

Universidade Federal de Juiz de Fora
Instituto de Ciências Exatas
Programa de Pós-Graduação em Ciência da Computação

Luã Moloise Fernandes da Silveira

**CodePLC: um Protocolo de Network Coding MAC para Power Line
Communication**

Juiz de Fora

2016

Luã Moloise Fernandes da Silveira

**CodePLC: um Protocolo de Network Coding MAC para Power Line
Communication**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Juiz de Fora, como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

Orientador: Alex Borges Vieira

Coorientador: Moisés Vidal Ribeiro

Juiz de Fora

2016

Ficha catalográfica elaborada através do Modelo Latex do CDC da UFJF
com os dados fornecidos pelo(a) autor(a)

Silveira, Luã Moloise Fernandes da.

CodePLC: um Protocolo de Network Coding MAC para Power Line
Communication / Luã Moloise Fernandes da Silveira. – 2016.

53 f. : il.

Orientador: Alex Borges Vieira

Coorientador: Moisés Vidal Ribeiro

Dissertação (Mestrado) – Universidade Federal de Juiz de Fora, Instituto
de Ciências Exatas. Programa de Pós-Graduação em Ciência da Computa-
ção, 2016.

1. PLC. 2. Network Coding. 3. TDMA-OFDM. I. Borges, Alex Vieira,
orient. II. Ribeiro, Moisés Vidal, coorient. III. Título.

Luã Moloise Fernandes da Silveira

**CodePLC: um Protocolo de Network Coding MAC para Power Line
Communication**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Juiz de Fora, como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

Aprovada em: 19 de agosto de 2016

BANCA EXAMINADORA

Prof. Dr. Alex Borges Vieira - Orientador
Universidade Federal de Juiz de Fora

Prof. Dr. Moisés Vidal Ribeiro - Coorientador
Universidade Federal de Juiz de Fora

Prof. Dr. Luiz Filipe Menezes Vieira
Universidade Federal de Minas Gerais

Prof. Dr. Marcos Augusto Menezes Vieira
Universidade Federal de Minas Gerais

Prof. Dr. Marcelo Ferreira Moreno
Universidade Federal de Juiz de Fora

AGRADECIMENTOS

Primeiramente agradeço à minha mãe Rossana, ao meu pai Luís Sérgio, aos meus irmãos (Júlia, Pedro e Luana), ao meu amor Fernanda, à minha segunda mãe/sogra Maria Goretti, às minhas tias, tios, primos e primas, por todo apoio, suporte, presença e amor.

Aos meus orientadores, prof. Alex B. Vieira e prof. Moisés Ribeiro, pela paciência e orientação. Aos membros da banca, pelos apontamentos e dicas de melhoria. Ao prof. Fabrício e ao Diogo pelas oportunidades. Ao quase Dr. Roberto, já que sem ele esse trabalho não teria saído da melhor forma. Aos amigos da Smarti9 e LCOM, pelo engrandecimento profissional. E a todos que, de alguma forma, tornaram o momento possível.

RESUMO

Recentemente, Power Line Communication (PLC) tem atraído o interesse de pesquisadores de todo mundo. Contudo, sistemas PLC enfrentam diversos desafios em termos da degradação da qualidade de comunicação de dados. Para superar tais problemas, propomos o CodePLC, um protocolo MAC para PLC que utiliza *network coding*. Utilizamos um nó intermediário para intermediar a comunicação, armazenar e repassar uma combinação linear de pacotes PLC. Avaliamos o desempenho do CodePLC através da simulação de uma topologia comum em um sistema PLC baseado no esquema *Orthogonal Frequency Division Multiplexing with Time Domain Multiple Access* (TDMA-OFDM), sob uma ampla gama de cenários. Em suma, os resultados mostram que, para transmissões do tipo *broadcast*, o uso de *network coding* melhora desempenho geral do sistema PLC baseado no TDMA-OFDM. De fato, quando comparado a um sistema que utiliza uma camada MAC PLC *multihop stop&wait* tradicional, podemos observar um pico de aumento de 110% no valor de pico do *goodput*. Além disso, nosso protocolo reduz em até 71,4% a ocupação dos *buffers* da rede. Finalmente, CodePLC pode reduzir a latência média em até quatro vezes.

Palavras-chave: PLC, Network Coding, TDMA-OFDM.

ABSTRACT

Recently, power line communication (PLC) has attracted interest of worldwide researchers. However, PLC systems face several challenges which degrade data communication quality. To overcome such issues, we propose CodePLC, a network coding PLC MAC protocol. We use a single relay node to intermediate communication, storing and forwarding linear combinations of PLC data packets. We evaluate CodePLC performance through simulations of a common topology for a PLC system based in Orthogonal Frequency Division Multiplexing with Time Domain Multiple Access (TDMA-OFDM) under a wide range of scenarios. In summary, our results show that in a broadcast like transmission, the use of network coding enhances overall system performance. In fact, when compared to a traditional PLC system using multihop stop&wait MAC layer protocol, we have observed an peak of 110% goodput increase. Moreover, our protocol reduces in 71,4% network occupancy buffers. Finally, CodePLC reduces mean latency by four times.

Key-words: PLC, Network Coding, TDMA-OFDM.

LISTA DE ILUSTRAÇÕES

| | |
|--|----|
| Figura 1 – Estrutura de pilha de protocolos de um sistema PLC IEEE 1901.2 (Cisco Developer Network, 2016) | 13 |
| Figura 2 – Perfil de <i>downlink</i> de alocação de um <i>frame</i> TDMA-OFDM | 15 |
| Figura 3 – Cenário PLC <i>indoor</i> em um cômodo de uma residência. | 16 |
| Figura 4 – Topologia do cenário estudado. | 17 |
| Figura 5 – Exemplo de uma transmissão em uma rede <i>multihop</i> | 18 |
| Figura 6 – Sistema de topologia do tipo borboleta | 19 |
| Figura 7 – Aplicação de <i>network coding</i> em um sistema de topologia do tipo borboleta | 20 |
| Figura 8 – Topologia de três nós e dois enlaces | 20 |
| Figura 9 – Utilização de <i>network coding</i> em uma topologia de três nós e dois enlaces | 21 |
| Figura 10 – Exemplo de escuta e codificação oportunista para a implementação de <i>network coding</i> | 22 |
| Figura 11 – Processo de codificação XOR bit a bit de uma mensagem | 22 |
| Figura 12 – Exemplo de codificação XOR bit a bit de uma mensagem | 23 |
| Figura 13 – Processo de decodificação XOR bit a bit de uma mensagem. | 23 |
| Figura 14 – Exemplo de utilização de <i>network coding</i> em TDMA. | 27 |
| Figura 15 – Exemplo do protocolo de ARQ do CodePLC. | 29 |
| Figura 16 – Estrutura de um MPDU em uma subcamada MAC com CodePLC | 30 |
| Figura 17 – Exemplo de codificação de MPDU's no nó E | 30 |
| Figura 18 – Fluxograma do algoritmo de transmissão e recepção em um nó comum | 31 |
| Figura 19 – Fluxograma do algoritmo de transmissão e recepção em um nó codificador | 32 |
| Figura 20 – Ocupação total de <i>buffers</i> do sistema ao fim de 200 <i>frames</i> | 36 |
| Figura 21 – Média de ocupação de <i>buffer</i> no sistema em relação à variação de tamanho de MPDU. | 37 |
| Figura 22 – Média de ocupação de <i>buffer</i> no sistema em relação ao volume de tráfego gerado. | 38 |
| Figura 23 – Distribuição do <i>goodput</i> médio dos nó do sistema ao fim de 100 MPDU's corretamente recebidas. | 39 |
| Figura 24 – Média do <i>goodput</i> do sistema em relação à variação de tamanho de MPDU. | 39 |
| Figura 25 – Média do <i>goodput</i> do sistema em relação à variação de tamanho de MPDU. | 40 |
| Figura 26 – Distribuição da latência média (em <i>frames</i>) em um sistema com geração de MPDU a 0,7. | 41 |
| Figura 27 – Média do latência do sistema em relação à variação da geração de dados. | 41 |
| Figura 28 – Distribuição de jitter (em <i>frames</i>) em um sistema com geração de MPDU a 0,7. | 42 |
| Figura 29 – <i>Jitter</i> médio do sistema em relação à variação da geração de dados. | 42 |

| | |
|---|----|
| Figura 30 – Esquema de disponibilidade do canal PLC para comunicação de dados | 43 |
| Figura 31 – Médias do <i>goodput</i> do sistema em relação à variação de utilização do CodePLC | 44 |
| Figura 32 – Distribuição da latência do sistema em relação à variação da disponibilidade do canal. | 45 |
| Figura 33 – Distribuição do <i>goodput</i> médio do sistema em relação para o cenário de 1 nó indisponível. | 46 |
| Figura 34 – Distribuição da ocupação de buffer média do sistema em relação para o cenário de 1 nó indisponível. | 46 |
| Figura 35 – Distribuição da latência média do sistema em relação para o cenário de 1 nó indisponível. | 47 |

LISTA DE ABREVIATURAS E SIGLAS

| | |
|---------|--|
| PLC | Power Line Communication |
| TDMA | Time Division Multiple Access |
| OFDM | Orthogonal Frequency-division Multiplexing |
| MAC | Media Access Control |
| QoS | Quality of Service |
| ARQ | Automatic Repeat reQuest |
| PER | Packet Error Rate |
| BER | Bit Error Rate |
| HS-OFDM | Hermitian Symmetric - Orthogonally Frequency Division Multiplexing |
| BPSK | Binary Phase Shift Keying |
| SBRC | Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos |
| LCN | Local Computer Networks |
| OSI | Open Systems Interconnection |
| IEEE | Institute of Electrical and Electronics Engineers |
| WiMAX | Worldwide Interoperability for Microwave Access |
| ITU-T | Telecommunication Standardization Sector |
| ARQ | Automatic Repeat reQuest |
| CRC | Cyclic Redundancy Check |
| MPDU | MAC Protocol Data Unit |
| CSMA/CA | Carrier Sense Multiple Access with Collision Avoidance |
| CSMA/CD | Carrier Sense Multiple Access with Collision Detection |

SUMÁRIO

| | | |
|----------|---|-----------|
| 1 | INTRODUÇÃO | 10 |
| 1.1 | CONTRIBUIÇÕES DA DISSERTAÇÃO | 11 |
| 1.2 | ESTRUTURA DA DISSERTAÇÃO | 12 |
| 2 | CENÁRIO E DEFINIÇÃO DO PROBLEMA | 13 |
| 2.1 | SOBRE POWER LINE COMMUNICATIONS (PLC) | 13 |
| 2.2 | DEFINIÇÃO DO PROBLEMA | 16 |
| 3 | NETWORK CODING | 19 |
| 3.1 | CONSIDERAÇÕES SOBRE NETWORK CODING | 19 |
| 3.2 | XOR | 22 |
| 4 | TRABALHOS RELACIONADOS | 24 |
| 5 | O PROTOCOLO CODEPLC | 26 |
| 5.1 | O PROTOCOLO DE COMUNICAÇÃO CODEPLC | 27 |
| 5.1.1 | Controle do meio de transmissão | 28 |
| 5.1.2 | Gerenciamento de erros e retransmissão | 28 |
| 5.1.3 | Codificação de MPDU's | 29 |
| 5.2 | ALGORITMO CODEPLC | 30 |
| 5.3 | LIMITAÇÕES DO CODEPLC | 32 |
| 6 | METODOLOGIA E AVALIAÇÃO | 34 |
| 6.1 | METODOLOGIA | 34 |
| 6.2 | OCUPAÇÃO DE BUFFERS | 36 |
| 6.3 | GOODPUT | 38 |
| 6.4 | LATÊNCIA e <i>JITTER</i> | 40 |
| 6.5 | VARIAÇÃO NA DISPONIBILIDADE DO NÓ CODIFICADOR | 43 |
| 6.6 | INDISPONIBILIDADE DE NÓS VIZINHOS | 45 |
| 7 | CONCLUSÃO E TRABALHOS FUTUROS | 48 |
| 7.1 | CONTRIBUIÇÕES | 48 |
| 7.2 | TRABALHOS FUTUROS | 48 |
| | REFERÊNCIAS | 50 |

1 INTRODUÇÃO

Comunicação via rede de energia elétrica (Power Line Communications - PLC) atrai cada vez mais a atenção, tanto da comunidade acadêmica, quanto da indústria. A demanda por esse tipo de rede se dá por sua flexibilidade e facilidade na conexão de dispositivos, baixo custo de instalação de infraestrutura quando comparado a tecnologias sem fio e, até mesmo, por suas altas taxas de transmissão (EZZINE et al., 2014). Esse interesse crescente se traduz em uma projeção de crescimento maior que 15% e um valor mundial de investimentos na área em torno de US\$11.27 bilhões até 2022 (Global Industry Analysts, 2015; Markets and Markets, 2016). Interesse esse impulsionado pela crescente demanda de empresas de energia elétrica para viabilizar tecnologias de comunicação de dados e aplicações que utilizam sistemas PLC.

Exemplos de aplicação de tecnologias PLC mais comuns são sistemas veiculares (automotivo, naval, aeroespacial), *backbones* de alta velocidade, redes banda larga de acesso à internet, aplicações *smart grid* e na área de IoT. Para cada uma delas, é fundamental que as novas gerações de dispositivos de comunicação de dados sejam confiáveis, de baixo-custo e sustentáveis. Isso gera uma demanda por soluções que utilizem um menor poder de processamento e mais resiliência, podendo oferecer, inclusive, garantias de qualidade de serviço (QoS) (CAMPOS, 2013).

As redes de energia elétrica são utilizadas para transmitir e distribuir energia em frequências muito baixas e não foram projetadas para a comunicação de dados (FERREIRA; LAMPE; NEWBURY, 2010). Ainda hoje, técnicas para transmissão de dados utilizadas em tecnologias PLC podem apresentar altas taxas de erro devido, sobretudo, à dinâmica do sistema elétrico de potência e características intrínseca ou sua infraestrutura (GAO et al., 2008). Por esse motivo, há uma contínua investigação em torno da utilização de protocolos de comunicação adequados para introduzir melhorias que resultem em melhor confiabilidade e vazão em sistemas PLC. A maior parte das contribuições nessa área discute os tipos de melhorias que podem ser introduzidas na camada física, tal como o emprego de protocolos de cooperação com técnicas de combinação de sinais provenientes de canais distintos (BIAGI, 2011; CHENG; CAO; YANG, 2013; KIM et al., 2012; VALENCIA; OLIVEIRA; RIBEIRO, 2014).

De acordo com a literatura, há a possibilidade de implementação, tanto de protocolos de cooperação, quanto de *network coding*, em camadas superiores para buscar um aumento de resiliência e confiabilidade em uma comunicação de dados através de uma rede de energia elétrica (NOORI; LAMPE, 2013; BILBAO et al., 2014). Esse interesse deve-se ao sucesso em termos de obtenção de melhoria de desempenho que o *network coding* apresenta em sistemas sem fio (FAROOQIA et al., 2014), cria uma expectativa que a utilização dessa técnica em sistemas PLC seja positiva.

Os ganhos em vazão e resiliência na rede, a partir do uso de *network coding*, são bem conhecidos para cenários de *broadcast* e *multicast* em redes sem fio (FRAGOULI; BOUDEC; WIDMER, 2006). Entretanto, o uso de *network coding* ainda é pouco explorado em sistemas PLC. De fato, existem poucos trabalhos na literatura que propõem o uso desse tipo de técnica em PLC. Como principal consequência, ainda há uma falta de conhecimento sobre o comportamento de tais sistemas na prática. Por exemplo, não são bem delimitadas as circunstâncias nas quais *network coding* na camada de controle de acesso ao meio (MAC) poderia melhorar o desempenho de redes PLC de forma geral. Mesmo que os trabalhos conhecidos demonstrem ganhos na vazão, na latência ou na resiliência da rede, pouco é tocado no que se refere à sobrecarga imposta à sinalização (ou controle) pelo uso de protocolos com *network coding* ou como essa técnica pode impactar em questões de gargalos e congestionamentos de rede.

Nesse contexto, dada a necessidade contínua pela introdução de melhorias no cenário PLC e os notórios benefícios de *network coding* em termos de vazão e resiliência em redes, esse trabalho propõe um mecanismo de *network coding* na camada de acesso ao meio (MAC) em PLC. Nosso protocolo de comunicação, intitulado CodePLC, utiliza o auxílio de um nó intermediário, que é responsável pela codificação das unidades de dados (MPDU's) e o repasse aos nós usuários.

O protocolo proposto melhora o *goodput*, latência e utilização de *buffer* dos nós do sistema de uma rede PLC residencial. Para demonstrar tais melhorias, fizemos simulações de um sistema baseado no PLC TDMA-OFDM, utilizando taxas de erro de pacotes (PER), obtidas através de taxas de erro de bits (BER) medidas em um ambiente real. As medições foram feitas em uma residência na cidade de Juiz de Fora e são relativas à um circuito elétrico em um cômodo, com distâncias entre 2 a 10 metros entre os nós de um mesmo circuito (COLEN et al., 2012). A campanha de medição caracterizou uma rede elétrica de baixa tensão como um meio de comunicação de dados quando a banda de frequência é entre 1,705 e 50 MHz. Foi utilizado um esquema HS-OFDM (*Hermitian Symmetric - Orthogonally Frequency Division Multiplexing*) (RIBEIRO et al., 2014) transmitindo símbolos com a modulação de chaveamento binário de fase (BPSK) e uma potência $P = 30$ dBm.

1.1 CONTRIBUIÇÕES DA DISSERTAÇÃO

Em suma, nossas contribuições são: (i) criamos um protocolo de subcamada MAC que utiliza uma técnica de *network coding* em PLC. (ii) utilizamos o protocolo desenvolvido para delimitar os cenários nos quais o uso da técnica de *network coding* é viável e oferece melhorias no desempenho de redes PLC. Para analisar o protocolo, foram utilizadas métricas de ocupação de *buffer*, *goodput*, latência e, por fim, disponibilidade de utilização de um nó intermediário.

