

Brasil 6G

Projeto Brasil 6G

Otimizações do Transceptor Brasil 6G



Histórico de Atualizações:

Versão	Data	Autor(es)	Notas
1	07/08/2023	Anderson Reis Rufino Marins Diego Gabriel Soares Pivoto João Henrique Silva Delfino Juliano Silveira Ferreira	Elaboração de conteúdo
2	22/08/2023	Luciano Leonel Mendes	Revisão de texto

Lista de Figuras

1	Estrutura organizacional do relatório.	2
2	Diagrama simplificado da rede de acesso e etapas do transceptor Brasil 6G.	3
3	MIMO com multiplexação espacial: taxa de dados consumidos pela UE.	6
4	Taxa de comunicação obtida na BS para teste do enlace de <i>downlink</i>	7
5	Validação do transceptor Brasil 6G (versão de complexidade reduzida) no novo ambiente de desenvolvimento.	9
6	Rede de acesso Brasil 6G utilizada inicialmente como referência para testes.	10
7	Carga computacional com o <i>software</i> da camada MAC operando de forma isolada.	12
8	Carga computacional com o <i>software</i> da camada MAC operando em conjunto com o <i>software</i> GNU Radio executando projeto <i>loopback</i> do transmissor.	13
9	<i>Setup</i> para cenário de testes de desempenho da Camada MAC.	17
10	Testes de <i>uplink</i> para avaliação de desempenho da taxa de transferência da rede.	18
11	Testes de <i>downlink</i> para avaliação de desempenho da taxa de transferência considerando 1 <i>User Equipment</i> (UE) conectada à rede.	19
12	Testes de <i>downlink</i> para avaliação de desempenho da taxa de transferência considerando 2 UEs conectadas à rede.	20
13	Testes de <i>downlink</i> para avaliação de desempenho da taxa de transferência considerando 3 UEs conectadas à rede.	21
14	Diagrama do algoritmo <i>Low-Complexity Deep Neural Network for Signal-to-Interference-Plus-Noise Ratio Estimation</i> (LDNN-SNRE).	24
15	Cadeia de comunicação Ponto-a-Ponto.	25
16	Amostragem e geração do <i>Dataset</i>	26
17	Estrutura da rede neural criada.	27
18	Saída dos modelos para as modulações: QPSK, 16-QAM, 64-QAM e 256-QAM.	28
19	Diagrama da simulação.	29
20	Resultado da análise de desempenho para 256-QAM.	30
21	Diagrama em blocos do processo de DFT na matriz de entrada do modulador GFDM do transceptor Brasil 6G.	35
22	Diagrama em blocos do processo de IDFT na matriz de saída do modulador GFDM do transceptor Brasil 6G.	36
23	PAPR 4-QAM para $K = 8192$, $M = 4$, $(roll-off)\alpha = 0$	37
24	SER 4-QAM para $K = 8192$, $M = 4$, $(roll-off)\alpha = 0$	37

Lista de Tabelas

1	Recursos de software para desenvolvimento do transceptor	7
2	Especificações técnicas dos computadores alocados para <i>Base Station</i> (BS) e uma UE para análise de carga computacional exigida.	11
3	Resultados de uso médio por CPU para análise de carga computacional na BS e UE.	14
4	Limitações da Camada MAC.	15
5	Descrição de <i>Hardware</i>	23
6	Descrição de <i>Software</i>	23
7	Bibliotecas	23
8	Tabela de MCS	29

Acrônimos

5G Quinta Geração de Rede Móvel Celular

5G-NR 5G *New Radio*

6G Sexta Geração de Rede Móvel Celular

ADAM *Adaptative Momentum*

AGC *Automatic Gain Control*

API *Application Programming Interfaces*

BER *Bit Error Rate*

BS *Base Station*

CAB *Colônia Artificial de Abelhas*

CCDF *Complementary Cumulative Distribution Function*

CPU *Central Processing Unit*

CSV *Comma-Separated Values*

CSO-PTS *Cumulative Symbol Optimization PTS*

DFT *Discrete Fourier Transform*

DFT-s-GFDM *DFT-spread-GFDM*

DFT-s-OFDM *DFT-spread-OFDM*

DFT-s-OTFS *DFT-spread-OTFS*

DL *Deep Learning*

ELM-TR *Extreme Learning Machine-based TR*

F-OFDM *Filtered OFDM*

GFDM *Generalized Frequency Division Multiplexing*

GPU *Graphics Processing Unit*

IA *Inteligência Artificial*

IDFT *Inverse Discrete Fourier Transformer*

IFFT *Inverse Fast Fourier Transformer*

ISFFT *Inverse Symplectic Finite Fourier Transform*

JIT *Just-in-Time*

JSON *JavaScript Object Notation*

LC-GFDM *Low Complexity GFDM*

LDNN-SNRE *Low-Complexity Deep Neural Network for Signal-to-Interference-Plus-Noise Ratio Estimation*

LSA-TR *Least Square Approximation TR*

LTE *Long Term Evolution*

LTS *Long Term Support*

MAC *Media Access Control*
MCS *Modulation Coding Scheme*
MIMO *Multiple-Input Multiple-Output*
ML *Machine Learning*
MMSE *Minimum Mean Square Error*
MSE *Mean Squared Error*
NMSE *Normalized Mean Squared Error*
OFDM *Orthogonal Frequency Division Multiplexing*
OTFS *Orthogonal Time Frequency Space*
PAPR *Peak-to-Average Power Ratio*
PHY *Physical Layer*
PID *Proportional–Integral–Derivative*
PRNet *PAPR Reducing Network*
PTS *Partial Transmission Sequence*
SIMD *Single Instruction Multiple Data*
QAM *Quadrature Amplitude Modulation*
QoS *Quality of Service*
QPSK *Quadrature Phase Shift Keying*
RAN *Radio Access Network*
RF *Radio Frequency*
S-SCR *Scaling Signal to Clipping noise Ratio*
SC-FDMA *Single Carrier Frequency Division Multiple Access*
SDR *Software Defined Radio*
SER *Symbol Error Rate*
SLM *Selective Mapping*
SNR *Signal-to-Noise Ratio*
TR *Tone Reservation*
TUN *Tunnel Software Network Interface*
UE *User Equipment*
ZF *Zero Forcing*

Sumário

1	Introdução	1
2	Otimizações do Transceptor	3
2.1	Otimizações da camada física (PHY)	4
2.1.1	Implementação de melhorias e novas funcionalidades	4
2.1.2	Migração do ambiente de desenvolvimento	6
2.1.3	Validação no GNU Radio	8
2.2	Otimizações da camada de Enlace (MAC):	8
2.2.1	Teste 1: Análise de carga computacional do <i>software</i> da camada MAC	10
2.2.2	Teste 2: Otimizações do MAC com camada física emulada	14
3	Integração de ferramentas de I.A. na camada física	22
3.1	Recursos computacionais utilizados	22
3.2	Técnicas de Estimação de SNR	23
3.2.1	Estimador baseado em NMSE	24
3.2.2	Estimador baseado em LDNN-SNRE	24
3.3	Implementação da técnica baseada em IA	24
3.3.1	Etapas principais da Camada física do transceptor utilizadas para integração e testes	25
3.3.2	Função MSE modificada	25
3.3.3	Amostragem, pré-processamento e geração do <i>Dataset</i>	25
3.3.4	Treinamento e validação do modelo de DL	26
3.3.5	Integração a camada física	26
3.4	Análise e resultados	27
4	Forma de onda: adequações do GFDM para redução da PAPR	31
4.1	Estudo de técnicas para Redução de PAPR	31
4.1.1	Pré-codificação DFT	31
4.1.2	<i>Partial Time Sequence</i>	32
4.1.3	<i>Selective Mapping</i>	33
4.1.4	Inteligência Artificial	33
4.2	Integração da Técnica de Redução de PAPR no Transceptor Brasil 6G	35
4.3	Testes e Resultados alcançados	36
5	Conclusão	39

1 Introdução

Diego Gabriel Soares Pivoto, Juliano Silveira Ferreira

O projeto Brasil 6G tem como um de seus objetivos conceber e desenvolver uma plataforma de testes e avaliação de diferentes tecnologias que possam vir a integrar a Sexta Geração de Rede Móvel Celular (6G). Dentre os elementos que compõe a plataforma proposta, pode-se destacar o transceptor Brasil 6G, que é o elemento principal da *Radio Access Network* (RAN).

