UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Dept. of Communications

Study on the effect of application layer forward error correction in a 5G Broadcast transmitter

Master's Thesis

Master of Science in Telecommunication Technologies, Systems and Networks

AUTHOR: Montilla Vicent, Aarón

Tutor: Gómez Barquero, David

ACADEMIC YEAR: 2021/2022

Objectives - The main objective of the thesis is the design, development and validation of an AL-FEC method inside of a 5G Broadcast scenario. The deployment consists of a 5G Broadcast transmitter used for a file delivery application using FLUTE protocol. In addition, the implementation of the addition of the codes in a professional deployment made in the mark of a national pilot led by TelecomCLM has been studied. The trial was scheduled for the end of this year.

Prototype development — By an agreement with UPV, BitRipple provided RaptorQ codes libraries to be used in various project. The libraries have been included in the transmitter and receiver side of the implementations of the 5G Broadcast transmitter. To test the software, an SDR equipment based testbench has been developed.

Results — The measurements of this implementation give information about every part of the system individual performance and for the complete deployment. The performance of the code and the transmitter included tests regarding several attributes: memory management, transmission delays, executional duration…The results of the different tests are compared with the basic CNC mode of the transmitter, which does not include any type of FEC protection. The presented results proved the great performance of RaptorQ libraries and the programming mark developed in the thesis and its additional benefits comparing .with the basic deployment of the transmitter.

Future work — In the near future, the deployment of the RaptorQ codes implementation will be studied in several use cases. Among them the most important is the national pilot whose on-field trials are scheduled for the end of this year. Furthermore, the FLUTE transmitter is intended to support ROUTE protocol instead. The enhanced protocol will achieve real time transmissions protected by RaptorQ codes. Furthermore, 5G-MAG has several use cases targeted for 2023. Most of them are based in 5G media which would be benefited by the addition of an AL-FEC method in it as it has been proved in this thesis.

Abstract

In recent years, broadcast transmissions have become very important in mobile communications due to its great efficiency in the use of radio resources. However, the lack of a dedicated return channel to receive feedback about the best use of the transmissions to each of the users can make them not optimal, causing occasional packet loss. In this project, an application layer forward error correction method called RaptorQ will be included in the deployment of a 5G transmitter with broadcast capabilities. The RaptorQ encoder offers near-optimal capabilities allowing its performance with minimum overhead. This project consists in the implementation of the functions and compatibilities that will allow to include the codes in the implementation, as well as a subsequent study to observe the obtained improvements.

Resumen

En los últimos años, las transmisiones broadcast han tomado una gran importancia en las comunicaciones móviles dado la gran eficiencia que aportan en el uso de recursos radio. Sin embargo, la falta de un canal de retorno dedicado que permita saber cómo utilizar de la mejor forma las transmisiones a cada uno de los usuarios puede hacer que éstas no sean óptimas, ocasionando pérdidas ocasionales de paquetes. En este proyecto, se va a incluir en el despliegue de un transmisor 5G con capacidades broadcast, un método de corrección de errores en capa de aplicación llamado RaptorQ. El codificador RaptorQ ofrece unas capacidades cercanas a las óptimas, que permitirán llevar a cabo su función con una mínima sobrecarga. Este proyecto se basa en la implementación de las funciones y compatibilidades que permitirán incluir los códigos en la implementación, así como un posterior estudio que permita observar las mejoras obtenidas.

Resum

En els últims anys, les transmissions broadcast han obtingut una gran importància donada la gran eficiència que aporten en l'ús dels recursos ràdio. No obstant, la falta d'un canal de retorn dedicat que permeta rebre informació de tornada sobre com utilitzar les transmissions de la millor manera per cada usuari pot fer que aquestes no siguen òptimes, ocasionant pèrdues ocasionals de paquets. En aquest projecte, es va a incloure un mètode de correcció d'errors en la capa d'aplicació anomenat RaptorQ. El codificador RaptorQ ofereix unes capacitats properes a les òptimes, que permetran dur a terme les seues funcions amb una mínima sobrecàrrega. Aquest projecte es basa en la implementació de funcions i compatibilitats que permetran incloure els codis en la implementació, així com un posterior estudi de les millores obtingudes.

Autor: Aaron Montilla Vicent, email: aamonvi@iteam.upv.es
Director 1: David Gómez Barquero, email: dagobar@iteam.upv.es
Fecha de entrega: 02-09-22

Table of contents

Table of figures

1 INTRODUCTION AND MOTIVATION

The use of multimedia streaming services has suffered a massive increment in the last years. These services allow the end user to consume the desired content without the necessity of a previous download of the content. Nowadays, there is plenty of devices capable of receiving these contents in both static mode (as it happens at home) or in mobility (cars, trains, while walking...). According to Ericsson reports, this number will keep increasing in the following years.

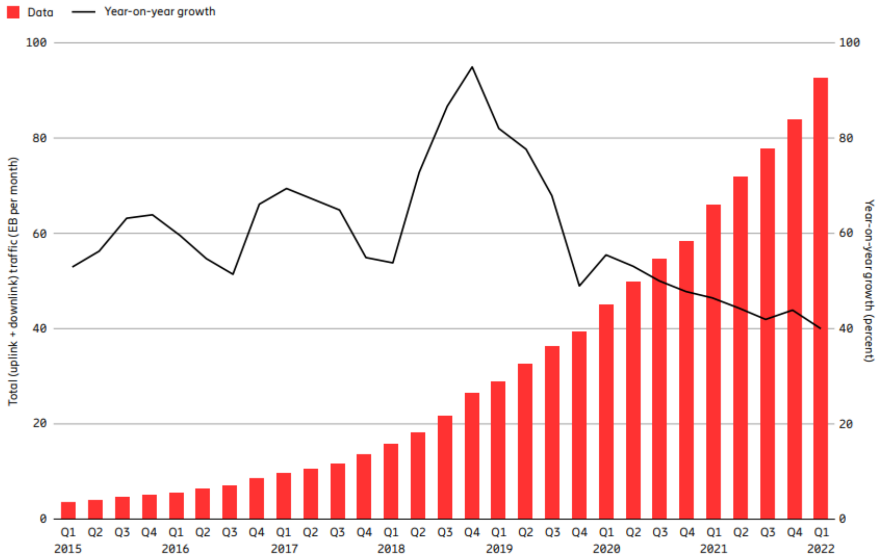

Fig. 1 Global mobile network data traffic and year-on-year growth [1]

In addition, the arrival of the fifth generation of mobile networks has opened a completely new world of possibilities with new use cases that were impossible to achieve before the update of the networks. Service providers continue to switch on to 5G increasing the number of commercial 5G deployments globally and augmenting the availability of these services. Therefore, the overall number of connected devices has increased and thus aggravating the global traffic issue. Ericsson reports predict to reach 4.4 billion of connected devices by 2027.

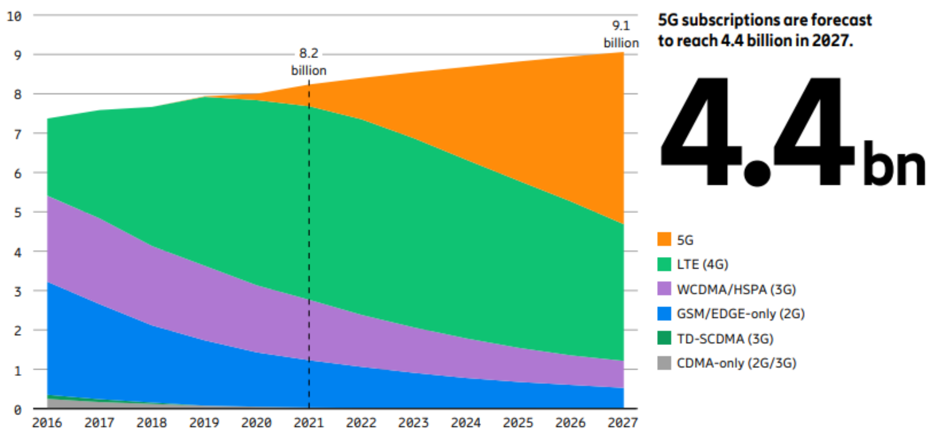

Fig. 2 Mobile subscriptions by technology (billion) [1]

Due to this fact, the increment of the bandwidth use has become incredibly challenging to manage with traditional transport methods. To overcome these problem, the exploitation of different delivery methods was studied. The known sending methods are:

- Unicast: By using this method the sender will need to send the packets individually to each end user that wants to receive the content. This method is very inefficient because it will flood the network with packets transporting the same information.
- Broadcast: This method will send the packets to every device connected to the network, independently if the users want to receive the content. This method is also inefficient in the use of resources because the devices will receive more packets than necessary and will waste some computational resources discarding them in case the user does not need the content.
- Multicast: This method achieves the most efficient use of the network resources, using the minimum required bandwidth as only the users that need the content are receiving the packets. The use of multicast transport in wireless communication will allow to send the same information to the users simultaneously using the same band, modulation and coding scheme.

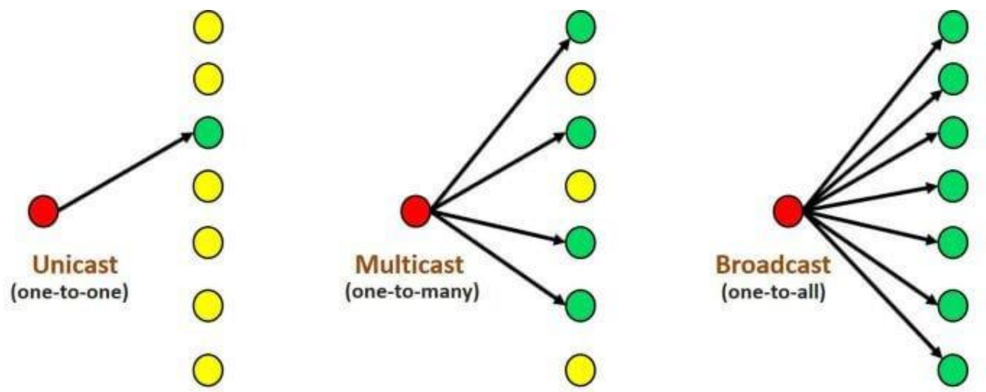

Fig. 3 Unicast, multicast and broadcast sending methods illustration [2]

The use of point to multipoint delivery methods can be beneficial for reducing the quantity of global traffic, as the information is only sent once while being received by many users and not repeated information is sent. Nevertheless, these delivery methods present additional drawbacks. Since it is not possible to have a feedback channel, it is impossible to perform the channel estimation and equalization to optimize each user's delivery. This results in a worse reception signal quality causing the occasional loss of packets. To overcome with the inconvenient several methods have been assessed, one of the most important and effective methods is the addition of forward error correction methods to repair the packets that were wrongly received.

This thesis will focus on the development of an application layer forward error correction method as a proposed solution. The used upper layer FEC codes are RaptorQ codes [3], the most modern and advanced version of Raptor codes. Usually, the use of AL-FEC generate a tradeoff with the additional protection increasing the time interleaving and the decrease of the system performance and increment of the latency. However, RaptorQ performance is near optimal, allowing the symbols to be recovered with very few extra symbols received and with minimum additional overhead. Thus, it includes minimum increment in the latency of the transmission. This software could be used in the thesis thanks to the collaboration agreement between UPV and BitRipple, owner of the RaptorQ libraries.

The RaptorQ codes implementation will be deployed and validated in 5G broadcast scenarios. LTE-based 5G Terrestrial Broadcast, widely known as 5G Broadcast, allows linear TV and radio to be broadcast to compatible 3GPP-based devices like smartphones, tablets, connected cars…As this service is designed and standardized by 3GPP it may be fully integrated into 3GPP equipment and complemented by conventional mobile broadband data. 5G Broadcast could be used to:

- Deliver public or commercial TV and radio services, free-to-air or encrypted to 3GPP compatible devices.
- Enable broadcast distribution of linear TV and radio services integrated into existing media applications with 3GPP-defined APIs.

5G Broadcast present several benefits in both the radio access and the service layer.

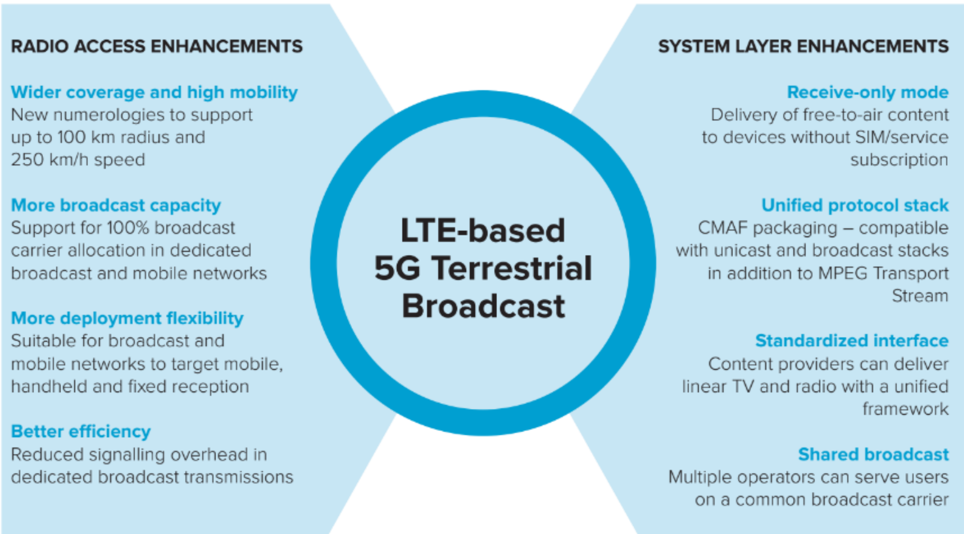

Fig. 4 LTE-based 5G Terrestrial Broadcast enhancements [4]

The architecture of a 5G Broadcast system is showed in the following figure. The main actors of the architecture are as follows:

- A 5G Broadcast TV/Radio Content Service Provider runs a head-end providing linear television and radio services.
- A 5G Broadcast TV/Radio Service application runs on devices that include a 5G Broadcast Receiver.
- A 5G Broadcast system operator runs a 5G Broadcast system with 5G Broadcast Transmitters for use by devices including 5G Broadcast Receivers.

- A 5G Broadcast TV/Radio Content Service Provider makes services available by using the 5G Broadcast system.
- A 5G Broadcast TV/Radio Service Application is able to consume the service by communicating with the 5G Broadcast Receivers via 5G Broadcast Client APIs.

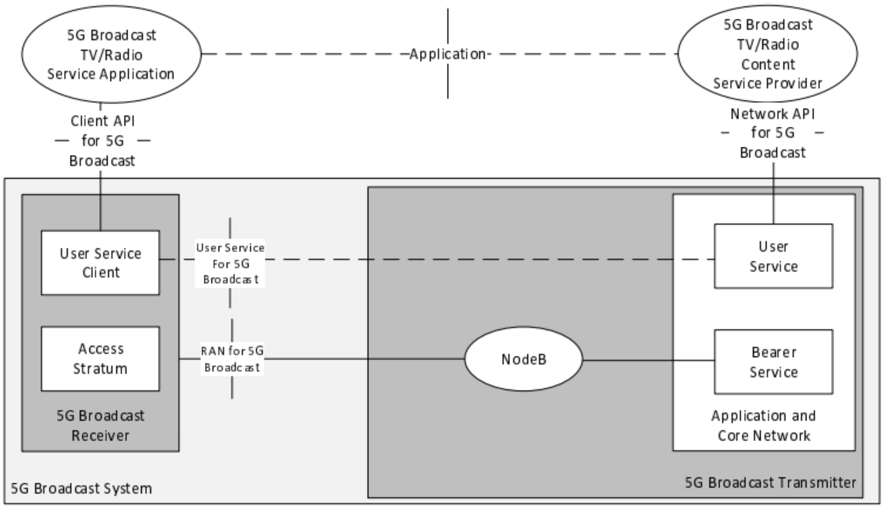

Fig. 5 Reference architecture for 5G Broadcast System [5]

This thesis focusses on two different implementations of a 5G Broadcast transceiver: the first one is an open-source implementation developed by 5G-MAG and the second one is being developed in the Red.es pilot.

