



HAL
open science

Physical-layer Network Coding: A Heterogeneous Software-Defined-Radio based Implementation

Mohammed Aissaoui, Chiraz Houaidia, Adrien van den Bossche, Thierry Val,
Leila Azouz Saidane

► **To cite this version:**

Mohammed Aissaoui, Chiraz Houaidia, Adrien van den Bossche, Thierry Val, Leila Azouz Saidane. Physical-layer Network Coding: A Heterogeneous Software-Defined-Radio based Implementation. 11th International Conference on Performance Evaluation and Modeling in Wired and Wireless Networks.(PEMWN 2022), IFIP; IEEE, Nov 2022, Roma, Italy. hal-03772162

HAL Id: hal-03772162

<https://ut3-toulouseinp.hal.science/hal-03772162v1>

Submitted on 8 Sep 2022

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

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

Physical-layer Network Coding: A Heterogeneous Software-Defined-Radio based Implementation

Mohammed AISSAOUI

ENSI(University of Manouba),IRIT(UT2J)
mohammed.aissaoui@(ensi-uma.tn/univ-tlse2.fr)

Chiraz HOUAIDIA

ENSI-University of Manouba
chiraz.houaidia@ensi-uma.tn

Adrien VAN DEN BOSSCHE

IRIT-University of Toulouse - Jean Jaurès
vandenbo@univ-tlse2.fr

Thierry VAL

IRIT-University of Toulouse - Jean Jaurès
thierry.val@univ-tlse2.fr

Leïla AZOUZ SAIDANE

ENSI-University of Manouba
leila.saidane@ensi-uma.tn

Abstract—Collisions in multi-hop wireless networks downgrade the system’s performance and affect its reliability. Therefore, most of the existing solutions aim to avoid or reduce the interference cases. Counter to all previous works, the newly proposed technique called Physical-layer Network Coding (PNC) transforms the collision’s dilemma into an advantage in such a manner that multiple transmissions occur simultaneously and end nodes can extract the superimposed signals. Although there have been many theoretical studies on PNC efficiency, there have been few experiments with pure and heterogeneous PNC schematics. This paper presents the first heterogeneous PNC Software Defined Radio based implementation with a brief discussion of the main challenging related issues. Obtained results proved the efficacy of the proposed solution in indoor environment and serve as a proof for the practicability of PNC in heterogeneous real-time applications.

Index Terms—Physical-layer Network Coding (PNC), GNU Radio, Software Defined Radio (SDR), Two-Way-Relay-Channel (TWRC), Orthogonal Frequency Division Multiplexing (OFDM).

I. INTRODUCTION

Multi-hop wireless networks, also called ad-hoc networks, are wireless, mobile and potentially large-scale networks. They have been designed to break free from traditional wired networks and are distinguished by their ability of self-organization and to overcome any infrastructure. This type of network is nowadays a potential candidate for the collection networks of the Internet of Things (IoT).

Nevertheless, as the capacity of these networks becomes particularly limited when the number of users increases, the scientific community is rethinking the basic principles of transmission in order to adapt this kind of networks to a wider range of application domains.

For instance, long considered one of the most challenging issues in multi-hop wireless networks, interference management has recently undergone a notable paradigm shift. While interference management is traditionally carried out by the access layer protocols whose objective is to avoid interference between users, it is now possible to exploit the interference thanks to new advanced communication techniques called Physical-layer Network Coding (PNC) [1].

This new paradigm authorizes, under certain conditions, several simultaneous transmissions from several source transmitters and in the same frequency band to the same receiver by allowing the source and destination nodes to deduce the signal resulting from the “collision”. This, of course, improves spatial reuse in wireless networks by increasing the number of simultaneous transmissions.