Nossos resultados numéricos apontam que, em uma transmissão tipo *broadcast* que atravessa um nó intermediário, o uso de *network coding* pode aumentar o *goodput* médio em até 110% em relação a um cenário onde há um nó intermediário. Mais ainda, a ocupação média dos *buffers* dos dispositivos de rede pode diminuir em até 71,4%. Finalmente, a latência observada em um cenário onde é aplicado o protocolo CodePLC, em comparação ao cenário comum, é até 84,7% menor, a partir de uma taxa de geração de dados dos nós de probabilidade 0,3. Uma análise sobre a variação de disponibilidade de um nó que executa *network coding* também demonstra que a não-utilização dos enlaces de transmissão em certo período de tempo ainda apresenta bons resultados em termos de vazão, comparado ao cenário padrão.

A utilização de *network coding* em PLC proposta neste trabalho, bem como a definição de cenário e problema, é tema de nosso artigo publicado no Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC 2016) (SILVEIRA et al., 2016b). A definição do protocolo CodePLC, a aplicação do mesmo e a análise da variação de disponibilidade do nó intermediário foram tema de *short paper* aprovado no *IEEE Conference on Local Computer Networks* (LCN 2016) (SILVEIRA et al., 2016a).

1.2 ESTRUTURA DA DISSERTAÇÃO

Este trabalho está organizado da seguinte forma: o Capítulo 2 apresenta a motivação para utilização de PLC em sistemas de comunicação e a definição do sistema PLC utilizado neste trabalho e trata da definição do escopo do problema e delimitação do cenário em que a implementação e consequente análise serão feitas; o Capítulo 3 trata de definições sobre *network coding* e sobre a técnica que utilizaremos no protocolo proposto; no Capítulo 4, apresentaremos uma análise do estado da arte da aplicação de técnicas de *network coding* em camada de enlace de sistemas PLC e uma definição dos avanços na utilização de *network coding* em cenários PLC; no Capítulo 5, apresentaremos o algoritmo CodePLC, bem como detalhes de sua implementação e questões a serem levantadas sobre seu desempenho e limitações; e no Capítulo 6 iremos analisar os resultados da aplicação do protocolo proposto no cenário apresentado, que foi simulado e alimentado com dados de medição.

2 CENÁRIO E DEFINIÇÃO DO PROBLEMA

2.1 SOBRE POWER LINE COMMUNICATIONS (PLC)

A possibilidade de utilizar redes de energia elétrica para transmissão de dados motiva a comunidade científica e a indústria a procurarem soluções que superem os desafios reconhecidos ao trafegar informações em uma meio que não foi originalmente pensado para esse fim. Esse interesse crescente é refletido em um maior investimento em pesquisas na área de PLC (Global Industry Analysts, 2015; Markets and Markets, 2016). Uma das motivações da crescente utilização da tecnologia PLC em sistemas de comunicação está no baixo custo de instalação já que, em ambientes domésticos, veiculares e de redes de dados, a maior parte da infraestrutura utilizada já está previamente instalada (MANNAN; D.K.SAXENA; BANDAY, 2014).

A implementação dos componentes principais de um sistema PLC da-se nas camadas 1 e 2 do modelo de referência OSI, respectivamente, camadas física e enlace (International Organization for Standardization, 1996). A figura 1 mostra a implementação de um sistema PLC do ponto de vista da pilha de protocolos. Nela, é representado um sistema PLC para Internet das Coisas que utiliza os protocolos de PLC IEEE 1901.2, para camada física e enlace (IEEE-SA, 2013).

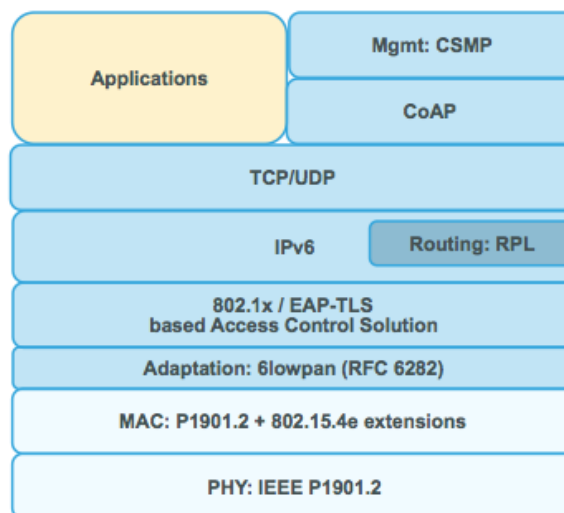


Figura 1 – Estrutura de pilha de protocolos de um sistema PLC IEEE 1901.2 (Cisco Developer Network, 2016)

A camada física tem como objetivo, de uma forma geral, condicionar os dados binários para que sejam transmitidos através da rede de energia elétrica (MALATHI; VANATHI, 2007). Devido às características dos canais PLC o esquema OFDM (do inglês, *Orthogonal frequency-division multiplexing*) é utilizado nas principais normas e

padronizações em PLC (TSS-ITU, 2014; TSS-ITU, 2012; IEEE-SA, 2010). Consiste na divisão da banda do canal em portadoras paralelas. Cada subcanal tem uma subportadora associada, sendo cada uma delas digitalmente e individualmente moduladas, considerando detecção coerente ou não coerente.

Na camada de enlace, responsável pela ligação entre dois nós adjacentes, a subcamada MAC (*Media Access Control*) é responsável pelo controle do acesso ao meio. Dentre as implementações possíveis, o controle de acesso ao meio em sistemas PLC pode ser feito com o uso de protocolos como o CSMA (*Carrier Sense Multiple Access*), TDMA (*Time Division Multiple Access*), Híbrido (que utiliza ambos) (IEEE-SA, 2013; IEEE-SA, 2010; TONELLO; CORTES; D’ALESSANDRO, 2009). Em normas como IEEE-SA (2013) e IEEE-SA (2010), a subcamada MAC acumula serviços relativos à camada de enlace como um todo, tais como: enquadramento de pacotes, entrega confiável, controle de fluxo e detecção e correção de erros.

O uso da tecnologia PLC para comunicação de dados é desafiador, uma vez que esse tipo de tecnologia tem como desafios minimizar as degradações introduzidas no sinal, quando propagado através da rede de energia elétrica, dentre as quais destacam-se: canais caracterizados por desvanecimento seletivo em frequência e variante no tempo, crescente atenuação do sinal quando a distância e a frequência aumentam, presença de ruído impulsivo de elevada potência e variação temporal da impedância de acesso (FERREIRA; LAMPE; NEWBURY, 2010). Por apresentar desafios similares e até maiores do que as redes sem fio (GÜZELGÖZ et al., 2011), é natural que técnicas utilizadas para minimizar esses problemas sejam objeto de pesquisa em sistemas PLC, em especial na camada de enlace e subcamada MAC. Um exemplo são técnicas de cooperação, que propõem a utilização de um ou mais *relay* entre os nós origem e destino, visando ganhos em termos de taxa de dados, resiliência e confiabilidade (OLIVEIRA et al., 2015).

Esse trabalho utiliza um cenário típico de sistema PLC banda-larga residencial, que faz uso do esquema OFDM e de Acesso Múltiplo por Divisão de Tempo. No contexto da interface entre as camadas física e subcamada MAC PLC, *frame* TDMA-OFDM é a unidade de divisão no canal PLC para a comunicação de dados, sempre iniciada por uma sequência conhecida de bits (preâmbulo). Um *frame* PLC TDMA-OFDM é dividido em *slots* de tempo (YEH; KWON; KIM, 2003).

O perfil de alocação de *slots* TDMA-OFDM em um *frame* i utilizado neste trabalho pode ser visto na figura 2. Dentro dos *time slots* t_u , referente a cada um dos nós u , ocorre o *uplink* de dados do respectivo usuário e o recebimento dos mesmos pelos usuários que compartilham o canal com o transmissor. Esse *time slot* é único para cada nó da topologia e é garantido que não há qualquer disputa pelo canal quando um dos nós assumir a transmissão de dados. No *slot* de tempo de controle, são enviados ACK e NACK relativos às transmissões no *frame* anterior ($i - 1$), além de outros dados de controle. A divisão do

tempo de controle também pode se dar por *slot* de cada usuário.

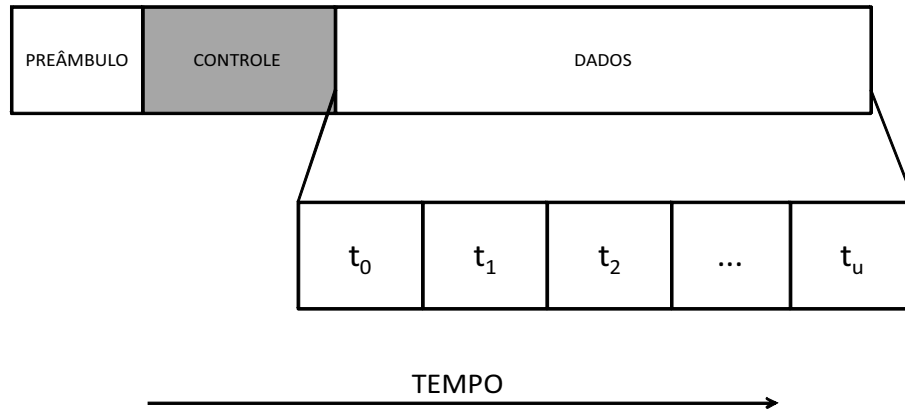


Figura 2 – Perfil de *downlink* de alocação de um *frame* TDMA-OFDM

A literatura mostra que existem vantagens para o controle do meio de transmissão quando utilizamos um esquema TDMA-OFDM em uma rede de comunicação. Por exemplo, o conhecimento prévio de qualquer intenção ou disponibilidade de transmissão ou recepção de dados, permite que qualquer elemento controlador possa planejar e prever a utilização de recursos de canal (HOESEL et al., 2004). Esse conhecimento pode se dar com a troca de informações de controle, no slot de tempo referente a esse tipo de informação, como acontece na implementação do padrão IEEE 802.16 (conhecido como WiMAX)(IEEE 802.16 Working Group on Broadband Wireless Access Standards, 2012).

O maior controle na utilização de recursos também pode ser fundamental para que requisitos de QoS sejam garantidos. A garantia de disponibilidade de transmissão para tráfego, torna sistemas de comunicação TDMA-OFDM mais previsíveis em aplicações multimídia e de tempo real, por proporcionar uma variação de atraso de dados (*jitter*) quase sempre fixa e previsível (HADDED et al., 2015).

Neste trabalho, consideramos um ambiente específico de um sistema PLC doméstico, onde há uma sequência de dispositivos interconectados por um circuito elétrico, contemplando quatro nós usuários e mais um dispositivo que atua como intermediário da comunicação entre os nós que não dividem um enlace a nível lógico. Esse último pode estar presente em um quadro de energia dentro de um cômodo, como podemos observar na figura 3.

Note que cada nó pode contemplar diversos dispositivos geradores de tráfego de dados, como computadores, sensores e aparelhos eletrônicos inteligentes. Assim como acontece em meios sem fio, alguns destes nós podem não ter contato em nível lógico com outros na rede, pelos desafios inerentes ao meio, como a crescente atenuação do sinal quando a distância e frequência aumentam.

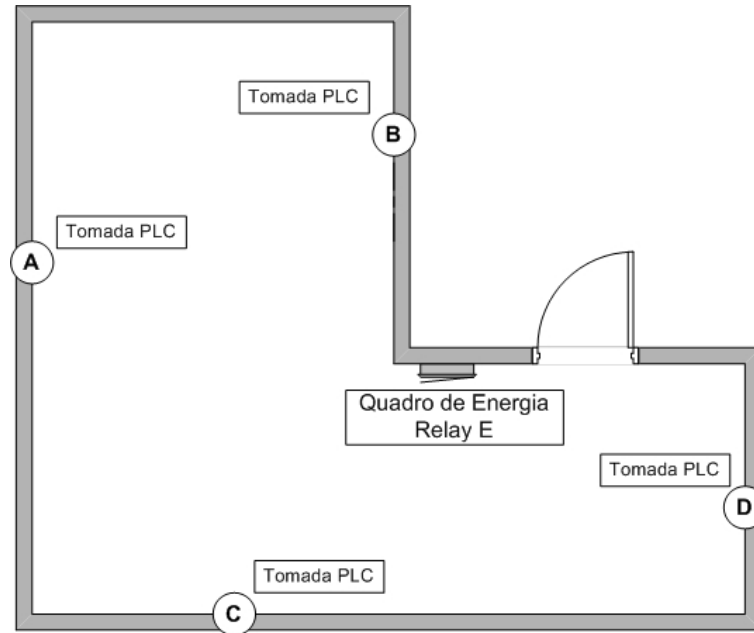


Figura 3 – Cenário PLC *indoor* em um cômodo de uma residência.

2.2 DEFINIÇÃO DO PROBLEMA

A figura 4 retrata um ambiente real de utilização do sistema PLC. É possível observar que um transceptor se comunicam com múltiplos transceptores. Cada nó é equipado com um *buffer* de dados no qual unidades de dados são armazenadas para posterior transmissão. Mais precisamente, de acordo com a topologia definida na figura 4, temos cinco nós (A, B, C, D e E). Não há conexão completa entre esses nós, e portanto, as linhas nessa figura indicam o compartilhamento do enlace entre os dispositivos PLC, caracterizando um *overhearing* perfeito (A e D podem escutar B e C; e vice-versa). O nó E, que exerce o papel de intermediário, tem capacidade de comunicação com todos os outros. Assumimos a existência de fluxos de dados de origem em cada um dos nós, que atravessa o nó coordenador E e tem como destino o nó oposto. A geração desses dados está sincronizada e respeita uma frequência determinada por uma probabilidade.

Cada enlace i , $i \in \{AB, AC, BD, CD, AE, BE, CE, DE\}$ possui taxas de erro de pacote (PER_i) independentes. Os valores de PER_i são extraídos a partir da taxa de erro de bits (BER_i), obtida na camada física para o i -ésimo enlace. Seu cálculo é dado por $PER_i = 1 - (1 - BER_i)^{N_i}$, onde N_i corresponde ao tamanho do pacote. Essa fórmula representa a probabilidade de ocorrência de erro em, pelo menos, um bit de um pacote com N_i bits (OLIVEIRA et al., 2015).

Os valores de BER foram estimado a partir de campanhas de medição realizadas em uma área típica urbana brasileira (COLLEN et al., 2012), onde foram coletadas informações sobre taxas de erros de bit em mais de 36 mil combinações de variações de distâncias relativas, considerando a frequência de banda entre 1,705 MHz e 50 MHz, com distâncias

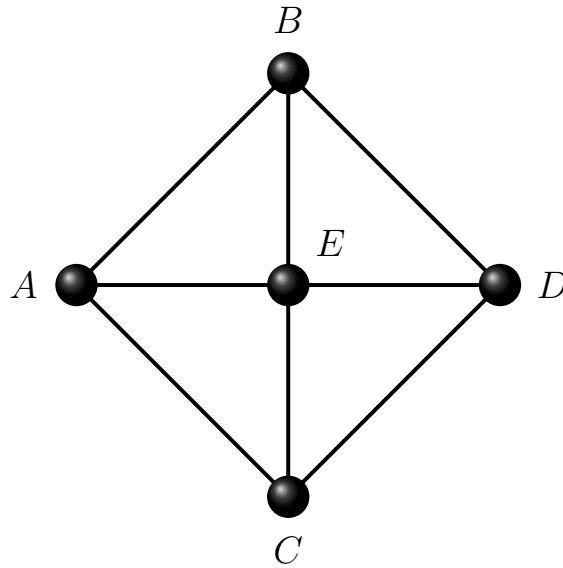


Figura 4 – Topologia do cenário estudado.

que variaram de 2 a 10 metros em cômodos de uma residência. Neste trabalho, foram utilizados dados das medições realizadas em canais com a potência total de 30 dBm, com taxa média de erro de 18% em cada enlace. Para a geração dos dados de campo utilizamos um esquema HS-OFDM sem codificação, juntamente com a modulação de chaveamento de fase binário (BPSK) e alocação uniforme de potência. Consideramos também perfeita sincronização e conhecimento completo do estado do canal por parte dos receptores.

Na topologia apresentada na figura 4, um fluxo de dados que inicie em uma das extremidades pode ter como destino uma extremidade oposta, sem possibilidade de comunicação direta. Em sistemas PLC que implementam padrões reconhecidos para camada de enlace, as soluções de transmissão se baseiam em encaminhamento de pacotes do tipo *multihop*, onde um nó ou um conjunto de nós servem como saltos intermediário para que um dado chegue ao destino. É o caso da recomendação ITU-T G.9903 e do padrão IEEE 1901, que definem padrões de implementação de sistemas PLC tanto para banda larga, quanto para banda estreita (IEEE-SA, 2010; TSS-ITU, 2014). Ambos os protocolos utilizam a técnica de vetor de distância para definir o percurso que o pacote percorrerá.

No caso do cenário proposto na figura 4, um fluxo que inicie em A e tenha como destino D (ou vice-versa) poderá, a qualquer critério, utilizar B, C e E como nós intermediários, já que consideramos as arestas com mesmo peso. No caso do sistema no problema analisado, o critério é a aleatoriedade. Neste trabalho, trataremos o cenário no qual esse procedimento é adotado como “cenário comum” ou “PLC comum”. A figura 5 ilustra um dos possíveis procedimentos de transmissão *multihop*.