1.1 5G-MAG

The deployment of the thesis includes many parts provided by the members of the 5G-MAG (5G Media Action Group) [6]. 5G-MAG is cross-industry, independent and not-for-profit association bridging the media and ICT (Information and communications technology) industries with its own legal funding, identity, governance and administrative rules. This association provides a framework to collaborate in the market-driven implementation of technologies in the digital content world. In particular, those related with the sector of 5G applied to content production, distribution and consumption. In addition, 5G-MAG is a 3GPP Market Representation Partner. By this collaboration, the association has the ability to offer market advice to 3GPP and to bring a consensus view of market within the scope of 3GPP.

In the end-to-end value chain of these associations, stakeholders of every sector are welcome no matter if they are content and service providers, network operators, technology solution suppliers, software developers, equipment manufacturers, R&D organizations, universities, regulators or policy makers.

The 5G Broadcast transceiver developed by 5G-MAG will be explained in later sections.

1.2 TelecomCLM pilot

The objective of this pilot is the development of an AL-FEC enhanced 5G Broadcast Transmitter and its deployment in Castilla La Mancha, concretely in Toledo [7].

To summarize, the idea of the pilot is the transmission and reception of professional TV and radio content using a HPHT (High Power High Tower) repeater, 5G Broadcast by SDR equipment and open-source mobile communication platforms. The pilot will be completely deployed in Toledo, not only the distribution part but also the creation, processing and emission of the multimedia content from the DVB-T (Digital Video Broadcasting Terrestrial) transmission center placed in Cerro de Palos. This trial is completely innovative since is the first 3GPP Release 16 compliant TV and radio content broadcast transmission.

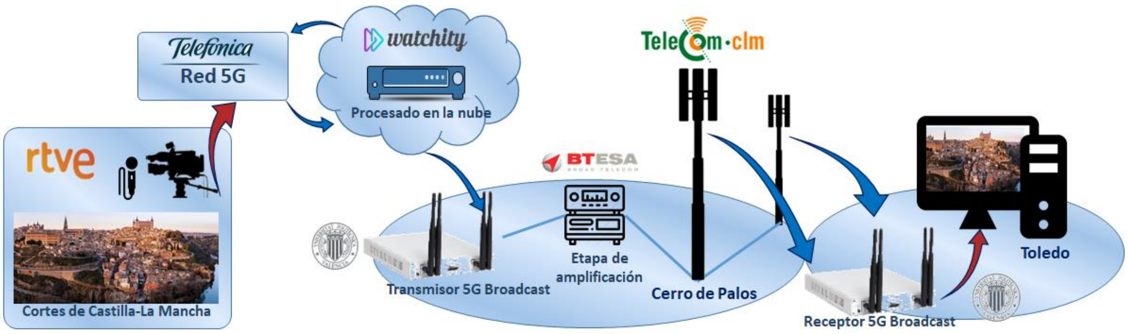

Fig. 6 TelecomCLM pilot architecture

This trial will be performed by a consortium formed by both private and public entities and will be divided into two groups, one group will be in charge of the production and processing of the content and the other group will be responsible of the transmission and reception of the content.

The first group is formed by (i) Radio Televisión Española (RTVE) who will provide the content to be distributed, (ii) Telefónica responsible of providing the 5G network to upload the content to the Cloud to be processed afterwards, (iii) Watchity in charge of providing the cloud server to make the real time signal production possible.

The second group is composed by: (i) TelecomCLM, it will provide the HPHT transmitter to distribute the content and coordinate the whole consortium, (ii) BTESA who will give the necessary professional equipment (amplifiers, RF re-emitters...), will make the coverage study in the whole area to optimize the power of the transmitter and will create the correct previous preamplifier stage of the HPHT transmitter, (iii) Instituto de Telecomunicaciones y Aplicaciones Multimedia (iTEAM) who will provide its previous experience in 3GPP SDR based implementations developing the 3GPP Release 16 compliant equipment.

On top of that, iTEAM will add the RaptorQ functions evaluated in this thesis to ensure the correct reception even in the most challenging transmission environments and achieving the correct reception in an even wider coverage area.

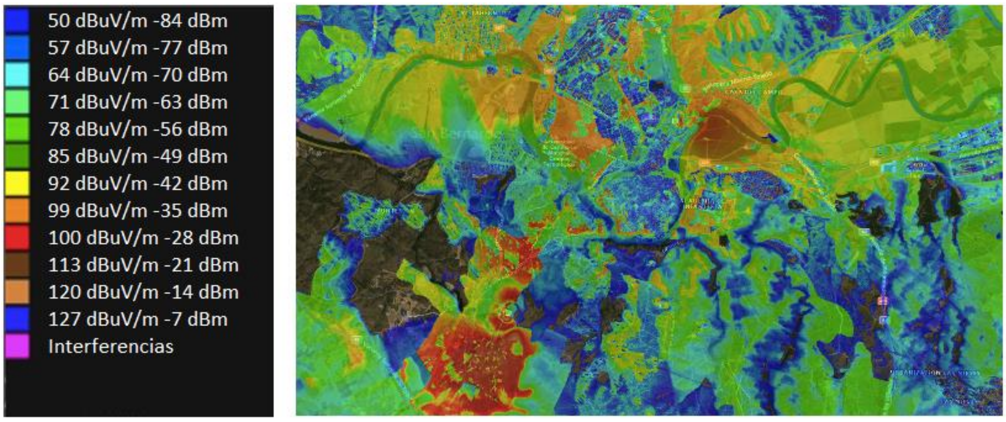

Fig. 7 Pilot coverage area

The study of the performance of this transmitter is out of the scope of the thesis.

2 OBJECTIVE

The objective of this thesis will focus on the improvement and performance study of the open source 5G Broadcast SDR transmitter currently being developed by 5G-MAG (5G Media Action Group) for the later deployment and implementation of the enhancements in the 5G Broadcast pilot of the national Red.Es 5G project in Castilla-La Mancha, more concretely in Toledo, led by TelecomCLM.

The results derived from this thesis will give a hint about what are the benefits of including an upper layer Forward Error Correction method in the implementation. The implementation will aim to achieve a better Quality of Service in the streaming and delivery of objects to a group of end users using point to multipoint transmissions. The better use of the network capabilities will achieve a better performance of the transceivers and in consequence a better quality of experience for the users. This thesis will study the crucial performance parameters of this version using laboratory trials to predict what performance can be expected from its deployment in a professional test case.

If the results of the tests are positive, this implementation will be used for the transmission and reception of a TV and radio channel property of Radio Televisión Española (RTVE) using 5G Broadcast.

3 STATE OF THE ART

3.1 5G Broadcast

The standards that 3GPP provides are organized by what is called releases. These are a set of several technical specifications, released every 15 to 18 months. Among the standards done by the 3GPP, there is a dedicated branch about point-to-multipoint communications to a group of end users inside of a certain area. This branch is named MBMS (Multimedia Broadcast Multicast Service) whose first appearance came in Release 6 in 2004. These capabilities could be used for the distribution of multimedia content. The performance of these capabilities was enhanced in the next generations and with the arrival of the fourth generation of mobile communications, 3GPP standardized in Release 9 a new version of the multicast standards named eMBMS (evolved MBMS). This feature, also known as LTE Broadcast, was released in Release 9, just after the initial LTE standard in Release 8. The capabilities of this feature were updated and increased in every release. However, until Release 13 it presented certain limitations that made the content creators to find it unappealing. These limitations were [8]:

- A maximum of 60% of radio resources assigned to broadcast and multicast purposes. Leaving the rest to control signaling and unicast transmissions.
- A limited coverage similar to 3G/4G, with maximums of few kilometers.

Finally, fifth generation of mobile communications was standardized by the 3GPP. By these means, the first release that brought 5G specifications was Release 15. In fact, when taken to practice every specification after Release 15 is named "5G", even LTE systems with Rel 15 enhancements. For that reason, the LTE systems, whose multicast capabilities were considered to be incomplete, that were further enhanced after Release 15 are named LTE-based 5G Broadcast.

4 APPLICATION LAYER FEC FOR 5G BROADCAST

4.1 *FEC*

FEC mechanisms rely on the transmission of repair information to protect loss events on underlying levels without a need for feedback (return channel), such that the receiver can detect, and possibly correct errors occurred during the transmission [9]. The error correction capability of a FEC code depends on the error distribution over time and in the robustness of the FEC code. Large burst of errors can make the decoding of the information even more challenging depending on the used code. This type of burst is especially usual in wireless communications, caused by signal fading due to multipath propagation, the missed synchronization in the radio channel or shadowing periods inside of the service area. In addition, channel noise may help to produce these error patterns. Furthermore, in some cases the transmission may be completely interrupted causing the missed reception of all the packets sent in this period.

The performance degradation due to bursts of errors can be mitigated by the addition of time interleaving to distribute the coded data over time so these errors are more uniformly distributed after performing the de-interleaving and thus would be easier for the FEC codes to correct the errors. Furthermore, the channel noise would be less variable when averaged over longer time periods causing lower amount of protection needed for larger interleaving.

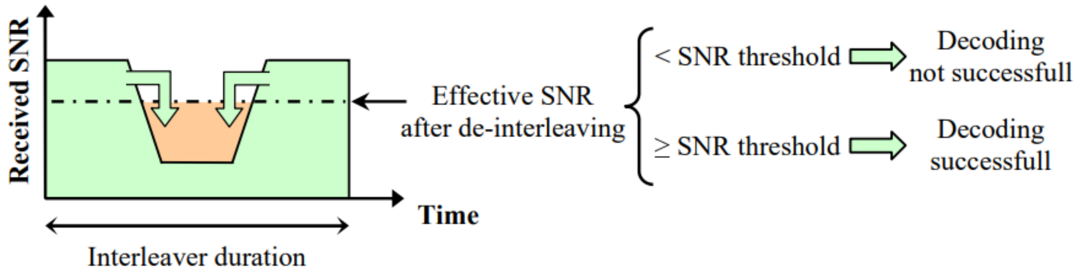

Fig. 8 Time interleaving effect in overall SNR

Averaging good and bad reception conditions, it is possible to correctly decode the information, as soon as the average SNR surpass the minimum threshold.

4.2 *Physical layer FEC vs upper layer FEC*

FEC mechanisms can be divided in two groups, the physical layer FEC codes and any FEC code working in any upper layer above it.

Physical layer FEC codes work at the bit level and are traditionally implemented as part of the radio interface for wireless communication systems. Theoretically, they have the attributes to offer the most effective protection against channel noise. However, due to practical limitations the maximum interleaving depth is rather small. The memory requirement for FEC decoding is proportional to the transmission data rate, interleaving duration and the rate of parity data transmitted, which is actually large at the physical layer. Therefore, physical layer FEC codes are usually combined with an upper-layer FEC code achieving a better tradeoff between overall system error protection and system implementation.

On the other hand, an upper layer FEC code is any code working on link layer or application layer. By operating above the physical layer, it is possible to provide larger interleaving depths, that physical layer cannot support. Contrasting with the physical layer FEC codes, which work at bit level, these codes recover from packet losses, which means fixed-size bit blocks.

The studied transmission chain in this thesis has several attributes that can lead to a deficient performance of the system. On one hand, the use of multicast transmissions without feedback channels makes impossible the estimation of the channel for each user. On the other hand, this project will use SDR (Software Defined Radio) devices in the transmission chain, whose performance has had an increasing interest in the industry and has been improved in the last years, but it is still far from being optimal. Both of these drawbacks could cause the lost or the wrong decoding of some of the sent packets leading to the necessity of the addition of a new level of protection inside of the transmission chain. To meet that end, this thesis will focus on the addition of an upper layer FEC code, concretely an AL-FEC (Application Layer Forward Error Correction). This feature includes protection in the distributed content to favor the correct decoding in the receiver without the need of interaction between sender and receiver.

4.3 Raptor codes

The standardized AL-FEC method in 3GPP is Raptor codes [10], developed by Amin Shokrollahi in 2000 in extension to LT codes (Luby transform) with linear encoding and decoding cost. Raptor codes are a computationally efficient implementation of fountain codes that achieve close to ideal performance. One of the biggest benefits of Raptor codes is the capability to be fully implemented in software without the need of dedicated hardware. The following image shows the case of an ideal and systematic fountain code transmission with limitless repairing data:

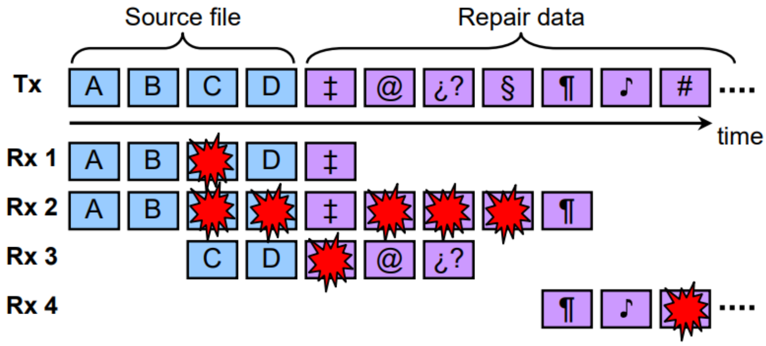

Fig. 9 Fountain code transmission over broadcast channel

To describe the functioning of fountain codes an example will be explained in this section. Suppose to have a block of data hereafter called a source block that are to be reliably transmitted. This block of data will be partitioned into parts of equal size which will me denominated source symbols that are supposed to be fitted into a packet. Let k be the number of source symbols in a source block. The code will generate $n \geq k$ encoding symbols out of the original k symbols, the relation k/n is called coding rate. The difference $n - k$ is called repairing symbols or more generally FEC overhead. Raptor codes belong to a group called systematic codes in which the first k symbols of the encoded symbols are the original data. If the code is non-systematic the encoding symbols do not contain the original symbols and different ones are generated.

In the raptor code situation, as they are systematic, if no packets are lost then the parity data will not be needed at all. Under poor receiving conditions, some of the packets will not be received. Once this occurs the decoder will be able to decode the original information with any subset of received symbols slightly bigger than the original, ideally of equal size. That is where the “fountain” term comes from: someone who wants to fill a glass of water under a regular fountain does not care about the particular drops filling the glass; instead, only the amount of water filling the glass matters. Similarly, with a fountain code the particular packets received do not matter; only the number of received packets matters. Each additional packet is beneficial for reconstruction of the original content and no receiver receives useless information. Furthermore, fountain codes can potentially create unlimited parity data, a quality called “rateless”.

On the receiver side, the increment in temporal cost depends on the transmission. Under fountain codes conditions, once the receiver has enough symbols to perform the decoding the original data can be rearranged, which in the ideal case will be after receiving k symbols. However, under poor receiving conditions, the receiver loses some packets, and it will need to wait for the following packets to arrive, increasing the transmission duration.

To perform Raptor codes enhancements over the original LT codes, a high-rate binary block code is applied before the LT-code. These new symbols will be called intermediate symbols. Then a suitable LT code will be applied to the intermediate symbol creating the final encoding symbols. After the first decoding process, some LT code symbols could not be recovered, nevertheless with an appropriate chose of the precode an erasure decoding algorithm can be applied to the precode recovering the lost symbols. As a consequence, in most practical settings Raptor codes outperform LT-codes in terms of efficiency, range of source block sizes over which it is effective, smaller reception overhead and lower failure probability.