The power of the PNC technique could be illustrated using the network model known as “Two-Way Relay Channel (TWRC)”, as shown in Figure 1 [2]. In the TWRC network, two nodes (N_1 and N_2) exchange data via a relay node (R). With the half-duplex constraint and using the traditional store-and-forward scheme without interferences, the exchange of two packets (P_1 and P_2) between the two edge nodes N_1 and N_2 via the relay node R takes four time slots. However, using the PNC technique, the two nodes N_1 and N_2 send their data (P_1 and P_2 respectively) to the relay simultaneously in one time slot. The relay, then, maps the received data into PNC data ($P_1 XOR P_2$) using *eXclusive OR*. In the second time slot, the relay broadcasts the received PNC data ($P_1 XOR P_2$) to the edge nodes N_1 and N_2 . Each source node decodes the PNC data and extracts the other node’s original data by applying the *XOR* between its memorized data and the received PNC data.

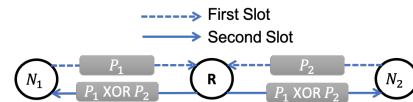


Fig. 1: TWRC Network.

Since the PNC requires the two edge nodes to be perfectly synchronized (at the symbol level), its implementation is very challenging. Therefore in the literature, there are at most theoretical demonstrations or numerical simulations of the PNC technique under the TWRC network compared to a few real-channel implementations. These implementations use, almost, a simplified and homogeneous Software Defined Radio (SDR) hardware.

This paper presents the first PNC implementation under the TWRC network using heterogeneous SDRs. We present the challenges of the heterogeneous SDRs and the solutions proposed to overcome them.

In the following, Section 2 gives an overview of the existing TWRC PNC implementations and their limitations. Section 3 and 4 present the proposed prototype of the PNC implementation and the solutions proposed for the heterogeneous SDR challenges, respectively. Section 5 shows the results of our TWRC PNC implementation and their analysis. Finally, section 6 concludes our work and outlines future works.

II. RELATED WORKS

In current practice, PNC based wireless network evaluation is mainly performed either through theoretical analysis or software simulations. These approaches lack the fidelity and flexibility needed for accurate and realistic evaluation of a wireless network [12]. Software simulations, mainly ns-2/3, typically use simplistic physical layer modeling that neglects and ignores various effects such as nonlinearity, filtering and inter modulation by hardware. However, with the growing proliferation of commercial off-the-shelf software-defined radios (SDRs), it has become feasible to deploy a multi-hop, multichannel system and to validate the throughput gain of the PNC technique in a real-world real-time context.

In this research focus, there have been various efforts to develop either device level SDR solutions or isolated protocol-level solutions. But the main challenging issue was the complicated processing at the relay node because of the simultaneous transmission of edge nodes. In fact, decoding techniques used for PNC at the relay node, commonly assume perfect synchronization of the two signals received from edge nodes (e.g., user transmissions begin and end at the same time, perfect frequency translation, etc.).

Furthermore, the relay node must perform channel estimation of each signal before the reception of those signals.

One of the first implementations of NC was proposed in [5], where authors developed a simplified Analog Network Coding (ANC). The basic idea of ANC is that the relay node simply amplifies then retransmits the received superimposed signals without coding. The proposed implementation was based on a SDR platform formed by the Universal Software Radio Peripheral (USRP) [3] with *RFX2400* boards as the hardware and the open source GNU Radio [4] as the signal processing software. The main drawback with this implementation is the error propagation as the relay amplifies the noise along with the signal when forwarding it to edge nodes.

Unlike the work in [7] which operates only in offline manner, the first real-time PNC implementation was reported in [6], based on USRP N210 with *XCVR2450* boards. Authors, in this proposal, used frequency division multiplexing (FDM) to separate the signals from edge nodes to the relay from those in the opposite direction. To overcome some technical challenges such as synchronization and channel estimation the authors used orthogonal frequency division multiplexing

(OFDM) with beacons. PHY-layer and MAC-layer error controls were also introduced to ensure more reliability. However, this implementation requires a change in the frame format and a balance between the power of the end nodes' signals received at the relay.