A literatura nos mostra que latência é um problema em sistemas que utilizam

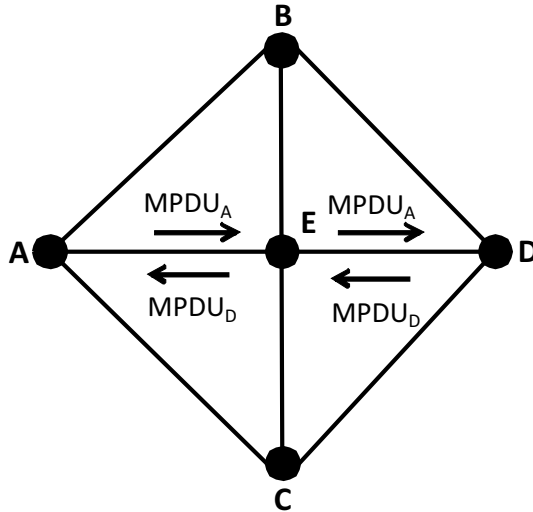


Figura 5 – Exemplo de uma transmissão em uma rede *multihop*

transmissões *multihop* e esquema TDMA. Redes que utilizam um esquema TDMA podem introduzir um atraso de escalonamento se, no mesmo caminho, *uplink* de um nó que atua como roteador for programado para transmitir antes de um *downlink* desse roteador (DJUKIC; ROGERS; VALAEE, 2009). Além disso, transmitir através de *multihop* pode introduzir um atraso de tempo de *buffer* se a vazão deste nó não for suficiente para repassar pacotes próprios e pacotes roteados, ou seja, na existência de um congestionamento. Esse atraso pode ser elevado, já que é acumulado em cada salto executado em uma rota (GOLESTANI; COMMUN, 2002). Algumas técnicas podem ser utilizadas para atenuar ou dirimir esses problemas (GOLESTANI; COMMUN, 2002; LEU; CHEN; LIU, 2014; LIANG et al., 2010), sendo que cada técnica tem custos computacionais e estruturais diferentes.

Utilizar um sistema *multihop* também gera uma necessidade de controle de retransmissão que seja feito por outras instâncias da camada de enlace (ou camadas superiores), inclusive com a possível necessidade de tráfego de controle extra (TSS-ITU, 2014). Isso porque, dado uma rota definida para um pacote, a confirmação de recebimento se dá apenas na comunicação ponto-a-ponto. As perdas relativas à pacotes roteados (que não estão no mesmo enlace) necessitam de um outro tipo de controle, que não é presente nos serviços previstos de uma subcamada MAC. Por exemplo, na figura 5, se o nó E perde o pacote de A destinado a D, o conhecimento da necessidade de retransmissão por A não tem suporte da subcamada MAC.

Nosso trabalho visa procurar uma técnica que também resolva problemas relativos ao congestionamento causados pelo uso de um esquema *multihop* em um sistema PLC que utiliza esquema TDMA-OFDM e diminuir problemas de encaminhamento sem a necessidade de que muitos recursos computacionais sejam utilizados para tanto.

3 NETWORK CODING

3.1 CONSIDERAÇÕES SOBRE NETWORK CODING

Yeung et al. (2006) define como *Network Coding* o método utilizado para otimizar um fluxo de dados em uma rede a partir da transmissão de uma combinação das mensagens trafegadas. Essa combinação é a composição de duas ou mais mensagens, que chegam ao receptor e são deduzidas para, só após isso, serem desmontadas e interpretadas diretamente. Neste trabalho, seguimos a definição de Ahlswede et al. (2000), que entende como “codificação” e “decodificação”, respectivamente, os processos de operação e posterior dedução das mensagens que compõem a combinação recebida por um nó em um rede.

Em redes de comutação de pacotes, o fluxo de dados é definido por mensagens, que partem de uma origem em direção a um destino. Na estação transmissora, a mensagem de saída é dividida em pacotes, cada um dos quais contém alguns dos dados da mensagem. Os pacotes não viajam necessariamente pela mesma rota, mas todos eles deveriam chegar a um mesmo destino, cujo o dispositivo receptor é capaz de reagrupá-los para constituir a mensagem original. O maior problema nesse método é que, quando as taxas de dados ou de erros são altas, é comum a criação de gargalos, congestionamentos e rotas subutilizadas, criando um aumento de latência e perda de desempenho em geral (YEUNG et al., 2006).

Como exemplo, podemos ver na figura 6 uma rede de topologia do tipo borboleta. O nó s indica a origem do tráfego de dados. O nós $t1$ e $t2$ são os nós de destino das transmissões e os nós 1, 2, 3 e 4 são nós intermediários, responsáveis pelo roteamento dos pacotes transmitidos por s ($m1$ e $m2$). Cada arco representa um enlace entre dois nós e capaz de transportar um único pacote. O fluxo de dados em uma rede de comutação parte de s e precisa percorrer uma das rotas entre a origem e os nós $t1$ e $t2$.

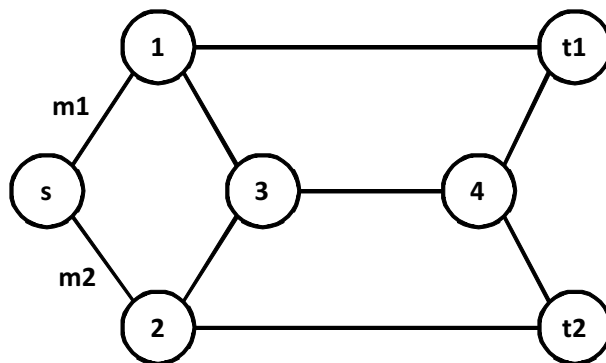


Figura 6 – Sistema de topologia do tipo borboleta

Utilizando *network coding*, alguns nós da rede tem a capacidade de serem codificadores. Dados que chegam de duas ou mais origens podem ser combinados em um único pacote. Na figura 7, podemos ver o mesmo exemplo da topologia da figura 6, com a utilização de *network coding*.

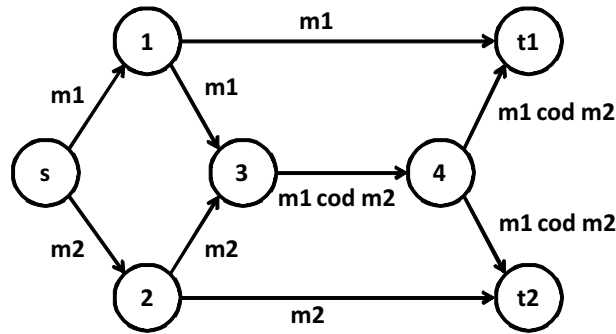


Figura 7 – Aplicação de *network coding* em um sistema de topologia do tipo borboleta

Dois pacotes, $m1$ e $m2$, saem da origem s em direção a $t1$ e $t2$ por rotas diferentes. Esses pacotes chegam em um nó com capacidade de combiná-los em um único pacote (nó 3). Essa informação combinada chega a $t1$ e $t2$ que, em posse de uma das mensagens, é capaz de interpretar a outra.

Este método aumenta os ganhos em vazão de uma rede através da minimização do número e da gravidade dos pontos congestionamento (HO; LUN, 2008). Em *network coding*, os dados não dependem apenas de uma mensagem transmitida, mas também sobre o conteúdo de outras mensagens que estejam compartilhando uma rota ou um enlace no momento de suas respectivas transmissões. Os ganhos em desempenho vão depender da topologia, além da frequência e severidade dos gargalos e congestionamentos. A literatura mostra que nenhum caso a utilização de *network coding* irá reduzir a vazão em comparação ao método tradicional de roteamento (YEUNG et al., 2006).

Network coding também é capaz de prover um aumento de robustez em termos de erros do enlace. A figura 8 representa um esquema tradicional de rede com três nós (1, 2 e 3) onde cada arco entre os nós representa um enlace direto, com capacidade para uma mensagem. Os nós 1 e 2 têm a capacidade de transmitir apenas um pacote por vez em seus enlaces de transmissão. Em sistemas tradicionais, um controle de erro no enlace através de um processo de mensagem de confirmação que é capaz de identificar a perda de $m1$ ou $m2$ em um mesmo enlace (1 e 2). Mas esse controle não é suficiente para que o nó 1 identifique a falha na transmissão de $m1$ quando ela ocorrer no enlace entre os nós 2 e 3.

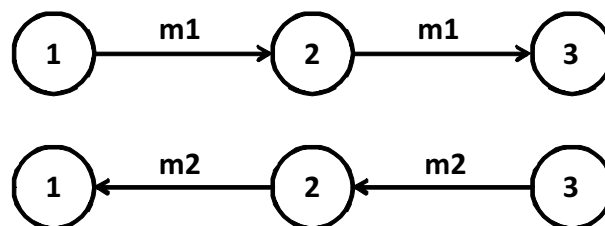


Figura 8 – Topologia de três nós e dois enlaces

Em um sistema de mesma topologia onde *network coding* é aplicado no nó 2, as

perdas ocorrerão em uma comunicação de nós em um mesmo enlace. Na figura 9, as mensagens $m1$ e $m2$ podem ser combinadas no nó 2, que transmite apenas um pacote com a combinação das duas mensagens. Nessa comunicação só há mensagens trocadas entre enlaces.

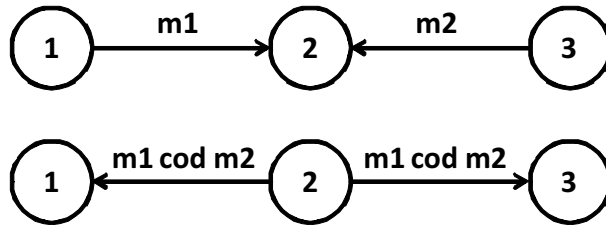


Figura 9 – Utilização de *network coding* em uma topologia de três nós e dois enlaces

Para aplicação de *network coding* em um sistema é importante destacarmos dois conceitos: o de escuta oportunista e o de codificação oportunista. O conceito de escuta oportunista, já estudado e utilizado em sistemas sem fio, também está presente em sistemas PLC (TSENG et al., 2015). Redes em que os nós se comunicam pelo mesmo meio de transmissão criam várias oportunidades para que todos eles escutem todos os pacotes trafegados na rede. Se fizermos os nós armazenarem todos esses dados, podemos utilizar o conhecimento desses pacotes transmitidos por outros usuários próximos, para obter pacotes que façam parte de uma combinação gerada pela *network coding*. Dessa forma, teremos o número necessário de pacotes para realizar o processo de decodificação (KATABI et al., 2006).

Codificação oportunista se refere à entendermos quando e em quais pacotes utilizar *network coding*. Essa questão tem que ser respondida pelo nó de forma local, sem necessitar de consulta à outros nós da rede (KATTI et al., 2005).

A figura 10 exemplifica esses dois conceitos. As mensagens $m1$ e $m2$ partem, respectivamente, das origens $s1$ e $s2$ em direção aos destinos múltiplos $t1$ e $t2$. Para isso, as duas serão transmitidas ao nó 1, que executa o *network coding* em ambas, repassa essa combinação ao nó 2 que, eventualmente, irá repassar a mensagem codificada aos nós destino. Os arcos representam um enlace com capacidade para uma transmissão por vez. Como se trata de um meio *broadcast*, $s1$ e $s2$ enviam suas respectivas mensagens diretamente para $t1$ e $t2$. Com posse de uma única mensagem das duas que compõe o pacote codificado, é possível que os dois destinos obtenham o pacote que falta. Além disso, o nó 1 tem que estar preparado para saber que a combinação de $m1$ e $m2$ tem que acontecer, ou seja, é preciso que exista um mecanismo que garanta a correta recepção de todos os pacotes em tempo da interpretação da mensagem codificada.

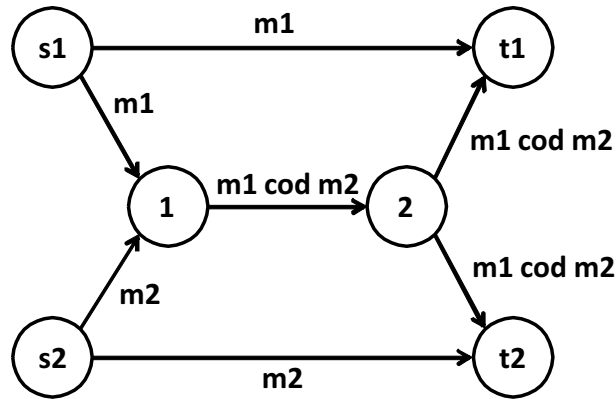


Figura 10 – Exemplo de escuta e codificação oportunista para a implementação de *network coding*

3.2 XOR

Existem várias técnicas de *network coding* que podem ser utilizados para promover a combinação entre pacotes (OSTOVARI; WU; KHREISHAH, 2014). Neste trabalho escolhemos uma técnica específica de *network coding* que pudesse nos proporcionar as vantagens esperadas sem um grande acréscimo da necessidade de capacidade computacional de dispositivos. A utilização de uma disjunção exclusiva (XOR) como técnica de *network coding* oferece um melhor desempenho computacional, com baixa utilização de memória e processamento, tanto para codificação, quanto para a decodificação (KATTI et al., 2006).

Generalizando a utilização da operação de XOR em *network coding*: para uma sequência de m mensagens com n bits cada, a combinação linear é gerada a partir da disjunção exclusiva bit a bit de todas as mensagens. Essa operação pode ser vista na figura 11.

$$\text{XOR} \left\{ \begin{matrix} a_1 & a_2 & a_3 & \dots & a_n \\ b_1 & b_2 & b_3 & \dots & b_n \end{matrix} \right. = \begin{matrix} (a_1 \text{ XOR } b_1) & \dots & (a_n \text{ XOR } b_n) \\ \vdots & & \vdots \\ z_1 & z_2 & z_3 & \dots & z_n \end{matrix}$$

$$\underline{(a_1 \text{ XOR } b_1 \text{ XOR } \dots \text{ XOR } z_1) \dots (a_n \text{ XOR } b_n \text{ XOR } \dots \text{ XOR } z_n)}$$

Figura 11 – Processo de codificação XOR bit a bit de uma mensagem

A vantagem de utilizar a operação XOR está em compor um *payload* codificado de tamanho n , o mesmo dos pacotes que compõe a evidência digital. O único acréscimo em termos de memória que pode acontecer será o tamanho do cabeçalho de controle do novo pacote gerado. Todos os pacotes codificados precisam ter um mesmo tamanho n . Para a

decodificação de uma das m mensagens, é necessário ter $m - 1$ das mensagens que estejam na codificação.

As figuras 12 e 13 apresentam um exemplo da utilização de uma disjunção exclusiva para a codificação e posterior decodificação de uma mensagem de bits.

$$\begin{array}{r} \text{XOR} \left\{ \begin{array}{l} 0100111010 \text{ MSG1} \\ 1010110101 \text{ MSG2} \\ \hline 1110001111 \text{ MSG1 xor MSG2} \end{array} \right. \\ \text{XOR} \left\{ \begin{array}{l} 1110001111 \text{ MSG1 xor MSG2} \\ 1110010101 \text{ MSG3} \\ \hline 000011010 \text{ MSG1 xor MSG2 xor MSG3} \end{array} \right. \end{array}$$

Figura 12 – Exemplo de codificação XOR bit a bit de uma mensagem

$$\begin{array}{r} \text{XOR} \left\{ \begin{array}{l} 000011010 \text{ MSG1 xor MSG2 xor MSG3} \\ 1110010101 \text{ MSG3} \\ \hline 1110001111 \text{ MSG1 xor MSG2} \end{array} \right. \\ \text{XOR} \left\{ \begin{array}{l} 1110001111 \text{ MSG1 xor MSG2} \\ 1010110101 \text{ MSG2} \\ \hline 0100111010 \text{ MSG1} \end{array} \right. \end{array} \quad \begin{array}{r} \text{XOR} \left\{ \begin{array}{l} 000011010 \text{ MSG1 xor MSG2 xor MSG3} \\ 0100111010 \text{ MSG1} \\ \hline 0100100000 \text{ MSG1 xor MSG2} \end{array} \right. \\ \text{XOR} \left\{ \begin{array}{l} 0100100000 \text{ MSG1 xor MSG2} \\ 1010110101 \text{ MSG3} \\ \hline 1010110101 \text{ MSG2} \end{array} \right. \end{array}$$

(a) Decodificando mensagem 1.

(b) Decodificando mensagem 2.

$$\begin{array}{r} \text{XOR} \left\{ \begin{array}{l} 000011010 \text{ MSG1 xor MSG2 xor MSG3} \\ 0100111010 \text{ MSG1} \\ \hline 0100100000 \text{ MSG1 xor MSG2} \end{array} \right. \\ \text{XOR} \left\{ \begin{array}{l} 0100100000 \text{ MSG1 xor MSG2} \\ 1010110101 \text{ MSG2} \\ \hline 1110010101 \text{ MSG3} \end{array} \right. \end{array}$$

(c) Decodificando mensagem 3.

Figura 13 – Processo de decodificação XOR bit a bit de uma mensagem.

Na codificação, MSG1, MSG2 e MSG3 ($m = 3$) são mensagens de um mesmo tamanho $n = 10$ bits a serem codificadas por um usuário u . Uma quarta mensagem de 10 bits, que representa a codificação dessas três, é gerada por u . No exemplo, podemos ver a decodificação de cada uma das mensagens, sendo que duas delas são previamente conhecidas por u . É possível perceber também que a ordem de execução das operações de XOR nos processos, tanto de codificação quanto de decodificação, não influencia no correto processamento das mensagens.

Com as vantagens computacionais e de utilização dentro da topologia que o cenário proposto apresenta, entendemos que realizar *network coding* através de XOR é uma opção plausível, principalmente se levarmos em consideração a necessidade de entregar melhorias em dispositivos PLC que demandem o menor aumento possível de recursos de *hardware*.

