Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

ORIGINAL PAPER

A Dynamic Network Coding MAC Protocol for Power Line Communication

Roberto Massi de Oliveira[1] | Luiz Filipe Menezes Vieira[2] | Marcos Augusto Menezes Vieira[2] | Alex Borges Vieira[1]

[1]Computer Science Department, Universidade Federal de Juiz de Fora (UFJF), Juiz de Fora - MG, Brazil
[2]Computer Science Department, Universidade Federal de Minas Gerais (UFMG), Belo Horizonte - MG, Brazil

Correspondence
*Alex Borges Vieira, Computer Science Department, Universidade Federal de Juiz de Fora (UFJF), Juiz de Fora - MG, Brazil.
Email: alex.borges@ufjf.edu.br

Abstract

Recently, Power Line Communication (PLC) is receiving attention from academic and industry researchers worldwide. In particular, Power Line Communication systems have been largely investigated as a medium for transmitting control signals, diagnostic information, or data measured by sensors in Smart Grid. However, PLC systems face several challenges, such as multipath effects and impulsive noise, which may degrade data communication performance. To surpass such issues we propose CodePLC, a Dynamic Network Coding MAC protocol for Power Line Communication. CodePLC relies on a single, yet dynamically chosen, relay node. This node stores and forwards linear combinations of PLC frames, which are then combined on their final destinations. We have evaluated the performance of CodePLC through simulations in a PLC system based on a time division multiple access orthogonal frequency-division multiplexing (TDMA-OFDM) scheme. Simulation results indicate that, in broadcast transmissions, CodePLC enhances system performance. When compared to a PLC system that does not consider the use of CodePLC, based on a stop&wait MAC layer protocol, our new protocol presents an average of 115% of goodput increase. Moreover, CodePLC reduces the average network buffers occupancy by 112%. Finally, our protocol is capable of reducing the mean end-to-end latency by 400%.

KEYWORDS:
PLC, network code, xor code

1 | INTRODUCTION

Power Line Communication (PLC) still attracts the attention of industry and academy. The growth projections for this area until 2022 surpass 15% and worldwide investments will reach up to US$11.27 billion[1,2]. Nowadays, Power Line Communication technology is receiving attention for a variety of services and applications[3], mainly thanks to the advantage of using the existing electrical infrastructures, which reduces deployment costs[4,5]. Moreover, in-home networks and high-speed backbones based on PLC systems are emergent trends, as well as research and models regarding PLC in such environments[6,7].

Despite this effervescent scenario, it is well-known that many factors may degrade PLC communication performance. For instance, data communication performance over PLC systems may suffer a high data loss rate due to physical environment characteristics such as signal attenuation, impedance mismatch, and background noise generated by other electrical appliances[8,9,10].

Power lines at a residence are a hostile communication environment. Electrical appliances such as motors, light dimmers, and fluorescent lights inject noises that mask the low-level signals. Moreover, these electrical characteristics often vary, presenting changes in communication channel properties continuously[11]. Thus, we advocate that the use of cooperative protocol and network coding techniques is an interesting approach towards performance and reliability improvements on PLC systems.

Cooperative communication is a promising approach to enhance the performance of many types of networks, such as wireless, ad-hoc networks and also PLC systems[12,13,14]. However, we highlight that there is a lack of studies about cooperative protocols for PLC systems, in particular, at the link-layer level. In fact, works that enhance PLC systems usually focus on the physical layer[15,16,17,18,10,4,19] or propose the use of repeaters and amplifiers[20,21,22,16]. The use of physical layer solutions is highly dependent on a given system or technology, imposing the use of special hardware, which in turn, increases system overall costs.

In this sense, we propose *CodePLC*, a novel PLC Dynamic MAC protocol that takes advantage of network coding. Differently from previous works, most focused on wireless networks[23,24,25], CodePLC considers the flawed nature of the PLC scenario and the benefits of network coding for enhancing network performance without relying on specific devices. Indeed, CodePLC dynamically uses a single relay, which stores/forwards linear combinations of PLC data. In other words, the relay intermediates the communication and enhances the overall system throughput and error rate. Moreover, besides the notable text enhancement and system discussion, when compared to our previous work[26], the current protocol supports a multi-hop scenario, with a dynamically chosen relay.

Our CodePLC evaluation relies on simulations of a PLC system topology based on time division multiple access orthogonal frequency division multiplexing (TDMA-OFDM). We have evaluated CodePLC over a set of simple and clear scenarios, using a wide range of configurations (e.g., variable relay availability and distinct packet sizes). Again, differently from our previous work, we evaluate several distinct scenarios, varying network parameters, and relay availability. These scenarios allow readers to easily understand how to use it and the results it can provide. Moreover, the scenarios we consider can be used as a building block for more complex network topologies. In this case, we believe the results we present are a lower bound of the CodePLC benefits.

Our numerical results show that, in a broadcast transmission that uses a single relay node, the use of *network coding* can increase the average *goodput* by up to 115%, when compared to a PLC system that does not use the proposed protocol. Moreover, the average buffer occupancy of network devices can decrease by as much as 71.4%. Finally, in a common scenario, where we apply the CodePLC protocol, the observed latency is up to 84.7% lower.

In sum, our main contributions are as follows:

- The proposal of a simple, yet effective, network coding protocol, named CodePLC, which works at the link-layer level using network nodes as relays adaptively to improve PLC system performance. As the proposed protocol operates at the link-layer level, it can be jointly used with other network coding protocols or schemes from lower/upper layers of the network protocol stack, complementing these other protocols/schemes.

- The performance analysis and comparison between a PLC system using CodePLC and a common PLC system that does not consider a network coding technique in terms of goodput, buffer occupancy, and end-to-end latency. The main objective of these analyzes is to verify the viability of the proposed protocol. In this sense, we investigate scenarios and conditions in which the proposed protocol achieves relevant performance results.

Finally, the remainder of this paper is organized as follows. In Section 2, we present background about Network Coding. In Section 3, we analyze the state of the art regarding network coding proposals for PLC systems. In Section 4, we describe the scenario. In Section 5, we present our proposal, describing our CodePLC protocol. In Section 6, we show the numerical results. Finally, in Section 7, we detail our main conclusions.

2 | NETWORK CODING BACKGROUND

Real-time applications are becoming more popular on the Internet, and the amount of traffic-related to these applications grows every day. Most of the real-time applications, such as VoIP and video conferences, generate a large amount of traffic. Moreover, they are often sensitive to delay and packet losses. In this context, in which there is high traffic on network links, network coding may present its maximum benefit[27].

Yeung et al.[28] define Network Coding as a method used to optimize data flow in a network by transmitting combined traffic messages. This combination is composed of two or more messages that arrive at the receiver and later are decomposed, or “decoded”, into the original messages. This is a new departure from the traditional network routing and forwarding scheme (store-and-forward[27]). For instance, in a simple scenario in which there is a single relay between two nodes that transmit data to each other, network coding enables data transmission among them simultaneously, regardless of link occupation. In other words, the system transmits both messages in a single time interval[29].

In packet switching networks, the data flow is defined by messages that leave one source towards a destination. In the transmitting station, the transmitted message is divided into packets, each packet contains some data from the message. These packets do not necessarily travel through the same route, but they all should arrive at the same destination, where the end receiver is capable of regrouping them to compose the original message. The main problem with this method is that, when the data rate or error rate is high, it commonly creates bottlenecks, congesting underused routes, increasing latency, and degrading performance in general[28].

As an example, Figure 1 depicts a butterfly topology network. Node s is the source of data traffic. Nodes $d1$ and $d2$ are the destination nodes and nodes 1, 2, 3 and 4 are intermediate nodes responsible for routing the transmitted packets $m1$ and $m2$ from s. Each arc represents one link between two nodes. The data flow in a packet switching network starts at s and needs to go through a route between source and nodes $d1$ and $d2$. The use of network coding allows nodes to codify packets. Data that arrives from two or more origins can be combined into one packet.

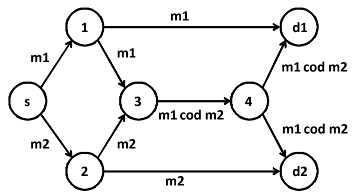


FIGURE 1 Network coding in a butterfly topology.

Two packets, $m1$ and $m2$, leave the same origin s towards $d1$ and $d2$ using different routes. These packets arrive in a node with the capacity to combine them into one unique packet (node 3) with the same size as the original packets. Note that this combined packet is not a concatenation of packets but a coding involving a combination of the packets’ information. This combined information arrives at $d1$ and $d2$ that, in possession of one of the messages, can decode the others. This method increases the throughput gains of a network by minimizing the number of congestion points[30].

For the application of network coding in a system, it is important to highlight two concepts: opportunistic listening and opportunistic coding. The former, already studied and used in wireless systems, is also present in PLC systems[31]. More specifically, opportunistic listening occurs when, after transmission between the source and destination nodes, their neighbors also receive the transmitted packet and store it for data communication improvement purposes [32].

Opportunistic coding refers to understanding when and in which packets to use network coding. This question has to be answered by the node locally, without needing to query the other nodes of the network[33].

Figure 2 exemplifies these two concepts. Let assume that the destination nodes, $d1$ and $d2$, need to receive both packets, $m1$ and $m2$, transmitted by the source nodes $s1$ and $s2$. These packets $m1$ and $m2$ depart, respectively, from the sources $s1$ and $s2$ toward destinations $d1$ and $d2$. However, the neighbour node 1 opportunistically listens to these transmissions and opportunistically codes these packets in a single one, $m1codm2$, which is transmitted to node 2. Then, node 2 broadcasts the coded packet to nodes $d1$ and $d2$. Lastly, nodes $d1$ and $d2$ use the previously received packets $m1$ and $m2$, respectively, to decode $m1codm2$ and recover the remaining packet.