4.4 *RaptorQ*

As it was stated in the introduction section, RaptorQ, the last and most improved version of raptor codes will be the codes used in this thesis. This version of the codes achieves near optimal capabilities. RaptorQ libraries offer the following key properties [11]:

- Exceptionally fast encoding and decoding – linear-time encoding and decoding, enabling deployment in even the most CPU constrained environments.
- Exceptional lost recovery properties, independently of which data is received and which data is lost.
- Flexibility to operate in a wide range of possible number of blocks, number of sub-blocks and as much encoded data as necessary.

The recovery properties of the RaptorQ decoder are exceptional. As an example, if there are k source symbols in a source block, then the RaptorQ decoder can recover the source block with probability greater than:

- 99% from reception of k encoded symbols.
- 99.99% from reception of k+1 encoded symbols.
- 99.9999% from reception of k+2 encoded symbols.

Furthermore, this version of the codes achieves the maximum flexibility in the creation of repair symbols and in the efficiency of the symbol recovery.

5 REFERENCE TOOLS IMPLEMENTATIONS

5G-MAG Reference Tools is an. activity sponsored and fostered by 5G-MAG to establish a developer community and at creating common open-source reference tools to support the implementation and interoperability of 5G Media technologies.

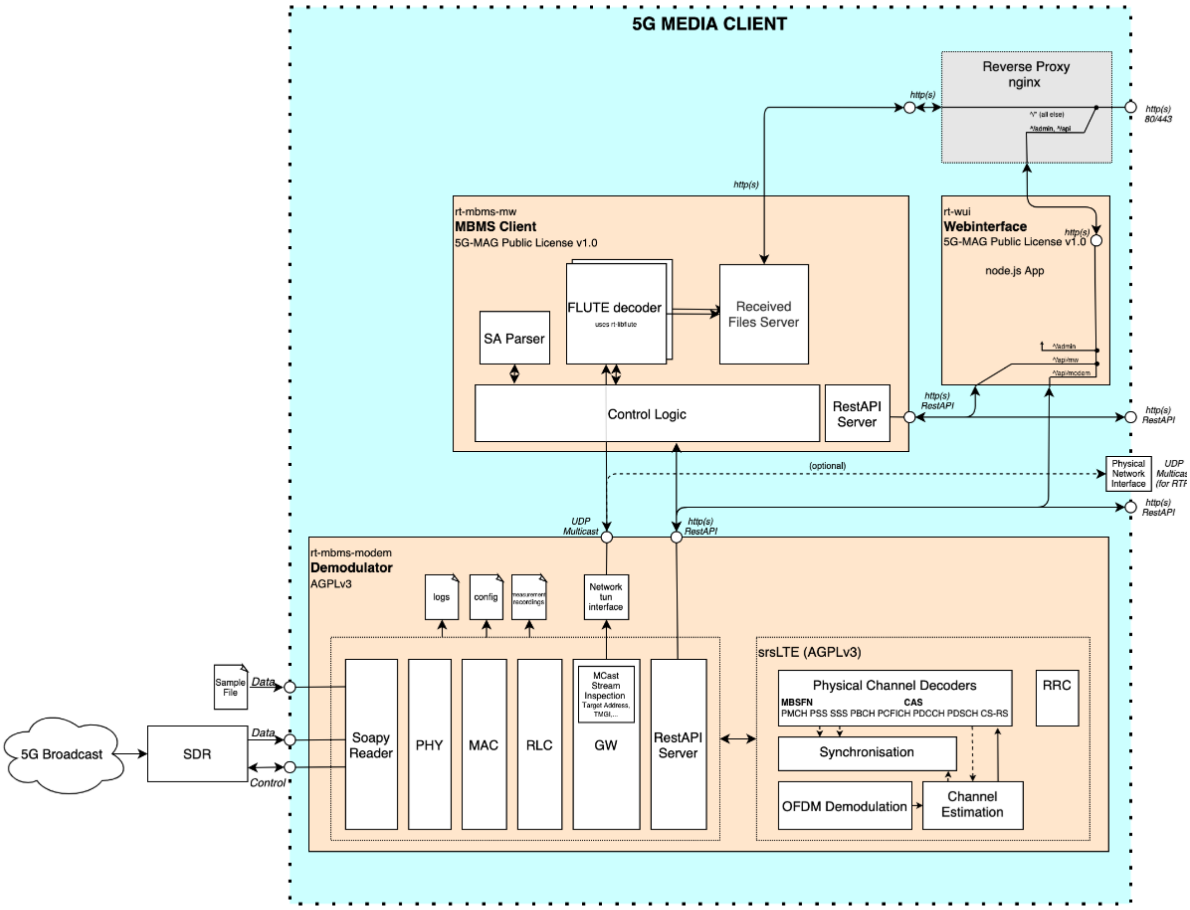

Fig. 10 Reference tools implemented system architecture

The system implementation of this thesis used the rt-mbms-modem and the FLUTE transceiver parts developed by Reference Tools.

5.1 *MBMS modem*

The MBMS Modem [12] builds the lower part of the 5G-MAG Reference Tools. Its main task is to convert a 5G BC input signal (received as I/Q raw data from the SDR) to Multicast IP packets on the output.

The main components of the MBMS Modem are implemented as modules for a better overview and to easier improve parts later:

- Reception of I/Q data from the SDR equipment
- PHY: synchronization, OFDM demodulation, channel estimation, decoding of the physical control and user data channels
- MAC: evaluation of DCI, CFI, SIB and MIB. Decoding of MCCH and MTCH
- RLC / GW: Receipt of MTCH data, output on tun network interface
- Rest API Server: provides an HTTP server for the RESTful API
- Logging of status messages via syslog

This component will be installed in the receiver VM and will be in charge of forwarding the created multicast packets to the application listening in the multicast destination IP address. In this case, the application will be the FLUTE transceiver developed by 5G MAG.

5.2 FLUTE transmitter

FLUTE [13] (File Delivery over Unidirectional Transport) is a protocol intended for the unidirectional file delivery over the Internet, especially in multicast use cases. This protocol builds on top of Asynchronous Layered Coding (ALC), a protocol designed for massively scalable multicast distribution. However, Flute protocol is designed to transport files, and in these use cases, basic object transport is insufficient, as the receivers need to know what the object represent.

To this end, FLUTE is applicable for the transport of files of every dimension. Its mechanism creates delivery sessions where the sender and receivers will share information that permits the receiver to map these parameters to correctly process the incoming data to get the file successfully. FLUTE protocol achieves massive scalability combined with an exceptionally reliable transport without requiring connectivity from the receiver to the sender.

The FLUTE protocol functioning works with File delivery sessions to which the potential receivers will connect to receive the data sent through. The sessions will be identified using both the source IP of the transmitter and a Transport session identifier (TSI). This is the information the receiver will map to know to which session an incoming packet belongs to. Additionally, each session can carry multiple objects. For that reason, the packets will also carry a Transmission Object Identifier (TOI) to distinguish the packets for each object.

In each session, there will be also File Delivery Tables (FDT) sent among the Object packets. FDT are used to let the receivers know about the attributes of the files that are distributed within the session. Some of the attributes related with the FDT are:

- TOI value that represents the file
- FEC object transmission information
- Size of the transmission object carrying the files
- Name, Identification and location of file (specified using an URI)
- Size of the file

All these information will be sent inside of the table following an XML format. The packets transporting the FDT will be recognized because are marked with the TOI 0. FDTs are recommended to be sent before the sending of the first packet of the object, even so it is compulsory to send the FDT periodically so receivers that connect to the session once it is started still can get the attributes of the file achieving the successful reception.

The implementation of a FLUTE transmitter developed by 5G-MAG programmers [14] follows all the specification of the RFC document and is the base of the implementation done in this thesis. This transmitter only had implemented a mode with no coding of the information, which was named as Compact No Code (CNC). In the result section, the performance of this mode is compared to the new and enhanced regarding protection RaptorQ mode. Notice that CNC code will be more efficient, as it does not require an extra processing of the receive data apart from the writing of the data on the file with the details received in the File Delivery Table. This could cause that the efficiency of the Compact No Code mode of the FLUTE transceiver will have a slightly better performance than the RaptorQ mode of the system without protection.

6 DESIGN AND DEVELOPMENT OF THE SYSTEM

6.1 *Algorithm implementation*

By the collaboration between both entities, UPV got rights to use the software from BitRipple in various projects. By this means, BitRipple provided RQ libraries, capable of performing the necessary actions to use RaptorQ encoding and decoding. The libraries implement functions with huge granularity, which is particularly useful to be able to modify every detail of the functioning, but also need to be well configured and need to have a well-programmed mark to work properly.

Considering this information, two bigger functions called “EncodeRQ” and “DecodeRQ” were created in the mark of this thesis. These functions contain every RQ library function that was needed to perform the encoding and decoding of the data chunks in both ends. To be able to take on these actions the functions need several input data:

- Memory location with the input data.
- Quantity of data to be processed.
- Input ESIs.
- Output ESIs.
- Memory location where to store the output data.

On one hand, the encoding function takes the data from the chunk of memory where the file is stored, it processes the input ESIs of the symbols in which the data will be partitioned (in the encoding case it will just be a vector with consecutive numbers from 0 to the number of source symbols), it computes the quantity of data that needs to be allocated for each process, it creates the intermediate and output symbols from the initial data and finally it will place each output ESI in each output symbol, returning the output data to the memory location specified in the function input to be prepared to be sent. Note that the output ESIs will also be a consecutive vector of numbers from 0 to the addition of the number of source symbols and the number of repairing symbols.

On the other hand, the decoding function has an extra complexity in the ESIs handling. It will receive the same input parameters but the content of some of them will be different. Upon receiving a symbol, the receiver will process the information of these symbols. The content of the header of the packets containing the symbols will be stored in the input ESIs vector that is passed to the decoding function. It is extremely important to notice that in this case the vector will not contain a consecutive number list, since the lost symbols identifiers will be missing in the vector. In addition, the symbols will be consecutively stored in a chunk of memory previously prepared for that end. The code of the receiver is constantly checking how much information has been received and if it is sufficient to decode the data. Once enough symbols are stored in the memory chunk, the memory location of the chunk and the vector of symbol identifiers will be passed to the function. The algorithm will then decode the data and will return the original source symbols to the receiver to be further processed. When the receiver does not receive enough symbols, the decoding function will not be activated and the symbols with an identifier lower than the number of source symbols will be placed in order just filling the gaps of the lost symbols with zeros and notifying the error. Additionally, some enhancements, such as the drop of the extra symbols if enough symbols of a block were received, were include in the code so the extra processing of the code was minimized. The same process will be executed for every source block of the file. Once all the blocks are marked as completed the original file can be restored. As an additional feature and for testing purposes, a method to force symbol loss was implemented inside of the code. This method made possible several test making possible the simulation of the loss of the desired number of symbols. This method will be further explained in the results section.

However, one may note this algorithm is not enough with the presented information. For that reason, additional enhancements and modifications have been added to the transmitter and receiver code to give the extra support that the encoding method will need. The first addition was to include RaptorQ headers as stated in the standard to include the compulsory information of the delivery. The following graphs show the different header structure and his members:

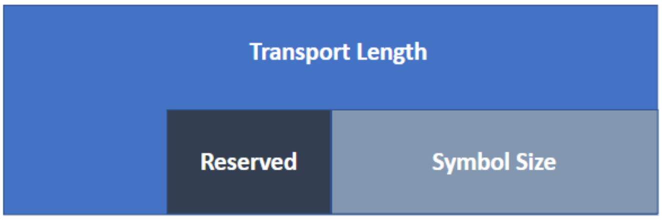

Fig. 11 Encoded Common FEC OTI for RaptorQ FEC Scheme

This header is common for the whole object that is being transported. This header is formed by 40 bits representing the transport length of the object, 8 bits reserved for its use in future releases of the standard and 16 bits to represent the size of the symbols being transmitted.

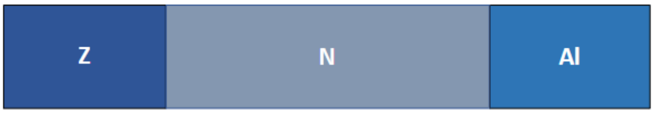

Fig. 12 Encoded Scheme-Specific FEC Object Transmission Information

The Scheme-Specific FEC Object transmission information is formed by the Z parameter representing the number of source blocks, N the number of sub-blocks and Al the alignment parameter. This header is represented by 4 octets distributed as 1 octet for the Z parameter, 2 octets for N and 1 octet for the alignment parameter.

The concatenation of the encoded Common FEC Object Transmission Information and the encoded Scheme-specific FEC Object Transmission Information forms the encoded FEC Object Transmission Information.

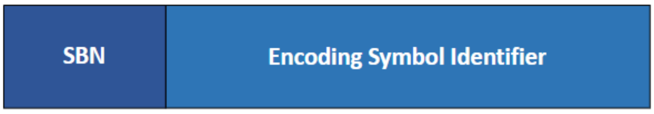

Fig. 13 Encoded Common FEC OTI for RaptorQ FEC Scheme

This header is common for all the symbols inside of a block. This header is formed by 8 bits representing the number of the block, which the symbol belongs to, and 24 bits to indicate which symbol is being carried. With this information the receiver will be able to process the received symbols correctly and thus to decode the original information.

6.2 *Setup of the transceiver*

This section will give details about the testbench mounted in UPV's Campus 5G lab to take on the tests regarding both hardware and software. The software that is being used in the setup is installed inside of 2 virtual machines, both having installed Ubuntu 20.04 version. This version is compulsory for both the 5G-MAG MBMS modem and FLUTE transmitter. To send the information via air interface, this system uses two SDR devices connected to each VM so the information can be sent and received. The devices are USRP (Universal Software Radio Peripheral), commercial product by Ettus Research [15]. Concretely the devices are a N310 model [16].

Fig. 14 USRP model N310

Thanks to the flexibility provided by these devices, the connection between both devices can be done through air interface using antennas or through a connected connection using a cable. For testing purposes both deployments have been tested.

6.2.1 *Transceiver software installation*

In the previous section, the hardware components, which make part of the system, have been explained. However, these components do not work by themselves, they need to be configured and used using a specific software that allows the correct functioning of the hardware. This section will summarize the different steps that need to be done to have the components ready and working.

The first step is the installation of the UHD repository inside of both virtual machines. The UHD repository is a free and open-source software driver and API for the Universal Software Radio Peripheral (USRP). This software is also distributed by Ettus Research. Although, this equipment was not among the options previously evaluated by the 5G-MAG and its installation was not included in the tutorials, it could be configured to be used in the system by adding few lines to the configuration files. In addition, the products from Ettus Research have shown good performance and results, as it will be presented in the following sections. To get the software installed just one command is needed:

$ sudo apt-get install libuhd-dev libuhd3.15.0 uhd-host

After finishing the installation, the devices must be detected by the PC and its information should be shown once the *uhd_find_devices* command is applied.

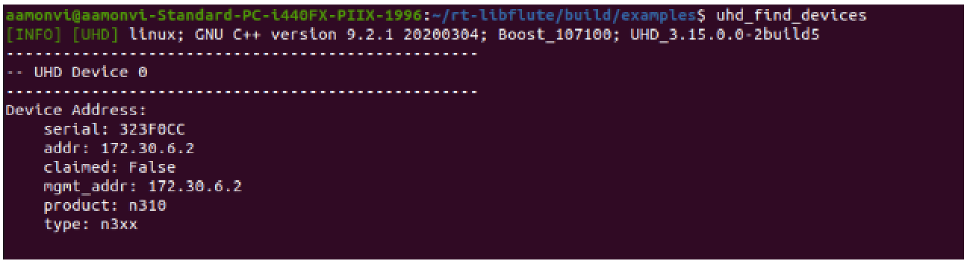

Fig. 15 Uhd_find_devices command output in the VM

Once the device is detected, it is ready to be used by any other software installed in the virtual machine.