4 TRABALHOS RELACIONADOS

A principal ideia por trás de *network coding* é deixar que nós em um enlace de comunicação fim-a-fim combinem pacotes através de algum tipo de operação algébrica com propósito de aumentar vazão e resiliência de uma rede. Assim, há uma quebra no esquema tradicional de encaminhamento de mensagens na rede (*store-and-forward*) (YEUNG et al., 2006). Por exemplo, em um cenário simples com um *relay* intermediando a comunicação entre dois nós, o *network coding* possibilita a transmissão de dados entre esses dois nós simultaneamente. Em outras palavras, o sistema virtualmente transmite as duas mensagens em um único pacote. Em um cenário com escuta oportunista, o nó central pode combinar pacotes que trafegam em direções opostas e, conseqüentemente, são gerados ganhos de codificação de até 1,33 (KATTI et al., 2006).

Os ganhos em vazão e resiliência na rede, a partir do uso de *network coding*, são bem conhecidos para cenários de *broadcast* em redes sem fio. Fragouli, Boudec e Widmer (2006) dissertaram sobre esses ganhos teóricos quando aplicamos *network coding* em uma rede de infraestrutura estática. Nesse contexto, há ainda trabalhos que avaliam as condições de contorno do uso de *network coding* e apresentam os ganhos máximos esperados (KESHAVARZ-HADDAD; RIEDI, 2014).

Existem poucos trabalhos na literatura que propõem o uso desse tipo de técnica em um sistema com base em PLC (EZZINE et al., 2014; NOORI; LAMPE, 2013; PHULPIN; BARROS; LUCANI, 2011; PRIOR et al., 2014; TSOKALO; LEHNERT; FITZEK, 2016). Mesmo que um ambiente sem fio tenha várias características em comum com PLC, os ganhos em um ambiente não podem ser inferidos automaticamente ao outro, e assim, ainda há uma falta de conhecimento sobre o comportamento de *network coding* em PLC na prática (PARAG; CHAMBERLAND, 2010).

Os trabalhos existentes tentam verificar o efeito de *network coding* em PLC, na sua maior parte, em condições ideais ou/e teóricas. Por exemplo, Ezzine et al. (2014) focam seu trabalho na proposta de um algoritmo de alocação de subportadoras em um sistema PLC com acesso ao meio baseado em OFDMA (Orthogonal Frequency Division Multiplexing Access). Eles avaliam o ganho de vazão na rede, quando é utilizado um mecanismo de *network coding* simples entre dois nós e um *relay*. Os autores também avaliam o sistema quando os nós estão em diferentes posições em relação ao *relay*. Apesar dos autores terem utilizado exemplos com *network coding*, esse não foi o objetivo do artigo. As avaliações elaboradas são baseadas em ganhos assintóticos, sem a consideração de um protocolo apropriado e erros realistas que podem influenciar no desempenho do sistema. A conclusão do trabalho demonstra que, em um cenário com três nós e dois enlaces, o melhor desempenho em termos de taxa de transmissão de dados se dá quando o nó intermediário se encontra em posição central e equidistante dos nós periféricos.

Existem análises sobre utilização de *network coding* na camada física de um cenário PLC de banda estreita. Em Noori et al. (NOORI; LAMPE, 2013), os autores utilizaram um simulador em MATLAB para demonstrar ganhos em vazão de dados quando o *network coding* é aplicado a nível de camada física de um sistema PLC doméstico de quatro usuários e um *relay*. Entretanto, o modelo utilizado para a simulação não leva em conta serviços relativos ao controle de acesso ao canal PLC (enquadramento de pacotes, entrega confiável, controle de fluxo e detecção e correção de erros) ou protocolo específico de comunicação para a subcamada MAC.

Phulpin, Barros e Lucani (2011), Prior et al. (2014) também dissertam acerca de prováveis benefícios de *network coding* a um sistema PLC. Os autores propõem o uso de um esquema de *linear code* (DOUGHERTY; FREILING; ZEGER, 2005) em um ambiente de *smart grids*, onde tanto PLC quanto redes sem fio podem ser utilizados. O principal objetivo é o aumento da confiabilidade e vazão da comunicação de dados para redes elétricas inteligentes (REI). Entretanto, os autores também não definem um ambiente e um protocolo para o uso de *coding*. Mais ainda, não há limitações realistas no cenário descrito, como taxas de erro nos canais ou limitações no tamanho dos *buffers*.

Na linha do trabalho que apresentamos, Bilbao et al. (2014) propõem um protocolo também em camada de acesso ao meio PLC que utiliza *network coding*. Eles dedicam parte do trabalho à caracterização de erro em PLC de banda estreita e mostram resultados que indicam que *network coding* pode aumentar a vazão da rede em até (aproximadamente) três vezes, se comparado a um protocolo tradicional, tal como *stop-and-wait*. Apesar das semelhanças, destacamos que o ambiente experimentado é diferente do proposto nesse trabalho. Nesse trabalho, os autores apresentam um limite superior para os ganhos de *network coding* em PLC. Mais ainda, os autores não avaliam questões importantes, como a ocupação de *buffers* dos dispositivos da rede e a latência total para entrega de dados, bem como condições em que o uso de *network coding*.

Mais recentemente, Tsokalo, Lehnert e Fitzek (2016) propuseram a utilização de *network coding* na camada de enlace definida pelo padrão G.9961 (Telecommunication Standardization Sector of ITU, 2015), que define um sistema PLC banda-larga OFDM. Uma combinação entre os mecanismos de ARQ e uma técnica de *network coding* aplicada a uma simulação (utilizando a biblioteca em C++ Kodo) apresenta um ganho de até 15% em vazão e latência. Ainda assim, o trabalho não fez uso de dados reais para a geração de erro no enlace de comunicação e nem se utiliza do controle de acesso ao meio como forma de maximizar os ganhos relativos à utilização de *network coding*.

5 O PROTOCOLO CODEPLC

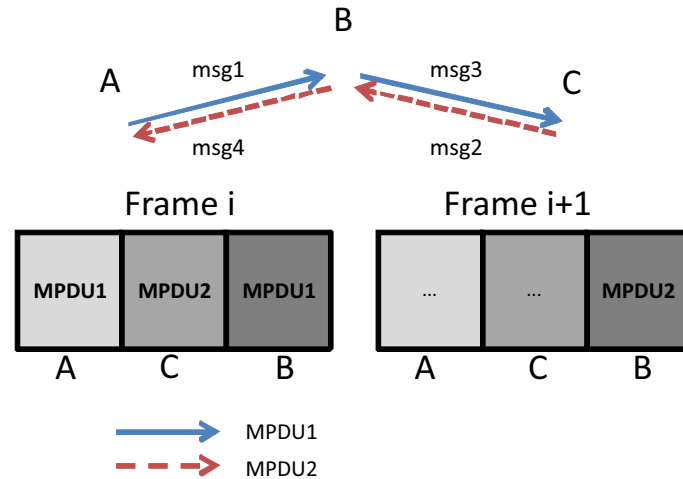
Para melhorar o desempenho da topologia apresentada na figura 4, nosso trabalho propõe a utilização de *network coding* XOR, descrita no capítulo 3, implementando-a junto a uma subcamada de controle de acesso ao meio de um sistema PLC banda-larga com esquema TDMA-OFDM.

Em um contexto de sistemas PLC utilizando tal esquema, uma das principais vantagens da escolha por implementar *network coding* na camada de enlace é podermos utilizar um escalonamento de *time slots* TDMA que favoreça a codificação de um ou mais *MAC Protocol Data Unit* (MPDU). Uma das formas de fazê-lo, é estabelecer o *time slot* final de um *frame* TDMA-OFDM ao nó que executará a combinação (XOR, no caso desse trabalho). Como o tamanho dos *time slots* de tempo da divisão de canal utilizada em um esquema TDMA-OFDM é fixo, a abordagem de codificação utilizando XOR se encaixa muito bem nesse contexto. Se aos MPDU's codificados tivessem tamanhos variados, o uso desse tipo de multiplexação de canal poderia não ser tão interessante do ponto de vista de complexidade do protocolo a ser implementado. Um exemplo de como *network coding* pode propiciar a economia de mensagens em um esquema *multihop* é dado na figura 14.

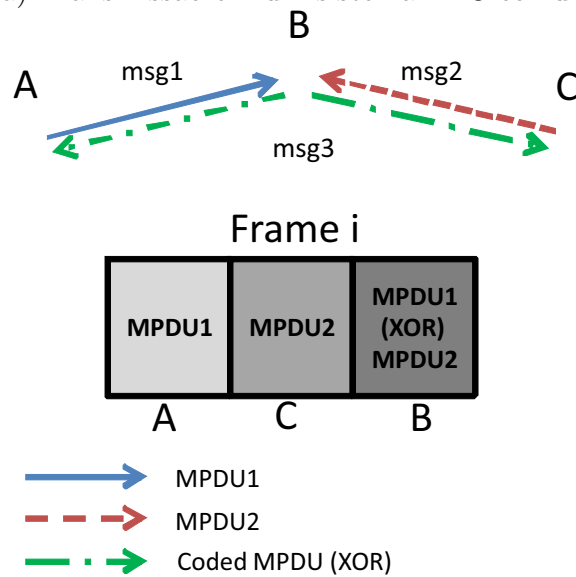
No exemplo, é possível observar que os MPDU1 e MPDU2 em (a) da figura 14 serão sempre enviados em dois saltos cada, representados por *msg1*, *msg2*, *msg3* e *msg4*. Com isso, se faz necessário que o nó B gaste um *time slot* a mais para terminar de transmitir corretamente os dois MPDU's. Em (b) é possível ver que MPDU1 e MPDU2 são codificados em uma única mensagem (*msg3*), que pode ser enviada ainda no mesmo *frame*, dentro do *time slot* designado ao nó codificador. Com isso, em um ambiente em que sejam executadas transmissões com fluxos de dados de origem em cada um dos nós, atravessando o nó central e com destino o nó oposto, é possível que uma troca de mensagens sejam executadas em apenas um *frame* TDMA-OFDM, além de proporcionar a economia de uma mensagem para concretizar a comunicação. Dentro da topologia definida no problema, que é mais complexa e intrincada do que o exemplo da figura 14, esperamos que a mesma situação ocorra e que os ganhos em termos de economia de mensagem se traduzam em um melhor desempenho geral do sistema.

Para a implementação da proposta deste trabalho, qualquer um dos nós da topologia poderia ser escolhido para executar o papel de nó codificador. Ele seria o responsável por armazenar em um *buffer* todo MPDU recebido, executar uma disjunção exclusiva em todos eles para gerar o *payload* de um MPDU novo que será a combinação de todos outros. O nó codificador gera uma mensagem múltipla para todos os usuários vizinhos, que tem informação suficiente para decodificar o MPDU codificado.

Contudo, espera-se obter melhor desempenho se o nó escolhido for posicionado de maneira central, tal qual o nó E da topologia apresentada na figura 4. O nó E é



(a) Transmissão em um sistema PLC comum.



(b) Transmissão utilizando network coding XOR.

Figura 14 – Exemplo de utilização de network coding em TDMA.

beneficiado por sua posição relativa aos outros nós da topologia por ter acesso físico a 1-hop de distância de qualquer outro nó. Com isso, torna-se também interessante a análise da porcentagem de ocupação que esse nó deve ter para ser um executor competente da função de codificar mensagens dos nós vizinhos. Na seção 6.5 veremos essa análise.

5.1 O PROTOCOLO DE COMUNICAÇÃO CODEPLC

Um dos maiores desafios para a implementação de *network coding* proposta neste trabalho é definir um protocolo de comunicação consistente, que entregue os serviços que devem ser fornecidos por uma subcamada MAC. Dentre esses serviços estão: controle de erros, transmissão e recepção correta de quadros, controle do fluxo de dados e controle do meio de transmissão.

Nesta seção, veremos o projeto de protocolo do CodePLC e as técnicas para prover esses serviços da subcamada MAC em combinação com a implementação de um algoritmo de *network coding* em uma rede PLC TDMA-OFDM.

5.1.1 Controle do meio de transmissão

Como vimos na seção 2.1, o sistema PLC que utiliza um esquema TDMA-OFDM tem em seu *frame* uma subdivisão de tempo de troca de controle e é finalizado por um período de *downlink* e *uplink* de dados de cada usuário que utiliza o canal. Neste trabalho, inferimos a perfeita sincronização dos nós do enlace. A subcamada MAC dos nós reconhece a temporização do canal e sempre estará disponível para receber o transmitir de acordo com ela.

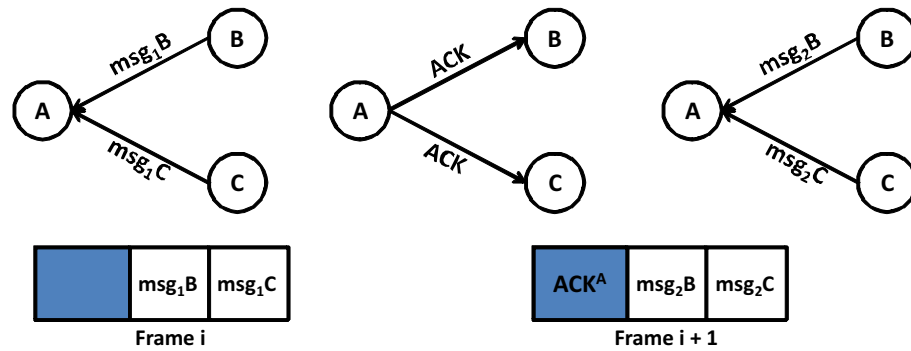
Durante o *time slot* cada nó executa *uplink* de um MPDU, se existir, do *buffer* de transmissão. Durante esse período, apenas um dos nós transmite. Durante os *time slots* de outros usuários, cada nó recebe apenas um MPDU de cada um dos vizinhos e os armazena em um *buffer* de recepção. O armazenamento desses dados é importante para a correta decodificação dos MPDU's em que foi utilizado o *network coding*.

5.1.2 Gerenciamento de erros e retransmissão

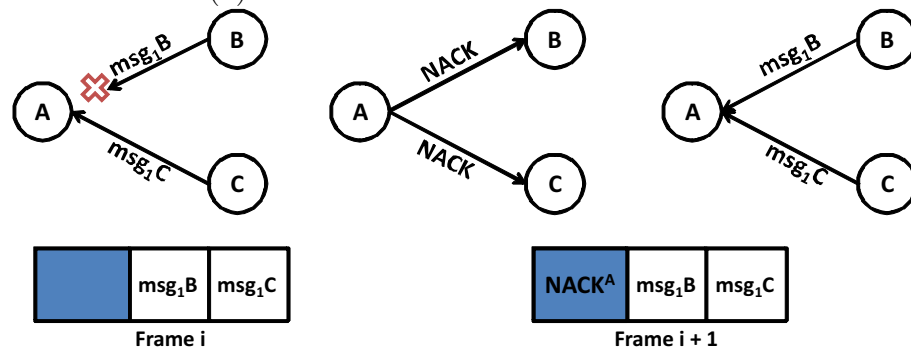
Para o controle de erros e retransmissão, CodePLC prevê um protocolo de *Automatic Repeat reQuest* (ARQ) do tipo *stop-and-wait* com mensagens de ACK e NACK, combinado a um campo próprio de conferência de integridade de dados através do uso de um método de verificação de redundância cíclica (CRC). De acordo com as probabilidades de erro de bit obtidas por medição (como abordado no capítulo 2), MPDU's são enviados pelo transmissor e, se houver integridade comprometida, identificados como tal no receptor, através de erro no campo de CRC. Por simplificação, consideramos que essa detecção de integridade sempre será eficiente.

A partir da recepção um MPDU de integridade comprometida ou não, recebido de um usuário u , cria-se no receptor uma mensagem de ACK ou NACK que acompanha a informações referentes à u . Essa mensagem será enviada em um *broadcast* ou *multicast* dentro do tempo de canal estabelecido, o *time slot* de controle. Um usuário u que receba alguma mensagem de NACK referente à transmissão de dados do *frame* anterior, retransmite o mesmo MPDU no *frame* atual. O processo pode ser ilustrado pela figura 15.

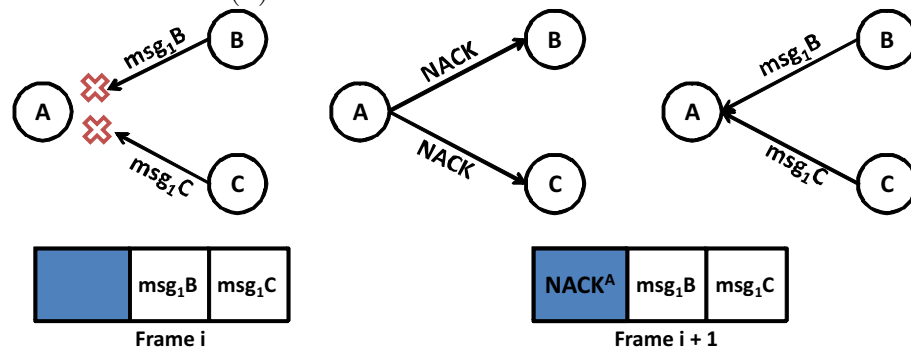
Na implementação simulada, não utilizamos limite superior para o número de tentativas de envio de um MPDU. Portanto, podemos considerar que o serviço da camada de enlace é confiável ao utilizarmos CodePLC. Isso terá impacto na latência medida mas, como veremos no capítulo 6, esse impacto ainda é menor do que o causado pelo congestionamento na utilização do *multihop*;



(a) Transmissão sem erros no MPDU.



(b) Erro de MPDU de um usuário.



(b) Erro de MPDU de dois usuários.

Figura 15 – Exemplo do protocolo de ARQ do CodePLC.

5.1.3 Codificação de MPDU's

O protocolo CodePLC prevê um acréscimo de informação de controle no MPDU PLC. Essa adição pode ser vista na figura 16.