In [5], the authors propose and implement a TWRC real-time Physical-layer Network Coding technique called RPNC. RPNC offers tighter coordination of the two edge nodes, real-time PNC decoding at the relay, and a PNC-compatible re-transmission scheme to ensure the reliability of packet delivery. To improve the proposed TWRC PNC system, in [6], the authors improve the RPNC to support real TCP/IP applications. This improvement allows the proposed RPNC implementation to be used in real-world applications.

Different from the previous implementations, the proposed implementations in [7] and [8] used one antenna to transmit and receive in half-duplex mode. Therefore, the authors add a coordination layer or a static MAC layer to coordinate the transmissions of the edge nodes and the relay.

Authors in [9] used the full-duplex and the Multiple Input Multiple Output (MIMO) for the transmissions and the receptions to avoid switching between transmit and receive mode when using one antenna in each SDR hardware.

Recently, in [10] and [11], the authors propose a full-duplex TWRC PNC system, where each SDR node (relay or edge node) can transmit and receive at the same time using two antennas. The proposed implementation uses the bladeRF [12] as hardware and the GNU Radio software platform to send and receive multimedia content.

As we can see from the state of art, the proposed implementations of the TWRC PNC systems are rare compared to the different propositions relayed to mathematical analysis or simulations based evaluation.

According to our research in this field [13], this rarity in the TWRC PNC implementation is due to the following reasons:

- **First:** The high prices of the SDRs for the researchers. The prices of some available SDRs are presented in Table I. Remember that the higher the specifications and the options the SDR provides, the higher the price is.

SDR's Name	Price	Web site
USRP B210	1811.00 USD	https://www.ettus.com/all-products/ub210-kit
Adalm-Plato	506.39 USD	https://www.passion-radio.fr/satellite-go-100/adalm-kit-100-948.html
HackRF One	389.95 USD	https://www.sparkfun.com/products/13001
RTL XYNC	4300 USD	https://www.rtl-sdr.com/xync-a-massive-mimo-sdr-with-up-to-32x32-tx-rx-channels/
BLADE RF 2.0	658.44 USD	https://lab401.com/fr/products/blade-rf-sdr-2-micro-x4
LimeSDR mini	272.91 USD	https://www.passion-radio.com/sdr-transceivers/limesdr-mini-667.html

TABLE I: Prices of the different SDRs

- **Second:** Some SDRs, such as HackRF and RLT, that have one antenna to transmit and receive and which could be used only on half-duplex, are not supported to be used in loopback (send in a time slot and receive in another one) by the GNU Radio platform. Figure 2 shows an example of an error when using the HackRF in loopback.
- **Finally:** Since the transmission frequencies are adjusted by the manufacturer, when we deal with heterogeneous SDRs (e.g. USRP, HackRF), there's always some offsets which are introduced and subsequently affect the obtained

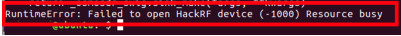


Fig. 2: Using HackRF as transceiver and receiver.

gain values. Therefore, it is challenging to set the optimal gains and frequencies that lead to a successful transmission and reception. Since each SDR uses a different driver with different signal processing operations and with different clocks and delays.

Work	Software Platform	Hardware	Antennas in each SDR	Loop-back problem
[14]	GNU Radio	USRP	Multiple	No
[5]	GNU Radio	USRP	Multiple	No
[6]	GNU Radio	USRP	Multiple	No
[7]	GNU Radio	USRP	One	No
[8]	GNU Radio	USRP	One	No
[9]	GNU Radio	USRP	Multiple	No
[10]	GNU Radio	BladeRF	Multiple	No
[11]	GNU Radio	BladeRF	Multiple	No

TABLE II: Prices of the different SDRs

As shown in Table II, all the existing TWRC PNC implementations use the same type and version of the SDR for the different nodes of the TWRC system to avoid the issues linked to the heterogeneous SDR devices. Furthermore, all used SDRs support multiple antennas to prevent the loop-back error shown in Figure 2. In our solution, we used three types of SDRs: USRP b210, HackRF One, and the Adalm-Pluto. Besides the problem of hardware heterogeneity, the HackRF is limited to one antenna and does not support the loop-back under the GNU Radio platform. Therefore, to implement the TWRC PNC system using those three types of SDRs together, we have to deal with, primary, the impact of the hardware variety on the frequency setting and, secondly, with the problem of loop-back of the mono-radio SDRs.