The following step was the installation 5G broadcast transmission system. To get the system ready several actions were needed in both the transmitter and receiver virtual machines. The transmitter side implementation is based on the srsRAN software [17]. srsRAN is a 4G/5G software radio suite developed by SRS. The implementation is divided in three parts:

- SrsUE: a full-stack SDR 4G/5G-NSA UE application.
- SrsENB: a full-stack SDR 4G/5G-NSA eNodeB application.
- SrsEPC: a lightweight 4G core network implementation with MME, HSS and S/P-GW.

However, this thesis only uses the eNodeB and the EPC part of the software. The steps to activate the transmitter correctly are:

- Build srsRAN at the latest commit on branch fembms-tx and install.
- Run sudo srsepc in one shell.
- Run sudo srsmbs in another shell.
- Set up the multicast routing from the Ethernet interface to the sgi_mb tunnel.
- Configure sib.conf.mbsfn and enb.conf files considering our system parameters.
- Run sudo srsenb in yet another shell.

At this point, the information the user sends to the multicast IP address, which in this thesis was 239.255.1.1, will be coded and transmitted by the srsRAN software using the configured hardware, which in this setup will be the USRPs.

On the receiver side, the 5G-MAG MBMS modem was installed to convert the incoming data received by the USRPs into multicast packets and will forward to the same multicast IP that was configured in the transmitter (239.255.1.1). The MBMS modem installation tutorial is included in the reference URL.

On top of that, the FLUTE transceiver will be working. The transmitter side of the FLUTE implementation will send the packets to the multicast IP address. The packets will be forwarded through the transmission chain wo be finally received in the same multicast IP address, where the receiver will be listening. Inside of both codes, RaptorQ encoder and decoder will be added to include the parity information. The complete setup for the trials is shown in the following graph:

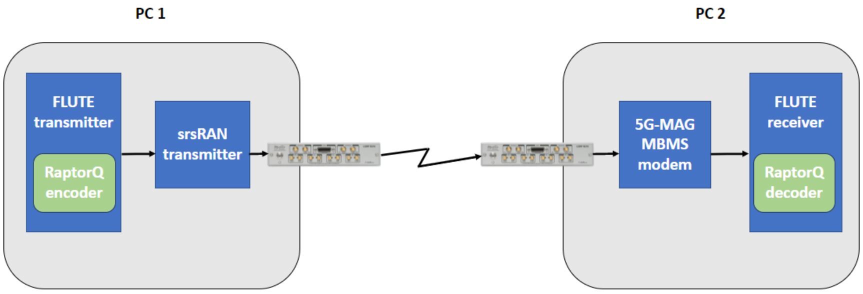

Fig. 16 Thesis laboratory trial setup

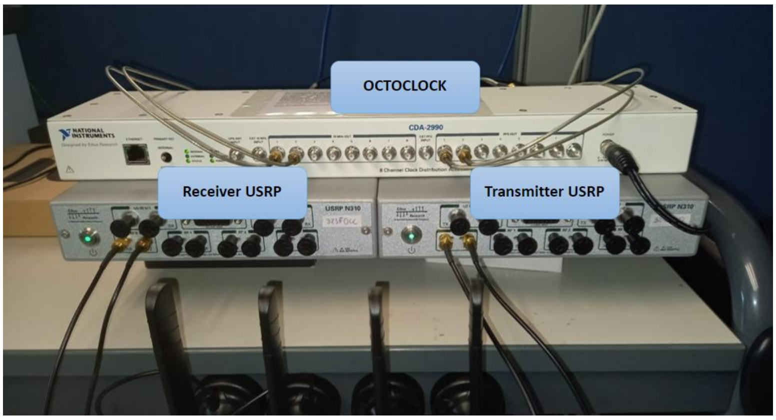

Fig. 17 SDR deployment used in the thesis

As it can be seen in the figure 17, the SDR devices are connected to an octoclock, which will provide the correct synchronization between both devices. This element is extremely necessary in the system, since the lack of synchronization between the USRPs will result in the wrong reception of the information.

7 RESULTS

7.1 Initial tests

As a first trial and remarking that the setup of these thesis is completely experimental and SDR based, it was necessary to assess what is the actual performance of the different addition of code that has been included to perform RaptorQ related actions. For that reason, this section starts working with a simplified set up with both of the virtual machines directly connected through the network. In this transmission chain neither the srsRAN nor the MBMS modem has been included. The complete set up will be used in the following sections and the derived limitations will be explained.

The connection between both VM has been tested using Iperf to obtain what is the maximum bandwidth between both sides. The obtained result showed a maximum bandwidth of 125 Mbps, so the test performed in this part of the section should remain under this threshold.

7.1.1 Speed test

7.1.1.1 Algorithm speed

As it was introduced in previous sections, the enhancements are focused in two functions: EncodeRQ and DecodeRQ which will encode and decode the data in the transmitter and in the receiver respectively. To get these results, the following test have been based on some measurements about the memory usage and the execution time of the actions that the RaptorQ libraries and the external functions perform. These results have been compared to the basic mode of the transceiver to study the tradeoff between protection and execution speed.

The speed test of the new algorithm has been studied in both the transmitter and receiver separately. In the transmitter side, the calculated time represent the time the transmitter lasts to process the input file, generate the partitioned file, calculate the different memory chunks the process will need to store the information, generate the repairing symbols and finally prepares the symbols to be sent. This means that not only RaptorQ related processes have been considered inside of this measurement and in consequence, the execution time of the basic mode of the transmitter need to be calculated and compare to study the actual increment in time of the RaptorQ enhancements.

This thesis has used the “chrono” library [18] to get this information. This library provides many tools to calculate the duration of a class or a program. In this case, the steady_clock::time point structure has been used. The object needs two different instances at the beginning and end points of the functions whose execution time the user wants to be computed. After the execution, the last step is to subtract the end and initial time point and cast with "duration_cast" to get the time interval. This method computes the execution time of a process with high precision, however, the time that the virtual machine takes to get an action done depends on external processes of the same virtual machine. Thus, the computed execution time will vary in each of the tests. Considering this information, the time has been calculated with the mean of ten different trials to limit the effect of the variance. The next graph shows how the results vary in the different trials:

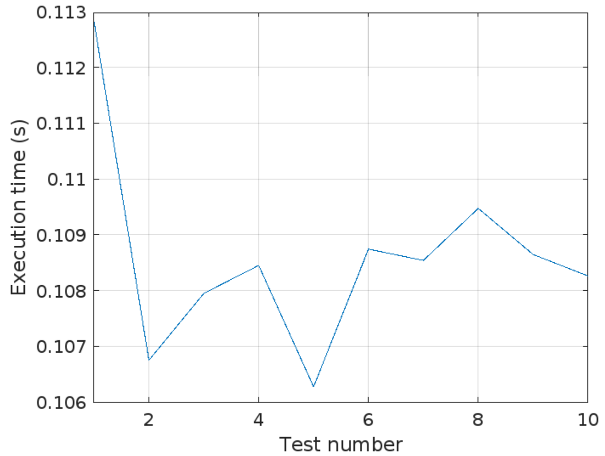

Fig. 18 Execution time variation in the different trials

As it can be seen in the graph, the results vary one from each other in the order of 5 milliseconds. Considering the mean value as the effective result will give the study of the thesis extra reliability.

Once the measurement method has been explained, it is time to extract information about what is the extra execution time that the RaptorQ transmitter adds to the original performance and how the protection percentage affects to this. In this trial, the clock instances have been placed in the code as it was explained previously, and the transmitter has been activated with a file of 10 MB varying the coding rate of the transmission. The following graphs shows the obtained information:

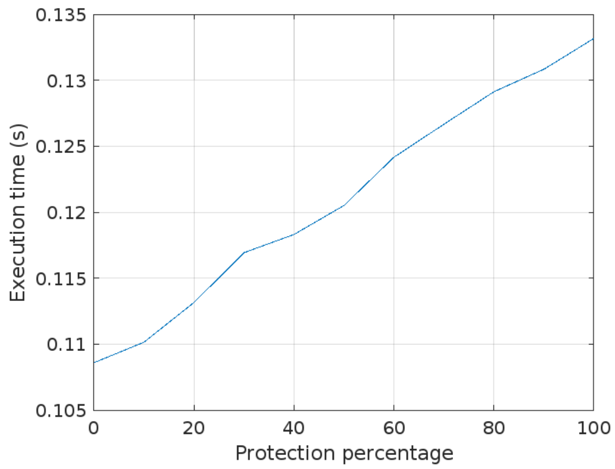

Fig. 19 EncodeRQ execution time vs protection percentage in a 10 MB file encoding

The first detail worth to be emphasized is the linear progression of the graph. This means that the execution time of the algorithm increases linearly with the amount of information that needs to be processed and in consequence, the number of repairing symbols that need to be created. To extract the real time that it lasts to create the RaptorQ repairing symbols additional measurements have been added to compute the execution time of the "EncodeRQ" function itself and the execution time of the basic CNC mode of the transmitter. The result of the first measurement showed that the encode function lasts around 0,102 seconds to be completed, this constitutes around 95 % of the total execution time. On the other hand, the execution time computed in the CNC was around 0,006 seconds. This information shows that the algorithm created for the RaptorQ mode of the transmitter only adds the time that RaptorQ libraries take to perform the calculation of the necessary memory chunks sizes, the creation of the intermediate symbols and the creation of the repairing symbols.

In order to get additional information about all the aspects that have been studied in the previous paragraph, the test have been redone using this time a file of 2 MB. The following graph shows the results obtained in the trial:

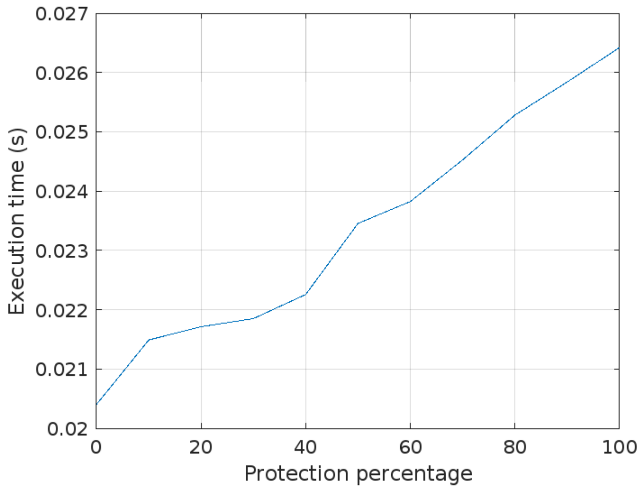

Fig. 20 EncodeRQ execution time vs protection percentage in a 2 MB file encoding

The graph shows how the obtained measurements, extracted again from the mean of 10 different measurements, do not present the same linearity as the previous graph. Nevertheless, this makes sense, as the variations are bigger in comparison to the mean value. Furthermore, the mean value is still increasingly progressive with the increment of the protection in the transmission.

Following with the guidelines of the previous trial, the execution time of the "EncodeRQ" function itself and the function of the CNC mode of the transmitter have been calculated. The result of the function duration was around 0.0185 seconds, again above of 90% of the total execution time, concretely 92.5%. On the other hand, the calculation done in the CNC mode give an execution time of 0.0012 seconds. Although it is true that the total time is bigger than the addition of the CNC execution time and the duration of the RaptorQ function, the difference is insignificant and may be due to the variation in the different calculation, confirming that the algorithm does not present extra overhead apart from RaptorQ libraries addition, which is minimal. It is important to remark that this process occurs before the sending of the file. For that reason, the throughput of the transmitter does not have influence in the measured values.

In the reception side the speed of the different processes has been calculated similarly to the transmitter side, using the “chrono” library. In this part of the study, the measurements will be about the DecodeRQ function and in the total reception time since the first symbol arrives until the file is correctly decoded.

For the first part of the test, the 2 MB file has been chosen to conduct the experiment. In this process, the sending is activated without any simulated lost packet. Once the last symbol has been received in the receiver side the clock start counting and the count is stopped once the original symbols are correctly placed in the final file. As it was done in the transmitter experiment, the protection percentage has been modified in each of the trials. The following graph shows the extracted results:

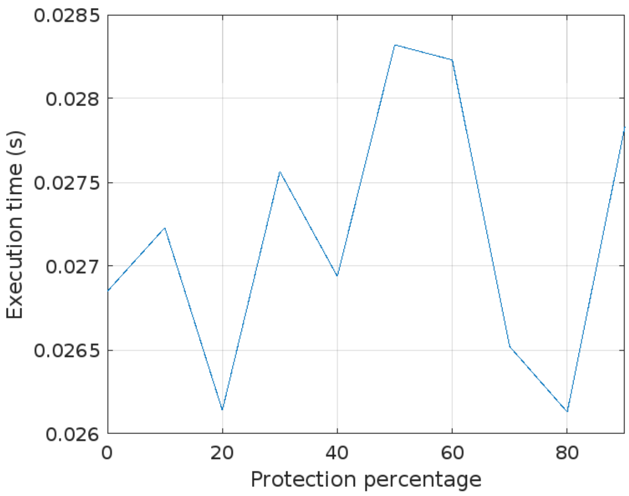

Fig. 21 DecodeRQ execution time for each coding rate

This representation confirms that the coding rate applied to the transmission and the execution time of the function has no correlation. This aspect is reasonable as the receiver is discarding the extra symbols, the unnecessary and redundant information. In this way, the algorithm achieves a fixed decoding time, setting aside the variation from the different trials. This test has huge importance in this study, showing how the coding rate does not compromise the satisfactory performance of the transmission chain in the decoding time nor in the computing workload. These results concur with the information explained in the FEC section, showing how when the last necessary symbol is received, the decoding process can start adding minimum overhead.

7.1.1.2 Reception duration

Many parameters are involved in the results of the total reception time: the rate limiter of the transmitter, the amount of loss packets and of top of that the own variation of the VM speed. The timer used in this test started with the arrival of the first symbol to the receiver and stopped once the file was correctly decoded. At first, no packet loss was introduced in the transmission to compare the performance of both modes in equal conditions. The following graphs show how the reception time increase depending on the channel bandwidth.

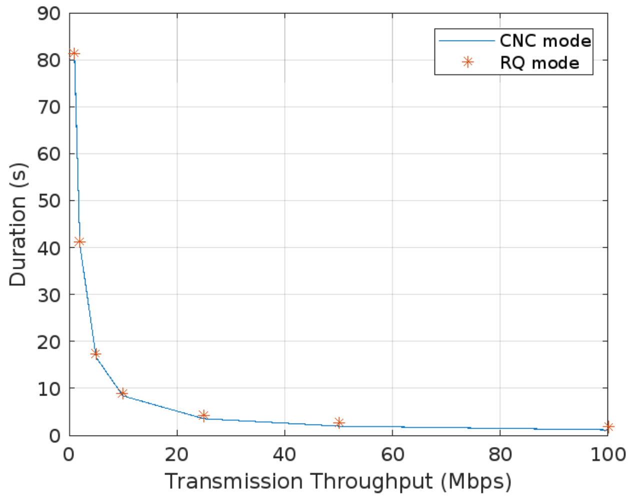

Fig. 22 Reception duration graph for 10 MB file transmission

The figure shows how little the difference between both modes' performance is. In fact, for bigger transmission latencies the relative difference decrease. This is because the main difference in time is the RaptorQ decoding whose duration does not depend on the transmission throughput as it was seen in the previous tests.

The difference is even less noticeable in a 2 MB transmission file.