As técnicas de processamento digital de sinais e as funcionalidades implementadas no transceptor impactam diretamente na capacidade de transmissão, na taxa de dados suportada, na cobertura da rede de acesso e na confiabilidade das comunicações estabelecidas. A realização de melhorias contínuas no transceptor envolvendo ajustes, novos parâmetros de operação, funcionalidades e tecnologias são essenciais para se alcançar uma solução técnica mais completa e efetiva, tanto para a camada *Media Access Control* (MAC) quanto para a *Physical Layer* (PHY).

Dentre as tecnologias eleitas como habilitadoras para as redes 6G, podem-se destacar aquelas associadas à Inteligência Artificial (IA). A expectativa é que tais tecnologias sejam empregadas em diferentes camadas da rede para otimizar o gerenciamento de recursos e aplicações, melhorar a eficiência espectral, garantir a segurança da rede, dentre várias outras possibilidades. Com relação à camada física, as tecnologias de IA também podem ser utilizadas de diversas formas, visando assegurar, por exemplo, maior cobertura da rede e qualidade do sinal. Um exemplo de uso de IA refere-se ao monitoramento de parâmetros relacionados às condições do enlace de comunicação como, por exemplo, a estimativa de *Signal-to-Noise Ratio* (SNR) que, por sua vez, impacta na definição do *Modulation Coding Scheme* (MCS) empregado por cada terminal de usuário ou UE pertencente à rede. Vale destacar que esta funcionalidade inovadora foi desenvolvida ao longo das atividades do projeto Brasil 6G.

O processo de desenvolvimento de otimizações do transceptor Brasil 6G também envolve pesquisas relacionadas ao *Generalized Frequency Division Multiplexing* (GFDM), que é a forma de onda adotada no rádio. O GFDM é uma forma de onda bastante versátil, podendo ser configurado para operar como o *Orthogonal Frequency Division Multiplexing* (OFDM). As investigações e testes realizados tiveram como foco técnicas de redução de *Peak-to-Average Power Ratio* (PAPR) da referida forma de onda. Como possíveis contribuições e resultados, a redução de PAPR permite otimizar a eficiência energética do sistema, uma vez que a potência média de RF irradiada pelos transceptores pode ser aproximada à potência de pico.

Em resumo, a evolução dos sistemas de comunicações móveis para o 6G e a implementação de melhorias nos transceptores por meio de otimizações das camadas PHY e MAC, ferramentas de IA e otimização das formas de onda, desempenham um papel crucial para proporcionar redes móveis mais rápidas, confiáveis e eficientes. Baseado nessa premissa, este relatório apresenta a evolução da rede de acesso a ser implementada na plataforma do Projeto Brasil 6G, destacando as principais pesquisas e melhorias realizadas sob esta pauta. A Figura 1 ilustra a estrutura organizacional deste documento.

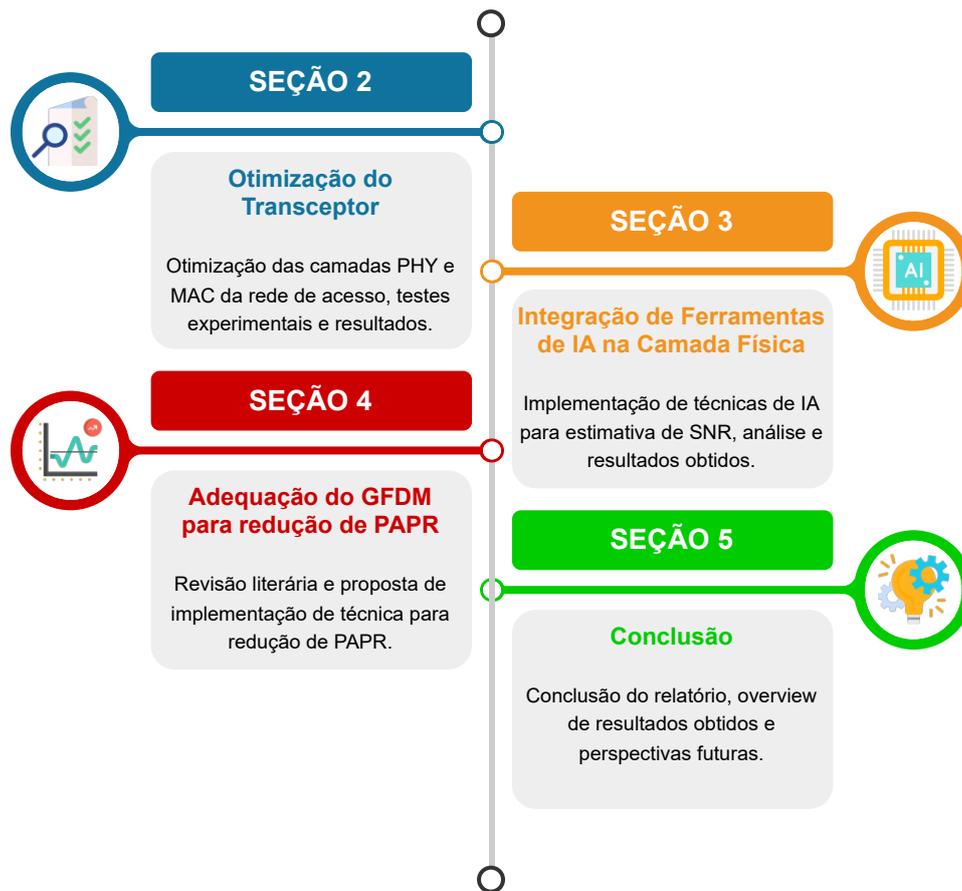


Figura 1: Estrutura organizacional do relatório.

A Seção 2 apresenta as principais otimizações realizadas nos transceptores Brasil 6G, especificamente nas camadas física e de enlace, bem como resultados de testes laboratoriais. A Seção 3, por sua vez, introduz o uso de ferramentas de IA na camada física para propôr a estimação de SNR do canal de forma otimizada. Já a Seção 4 apresenta uma breve revisão literária e análise relacionada à técnicas para redução de PAPR, considerando a forma de onda GFDM. Por fim, a Seção 5 aborda a conclusão deste documento, apresentando resultados e considerações finais.

2 Otimizações do Transceptor

Diego Gabriel Soares Pivoto, João Henrique Silva Delfino, Juliano Silveira Ferreira

A plataforma Brasil 6G tem como um elemento de destaque sua rede de acesso, cujo dispositivo principal, por sua vez, é o transceptor Brasil 6G. A Figura 2 apresenta um diagrama bastante simplificado da rede de acesso Brasil 6G, na qual os transceptores são colocados em evidência. O transceptor Brasil 6G é formado por duas etapas principais de processamento, conforme ilustrado na figura citada, que são: camada de enlace ou camada MAC e camada física ou PHY. O transceptor desenvolvido pode ser configurado para operar como a BS ou UE da rede de acesso. A BS é, em linhas gerais, o terminal responsável por prover acesso à internet e a outros servidores, além de definir os parâmetros de operação tanto da rede de acesso quanto dos terminais configurados como UE.