In the following section, we present the proposed PNC prototype. Next, we present the solutions proposed for the implementation to fix the mentioned challenges.

III. PROPOSED PNC PROTOTYPE

In this section, the purpose is the modeling of a real time PNC-TWRC system based on SDR in two time slots totally to demonstrate the exchange of two data streams between two end nodes via a relay node. The operation at the relay node is XOR on bits without symbols.

As shown in Figure 1, the PNC transmission occurs in two phases: The first phase is the Multiple Access Phase (MAP), where the two edge nodes N_1 and N_2 transmit their data (P_1 and P_2 , respectively) simultaneously to the relay in the first slot. The second phase is the Broadcast Phase (BCP) where the relay broadcasts the coded PNC data ($P_1 XOR P_2$) to the two edge nodes. Since the BCP is traditional broadcast transmission and we use the half-duplex in our implementation, the second BCP requires a MAC layer to schedule and coordinate the MAP, and the BCP transmissions [15], which is not in the scope of our work. Therefore, in our implementation, we focus on the first MAP, which is the most challenging phase of the PNC system and the most complex signal processing and where PNC mapping operations occur.

For a reliable transmission, uplink signals must be distinguished at relay node, time and frequency synchronization should be respected, and channel estimation must be precisely provided. For that purpose, we used frequency multiplexing, the preamble part of OFDM frame (the structure proposed by the Schmidl-Cox) and beacon signals.

As illustrated in Figure 3, we used the OFDM technique for transmissions to avoid the perfect synchronization challenge required by the PNC system. In this work, we adopt the OFDM transmitter and receiver proposed in [16] where each node N_1 and N_2 sends its data in different OFDM frequency. The node N_1 sends its data to the relay using a frequency slightly shifted to the right ($+\beta$) from the central frequency f used by the relay node R to receive the superimposed signals of the PNC transmissions. The node N_2 also sends its data to the relay node R using a shifted frequency, but reverse to the node N_1 . It uses a frequency slightly shifted to the left ($-\beta$) from the central frequency f .

Using one antenna and one central frequency, as shown in Figure 4, the relay recovers the signal and the data P_1 of the node N_1 by shifting the received signal on the central signal f to the right by a value of $+\beta$. The relay also applies the same process to recover the signal and the data P_2 of the node N_2 by shifting the received signal on the central signal f to the left by a value of $-\beta$. After, recovering the data P_1 and P_2 of the two edge nodes N_1 and N_2 , respectively, the relay applies the eXclusive Or (XOR) on P_1 and P_2 and broadcasts the PNC resulting data $P_1 XOR P_2$ to the two edge nodes N_1 and N_2 in the second time slot or the BCP. The relaying technique used here is called the Decode And Forward (DAF) relaying technique. We refer to the work presented in [13] for more relaying techniques.

Figure 3 presents the prototype of the edge nodes N_1 and N_2 , and figure 4 illustrates the prototype of the relay node R .

IV. IMPLEMENTATION OF PNC UNDER TWRC MODEL USING HETEROGENEOUS SDRs

In order to implement the PNC technique under the TWRC channel using heterogeneous SDRs (USRP b210, HackRF One and Adalm-Pluto), as we mentioned before we have to deal with two major challenges:

- 1) **The loop-back on each SDR:** the SDR should be able to play a double role, transmitter and receiver in half-duplex, but given our use case, the Hackrf driver does not support this option.
- 2) **The heterogeneity of SDRs:** the heterogeneity of the SDRs or drivers used (three types of SDRs) led to multiple configurations and, therefore, different outputs (frequency, transmission power, sample rate, bandwidth, etc.).