There are several techniques of network coding that can be used to conduct the combination of packets. In this work, we chose the exclusive or (XOR) operator as the network coding technique. Such a technique offers better computational performance with low memory utilization and processing for both encoding and decoding[29]. Due to this advantage, we understand that performing network coding through XOR is a plausible option for PLC systems, especially, if we consider that XOR coding demands low hardware resources, which in turn, may not impact the final cost of PLC devices.

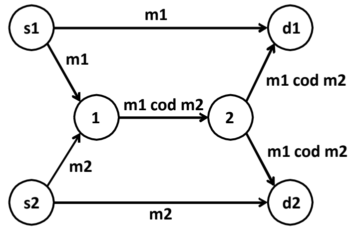


FIGURE 2 Example of opportunistic listening and coding for implementing network coding.

3 | RELATED WORK

It is well known that network coding enhances communication performance in the wireless broadcast scenario. In fact, Network Coding benefits wireless network resilience and improves transmission rates[34,23,35]. Existing works about Network Coding and wireless networks already evaluate maximum expected gains and boundary conditions[36,24]. Moreover, several protocols exist to help disseminate information in wireless scenarios, such as CodeDrip[25].

Despite the similarities between wireless and PLC systems, the gains in an environment cannot be automatically assumed into the other. Thus, there is still a lack of knowledge about the practical utilization of Network Coding in PLC systems[37]. For example, the circumstances in which Network Coding could improve PLC networks are unknown. Despite the positive results of the aforementioned work, there is no analysis of signaling (or control) overheads related to the use of Network Coding.

Only recently, a few works in the literature propose and discuss the use of Network Coding techniques in PLC systems[38,6,37,10,39,4]. Works that enhance PLC systems usually focus on the physical layer[15,16,17,18,10,4] or propose the use of repeaters and amplifiers[20,21,22,16]. While the use of physical layer solutions is highly dependent on a given system, the use of special hardware, as repeaters, increases system costs.

Previous works that address the effects of Network Coding in PLC systems, generally, evaluate its gains over ideal conditions. They do not propose MAC layer protocols nor do they evaluate with a realistic workload. For instance, Ezzine et al.[6] focus their work on proposing an algorithm for allocating subcarriers in a PLC system with a medium access layer based on OFDMA (Orthogonal Frequency Division Multiple Access) schemes. They evaluate network throughput gains using a simple Network Coding mechanism between two nodes and a relay. The authors also analyze the system varying the relative relay position between end nodes. Although the authors have used examples with Network Coding, they present an analysis of asymptotic gains without proposing a protocol to properly handle Network Coding neither consider realistic errors that can affect system performance.

Phulpin et al.[38,27] have also discussed the possible benefits of Network Coding on PLC systems. The authors propose the use of a linear code scheme in a smart grid environment, in which both PLC and wireless networks coexist. Their main goal is to improve efficiency on smart grids data collecting. However, the authors do not propose a communication protocol. Moreover, they do not evaluate realistic system metrics such as transmission rate, channel error rate, and buffer size limitations.

Bilbao et al.[40] propose a Network Coding PLC MAC protocol. They dedicate part of their work to characterize the error in a narrowband PLC system. They present results indicating that Network Coding can improve overall throughput up to (approximately) three times, compared with a traditional system using a stop&wait protocol. Despite the similarities, we highlight that our simulated scenario is different. In their work, Bilbao et al. do not design their protocol considering relay node availability. In summary, the authors show the upper limits of Network Coding gains on PLC systems and only address experiments in a controlled environment, without addressing error rates of realistic scenarios. Note that, in the current work, we evaluate important realistic system metrics such as buffer conditions and total latency for end-to-end data transmission.

Karaarslan et al.[10] and Chiti et al.[4] also propose the use of network coding on lower network layers of PLC. Indeed, Karaarslan et al. reduce the error bit rate by using adaptive modulation and coding on broadband PLC under multipath fading and impulsive noise. Authors employ QPSK, 16-QAM, and 64-QAM schemes and turbo coding at 1/2 and 3/4 coding rates. Despite authors carefully modeled channel characteristics, they assume a perfect/static environment. For example, they assume an always available relay. Chiti et al. also propose the use of Network Coding, but in this case, on a multicast-based PLC system. Their scheme performs multiple transmissions of the same symbol, in the frequency domain, to overcome impulsive noise issues. In other words, replicas of each symbol of the same packet are coherently combined at each receiving end to increase the data detection reliability. They also propose an optimized procedure for the data rate selection for the transmission of the network coding data flow. As we previously discussed, the use of Network Coding in lower layers (e.g. PHY.) turns the proposal highly dependent on a given system. Moreover, the use of a static scenario and a dedicated relay may incur high costs.

Tsokalo et al.[39] propose the use of Network Coding in a single-source-destination routing problem, in a mesh network with a broadcast channel. At a glance, authors assume an always available relay, which is dynamically chosen between special network nodes. The receiver node decides the relaying based on the route quality estimator and, additionally, on such information as the origin of the received data, the history of previous receptions, among others.

More recently, Borovina et al.[41] presents an error performance analysis and a model of a narrow-band power line carrier system for smart metering. This highlights that our protocol can also be used for smart metering. Ha et al.[42] investigates a version of TCP with Network Coding for Power Line Communication Networks, simulating in ns-3, that skips ACKs. They worked at the transport layer, modifying TCP, while we worked at the MAC layer.

Finally, besides the notable text enhancement and system discussion, when compared to our previous work[26], we highlight the following advances: first, the current protocol supports a multi-hop scenario, with a dynamically chosen relay. Second, we evaluate several distinct scenarios, varying network parameters, and relay availability. In the previous work, the protocol works considering fixed environments, with a single relay intermediating the source and the destiny by a single hop distance. In the current work, the protocol is much more flexible. The relay (or relays) of the network can be dynamically chosen. The network relay can be any network node, i.e., it can be a node that dynamically cooperates with the network, while it is idle. In this sense, the coding of messages may occur at any time, depending only on the availability of the relays. We evaluate the protocol using simple and illustrative network topologies. However, the current protocol supports multi-hop scenarios, which benefit a wide number of real networks.

In sum, we highlight the following contributions of the current work, when compared to the state-of-the-art: first, the Network Coding communication protocol for PLC systems we propose considers a dynamic scenario, where the relay may change and also be unavailable. Second, our protocol may coexist with actual lower layer proposals, which will enhance system performance even more once our proposal can mitigate problems that can only be detected after data packet reconstruction. Finally, we implement and evaluate CodePLC on realistic scenarios in which we evaluate the feasibility and benefits of Network Coding on PLC systems.

4 | SYSTEM MODEL

In this work, we consider a typical power line communication environment (also named as “Common PLC system” in the remainder sections), in which several nodes (residences) are interconnected by a base station (central node). Each node can communicate with a set of devices generating a wide range of custom network traffic. In this case, some of these nodes may not have direct contact with others, because of phase mismatch or high link attenuation[43]. Additionally, we consider an unencoded HS-OFDM (Hermitian symmetric-OFDM) scheme together with a binary phase-shift keying (BPSK) modulation; complete channel state information (CSI) at the receiver, and perfect synchronization. OFDM is a very efficient modulation technique, which is widely used in broadband communication, to handle the inter-symbol interference, and it also mitigates multipath problems[10]. Finally, we assume that transmission power is equally distributed among N sub-carriers of HS-OFDM symbols (P_0/N and P_1/N for nodes S and R, respectively) during a data communication cycle.

Figure 3 depicts an example of the considered scenario, where a source node (e.g., node A) broadcasts data to other nodes (e.g., nodes B, C, and E). Despite the simplicity of this scenario, it allows readers to easily understand how to use it and the results it can provide. Moreover, this scenario can be used as a building block for more complex network topology. In this case, we believe the results we further present are a lower bound of the CodePLC benefits.

Note that there is not a complete connection among all nodes in Figure 3. In this case, lines in this image indicate a shared link among power line communication devices, characterizing a perfect overhearing (e.g., node E can overhear nodes A, B, C, and D). Node E is the base station and can communicate with all remaining nodes. Finally, each node presents a data buffer, in which data packets are stored for later transmissions.

As previously mentioned, pair-wise communication may not be always possible due to lack of link or high error rate of a given direct link. In this case, nodes must rely on multi-hop packet forwarding, where a node or set of nodes serve as intermediate hops to the final destination. In the current work, we may assume a broadcast-like mechanism. However, any mechanism can be used as the ITU-T G.9903 and the IEEE 1901 standards, which define standards for the implementation of PLC systems for

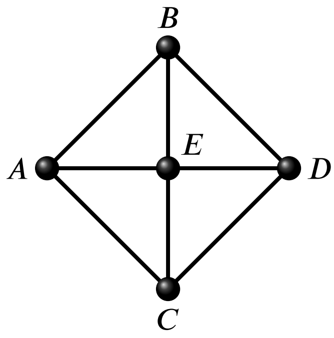


FIGURE 3 Evaluated Topology.

both broadband and narrowband[44,45], recommend multi-hop packet forwarding and both standards route data packets based on distance vectors.