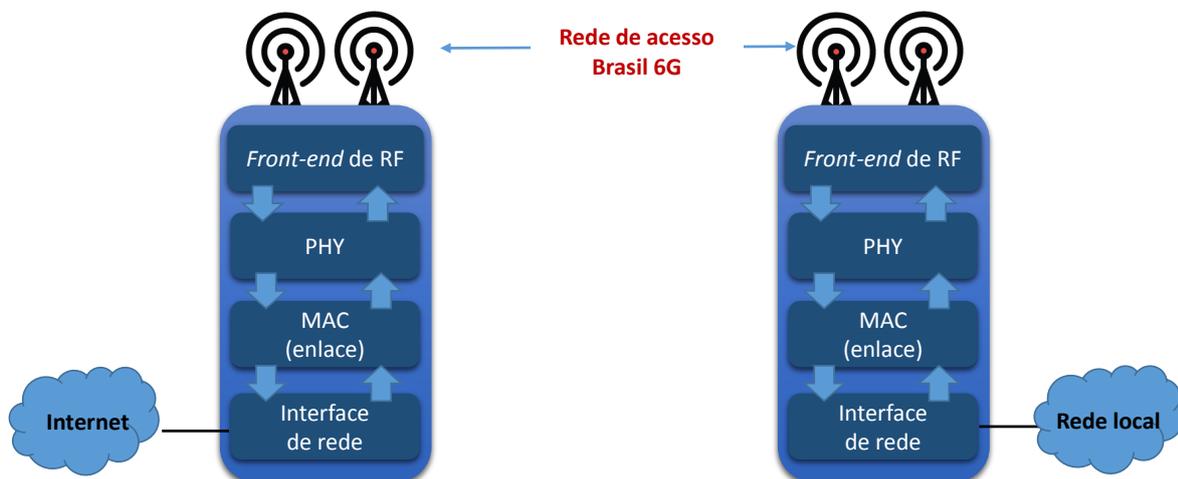


Figura 2: Diagrama simplificado da rede de acesso e etapas do transceptor Brasil 6G.

O desenvolvimento do transceptor Brasil 6G adotou o paradigma de implementação denominado de *Software Defined Radio* (SDR). Tal abordagem parte do princípio que as etapas de processamento digital de sinais são executadas em sua maioria através de uma plataforma de *software*. Esta é integrada a um dispositivo de *hardware*, responsável pelas conversões digital-analógico assim como as conversões de sinal em banda base para o canal desejado, bem como as conversões inversas à estas. A etapa de *hardware* é denominada de *front-end Radio Frequency* (RF) e se encontra também ilustrada na Figura 2. Vale destacar que as etapas principais de processamento que compõe as camadas MAC e PHY do transceptor, mencionada anteriormente, são executadas na plataforma de *software* desenvolvida.

Vale ressaltar que o desenvolvimento do transceptor Brasil 6G considera duas versões distintas, como detalhado em [1], que são:

- Transceptor Brasil 6G de complexidade reduzida: com suporte à comunicação ponto-a-ponto e à funcionalidades de menor complexidade;
- Transceptor Brasil 6G de alta capacidade: com suporte a múltiplos usuários e funcionalidades e aplicações mais avançadas, que incluem a camada MAC. Esta versão emprega um

hardware de maior complexidade e capacidade em comparação com o transceptor Brasil 6G de complexidade reduzida.

A presente seção apresenta as otimizações realizadas nas duas versões do transceptor Brasil 6G, cujos avanços estão mostrados em duas subseções: a subseção 2.1 detalha as otimizações implementadas na camada PHY, enquanto a subseção 2.2 descreve as melhorias realizadas na camada MAC.

2.1 Otimizações da camada física (PHY)

Conforme mencionado, foram realizadas correções, melhorias e ajustes em diferentes etapas que compõe a camada física do transceptor Brasil 6G, bem como desenvolvimento de novas funcionalidades. As principais atividades realizadas nestes quesitos encontram-se apresentada na subseção 2.1.1.

Um desenvolvimento inovador realizado e que pode-se destacar refere-se à integração de IA na camada PHY, cujos resultados foram detalhados em uma seção específica, a Seção 3. Para viabilizar que esta integração fosse possível, foi necessário realizar previamente uma otimização relacionada à atualização das ferramentas empregadas no ambiente de desenvolvimento e execução da camada PHY do transceptor. As justificativas da necessidade deste processo de atualização, bem como as atividades envolvidas e resultados alcançados, encontram-se apresentadas na subseção 2.1.2.

2.1.1 Implementação de melhorias e novas funcionalidades

A presente seção apresenta um resumo das principais melhorias e desenvolvimento de novas funcionalidades aplicáveis principalmente à versão do transceptor de alta capacidade. Esta versão do transceptor é aquela que integra funcionalidades mais avançadas como o suporte à múltiplos usuários e tecnologias *Multiple-Input Multiple-Output* (MIMO).

Um ajuste importante realizado no transceptor envolve a definição do MCS a ser utilizado na comunicação com cada UE. A definição do MCS de operação do transceptor é realizada com base na SNR identificada pelo receptor de um dado enlace. A SNR estimada pela UE, por exemplo, é transmitida para o terminal configurado como BS. A camada de enlace da BS define os MCS do enlaces de *uplink* de cada UE com base em tais estimativas. Estes parâmetros podem variar rapidamente em função das condições momentâneas do canal de comunicação, impactando na frequente troca do MCS. A mudança contínua da modulação e forma de codificação pode impactar negativamente na taxa de comunicação do enlace, uma vez que um MCS restritivo pode ser adotado em função de uma condição momentânea. Por conseguinte, a implementação de um filtro permite reduzir essa mudança constante de operação, visto que o MCS é alterado caso seja identificado que as condições médias de SNR do enlace encontram-se mais estáveis.

De maneira similar ao feito para a definição do MCS, foram realizados também ajustes no algoritmo do *Automatic Gain Control* (AGC), visando otimizar as trocas de ganho realizadas na recepção do transceptor. Para isso, otimizou-se a operação da malha de controle fechada, a qual implementa a técnica *Proportional-Integral-Derivative* (PID) para o controle do AGC. Isso permite reduzir a variação do ganho de recepção em função da incidência de interferências momentâneas no sinal recebido.

Além disso, foram implementadas novas funcionalidades na camada PHY do transceptor Brasil 6G, mais especificamente o transceptor de alta capacidade, relacionado à técnicas MIMO.

Foram implementados novos detectores MIMO para multiplexação espacial. Vale ressaltar que a técnica MIMO para diversidade espacial já se encontrava previamente implementada, sendo a mesma utilizada nos testes já documentados anteriormente. As técnicas MIMO associadas à multiplexação espacial permitem aumentar a vazão de dados de um sistema de comunicação, pois, empregam as múltiplas antenas para a transmissão de múltiplos fluxos de dados de maneira simultânea. Um sistema que utiliza M antenas de transmissão, nesse caso, pode teoricamente ampliar a vazão de comunicação em até M vezes quando comparado com o mesmo sistema com antena única [2], considerando que o número de antenas receptoras seja $N \geq M$.

A operação de sistemas MIMO com multiplexação espacial pode ser considerada bastante desafiadora, uma vez que o transmissor irradia simultaneamente sinais distintos na mesma faixa de frequência causando assim, interferência co-canal. No caso dos detectores MIMO implementados no transceptor Brasil 6G, que são detectores lineares de baixa complexidade, *Zero Forcing* (ZF) e *Minimum Mean Square Error* (MMSE), o canal de transmissão envolvido deve possuir características adequadas para viabilizar que o algoritmo de detecção possa separar os sinais oriundos de cada antena transmissora. A capacidade de um canal de comunicação depende tanto da largura de banda empregada pelo sistema e da SNR. O condicionamento do canal de comunicação é tido como um fator adicional que influencia as condições de operação de certas técnicas MIMO e que impacta na capacidade do canal de comunicação. Sendo assim, foi implementado no transceptor Brasil 6G, além dos equalizadores ZF e MMSE, o algoritmo para cálculo da métrica do condicionamento do canal, conforme melhor detalhado no relatório [3]. A partir do valor estimado do condicionamento do canal é verificada a viabilidade de uso da técnica de multiplexação espacial. Um valor inicialmente adotado para validar esta viabilidade é que o condicionamento do canal seja inferior a 10 dB [3].

A figura 3 mostra o resultado de testes laboratoriais considerando o uso dos detectores MIMO desenvolvido. A Figura em questão mostra a vazão de dados para um UE que emprega a técnica de multiplexação espacial. As condições de testes consideram o emprego de um canal de comunicação adequado para o uso de tal técnica, condicionamento de canal igual a 0, além de uma elevada SNR. Nestas condições, foi possível obter a taxa de aproximadamente 44 Mbps como destacado na Figura a 3. Este valor de taxa equivale ao dobro daquele alcançado quando se emprega a técnica MIMO de diversidade espacial, que é de cerca de 22 Mbps. Tais resultados foram obtidos no enlace de *uplink* e considerando a alocação de 6 MHz para a UE em teste. Ou seja, o aumento da vazão mencionado foi alcançado sem demandar o aumento da largura de banda ocupada pela UE. Maiores detalhes sobre as implementações e testes relacionados a MIMO, podem ser encontrados em [3]. Vale ressaltar que os algoritmos de detecção de multiplexação espacial demandam maior tempo de processamento, devido sua maior complexidade, em relação a diversidade espacial. Os testes iniciais de tempo de processamento também foram apresentados no relatório [3].