O campo de “Bitmap Coding” apresenta um conjunto de bits onde cada um deles, a partir do menos significativo, representa um usuário. A ordem da relação bit-usuário é dada pela ordem dos *time slots* de cada um dos usuários. Quando o bit estiver ativo em um MPDU codificado, o CodePLC entende que o MPDU daquele usuário está presente na codificação. Com essa informação, também é possível saber quantas MPDU's foram codificadas na mensagem.

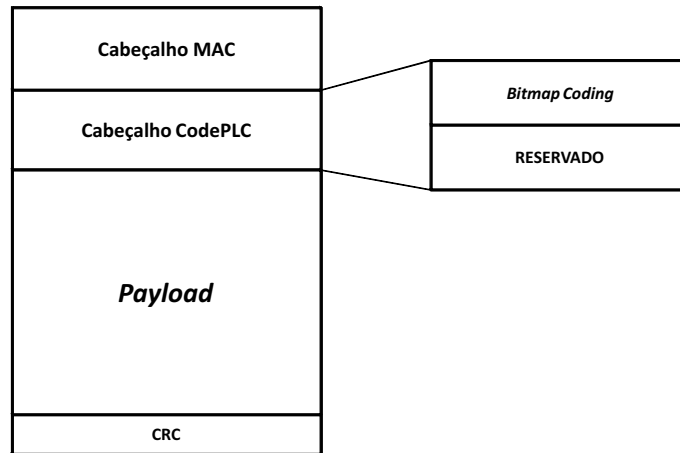


Figura 16 – Estrutura de um MPDU em uma subcamada MAC com CodePLC

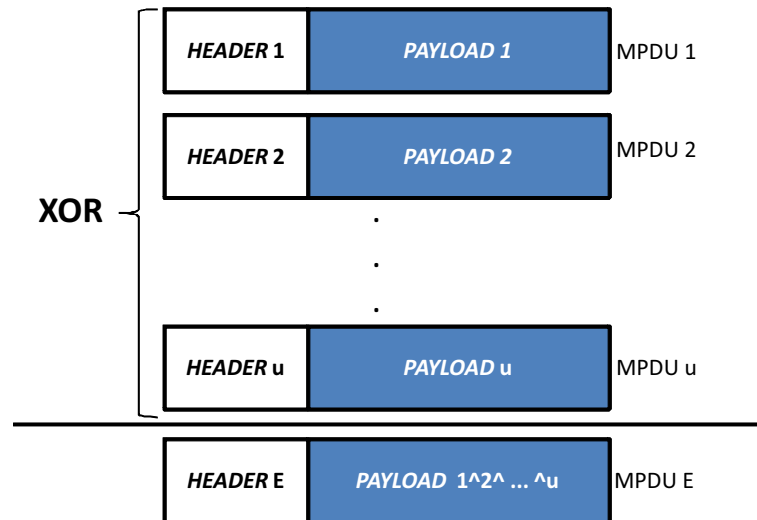


Figura 17 – Exemplo de codificação de MPDU's no nó E

5.2 ALGORITMO CODEPLC

Os algoritmos que controlam os processos de transmissão e recepção dos nós do sistema simulado são descritos nos fluxogramas das figuras 18 e 19. A figura 18 descreve os procedimentos de transmissão, recepção e decodificação de MPDU's de um nó U em seu respectivo *time slot*. Quando um erro de MPDU ocorre, o algoritmo compreende também o processo de retransmissão, que está inserido no *time slot* de controle.

Na figura 19, o fluxograma descreve os procedimentos de recepção, transmissão e codificação de MPDU's pelo nó responsável por executar o *network coding* PLC.

Mais precisamente, dentro do algoritmo proposto para o *network coding* PLC, no *time slot* correspondente, o nó atual U executará os seguintes procedimentos:

- Análise de MPDU de controle do último *time slot*: em um *frame* i , no *time slot* de controle, um nó U recebe um mensagem de ACK/NACK de seus respectivos vizinhos.

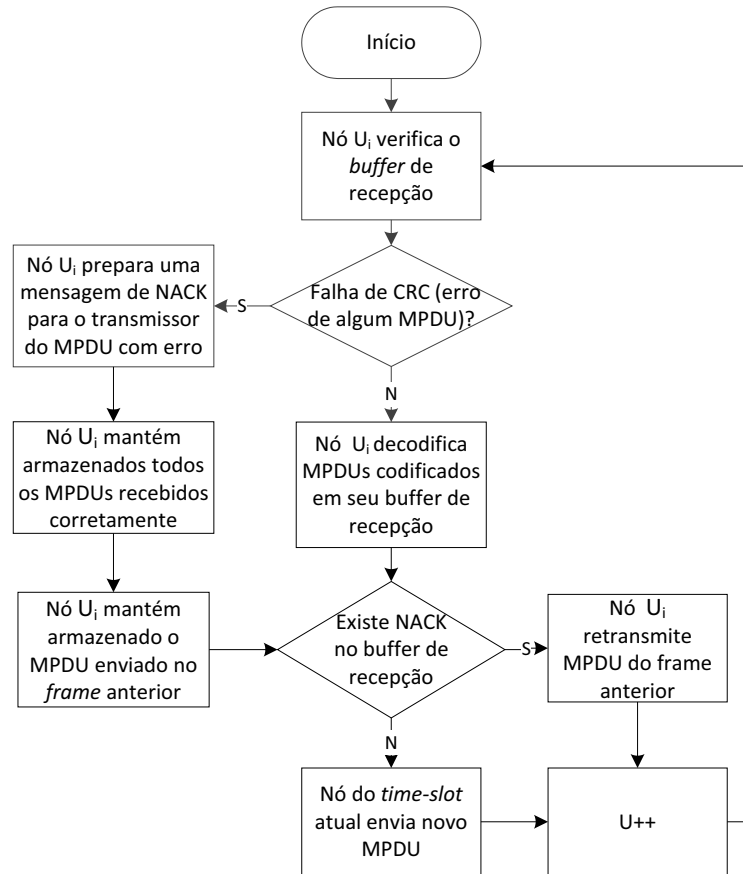


Figura 18 – Fluxograma do algoritmo de transmissão e recepção em um nó comum

Se ACK, a mensagem no *frame* $i - 1$ foi corretamente enviada. Se NACK, a mensagem do *frame* $i - 1$ é preparada para ser retransmitida no *frame* i , dentro do *time slot* relativo ao nó U.

- Recepção de MPDU de dados: em cada *time slot* t_u de um *frame* i , um nó U recebe o MPDU dos seus vizinhos. Se o MPDU foi corretamente recebido, ele é armazenado em um buffer de recepção e uma mensagem de ACK é preparada para ser enviada a todos os vizinhos, no *time slot* de controle do *frame* $i + 1$. Se a mensagem tem erro, um NACK é enviado para todos os vizinhos e os MPDU's recebidos no *frame* i são descartados.
- Recepção de MPDU codificado: em um *frame* i , no último *time slot*, todo nó U recebe uma mensagem codificada do nó responsável pelo *network coding*, contendo as mensagens corretamente recebidas e ele codificadas dentro de um MSDU codificado. Em casos de erro no envio do MPDU codificado, no *time slot* de controle do *frame* $i + 1$, o nó U envia mensagem de NACK.
- Decodificação de MPDU codificado, caso seja possível: para a decodificação de um MPDU codificado em U, é necessário que as mensagens dos vizinhos que chegaram sem erro ao nó de codificação, também sejam corretamente recebidas por ele. Se essas mensagens estiverem no *buffer* de recepção de um nó U, esse nó consegue extrair o

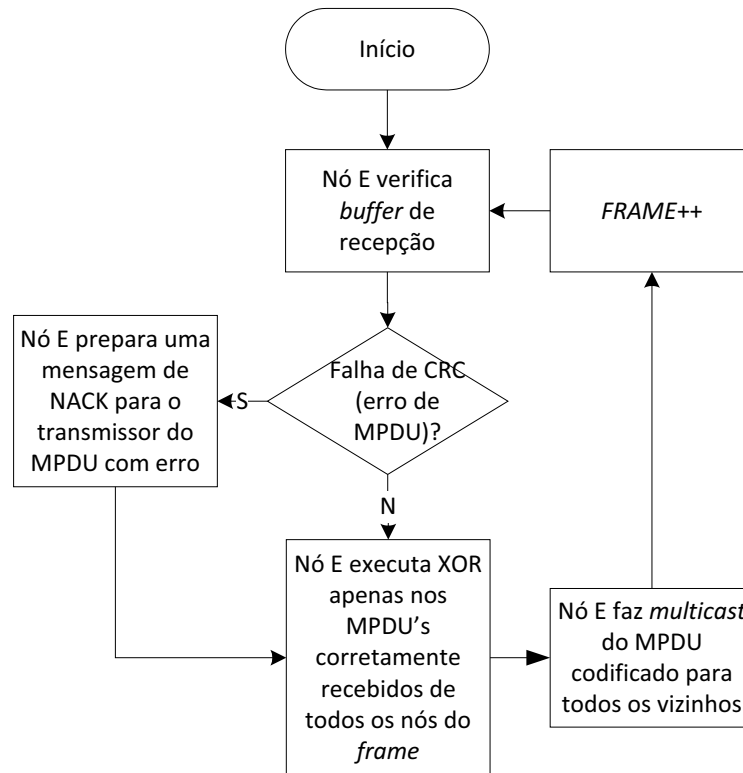


Figura 19 – Fluxograma do algoritmo de transmissão e recepção em um nó codificador

MPDU que é destinado a ele. Caso os dados de decodificação não estiverem presentes no *frame* atual, o MPDU codificado é armazenado em *buffer* até que todas as mensagens do *frame* anterior sejam retransmitidas.

- Codificação com disjunção exclusiva: em todo *frame* i , o último *time slot* t é designado ao usuário E , que executa a codificação de todos os MPDU's corretamente transmitidos pelos demais usuários no *frame* e transmite em múltiplos MPDU's para todos usuários.
- Transmissão de MPDU gerado pela camada superior: quando um dado é gerado na camada superior de um usuário U , um MPDU é criado e armazenado na fila de transmissão. Ao ser transmitido, em um *frame* i , o MPDU fica armazenado nesse *buffer* até a recepção do ACK ou NACK no *time slot* de controle do *frame* $i + 1$.

5.3 LIMITAÇÕES DO CODEPLC

As limitações mais evidentes do protocolo proposto neste trabalho estão ligadas ao cenário analisado. A utilização de *network coding* é limitada pela necessidade de MPDU's serem recebidos para decodificar uma mensagem. Para um número m de MPDU's codificados dentro de uma mensagem, precisamos de $m - 1$ dos MPDU's contidos nessa mensagem, para a correta decodificação. Em um cenário onde existe indisponibilidade de uma comunicação *broadcast* entre os nós que geram MPDU's incluídos em uma mensagem

codificada, o CodePLC não é aplicável. O protocolo também não prevê a utilização de mais de um nó intermediário como operador do *network coding*.

A implementação do CodePLC em um nó intermediário de um sistema que utiliza um esquema TDMA-OFDM, pode torná-lo indisponível para *uplink* de mensagens próprias, sejam elas geradas no próprio nó ou em uma subrede conectada a ele. Essa indisponibilidade poderia ser resolvida em um sistema que utilizasse um esquema OFDMA-TDMA, com outra faixa de frequência reservada para a transmissão de dados de forma concorrente, ou com a intercalação entre *frames* em que o protocolo CodePLC é utilizado e *frames* indisponíveis para a utilização do mesmo. Nesse sentido, a análise proposta na seção 6.5 é importante para determinar o desempenho resultante dessa abordagem.

O protocolo também não prevê a segmentação de MPDU's e montagem de pacotes, caso o dado a ser transmitido exceda um tamanho máximo em bytes. Para esse serviço, seria necessário um protocolo ARQ mais complexo, que levasse em conta perdas de segmentação, no nó de destino e necessidades de retransmissões seletivas, no nó de origem.

6 METODOLOGIA E AVALIAÇÃO

6.1 METODOLOGIA

Para analisar o desempenho do algoritmo proposto no sistema analisado, criamos, na ferramenta MATLAB, um simulador da topologia definida na figura 4. Utilizamos um simulador próprio por tratarmos apenas de camada de enlace em nosso trabalho. Consideramos a camada física como “caixa-preta” e os únicos dados que temos vindo dela são as taxas de erro de bit, que é relevante por se tratar de dados realistas, coletados em uma campanha de medição. Se utilizássemos um simulador de terceiros, precisaríamos definir e configurar parâmetros na camada física que fogem do nosso escopo de trabalho. Para a análise do protocolo proposto, foram testados três cenários:

- Rede *multihop* com todos os cinco nós iguais. A camada de enlace é responsável por transportar MPDU's utilizando um nó intermediário aleatório entre transmissor e destino, fazendo com que essas MPDU's alcancem usuários que não são acessíveis a 1-hop pelo nó de origem da transmissão. Todos nós transmitem para todos outros em *multicast*. Assume-se que, para todo nó (exceto o central), há uma taxa de geração de uma MPDU de tamanho fixo a ser transmitido em cada *frame* TDMA-OFDM (ou seja, um nó transmite com uma dada probabilidade). Assume-se também que a decisão de qual nó será escolhido como caminho para um MPDU em *multihop* será aleatória.
- Rede *multihop* com 4 nós iguais e o nó E (central) responsável pelo *network coding*. O nó central está sempre disponível para receber MPDU's de seus vizinhos, codificá-los e colocá-los em uma fila de transmissão. Assume-se que, para todo nó (exceto E), há uma taxa de geração de uma MPDU de tamanho fixo a ser transmitido em cada *frame* TDMA-OFDM.
- Rede *multihop* com 4 nós iguais e o nó E (central) responsável pelo *network coding*. O nó central tem uma probabilidade p de estar disponível para receber MPDU's de seus vizinhos, codificá-los e colocá-los em uma fila de transmissão. Durante os períodos de indisponibilidade, a recepção e transmissão em todo o sistema estão inativos. Assume-se que, para todo nó (exceto E), há uma taxa de geração de uma MPDU de tamanho fixo a ser transmitido em cada *frame* TDMA-OFDM.
- Rede *multihop* com 3 nós iguais, o nó E (central) responsável pelo *network coding*. O nó central está sempre disponível para receber MPDU's de seus vizinhos, codificá-los e colocá-los em uma fila de transmissão. Assume-se que, para todo nó (exceto E), há uma taxa de geração de uma MPDU de tamanho fixo a ser transmitido em cada *frame* TDMA-OFDM.

Nesses cenários, dois parâmetros foram variados: o tamanho das MPDU's (40, 80, 120, 160, 200 e 248 Bytes de *payload*) e a taxa de novos MPDU's gerados pela camada superior e adicionados à fila de transmissão ($t, 0 < t < 1$). Essas variações são interessantes pois impactam na taxa de erro de pacotes (tamanho da MPDU) e o tráfego de dados no sistema simulado (taxa de transmissão). As probabilidades de transmissão e o tamanho do MPDU são métricas que apresentam um bom compromisso para demonstrar o comportamento da rede em baixa, média e alta carga de trabalho.

Conforme definimos na seção 3, consideramos também um sistema *uncoded* TDMA-OFDM, que aloca todas as subportadoras para um nó quando o mesmo está usando seu próprio *time slot* para transmissão. Adotamos a modulação digital BPSK e uma potência $P = 30$ dBm. A frequência de banda utilizada é de 1,705 até 50 MHz. O tráfego na rede simulada atravessam o nó central em direções opostas (ou seja, A e B enviam pacotes destinados a C e D, e vice-versa).

Em cada cenário avaliado, e em cada uma de suas variações, foram executados os mesmos experimentos, a fim de apurar as seguintes métricas: ocupação média dos *buffers*, *goodput*, latência média e jitter. A ocupação média de *buffers* é definida pelo total de MPDU's válidas nos *buffers* de todos usuários do sistema em um determinado intervalo de tempo. O *goodput* é definido pela razão entre o número de transmissões efetuadas com destino a um nó u e o número de MPDU's corretamente recebidos por esse nó. A latência média foi dada pela média do número de *frames* TDMA-OFDM que um MPDU levou para ser entregue corretamente em seu destino, em um determinado intervalo de tempo. O jitter é a variação estatística da latência.

Para medições de ocupação de *buffers*, foram realizadas 1.000 iterações, onde cada uma corresponde a 200 *frames* TDMA-OFDM. Para medições de *goodput*, foram realizadas 1.000 iterações, onde cada uma corresponde à chegada correta de 100 MPDU's a um nó especificado. No caso da medição de latência média, foram realizadas 1.000 iterações, onde cada uma compreende os envios realizados durante a execução de 200 *frames* TDMA-OFDM.

Os dados que representam as taxas de erro utilizadas tem como base valores de PER gerados da forma descrita no capítulo 2 e representam os erros relativos a cada um dos enlace. Os valores PER, calculados com base na equação $PER_i = 1 - (1 - BER_i)^{N_i}$, onde N_i corresponde ao tamanho do pacote, são acessados de forma aleatória para cada evento de transmissão e determinam uma probabilidade de erro de um MPDU.

Com objetivo de simplificar o cenário em relação às métricas, consideramos que os *time slots* de usuários dentro do *frame* TDMA-OFDM é suficiente para transmitir o maior MPDU na modulação digital utilizada. MPDU's menores são preenchidos com *padding*. Consideramos também que a contagem de preenchimento de *buffer* é realizada de acordo com o número de MPDU's em memória, independente do tamanho. Todos os *buffers*

utilizados têm tamanho infinito.

6.2 OCUPAÇÃO DE BUFFERS

Neste trabalho, *buffer* é uma métrica importante a ser considerada em termos de desempenho quando propomos uma utilização de *network coding*. Apesar da técnica XOR não inserir muita informação de controle ao MPDU, o que acontece em outras técnicas de *network coding* (OSTOVARI; WU; KHREISHAH, 2014), a análise do *buffer* é capaz de revelar problema de congestionamentos típicos de sistemas de encaminhamento de pacotes.