In terms of the physical layer, we adopted a simplified model of the impulsive additive noise at the output of the PLC channel, which is adequate to illustrate the worst possible scenario[46,47]. The discrete-time domain representation of the adopted impulsive additive noise model is given by

$$v[n] = v_{background}[n] + v_{ps}[n] + v_{imp}[n] \tag{1}$$

in which $v_{background}[n] \sim \mathcal{N}(0, \sigma_v^2)$ represents the background noise, $v_{ps}[n] \sim \mathcal{N}(0, K_1\sigma_v^2)$ denotes the synchronous periodic impulsive noise, whose time of arrival of its impulses ($t_{arr,r} = 1/f_o$) is related to the power grid frequency ($f_o = 60$ Hz), $v_{imp}[n] \sim \mathcal{N}(0, K_2\sigma_v^2)$ refers to the asynchronous impulsive noise, whose inter-arrival time in the continuous-time domain between two consecutive impulses, $t_{arr,s}$, is an exponential random variable with a mean value of 100 ms. Furthermore, we assume that the duration of each synchronous periodic and aperiodic noise burst in the continuous-time domain is $t_{w,s} = 100\mu$s. Moreover, the channel impulse responses are those obtained in measurement campaigns performed in a typical urban area[48,49], considering a single relay model. We have obtained more than 36, 000 estimates of PER considering the frequency band from 1.705 MHz to 100 MHz. These frequency bands refer to the PLC system regulated frequency bands in European countries[50] and in Brazil[51], respectively. The 1.7-100 MHz band is a new alternative to increase the data rate in PLC systems[52,53]. In this work, we have considered measurements performed with a total power of 30 dBm and mean error ratio of 18% associated with each link. Finally, it is important to emphasize that we adopt a single relay model because the measurement campaign is related to this scenario. Although, the protocol we propose may use multiple relays.

The topology shown in Figure 3 is similar to the topology of Massi et al.[54], where each link $i \in \{AB, AC, BD, CD, AE, BE, CE, DE\}$ presents independent packet error rate (PER_i). Since the PLC environment and system model of[54] has been previously scrutinized, in the current work, we use the same methodology and model to estimate the system errors. The values of PER_i are estimated given the bit error rate (BER_i), which we can obtain from the physical layer regarding the i-th link. In this case, the PER is given by

$$PER_i \approx 1 - \left[(1 - \eta BER_i)^{\nu N_i} + (1 - BER_i)^{(1-\nu)N_i}\right], \tag{2}$$

in which $\eta \in \mathbb{R}_+^*$ is an augment ratio which varies according to the noise in a given time interval, $0 \leq \nu \leq 1$ and N_i corresponds to the packet size. Note that the η value describes the presence or not of impulsive noise by increasing or decreasing the bit error rate. Moreover, νN and $(1 - \nu)N$ refer to the subsequences of bits associated with the presence of the background noise plus the impulsive noise and background noise, respectively. We refer the readers to[54], in which the authors consider a similar model.

Table 1 shows how the presence of the impulsive noise affects the mean packet loss rate. Note that, for any η value the mean packet loss rate results are similar. This is since impulsive noises have inter-arrival times, as defined in Section 4, and, thus, they do not occur every frame. Moreover, regardless of the occurrence of bursts of erroneous bits, a data packet is corrupted, at the link-layer level, if at least one of its bits is erroneous.

Table 2 shows the PER variation according to the packet size we further use to evaluate the protocol we propose. Note that mean PER values increase as the packet size increases, which is expected according to the equation (2).

TABLE 1 Influence of impulsive noise on the packet loss rate.

Mean Packet Loss Rate		
$\eta = 1$	$\eta = 10$	$\eta = 100$
0.3451	0.3614	0.3673

TABLE 2 Mean PER values for each packet size ($\eta = 100$).

Packet Size (B)	Mean PER	Standard Deviation
80	0.0259	0.0082
120	0.0310	0.0093
160	0.0352	0.0102
200	0.0385	0.0107
240	0.0413	0.0109

The TDMA-OFDM based PLC system presents a subdivision in its frame: a control period followed by a data downlink/uplink period, regarding each node that uses the communication channel. As previously stated, we consider a perfect node synchronization at the link layer. The MAC sublayer of the nodes recognizes the channel timing division and will always be available to receive/transmit data according to this division.

During the corresponding time-slot, each node sends a packet (uplink), case it has any data in its transmission buffer. Clearly, during a specific time-slot, only one of the nodes transmits. In this case, the remaining nodes only receive the packet that the previously mentioned node sent and store it in their respective receiving buffer. We assume nodes present enough storage capacity, as buffering is a key factor in network coding.

In this work, we also assume an Automatic Repeat Request (ARQ) protocol based on stop-and-wait with ACK and NACK messages combined with a proper integrity check field through the use of a cyclic redundancy check (CRC) method. For the sake of simplicity, we consider CRC is effective and that channel error follows the error rate probabilities obtained by the measurement campaign. In other words, the source node stores the message it sends to the destination until it receives an acknowledgment. The source node tries to retransmit this message (for a limited number of attempts) until it receives an acknowledgment. The source node is not able to transmit other messages until the current message has been properly treated, either being confirmed or discharged. In this sense, the source node enqueues all new messages it wants to transmit. Queues may increase in this case until the network conditions allow to correctly transmit all messages in the transmission buffer.

5 | CODEPLC: A DYNAMIC NETWORK CODING PROTOCOL FOR A PLC SYSTEM

As we previously discussed, in Section 2, network coding can enhance data transfer in several network topologies and applications. In this sense, we propose the use of this technique at the MAC sublayer of a PLC system. In short, a PLC system can use any node as a relay which stores all MAC data units (*MAC Protocol Data Units* or simply, MPDUs) in buffers for later coding. Once it groups a given number of MPDUs, the relay codes these messages and forwards the final MPDU to all system nodes. When a node receives a coded MPDU, it decodes and obtains the desirable data.

Different network coding protocols use different operators to combine messages. In this work, we rely on the exclusive or (XOR) operator instead of a more complex finite field. This choice allows CodePLC to run efficiently on resource constraint nodes. On most CPUs, the XOR operator is just one hardware instruction.

Note that any system node may act as a relay. Although we may expect better results if we choose a central node, e.g. node E in Figure 3. Indeed, node E, in this figure, places a central position in network topology, being 1-hop distant from any other system node. Moreover, by previously choosing a suitable relay, we can properly sort control and data messages in order to favor the network coding operation. For example, in a time division (TDMA) based system, we may associate the last message to the relay. As a consequence, the relay will be able to efficiently combine all messages received in this frame from the other nodes in the previous time-slots.

Figure 4 details a simple network coding operation example, in a multi-hop PLC communication. In this figure, the blue, red, and green arrows refer to the first, the second, and the xor codec MPDU, respectively. We have also represented these MDUs as solid, dashed, and point-dashed lines (to make text black and white mode compliance). As we stated in Section 4, we consider a TDMA system. In this example, in the first frame, node A sends a message to all its neighbors (MDPU1 in msg1). Node C does the same in its time slots (MPDU2 in msg2). Then, as node B overhears the channel, it can opportunistically get both messages. Then, the two messages, MPDU1 and MPDU2, from the first frame[1] are encoded in one single message (*msg*3) by relay node B. The encoded message can be sent to both A and C nodes, in a single *time-slot* assigned to the relay node B. As a consequence, the transverse data exchanged occurred in a total of two frames, instead of three, as usual. In an environment with a larger number of transverse data flows, we may expect more opportunities to code and a better communication enhancement, reducing the number of system messages, increasing throughput, and also reducing the overall error rate.

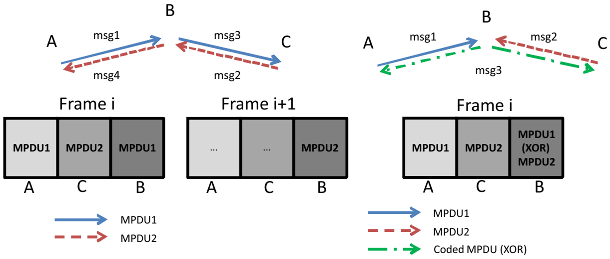


FIGURE 4 Transversal data transference using network coding in a TDMA based system.

To implement network coding in the power line communication context, we define transmission and reception procedures for both, peripheral and central (relay) nodes. Figure 5 presents a frame *i* and its slots assignment profile, for a system based on TDMA-OFDM. During each time-slot t_u, a node *u* sends its MPDU and each other system node connected (1-hop) to *u* receives this MPDU. The control time-slot is used by system nodes to send ACK or NACK messages, corresponding to the acknowledgment (or not) of previous MPDU transmission.

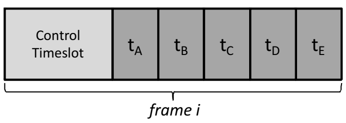


FIGURE 5 Allocation profile in a TDMA-OFDM frame.

Figure 6 presents a simple MPDU structure. The MPDU we show contains the MAC header and an extra Network Coding header. The MAC header contains the base structure of a PLC system, based on TDMA, which does not use any coding scheme. On the other hand, the Network Coding header contains additional fields to implement CodePLC.

The Network Coding header contains an extra bitmap field (Bitmap Coding). This bitmap indicates which data packets or MPDUs are XOR coded in the current frame. More precisely, the Bitmap Coding field presents an ordered array of bits. Each bit represents a system node. The bitmap order maps nodes in the same order of the frame time-slot. When the corresponding node *i* bit is set in the bitmap, CodePLC understands that the relay used node *i* MPDU as part of the xor coding process.

Network Coding Header presents redundant fields as the *"Components"* and *"Frame number"* fields. The *Components* field indicates the number of MPDUs that the current frame encodes. The number of bits set in the bitmap is equivalent to this field and indicates the number of encoded MPDUs. The *Frame number* is a field that can be used to generalize a protocol, and it can be inferred from the existing PLC header.