Thus, in order to make our performance results more reliable, we are deploying techniques that should disregard this heterogeneity and homogenize the hardware and software context.

First, we have integrated two intermediate cross-layers between the SDRs and the GNU Radio platform to deal with

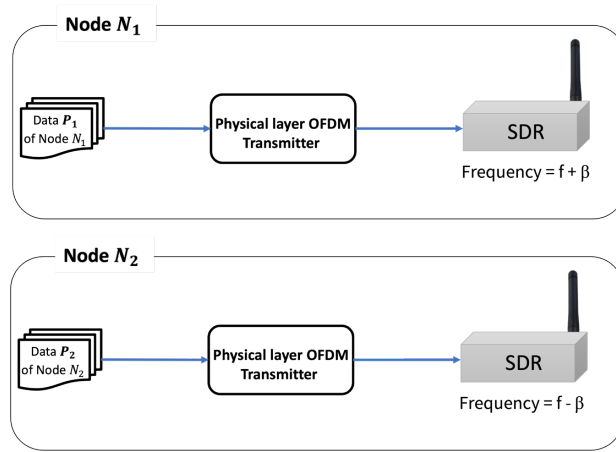


Fig. 3: The prototype of the two edge nodes N_1 and N_2 .

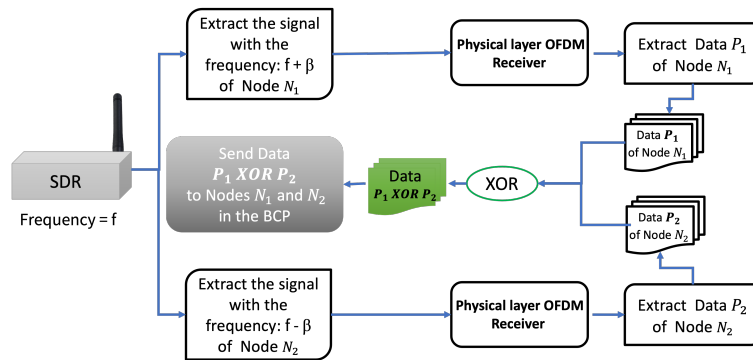


Fig. 4: The prototype of the Relay node R .

these challenges, as shown in Figure 5. The integrated layers are the **SoapySDR** [17] and the **SoapyUHD** [18] layer. The reason for using each layer is as follows:

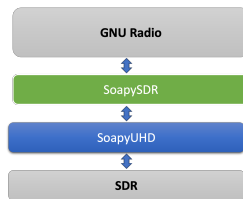


Fig. 5: Integrated layers between the SDRs and GNU Radio.

The SoapySDR: SoapySDR allows us to make abstraction on the SDR used and make the implementation functional on a single SDR system, which is the SOAPY system, rather than using different SDR systems, which lead to different outputs.

The SoapyUHD: SoapyUHD allows us to run and use the different types of SDRs as a single type, which is the USRP Hardware Driver (UHD) [19]. The GNU Radio platform will consider all the SDRs as USRPs. Thanks to the Soapy layers, all the SDRs (USRP, HackRF and Adalm-Pluto) will be treated and detected as UHD devices. This leads to generating uniform configurations for the different types of SDRs used, which

solve the heterogeneity problem of the SDRs. Furthermore, this UHD offers a virtual radio (UHD) capable of playing dual roles (transmit or receive) for the SDRs that do not support the loop-back function (SDRs using one antenna like HackRF).

