Open-field Emulation of Cooperative Relaying in LTE-A Downlink Using the GNU Radio Platform

Patricia Atungire(1), Talha Faizur Rahman(2) *Student member IEEE*, Fabrizio Granelli(2) *Senior member IEEE*, and Claudio Sacchi(2) (*) *Senior member IEEE*

(1)MTN Uganda, Plot 22, Hannington Road, Kampala (UGANDA) – E-mail: atungip@mtn.co.ug

(2)University of Trento, Dept. of Information Engineering and Computer Science (DISI), Via Sommarive 5, I-38123, Trento (ITALY) – Phone: +39-0461-283907, Fax: +39-0461-282093. E-mail: Talha.Rahman@unitn.it, granelli@disi.unitn.it, sacchi@disi.unitn.it

(*) *corresponding author*

ABSTRACT

Long Term Evolution-Advanced (LTE-A) plans the adoption of cooperative relaying in order to extend coverage and improve Quality-of-Service of future-generation mobile wireless networks. In the present paper, we propose the usage of open-source GNU radio tools to build a full-Software-Defined Radio (SDR) implementation of cooperative relaying for an LTE-A downlink connection. Experimental results demonstrate the effectiveness, tested in open-field, of the cooperative relaying in terms of augmented coverage and improved throughput, with a substantial added value provided by the SDR implementation in terms both of algorithmic flexibility and network reconfigurability.

I. INTRODUCTION

The growing requirement for wireless and mobile access on an “anytime-anywhere-any device” basis is pushing cellular networks to rapidly evolve towards improved capacity and lower access delay. While LTE is being deployed in several countries around the world, its evolution (LTE-Advance or LTE-A) is under standardization, with the target to extend coverage, improve Quality of Service and reduce power consumption in next generation mobile wireless networks.

In this scenario, given the high costs required for the deployment of a significant testbed, a relevant role (especially by the scientific community) is played by open source experimental platforms able to guarantee the possibility to perform significant emulations of next generation functionalities and technological solutions.

An example of advanced networking functionality that will be introduced in LTE-A is cooperative relaying. Although relay has been proposed very recently, three relay standards have been already specified for WiMAX and LTE, namely: IEEE 802.16j, IEEE 802.16m, and 3GPP LTE-A release 10 [1]. The basic idea behind cooperative relaying has been presented more than 30 years ago in the pioneering work of Cover and El Gamal [2]. Wireless systems using multicarrier transmission techniques may enable cooperative relaying in easier and more efficient manner, thanks to their improved flexibility in managing the available radio resources. This is the reason why LTE and LTE-A are considering the cooperative relaying as a pillar of their future realizations. Two very recent works investigated the use of cooperative relaying in LTE-A networks from a merely theoretical viewpoint. In [3], Alam, Mark and Shen formulated in terms of joint optimization problem relay selection and resource allocation for multi-user cooperative OFDMA networks. In [4], Li et. al. investigated the radio resource allocation for heterogeneous networks with cooperative relaying, where the relay nodes (RNs) with in-band backhaul act as micro base stations (BSs), able at serving the user equipment (UE) independently or cooperatively with the BSs. Both [3] and [4] focus on MAC layer optimization for LTE-A networks with cooperative relaying, but do not consider any experimental evaluation of the impact of cooperative relaying to end-to-end uplink/downlink connections. We think that a useful step-ahead would be represented by the emulation of this new functionality. A Software Defined Radio

(SDR) platform represents a viable approach to facilitate experiments on such next generation functionalities.

The SDR is built using reprogrammable Digital Signal Processors (DSPs) and General Purpose Processors (GPPs), whose functionality can be altered by varying some of their operational parameters in real time. These reprogrammable processors thus yield an increase in design flexibility and allow reconfigurable radio design [5].

This paper aims at presenting an experimental study of cooperative relaying, based on the SDR implementation and the open-field emulation of a relay-aided LTE-A downlink multi-user transmission. The SDR implementation has been realized by means of open source GNU Radio software tools [6] and the transmission emulation has been performed by using commodity PCs connected by means of low-cost programmable transceivers. Our configuration is based on two relays using decode and forward relaying methodology. The transmission parameters in terms of data rates and bandwidth have been properly scaled in order to cope with the hardware constraints. We focus in particular on the Physical Downlink Shared Channel (PDSCH) emulation to demonstrate the benefit attained with the inclusion of cooperative relaying into the downlink of a LTE-A like testbed.

II. RELATED WORK

GNU Radio represents a mature open source software development toolkit that provides signal processing blocks to implement software radios [6]. It can be used with readily-available low-cost external RF hardware (namely: Universal Software Radio Peripheral – USRP, supplied by Ettus Research [7]), or without hardware in mere software simulation environment.

GNU Radio project has attracted the interest of the research community working with Software Defined Radio since almost 10 years. In [8], a software receiver for the Emergency Weather Information Network (EWIN) fully implemented on a GNU Radio platform has been presented. In [9], GNU Radio tools demonstrated their capability to support the software integration of satellite radio localization and terrestrial communications in emergency scenarios by means of reconfigurable receiver architecture. Another recent work [10] deals with the integration of GNU Radio tools with adaptive antenna arrays. The results shown in [8-10] prove that GNU Radio environment can effectively support and emulate very complex (and safety-critical) communication scenarios, providing practical testbeds for advanced transmission and networking technologies. In the framework of SDR-based relaying implementation, it should be mentioned here the work of Zhang et. al. [11] where a cooperative testbed based on GNU Radio and USRP platform has been proposed and assessed. The focus of [11] was to evaluate the performance in terms of BER of cooperative relaying schemes and to discuss relay selection and rate adaptation. The PHY-layer of [11] has been implemented by means of a single-carrier GMSK modulation. In this framework, we can also mention the Open Relay Demonstrator [12], a GNU Radio-based platform that has been developed at Linköping University of Sweden. In such a platform, cooperative relaying has been implemented and tested using one relay node and amplify-and-forward relaying strategy.

