CodePLC: A Network Coding MAC Protocol for Power Line Communication

L. M. F. Silveira[†], Roberto M. Oliveira[‡], Moises V. Ribeiro[‡,*]
Luiz F. M. Vieira [◇], Marcos A. M. Vieira[◇], Alex B. Vieira[†]

[†]UFJF - Universidade Federal de Juiz de Fora, Computer Science Department, Brazil.
[‡]UFJF - Universidade Federal de Juiz de Fora, Electrical Engineering Department, Brazil.
[◇] UFMG - Universidade Federal de Minas Gerais, Computer Science Department, Brazil.

***Abstract*—Power line communication systems face several challenges which degrade data communication quality. To overcome such issues, we propose CodePLC, a network coding Power Line Communication MAC protocol. We use a single relay node to intermediate communication, storing, and forwarding linear combinations of data packets. We evaluate CodePLC performance through simulations of a common topology for a PLC system under a wide range of scenarios. In sum, our results show that in a broadcast like transmission, the use of network coding enhances overall system performance. When compared to a traditional PLC system, we have observed an average of 115% goodput increase. Moreover, our protocol reduces in 112% the average of network occupancy buffers. Finally, CodePLC reduces mean latency by four times.**

I. Introduction

Power line communication (PLC) attracts both, academic and industry attention. Indeed, we note an increasing number of in-home networks and high speed backbones based on PLC systems, as well as research and models about communication process on such environment.

Despite this effervescent scenario, data communication performance over PLC systems may present high data loss rate due to several physical environment characteristics. In fact, home power lines are an extremely hostile communication medium where electrical appliances such as motors and fluorescent light ballast inject noise and mask low-level signals.

The use of cooperation is an interesting and promising technique to improve communication performance in PLC systems [1]. However, there is a lack of studies about cooperation at the link layer of PLC systems. In fact, works addressing the enhancement of communication in PLC systems usually focus on physical layer [2], [3] or propose the use of repeaters and amplifiers [2], [4]. Nevertheless, PLC systems are a suitable environment for cooperative techniques and network coding (NC) in upper layers [5].

In this sense, considering the flaw nature of PLC scenario and notable benefits of network coding for improving networks performance [6], we propose CodePLC, a network coding Power Line Communication MAC protocol. We use a single relay node to intermediate communication. This node stores and forwards linear combinations of PLC data packets.

We evaluate CodePLC performance through simulations of a common topology for a PLC system with time division multiple access (TDMA-OFDM) under a wide range of scenarios (e.g. variable relay availability). Our results show that, in a broadcast like transmission, the use of network coding can improve mean goodput up to 115% when compared with a traditional scenario. Moreover, the mean buffer occupation of various network devices can decrease up to 115%. Finally, CodePLC reduces mean latency by four times.

II. CodePLC: Network Coding Protocol in a PLC System

It is well known that network coding can enhance data transfer in a wide range of topologies and applications. For instance, we may use such technique on MAC layer, in a PLC scenario, where we can use any node as a relay, storing all *MAC Protocol Data Units* (MPDUs) in buffers for latter coding. Once a group of MPDUs are coded, node relay forwards it to all system nodes connected to it. Once a node receives a coded MPDU, it performs the decoding operation and extracts the desirable information.

For the sake of simplicity, we choose a central node to act as relay. Moreover, we associate the last message –in a TDMA based system– to the relay. As a consequence, relay will be able to efficiently combine all messages received in this frame from the other nodes in the previous time-slots.

Figure 1 presents a simple example of XOR based network coding operation in a multihop communication. We consider a TDMA system and, in this example, MPDU1 and MPDU2 from the first frame are encoded in a single message ($msg3$) by relay B. This message can be sent to both, A and C nodes, in a single *time-slot*, assigned to B. As a consequence, in an environment with a large number of transversal data flows, most of all data exchange can be done in a single TDMA-OFDM frame, instead of two, as usual.