In our implementation, we used UBUNTU 16.04 as an operating system and GNU Radio 3.7.11 as a platform (software). GNU Radio is Free Software for implementing software radios and signal processing systems. It offers a graphical interface (GNU Radio Companion) and a set of tools that provides a graphical assembly of all the blocks and a reliable run of the desired simulations. Hence, the GNU Radio flow graph is a block sequence. Each block represents a signal processing operation or SDR job. For the hardware, we used the HackRF for the edge node N_1 , Adalm-Pluto for the edge node N_2 and the USRP b210 for the relay node R as shown in Figure 6.



Fig. 6: The used Testbed.

Figure 7 shows the flow graph of the edge nodes N_1 and N_2 and Figure 8 shows the flow graph of the relay node R .

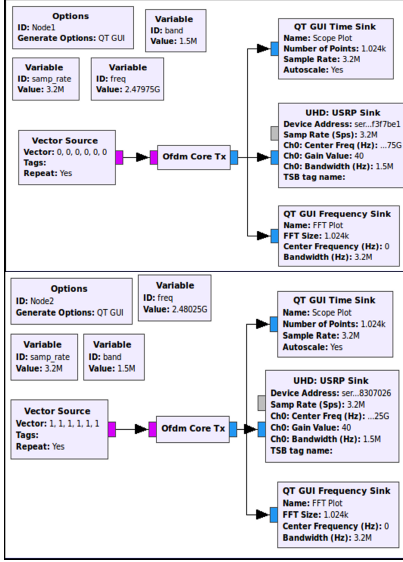


Fig. 7: The GNU Radio blocks of Node N_1 and N_2 .

As we can see, the flow graphs of the GNU Radio match perfectly the prototypes presented before. Ofdm Core TX is the OFDM transmitter and the UHD:USRP Sink is the block connected with the edge nodes N_1 or N_2 . The rest of the blocks are just blocks for visualization.

For the Relay's flow graph shown by Figure 8, UHD:USRP Sink is the block connected to the relay node R , the Multiply blocks are used to shift and extract the signals of N_1 and N_2 , Ofdm Core TX is the OFDM receiver and the XOR block is the PNC mapper that codes P_1 and P_2 to $P_1 XOR P_2$.

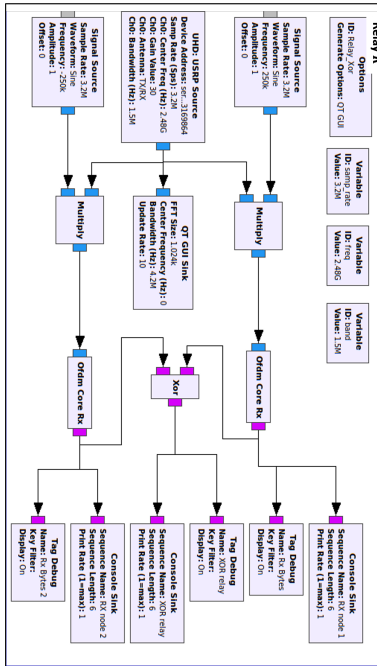


Fig. 8: The GNU Radio blocks of the relay R .

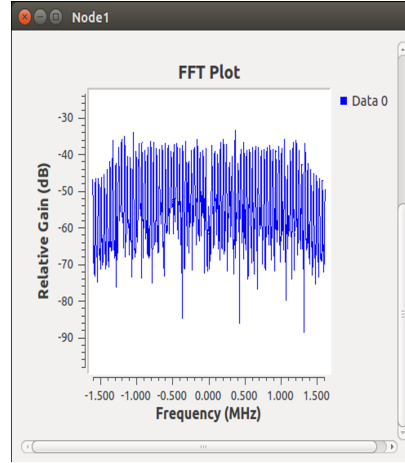
Table III summarizes the parameters used for the SDRs.