The major difference between the literature works [11-12] and the one presented in this paper is the usage of the GNU Radio for emulating the introduction of cooperative relaying in a standard scenario, i.e. the OFDMA-based LTE-A downlink. No reference to current and/or forthcoming standards is indeed present in those state-of-the-art contributions.

III. THE EMULATED LTE-A SCENARIO

The considered wireless network scenario, whose configuration and topology is shown in Fig.1, is related to a cooperative relaying-aided LTE-A downlink.

The employed topology consists of one main transmitter, emulating the LTE-A BTS (namely: “Main node” in Fig.1), one User Equipment (UE) receiver (namely: “RX3” in Fig. 1) and two relay nodes (namely: “RX1” and “RX2” in Fig.1). Four USRP boards are used to enable wireless transmission and reception. All the boards are using half-duplex RF daughterboards.

The GNU Radio platform is used to implement via software the overall TX/RX and relaying functionalities of the emulated LTE-A scenario. In particular, the transmission of the Physical Downlink Shared Channel (PDSCH) is emulated. Without relaying, the main node sends information packets to RX1, RX2 and RX3 nodes (direct data connections of Fig.1). As soon as the relay functionality is activated, the main node achieves information about the channel status related to the two possible relay paths: the first one passing through RX1 and the second one passing through RX2. Then, depending on channel condition, single or double relay connections are activated. In our emulation, it has been decided that the Channel State Information (CSI) is sent from the UE and the relays as feedback to the main transmitter node via Internet WLAN (IEEE 802.11n connections of Fig.1) and stored in a database (namely: “MySQL”) resident on the Main node.

The “MySQL” database is also used to support the half-duplex operation of the daughterboards, by providing a sort of network synchronization. This represents an essential part of the implementation, since the boards can operate only in half-duplex mode; therefore it is necessary to ensure that the Main node does not transmit while the relay node is retransmitting information to the UE. This is achieved by setting the status of each node so that is indicative of what operation is going on. Additionally, a separate thread is used to monitor the different states that each node assumes. In this way, the synchronization process does not hinder the data transfer, but it simply monitors when the data is transferred to avoid data loss due to half-duplex operation.

FIGURE 1

IV. HARDWARE SETUP OF THE EMULATION TESTBED

The hardware components of the emulation scenario of Fig.1 can be subdivided into two main categories:

- *Computing machines*, devoted at running GNU-radio baseband signal processing tasks, including: data modulation-demodulation, synchronization, channel estimation, radio resource management, relaying, etc. The computing devices are commodity PCs running UBUNTU 11.10 OS with the last release of GNU Radio platform installed on each PC;
- *Communication interfaces*, devoted at forming the waveforms transmitted on the air and to receive them; they perform the following operations: A/D and D/A conversion, RF generation and wave transmission. The SDR communication interfaces are the USRP N210 and USRP1 boards. All the boards embark XCVR2450 half-duplex RF daughterboards.

Each node of Fig.1 consists of a computing device connected to a communication interface. More in detail, two USRP1 boards are used for the Main node and for UE, and two USRP N210 boards are used for the relay nodes. The transmission trials have been performed in the 5GHz ISM band in order to make easier the connection between USRP1 and USRP N210 (some connection problems have been noticed using the 2.4GHz ISM band).

Although USRP N210 board is a more powerful communication interface, as compared to the USRP1, the hardware constraints imposed us to downscale rate and bandwidth of the OFDMA downlink transmission with respect to real LTE-A parameters. The link parameters used for the emulations are shown in Tab.1.

TABLE 1

V. SDR IMPLEMENTATION OF THE OFDMA DOWNLINK WITH COOPERATIVE RELAYING

A. Generalities about GNU Radio-based implementation

GNU Radio framework allows the use of Python and C++ for the software development of signal processing algorithms addressing specific communication tasks. GNU Radio offers to users a wide library of built-in blocks, written in C++, which are used to implement computationally-demanding processing tasks, for example: filtering, data modulation and demodulation, equalization, synchronization, etc. Additionally, developers are allowed to design their own blocks in C++ in order to implement new functionalities. Python is generally used to interconnect the blocks implemented in C++ and to implement some less computationally-intensive tasks. Regarding the scenario of Fig.1, some modifications have been made to some available C++ modules, new modules have been implemented to support cooperative relaying, and, finally, database functionality has been added to the Python implementations. We have chosen to use the non-graphical user interface in developing the software so as to increase the flexibility of the design.

B. OFDMA transmitter implementation

Within GNU Radio, a software implementation of the single-user OFDM transmitter and receiver is already available. For this work, we modified such implementation to extend the functionality of the transmitter and receiver to multi-user OFDMA, which is used in the LTE-A downlink. The following assumptions have been considered during the implementation:

- Multipath channel is flat over each subcarrier and slowly varying;
- Number of subcarriers is the same for each node;
- The outgoing queues for each user are full;
- All resource blocks are available for a given transmission burst;
- The users are already scheduled and they are homogeneous.

The flowchart (I) of Fig.2 illustrates the interconnection between the blocks in the “main” of the transmitter, namely “Benchmark_tx.py” file. The “AWGN Noise source” module is optional and used only for software simulations. The input data are generated for all users and assembled in packets, whose payload size has been specified in Tab.1. Then the data are passed to the module "Transmit_Path.py" one packet at a time.