Uma outra nova funcionalidade implementada na camada PHY do transceptor refere-se à captura e armazenamento de sinais e informações específicas de tal camada. Esta funcionalidade foi prevista para viabilizar que informações obtidas em condições de operação real do transceptor possam ser analisadas posteriormente por meio de, por exemplo, simulações computacionais. Além disso, tais informações capturadas podem ser utilizadas em pesquisas e avaliações de novos algoritmos ou técnicas que possam ser futuramente integradas ao transceptor ou à rede de acesso. Uma importante informação considerada inicialmente são as condições do canal de comunicação estimada pelo transceptor. Um primeiro exemplo de uso de tais informações refere-se à análise de desempenho de detectores MIMO, implementados inicialmente em simulações computacionais. Desta forma, novos detectores MIMO podem ser avaliados via simulação,

```
[Brasil 6G][BR6G Transceptor][PHY RX STATUS] =====
[UE - User Equipment] 1
Numerology : 0
MIMO : MULTIPLEXING
Precode : 0
Mod : 6
Code rate : 0,420238 (0,420238)
[PHY RX] MSE : -39.2518
RX Data Throughput (Mbps): 0.700812
RX SNR : 39.2518 [dB]
RX Channel Condition Number: 0 [dB]

[ 4] 20.00-21.00 sec 5.54 MBytes 46.5 Mbits/sec 0 2.47 MBytes
[ 4] 21.00-22.00 sec 5.48 MBytes 45.9 Mbits/sec 0 2.49 MBytes
[ 4] 22.00-23.00 sec 5.45 MBytes 45.7 Mbits/sec 0 2.51 MBytes
[ 4] 23.00-24.00 sec 5.53 MBytes 46.4 Mbits/sec 0 2.55 MBytes
[ 4] 24.00-25.00 sec 5.51 MBytes 46.2 Mbits/sec 0 2.60 MBytes
[ 4] 25.00-26.00 sec 5.53 MBytes 46.3 Mbits/sec 20 2.64 MBytes
[ 4] 26.00-27.00 sec 5.37 MBytes 45.2 Mbits/sec 0 2.65 MBytes
[ 4] 27.00-28.00 sec 5.56 MBytes 46.7 Mbits/sec 2 2.65 MBytes
[ 4] 28.00-29.00 sec 5.46 MBytes 45.7 Mbits/sec 0 2.65 MBytes
[ 4] 29.00-30.00 sec 5.49 MBytes 46.0 Mbits/sec 0 2.65 MBytes

-----
[ ID] Interval      Transfer    Bandwidth    Retr
[ 4]  0.00-30.00 sec   160 MBytes 44.8 Mbits/sec 53      sender
[ 4]  0.00-30.00 sec   159 MBytes 44.6 Mbits/sec           receiver

iperf Done.
root@modem:~# iperf3 -c 10.0.0.10 -t 30
```

Figura 3: MIMO com multiplexação espacial: taxa de dados consumidos pela UE.

porém, considerando condições de canal que incluem interferências e ruídos de RF, polarização de antenas, arredondamentos e ajustes realizados por etapas de processamento como conversão analógico-digital do *front-end* de RF, bem como estimação e equalização de canal. A influência destas etapas de processamento e condições reais de operação são normalmente difíceis de serem reproduzidas em simulações computacionais. Desta forma, as informações capturadas a partir do transceptor permitem que condições realistas e mais complexas possam ser consideradas durante a realização de pesquisas, análises e avaliações de diferentes algoritmos e técnicas.

Foram implementadas melhorias conjuntas na camada MAC e PHY que permitiram ao transceptor operar com a modulação *256-Quadrature Amplitude Modulation* (QAM), viabilizando alcançar em condições de laboratório taxas de comunicação acima de 100 Mbps. As melhorias realizadas estão relacionadas principalmente à otimização do uso de recursos computacionais, através da adoção de recursos de processamento paralelo onde as instruções de execução consideram a manipulação de múltiplos dados, *Single Instruction Multiple Data* (SIMD), na camada MAC e em algumas etapas da camada PHY. A Figura 4 mostra a taxa de dados de *downlink* alcançada a partir da implementação de tais melhorias. Pode-se verificar, destacado na parte inferior da Figura 4, que foi alcançada a taxa de comunicação da ordem de 130 Mbps. Reforçando que tal condição foi alcançada a partir de condições laboratoriais com canal de comunicação e SNR favoráveis. De qualquer forma, o teste evidencia que as otimizações de uso dos recursos computacionais favoreceram a execução das rotinas do transceptor, permitindo ao mesmo prover maiores taxas de comunicação.

2.1.2 Migração do ambiente de desenvolvimento

Foi definido e adotado um ambiente inicialmente para o desenvolvimento e teste das etapas de processamento em *software* que compõe a camada PHY. O ambiente inicial é formado principalmente pelo sistema operacional Ubuntu versão 18.04, GNU Radio versão 3.7 e Python versão 2.7, como descrito no relatório [1]. Conforme também mencionado no referido relatório, um novo ambiente de desenvolvimento seria alvo de avaliação durante a execução das ativi-

```
[Brasil 6G][BR6G Transceptor][PHY RX STATUS] =====
[Base Station]
Numerology : 0
MIMO : DIVERSITY
Mod : 8
Code rate : 0.645089 (0.645089)

RX SNR : 41.878 [dB]
RX Channel Condition Number: 0 [dB]

[ 4] 20.00-21.00 sec 15.0 MBytes 126 Mbits/sec 0 6.00 MBytes
[ 4] 21.00-22.00 sec 15.0 MBytes 126 Mbits/sec 0 6.00 MBytes
[ 4] 22.00-23.00 sec 15.0 MBytes 126 Mbits/sec 0 6.00 MBytes
[ 4] 23.00-24.00 sec 15.0 MBytes 126 Mbits/sec 0 6.00 MBytes
[ 4] 24.00-25.00 sec 15.0 MBytes 126 Mbits/sec 0 6.00 MBytes
[ 4] 25.00-26.01 sec 15.0 MBytes 125 Mbits/sec 0 6.00 MBytes
[ 4] 26.01-27.00 sec 15.0 MBytes 127 Mbits/sec 0 6.00 MBytes
[ 4] 27.00-28.00 sec 16.2 MBytes 136 Mbits/sec 0 6.00 MBytes
[ 4] 28.00-29.01 sec 15.0 MBytes 125 Mbits/sec 0 6.00 MBytes
[ 4] 29.01-30.00 sec 15.0 MBytes 126 Mbits/sec 0 6.00 MBytes

[ ID] Interval      Transfer  Bandwidth  Retr
[ 4] 0.00-30.00 sec 470 MBytes 131 Mbits/sec 26
[ 4] 0.00-30.00 sec 447 MBytes 125 Mbits/sec

iperf Done.
root@modem:~# iperf3 -c 10.0.0.11 -t 30
```

Figura 4: Taxa de comunicação obtida na BS para teste do enlace de *downlink*.

dades do projeto Brasil 6G envolvendo principalmente o emprego de versões mais recentes do sistema operacional Ubuntu, bem como do GNU Radio e Python. A adoção de versões mais recentes permitem que ferramentas mais novas possam ser utilizadas pela plataforma. Uma motivação adicional para a atualização do ambiente se refere à integração de ferramentas de IA no transceptor. Ferramentas eleitas como base para o desenvolvimento de IA, como o TensorFlow, oferecem suporte apenas para versões mais recentes da linguagem Python (versão 3.0 ou superiores). Dessa forma, tornou-se uma necessidade a migração do projeto para um ambiente de desenvolvimento atualizado e compatível com as principais ferramentas de desenvolvimento de IA.