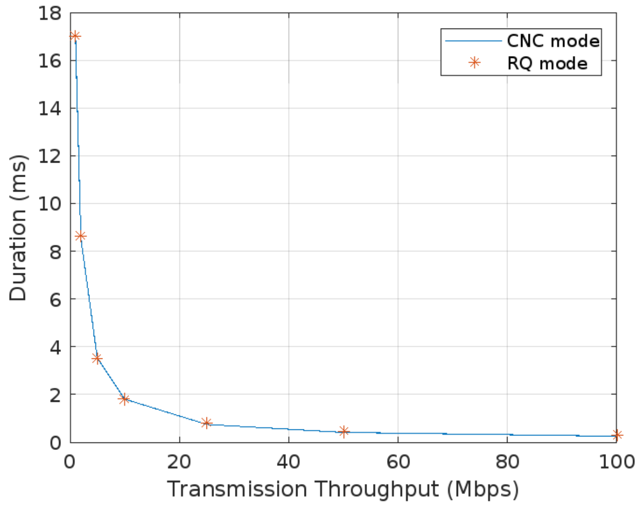

Fig. 23 Reception duration graph for 2 MB file transmission

As the number of symbols is smaller, the decoding time also decreases. At this point, it is not possible to notice the difference between both modes' performance proving RaptorQ's minimum overhead that was stated in the theory.

Nevertheless, the packet loss will increase the system latency since the receiver needs to wait to the additional repairing symbols to be able to decode the block. To assess how this could affect the overall system performance the increment in time caused for different number of lost packets have been studied.

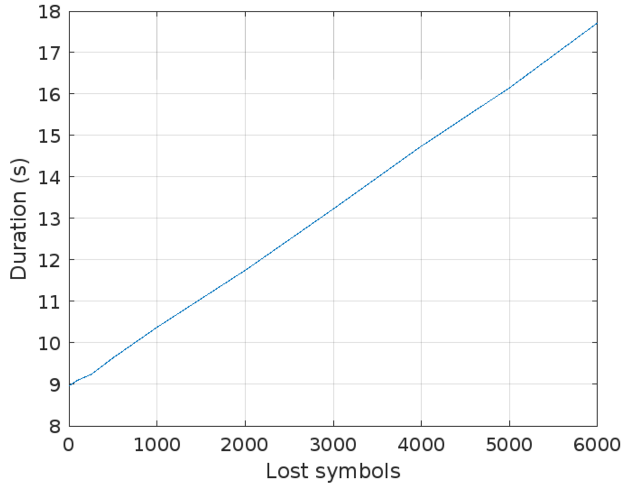

Fig. 24 Reception duration vs number of lost symbols in 10 MB transmission

This graph shows the linearity in duration that the code present. The file of 10 MB is divided into 6800 symbols by the encoder thus, it is reasonable that the case of 6000 lost packets is slightly smaller than the double duration of the test case where no symbols are lost, since it means that the receiver needs to wait for almost the same number of symbols as the original data.

7.1.1.3 *Packet loss distribution*

Since the packet loss can be simulated by code, it exists the possibility to make test depending on the distribution of the lost packets. By this method, it is possible to arrange the lost packets in a big burst of lost symbols, simulating a period where the transmission line would be blocked and thus the receiver shadowed receiving little signal quality, or, in contrast, losing symbols occasionally during all the transmission duration with lower frequency of the losses.

To simulate this situation, two different scenarios were implemented (both of them using the 10 MB file). The first one simulates the burst of errors which will be of 680 symbols, in the second one 1 out of 10 symbols will be discarded until reaching that number. Of course, enough parity information will be transferred with the symbols to be able to repair the simulated packet loss.

The measurements of the test cases gave exactly the same results, as it was expected, since in both cases the receiver needs to wait for the same number of symbols, no matter which ones or which order the symbols were lost.

7.1.1.4 5G-MAG MBMS modem performance

Once the performance of the additional code of the FLUTE transmitter has been studied, it is time to start testing the functioning of the other part of the transmission chain, which is the FeMBMS checkout of the srsRAN software in the transmitter side and the addition of the 5G-MAG MBMS modem in the receiver side.

The first step before start performing test was to characterize what were the attributes of the network used in the trials. For that purpose, the first test was to use the Iperf software to know how many packets were dropped in the transmission between both computers by the 5G-MAG modem transceiver. The following screenshots show what the results were.

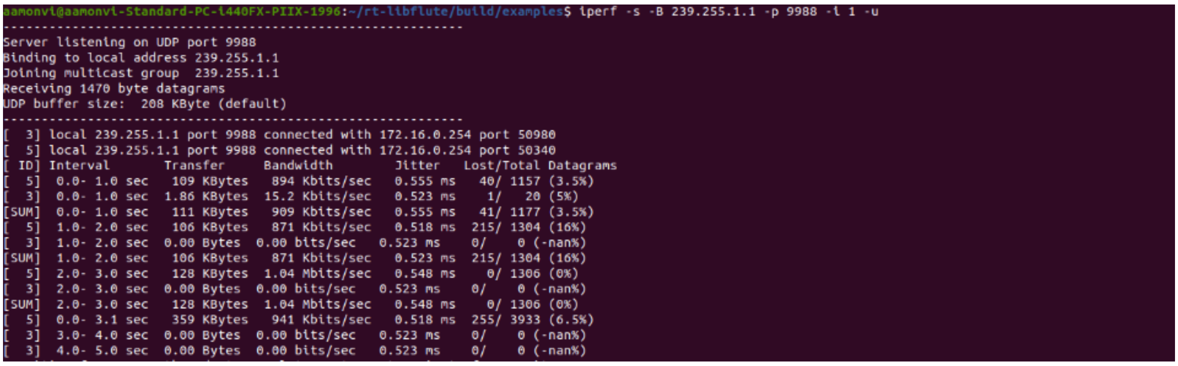
```
aamonvi@aamonvi-Standard-PC-i440FX-PIIX-1996:~/rt-libflute/build/examples$ iperf -s -B 239.255.1.1 -p 9988 -i 1 -u
------------------------------------------------------------
Server listening on UDP port 9988
Binding to local address 239.255.1.1
Joining multicast group  239.255.1.1
Receiving 1470 byte datagrams
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[  3] local 239.255.1.1 port 9988 connected with 172.16.0.254 port 50980
[  5] local 239.255.1.1 port 9988 connected with 172.16.0.254 port 50340
[ ID] Interval       Transfer     Bandwidth        Jitter   Lost/Total Datagrams
[  5]  0.0- 1.0 sec   109 KBytes   894 Kbits/sec   0.555 ms   40/ 1157 (3.5%)
[  3]  0.0- 1.0 sec  1.86 KBytes  15.2 Kbits/sec   0.523 ms    1/   20 (5%)
[SUM]  0.0- 1.0 sec   111 KBytes   909 Kbits/sec   0.555 ms   41/ 1177 (3.5%)
[  5]  1.0- 2.0 sec   106 KBytes   871 Kbits/sec   0.518 ms  215/ 1304 (16%)
[  3]  1.0- 2.0 sec  0.00 Bytes  0.00 bits/sec   0.523 ms    0/    0 (-nan%)
[SUM]  1.0- 2.0 sec   106 KBytes   871 Kbits/sec   0.523 ms  215/ 1304 (16%)
[  5]  2.0- 3.0 sec   128 KBytes  1.04 Mbits/sec   0.548 ms    0/ 1306 (0%)
[  3]  2.0- 3.0 sec  0.00 Bytes  0.00 bits/sec   0.523 ms    0/    0 (-nan%)
[SUM]  2.0- 3.0 sec   128 KBytes  1.04 Mbits/sec   0.548 ms    0/ 1306 (0%)
[  5]  0.0- 3.1 sec   359 KBytes   941 Kbits/sec   0.518 ms  255/ 3933 (6.5%)
[  3]  3.0- 4.0 sec  0.00 Bytes  0.00 bits/sec   0.523 ms    0/    0 (-nan%)
[  3]  4.0- 5.0 sec  0.00 Bytes  0.00 bits/sec   0.523 ms    0/    0 (-nan%)
```

Fig. 25 Iperf results using 5G-MAG modem

The Iperf data shows what is the real bandwidth between both of the VM acting as transmitter and receiver, which in total is around 1,05 Mbps. It is important to note that this attribute can act as bottleneck in the following trials as no higher speed could be achieved.

Another important characteristic is the number of lost packets. This aspect emphasized even more the importance of the addition of RaptorQ in the system since the packet lost will derive in the wrong reception of the packet and in consequence the malformation of the final object.

The limitations of the transmission chain are not caused by the addition of the RaptorQ code but by the non-optimal transmission chain implementations. An improvement in the hardware used in the trials will derive in the increment of the bandwidth and therefore in a better performance of the system. Nevertheless, the system implemented in the thesis was the best possible performance out of the possibilities of the thesis.

7.1.2 Memory measurements

One of the most challenging parts in a software development is to achieve the optimal memory management. Inside of a program, it is important to allocate the amount of memory that is going to be used and freeing it at the right moment. This section will study what is the code memory usage all along the transmission process.

In the transmitter side, the program allocates memory for the input files to be transmitted and for the created repair symbols, so it depends on the amount of protection that is applied to the source symbols. Nevertheless, this is not all the used memory, as the program also needs to allocate enough memory to store all the variables of the code. Something similar happens in the receiver side. However, in this case the amount of allocated memory is fixed and equal to the number of source symbols plus the number of extra repair symbols that one may want to include to add extra reliability to the decoding process. Since, as it was explained in the theory section, only the ideal case is able to decode the original information with the same number of symbols in the totality of the cases. In practice, a little number of extra repair symbols is needed to ensure the correct decoding.

The measurements were made with the “ps” command from Linux, which represent different information about the processes running in a machine. The different memory measurements that this command show include the following parameters:

- RSS (Resident Set Size): the amount of non-swapped physical memory that a task has used (in kilobytes). This is the quantity of memory allocated to that process in the RAM. It includes all stack and heap memory.
- VSZ (Virtual Memory Size): All memory that the process can access. It includes swapped memory, allocated but not used memory and memory from shared libraries.
- DRS (Data Resident Set size): Amount of physical memory devoted to other than executable code.
- %MEM: Ratio of the process’s resident set size to the physical memory on the machine, expressed as percentage.

To make the study more reliable, only the RSS measurements have been considered for the comparisons.

As a first step, the memory used in the transmitter has been measured for different amount of created repair symbols.

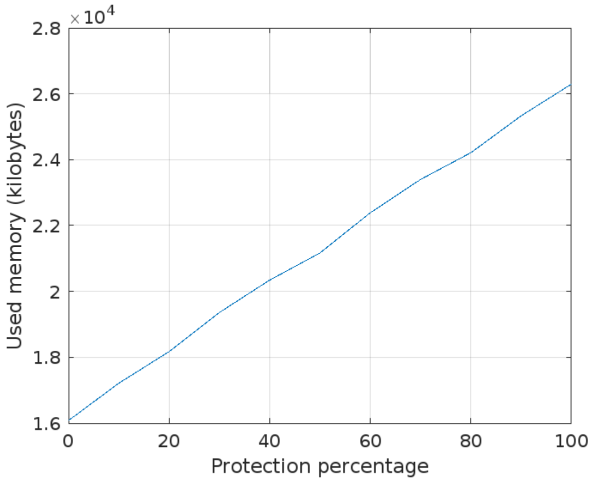

Fig. 26 Memory usage vs protection percentage for a 10 MB transmission

The measurement for a transmission with no protection was 16080 kilobytes, extracting the size of the file (10 MB), the rest is additional memory used for storing the other variables, which gives a total of 6 MB. These results need to be compared with the performance of the CNC mode of the transmitter, which uses a total RSS of 15952. The difference between both modes is negligible and can be due to additional variable that are needed in the RaptorQ mode code. For the rest of coding rates, the increment is linear, with a slope of just the additional information added by the additional repair symbols.

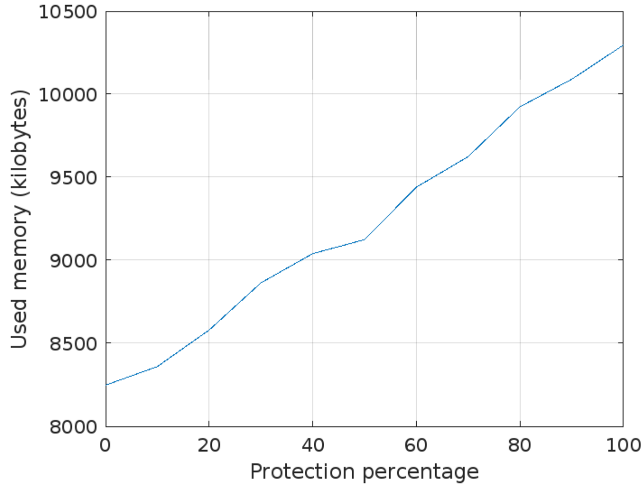

Fig. 27 Memory usage vs protection percentage for a 2 MB transmission

The same measurements have been performed with a smaller file. In this case, the base memory usage, for no protected transmission is 8248 kilobytes. Considering the size of the file, 2 MB, it can be considered that the memory usage of the code is the same than in the first situation. Yet again, comparing this value with the CNC mode performance, the difference is 40 kilobytes, being the memory usage of the basic mode 8208 kilobytes.

In the receiver side, as it happens in the executional time measurements, the result remains constant no matter the amount of protection the transmission has. The necessary symbols for the correct decoding of the information remain the same, thus, the amount of allocated memory remains the same. In the 2 MB transmission, the calculated RSS value was 8404 kilobytes, including the size of the file, the size of the extra received symbols for a more reliable decoding and the memory used for other purposes. In comparison with this, the basic mode of the receiver uses an amount of 8264, again really close to RaptorQ Flute receiver usage. Similarly, the 10 MB transmission values of 16724 kilobytes in the RaptorQ case and 16232 bytes in the basic case. Considering these results, the memory consumption of the FLUTE decoder can be considered optimal, since no additional allocated is needed to perform the decoding no matter the used coding rate.

7.2 Complete transmission chain results

The complete chain uses both 5G-MAG MBMS modem and the 5G-MAG FLUTE transceiver. To realize the real attributes in this situation, the original version of the FLUTE transceiver has been used. This means that the CNC mode is used.

To take on these tests the rate-limit of the system has been taken as varying attribute. The results show how the Compact No Code of the implementation could work fine in most of the trials with a rate limit of 2 MB/s. Nevertheless, the performance struggle to be enough for rate limits between 2 and 2.5 MB/s having no successful results for the upper bound. For that reason, the rate limit used in the tests of the thesis has been varying in this margin to be able to map the benefits that the new implementation is adding to the system. It is important to remember that the channel bandwidth that was calculated while using 5G-MAG MBMS modem was slightly better than 1 Mbps. In other words, this trials test transmissions where the transmission rate is higher than the channel can handle, and this could cause packet loss. Another important trial was the testing of an ffmpeg stream using the 5G MAG transmission chain. The extracted bandwidth by ffmpeg was around 2 MB, which gives a hint about why the CNC transmission was possible with that throughput.

Another aspect worth to remark is the problems related with the late reception of the File Delivery Table. As it was explained in the FLUTE section, the FDT is recommended to be sent before the start of the content transmission and several times during the streaming duration. Thanks to this, the receiver can still know to which file some packets belong to even with the late reception of the FDT. However, in a file delivery service using the CNC mode of the implementation, the late reception of the FDT is a huge issue, because the previous packets would have been discarded before the FDT reception and thus the packet could never be completed. This issue is also solved by the addition of RaptorQ encoding, where the previous symbols can be considered as lost and if sufficient symbols are received, the file will be able to be successfully received.

The first test to explore implies the testing of the maximum speed the transmission can reach achieving the correct reception of the file for each coding rate. It is important to remember that the transmission rate that could be achieved in the Compact No-Code mode was between 2 and 2.5 Mbps. This aspect implies that for throughputs higher than this threshold the transmission rates would be too high for the channel to transport all the symbols and some of them will be lost.

Rate-limit in the transmitter	Percentage of protection
1 Mbps	0
2 Mbps	0
3 Mbps	10
4 Mbps	10
5 Mbps	20
6 Mbps	30
7 Mbps	50
8 Mbps	70
9 Mbps	70
10 Mbps	80

Table 1 Necessary coding rate to correctly decode using each transmission rate

The table shows how more symbols are lost with the increment of the transmission rate due to the channel conditions. Nevertheless, RaptorQ codes are able to recover from this challenging environment and by the addition of parity information it is able to decode the original file.