The module "Transmit_Path.py", whose structure is exploded in the flowchart (II) of Fig. 2, encloses a core sub-module that is named "Modulator.py". This sub-module, detailed in flowchart (III) of Fig.2, manages the baseband OFDMA transmission tasks, in particular: radio resource allocation, subcarrier mapping, IFFT, and cyclic prefix addition. Radio resource allocation and subcarrier mapping are performed by the "Mapper_block" (written in C++). Within this script, three main functions have been created:

1. The incoming user data are redirected into separate queues for each user prior the transmission.
2. Each user is assigned to a Resource Block (RB) on the basis of the Maximum Sum Rate (MSR) criterion shown in [13]. The choice of MSR as radio resource allocation strategy is motivated by its low complexity and implementation ease. In this scheme, we acquired feedback on the CSI from the "MySQL" database. The feedback information is the channel gain measured over each subcarrier. The information on the assigned resource blocks is then passed down to the mapper module. In such a way, it is possible to assign to each user those subcarriers in a resource block over which that user has the best channel gain (greedy optimization for the MSR resource allocation scheme is adopted). To ensure fairness, we allocate an equal number of resources to each user before providing more resource blocks to the best performers.
3. The user messages are retrieved, one at a time, from each of the user queues, and mapped onto the delineated available subcarriers in accordance with the resource allocation scheme in use.

The multi-user signal, generated by the OFDMA baseband transmitter, is finally amplified and sent onto the wireless channel by means of the RF modules connected to the USRP boards.

FIGURE 2

C. OFDMA receiver implementation

The single-user OFDM receiver implementation available in GNU Radio performs the basic OFDM demodulation tasks. In our implementation, such a receiver has been modified in order to encompass the functionalities required both by multi-user OFDMA detection and cooperative relaying. The flowchart of Fig. 3 shows the interconnections among the software blocks required to implement the OFDMA downlink receiver.

The core module "ofdm_receiver.py" (see flowchart (IV) of Fig.3) is left unchanged with respect to the standard OFDM GNU Radio receiver except the "Frame_acquisition" block, written in C++. In this block, the channel estimation is performed. We transmit these estimates to the next block, i.e. the "frame_sink" C++ block (see flowchart (III) of Fig.3). The input signal is then demodulated according to the intended receiver Id. The "Frame_sink" block performs this operation in a series of states like a state machine (see Fig. 4). In the "control handling" state, the receiver extracts the correspondingly allocated resource blocks sent alongside the receiver Id and then extracts the data symbols corresponding to the given Id. In this way, only the receiver Id to which the transmitter sent the information is enabled to decode the expected data. Once the resource blocks are determined, the "control handling" state sets the receiver to acquire the transmitted information from the subcarriers in the assigned resource blocks and then reassigns the machine to the previous state. The operation then continues on through the other states until the next start of frame flag is received, regardless of the specific state of "frame_sink". Indeed, whenever it receives a "Synch rxd" flag, the state machine resets to the "Synch search" state again, starts the detection process afresh and the previous packet is dropped.

FIGURE 3

FIGURE 4

Additionally, in this block, we implemented the capability of the receiver to insert the estimated channel gains into an object that could be accessed from Python. Within the Python implementation of the demodulator, we placed a thread running in parallel to the main flow graph program that pushes an update into the database about the receiver channel state information whenever a new channel estimate is available. Thus, for every packet, as soon as the preamble is received, this thread inserts an update about the channel feedback to the "MySQL" database. In such a way, the CSI updates do not slow down the USRP device during data demodulation, as they are executed and stored concurrently. Such an added functionality will be exploited by the cooperative relaying.

D. Cooperative relaying implementation

The adopted relay strategy is of the type 1: "decode and forward". Such a strategy was also included in the LTE Release 8 specification. This protocol enables the relay node to decode the received signals and re-encode them for transmission to their destination. Decode and forward relaying exhibits higher performance gain with respect to other relaying alternatives at the cost of an increased algorithmic complexity. Since we are considering a downlink scenario, the node chosen as relay is unidirectional. Moreover, since the RF daughterboard used by the USRP transceiver is half-duplex, the relay is implemented such that only after having received the data, it can retransmit them to the UE. From the emulation viewpoint, this means that we must adhere to the rules included in the LTE-A protocol in order to avoid self-interference at the type 1 in-band relay station. The relay receives information over an assigned group of subcarriers and, only after having completed the demodulation tasks, it is enabled to retransmit this information to the UE.

The cooperative relaying algorithm is described in the flowchart of Fig. 5. Each relay node is designed to start its functionality as a receiver. For sake of simplicity in emulation, the system is implemented to retransmit to a single UE. In order to facilitate cooperative relaying, the design included a further feature where the relay could retransmit or not depending on a parameter contained in the control information sent by the transmitter along with data. This parameter describes the channel performance over the path connecting the relay node with the UE. Depending on the value of this parameter, the relaying is selected or not, as shown in Fig. 5. If both relay paths are evaluated as appropriate, the double relaying is issued and two relay nodes are both activated. A selective diversity combining of the two relay paths is performed by selecting the path that exhibits the highest Signal-to-Noise ratio.

FIGURE 5

VI. EXPERIMENTAL RESULTS

The experiments shown in this section have been carried out in indoor environment, more precisely in two corridors located in the premises of the Department of Information Engineering and Computer Science (DISI) at University of Trento. The first corridor (namely: "wide corridor") is wider and longer than the second corridor (namely: "narrow corridor"). More precisely, the dimensions are 3mt.x55mt. for the wide corridor and 1.70mt.x30mt. for the narrow corridor respectively. Both the environments are open spaces with no furniture except for occasional persons passing through the corridors. In Fig. 6a and 6b, some pictures depicting the two considered emulation scenarios have been shown.

The Main node has been placed in a fixed position in the corridors. The UE terminal and relay nodes position have been then varied from close to the transmitter to further away from the transmitter. The relay-UE distance was also varied in the same manner. The relay(s) has (have) been successively placed at a 2 mt. distance from the UE for each emulation trial. The relay-UE distance of 2 mt. was finally chosen because it was observed from some ad-hoc experimental trials that this was the optimal distance to improve the coverage from the Main node to the UE.