[1]In this context, a frame is the set of all messages transmitted during all time-slots of a TDMA round.

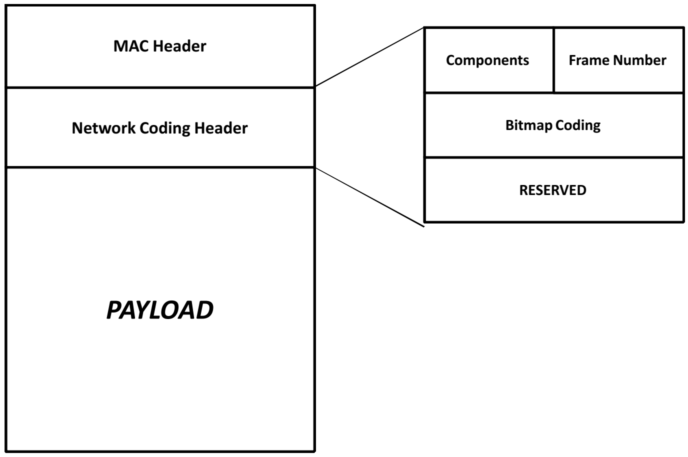


FIGURE 6 Structure of an MPDU in a MAC sublayer with CodePLC

As we previously stated, a frame is split between control and data phases, in this order. Thus, during a frame, nodes first check control messages, and then they may transmit/receive data. Figure 7 summarizes both transmission and reception procedures for a given node, during a frame i. In sum, nodes using our protocol works as follows:

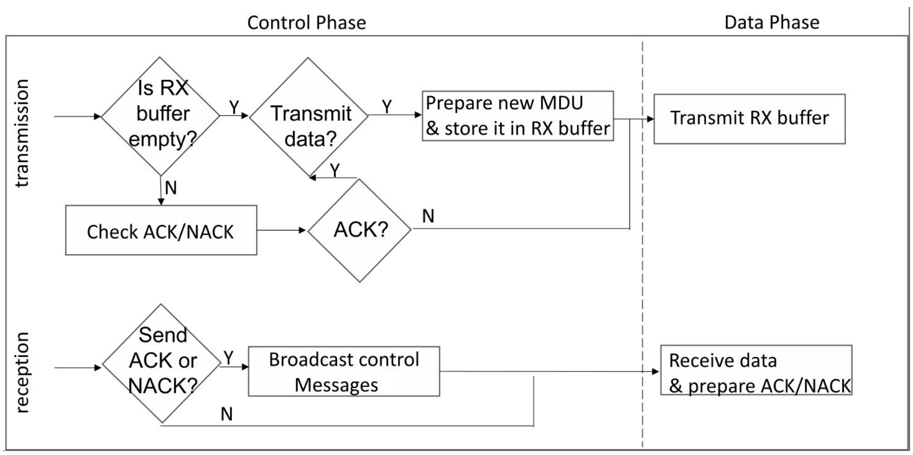


FIGURE 7 CodePLC transmission and reception procedure from a given node, during a frame i. Both, acknowledgment (ACK/NACK) and data transmission occur during a given node control and data time-slot, respectively.

First, during the control period from frame i, a node u checks ACK/NACK messages it receives from its neighbors. In the cases where u receives an ACK, the corresponding message of the previous frame $i-1$ has been correctly transmitted. Otherwise, in the case u receives a NACK, it needs to retransmit the corresponding message, during its data time-slot in the current frame i.

We assume that a given node u's transmission buffer is empty. Then, its upper layers generate data and request u's MAC layer to transmit it. During its time-slot, during the data phase in frame i, node u will try to transmit the data. Our protocol then creates an MPDU and stores it in the transmission queue. This MPDU will be transmitted and will remain in the transmission queue until u receives the corresponding ACK, in the following frames, from all addressed neighbors.

All nodes, including node u, may receive data (MPDUs). More precisely, a node j, during its time-slot t_j in frame i sends data to u. If u correctly receives the MPDU from a node j, it stores the corresponding data in a reception buffer and prepares an ACK message to j. Otherwise, u prepares j a NACK and discards the MPDU. Note that u will send an ACK/NACK, during the control time-slot in the next frame, for all nodes it has received an MPDU.

The last time-slot from frame i belongs to the relay. During this time-slot, all network nodes u at a 1-hop distance from the relay receive a broadcasted coded message. This message contains all corrected data received by the relay during this frame, coded into a single MPDU. The remaining network nodes confirm positively/negatively (ACK/NACK) this coded MPDU as they do to any other MPDU, during the next control time-slot, in frame $i + 1$.

Now, nodes may decode a given MPDU case it has correctly received all 1-hop distance messages (messages from its neighbors). Clearly, a node *u* stores MPDUs in its buffer and is able to perform a bitwise XOR using all MPDUs it received from its neighbors. To perform such an operation, *u* checks the CodePLC header bitmap to know which MPDUs are encoded. Case *u* misses any MPDU, it stores the encoded MPDU in its buffer and retries to decode by the end of the next frame. The flowcharts depicted in Figures 8 and 9 detail the steps of the CodePLC algorithm.

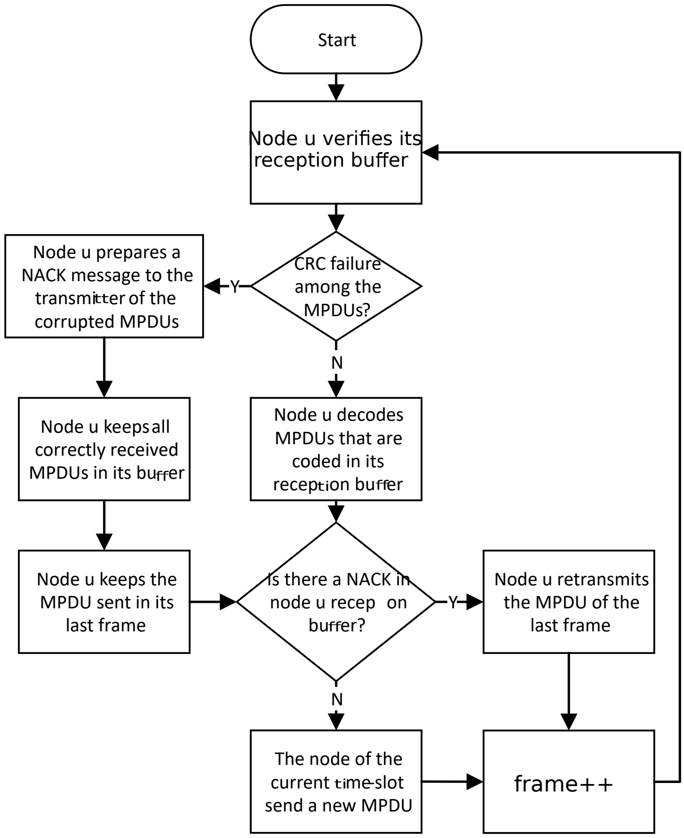


FIGURE 8 CodePLC transmission and reception of a common node

6 | SYSTEM EVALUATION METHODOLOGY AND RESULTS

In this section, we describe the simulation results and demonstrate the proposed protocol performance. First, in Section 6.1 , we explain the adopted simulation scenarios and evaluation methodology. Then, in Section 6.2, we present numerical results concerning the proposed protocol.

6.1 | Evaluation methodology

To analyze the performance improvements of our protocol, we developed a simulation tool using MATLAB, considering the topology depicted in Figure 3. Simulation, in this case, provides us with practical feedback when designing real-world systems. This allows us to determine the correctness and efficiency of a design before the system is actually constructed[55]. We focus on protocol simulation and we did not use any preexistent simulator. We follow realistic parameters of the physical layer (e.g., bit error rate) as discussed in Section 4. Both source code and packet error samples are available on http://netlab.ice.ufjf.br/plc.

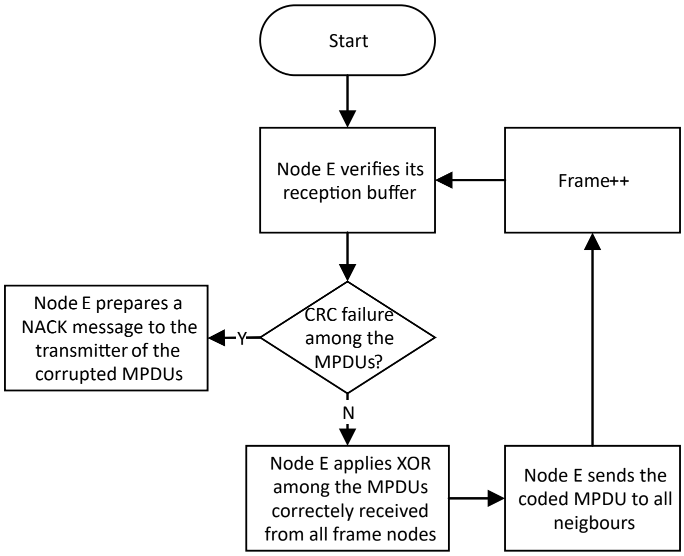


FIGURE 9 CodePLC transmission and reception of the coding node

In the following evaluations, we compare the current proposal, the CodePLC, against a “common PLC system”. A common PLC system does not use any coding technique. System nodes, in this case, only transmit data (MPDU) during their TDMA time slot. The so-called “common PLC system” serves as a base case. On the other hand, a PLC system using CodePLC (in short, CodePLC), is a PLC system as described in Section 5.