In order to implement network coding in our context, we define transmission and reception procedures for both, peripheral and central (relay) nodes. During each time slot t_u, a node u send its own MPDU and, each other system node connected (1-hop) to u receives this MPDU. During control time slot, system nodes send ACK or NACK messages, corresponding to the acknowledgment of previous MPDU transmission. For the sake of simplicity for this protocol proposal and during our simulations, we consider error free control messages.

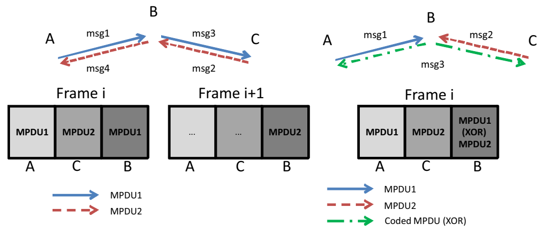

Fig. 1: Transversal data transference using network coding in a TDMA based system.

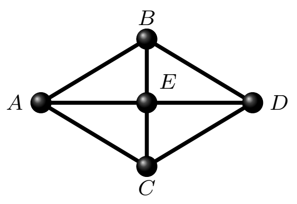

Fig. 2: Evaluation Scenario Setup.

Overall network code overhead is associated to a bitmap we add to existing MPDU header. This bitmap shows which data packets, or MPDUs are XOR coded. Note that, as a frame presents a low number of MPDUs (proportional to the system number of nodes), CodePLC overhead is O(1).

In sum, each node acts as in the following description:

- Previous frame MPDU analysis: in given frame i, during control period, a node u receives an ACK/NACK message from their neighbors. Case u receives an ACK, the message of frame $i-1$ was correctly transmitted. In the case it receives a NACK, the message of frame $i-1$ will be prepared to be retransmitted in the current frame, within the u time-slot.
- MPDU reception: in a given time slot t_u of a frame i, a given node receives MPDUs from its neighbors. In the case the MPDU has been properly received, it stores data in a reception buffer and an ACK is prepared to be sent to all neighbors during the control time-slot in the next frame. Otherwise, it sends a NACK and discards the MPDU.
- Encoded MPDU reception: in a given frame i, the last time-slot belongs to the relay. During this time slot, all network nodes u receive a coded message from the network relay. This message contains all corrected data received by node relay during this frame, coded in a single MPDU. Network nodes confirm this MPDU reception during the next control time-slot, in frame $i+1$.
- MPDU decoding: a given node u may decode a given MPDU in the case it correctly received all its neighbors messages, and the correspondent encoded MPDU. If those messages are stored on u buffer's then node u will be able to decode the messages addressed to it. Otherwise, it stores the coded MPDU in a buffer until all neighbors MPDU's from a previous frame are correctly retransmitted.
- Data XOR coding: during the last time slot from a given frame i, relay node E performs a bitwise XOR using all MPDUs it received from its neighbors. It transmits this coded MPDU for all its neighbors, in a broadcast like operation.
- MPDU transmission: when upper layer generates data and requests a node u to transmit it, CodePLC creates a MPDU and stores it in a transmission queue. This MPDU will be further transmitted and it will remain in a buffer until u receives an ACK from all its neighbors in the following frames.

III. PLC Evaluation Scenario

In this work, we consider a typical PLC environment in which a number of residences are interconnected by a base station. Each node can deal with a set of devices generating a wide range of network traffic. As occurs to wireless networks, some of these nodes may not have direct contact with others, because of phase mismatch or high link attenuation.

Additionally, we use an uncoded HS-OFDM scheme together with a binary phase shift keying (BPSK) modulation, a complete channel state information (CSI) at the receiver and, a perfect synchronization. We use a total transmission power $P = P_0 + P_1$, where P_0 and P_1 are transmission powers allocated to nodes S and R, in this order. These transmission powers are equally distributed among N sub-carriers of HS-OFDM symbols during a data communication cycle.

Figure 2 shows our scenario, in which the data source communicates with multiple destinations. According this figure, we observe five nodes (A, B, C, D and E).There is not a complete connection among these nodes and, thus, the lines in this figure indicate a link shared among PLC devices, characterizing a perfect overhearing. Node E is the base station and is able to communicate with all remaining nodes.