FIGURE 6a

FIGURE 6b

The performance metric chosen to assess the effectiveness of the proposed relaying methodology is the Packet-Error-Rate (PER) vs. link distance, measured at the UE terminal. Since the channel conditions may vary in time, several emulation runs have been performed and the sample average of the observed PER values has been computed. The confidence intervals for the statistical averages are 5% of the sample average for the wide corridor and 10% for the narrow corridor, respectively, with a confidence level of 95%. The emulation results have been checked against the theoretical performance evaluation of [14], where analytical expressions about BER achieved by a downlink transmission with single and double cooperative relaying and binary modulations have been derived. PER curves have been obtained by using the approximation: $PER = 1-(1-BER)^L$, being L the payload size. The AWGN Signal-to-Noise ratio vs. distance has been computed using the pathloss model of [15], while the variance of the multipath attenuation has been estimated using the CSI data stored in the "MySQL" database.

PER results are shown in Fig. 7 (wide corridor) and Fig. 8 (narrow corridor), respectively. In Fig. 7, results are shown both for single relaying (namely: "UE from relay1") and double relaying (namely: "UE from both relays"), whereas in Fig. 8 only double relaying results are plotted. Channel conditions have been revealed more severe in the narrow corridor scenario, as clearly evidenced by the achieved higher PER values. Anyway, experimental curves demonstrate that the introduction of the relay nodes significantly increases performance and coverage of the system by substantially decreasing PER with respect to the direct connection main node – UE (namely: "UE without relaying") for almost all the link distances and conditions considered in the emulations. Indeed, the usage of relays enables the system to operate nearly at the same performance level as in the case of a short-range direct link between transmitter and receiver, demonstrating the potential provided by this technology. The comparison of emulation results with theoretical results looks interesting. The trend of PER curves is rather similar, but analytical results represent lower bounds on real results. Indeed, analytical evaluation does not take into account non-controllable non-idealities of the open-field transmission, which are: high levels of in-band interference, real multipath propagation, nonideal and nonlinear behavior of low-cost communication hardware, finite precision arithmetic, etc.

FIGURE 7

FIGURE 8

The achieved results underline the potential of experimenting with cutting-edge technology using open source hardware and software platforms. In the presented test case, the availability of the GNU Radio software defined radio platform enabled to successfully analyze a technology in academia by performing real experiments. This allowed deriving interesting insights related to the usage of a real communication channel – which is quite hard to obtain through software simulations or analytical works. The drawback of this solution is related to the fact that in many cases – like the case presented in this paper – experiments can be performed only at a scaled-down level, since the low-cost hardware supporting GNU Radio is limited in performance. Nevertheless, even in the presence of some limitations, the results are significant and useful for the interested community.

VII. CONCLUSION

This paper has demonstrated the usage of an open source platform (GNU Radio) to emulate a system incorporating cooperative relaying in an LTE-A downlink. The system was developed and tested to measure and demonstrate the benefits of including cooperative relaying. Additionally, this system has resulted into a hardware/software testbed that can be used for testing the impact of other typologies of resource allocation on system performance and implement different relaying techniques. The experimental results achieved in open-field, although considering the necessary downscaling in terms of bandwidth and bit-rate in order to cope with hardware constraints, fully confirmed the effectiveness of the cooperative relaying in LTE-A-like scenarios. Moreover, the proposed analysis may represent a preliminary study about the viability of the SDR implementation of cooperative communication tasks in ongoing real-world LTE-A applications. Future work will be aimed at using the same platform to study cooperative relaying in the LTE-A uplink.

VIII. ACKNOWLEDGEMENTS

Authors wish to thank Emmanuel R. Ssebaggala for his valuable contribution in software implementation and results collection.

REFERENCES

[1] K. Loa, C-C. Wu, S-T. Sheu, Y. Yuan, M. Chion, D. Huo, L. Xu, "IMT-Advanced Relay Standards," *IEEE Comm. Mag.*, Aug. 2010, pp. 40-48.

[2] T.M. Cover, A.A El Gamal, "Capacity Theorems for the Relay Channel," *IEEE Trans. on Info Theory*, vol.25, no.5, Sept. 1979, pp. 572-584.

[3] M.S. Alam, J.W. Mark, X. Shen, "Relay Selection and Cooperative Resource Allocation for Multi-User Cooperative OFDMA Networks," *IEEE Trans. on Wireless Comm.*, vol.12, no.5, May 2013, pp. 2193-2205.

[4] Q. Li, R.Q. Hu, Y. Qian, G. Wu, "Intracell Cooperation and Resource Allocation in a Heterogeneous Network with Relays," *IEEE Trans. on Veh. Tech.*, vol.62, no.4, May 2013, pp. 1770-1784.

[5] T. Ulversoy, "Software Defined Radio: Challenges and Opportunities," *IEEE Comm. Surveys & Tutorials*, vol.12, no.4, 4th quarter 2010, pp. 531-550.

[6] D. Valerio, *"Open Source Software-Defined Radio: A survey on GNU Radio and its applications,"* Technical Report, issued by Forschungzentrum Telekommunikation, Wien (A): 2008, http available at: http://userver.ftw.at/~valerio/files/SDRreport.pdf

[7] Ettus Research: products data sheets: https://www.ettus.com/product

[8] E. Valles, K. Tarasov, J. Robertson, E. Grayver, K. King, “An EMWIN and LRIT Software Receiver using GNU Radio”, *Proc. of IEEE Aerospace Conf. 2009*, Big Sky (MT), March 2009, pp. 1-11.

[9] M. Panizza, C. Sacchi, J. Varela-Miguez, S. Morosi, L. Vettori, S. Digenti, and E. Falletti, “Feasibility study of an SDR-based reconfigurable terminal for emergency applications,” *Proc. of 2011 IEEE Aerospace Conf.*, Big Sky (MT), March 5–12 2011, pp. 1-18.

[10] M. Donelli, C. Sacchi, “Implementation of a Low-Cost Reconfigurable Antenna Array for SDR-based Communication Systems,” *Proc. of 2012 IEEE Aerospace Conf.*, Big Sky (MT), March 3–10 2011, pp. 1-7.