The effect of the bad transmission channel would be even more emphasized in longer transmissions, as the system will collapse by the higher throughput of the transmitter and due to the bottleneck effect, lots of symbols will be lost. To show this case a 10 MB transmission has been evaluated using different transmission rates.

Transceiver mode	Percentage of protection	Rate-limit in the transmitter	Test result
CNC	0	1 Mbps	Success
CNC	0	2 Mbps	Failed
RaptorQ	10	2 Mbps	Failed
RaptorQ	20	2 Mbps	Success
RaptorQ	30	3 Mbps	Failed
RaptorQ	40	3 Mbps	Failed
RaptorQ	50	3 Mbps	Success

Table 2 RaptorQ enhanced 5G-MAG MBMS modem results in a 10 MB transmission

It is important to remark that the failed cases are because of an insufficient quantity of received symbols, not due to RaptorQ performance. This test proved the improved capabilities added by RaptorQ, making the reception possible even with poor channel conditions. This would be impossible to achieve with the use of CNC mode. However, it is important to remark that the speed of the transmission would be similar to transmissions with a 1 MB rate limiter in the transmitter as this is the rate in which the receiver gives the symbols to the FLUTE application.

8 FUTURE RESEARCH

8.1 *TelecomCLM pilot*

The necessary systems to develop the TelecomCLM are already being deployed in the city of Toledo.

Fig. 28 TelecomCLM pilot deployment in Toledo

As the on-field trials of the TelecomCLM pilot were scheduled for the end of 2022, the results extracted from those trials are out of the scope of this thesis. However, considering the hardware set up used in this thesis and the positive results extracted from the trials using the open-source 5G-

Broadcast transmitter, only better results are expected to be acquired in future trials with professional hardware.

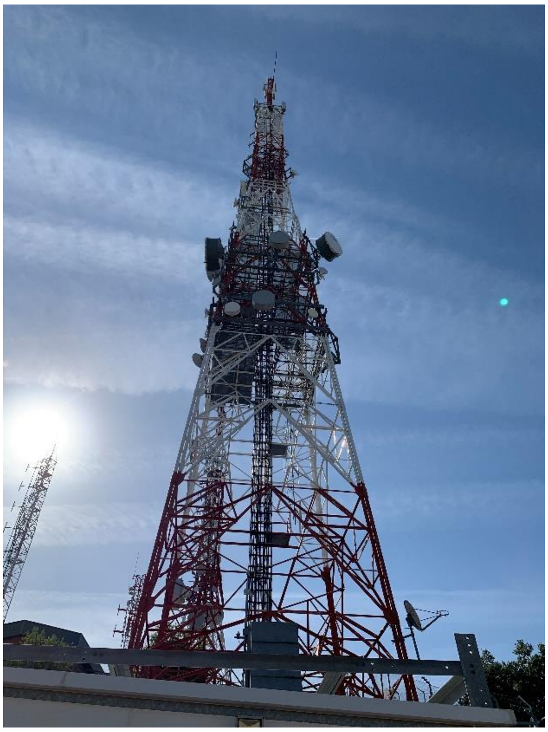

Fig. 29 High Power High Tower used in TelecomCLM pilot

8.2 *Route transmitter update*

As future research, the FLUTE transmitter used in the thesis can be updated to use more recent transport protocols. One of the best options to perform the update is ROUTE protocol (Real-time Transport Object delivery over Unidirectional Transport) [19] which can be used for robust delivery of application objects, including application objects with real-time delivery constraints, to receivers over a unidirectional transport. The application of this protocol targets services enabled media consumption devices such as smartphones, tablets and so on. Even if the protocol was design for time sensitive reception use cases it is also capable of transporting objects which are not intended to be delivered in real time. In fact, ROUTE protocol is very aligned with FLUTE protocol as well as the extensions defined in MBMS. It is aligned with FLUTE since it uses most of the basic features of the protocol. In addition, ROUTE protocol presents some enhancements as:

- Real-time delivery of object-based media data.
- Flexible packetization, including media-aware packetization as well transport-aware packetization of delivery object.
- Independence of Application objects and delivery objects.

Based on FLUTE protocol functioning, the ROUTE protocol is based on sessions where the different objects will be transferred. In this case, ROUTE receiver should get some metadata before being able to connect to the session. Some of the information included in the metadata is:

- Route version number.
- Connection ID: Identifier that represents the ROUTE session, usually represented by source IP/source port and destination IP/destination port.
- STSI: LCT TSI value corresponding of the Transport session for the source flow.
- Rt: A Boolean representing if the object being carried is intended for real time delivery or not.

All these enhancements could be applied to the 5G-MAG transmitter, using the old transmitter implementation as a base and thus improving the overall performance and enabling real time object delivery.

8.3 5G-MAG Reference tools future use cases

Since iTEAM is part of the 5G-MAG association the use of RaptorQ codes can be extended to additional 5G-MAG use cases. In the 5G-MAG agenda there are several 5G media uses cases targeted for 2023 in which AL-FEC protection could be beneficial. Some of the future use cases are:

- Reliable video on demand over mobile network.
- Sustainable 5G streaming over 5G.
- Dynamic Broadcast-Multicast.
- Broadcast on demand.
- Emergency alert and media services through 5G Broadcast.

In the near future, 5G-MAG developers will study the design of the architecture of these new use cases and iTEAM will take part on the process.

9 CONCLUSION

This document introduced the growing global data issue and the need to explore new solutions to address this problem. Among the possible solutions, a change in the transmission delivery method presented the suitable attributes to solve this issue, concretely the use of point-to-multipoint transmissions. Nevertheless, the use of point-to-multipoint transmissions derived in a worse quality of the received signal, due to the lack of a feedback channel enabling the correct channel estimation and equalization to each of the users. To still be able to receive the signal, the addition of forward error correction methods was presented as a possible solution, in particular the use of Application Layer Forward Error Correction codes. The addition of these codes permits the recovery of the lost data due to poor reception conditions allowing the correct reception of the transmitted object.

In the mark of this thesis, an open-source FLUTE transmitter was updated and enhanced with the addition of AL-FEC codes to the application. RaptorQ, a high-performance AL-FEC codes could be deployed into the code thanks to an agreement between RaptorQ codes owner, BitRipple, and UPV.

The performance of the codes has been studied using an open source 5G Broadcast transmitter as the transmission chain. BitRipple's RaptorQ libraries were able to achieve the decoding of the original information in the receiver even in the most challenging environments. Furthermore, as it has been proved in the results of this thesis, the codes present near optimal performance, adding minimal overhead to the transmissions and negligible increment in the transmission latency.

As an additional work, the code is being deployed into a professional deployment, in the mark of a national pilot led by TelecomCLM. As an extension, similar positive results are expected in the field trials of the pilot, scheduled for the end 2022.

ACKNOWLEDGEMENTS

The master thesis has been implemented in the mark of the research work of project from national and international domain. Some examples can be the European project H2020 5G-TOURS which consisted in the implementation of different pilots whose intention is to prove the benefits that the fifth generation of mobile communications can provide to the tourist visiting the city of Turin and the aforementioned TelecomCLM project.

To conclude, we thank BitRipple for providing RaptorQ software to iTEAM to make this thesis possible.

REFERENCES

[1] *Ericsson mobility report 2022, https://www.ericsson.com/49d3a0/assets/local/reports-papers/mobility-report/documents/2022/ericsson-mobility-report-june-2022.pdf,[29-8-2022] Online*

[2] *"What is Unicast, Broadcast and Multicast", The Study genius, https://www.thestudygenius.com/unicast-broadcast-multicast/, [29-8-2022] Online.*

[3] *Luby, M., Shokrollahi, A., Watson, M., Stockhammer, T., and L. Minder, "RaptorQ Forward Error Correction Scheme for Object Delivery", RFC 6330, DOI 10.17487/RFC6330, August 2011, <https://www.rfc-editor.org/info/rfc6330>.*

[4] *Lte-based-5g-terrestrial-broadcast-for-tv-and-radio-distribution, 5G-MAG https://www.5g-mag.com/post/lte-based-5g-terrestrial-broadcast-for-tv-and-radio-distribution , [29-8-2022] Online*

[5] *ETSI TS 103 720 V1.1.1 (2020-12) 5G Broadcast System for linear TV and radio services; LTE-based 5G terrestrial broadcast system*

[6] *5G-MAG, https://www.5g-mag.com/, [29-8-2022] Online*

[7] *A.Ibañez, A. Montilla, J. Sánchez, D. Gómez-Barquero (2022), Radiodifusión de contenidos 5G Broadcast.*

[8] *Gomez-Barquero, David & Gimenez, Jordi & Beutler, Roland. (2020). 3GPP Enhancements for Television Services: LTE-based 5G Terrestrial Broadcast.*

[9] ETSI TR 102 993: "Digital Video Broadcasting (DVB); Upper Layer FEC for DVB Systems".

[10] A. Shokrollahi: "Raptor Codes", IEEE Trans. on Information Theory, vol. 52, no. 6, pp. 2251-2567, June 2006.

[11] RaptorQ Technical Overview, www.qualcomm.com/content/dam/qcomm-martech/dm-assets/documents/RaptorQ_Technical_Overview.pdf, [29-8-2022] Online.

[12] 5G-MAG rt-mbms-modem, https://github.com/5G-MAG/rt-mbms-modem , [29-8-2022] Online.

[13] Paila, T., Walsh, R., Luby, M., Roca, V., and R. Lehtonen, "FLUTE - File Delivery over Unidirectional Transport", RFC 6726, DOI 10.17487/RFC6726, November 2012, <https://www.rfc-editor.org/info/rfc6726>.

[14] 5G-MAG rt-libflute, https://github.com/5G-MAG/rt-libflute , *[29-8-2022]* Online

[15] Ettus research, https://www.ettus.com/, *[29-8-2022]* Online

[16] USRP model N310, https://www.ettus.com/all-products/usrp-n310/ , *[29-8-2022]* Online

[17] srsRAN, https://docs.srsran.com/en/latest/ , [29-8-2022] Online

[18] Chrono libraries, https://cplusplus.com/reference/chrono/ , *[29-8-2022]* Online

[19] W. Zia, T. Stockhammer, L. Chaponniere, G. Mandyam, M. Luby, "Real-Time Transport Object Delivery over Unidirectional Transport (ROUTE)", RFC 9223, DOI 10.17487/RFC9223 April 2022, <https://www.rfc-editor.org/info/rfc9223>.

PUBLICATIONS

The following pages will include the publication done while this thesis was being developed.

Radiodifusión de contenidos 5G Broadcast

Alvaro Ibanez*, Aaron Montilla*, Jaime Sanchez*, David Gomez-Barquero*
*Instituto de Telecomunicaciones y Aplicaciones Multimedia, Universitat Politècnica de València, España.
Email:{alibla, aamonvi, jaisanro, dagobar}@iteam.upv.es

***Resumen*—The emergence of 5G networks will mark one of the greatest transformations in recent decades. The technical characteristics of 5G promise to make currently used technologies obsolete. 5G can be so disruptive, not only because of the launch of this technology itself, but also because of how it will be able to support other technologies that need this new connectivity and data rates offered by 5G. One of the issues in which 5G can be key is the Multicast/Broadcast capabilities, which can be very useful for the reception of the increasing multimedia content (UHD, 4K, 8K) and also the traditional content (TV and radio). 5G Broadcast consists of carrying out Point To Multipoint (PTM) transmissions delivering, in an efficient way, the same content to a large number of users. In that sense, this paper shows a technical overview of 5G technology, showing the its evolution and also the main features. This papers present an innovative demo in which 5G Broadcast network will be used for the transmission and reception of traditional television and radio multimedia content, in real time, in the city of Toledo, Spain.**

I. Introducción

El estándar de comunicaciones móviles de última generacion (5G), engloba muchas más características y mejoras de las centradas solamente en el aumento de las tasas de datos y reducción de la latencia. Se trata de un sistema de comunicaciones capaz de conectar no sólo a personas sino a todo tipo de máquinas y dispositivos, cambiando industrias y mercados enteros, como el de la conducción automatizada, la industria 4.0, la red inteligente, la salud, industria audiovisual, etc.

El contenido y servicios multimedia ofrecidos por compañías y medios audiovisuales ha evolucionado desde un número limitado de canales de radiodifusión tradicionales (radio y televisión) a una oferta mucho más extensa gracias a las plataformas de distribución digital, como son los servicios *streaming* basados en IP, cada vez más populares entre el público. Actualmente, se consume este tipo de contenido multimedia en un amplio abanico de dispositivos, como televisores convencionales e inteligentes, teléfonos móviles, tabletas, coches, sistemas de entretenimiento portátil, etc. cuya recepción puede ser fija, como ocurre en el hogar, o en movilidad, en un coche, tren, andando por la calle, etc.

En ese sentido, las tecnologías y estándares de transmisión terrestre convencionales, como *Digital Video Broadcasting* (DVB-T/DVB-T2), ATSC 3.0, etc. pueden ofrecer contenidos multimedia tradicionales a dispositivos portátiles y móviles. Sin embargo, estos estándares no son compatibles con estos dispositivos, ya que no incorporan el hardware necesario.

En el mercado existen millones de dispositivos, véase móviles o tablets, que pueden beneficiarse de la transmisión de contenido multimedia tradicional en una red ya compatible con los dispositivos actuales. El estándar 5G proporciona las herramientas para ello, particularmente para que los servicios de radiodifusión tradicionales se puedan transmitir a través de la red 5G en lo que se denomina 5G Broadcast, 5G para difusión o radiodifusión. Este 5G Broadcast permite la distribución de contenido mediante una transmisión *multicast*, enviando la misma información a un gran número de usuarios a través del mismo canal físico, lo que posibilita que una gran cantidad de dispositivos actuales compatibles con el estándar 5G puedan a través de este nuevo modo de transmisión recibir y decodificar contenido multimedia.

La radiodifusión de televisión mediante redes 5G como complemento a la Televisión Digital Terrestre (TDT), cuyo estándar es el DVB-T en España, es uno de los novedosos modos de transmisión que estandarizó el 3GPP (*The 3rd Generation Partnership Project*) en la segunda versión de 5G (*release* 16), mejorando las capacidades *multicast/broadcast* de LTE (*release* 14), conocidas como FeMBMS *(Further evolved Multimedia Broadcast Multicast Service)*.

En los últimos años se han realizado trabajos acerca de la distribución de contenido multimedia utilizando redes 4G y 5G en entornos reales, empleando en la mayoría de casos equipamiento comercial de la compañía *Rohde & Schwarz*. La *Radiotelevisione Italiana* (RAI) llevó a cabo en el Valle de Aosta la difusión de contenidos multimedia empleando el modo FeMBMS [1]. La prueba se implementó usando repetidores de alta altura y potencia, HPHT *(high power high tower)*, utilizando cinco transmisores operando en una red de frecuencia única, del inglés *single frecuency networks* o SFN. También como parte del proyecto 5G TODAY [2], se realizó una prueba basada en el modo FeMBMS en una red SFN en la región bávara de Oberland. Los transmisores se encontraban separados 63 km uno de otro, con una potencia de emisión 100 kW y un ancho de banda de 5 MHz (MCS 9, QPSK, *data rate* 3192 kbit/s).

Otra de las pruebas en las que se empleó el modo FeMBMS fue en Beijing (China) en 2019 [3], en la que se realizó una emisión a través de tres estaciones base operando en SFN con una potencia de salida de 1 kW a una frecuencia central de 754 MHz y un ancho de banda de 5 MHz. También se han realizado pruebas empleando una red 5G Broadcast con el equipamiento de R&S en Barcelona [4] y en Rio de Janeiro.