Each link $i \mid i \in \{AB, AC, BD, CD, AE, BE, CE, DE\}$; has an independent packet error ratio (PER_i). Despite the existence of only one relay node in this scenario, we can expand our proposal for using multiple relays, considering the protocol we propose in Section II. WLoG, in this work, we focus on the use of a simple scenario, in which network coding application is appropriate.

We obtained PER_i estimates at physical layer calculating the bit error ratio (BER) related to the ith link so that $PER_i = 1 - (1 - BER_i)^{N_i}$, where N_i is the packet size. We estimated error values through a measurement campaign performed in a typical urban area [7]. We have obtained more than $36,000$ estimates of PER considering the frequency band from $1,705$ MHz to 100 MHz. In this work, we considered measurements performed with total power of 30 dBm and mean error ratio of 18% associated to each link.

IV. System Evaluation Methodology and Results

A. *Evaluation methodology*

In order to analyze performance improvements by using CodePLC in PLC systems, we developed, in MATLAB, a simulation tool considering the topology depicted in fig. 2. In this sense, we have tested two scenarios:

- Multihop network with five equal nodes: In this scenario, link layer is responsible for transporting MPDU's utilizing a random intermediate node between source and destination. It makes feasible the communication between two nodes out of a 1-hop range. All nodes transmit its MPDU's in broadcast. We consider, for every node that there is a probability for a fixed size MPDU generation, which is transmitted in each TDMA-OFDM frame (a node transmits MPDUs it generates with a given probability as we detail in next paragraph).

- Multihop networks with four equal nodes and the node E, responsible for network coding: in this scenario, the central node is always available to receive MPDU's from its neighbors and then, it encodes these MPDU's, according to the algorithm we previously discussed (in Sec. II) and put them in a transmission queue. We also consider, for every node (except the central one), that there is a probability for a fixed size MPDU generation, which is transmitted in each TDMA-OFDM frame.

In these aforementioned scenarios, we have varied new MPDU generation probability (25%, 50% e 75%). Transmission probabilities present a good compromise to demonstrate network behavior in low, average and high workload. As we previously defined, in section II, we also consider an uncoded TDMA-OFDM system, which allocates all subcarriers for a node when it is using its own time-slot for transmission. We adopted digital BPSK modulation and a total transmission power of $P = 30$ dBm. Moreover, we considered a frequency band from $1,7$ to 100 MHz. Nevertheless, we have simulated network traffic crossing network (i.e., A and B send packets to C and D, respectively, and vice-versa).

In each scenario we have analyzed, and its variations, we have executed the same experiments, aiming to determine the following values: mean occupancy of buffers, goodput, mean latency. The mean occupancy of buffers is defined by the sum of all valid MPDU's in the buffers of all system users in a given time interval. The goodput is defined by the number of correct receptions by a node u in a given time period. The mean latency is defined as the number of TDMA-OFDM frames needed to deliver a MPDU from a node source to a node destination in a given time interval.

We have performed $1,000$ iterations for analyzing buffer occupancy, goodput and mean latency, in which each one corresponds to 100 TDMA-OFDM frames.

B. Numerical Results

Figure 3 shows cumulative distribution functions (CDFs) of total occupancy of buffers in each TDMA-OFDM frame of a common multihop scenario and of a network coding PLC scenario. Unless we say otherwise, we use as base simulation a scenario where nodes presents a 50% probability of transmission, which represents a good compromise between low and high network workload. Moreover, we execute simulations using MPDUs with a fixed size of 248 Bytes.

For up to 8% of cases, a common scenario presents lower buffer utilization then a system using CodePLC. Intuitively, when a system has low dispute over resources, direct communication (in this case, through 2-hops) may generate less occupancy of system buffers. Moreover, network coding presents a intrinsic buffer overhead. E.g., we may have to store a number of MPDUs just before the coding operation. For the remaining cases, a system using CodePLC presents better performance. In fact, at least 50% of cases in a system that not uses coding presents buffer occupation greater than the maximum value we have observed in a system using CodePLC. The reduction of mean occupancy of buffer by using CodePLC is highly considerable, since it reaches up tp 200% median difference, when compared to a common PLC system.