[11] J. Zhang, J. Jia, Q. Zhang, E.M.K Lo, “Implementation and Evaluation of Cooperative Communication Schemes in Software-Defined-Radio Testbed,” *Proc. of IEEE INFOCOM 2010*, pp.1-9.

[12] *“Open Relaying Demonstrator”*, http available at: https://www.cgran.org.

[13] A. Ghosh et al., *“Fundamentals of LTE,”* Pearson Education Inc., Boston (MA): 2011.

[14] P. Liu, I.M. Kim “Average BER Analysis for Binary Signallings in Decode-and-Forward Dissimilar Cooperative Diversity Networks,” *IEEE Trans. on Wireless Comm.*, vol.8, no.8, Aug. 2009, pp. 3961 – 3968.

[15] Y. Wang, W.J. Lu, H.B. Zhu, “An Empirical Path-Loss Model for Wireless Channels in Indoor Short-Range Office Environment,” *Int. Jour. of Antennas and Propagat.*, vol.2012, Article ID 636349, pp. 1-7.

BIOGRAPHIES

Patricia Atungire is currently a charging systems Engineer at MTN (Uganda). She holds an MSc Telecommunications Engineering degree from the University of Trento, Italy and a First Class Honors BSc Telecommunications Engineering degree from Makerere University, Uganda.

Talha Faizur Rahman is a PhD student at the ICT School of University of Trento (Italy). He received his BSc in Electrical and Electronic Engineering from Islamic University of Technology, Dhaka Bangladesh and his MSc in Telecommunication Engineering from University of Trento. His research interests include LTE-A (physical layer aspect, radio resource management), cooperative MIMO, Cognitive Radios, etc.

Fabrizio Granelli (M'01, SM'05) is Associate Professor at the University of Trento (Italy) and coordinator of the networking lab. He is IEEE ComSoc Distinguished Lecturer for 2014-15. He is author of more than 140 international papers, focused on network modeling, wireless networks, green networking and smart grid communications.

Claudio Sacchi (M'01, SM'07) is Assistant Professor at the University of Trento (Italy). He received the Laurea degree in Electronic Engineering and the PhD in Space Science and Engineering from University of Genoa, Italy. His main research interests include multi-user communications, satellite communications, software-defined radios, etc.