Parameter	Value
Frequency f	$2.48 GHz$
Shift value β	$250 kHz$
Sample Rate	$3.2 M$
Bandwidth	$1.5 M$
Gain	40 for TX, 30 for RX
Distance between two SDRs	50 CM

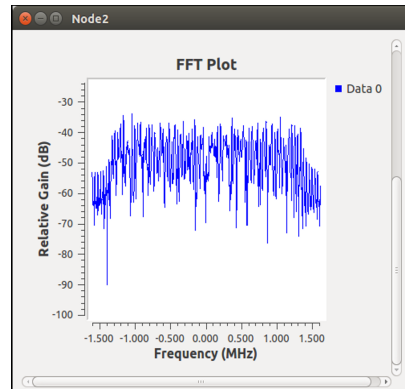
TABLE III: The parameters used for the implementation

V. PRELIMINARY RESULTS

This section presents the preliminary results of implemented TWRC PNC system using heterogeneous hardware SDR devices. Figures 9 (a) and 9 (b) represent the sent signals by the edge nodes N_1 and N_2 , respectively. From the resulting signals, we can see that the outputs (frequency, bandwidth, Gain ...) nearly perfectly matched the inputs, although the devices are heterogeneous. For example, from Figure 9 (a), the gain is approximately 40, which is the input gain, the bandwidth is $1.5 MHz$ (from 0.000 to 1.500 MHz or from 0.000 to -1.500) and the frequency is perfectly centralized to $0.000 MHz$, which represents $f + \delta$ for the node N_1 and $f - \delta$ for node N_2 .



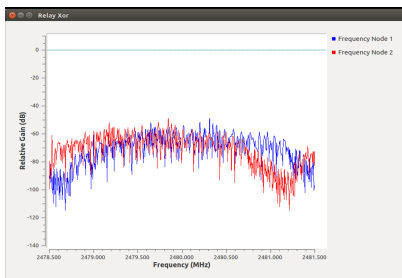
(a) Sent Signal by Node 1.



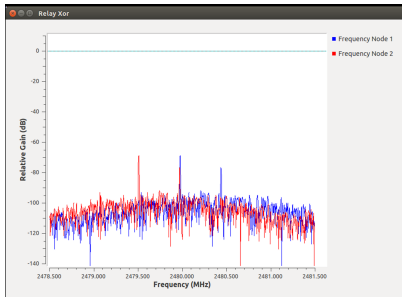
(b) Sent Signal by Node 2.

Fig. 9: Preliminary Results at Edge Nodes.

For the relay node R , Figures 10 (a) and 10 (b) represent, respectively, the received signals by the relay node in the idle state, where the relay does not receive data from N_1 and N_2 , and the receiving state, where the relay receives data from N_1 and N_2 simultaneously. Figure 10 (a) shows that the received signals are perfectly centralized to $f = 2480.000\text{MHz}$ (the peak in the middle), either for the signal of node N_1 (the blue one), or the node N_2 (the red one), this figure also shows that the two signals of nodes N_1 and N_2 are perfectly centralized and detected (the peak in the right and in the left for nodes N_1 and N_2 , respectively). For the receiving mode (MAP), Figure 10 (b) shows that the relay node detects the two signals of nodes N_1 and N_2 simultaneously and perfectly. We can also easily note the signal of node N_1 shifted to the right, and the signal of node N_2 shifted to the left, respecting the frequency inputs specified in the flow graph or the GNU Radio diagram block.



(a) Received Signal by the Relay in Idle Mode.



(b) Received Signal by the Relay in Receiver Mode.

Fig. 10: Preliminary Results at the Relay Node.

VI. CONCLUSION

This paper proposes the first PNC implementation under the TWRC network, using heterogeneous SDRs. We have presented the challenges of the heterogeneous SDRs and the solutions to overcome them. The implementation results prove the proposed solutions' power and the possibility of implementing the PNC using heterogeneous SDRs.

In future works, we will implement a well-designed MAC layer for the PNC technique above the presented PNC physical layer to adapt the PNC technique in larger multi-hop networks.