This improvement occurs due to the fact that, when we adopt CodePLC, network nodes do not act as intermediaries in a 2-hop communication anymore. Thus, we avoid bottleneck problems that would require sophisticate routing algorithms and extra signaling overheads. Also, the decoding algorithm contributes for the lower use of buffers. When a neighbor node does not have required data in order to decode an encoded MPDU, it discards MPDU's of last frame and keeps just the coded message. Then, it asks all neighbors but central node to retransmit last frame MPDU's.

Figure 4 presents overall PLC systems goodput CDF. As expected, we need less TDMA-OFDM frames to perform MPDUs delivery when we use the protocol we propose. As a consequence, system mean flow rate increases. Nevertheless, according to Figure 3 there is a buffer queue reduction and, thus, less MPDUs wait in buffer queues of intermediate users. The median difference between a system using CodePLC and a common PLC system is up to 40%.

We have also evaluated goodput, while varying MPDU sending rate. Figure 5 presents mean goodput values and a negligible confidence interval, for a 99% of confidence. According to this figure, a common PLC system saturates faster than a system using CodePLC. In fact, a common PLC system only presents marginal gains on goodput for transmission rates greater than 50%. On the other hand, CodePLC still presenting an increasing goodput up to a 75% transmission rate. Note that under low workload, the difference between PLC systems is quite low. In this case, relative systems difference is lower than 25%. However, considering a high loaded system, where each node presents transmission rate of 75%, the mean difference reaches more than 115%.

Figure 6 show buffer occupancy of both PLC systems, while we vary transmission rate. As observed for goodput, the growth of MPDU size decreases performance of booth systems. Under low network workload, the direct transmission without the use of CodePLC is slightly better. This result support our previously discussion, regarding to Figure 3. However, increasing transmission rate, network links becomes more disputed by system devices. In such scenario, a system using CodePLC suffers less impact and performance difference between these two systems is about 112%.

Figure 7 presents cumulative distribution function of system end-to-end latency, comparing a common PLC system to CodePLC. Clearly, CodePLC provides faster transmissions. In fact, virtually, there are parallel transmissions while using network coding, as we illustrated on previous section and, for this reason, mean transmission latency is expected to be shorter. While the maximum latency experienced in a system

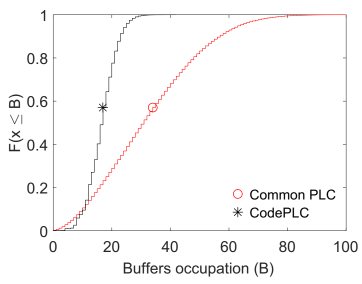

Fig. 3: Total buffer occupancy.

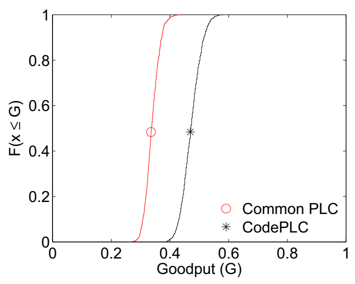

Fig. 4: Overall system goodput.

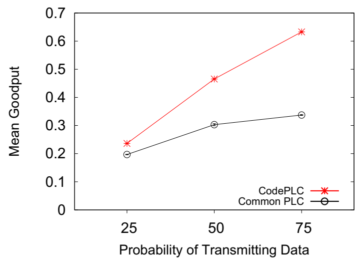

Fig. 5: System mean goodput.

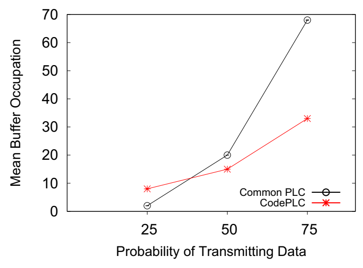

Fig. 6: Mean buffer occupancy.