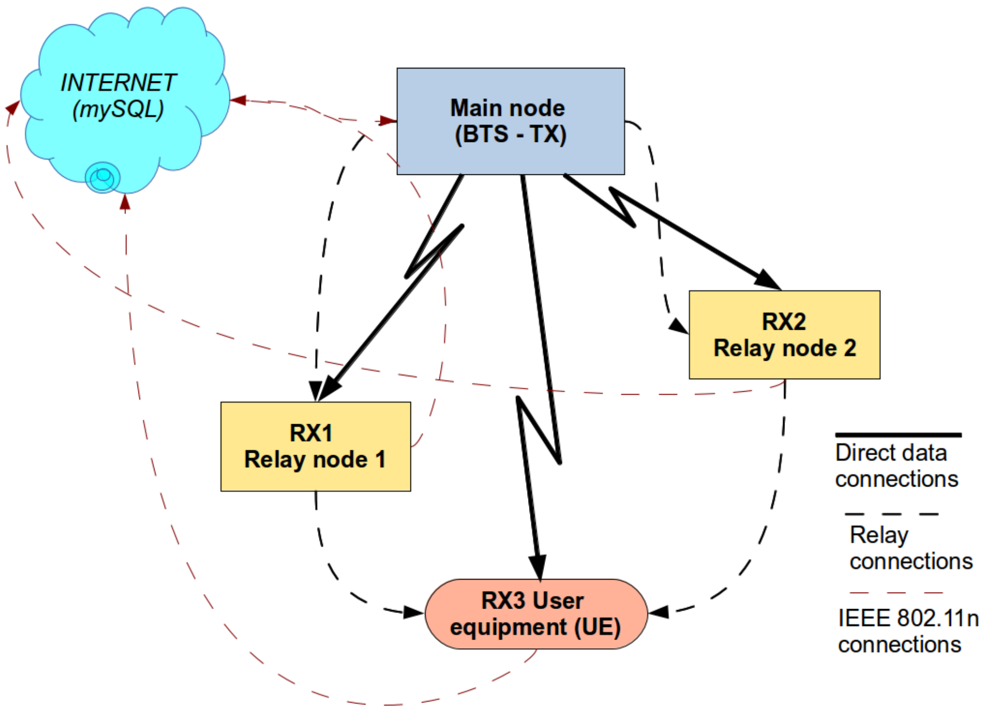

Figure 1: *emulation scenario: configuration and topology.*

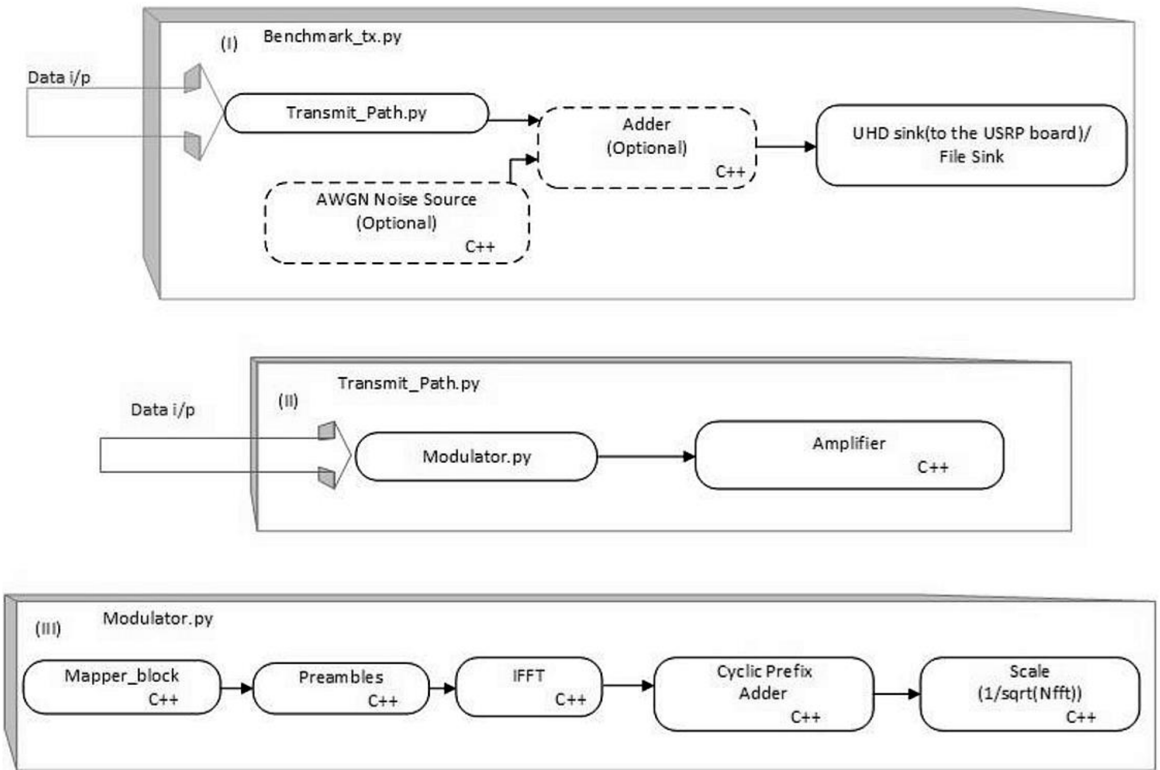

Figure 2: *software implementation of the OFDMA downlink transmitter.*

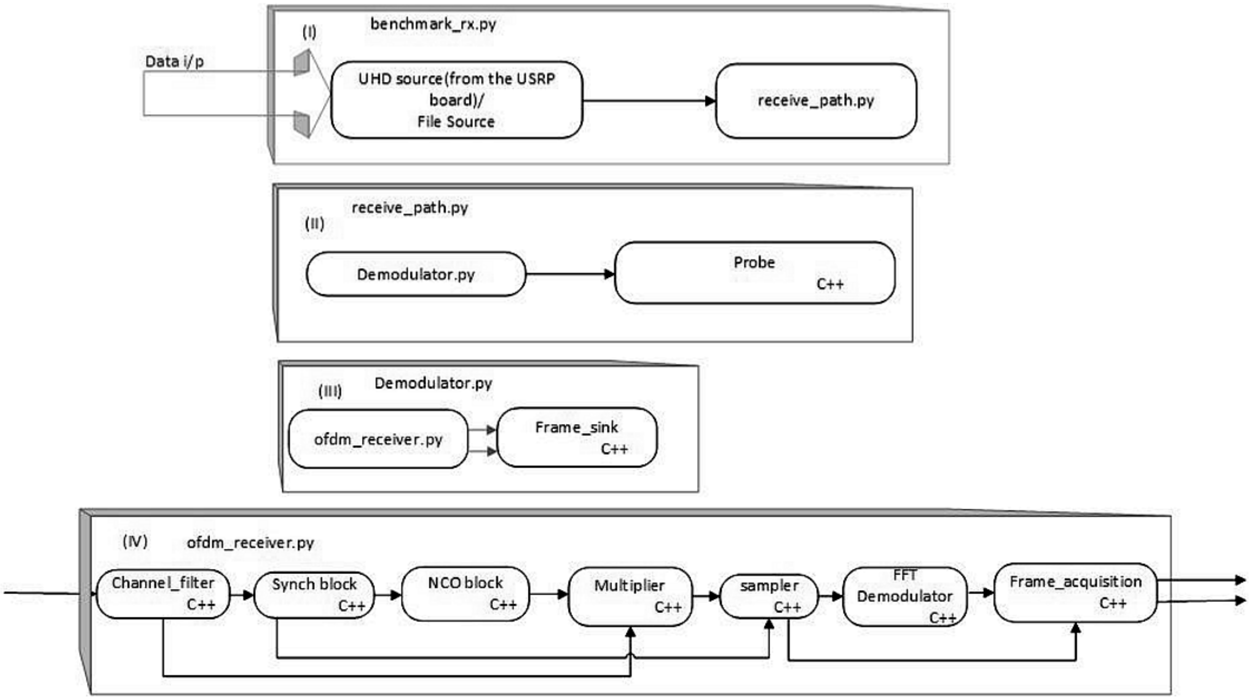

Figure 3: *software implementation of the OFDMA downlink receiver.*

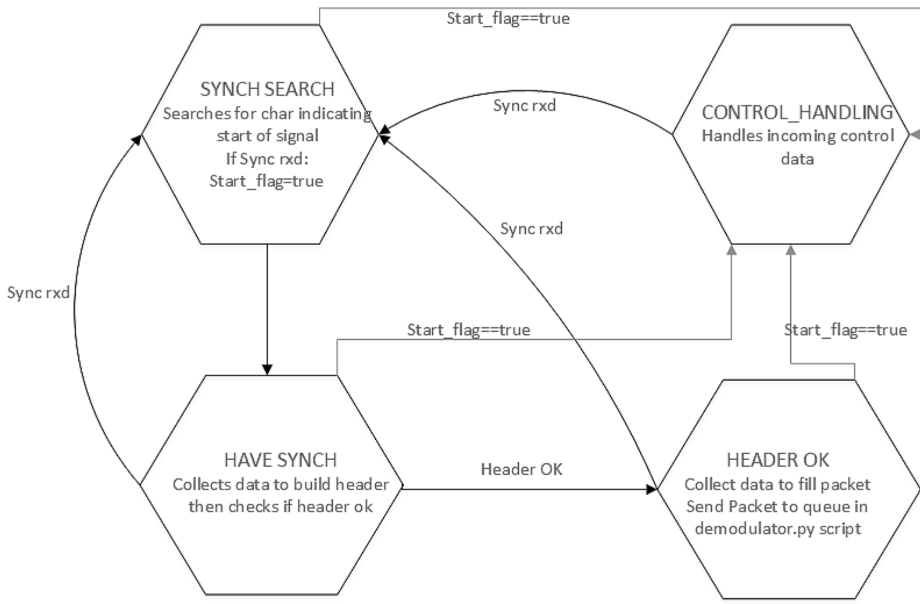

Figure 4: *frame synchronization state machine implemented in "frame_sink" block.*