A figura 20 mostra a distribuição acumulada (CDF) da relação entre a ocupação total dos *buffers* em cada *frame* TDMA-OFDM, em um cenário *multihop* PLC comum e um CodePLC. A taxa de geração de MPDU's na camada superior utilizada nesse cenário foi de 0.7, contexto no qual esperamos um desempenho melhor da técnica de *network coding*. Nesse caso, os testes foram executados com MPDU com tamanho fixo de 248 Bytes.

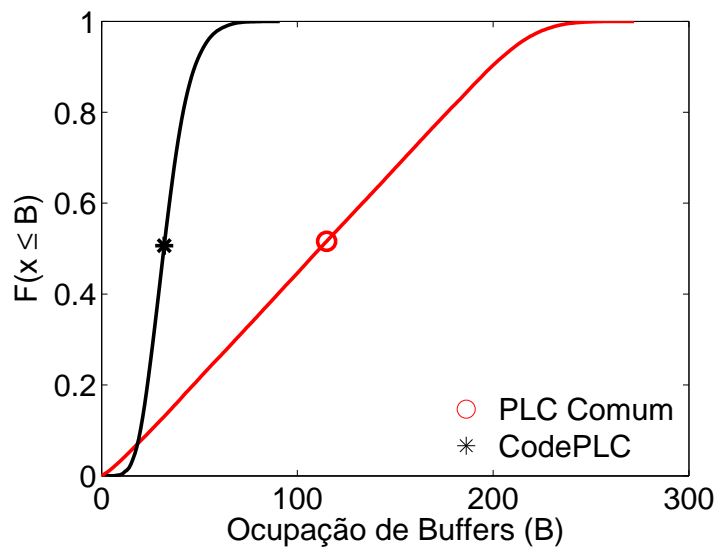


Figura 20 – Ocupação total de *buffers* do sistema ao fim de 200 *frames*.

Note que, há situações que a rede sem *network coding* (cenário comum) pode apresentar uma porcentagem de valores de ocupação de *buffer* menores que um cenário com o protocolo CodePLC. Intuitivamente, quando o sistema tem pouca disputa por recursos, a entrega direta (no caso, por dois saltos) pode gerar menor ocupação dos *buffers* do sistema. Entretanto, para cerca de 90% dos valores de ocupação, o cenário com CodePLC apresentou valores menores de preenchimento de buffer.

Essa melhoria está relacionada ao fato de, ao adotarmos o protocolo proposto, desobrigamos todos os nós do sistema de exercer um papel de intermediário entre a comunicação de dois nós *multihop*. Com isso, evitamos gargalos que teriam que ser tratados

com algoritmos de roteamento mais sofisticados e com maior *overhead* de mensagens de controle. Como o algoritmo utilizado para o *network coding* é o XOR, o aumento de tamanho do MPDU se dá apenas com alguns campos de controle (levados em conta na geração dos resultados).

Outro fator importante que contribui para o menor uso de *buffers* é a abordagem simplista do algoritmo de ARQ utilizado. Quando um nó vizinho não tem os dados necessários para a decodificação de um MPDU codificado, ele descarta os MPDU's do *frame* anterior, conserva apenas mensagem codificada e pede reenvio a todos os vizinhos (menos ao nó de *coding*). É provável que uma abordagem mais sofisticada no ARQ, onde apenas o usuário que falhou na última transmissão repetisse a tentativa, nos entregasse um melhor resultado em termos de *goodput*. Mas, em contrapartida, necessitaria de um gerenciamento complexo das mensagens do buffer e um gasto maior de memória para se acumular MPDU's que foram transmitidos com sucesso mas que, naquele momento, não podem ser utilizados no processo de decodificação.

As figuras 21 e 22 apresentam a ocupação de *buffer* no sistema, ao se variar o tamanho das MPDU's e a taxa de transmissão, respectivamente. O aumento do tamanho da MPDU piora o desempenho de ambos os sistemas. Como a taxa de erros de pacotes é proporcional ao tamanho deles, um aumento na MPDU aumenta a taxa de erro do sistema, o que gera mais retransmissões (e filas nos *buffers* dos dispositivos).

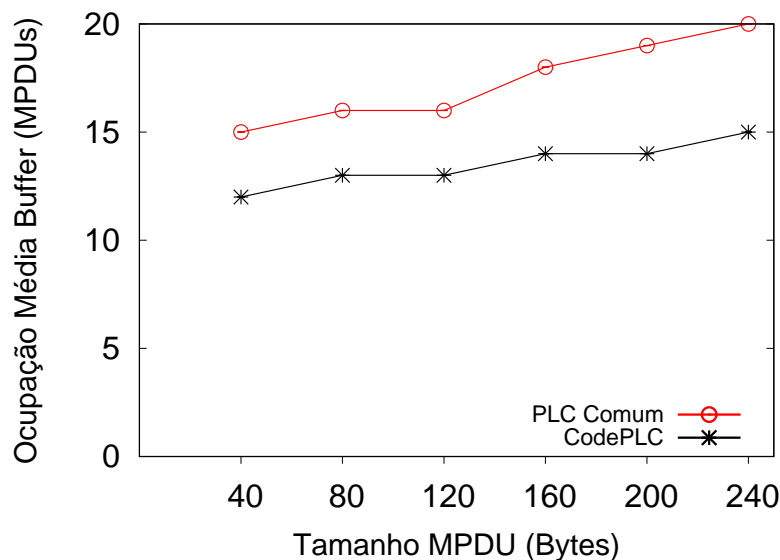


Figura 21 – Média de ocupação de *buffer* no sistema em relação à variação de tamanho de MPDU.

Em redes com taxas de geração de dados abaixo de 0,4, a transmissão direta, sem necessidade de codificação/decodificação apresenta vantagens em relação ao uso de *network coding*. Entretanto, com o aumento da taxa de transmissão (i.e. maior probabilidade de transmissões), a rede se torna mais disputada pelos dispositivos. Nesse cenário, o

sistema com *network coding* sofre menos impacto que o sistema tradicional. A melhoria de desempenho em termos de ocupação máxima de *buffers* chega a 71,4% com 0,6 de taxa de geração de MPDU.

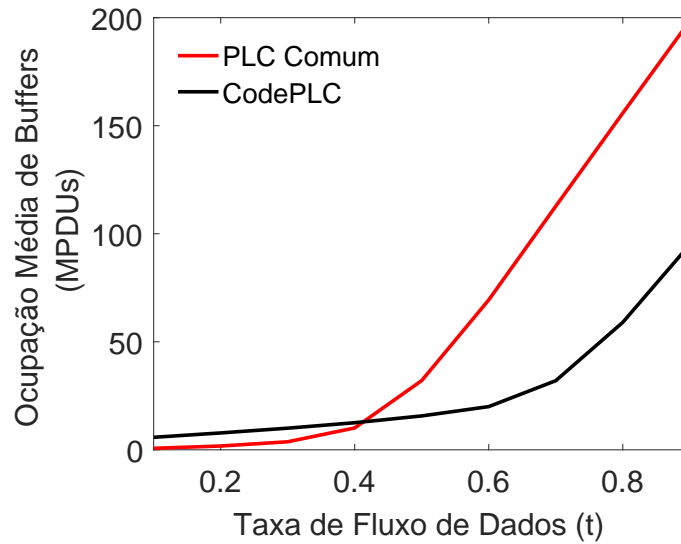


Figura 22 – Média de ocupação de *buffer* no sistema em relação ao volume de tráfego gerado.

6.3 GOODPUT

Goodput é uma métrica de desempenho apurada para o cenário tratado, já que lidamos com um sistema com alto índice de erros e congestionamentos. Saber a taxa na qual mensagens são corretamente entregues ao destino em um determinado intervalo de tempo nos proporciona uma visão sobre como a implementação de *network coding* influencia o sistema PLC estudado.

Na figura 23, é apresentada a função de distribuição acumulada da relação do *goodput* de todos os usuários do sistema a cada iteração de testes. A taxa de geração de MPDU's na camada superior utilizada nesse cenário foi de 0,7, criando uma alta taxa de trabalho. Como esperado, são necessários menos *frames* TDMA-OFDM para realizar a entrega correta dos MPDU's enviados de uma origem. Em consequência, a vazão média do sistema se eleva. Como vimos acima, há redução das filas dos *buffers* e assim, quase não há casos de MPDU's retidos em *buffers* de usuários intermediário. Esses dois fatores contribuem para o ganho observado no gráfico. Em média, a diferença é de cerca de 40% entre os dois cenários.

A figura 24 apresenta o *goodput* médio dos nós sistema, em relação a diferentes tamanhos de MPDU. Para os dados apresentados, o intervalo de confiança, para 99% de confiança, é desprezível (menor que 0,01%). Em ambos os cenários, observamos um leve decaimento na métrica de interesse ao se elevar o tamanho das mensagens. O impacto pelo

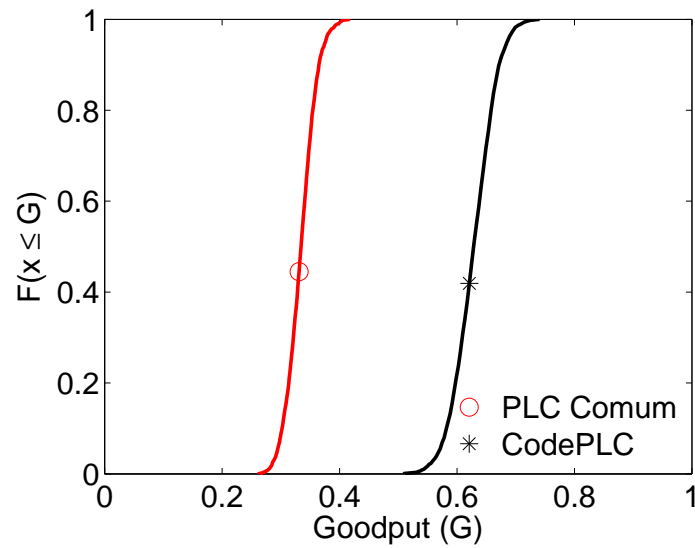


Figura 23 – Distribuição do *goodput* médio dos nós do sistema ao fim de 100 MPDU's corretamente recebidas.

crescimento do tamanho da MPDU pode estar relacionado a um leve aumento na taxa de erro de pacotes que, nesse caso, influencia negativamente ambos os cenários, gerando mais retransmissões.

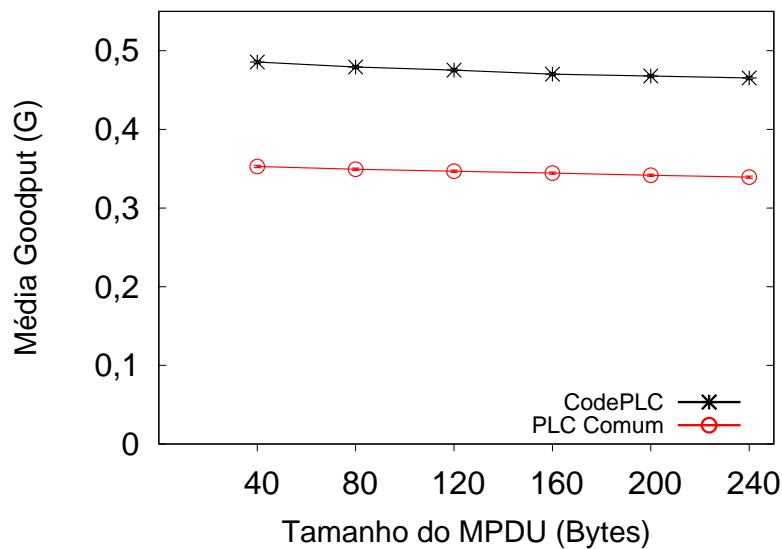


Figura 24 – Média do *goodput* do sistema em relação à variação de tamanho de MPDU.

A variação da taxa de transmissão destaca ainda mais a diferença de desempenho entre os sistemas avaliados. A figura 25 apresenta os valores de mediana de *goodput* ao se variar essa taxa.

Por essa figura, podemos observar que os valores de *goodput* nos cenários com e sem a utilização do CodePLC são similares até uma taxa de geração de dados de 0,4. A partir daí, a geração de pontos de gargalo e congestionamento do cenário *multihop* compromete

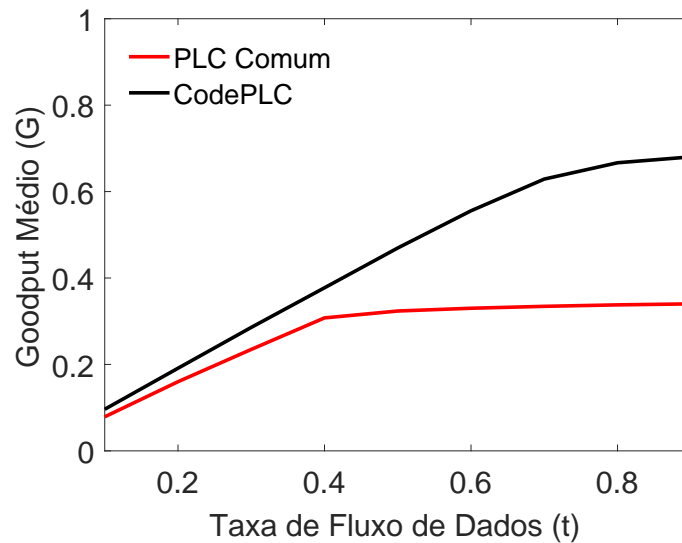


Figura 25 – Média do *goodput* do sistema em relação à variação de tamanho de MPDU.

a entrega de MPDU's corretas a seu destino dentro de um determinado intervalo de tempo observado, estabilizando o *goodput* de sistemas que utilizam *multihop*. Já no sistema CodePLC, o crescimento da taxa de MPDU's beneficia o aumento do *goodput* mediano de até 110% quando chegamos a uma taxa de geração de pacotes de 0,9.

6.4 LATÊNCIA e *JITTER*

Como vimos no Capítulo 2, latência e *jitter* são métricas relevantes por serem parte de um problema reconhecido de sistemas que utilizam *multihop* em suas transmissões. Também temos questões relativas à essas métricas quando tratamos de sistemas TDMA-OFDM já que, mesmo que o cenário estudado represente um sistema com número fixo de usuários, as retransmissões consequentes de erros de pacotes acarretam em uma degradação dessas duas métricas. A latência calculada é a média de tempo de entrega (em *frames*) das MPDU's a cada execução de *frame* TDMA-OFDM simulado. O *jitter* foi calculado a partir da média da variação nas latências em cada um dos *frames* TDMA-OFDM simulados. O valor do *jitter* médio também é dado em número de *frames* TDMA-OFDM.

Na figura 26 podemos observar como o comportamento da métrica de latência seja problemática no cenário *multihop* e a melhoria do parâmetro dada a utilização do CodePLC em um cenário com taxa de geração de dados de 0,7.

É possível atribuímos a diminuição da latência à resolução de um problema de congestionamento causado pela abordagem *multihop*. Ao invés do repasse de mensagens pelos nós intermediários, o CodePLC poupa os usuários periféricos dessa tarefa e a acumula no usuário central. Como a função deste nó central é utilizar *network coding*, o tempo de uma mensagem em um *buffer* de um usuário intermediário é substituído pelo tempo de

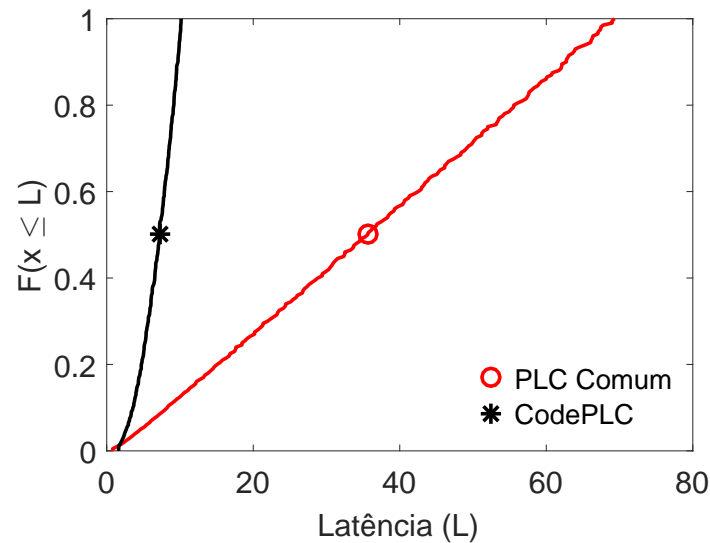


Figura 26 – Distribuição da latência média (em *frames*) em um sistema com geração de MPDU a 0,7.

codificação e envio correto da mensagem codificada aos seus destinos, bem como a correta decodificação da mensagem.

Os valores medianos de latência avaliados na variação de geração de fluxo de dados, são muito similares nos dois casos até o fluxo de dados em 40%. Como podemos observar na figura 27, após esse valor, a latência no cenário *multihop* é consideravelmente maior.

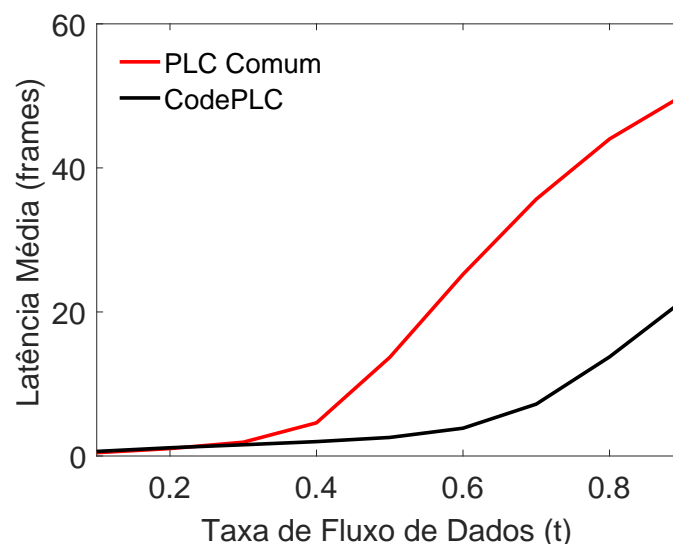


Figura 27 – Média do latência do sistema em relação à variação da geração de dados.