A migração de ambientes de desenvolvimento é uma tarefa que requisita uma análise e compreensão a cerca das versões e a compatibilização dos recursos utilizados no sistema. Dessa maneira, optou-se pela seleção de uma versão mais recente do sistema Operacional, sendo adotado o Ubuntu versão 20.04, por ser uma versão com suporte de longo prazo, *Long Term Support (LTS)*, e que já estava sendo avaliada pela equipe. A versão do Python presente nativamente neste sistema operacional é a versão 3.8.10, que é compatível com diversas ferramentas de desenvolvimento de IA. Além disso, foram adotadas versões mais recentes de bibliotecas como Numpy e Pandas, dentre outras, que são empregadas como base para o desenvolvimento de algoritmos empregando IA.

A tabela 1 apresenta as versões das principais ferramentas de *software* adotadas para o novo ambiente de desenvolvimento da camada PHY, bem como o link de acesso para as mesmas.

Tabela 1: Recursos de software para desenvolvimento do transceptor

Ferramentas e Bibliotecas	Versão	Acesso
GNU Radio	v3.9.7	https://github.com/gnuradio/gnuradio
Volk	v2.5.1	https://github.com/gnuradio/volk
AFF3CT	v2.3.5	https://github.com/aff3ct/aff3ct
NI PCIe Driver	v2022.Q4	https://www.ni.com/pt-br/support/downloads/drivers/download.ni-linux-device-drivers.html#467701
UHD	v4.3.0.0	https://github.com/EttusResearch/uhd

Vale destacar que há alguns pontos críticos que envolveram o processo de migração para

a versão atualizada do ambiente de desenvolvimento. Um dos pontos principais se refere, principalmente, à mudança da versão do GNU Radio, que é a ferramenta principal utilizada para o desenvolvimento dos blocos de processamento digital de sinais da camada PHY. A adoção da versão mais recente do GNU Radio demandou que todos os blocos desenvolvidos especificamente pela equipe do projeto Brasil 6G, denominados de blocos *Out-of-tree*, tivessem de ser adaptados. Um ponto de adaptação se refere a criação de um novo arquivo de integração e interfaceamento de cada bloco com o GNU Radio. A versão mais recente do GNU Radio adota um formato do tipo *YAML Ain't Markup Language* (YAML) para tal arquivo, enquanto a versão antiga utiliza o formato *Extensible Markup Language* (XML). O novo formato adotado é mais simples e recorda um mapa no estilo chave-valor, em que sua legibilidade é uma característica evidente.

Aliado a estes fatores, vale ressaltar que as estrutura de dados e algoritmos de processamento principais empregados pelo transceptor não necessitaram de grandes alterações. Porém, alguns métodos principais, como o construtor, tiveram de ser alterados em virtude da mudança da forma da declaração deste na nova versão do GNU Radio. Adicionalmente, os blocos ou funções de processamento desenvolvidas em linguagem Python também tiveram de ser atualizados e adaptados, visando a compatibilização dos mesmos com a nova versão adotada desta linguagem e de suas ferramentas.

2.1.3 Validação no GNU Radio

Após finalizada a etapa de migração de todos os blocos de processamento que compõe a camada PHY do transceptor Brasil 6G, foi desenvolvido um novo projeto para a integração dos blocos e reconstrução do transceptor.

A Figura 5 mostra uma visão de um dos testes de validação do transceptor Brasil 6G, neste caso a versão de complexidade reduzida, já no novo ambiente de desenvolvimento. Na parte esquerda da referida figura é possível observar o projeto com os novos blocos do transceptor já integrados na nova versão do GNU Radio [4]. À esquerda da mesma figura é possível observar a constelação de recepção do transceptor. No teste em questão, o sinal recebido e demodulado foi gerado por um transceptor desenvolvido no ambiente inicial de desenvolvimento. Desta forma, pode-se atestar que o processo de migração de ambiente foi realizado com sucesso e que esta versão do transceptor opera de forma similar em ambos os ambientes.

O processo de validação da migração de ambiente foi realizado primeiramente na versão de complexidade reduzida do transceptor, devido ao fato do mesmo envolver uma complexidade menor. O processo de migração dos blocos de processamento que compõe o transceptor Brasil 6G em sua versão de alta capacidade também foi realizada com sucesso e encontra-se atualmente em fase de validação.

2.2 Otimizações da camada de Enlace (MAC):

A camada de enlace, também conhecida como camada L2 ou MAC, tem como principal objetivo realizar a conexão lógica entre os dispositivos que estiverem trocando informações, além de lidar com funções relacionadas à sincronização, controle de erros e de fluxo, segmentação, sequenciamento e delimitação das unidades de dados. O MAC é responsável pelo preparo dos dados para transmissão e controle de como esses dados acessam o meio físico, enquanto que no destino ele recebe os dados provenientes da camada física para realizar a sua aceitação e processamento.

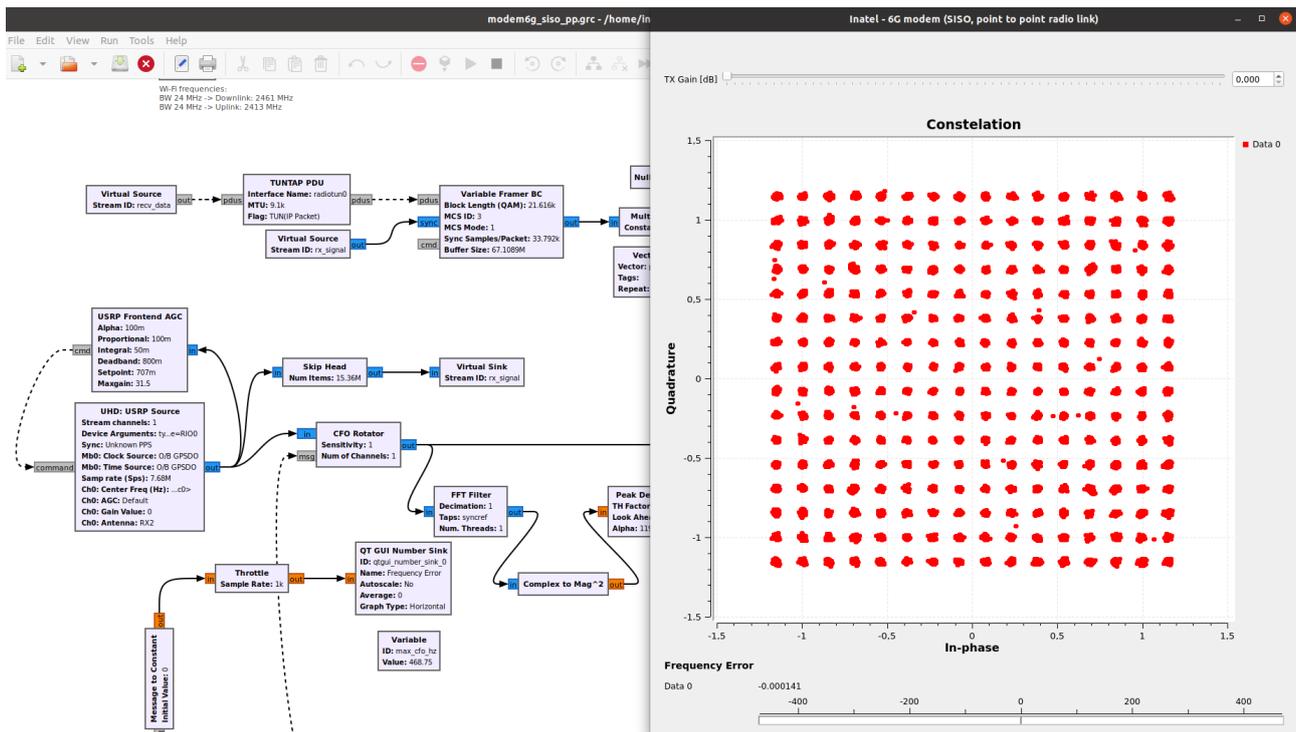


Figura 5: Validação do transceptor Brasil 6G (versão de complexidade reduzida) no novo ambiente de desenvolvimento.

Um dos desafios do projeto Brasil 6G é preparar uma rede de acesso capaz de operar com múltiplos usuários. Neste caso, a camada MAC combinada com a camada PHY dos transceptores possuem papel fundamental para o suporte a um número maior de terminais de usuário. A Figura 6 ilustra um diagrama de um exemplo de composição da rede de acesso Brasil 6G utilizado inicialmente como referência para testes que considera 1 BS e 3 UEs operando com suporte à uma vazão de dados da ordem de 100 Mbps.