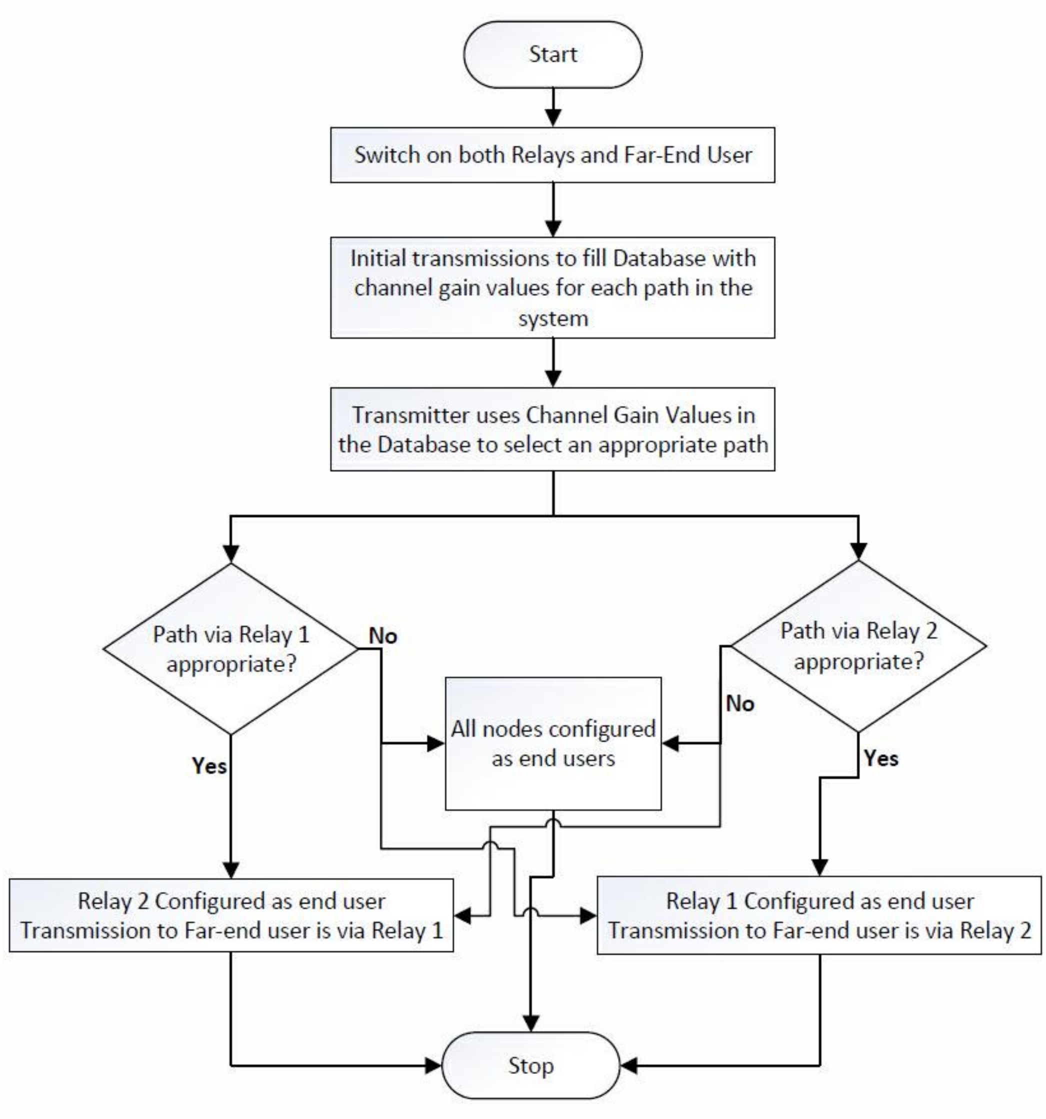

Figure 5: *flowchart of the cooperative relaying algorithm.*

(a)

(b)

Figure 6: *open-field emulation scenarios: (a) wide corridor, (b) narrow corridor.*