O comportamento do *jitter* com a utilização do protocolo CodePLC é mais estável durante toda a execução da transmissão. Na figura 28, podemos ver valores altos na variação de atraso em um sistema PLC comum. Essa variação se dá pelas questões apontadas na análise de distribuição de valores da ocupação de *buffers*. Durante a carga de

um sistema PLC comum, não há focos de congestionamento, assim as transmissões chegam ao destino sem grandes atrasos. No momento em que o número de MPDU's transmitidos e armazenados em *buffer* dos nós aumenta, os valores de latência também aumentam, causando um impacto na variação do atraso entre MPDU's.

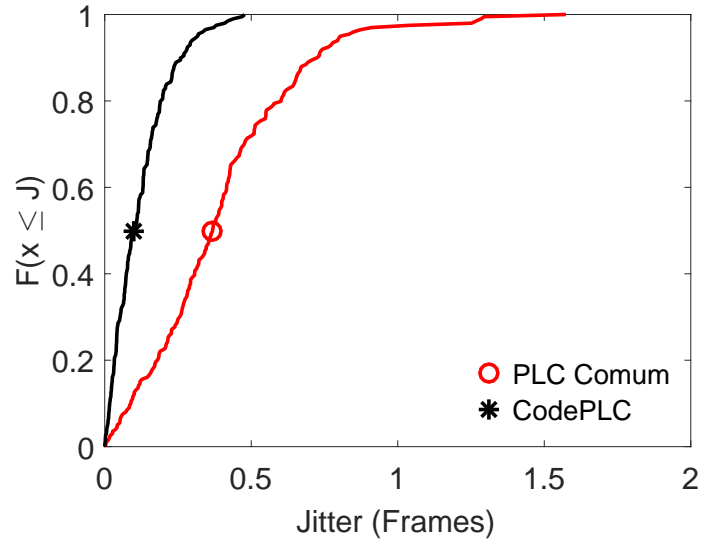


Figura 28 – Distribuição de jitter (em *frames*) em um sistema com geração de MPDU a 0,7.

Com a variação de fluxo, os valores de variação de latência são menores para o caso do CodePLC. Como mostrado na figura 29, para taxas de geração menores que 0,4, temos um *jitter* menor no cenário de PLC, devido à baixa carga de MPDU's nos *buffers* de cada um dos nós. Após esse valor de taxa, temos valores *jitter* até 71% menores para o caso do CodePLC.

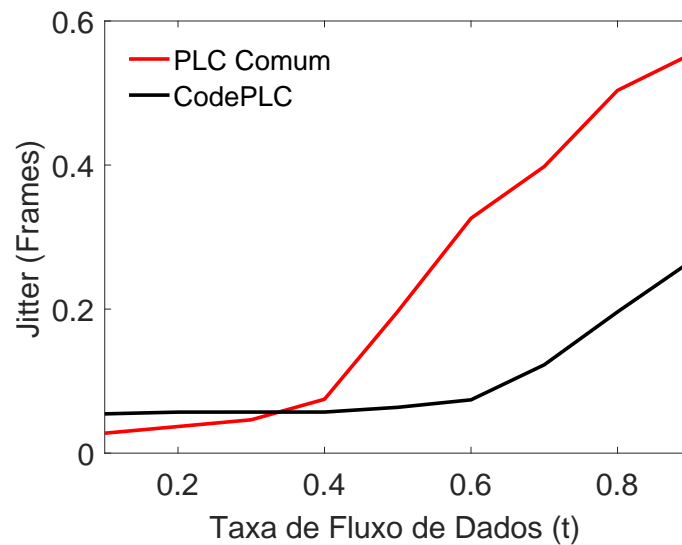


Figura 29 – *Jitter* médio do sistema em relação à variação da geração de dados.

Isso ocorre, entre outros motivos, pela variação da quantidade de MPDU's codificados em cada frame TDMA-OFDM. Quando a taxa é maior e mais MPDU's chegam ao nó codificador, o sucesso dessa codificação e posterior envio aos nós vizinhos está intrinsecamente ligado ao aumento da estabilidade no tempo de entrega.

6.5 VARIAÇÃO NA DISPONIBILIDADE DO NÓ CODIFICADOR

Outra análise realizada por esse trabalho trata-se da disponibilidade de utilização do nó central como codificador de MPDU's. Como variamos o fluxo de dados gerados em cada um dos nós, também é importante definirmos se essa variação na utilização da codificação no nó E insere algum tipo de queda de desempenho, já levando em conta os ganhos esperados pelo uso do protocolo CodePLC.

Essa é uma análise importante por englobar um possível cenário onde o nó E está indisponível para codificar mensagens de seus vizinhos, forçando os enlaces a ficarem inativos. No protocolo CodePLC, qualquer um dos nós poderiam assumir a posição de codificadores. Entretanto, a utilização de *network coding* em MPDU's de outros usuários pode não ser possível devido à erros no dispositivo, no canal e da necessidade do nó intermediário de transmitir os dados gerados por ele mesmo (ou uma subrede associada à ele), entre outros fatores.

Para os testes, executamos dentro do cenário apresentado uma análise sobre limitações e potencialidades da aplicação do protocolo CodePLC em uma situação onde o canal não estará 100% do tempo disponível. Definimos situações onde, tanto *uplink* e *downlink*, tenham uma probabilidade p tal que $0 < p < 1$, de ficarem indisponíveis por um frame TDMA-OFDM inteiro, como ilustrado na figura 30. Durante esse período, todos os *buffers* e estados do sistema serão mantidos e deixados para o próximo *frame* ativo, exceto o recebimento de novos MPDU's gerados em cada nó. A codificação e conseqüente envio de MPDU's codificados também só serão executadas durante o período de atividade. Nosso objetivo é, a partir das métricas, gerar informações suficientes para definir um *tradeoff* entre ganhos gerados pelo protocolo proposto e perdas geradas pela indisponibilidade do canal PLC.

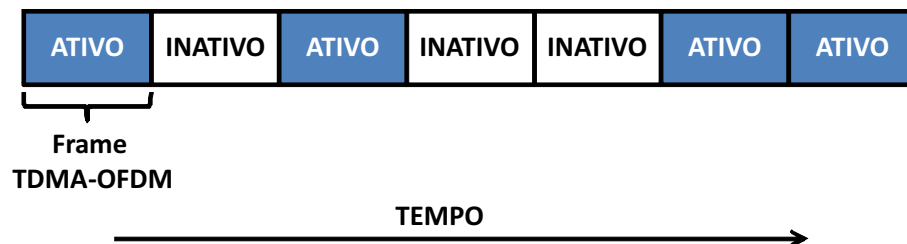


Figura 30 – Esquema de disponibilidade do canal PLC para comunicação de dados

Como sabemos pelos resultados anteriormente apresentados, existe uma tendência

de que o CodePLC proporcione resultados melhores, em comparação ao um cenário comum, quando utilizamos taxas de geração de MPDU's maiores que 0,4. Por isso, as análises serão feitas com uma taxa de geração a partir dos 0,5. Para os dados apresentados, o intervalo de confiança, para 99% de confiança, é desprezível (menor que 0,01%).

A análise inicial que executamos é ilustrada pelas funções de distribuição apresentadas na figura 31.

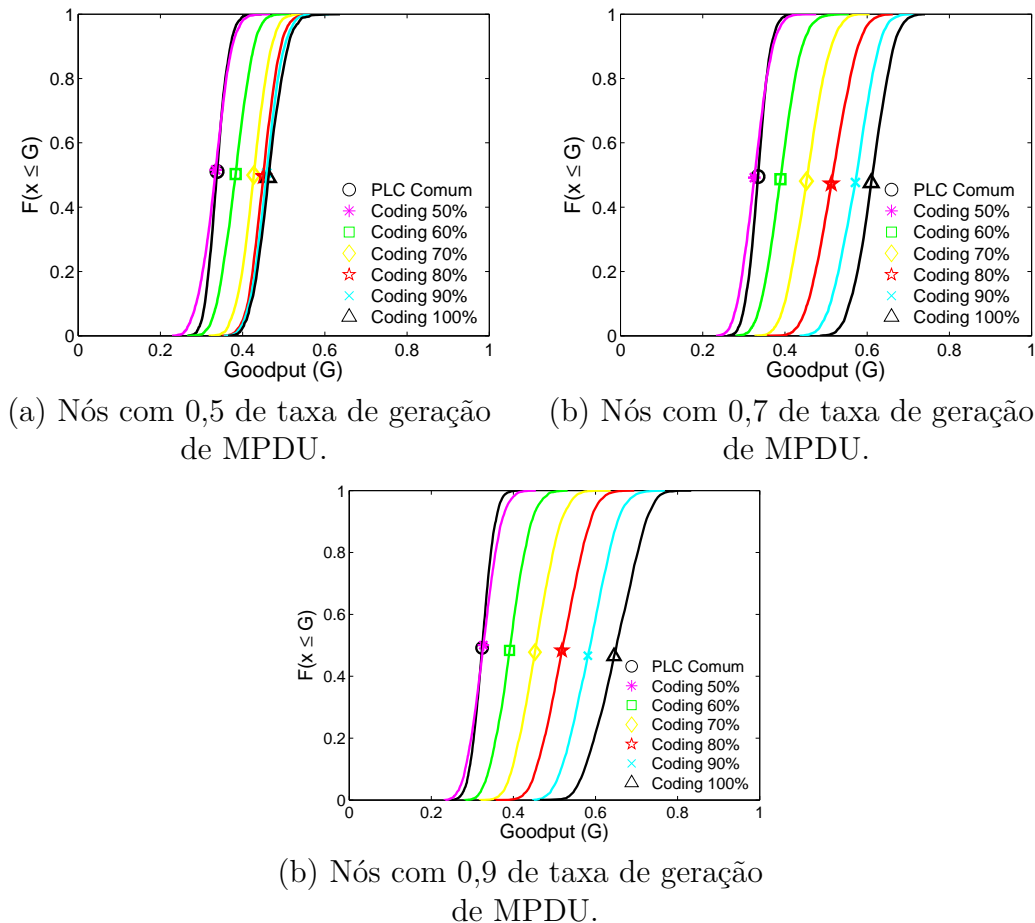


Figura 31 – Médias do *goodput* do sistema em relação à variação de utilização do CodePLC

Nas figuras, as curvas relativas à porcentagem de coding representam a função de distribuição acumulada em relação ao *goodput* de todos os usuários do sistema, de acordo com uma determinada taxa de geração de MPDU's. A curva relativa ao PLC comum se refere a um ambiente onde o canal está ativo 100% do tempo e o algoritmo de repasse utilizado é o *multihop* tradicional e a curva relativa Coding 100% representa a situação de enlaces ativos durante todo o tempo de simulação e utilizando o protocolo CodePLC.

Como é possível observar, a disponibilidade do nó que executa o *network coding* por mais de 50% do tempo de canal ativo gera uma melhoria no desempenho de *goodput* maior do que a utilização do método de PLC comum. As curvas demonstram o aumento de *goodput* ao se utilizar o protocolo CodePLC em relação à geração de MPDU's e mostram

também que esse aumento se estende à utilização da técnica somente durante a parte ativa do canal.

É esperado que um impacto da mudança de disponibilidade do canal se dará na latência dos pacotes de dados que trafegam por ele. A figura 32 apresenta esse impacto. É possível observar que a piora do *delay* dos pacotes acompanha a piora observada no *goodput*. Ou seja, para casos de canal ativo em 60% do tempo ou mais, a utilização do protocolo CodePLC promove ganhos maiores que a utilização do método PLC comum *multihop*.

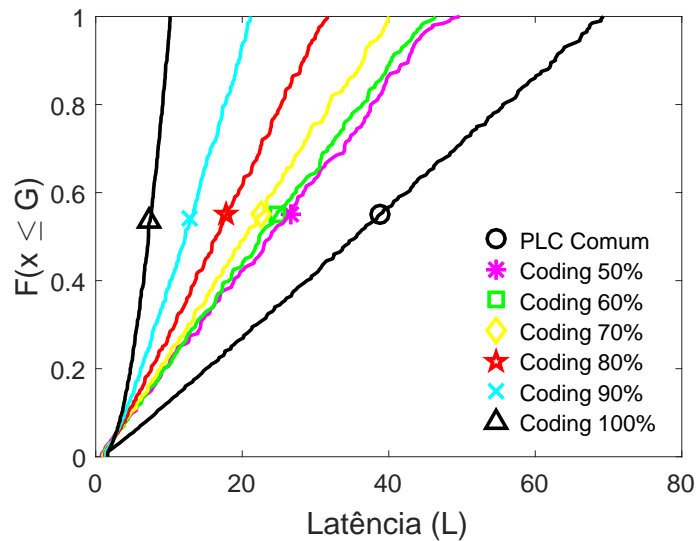


Figura 32 – Distribuição da latência do sistema em relação à variação da disponibilidade do canal.

Diante dos dados apresentados, concluímos que a utilização do CodePLC em cenários onde a disponibilidade da rede é 0,6 ou mais do tempo é mais eficiente, do ponto de vista de *goodput* e latência, que a utilização de um protocolo *multihop* comum em uma rede com canal totalmente disponível.

6.6 INDISPONIBILIDADE DE NÓS VIZINHOS

Outra análise feita nos cenários testados consiste em analisar um cenário onde um dos nós vizinhos está inativo. No cenário inicialmente proposto, fazemos uso de um topologia simétrica que beneficia o desempenho protocolo CodePLC, principalmente em altas cargas de transmissão. Isso se deve ao oportunismo do nó central em sempre ter um número grande de mensagens para codificar, maximizando assim a capacidade de um MPDU em transmitir pacotes aos demais nós da rede.

Nesse cenário, verificaremos o que acontece com o desempenho medido em termos de ocupação de *buffer*, *goodput* e latência quando um dos nós está indisponível, tanto para transmissão quanto para a recepção.

A figura 33 apresenta a comparação entre as funções de distribuição da média de *goodput* nos três casos analisados, para uma taxa de geração de pacotes de 0,7. É possível notar que a indisponibilidade de algum dos nós causa uma queda média de 4% de desempenho nos valores da distribuição.

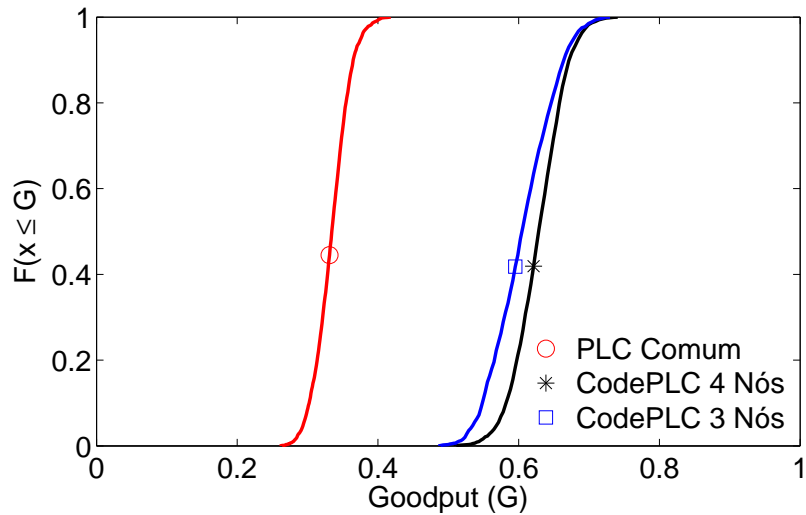


Figura 33 – Distribuição do goodput médio do sistema em relação para o cenário de 1 nó indisponível.

Essa queda acontece mesmo com a diminuição de ocupação dos buffers do sistema, que pode ser observada no gráfico de distribuição da figura 34. Menos MPDU's armazenadas em E e a falta das MPDU's na fila do nó não presente no sistema são responsáveis por essa diminuição. Como consequência, latência também apresentou uma melhoria, devida a menor carga nos *buffers*, como vemos na figura 35.

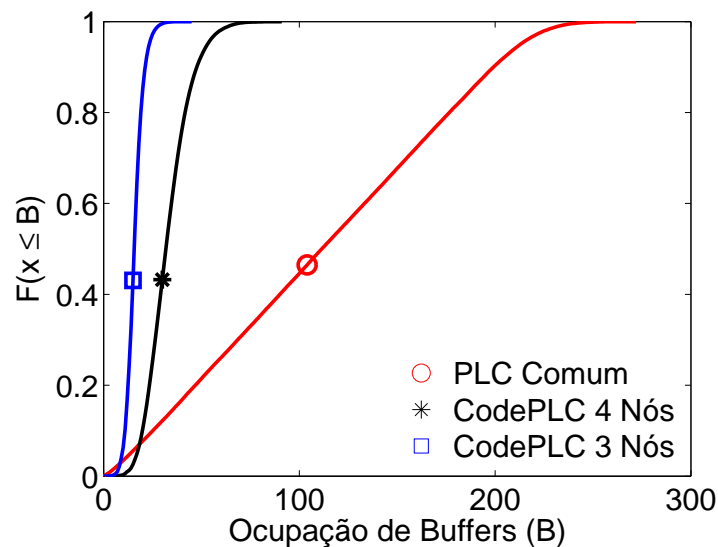


Figura 34 – Distribuição da ocupação de buffer média do sistema em relação para o cenário de 1 nó indisponível.