El objetivo principal de este documento es describir el piloto de Red.es llevado a cabo en Castilla-La Mancha, concretamente en Toledo, donde se realizará la transmisión y recepción de un canal de televisión y radio de Radio Televisión Española (RTVE), a través de la red 5G.

El contenido del artículo se estructura de la siguiente forma, en la sección 2 se hace un resumen de la tecnología 5G Broadcast, donde se describe su evolución a lo largo de las versiones de la especificación así como sus principales características. En la sección 3 se presenta el piloto de Castilla-La Mancha de Red.Es que empleará una red 5G Broadcast en la ciudad de Toledo, mostrando los objetivos principales, las innovaciones y describiendo todos los elementos que van a formar parte del piloto. Por último, se muestran las conclusiones del artículo.

II. Tecnología 5G Broadcast

A. *Broadcast en 3GPP*

Desde la aparición de las redes móviles una de las características que han sido introducidas y mejoradas con el paso de las especificaciones ha sido la transmisión de contenido *multicast/broadcast*. En la *release* 6 del 3GPP [5] se definió por primera vez el modo *broadcast* o difusión, denominado MBMS (*multimedia broadcast and multicast services*). Este tipo de transmisión permitía un aumento en la eficiencia del uso de recursos radio mediante la transmisión de datos en modo *multicast/broadcast* utilizando conexiones punto a multipunto (PTM, *point to multipoint*). MBMS abarca cambios tanto en el terminal de usuario como en el núcleo de red y la interfaz radio.

Con la llegada de la cuarta generación de redes móviles (4G), el soporte de transmisiones PTM fue añadido como un requisito. Por esta razón, en la *release 9* [6] se añadieron nuevas características a MBMS, pasando a denominarse eMBMS (*evolved* MBMS), también conocido como LTE Broadcast. eMBMS ha continuado desarrollándose en las posteriores versiones del estándar, sin embargo, hasta la *release* 14 estaba limitado por algunas características que lo hacían poco atractivo para los proveedores de contenido de radiodifusión. Entre estas limitaciones se destacan: (i) Límite del 60 % de los recursos radio para *broadcast/multicast*, dejando el resto, el 40 %, para datos de control y transmisiones de usuario dedicadas (*unicast*). (ii) Cobertura similar a 3G/4G, con un alcance muy limitado, lo que contrasta con las grandes zonas cubiertas por un único transmisor en las redes de radiodifusión.

Otro de los inconvenientes que presentaba las transmisiones PTM fue la dificultad a la hora de realizar la estimación del canal por usuario, pudiendo producirse pérdidas de paquetes o decodificaciones erróneas en recepción. Para solucionar este problema, en eMBMS (*release* 9) se añadió un mecanismo de corrección de errores en la capa de aplicación llamado AL-FEC (*aplication layer forward error correction*), el cual permitía la recuperación de los paquetes perdidos en la transmisión sin necesidad de interacción entre cliente y servidor. Los códigos utilizados en eMBMS para AL-FEC son los Raptor, que serán presentados más adelante.

Con la llegada de las especificaciones de la *release* 14, se mejoró significativamente el modo eMBMS pasando a denominarse *Further eMBMS (FeMBMS)*, introduciendo mejoras en la capa física, suponiendo así un gran avance en las especificaciones de radiodifusión en redes móviles.

B. *Raptor Q: Códigos de corrección de errores (AL-FEC)*

Los códigos de corrección de errores empleados en la capa de aplicación en el estándar del 3GPP son los códigos RaptorQ [7] desarrollados por Amin Shokrollahi en 2001 [8]. Estos códigos pertenecen a la clase de códigos fuente (en inglés *fountain codes*) los cuales son capaces de decodificar la información a partir de cualquier subconjunto de símbolos recibidos, siempre y cuando la cantidad de éstos sea mayor que la de símbolos fuente enviados.

RaptorQ ofrece prestaciones cercanas a las ideales, donde destaca el mínimo tiempo de procesado que introduce, la gran flexibilidad que ofrece en la generación de los datos de corrección y la eficiencia en la reparación de los datos originales. Proporciona una correcta recuperación de la información a partir de una secuencia de datos (originales y de reparación) ligeramente mayor a la enviada, lo que permite llevar a cabo la corrección de errores con un mínimo aumento en la latencia y mínima disminución en la capacidad.

C. *FeMBMS Release 14*

FeMBMS introdujo un modo de transmisión que puede utilizarse para transmitir contenido multimedia tradicional [9] para radiodifusión, [10]. Las principales novedades que se adoptaron en la *release* 14, a nivel de arquitectura son: (i) El 100 % de los recursos radios son dedicados a transmisión *broadcast*, ya no hay recursos reservados a transmisiones de usuario dedicadas (ii) Un modo de recepción única o ROM *(receive only mode)* que permite la difusión de contenido en abierto, pudiendo ser recibido por cualquier tipo de dispositivo sin necesidad de estar registrado en la red, por lo que ya no es necesaria una tarjeta SIM. (iii) Un modo de transmisión transparente, que permite la difusión de contenidos que hacen uso de protocolos no soportados por el 3GPP, por ejemplo, MPEG-2 TS sobre IP. (iv) Una nueva interfaz xMB para simplificar el envío de contenido a la plataforma FeMBMS por parte de los proveedores de contenidos. (v) Una nueva interfaz de programación de aplicaciones abierta (MBMS-API) para simplificar el acceso a los procedimientos de FeMBMS en los receptores.

Las mejoras en la red de acceso radio RAN (*radio access network*), introducidas en FeMBMS son: (i) Nuevo tipo de subtrama, llamada CAS *(cell acquisition signalling)*, con información de sincronización, señalización y control. (ii) Soporte para mayores distancias entre emplazamientos en una red SFN, mediante nuevas numerologías OFDM. En concreto, se añadió una con prefijo cíclico de 200 μs para cubrir una distancia ISD *(inter-site distance)* de 15 km, y un espaciado entre subportadoras de 1.25 kHz. (iii) Se añadió la opción para compartir la misma red de acceso radio entre los distintos operadores de telecomunicaciones, evitando la transmisión del mismo contenido a través de varias redes distintas simultáneamente.

Pese a todas las mejoras introducidas, FeMBMS todavía no satisfacía las necesidades de los radiodifusores. Por otro lado, en las primeras especificaciones de 5G, en la *release* 15, no se incluyó ninguna característica nueva referente a *broadcast/multicast*. No fue hasta la *release* 16 cuando se incorporaron nuevas mejoras para FeMBMS en la capa física, con el objetivo de aumentar la cobertura y permitir una recepción óptima en recepción móvil a gran velocidad incluso bajo presencia de un mayor nivel de interferencias y ruido.

D. *5G Broadcast LTE-Based release 16*

Las mejoras planteadas en *release* 16, como se ha comentado con anterioridad, se centran en la capa física, concretamente se añaden dos nuevas numerologías OFDM y se mejora la robustez de la señalización en el CAS, para mejorar la recepción y decodificación en escenarios con peor señal a ruido. [11].

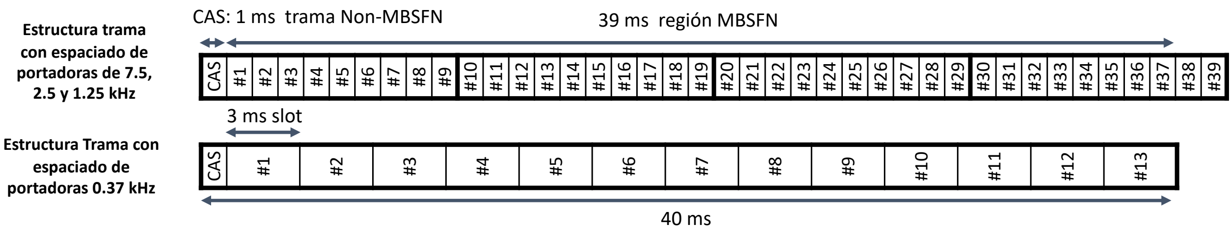

Fig. 1. Estructura de trama para las numerologías OFDM FeMBMS y 5G *Broadcast* [11]

TABLA I
NUEVAS NUMEROLOGÍAS FEMBMS-RELEASE 16

Release	SCS (kHz)	OFDM Símbolos por subtrama	Subportadoras por *resource block*	T_{CP} (μs)	T_u (μs)	ISD (km)
14	1.25	1	144	200	800	60
16	2.5	2	72	100	400	30
16	0.37	1	486	300	2700	100

1. Nuevas numerologías OFDM: Las dos numerologías OFDM introducidas en *release* 16 tienen como objetivos principal aumentar la zona de cobertura, pudiendo alcanzar distancias entre transmisores mayores y mejorar la recepción en condiciones de movilidad.

(i) La primera numerología añadida tiene un espaciado entre subportadora de 0.37 kHz y una duración de prefijo cíclico (CP, *cycle prefix*) de 300 μs con un tiempo de símbolo OFDM de 2700 μs. Esta numerología rompe con la estructura de trama tradicional de LTE, como se muestra en la Figura 1, siendo el tiempo de subtrama tres veces superior al que se usaba en LTE. El aumento del tiempo del prefijo cíclico está pensado para aumentar las distancias entre transmisores en recepción fija hasta los 100 km. (ii) La numerología con un espaciado de 2.5 kHz y un prefijo cíclico de 100 μs con un tiempo de símbolo OFDM de 400 μs, permite una recepción a altas velocidades. En la Tabla I se representan las características principales de estas dos numerologías y la comparativa con la introducida en FeMBMS.

2. Mejora del CAS: Por otro lado, los cambios en el CAS se centran en mejorar la recepción y decodificación de esta subtrama en escenarios con peor relación señal a ruido, escenarios con interferencias, en movilidad con altas velocidades, etc. ya que podía darse la situación en la que los datos podían decodificarse, pero el CAS que emplea una numerología con un espaciado entre de subportadoras de 15 kHz, no era recibido correctamente haciendo imposible la decodificación de la información.

Estas mejoras se llevaron a cabo mediante los siguientes cambios. (i) Repetición del canal PBCH (*physical broadcast channel*): El canal físico de difusión transporta la información de bloque principal, fundamental para poder decodificar los datos. Contiene por ejemplo el ancho de banda del sistema. En la *release* 16 se añadió la repetición del PBCH, mejorando así la recepción y sincronización por parte del terminal de usuario. (ii) Nuevo nivel de agrupación, AL *(agreggation level)*: El AL indica cuántos elementos de control se utilizan para transmitir el canal de control PDCCH (*physical downlink control channel*). Se añadió el nivel 16, siendo hasta entonces el nivel 8 el valor máximo en FeMBMS. Se consigue así una mejora en la recepción en condiciones de baja relación señal a ruido, como se demuestra en [12].

3. Futuras mejoras: Tras la *release* 16, los esfuerzos del 3GPP en cuanto a la transmisión *multicast/broadcast*, se centran en una solución nativa 5G NR, más eficiente y flexible, basada en la propuesta del proyecto 5G-Xcast [13]. El 3GPP se encuentra trabajando en la *release* 17 en lo que se denomina modo mixto *(mixed mode)*, o 5G NR MBS *(multicast broadcast service)* para permitir a las redes 5G hacer transmisiones *multicast/broadcast* tanto a nivel radio como a nivel de núcleo de red de manera dinámica y bajo demanda. Sin embargo, el modo de radiodifusión terrestre de NR no llegará hasta la *release* 18. Por otro lado, en la *release* 17, la unión de radiodifusores europeos, EBU (European Broadcasters Union) propuso añadir a 5G Broadcast LTE-Based los anchos de banda de TDT, algo que ha sido aprobado y cuyo estudio y posterior estandarización empezará en los próximos meses [14].

III. PILOTO 5G BROADCAST RED.ES DE CASTILLA-LA MANCHA

En esta sección se presenta y detalla el piloto de Castilla-La Mancha de Red.Es. El objetivo principal es realizar una emisión de contenido de TV a través de una red 5G Broadcast en la ciudad de Toledo (véase Fig. 2) mediante un repetidor HPHT (*high power high tower*) y usando equipos de radio definida por software, SDR (*software defined radio*), tanto en transmisión como en recepción.

El piloto se realizará en su totalidad en la ciudad de Toledo, donde se grabará contenido de televisión y radio en tiempo real para ser procesado y emitido desde el centro emisor de TDT (Televisión Digital Terrestre) en el Cerro de Palos, donde actualmente se radia la señal de TV en el término municipal de Toledo. Se trata por tanto de una prueba totalmente innovadora, ya que hasta la fecha no se han realizado transmisiones de radiodifusión de contenidos de TV y radio con equipos compatibles con las especificaciones de *release* 16 del 3GPP.

A continuación detallar de forma más exhaustiva el Piloto de Red.es, para ello, se presenta el consorcio de empresas e instituciones que están involucradas, así como todos los componentes y los estudios realizados.

A. *Demostración 5G Broadcast*

El desarrollo del piloto utilizará la grabación de un evento en directo realizado en el edificio de las Cortes de Castilla-

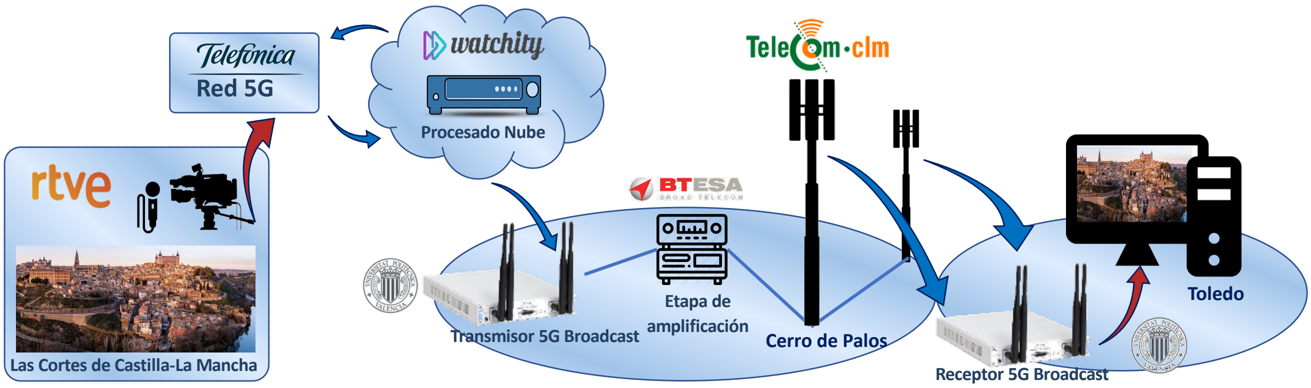

Fig. 2. Arquitectura del piloto TelecomCLM de Red.es

Fig. 3. Localización de centro de transmisión de Cerro Palos

La Mancha. El contenido será procesado en tiempo real y enviado al centro de emisión de TDT en Cerro de Palos (véase Figura 3). Al no haber en la actualidad equipos comerciales compatibles con las especificaciones 5G Broadcast, se hará uso de dispositivos SDR *(software defined radio)*, necesarios para poder implementar de forma rápida y eficaz las novedades introducidas en la últimas versiones del estándar. Estos equipos SDR servirán para poder transmitir el contenido de TV y también servirán para la recepción.

Estas soluciones SDR ofrecen una comunicación extremo a extremo transparente para el proveedor de contenidos, sin embargo al estar diseñados para uso en laboratorio presentan una serie de inconvenientes en cuanto al uso en pruebas de campo ya que trabajan con potencias bajas y no cuentan con algoritmos optimizados para entornos de recepción desfavorables, entre otros.

Por todo ello, será necesario optimizar la cadena de transmisión y recepción. En ese sentido, se ha realizado un estudio de la cobertura desde el centro emisor de Cerro de Palos, para hacer una planificación óptima de las potencias de transmisión en la banda de frecuencia de emisión, entorno a los 700 MHz. Otro de los aspectos a tener en cuenta para la realización de la demostración es la integración de los equipos SDR con el emisor HPHT. Siendo necesario un sistema de preamplificación en la etapa de radio frecuencia (RF).