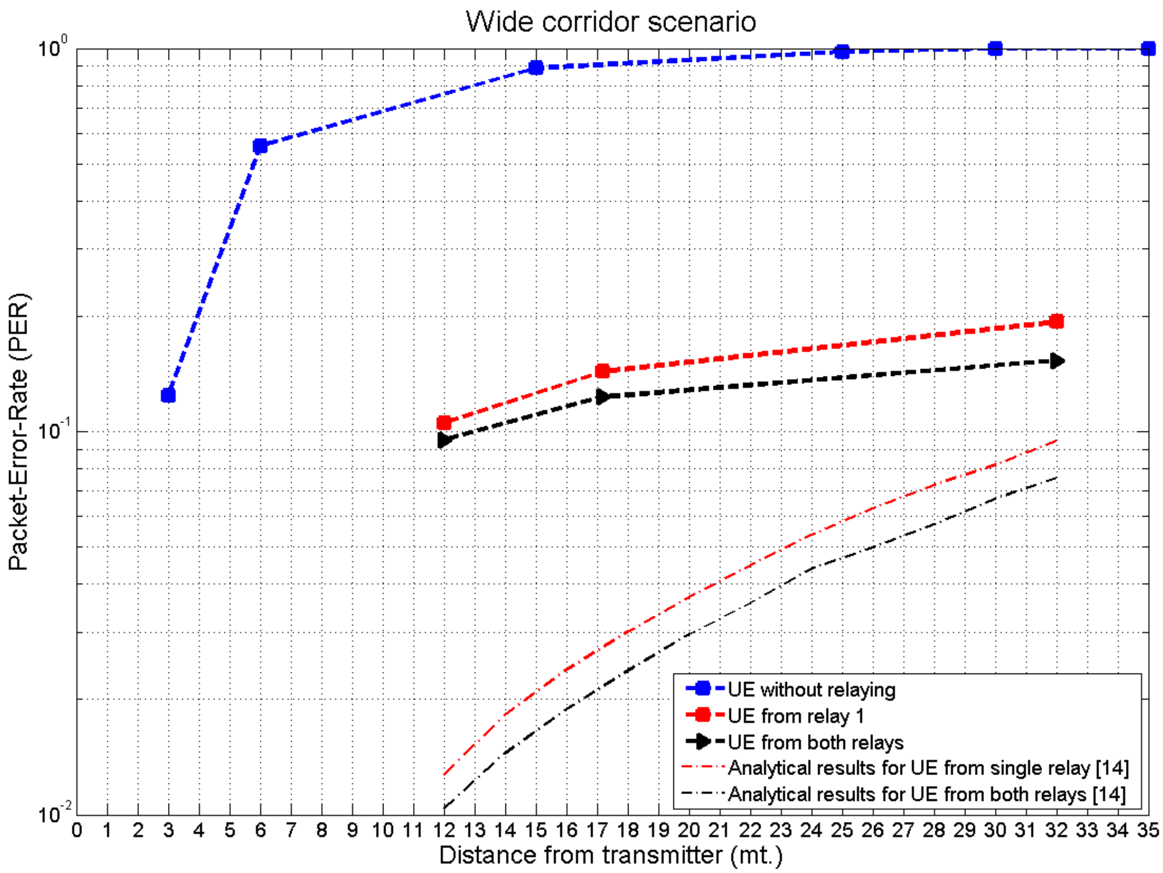

Figure 7: *emulation results in terms of PER vs. link distance (wide corridor).*

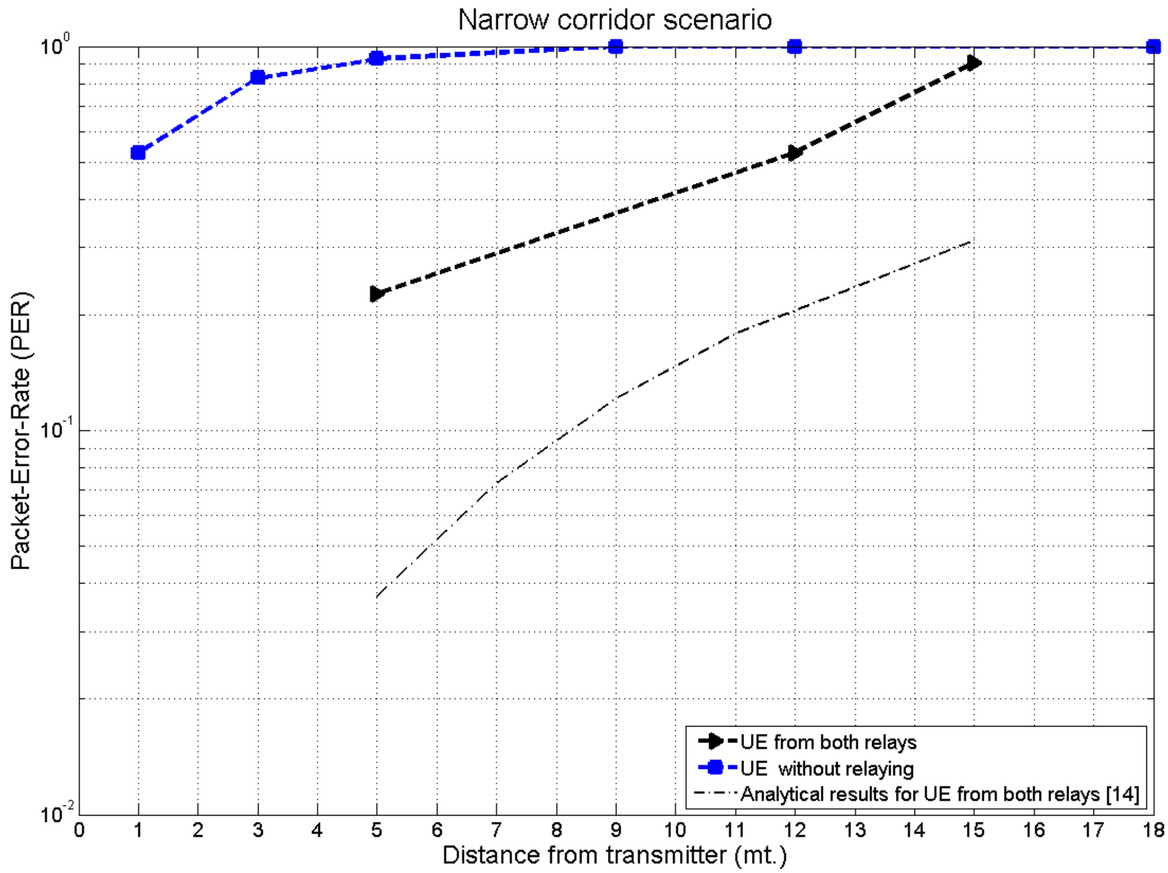

Figure 8: *emulation results in terms of PER vs. link distance (narrow corridor).*

Table 1: *emulated system parameters.*

PARAMETER	NUMERICAL VALUE
Subcarrier number	128
Max. number of users	3
Size of payload (Bytes)	400
Size of transmitted data per user (Mbytes)	1
Signal bandwidth (KHz)	500
Modulation scheme	QPSK