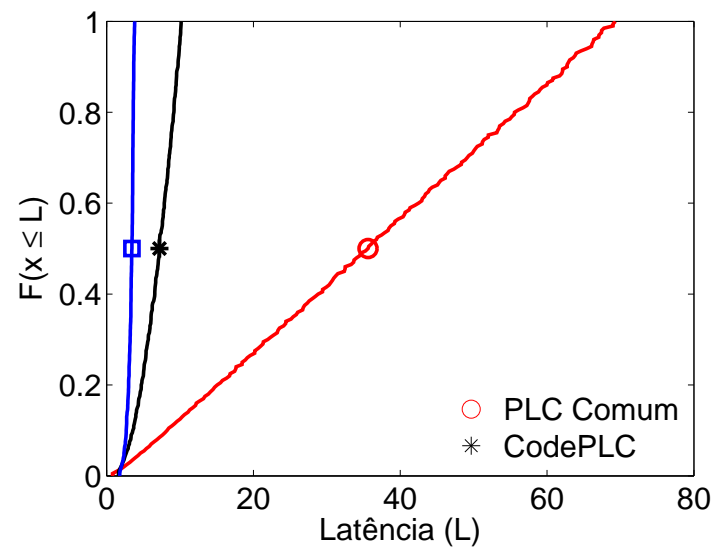


Figura 35 – Distribuição da latência média do sistema em relação para o cenário de 1 nó indisponível.

Com isso, podemos concluir que a queda nos valores de *goodput* advém do menor uso das vantagens em economia de mensagens e *timeslots* proporcionada pela utilização do *network coding*.

7 CONCLUSÃO E TRABALHOS FUTUROS

7.1 CONTRIBUIÇÕES

Neste trabalho, propusemos o protocolo CodePLC, uma implementação de *network coding* para a subcamada MAC de um sistema PLC banda larga *indoor* que utiliza um esquema TDMA-OFDM. Pudemos observar uma melhoria de desempenho satisfatória dentro do ambiente analisado, o que abre caminho para a viabilidade desse tipo de solução em um ambiente de transmissão de dados via rede elétrica real. Resumindo, as conquistas desse trabalho foram:

1. Aumento mediano de até 110% do *goodput* médio dos nós da rede, dependendo da taxa de geração de pacotes.
2. Ocupação média dos *buffers* dos dispositivos reduzida em até 71,4%, dependendo da taxa de geração de pacotes.
3. Latência mediana de MPDU's reduzida em até 4x.
4. Diminuição do problema de congestionamento gerado pela utilização de um algoritmo simples de transmissão *multihop*.
5. Aumento da resiliência do sistema. Com a implementação do CodePLC no cenário analisado, não foi necessário o acréscimo de controle para a confirmação de entrega dos MPDU's que necessitam de mais um salto para chegarem ao destino.
6. Definição de parâmetros para a análise do *tradeoff* entre os ganhos na utilização do CodePLC e a disponibilidade do canal.

Todas as conquistas listadas foram atingidas com uma implementação com baixo custo computacional e com acréscimo de apenas alguns bytes de controle. A utilização da técnica de XOR se mostrou satisfatório dentro do cenário analisado, principalmente por não inserir gastos com memória, recurso que pode ser escasso em sistemas embarcados, cenário natural de dispositivos PLC.

A análise disponibilidade do nó codificador em um sistema CodePLC foi capaz de mostrar que utilizar o protocolo é preferível a utilizar um encaminhamento tradicional *multihop*, mesmo em um cenário onde os enlaces não estão disponíveis em determinado período de tempo.

7.2 TRABALHOS FUTUROS

Como visão de futuro, algumas linhas de pesquisa emergem da comprovação de viabilidade de aplicação de *network coding* na camada de enlace de sistemas PLC. A

primeira é a comparação do protocolo proposto com outros protocolos de *network coding* implementados em ambientes com e sem fio.

Uma segunda linha seria a implementação do protocolo CodePLC em um ambiente de transmissão via rede elétrica, utilizando dispositivos reais. Como o protocolo propõe uma adição de informações de controle, a integração do método em normas e padronizações já existentes pode ser possível utilizando espaços reservados que essas normas oferecem.

Outro problema gerado a partir da técnica de *network coding* analisada é a necessidade, em uma rede real e extensa, da descoberta de nós mais propensos a atuarem como *relays*, dada uma determinada topologia, a fim de se extrair o máximo desempenho da rede em termos de parâmetros como vazão e latência. Esse é um problema de caráter combinatório que, provavelmente, poderá ser analisado no campo de grafos ou outro método matemático propício.

Finalmente, a utilização de TDMA como forma de acesso ao canal foi um dos parâmetros importantes nesse trabalho. Mas é necessário verificar se os ganhos obtidos podem ser estendidos à sistemas que implementem outras formas de acesso ao canal como CSMA/CA, CSMA/CD, híbrido (CSMA + TDMA), Token Ring e Slotted ALOHA.

REFERÊNCIAS

- AHLISWEDE, R.; CAI, N.; LI, S.-Y. R.; YEUNG, R. W. Network Information Flow. *IEEE Transactions On Information Theory*, v. 46, p. 204–1216, Jul. 2000.
- BIAGI, M. MIMO Self-Interference Mitigation Effects on Power Line Relay Networks. *IEEE Communications Letters*, v. 15, n. 8, p. 866–868, Aug. 2011.
- BILBAO, J.; CALVO, A.; ARMENDARIZ, I.; CRESPO, P. M.; MEDARD, M. Reliable communications with Network Coding in narrowband powerline channel. *Proc. IEEE International Symposium on Power Line Communications and its Applications*, p. 316–321, Apr. 2014.
- CAMPOS, F. P. V. de. *Transceptores com modulação multiportadora e técnicas de alocação de recursos em sistemas de transmissão de dados cia rede de energia elétrica*. Tese (Doutorado) — Universidade Federal do Rio de Janeiro (COPPE, Rio de Janeiro, Brasil, Fev. 2013.
- CHENG, X.; CAO, R.; YANG, L. Relay-Aided Amplify-and-Forward Powerline Communications. *IEEE Trans. on Smart Grid*, v. 4, n. 1, p. 265–272, Mar. 2013.
- Cisco Developer Network. *SmartGrid Multiservices FAN Infrastructure Overview*. 2016. <<https://solutionpartnerdashboard.cisco.com/web/cegd/overview>>. Acessado em: 01-08-2016.
- COLEN, G. R.; MARQUES, C. A.; OLIVEIRA, G.; CAMPOS, F. P. V.; RIBEIRO, M. V. Measurement setup for characterizing low-voltage and outdoor electric distribution grids for PLC systems. *Proc. IEEE International Symposium on Wireless Communication Systems*, p. 366–370, 2012.
- DJUKIC, P.; ROGERS, E. S.; VALAEE, S. Delay Aware Link Scheduling for Multi-Hop TDMA Wireless Networks. *IEEE/ACM Transactions on Networking*, IEEE, v. 17, n. 3, p. 870–883, 2009.
- DOUGHERTY, R.; FREILING, C.; ZEGER, K. Insufficiency of Linear Coding in Network Information Flow. *International Symposium on Information Theory*, p. 264–267, Set. 2005.
- EZZINE, S.; ABDELKEFI, F.; CANCES, J. P.; MEGHDADI, V.; BOUALLEGUE, A. Joint network coding and OFDMA based MAC-layer in PLC networks. *18th IEEE International Symposium on Power Line Communications and its Applications (ISPLC)*, p. 311–315, 2014.
- FAROOQIA, M. Z.; TABASSUMA, S. M.; REHMANIA, M. H.; SALEEMB, Y. A survey on network coding: From traditional wireless networks to emerging cognitive radio networks. *Journal of Network and Computer Applications*, Elsevier, v. 46, p. 166–181, 2014.
- FERREIRA, H. C.; LAMPE, L.; NEWBURY, J. *Power Line Communications Theory and Applications for Narrowband and Broadband Communications over Power Lines*. 1. ed. Reino Unido: John Wiley Sons, 2010.
- FRAGOULI, C.; BOUDEC, J.-Y. L.; WIDMER, J. Network coding: an instant primer. *ACM SIGCOMM Computer Communication Review*, ACM, v. 36, n. 1, p. 63–68, 2006.

GAO, Q.; YU, J.; CHONG, P.; SO, P.; GUNAWAN, E. Solutions for the “silent node” problem in an automatic meter reading system using power-line communications. *IEEE Transactions on Power Delivery*, IEEE, v. 23, n. 1, p. 150–156, 2008.

Global Industry Analysts. *MCP-7073: Power Line Communication (PLC) Systems – A Global Strategic Business Report*. 2015. <<http://www.strategyr.com/pressMCP-7073.asp>>. Acessado em: 18-07-2016.

GOLESTANI, S. J.; COMMUN, B. A Framing Strategy for Congestion Management. *IEEE Journal on Selected Areas in Communications*, IEEE, v. 9, n. 7, p. 1064–1077, 2002.

GüZELGÖZ, S.; ARSLAN, H.; ISLAM, A.; DOMIJAN, A. A Review of Wireless and PLC Propagation Channel Characteristics for Smart Grid Environments. *Journal of Electrical and Computer Engineering*, v. 2011, 2011.

HADDED, M.; MUHLETHALER, P.; LAOUITI, A.; ZAGROUBA, R.; SAIDANE, L. A. TDMA-Based MAC Protocols for Vehicular Ad Hoc Networks: A Survey, Qualitative Analysis, and Open Research Issues. *IEEE Communications Surveys & Tutorials*, v. 17, p. 2461–2492, Jun. 2015.

HO, T.; LUN, D. S. *Network Coding: An Introduction*. 1. ed. Reino Unido: Cambridge University Press, 2008.

HOESEL, L. F. W. van; NIEBERG, T.; KIP, H. J.; HAVINGA, P. J. M. Advantages of a TDMA based, energy-efficient, self-organizing MAC protocol for WSNs. *IEEE Vehicular Technology Conference*, v. 3, p. 1598–1602, Mai. 2004.

IEEE 802.16 Working Group on Broadband Wireless Access Standards. *IEEE Std 802.16.1-2012*. [S.l.], 2012. 30-513 p. Disponível em: <<http://standards.ieee.org/getieee802/download/802.16.1-2012.pdf>>.

IEEE-SA. *1901-2010 - IEEE Standard for Broadband over Power Line Networks: Medium Access Control and Physical Layer Specifications*. [S.l.], 2010. Disponível em: <<https://standards.ieee.org/findstds/standard/1901-2010.html>>.

IEEE-SA. *1901.2-2013 - IEEE Standard for Low-Frequency (less than 500 kHz) Narrowband Power Line Communications for Smart Grid Applications*. [S.l.], 2013. Disponível em: <<https://standards.ieee.org/findstds/standard/1901.2-2013.html>>.

International Organization for Standardization. *ISO/IEC 7498-1*. [S.l.], 1996. Disponível em: <<http://www.ecma-international.org/activities/Communications/TG11/s020269e.pdf>>.

KATABI, D.; KATTI, S.; HU, W.; RAHUL, H.; MEDARD, M. On Practical Network Coding for Wireless Environments. *International Zurich Seminar on Communications*, IEEE, v. 1, p. 84–85, 2006.

KATTI, S.; KATABI, D.; HU, W.; RAHUL, H.; MÉDARD, M. *The Importance of Being Opportunistic: Practical Network Coding for Wireless Environments*. 2005. Online. Disponível em: <<https://www.cl.cam.ac.uk/research/srg/netos/papers/2005-allerton-netcoding.pdf>>.

- KATTI, S.; RAHUL, H.; HU, W.; KATABI, D.; MÉDARD, M.; CROWCROFT, J. XORs in the air: practical wireless network coding. *ACM SIGCOMM Computer Communication Review*, v. 36, n. 4, p. 243–254, 2006.
- KESHAVARZ-HADDAD, A.; RIEDI, R. H. Bounds on the benefit of network coding for wireless multicast and unicast. *Mobile Computing, IEEE Transactions on*, IEEE, v. 13, n. 1, p. 102–115, 2014.
- 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*, v. 27, n. 2, p. 886–894, Apr. 2012.
- LEU, F.-Y.; CHEN, H.-L.; LIU, J.-C. Improving Multi-path Congestion Control for Event-Driven Wireless Sensor Networks by Using TDMA. *International Conference on Broadband and Wireless Computing*, v. 1, p. 300–305, Nov. 2014.
- LIANG, L.; GAO, D.; QIN, Y.; ZHANG, H. An adaptive congestion-aware MAC protocol for wireless sensor networks. *IEEE International Conference on Broadband Network and Multimedia Technology*, v. 1, p. 1074–1078, Oct. 2010.
- MALATHI, P.; VANATHI, P. T. Power Line Communication using OFDM and OGA. *AIML Journal*, v. 7, p. 23–30, Jun. 2007.
- MANNAN, A.; D.K.SAXENA; BANDAY, M. A Study on Power Line Communication. *International Journal of Scientific and Research Publications*, v. 4, p. 1–4, Jul. 2014.
- Markets and Markets. *Power Line Carrier Communication & PoE Controller Market worth 12.31 Billion USD by 2022*. 2016. <<http://www.marketsandmarkets.com/Market-Reports/power-line-communication-plc-market-912.html>>. Acessado em: 18-07-2016.
- NOORI, M.; LAMPE, L. Improving data rate in relay-aided power line communications using network coding. *IEEE Global Communications Conference (GLOBECOM)*, IEEE, v. 1, p. 2975–2980, 2013.
- OLIVEIRA, R. M.; FACINA, M. S.; RIBEIRO, M. V.; VIEIRA, A. B. Performance Evaluation of In-Home Broadband PLC Systems Using a Cooperative MAC Protocol. *Computer Networks*, v. 95, p. 62–76, Dec. 2015.
- OSTOVARI, P.; WU, J.; KHREISHAH, A. Network coding techniques for wireless and sensor networks. In: AMMARI, H. M. (Ed.). *The Art of Wireless Sensor Networks*. [S.l.]: Springer, 2014. cap. 5, p. 129–162.
- PARAG, P.; CHAMBERLAND, J.-F. Queueing analysis of a butterfly network for comparing network coding to classical routing. *IEEE Transactions on Information Theory*, IEEE, v. 56, n. 4, p. 1890–1908, 2010.
- PHULPIN, Y.; BARROS, J.; LUCANI, D. Network coding in smart grids. *In. proc. IEEE International Conference on Smart Grid Communications (SmartGridComm)*, p. 49–54, 2011.
- PRIOR, R.; LUCANI, D. E.; PHULPIN, Y.; NISTOR, M.; BARROS, J. Network coding protocols for smart grid communications. *IEEE Transactions on Smart Grid*, IEEE, v. 5, n. 3, p. 1523–1531, 2014.

- RIBEIRO, M. V.; COLEN, G. R.; CAMPOS, F. V. P.; QUAN, Z.; POOR, H. V. Clustered-orthogonal frequency-division multiplexing for power line communication: When can it be beneficial? *IET Communications*, v. 8, n. 13, p. 2336–2347, Sep. 2014.
- SILVEIRA, L. M. F.; OLIVEIRA, R. M.; RIBEIRO, M. V.; VIEIRA, L. F. M.; VIEIRA, M. A. M.; VIEIRA, A. B. CodePLC: A Network Coding MAC Protocol for Power Line Communication. *Proc. IEEE Conference on Local Computer Networks*, Nov. 2016.
- SILVEIRA, L. M. F.; OLIVEIRA, R. M.; RIBEIRO, M. V.; VIEIRA, L. F. M.; VIEIRA, M. A. M.; VIEIRA, A. B. Um Protocolo de Acesso ao Meio com Network Coding em Ambiente PLC. *XXXIV Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, Jun. 2016.
- Telecommunication Standardization Sector of ITU. *G.9961: Unified high-speed wireline-based home networking transceivers*. [S.l.], 2015. Disponível em: <<http://www.itu.int/rec/T-REC-G.9961-201507-I/en>>.
- TONELLO, A. M.; CORTES, J. A.; D’ALESSANDRO, S. Optimal time slot design in an OFDM-TDMA system over power-line time-variant channels. *IEEE International Symposium on Power Line Communications and Its Applications*, p. 41–46, Abr. 2009.
- TSENG, S.-M.; LEE, T.-L.; HO, Y.-C.; TSENG, D.-F. 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*, v. 1, Mar. 2015.
- TSOKALO, I.; LEHNERT, R.; FITZEK, F. H. Intraflow Network Coding on the Data Link Layer for Broadband PLC. *International Symposium on Power Line Communications and its Applications*, IEEE, p. 109–114, Mar. 2016.
- TSS-ITU. *G.9904: Narrowband orthogonal frequency division multiplexing power line communication transceivers for PRIME networks*. [S.l.], 2012. Disponível em: <<https://www.itu.int/rec/T-REC-G.9904/en>>.
- TSS-ITU. *G.9903: Narrowband orthogonal frequency division multiplexing power line communication transceivers for G3-PLC networks*. [S.l.], 2014. 175-199 p. Disponível em: <<https://www.itu.int/rec/T-REC-G.9903>>.
- VALENCIA, J.; OLIVEIRA, T. R.; RIBEIRO, M. V. Cooperative power line communication: Analysis of Brazilian In-Home channels. *Proc. IEEE International Symposium on Power Line Communications and its Applications*, p. 301–305, Apr. 2014.
- YEH, C. I.; KWON, D. S.; KIM, W. W. An efficient TDMA based MAC frame structure in OFDM systems. *IEEE Proceedings on Personal, Indoor and Mobile Radio Communications*, v. 1, p. 2461–2492, Set. 2003.
- YEUNG, R. W.; LI, S.-Y. R.; CAI, N.; ZHANG, Z. *Network Coding Theory*. 2. ed. USA: Foundation and Trends in Communications and Information Theory, 2006.