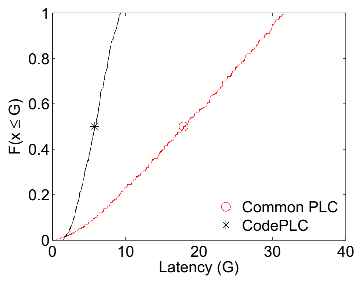

Fig. 7: Overall end-to-end latency.

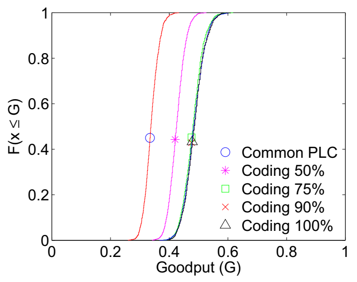

Fig. 8: System mean goodput.

using CodePLC is only 10 time unities (frame executions), in a traditional system, less than 20% of transmissions have achieved this threshold. In more than 50% of the cases, the latency nodes experienced was superior than 20 unities.

Finally, Figure 8 presents system goodput while we vary the probability a relay node is active. A 0% of coding, or a totally unreliable relay, is just the same as a traditional system and a 100% of coding corresponds to a fully reliable relay node. The more reliable a relay is, the better the results our system provides. For example, while a 50% reliable relay provides a 25.4% goodput on median, a 100% reliable relay provides 40%. Note that goodput presents only marginal gains for 75% and upper values for relay reliability. In other words, even in a PLC system where a relay is not totally available to perform network coding all the time, we would achieve goodput results as good as a system using a fully dedicated relay.

V. Conclusions and Future Works

In this paper, we proposed CodePLC, a communication protocol that uses Network Coding in the MAC layer for PLC networks. Our protocol uses one relay node to encode messages with the XOR operator enhancing system goodput, latency, and robustness. We evaluated CodePLC performance through simulations of a common topology for a PLC system under a wide range of scenarios. In sum, our simulations show expressive gains with CodePLC. It is capable of reducing packet losses and increase goodput. When compared with a traditional system such as stop&wait, CodePLC can achieve goodput gains of 116%. Moreover, the system buffer occupation reduces to half. Finally, the end-to-end latency with CodePLC is four times smaller.

Future works include extension for the protocols, especially in hybrid environments such as wireless/PLC systems.

Acknowledgment

Authors would like to acknowledge the support of research agencies CNPq, CAPES and FAPEMIG.

References

[1] A. Dubey and R. K. Mallik, "PLC system performance with AF relaying," *IEEE Trans. on Communications*, vol. 63, no. 6, pp. 2337–2345, 2015.

[2] X. Cheng, R. Cao, and L. Yang, "Relay-aided amplify-and-forward powerline communications," *IEEE Trans. on Smart Grid*, vol. 4, no. 1, pp. 265–272, Mar. 2013.

[3] J. Valencia, T. R. Oliveira, and M. V. Ribeiro, "Cooperative power line communication: Analysis of brazilian in-home channels," in *Proc. IEEE International Symposium on Power Line Communications and its Applications*, Apr. 2014, pp. 301–305.

[4] L. Lampe and A. J. Han Vink, "On cooperative coding for narrow band PLC networks," *International Journal of Electronics and Communications*, vol. 65, no. 8, pp. 681–687, Aug. 2011.

[5] J. Bilbao, A. Calvo, I. Armendariz, P. M. Crespo, and M. Medard, "Reliable communications with network coding in narrowband powerline channel," in *Proc. IEEE International Symposium on Power Line Communications and its Applications*, Apr. 2014, pp. 316–321.

[6] L. F. M. Vieira, M. Gerla, and A. Misra, "Fundamental limits on end-to-end throughput of network coding in multi-rate and multicast wireless networks," *Computer Networks*, vol. 57, no. 17, pp. 3267–3275, 2013.

[7] R. de Oliveira, M. S. P. Facina, M. Ribeiro, and A. B. Vieira, "Performance evaluation of in-home broadband PLC systems using a cooperative MAC protocol," *Elsevier Computer Networks*, vol. 95, pp. 62–76, 2015.