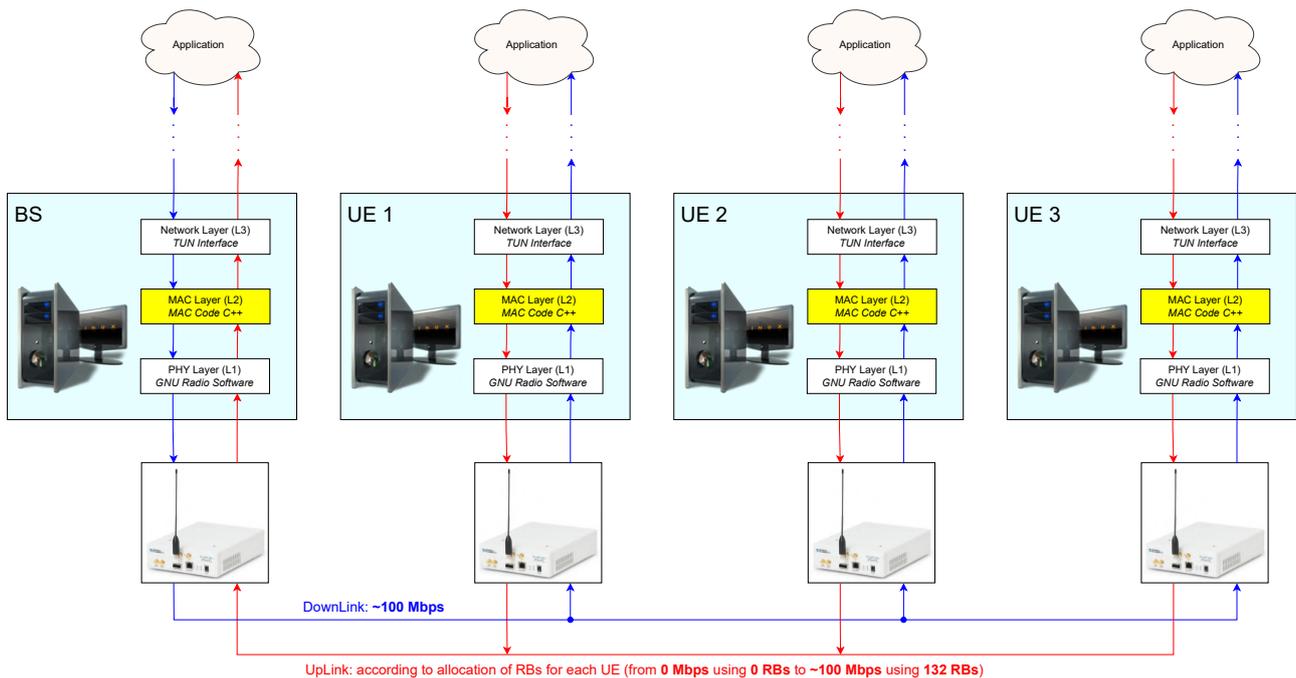


Figura 6: Rede de acesso Brasil 6G utilizada inicialmente como referência para testes.

A partir da rede de referência adotada para testes, foram executados dois testes distintos. O Teste 1 apresenta um cenário com uma BS conectada à 3 UEs usando um código emulador da camada física denominado *StubPHYLayer* para analisar a carga computacional exigida pelo MAC de forma isolada e posteriormente em conjunto com o *software* GNU Radio executando uma versão da camada física do transceptor denominada de *loopback*. Esta versão do transceptor é empregada para testes dos blocos de processamento, porém, sem considerar o uso do *front-end* de RF, ou seja, considera apenas os blocos de processamento em *software* da versão de complexidade reduzida do transceptor. O Teste 2, com o cenário similar ao Teste 1, tem como objetivo verificar problemas no *software* da camada MAC, de modo a apresentar possíveis soluções e otimizações no código para melhorar seu desempenho.

Para o Teste 1, o objetivo é investigar a carga computacional de processamento exigida apenas pelo MAC e seu possível impacto ao combiná-lo posteriormente com os SDRs no PHY. O Teste 2, por sua vez, visa a correção de *bugs* e otimizações no MAC de forma isolada. Os Testes 1 e 2 estão detalhados a seguir.

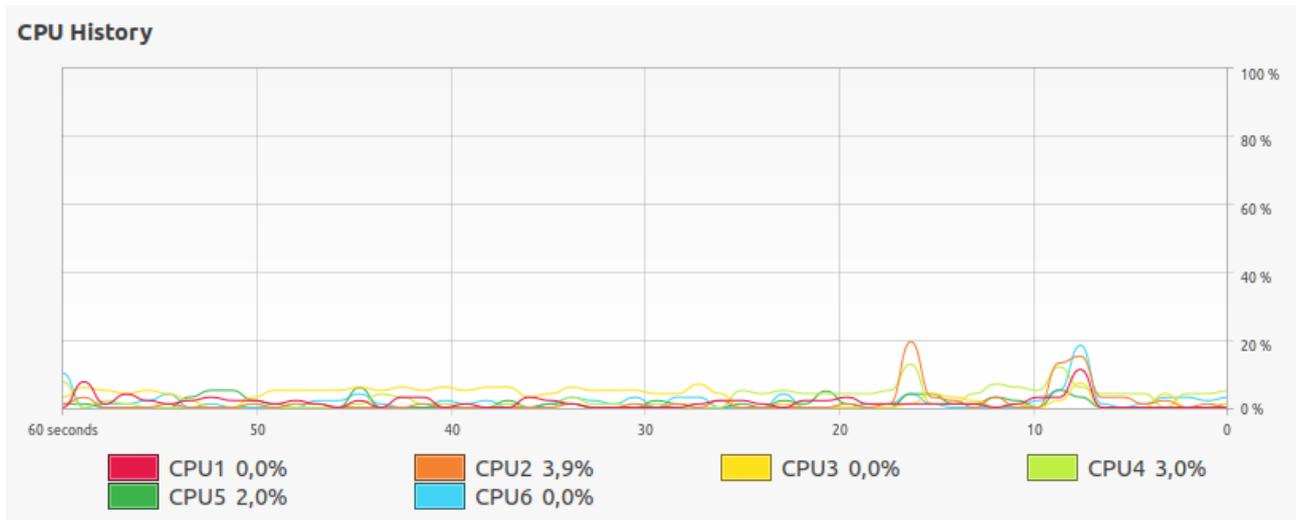
2.2.1 Teste 1: Análise de carga computacional do *software* da camada MAC

A Tabela 2 apresenta as especificações técnicas dos computadores que foram utilizados como BS e UE e que foram utilizados para a análise de exigência de recursos computacionais dada a execução da camada MAC e considerando o Cenário 1 acima descrito.

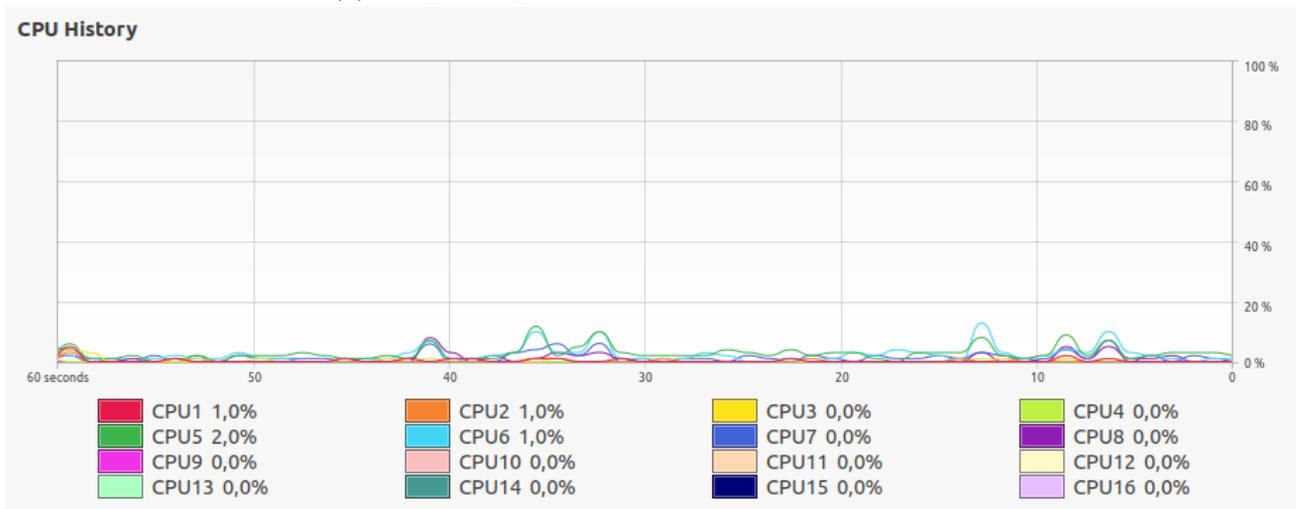
Tabela 2: Especificações técnicas dos computadores alocados para BS e uma UE para análise de carga computacional exigida.