Por otro lado y de forma innovadora, se hará uso de códigos de corrección de errores en la capa de aplicación, implementados mediante códigos RaptorQ [8], en su versión más moderna, los cuales permitirán una mejora en la recepción de contenidos multimedia.

B. *Consorcio Piloto TelecomCLM*

El consorcio estará formado por diferentes empresas e instituciones de carácter privado y público. Se divide en dos grupos, el primero lo constituyen las compañías que se dedicarán a la producción y procesado del contenido de TV y radio, y el segundo por los encargados de transmitir y recibir ese contenido.

El primer grupo lo componen: (i) Radio Televisión Española (RTVE): encargada de proporcionar el contenido que va a transmitirse. (ii) Telefónica: encargada de poner a disposición la red 5G qué llevará el contenido a la nube para su posterior procesado. (iii) Watchity: encargado de suministrar el servidor en la nube que permite en tiempo real la realización y producción de la señal desde un navegador web.

El segundo grupo lo componen: (i) TelecomCLM: encargado de proporcionar el transmisor HPHT para la radiodifusión del contenido además de coordinar del consorcio. (ii) BTESA: encargada de realizar el estudio de cobertura y dimensionar la etapa de preamplificación previa al repetidor HPHT, así como de proveer los equipos profesionales (amplificadores y reemisores RF). (iii) El Instituto de Telecomunicaciones y Aplicaciones Multimedia (ITEAM) de la Universidad Politècnica de València (UPV) que aportará su experiencia previa en demostraciones y ensayos en implementaciones del estándar del 3GPP mediante soluciones SDR, desarrollando los equipos compatibles con las especificaciones de la *release* 16 para el transmisor y receptor.

C. *Estudio de cobertura*

Previo al desarrollo del Piloto, se requiere un estudio de cobertura del transmisor de televisión digital terrestre instalado en Cerro de Palos (Toledo), el cual cuenta con un emisor adicional o *gap filler* en Salto del Caballo (Toledo) para extender la cobertura.

TABLA II
PARÁMETROS PARA ESTUDIO COBERTURA

Parámetro	Valor
P_N Potencia ruido receptor	-131 dBW
Factor Ruido receptor	6 dB
Ancho Banda	5 MHz
P_S Potencia minima receptor	-111 dBW
C/N	20.4 dB
Frecuencia	700 MHz
Intensidad mínima campo	50.5 dBμV/m
Potencias emisión	200-500W
Altura antena transmisora	50m
Ganancia antena transmisora	11.3 dBi

Para el estudio, se ha utilizado un software específico para el análisis de coberturas, con mapas cartográficos de alta resolución (5 m/pixel), con información cartográfica de edificios (clutter). El estudio de cobertura se basa en las recomendaciones de la ITU-R P.525/526-15 [15], [16], utilizando una geometría de difracción basada en el método *Delta-Bullington*. El objetivo del estudio es obtener la potencia necesaria de emisión para conseguir un valor mínimo de campo de 50 dbuV/m en el área de recepción de la señal de televisión (alternando entre el Canal 24h y la señal en Las Cortes) y la señal de Radio 5. Debido a que los anchos de banda de LTE difieren de los de TDT, se ha elegido el ancho de banda de 5 MHz para la transmisión del contenido. La modulación utilizada para alcanzar un tasa de transmisión en torno a los 5 Mbps es la 16 QAM. Con dicha modulación la C/N (*carrier to noise ratio*) mínima requerida en el receptor ronda los 20 dB.

Según la EBU TECH 3348 *(Frequency and Network Planning Aspects of DVB-T2)* [17], los valores de campo mínimo en condiciones de visión directa entre antena transmisora y receptora se sitúan en torno a los 50 dBμV/m. En la Figura 4 se muestran los resultados del estudio de cobertura, donde se puede observar un campo mínimo de 50 dBμV/m en casi toda la ciudad y valores por encima de 65 dBμV/m en muchos emplazamientos y una buena parte entre 71 y 78 dBμV/m. En base a estos resultados se va a emplear una potencia máxima de 500 W.

D. Plataformas y equipos SDR

1. Plataformas SDR: En la actualidad existen varias plataformas y equipos SDR compatibles con las especificaciones FeMBMS del 3GPP. Dentro de toda la oferta de código libre u *open source* disponibles en el mercado, se encuentra la plataforma OpenAirinterface (OAI) [18] y la de ORS (Austrian-Broadcasting-Services), OBECA (Open Broadcast Edge Cache Appliance) [19]. La versión FeMBMS de OAI ya fue usada en un escenario real [2], sin embargo, la versión actual no implementa las novedades de *release* 16, por lo que será necesario añadirlas. Por otro lado, la versión del software OBECA ofrece una solución FeMBMS completa y en un futuro cercano dispondrán de una versión compatible con las especificaciones de *release* 16. Se contempla el uso de ambas plataformas para el desarollo del Piloto.

2. Equipos SDR: Las plataformas SDR requieren de unos dispositivos físicos para convertir la implementación software (señal digital) a señal de RF. En el mercado existe una inmensa variedad de dispositivos que realizan esta conversión, entre ellos se encuentran los USRPs (*universal software radio peripheral*), del fabricante Ettus Research. Un USRP es un dispositivo de radio frecuencia reconfigurable que incluye una combinación de procesadores basados en FPGA y terminales RF. Dentro de la gama de USRPs que ofrece Ettus el más interesante es el USRP B210 [20], (véase Figura 5), ya que es el más estable para trabajar con las plataformas *open source* nombradas anteriormente y el que ofrece una mayor relación calidad precio.

E. Etapa de preamplificación y potencia

Uno de los inconvenientes que presentan los dispositivos SDR (como los USRPs) es su reducida potencia de emisión (sobre los 10 mW), relativamente baja para entornos reales. Por ello, si se va a transmitir el contenido a través de un emisor HPHT, es necesario altas potencias de emisión, como se ha visto anteriormente. Por lo tanto se hace necesaria una etapa de preamplificación y una etapa de potencia. La etapa de amplificación constará de un excitador (*driver*) que sea capaz de trabajar en el rango de frecuencias entorno a los 700 MHz, y también se necesitará el propio sistema de amplificación.

La etapa de amplificación además de recibir la señal del modulador FeMBMS, se encarga de proteger el transmisor en todo momento tanto por sobreexcitación como por VSWR (*voltage standing wave ratio*) excesivo, mediante circuitos de corte rápido. Igualmente, se deberá manejar el sistema de CAG/CAP (control automático de potencia) para conseguir una potencia de tranmisión estable. Se requerirá un módulo de precorrección digital adaptativo que asegura una buena linealidad, traducida en un buen MER *(modulation error rate)*. Finalmente, un amplificador con capacidad máxima de 1 Wrms (para señales OFDM) con una configuración redundante y balanceada que asegura la potencia de excitación necesaria.

La etapa de potencia constará de dos módulos de 3RU sumados mediante combinación híbrida dando una potencia máxima de 600 W. Cada módulo lleva incorporado dos etapas de potencia dobles en configuración balanceada, basados en transistores de estado sólido tipo LDMOS y en clase AB.

F. Raptor Q

Debido a que los mecanismos de procesado de señal en recepción y transmisión de las plataformas SDR no están completamente optimizados, se necesita una protección extra mediante mecanismos de corrección de errores en la capa de aplicación. Por esta razón en el piloto se ha incluido una implementación de AL-FEC (*aplication layer forward error correction*) basada en los códigos RaptorQ, en su versión más moderna y mejorada cuyas características superan en gran medida a los Raptor convencionales.

Estos mecanismos de protección y corrección de errores, que ya se implementan en la capa física de LTE, serán aplicados previamente a la transmisión del contenido por el transmisor SDR (capa de aplicación). Esto supone un nivel de protección extra al que ya se aplica en la capa física, aumentando el entrelazado temporal. Sin embargo, se debe tener en cuenta que su aplicación va ligada a una disminución de la capacidad del sistema y un aumento de la latencia, dos parámetros de calidad claves en las transmisiones *broadcast*.

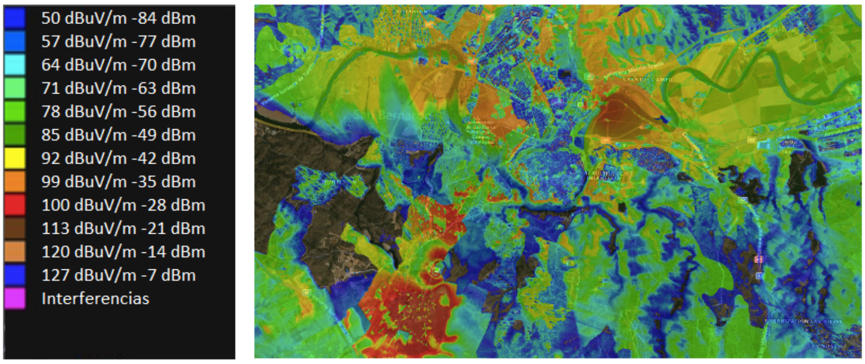

Fig. 4. Intensidad de campo en la ciudad de Toledo (dBμV/m)

Fig. 5. USRP B210

Estas características son ajustables usando dos parámetros de configuración: el periodo de protección y la tasa de codificación.

La tecnología RaptorQ genera paquetes adicionales o de reparación a partir de los datos originales y transmite una secuencia formada por ambos tipos. En recepción, la decodificación de los datos originales puede realizarse si el tamaño de la secuencia es ligeramente mayor a la original, independientemente de si los datos recibidos son originales o de reparación. Dentro de la arquitectura utilizada en el piloto, el codificador RaptorQ se incluirá en la etapa de distribución implementada por la UPV. De forma similar, el decodificador RaptorQ estará incluido dentro de la etapa de recepción también implementada por la UPV. La inclusión de los datos de reparación creados por RaptorQ añadirá un grado más de fiabilidad a la transmisión, además dado que RaptorQ presenta unas propiedades cercanas a las óptimas, el aumento de latencia introducido será mínimo y la capacidad se verá mínimamente reducida.

IV. Conclusiones

En este artículo se analiza con detalle la tecnología 5G Broadcast, mostrando las mejoras que se han ido introduciendo desde su incorporación en las especificaciones del 3GPP y a continuación las características principales de la tecnología. En este artículo también se explica el piloto de Red.es de Castilla-La Mancha, donde se retransmitirá contenido de TV y radio en tiempo real mediante una red 5G Broadcast.

Agradecimientos

Este trabajo se está desarrollando en la segunda convocatoria de proyectos piloto 5G de Red.es, bajo la financiación del Fondo Europeo de Desarrollo Regional (FEDER). Cuenta con el apoyo del proyecto europeo del horizonte 2020 (H2020) 5G-TOURS bajo la financiación N. 856950.

Referencias

[1] Radiotelevisione Italiana, “Trials on 5G broadcast technology in Aosta Valley during European Championships 2018,” *Available: http://www.crit.rai.it/CritPortal/progetti/?p=2342.com.*

[2] 5G TODAY, “Towards 5G Broadcast: Large-scale TV broadcasting” *January, 2019.*

[3] Rohde & Schwarz 5G Broadcast “5G Broadcast trials in China ” *Available: https://www.tvtechnology.com/news/rohde-schwarz-taking-part-in-chinas-5g-broadcast-trial.*

[4] Rohde & Schwarz 5G Broadcast “5G Broadcast trials in Barcelona” *Available: https://www.broadbandtvnews.com/2020/02/24/rohde-schwarz-cellnex-rtve-trial-5g-broadcast-in-barcelona/.*

[5] 3GPP “Technical Specification Group Services and System Aspects; Multimedia Broadcast/Multicast Service (MBMS) User Services; Stage 1 (Release 6) ” *January, 2004 Available: https://www.3gpp.org/ftp/Specs/archive/22_series/22.246/22246-600.zip/.*

[6] 3GPP “Technical Specification Group Services and System Aspects; Multimedia Broadcast/Multicast Service (MBMS) User Services; Stage 1 (Release 6) ” *June, 2008 Available: https://www.3gpp.org/ftp/Specs/archive/22_series/22.146/22146-900.zip./.*

[7] A. Shokrollahi, “Raptor Codes ””, in *IEEE Information Theory Workshop on Information Theory for Wireless Networks* , 2007, pags. 1-1.

[8] Qualcomm “RaptorQ Technical Overview” *Disponible: https://www.qualcomm.com/media/documents/files/raptorq-technical-overview.pdf.*

[9] T. Stockhammer, G. Teniou, G. Frederic “3GPP based tv service layer” in *Proc. International Broadcasting Convention (IBC), Amsterdam, The Netherlands September, 2017.*

[10] T. Stockhammer, I. Bouazizi, F. Gabin, JM Guyot, C. Lo, T. Lohmar and C. Thienot “Enhanced TV Services over 3GPP MBMS ” in *Proc. International Broadcasting Convention (IBC), Amsterdam, The Netherlands September, 2017.*

[11] D. Gomez Barquero, J.J. Gimenez and R. Beutler “3GPP Enhancements for Television Services: LTE-based 5G Terrestrial Broadcast ” in *Encyclopedia of Electrical and Electronics Engineering January, 2020.*

[12] D. He et al.“Overview of Physical Layer Enhancement for 5G Broadcast in Release 16 ” in *IEEE Transactions on Broadcasting, vol. 66, no. 2, pp. 471-480, June, 2020.*

[13] E. Garro, M. Fuentes, J.J. Gimenez, J.L. Carcel, et al. “Deliverable D3.2: Air Interface ” in *5G-Xcast July, 2019 Disponible: http://5g-xcast.eu/wp-content/uploads/2019/07/5G-Xcast_WP3_D3.2_v3.00_web.pdf.*

[14] 3GPP “WI - 911020 - New bands and bandwidth allocation for 5G terrestrial broadcast ” *Disponible: https://portal.3gpp.org/desktopmodules/WorkItem/WorkItemDetails.aspx?workitemId=911020.*

[15] Recomendación P.525 ITU-R, “Cálculo de la atenuación en el espacio libre ” *Disponible: https://www.itu.int/rec/R-REC-P.525/es.*

[16] Recomendación P.526 -15 ITU-R, “Propagación por Difracción” *Disponible: https://www.itu.int/rec/R-REC-P.526-15-201910-I/es.*

[17] ITU-R “Frequency and Network planning aspects of DVB-T2 ” *Disponible: https://www.itu.int/pub/R-REP-BT.2254/es.*

[18] OpenAirinterface Alliance “Home Page ” *Disponible: https://www.openairinterface.org/.*

[19] ORS “OBECA SDR Receiver ”*Disponible: https://www.broadbandtvnews.com/2021/02/19/ors-broadcast-develops-5g-receiver/.*

[20] National Instruments “USRP (Universal Software Radio Peripheral) Modelo B210 ” *Disponible: https://www.ettus.com/all-products/ub210-kit/.*

Virtual del 20 al 24 de septiembre de 2021

URSI 2021 Vigo XXXVI Simposium Nacional de la Unión Científica Internacional de Radio

Organiza: Escola de Enxeñaría de Telecomunicación Universida de Vigo

EL COMITÉ ORGANIZADOR DEL XXXVI SIMPOSIUM NACIONAL DE LA UNIÓN CIENTÍFICA INTERNACIONAL DE RADIO (URSI 2021)

CERTIFICA QUE:

D. ÁLVARO IBÁÑEZ LATORRE

Ha participado en dicho Simposium, organizado por la Universidad de Vigo y el Comité Nacional Español de URSI, y celebrado en modo ***on-line*** del 20 al 24 de septiembre de 2021. Además, ha presentado la ponencia titulada:

"Radiodifusión de contenidos 5G Broadcast"

En Vigo, 27 de septiembre 2021

D. Antonio Pino García ***D. Iñigo Cuiñas Gómez***

Presidentes del Comité Organizador de URSI 2021,

URSI 2021 Vigo