We evaluate CodePLC under two distinct scenarios:

1. A scenario with a dedicated relay: in this scenario, the central node, which is responsible for network coding, is always available to receive and to encode MPDUs from its neighbors, according to the mechanism we previously discussed (in Sec. 5). We also consider that standard system nodes transmit data, at each TDMA-OFDM frame, according to a given distribution. The evaluation of *BufferOccupancy, Goodput, and End-to-EndLatency* relies on this configuration.

2. A scenario with a dynamically chosen relay: In this scenario, CodePLC dynamically chooses a proper node to act as a relay. For the sake of simplicity, we do not employ any policy to choose a relay. In other words, we randomly choose an intermediate node between source and destination. Again, nodes transmit data, at each TDMA-OFDM frame, according to a given distribution. The evaluation of *Relay Availability* relies on this configuration.

In these aforementioned scenarios, we have varied the size of MPDU and the data transmission rate (t). These variations are interesting because they influence the packet error ratio results (i.e. MPDU size) and on data traffic in the simulated system (i.e. transmission rate). The MDPU size varies from 40, 80, 120, 160, 200 to 248 bytes, which are reasonable sizes for narrowband PLC applications, such as Smart Meters and Smart City, for example, as shown by Andreadou et al.[56] and Ikpehai et al.[57]. In particular, Andreadou et al.[56] consider the MDPU size of 100 bytes in the scope of smart meters. Ikpehai et al.[57] quote that the packet size for street lighting applications, based on PLC systems, varies between tens to a few hundreds of bytes. Moreover, these sizes are limited by physical blocks which are similar to the “IEEE Standard for Broadband over Power Line Networks: Medium Access Control and Physical Layer Specifications”[58].

In real scenarios, we may expect bursts of traffic. However, in this work, we prefer to keep the simulation as simple as possible to highlight protocol functionalities. We consider independent events and, in this case, the Bernoulli distribution is a good approximation. Nodes transmit data, with a probability of 50%, 70%, and 90% according to a Bernoulli distribution. The transmission rate is then normalized between 0 and 1, where 1 is the maximum channel capacity. We believe that these data transmission probabilities present a good compromise to demonstrate network behavior under low, average, and high network load. Furthermore, during impulsive noises, the BER is augmented by $\eta = 10^2$[54], affecting the Equation 2, which may lead to bursts of errors.

As we previously defined, in Section 5, we also consider an uncoded TDMA-OFDM system, which allocates all sub-carriers for a node when it is using its time-slot for transmission. We adopted digital BPSK modulation and the total transmission power of $P = 30$ dBm. Moreover, we considered a frequency band from 1.7 to 100 MHz. The 1.7–100 MHz band is a new alternative to increase the data rate in PLC systems[52]. Nevertheless, we have simulated network traffic crossing networks (i.e., A and B send packets to C and D, respectively, and vice-versa).

We consider that user time-slot within a TDMA-OFDM frame is sufficient to transmit the entire MPDU, without fragmentation. Moreover, we consider that MPDUs are padded if the data does not fill the payload field. Finally, we consider that all nodes present enough resources to process and store buffers.

In each scenario and its variations, we have executed the same simulations to determine the following metrics: mean occupancy of buffers, goodput, and mean latency. We pay attention to both transmitter and receiver buffers. The transmitter has to store MPDUs until it receives the correspondent acks. The receiver, on the other hand, has to store messages until it can group several MPDUs and properly decode data. Then, the mean occupancy of buffers is defined by the sum of all MPDUs in the buffers of all system nodes in a given time interval. Nodes discharge invalid MPDU and store only valid ones to further decode a message. The goodput is defined by the number of correct receptions by a node u in a given period. The mean latency is defined as the number of TDMA-OFDM frames a node demands to deliver an MPDU to the destination.

Unless we say otherwise, the results we present are mean values, or distributions, regarding $1,000$ repetitions of each simulation. We consider a system running in a stable regime. We discharge the initial and the final simulation run. For instance, to analyze buffer occupation and latency, each simulation remains for 200 TDMA-OFDM frames. In the current work, we sample the system every time a given probe node correctly transmits 100 MPDUs.

6.2 | Numerical Results and Performance Analysis

In what follows, we analyze buffer occupancy, goodput, and latency. We compare a common PLC system, without any cooperative technique and a system using CodePLC. In the following analyzes, we assume that packet generation and transmission occur at the same rate, so that $rho = t$.

Buffer Occupancy

Buffer occupation is a key metric that may enlighten protocol overhead and typical network problems, as network congestion. It is well known that XOR coding does not present a considerable overhead. Indeed, as shown in Section 5, the main CodePLC overhead refers to a bitmap, which occupies only a few bytes on the MAC layer header. On the other hand, if network traffic surpasses link capacity or the number of errors is high, nodes will enqueue frames to (re)transmissions, which will lead to a higher buffer occupation.

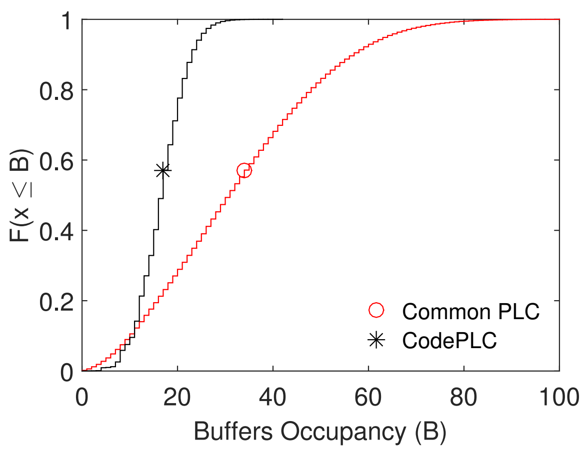


FIGURE 10 Cumulative Distribution Function of Nodes' Total Buffer Occupancy ($\rho = 0.7$, MPDU = 248 bytes).

Figure 10 shows the cumulative distribution functions (CDFs) of buffer occupancy, observed at each TDMA-OFDM frame in a common multi-hop scenario. In this scenario, the relay is always available and network nodes generate data at following a Bernoulli distribution 0.7 rate (probability of generating data at each frame is $\rho = 0.7$), which represents a good compromise between low and high network workload. Moreover, we execute simulations using MPDUs with a fixed size of 248 bytes.

This improvement occurs because when we adopt the proposed protocol, network nodes do not act as intermediaries in a 2-hop communication. Thus, we avoid bottleneck problems that would require sophisticated routing algorithms and extra signaling overheads. Moreover, as the proposed protocol uses an XOR encoding/decoding algorithm, the MPDU size increment is of just a few control fields. Finally, the decoding algorithm also contributes to the lower use of buffers. When a given node does not have all the required data to decode an encoded MPDU, it just discards all last frame MPDUs and just keeps in its buffer a single coded message. This node just asks its neighbors (1-hop distance nodes, but the relay) for retransmissions. More sophisticated ARQ schemes would incur larger buffers and complex management policies.

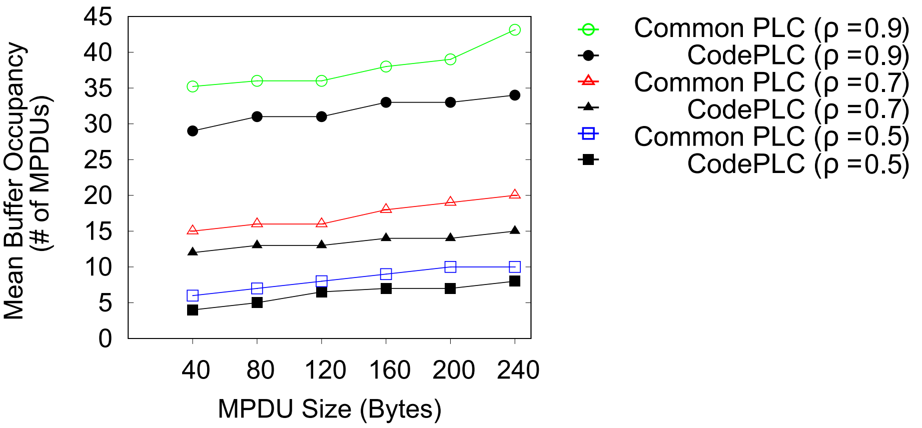


FIGURE 11 Mean buffer occupancy while varying MPDU size.

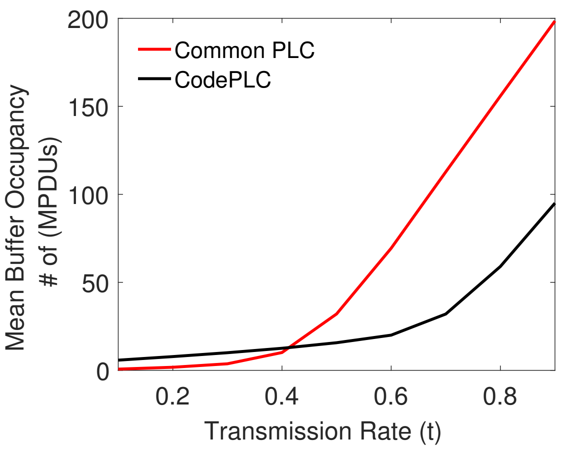


FIGURE 12 Mean buffer occupancy while varying the transmission rate (MPDU = 248 bytes).

Figures 11 and 12 present buffer occupation while we vary the MPDU size and the transmission rate, respectively. For both figures, the confidence interval, for a 99% confidence level, is practically negligible ($< 0.1\%$). As expected, buffer occupation increases almost linearly with the MPDU size. Moreover, buffer occupation is related to the number of packets a node has to store until it can decode a message. In this sense, The higher the transmission rate, the more packets a node receives. As a consequence, it has to store more combinations of messages, from more other nodes, until it can decode properly the original data. Moreover, note that for any ρ value, CodePLC can reduce the mean buffer occupancy. Furthermore, as the ρ increases, the distance between CodePLC and Common PLC curves increases. This result shows that the CodePLC tends to achieve higher mean buffer occupancy reduction in saturated networks when compared to a Common PLC system that does not consider any network coding technique at the link-layer level.