	Processador			RAM	Placa de Vídeo
	Modelo	Núcleos	Threads		
BS	Intel Core i9-11900K 3.50GHz	8	16	64GB	NVIDIA GTX 1050
UE	Intel Core i7-8700 3.20GHz	6	12	16GB	Intel UHD Graphics 630

Baseado nas especificações técnicas dos equipamentos de BS e UE descritos na Tabela 2, foram realizados testes de carga computacional de modo a investigar o impacto do *software* responsável pelo funcionamento da camada MAC. A Figura 7 apresenta a carga computacional exigida pela BS e em uma UE com o *software* da camada MAC operando de forma isolada, considerando 3 equipamentos de usuário conectados à estação rádio-base.



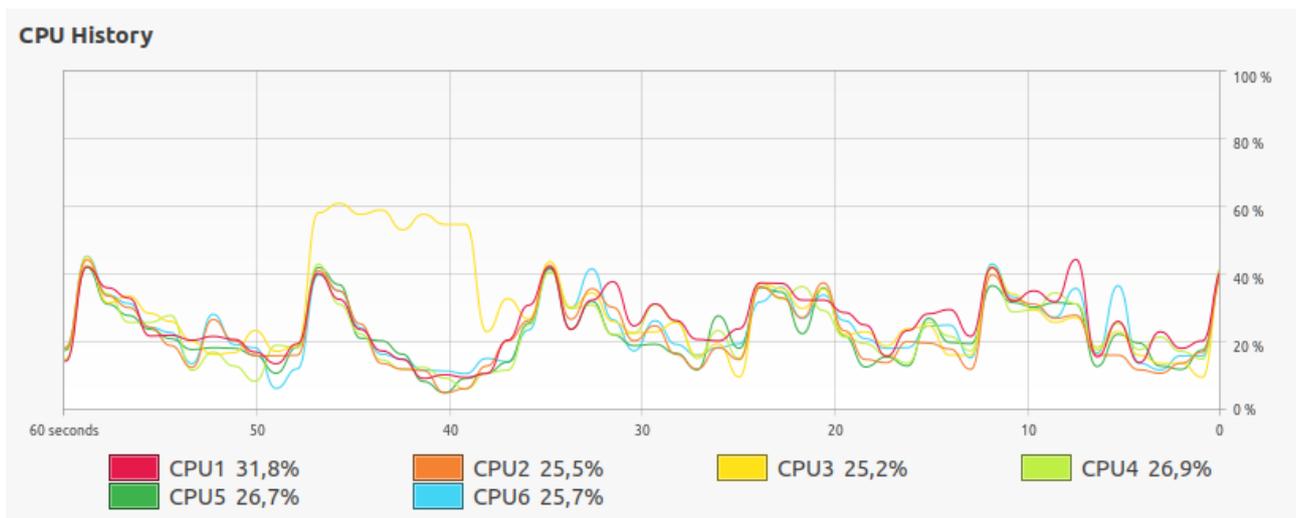
(a) Carga computacional da camada MAC na UE.



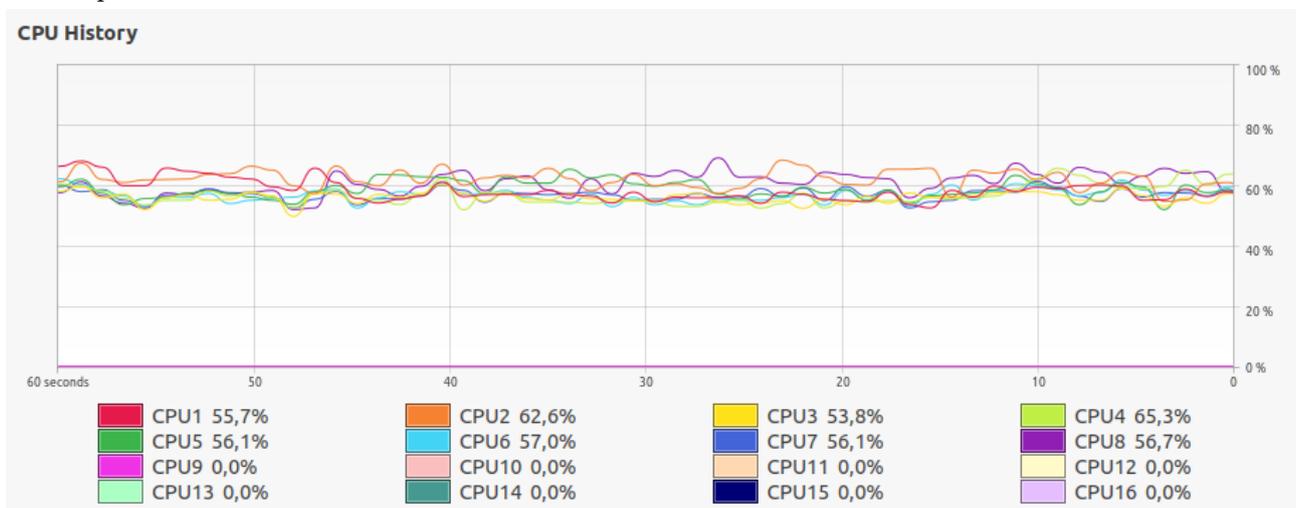
(b) Carga computacional da camada MAC na BS.

Figura 7: Carga computacional com o *software* da camada MAC operando de forma isolada.

Como pode ser observado pela Figura 7, o *software* responsável pela camada MAC por si só não exige uma carga computacional considerável. A Figura 7a apresenta uma porcentagem de aproximadamente 1,48% de uso médio por *Central Processing Unit* (CPU) em uma UE, enquanto a Figura 7b destaca 0,31% de uso médio por CPU na BS. Como meio comparativo, a Figura 8 apresenta os gráficos de carga computacional na BS e UE ao manter o *software* da camada MAC funcional e executar o *software* GNU Radio executando o projeto, descrito anteriormente, denominado de *loopback* do transceptor.



(a) Carga computacional da camada MAC na UE com GNU Radio executando projeto *loopback* do transceptor.



(b) Carga computacional da camada MAC na BS com GNU Radio executando projeto *loopback* do transceptor.

Figura 8: Carga computacional com o *software* da camada MAC operando em conjunto com o *software* GNU Radio executando projeto *loopback* do transmissor.

Através das Figuras 8a e 8b, percebe-se uma porcentagem de uso médio por CPU de aproximadamente 26,97% em uma UE e 28,96% na BS ao executar o *software* GNU Radio com a execução do projeto de *loopback* do transmissor em conjunto com o *software* da camada MAC. Comparado com o MAC operando de forma isolada apresentado na Figura 7, pôde-se observar um aumento de 25,49% do uso médio por CPU na UE, além de um acréscimo de 28,65% no uso médio por CPU na BS. Baseado nestes resultados, pode-se concluir que a carga computacional necessária para a execução do *software* da camada MAC não é significativo, visto que o impacto maior está relacionado ao uso do GNU Radio para a camada física. A Tabela 3 resume os resultados obtidos de carga computacional exigidos pela estação radio-base e equipamentos de usuário, respectivamente.

Tabela 3: Resultados de uso médio por CPU para análise de carga computacional na BS e UE.