REFERENCES

- [1] S. Zhang, S. C. Liew, and P. P. Lam, "Hot topic: Physical-layer network coding," in *Proceedings of the 12th annual international conference on Mobile computing and networking*, 2006, pp. 358–365.
- [2] C. Fragouli, J.-Y. Le Boudec, and J. Widmer, "Network coding: an instant primer," *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 1, pp. 63–68, 2006.
- [3] E. Inc, "Universal software radio peripheral," <https://www.ettus.com>.
- [4] G. FSF, "Gnu radio - gnu fsf project," <https://www.gnuradio.org>.
- [5] L. Lu, L. You, Q. Yang, T. Wang, M. Zhang, S. Zhang, and S. C. Liew, "Real-time implementation of physical-layer network coding," in *Proceedings of the second workshop on Software radio implementation forum*, 2013, pp. 71–76.
- [6] L. You, S. C. Liew, and L. Lu, "Reliable physical-layer network coding supporting real applications," *IEEE Transactions on Mobile Computing*, vol. 16, no. 8, pp. 2334–2350, 2016.
- [7] K. H. Ngo and Q. T. Nguyen, "Implementation of network coding scheme in universal software radio peripheral," in *ICDV: The 5th International Conference on Integrated Circuits, Design, and Verification*, 2014.
- [8] N. K. Hoang, "Sdr implementation of ofdm-based physical layer network coding," Ph.D. dissertation, VIETNAM NATIONAL UNIVERSITY, HANOI UNIVERSITY OF ENGINEERING AND TECHNOLOGY, 2014.
- [9] D. Kramarev, A. Sakzad, and E. Viterbo, "Implementation of a two-way relay network with compute-and-forward in gnu radio," *Transactions on Emerging Telecommunications Technologies*, vol. 27, no. 4, pp. 484–493, 2016.
- [10] T. T. T. Quynh, T. V. Khoa, L. Van Nguyen, and N. Linh-Trung, "Network coding with multimedia transmission: A software-defined-radio based implementation," in *2019 3rd International Conference on Recent Advances in Signal Processing, Telecommunications & Computing (SigTelCom)*. IEEE, 2019, pp. 109–113.
- [11] T. T. T. Quynh, N. K. Hoang, N. Van Ly, N. L. Trung, N. Q. Tuan, E. Bastug, S. Azarian, V. N. Q. Bao, T. X. Nam, M. Debbah *et al.*, "Network coding with multimedia transmission and cognitive networking: An implementation based on software-defined radio," *REV Journal on Electronics and Communications*, vol. 10, no. 3-4, 2021.
- [12] Nuand, "bladerf," <https://www.nuand.com>.
- [13] M. Aissaoui, C. Houaidia, A. Van Den Bossche, T. Val, and L. A. Saidane, "A new taxonomy of physical-layer network coding techniques in two way relay channel model with single antenna," in *2021 10th IFIP International Conference on Performance Evaluation and Modeling in Wireless and Wired Networks (PEMWN)*. IEEE, 2021, pp. 1–6.
- [14] S. G. S. Katti and D. Katabi, "Embracing wireless interference: Analog network coding," *ACM SIGCOMM Computer Communication Review*, vol. 37, pp. 397–408, 2007.
- [15] M. Aissaoui, C. Houaidia, A. van Den Bossche, T. Val, and L. A. Saidane, "Dynamic slotted network coding protocol," in *17th International Conference on Software Technologies (ICSOFT 2022)*, 2022.
- [16] K. Bansal and V. Tripathi, "Ofdm transmission and reception of packets using gnu-radio and usrp-communications lab project," 2012.
- [17] Pothosware, "Soapysdr," <https://github.com/pothosware/SoapySDR>.
- [18] —, "Soapyuhd," <https://github.com/pothosware/SoapyUHD>.
- [19] EttusResearch, "Uhd," <https://github.com/EttusResearch/uhd/wiki>.