Overall, increasing the MDPU size degrades buffer occupation for both systems. Indeed, increasing MPDU size leads to a higher packet error rate, which in turn generates a higher number of retransmissions. As consequences, both systems present larger buffer occupation as nodes may enqueue several packets for retransmission.

Higher nodes' transmission rate also incurs the worst network performance. Although, a system using CodePLC is less impacted than a common PLC system. In fact, underloaded systems (i.e. transmission rate ≤ 0.4) present similar performance. However, while we increase the transmission rate, we also increase the dispute over network resources. Nodes may have to enqueue packets for transmission and, as we previously discussed, in this case, coding messages may lead to better performance. For example, for a 0.6 data transmission rate, CodePLC presents more than 71% better performance, when compared to the traditional system.

Goodput

Goodput is also a key metric to evaluate the proposed protocol. In sum, as shown in Section 2, it is expected that coding reduces the amount of time to transfer concurrent data packets and, as a consequence, increases system goodput.

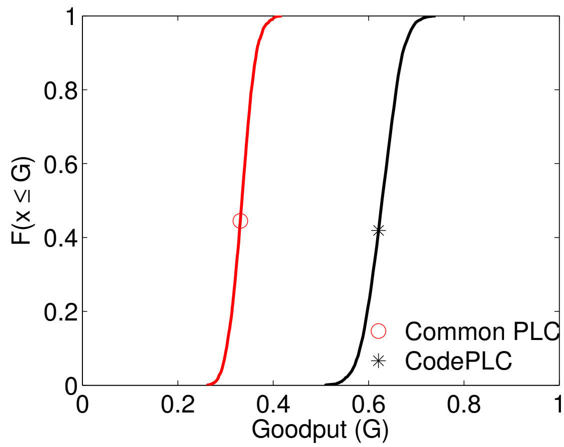


FIGURE 13 Cumulative Distribution Function of Goodput ($\rho = 0.7$, MPDU = 248 bytes).

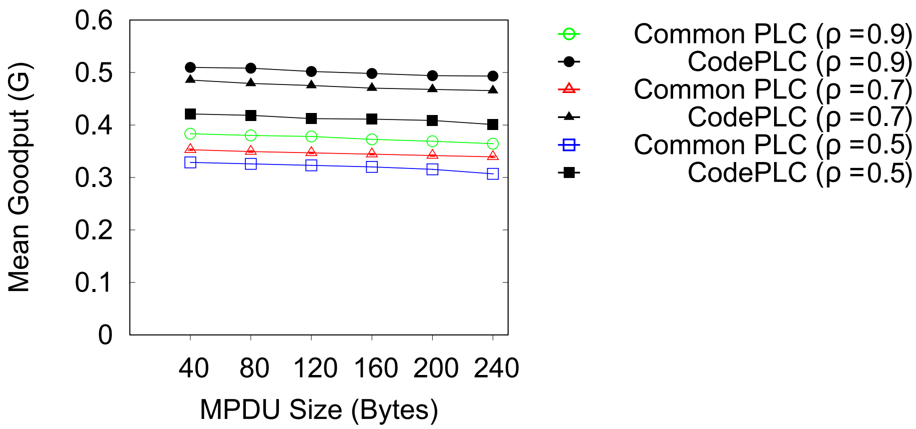


FIGURE 14 System mean goodput while varying MPDU size

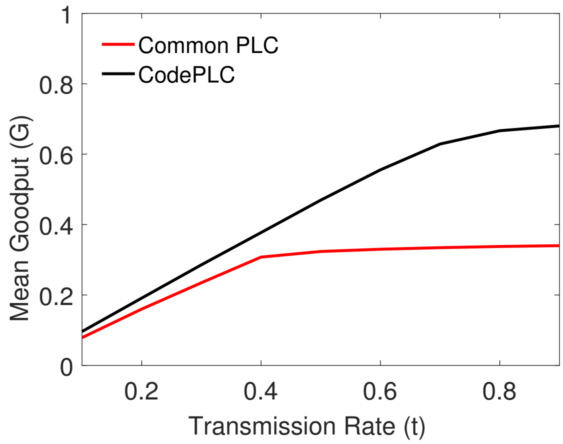


FIGURE 15 System mean goodput while varying system nodes transmission rate (MPDU = 248 bytes).

Figure 13 presents the goodput cumulative distribution functions (CDF) for all transmissions at each TDMA-OFDM frame, in a common multi-hop scenario. In this scenario, the relay is always available and again, network nodes generate data by following a Bernoulli distribution 0.7 rate and MPDUs present a fixed size of 248 bytes. As expected, goodput presents a notable difference between the traditional PLC system and a system using CodePLC. Indeed, coding allows concurrent transmission and, as a consequence, a better goodput. Moreover, as shown in Figure 10, buffer queues are smaller which also contributes to faster end-to-end delivery. In sum, the median difference between a system using CodePLC and a traditional PLC system is up to 100%.

Figure 14 presents the mean system goodput when varying the MPDU size. In this figure, the confidence interval, for a 99% confidence level, is practically negligible ($< 0.01\%$). For both systems, we note a slight decay in goodput when we increase MPDU size regardless of the ρ value. As we previously discussed, increasing the MPDU size incurs a higher packet error rate. As a consequence, when growing MPDU size, we may experience more retransmissions and a lower goodput. For all MPDU sizes, when $\rho = 0.7$, for instance, we observe a difference between both the systems up to 41%. In other words, a system using CodePLC remains roughly 41% better than a common PLC system, for all MPDU sizes we have simulated.

We have also evaluated goodput while varying the transmission rate. Figure 15 shows the mean goodput values and a negligible confidence interval, for a 99% confidence level. Note that, according to this figure, a common PLC system saturates faster in its goodput than a system using CodePLC. In fact, a common PLC system only presents negligible gains on goodput for transmission rates greater than 0.4. Differently, a system using CodePLC presents an increasing goodput up to a 0.7 transmission rate. Under low workload (transmission rate $\leq 25\%$), the difference between PLC systems is low, but not negligible. However, considering a high loaded system, where each node presents a transmission rate of 0.7%, the mean difference reaches more than 110%.

End-to-End Latency

Finally, we have evaluated end-to-end system latency, under the same conditions of previous simulations (relay always available, the transmission rate of 0.7 and MDPU of 248 bytes). Overall, end-to-end latency follows close to the previous results. For instance, as shown in Figure 16 –which presents the cumulative distribution function of end-to-end system latency – CodePLC presents a better performance when compared to a traditional PLC system. Clearly, CodePLC provides faster transmissions. Virtually, there are parallel transmissions while using network coding and, for this reason, mean transmission latency is expected to be shorter. More precisely, according to Figure 16, while the maximum latency experienced in a system using CodePLC is only 10 time units (TDMA-OFDM frames), in a traditional system, less than 18% of transmissions have achieved this threshold. In more than 50% of the cases, the latency nodes experienced was superior to 35 time units.

Figure 17 presents mean end-to-end latency while we vary nodes transmission rate. The confidence interval is negligible for a 99% confidence level. Again, for low network loads, both systems present similar behavior. However, CodePLC presents a considerably better performance for transmission rates higher than 0.4. For a high network load, the performance difference reaches more than 100%.

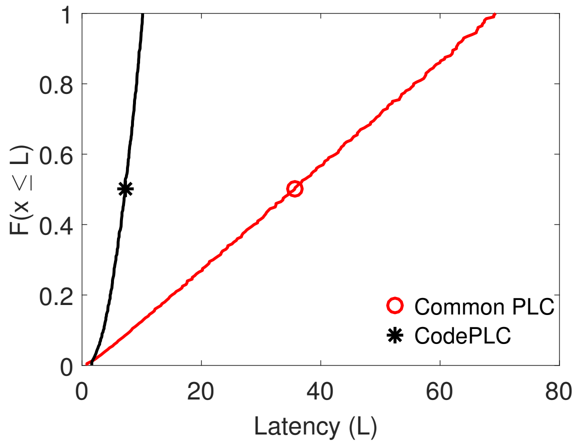


FIGURE 16 End-to-End Latency (# of TDMA *frames*) Cumulative Distribution Function (ρ = 0.7, MPDU = 248 bytes).

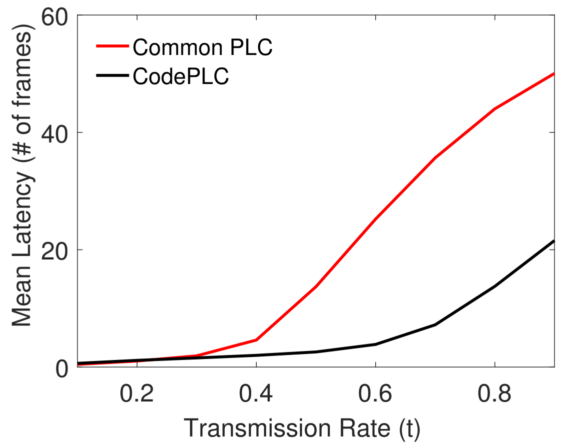


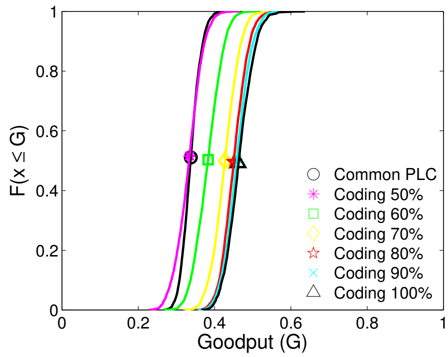
FIGURE 17 Mean Latency While Varying Transmission Rate (MPDU = 248 bytes).

Evaluating Relay Availability

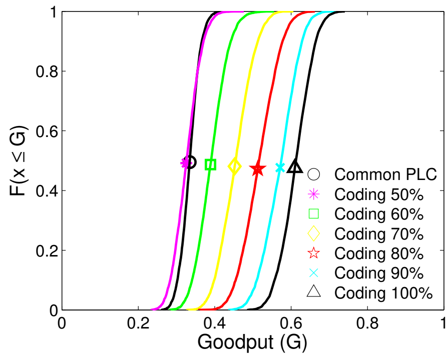
CodePLC relies on dynamically chosen relays to work. Many factors may impact relay availability. For example, a node may transmit its data and may not have enough resources to act as a relay. Moreover, the PLC channel may suffer electromagnetic interferences which may turn some nodes unavailable. In this sense, in the following, we evaluate the relay availability impact on CodePLC performance.

Figure 18 shows the system goodput while varying the relay availability, from 50% to 100%, distributed during the simulation. At a glance, a 100% availability means a node is always available to act as a relay, coding data it receives according to CodePLC protocol. On the other hand, a 0% availability means that the dynamically chosen node will not be able to code data, and the PLC system works traditionally. In this figure, we also varied the nodes' transmission rate, from 0.5, 0.7 to 0.9.

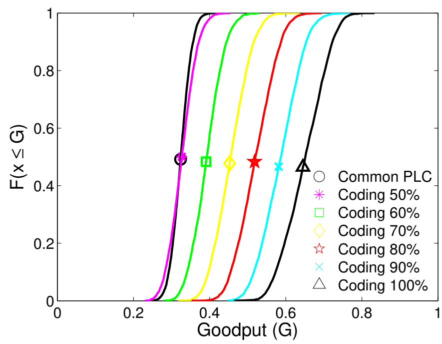
As expected, for all transmission rate scenarios, the more reliable a relay node is, the better the results our system provides. Moreover, the more overloaded the system, the more accentuated is the difference between high and low availability scenarios. More in-depth, for a 0.5 transmission rate scenario, a 50% availability about 0.3 median goodput, while a 100% relay availability provides about 0.45 median goodput. In a high overloaded scenario, where nodes present a 0.9 average transmission rate, CodePLC may provide almost a 0.7 median goodput. As we previously discussed, when nodes generate more data, CodePLC will have more opportunity to code data, and jointly to high relay availability, CodePLC enhances system performance.



(a) Transmission rate of 0.5.



(b) Transmission rate of 0.7.



(c) Transmission rate of 0.9.

FIGURE 18 Mean Goodput While Varying the Relay Availability.

Relay availability also interferes with system overall data transference latency. As shown in Figure 19, which presents the cumulative distribution function of transference latency for different relay availability (and a 0.5 transmission rate), the more the relay availability, the lower will be the latency. Indeed, while the median latency is up to 25 ms for a 50% available relay availability, the median latency is lower than 10 ms, when the relay is always available.

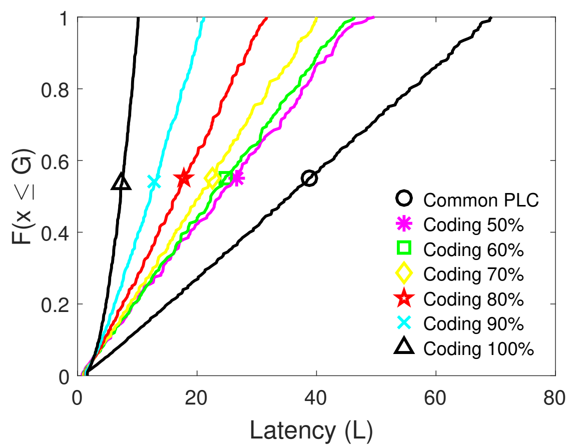


FIGURE 19 Cumulative Distribution Function for End-to-End Latency While Varying Relay Availability.

In sum, our results show CodePLC enhances overall system performance. In particular, both goodput and latency enhancements are more accentuated when relay availability and system nodes transmission rate increase. In some cases, the goodput turns 2.5 times better and latency falls in half.

7 | CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a MAC sublayer network coding communication protocol for PLC networks, named CodePLC. The proposed protocol uses one relay node to encode messages with the XOR operator. CodePLC allows a gain in goodput, latency, and robustness.

We verified and evaluated CodePLC with realistic simulations. The PLC system is simulated using time division multiple access (TDMA-OFDM). We used in our simulations measured values collected from real PLC scenarios.

Our simulation results show CodePLC expressive gains. Our protocol is capable of reducing packet losses and increasing goodput. In fact, when compared with a traditional system that considers the stop&wait protocol, CodePLC was able to achieve goodput gains of 116%. Additionally, the system buffer occupation reduces to half. It is also important to notice that the end-to-end latency is four times smaller with CodePLC.

Note that, as CodePLC is limited to a single network stack layer (i.e., the link-layer level), implementing it in an actual system may not demand big efforts. In fact, as we show in the example in Section 5, Figure 6, one has only to add very few more data to a standard PLC header. CodePLC can be jointly used with other network coding protocols or schemes from lower/upper layers of the network protocol stack, complementing these other protocols/schemes.

In most of the scenarios we evaluate, nodes of CodePLC do not need a considerable amount of specialized memory for the coding process. Their coding buffer can be small and CodePLC reduces retransmissions, which reduces the overall system buffer occupation. However, depending on the system error rate, relays may demand large buffers to deal with the coded messages. Moreover, in extreme situations, where all nodes are producing and transmitting data at a high rate, the dynamic relay election may not work. In this scenario, the system performs as a standard PCL system, without any coding.

Finally, note that there is a tradeoff between implementation complexity and the size of the system. The higher the number of nodes, the better the probable performance the system achieves once we encode more messages in a single message. However, manage distinct buffers and a high number of entities in a distributed system increases the complexity and the number of control messages.

For future work, we intend to integrate CodePLC in a hybrid environment for wireless/PLC systems. Furthermore, we plan to use network coding aggregation with opportunistic collaboration, reducing the need for a specific relay.

ACKNOWLEDGEMENTS

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001. Authors also acknowledge the financial support of CNPq and FAPEMIG.

References

1. Global Industry Analysts . MCP-7073: Power Line Communication (PLC) Systems: A Global Strategic Business Report http://www.strategyr.com/pressMCP-7073.asp, Date accessed: 18-07-2016; 2015.

2. Markets and Markets . Power Line Carrier Communication & PoE Controller Market worth 12.31 Billion USD by 2022 http://www.marketsandmarkets.com/Market-Reports/power-line-communication-plc-market-912.html, Date accessed: 18-07-2016; 2016.

3. Oliveira R. M., Vieira A. B., Latchman H. A., Ribeiro M. V.. Medium Access Control Protocols for Power Line Communication: A Survey. *IEEE Communications Surveys Tutorials.* 2019;21(1).

4. Chiti Francesco, Fantacci Romano, Tani Andrea. An efficient network coding scheme for reliable multicast power line communications. *Physical Communication.* 2017;25:34-42.

5. Dostert K., M. Zimmermann, T. Waldeck, M. Arzberger. Fundamental properties of the low voltage power distribution grid used as a data channel. *Transactions on Emerging Telecommunications Technologies.* 2000;11:297–306.

6. Ezzine S., Abdelkefi F., Cances J. P., Meghdadi V., Bouallegue A.. Joint network coding and OFDMA based MAC-layer in PLC networks. In: *Proc. IEEE International Symposium on Power Line Communications and its Applications*:311-315; 2014.

7. Muniz Adriano, Tsukamoto Kazuya, Kawahara Kenji, Tsuru Masato, Oie Yuji. Cooperative transmission scheme between PLC and WLAN to improve TCP performance. In: *Proc. IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*:142-147; 2013.

8. Gao Q, Yu JY, Chong PHJ, So PL, Gunawan E. Solutions for the “silent node” problem in an automatic meter reading system using power-line communications. *IEEE Transactions on Power Delivery.* 2008;23(1):150-156.

9. Zimmermann M., Dostert K.. A multipath model for the powerline channel. *IEEE Trans. on Communications.* 2002;50(4):553-559.

10. Karaarslan Güray, Ertuğ Özgür. Adaptive modulation and coding technique under multipath fading and impulsive noise in broadband power-line communication. In: *Proc. IEEE International Conference on Electrical and Electronics Engineering*:1430-1434; 2017.

11. Kim Charles J, Chouikha Mohamed F. Attenuation characteristics of high rate home-networking PLC signals. *IEEE Transactions on Power Delivery.* 2002;17(4):945-950.

12. Dubey A., Mallik R. K.. PLC System Performance With AF Relaying. *IEEE Trans. on Communications.* 2015;63(6):2337-2345.

13. Lee J., Kim Y.. Diversity Relaying for Parallel Use of Power-Line and Wireless Communication Networks. *IEEE Trans. on Power Delivery.* 2014;29(3):1301-1310.

14. Mohsen S. H., Ghosheh A. H., Mohammad M. K.. Capacity Analysis of PLC Point-to-Point and Relay Channels. *Transactions on Emerging Telecommunications Technologies.* 2016;47:200-215.

15. Biagi M.. MIMO Self-Interference Mitigation Effects on Power Line Relay Networks. *IEEE Communications Letters.* 2011;15(8):866-868.

16. Cheng X., Cao R., Yang L.. Relay-Aided Amplify-and-Forward Powerline Communications. *IEEE Trans. on Smart Grid.* 2013;4(1):265-272.

17. Kim Y., Choi S., Kim S., Lee J.. Capacity of OFDM Two-Hop Relaying Systems for Medium-Voltage Power-Line Access Networks. *IEEE Trans. on Power Delivery.* 2012;27(2):886-894.

18. Valencia J., Oliveira T. R., Ribeiro M. V.. Cooperative power line communication: Analysis of Brazilian In-Home channels. In: *Proc. IEEE International Symposium on Power Line Communications and its Applications*:301-305; 2014.

19. Jiang J Vincent Poor H.. Achievable rates for discrete memoryless relay channels with generalised feedback. *Transactions on Emerging Telecommunications Technologies.* 2013;24:212–231.

20. Noori M., Lampe L.. Improving data rate in relay-aided power line communications using network coding. In: *Proc. IEEE Global Communications Conference*:2975-2980; 2013.

21. Lampe L., Han Vink A. J.. On cooperative coding for narrow band PLC networks. *International Journal of Electronics and Communications.* 2011;65(8):681-687.

22. Kim K., Lee H., Kim Y., Lee J., Kim S.. Cooperative Multihop AF Relay Protocol for Medium-Voltage Power-Line-Access Network. *IEEE Trans. on Power Delivery.* 2012;27(1):195-204.

23. Vieira Luiz Filipe M, Misra Archan, Gerla Mario. Performance of network-coding in multi-rate wireless environments for multicast applications. In: *Proc. IEEE Military Communications Conference*:1-6; 2007.

24. Vieira Luiz Filipe M., Gerla Mario, Misra Archan. Fundamental limits on end-to-end throughput of network coding in multi-rate and multicast wireless networks. *Computer Networks.* 2013;57(17):3267-3275.

25. Júnior Nildo dos Santos Ribeiro, Vieira Marcos A. M., Vieira Luiz F. M., Gnawali Omprakash. CodeDrip: Data dissemination protocol with network coding for wireless sensor networks. In: *Wireless Sensor Networks*Springer International Publishing 2014 (pp. 34-49).

26. Silveira LMF, Oliveira Roberto M, Ribeiro Moises V, Vieira Luiz FM, Vieira Marcos AM, Vieira Alex B. CodePLC: A Network Coding MAC Protocol for Power Line Communication. In: *2016 IEEE 41st Conference on Local Computer Networks (LCN)*:643–646; 2016.

27. Prior Rui, Lucani Daniel E, Phulpin Yannick, Nistor Maricica, Barros João. Network coding protocols for smart grid communications. *IEEE Transactions on Smart Grid.* 2014;5(3):1523-1531.

28. Yeung R. W., Li S.-Y. R., Cai N., Zhang Z.. *Network Coding Theory*. USA: Foundation and Trends in Communications and Information Theory; 2 ed.2006.

29. Katti Sachin, Rahul Hariharan, Hu Wenjun, Katabi Dina, Médard Muriel, Crowcroft Jon. XORs in the air: practical wireless network coding. *ACM SIGCOMM Computer Communication Review.* 2006;36(4):243-254.

30. Ho Tracey, Lun Desmond S.. *Network Coding: An Introduction*. Reino Unido: Cambridge University Press; 1 ed.2008.

31. Tseng Shu-Ming, Lee Te-Lun, Ho Yo-Chain, Tseng Der-Feng. Distributed space-time block codes with embedded adaptive AAF/DAF elements and opportunistic listening for multihop power line communication networks. *International Journal of Communication Systems.* 2015;1.

32. Katabi Dina, Katti Sachin, Hu Wenjun, Rahul Hariharan, Medard Muriel. On Practical Network Coding for Wireless Environments. *IEEE International Zurich Seminar on Communications.* 2006;1:84-85.

33. Katti Sachin, Katabi Dina, Hu Wenjun, Rahul Hariharan, Médard Muriel. The importance of being opportunistic: Practical network coding for wireless environments. *Newsletter ACM SIGCOMM Computer Communication Review.* 2006;36(4).

34. Fragouli Christina, Le Boudec Jean-Yves, Widmer Jörg. Network coding: an instant primer. *ACM SIGCOMM Computer Communication Review.* 2006;36(1):63-68.

35. Vieira Luiz F. M., Vieira Marcos A. M.. Network Coding for 5G Network and D2D Communication. In: *Proc. Symposium on QoS and Security for Wireless and Mobile Networks*:113-120; 2017.

36. Keshavarz-Haddad Alireza, Riedi Rudolf H. Bounds on the benefit of network coding for wireless multicast and unicast. *IEEE Transactions on Mobile Computing.* 2014;13(1):102-115.

37. Parag Parimal, Chamberland Jean-Francois. Queueing analysis of a butterfly network for comparing network coding to classical routing. *IEEE Transactions on Information Theory.* 2010;56(4):1890-1908.

38. Phulpin Yannick, Barros João, Lucani D. Network coding in smart grids. In: *Proc. IEEE International Conference on Smart Grid Communications*:49-54; 2011.

39. Tsokalo I., Matthiesen B., Fitzek F. H. P.. Approaching the limits in routing in power line communication exploiting Network Coding. In: *2017 IEEE International Symposium on Power Line Communications and its Applications (ISPLC)*:1-6; 2017.

40. Bilbao J., Calvo A., Armendariz I., Crespo P. M., Medard M.. Reliable communications with Network Coding in narrowband powerline channel. In: *Proc. IEEE International Symposium on Power Line Communications and its Applications*:316-321; 2014.

41. Borovina Dzemo, Zajc Matej, Mujcic Aljo, Tonello Andrea, Suljanovic Nermin. Error performance analysis and modeling of narrow-band PLC technology enabling smart metering systems. *International Journal of Electrical Power and Energy Systems.* 2020;116:105536.

42. Ha N. V., Van Hau L., Tsuru M.. Dynamic ACK skipping in TCP with Network Coding for Power Line Communication Networks. In: *2020 22nd International Conference on Advanced Communication Technology (ICACT)*:29-34; 2020.

43. Ferreira H. C., Lampe L., Newbury J.. *Power Line Communications Theory and Applications for Narrowband and Broadband Communications over Power Lines*. United Kingdom: John Wiley Sons; 1st ed.2010.

44. Goldfisher Shmuel, Tanabe Shinji. IEEE 1901 access system: An overview of its uniqueness and motivation. *IEEE Communications Magazine.* 2010;48(10).

45. Recommendation I.. Narrowband orthogonal frequency division multiplexing power line communication transceivers for G3-PLC networks https://www.itu.int/rec/T-REC-G.99032017.

46. Picorone A. A. M., Amado L. R., Ribeiro M. V.. Linear and periodically time-varying PLC channels estimation in the presence of impulsive noise. In: *Proc. IEEE International Symposium on Power Line Communications and Its Applications*:255-260; 2010.

47. Hormis R., Berenguer I., Wang X.. A simple baseband transmission scheme for power line channels. *IEEE Journal on Selected Areas in Communications.* 2006;24(7):1351-1363.

48. Colen Guillherme R, Marques Cristiano AG, Oliveira Thiago R, Campos Fabricio PV, Ribeiro Moisés V. Measurement setup for characterizing low-voltage and outdoor electric distribution grids for PLC systems. In: *Proc. IEEE PES Conference On Innovative Smart Grid Technologies Latin America*:1-5; 2013.

49. Oliveira Roberto, Facina Michelle S. P., Ribeiro Moises, Vieira Alex Borges. Performance evaluation of in-home broadband PLC systems using a cooperative MAC protocol. *Computer Networks.* 2015;95:62-76.

50. Galli Stefano, Logvinov Oleg. Recent developments in the standardization of power line communications within the IEEE. *IEEE Communications Magazine.* 2008;46(7).

51. Brazilian National Telecomunications Agency . Technical Report, Resolučão Normativa 527 http://www.anatel.gov.br/legislacao/resolucoes/2009/101-resolucao-5272009.

52. Tonello A. M., Versolatto F., Pittolo A.. In-home power line communication channel: Statistical characterization. *IEEE Trans. on Communications.* 2014;62(6):2096-2106.

53. Tlich Mohamed, Zeddam Ahmed, Moulin Fabienne, Gauthier Frederic. Indoor power-line communications channel characterization up to 100 MHz—Part II: Time-frequency analysis. *IEEE transactions on power delivery.* 2008;23(3):1402-1409.

54. Oliveira R. M., Vieira A. B., Ribeiro M. V.. EPLC-CMAC: An Enhanced Cooperative MAC Protocol for Broadband PLC Systems. *Computer Networks.* 2019;153:11-22.

55. Craig Donald C. Extensible hierarchical object-oriented logic simulation with an adaptable graphical user interface. PhD thesisMemorial University of Newfoundland1996.

56. Andreadou N., Kotsakis E., Masera M.. Smart Meter Traffic in a Real LVDistribution Network. *Energies.* 2018;11(5):1-27.

57. Ikpehai A., Adebisi B., Kharel R.. Smart Street Lighting over Narrowband PLC In a Smart City: The Triangulum Case Study. In: *2016 IEEE 21st International Workshop on Computer Aided Modelling and Design of Communication Links and Networks (CAMAD)*:242-247; 2016.

58. *IEEE Standard for Broadband over Power Line Networks: Medium Access Control and Physical Layer Specifications.* 2010.