Equipamento	Uso Médio por CPU (%)		Aumento (%)
	MAC isolado	MAC + GNU Radio (<i>Loopback</i> do Transceptor)	
BS	0,31	28,96	28,65 (↑)
UE	1,48	26,97	25,49 (↑)

É importante destacar que, considerando o projeto completo da camada física do transceptor, ou seja, considerando a integração com o *front-end* de RF, os valores de carga computacional exigidos dos computadores deve ser superior. Também é válido ressaltar que os computadores considerados para a versão final dos transceptores possui uma capacidade de recursos computacionais, a princípio, superior aos computadores utilizados nos testes apresentados.

2.2.2 Teste 2: Otimizações do MAC com camada física emulada

Para o Teste 2, foi realizado um mapeamento detalhado da camada MAC para identificar *bugs* a serem corrigidos e também limitações no código. A partir deste mapeamento, foi possível definir um conjunto de ações para superar os principais desafios previamente conhecidos e consequentemente otimizar o uso do código da camada MAC para os transceptores da BS e da(s) UE(s).

Através de um estudo detalhado da camada de enlace, pôde-se realizar um mapeamento das principais limitações do código MAC, além de possíveis causas e soluções para estes problemas, conforme apresentado na Tabela 4.

Baseado na Tabela 4, a Limitação 1 (LIM1), referente à estabilidade para transmissão de multi-usuários, é uma das principais limitações da camada MAC. Através dos testes realizados, pôde-se verificar que a inserção de 3 ou mais UEs em um cenário de multi-usuários implica na falha de transmissão da BS para os usuários correspondentes. Como possível causa deste problema, foi identificado o processamento necessário para que o computador da BS seja capaz de realizar esta tarefa.

Com o escalonamento do número de UEs, o código MAC consequentemente necessitará de maior processamento interno para lidar com o tratamento de dados a serem transmitidos para diferentes destinos. Além disso, o código MAC cria uma interface virtual para se comunicar com a camada de rede. Desta forma, os pacotes de rede são transmitidos através de uma interface de rede virtual que permite a comunicação entre o espaço do usuário e o *kernel* do sistema operacional para processamento. Ao contrário de uma interface física diretamente conectada a dispositivos de rede físicos, como placas de rede ou adaptadores *Ethernet*, esta

Tabela 4: Limitações da Camada MAC.

Acrônimo	Limitação	Descrição	Causa	Solução
LIM1	Estabilidade para transmissão de multi-usuários.	Falha de transmissão da BS em um cenário de multi-usuários com 3 ou mais UEs.	Carga computacional necessária para processamento do código MAC para multi-usuários e dos pacotes de rede a serem transmitidos por uma interface virtual.	Aumento da capacidade computacional de processamento do computador da BS.
LIM2	Transmissão de <i>Downlink</i> para multi-usuários de forma simultânea.	Para 2 ou mais UEs, a vazão máxima da rede é direcionada para uma única UE, não havendo transmissão simultânea no <i>Downlink</i> .	Problema na seção do código MAC de Scheduling, relacionada à alocação de recursos para cada uma das UEs.	Correção do algoritmo de Scheduling para a alocação de recursos das UEs.
LIM3	Taxa de transferência de 100 Mbps distribuída para as UEs.	A vazão para multi-usuários não é realizada de forma distribuída para alcançar a vazão máxima aproximada da rede de 100 Mbps.	Processamento e <i>bug</i> na função de alocação de recursos do código MAC (causas similares às limitações LIM1 e LIM2).	Aumento da capacidade de processamento da BS e correção de função de alocação de recursos do código MAC (soluções similares para as limitações LIM1 e LIM2).
LIM4	Problemas de roteamento de rede.	A BS fornece acesso à Internet para as UEs, mas não fornece conectividade para outras redes, como por exemplo o Core.	Problemas de roteamento e de <i>Firewall</i> .	Correção no script do código MAC para realizar o roteamento para a rede desejada, além de comandos de permissão no <i>Firewall</i> para autorizar o tráfego de dados na rede.

interface *Tunnel Software Network Interface* (TUN) opera no nível de *software* e pode exigir maior processamento devido a alguns fatores, como citados a seguir:

- i Processamento no espaço do usuário: Como a interface TUN permite a comunicação entre o espaço do usuário e o *kernel*, os pacotes de rede são enviados para o espaço do usuário para processamento adicional. Isso pode envolver operações como análise de cabeçalhos, filtragem de pacotes, criptografia/descriptografia, entre outros. O processamento adicional realizado no espaço do usuário requer recursos da CPU, o que pode aumentar a carga de processamento em comparação com uma interface física que realiza parte desse processamento no hardware da placa de rede.
- ii Encapsulamento e desencapsulamento de pacotes: A interface TUN encapsula pacotes de rede em estruturas de dados adequadas para transmissão no espaço do usuário. Isso envolve operações adicionais, como cópia dos dados dos pacotes para o espaço do *kernel* e vice-versa. Essas operações de encapsulamento e desencapsulamento de pacotes podem consumir recursos adicionais da CPU.
- iii Sobrecarga de chamadas de sistema: A interface TUN faz uso extensivo de chamadas de sistema para transferir pacotes entre o espaço do usuário e o *kernel*. Cada chamada de sistema envolve uma troca de contexto entre o espaço do usuário e o *kernel*, o que pode adicionar sobrecarga ao processamento dos pacotes.

Visto que o processamento dos pacotes de rede requer recursos da CPU, o aumento do número de UEs implica conseqüentemente na necessidade de maior carga de processamento e este pode ser um fator limitante de escala. Para mitigar este problema, foram realizados testes substituindo o computador da BS por um computador com maior capacidade computacional e de processamento. Nos testes iniciais, o computador da BS tinha especificações técnicas

similares aos computadores das UEs, previamente mapeados na Tabela 2. Após a substituição do computador da BS por um equipamento de maior capacidade computacional, conforme apresentado também na Tabela 2, pôde-se verificar que a estação rádio-base é capaz de realizar a transmissão de dados para 3 UEs simultaneamente de forma estável.

Em relação à Limitação 2 (LIM2), referente à transmissão de *downlink* para multi-usuários de forma simultânea, pôde-se verificar que a BS estava direcionando sua vazão máxima para uma única UE. Desta forma, era necessário o término de transmissão deste equipamento de usuário para que a BS pudesse alocar recursos para transmitir para outra UE, inviabilizando a transmissão de *downlink* para multi-usuários de forma simultânea. Para solucionar este problema, foram detectadas inconsistências no bloco de código do MAC responsável pela função de *Scheduling*, que por sua vez tem como objetivo realizar a alocação de recursos para as UEs disponíveis na rede. Após esta correção, novos testes de *downlink* foram realizados, e os resultados mostraram que a BS foi capaz de atender até 3 UEs de forma simultânea, dividindo sua taxa de transferência máxima entre os usuários da rede de acordo com a demanda de *bytes* a serem transmitidos. Apesar de atender a multi-usuários, a alocação de recursos ainda apresenta variações quanto à distribuição de vazão para as UEs, dado que o fluxo de dados recebidos da camada de rede pelo código MAC é diferente do fluxo de dados enviados à camada física.

Com as soluções apresentadas para as limitações previamente descritas (LIM1 e LIM2), foi possível solucionar consequentemente a Limitação 3 (LIM3), na qual a BS não era capaz de disponibilizar uma vazão mínima de 100 Mbps de forma distribuída para as UEs, tendo uma taxa de transferência oscilante. A partir destas correções, o uso dos transceptores foi capaz de atender multi-usuários com a vazão mínima distribuída desejável.

Por fim, a Limitação 4 (LIM4) refere-se à problemas de roteamento de rede. Em testes iniciais, a BS tinha como objetivo fornecer acesso à Internet para UEs localizadas em áreas remotas sem conectividade de forma direta. Tendo em vista que o Projeto Brasil 6G propõe o uso de um núcleo de rede capaz de realizar, por exemplo, o gerenciamento das UEs e fornecer conectividade à redes de acordo com a demanda do usuário, é necessário que a BS encaminhe o tráfego de dados para o núcleo. Inicialmente, problemas foram encontrados devido à permissões de *Firewall* e roteamento adequado. Após essas correções, um computador foi conectado à BS para simular o núcleo e testes foram realizados com a nova rede implementada. Como resultado, o novo dispositivo atrelado à BS foi capaz de fornecer conectividade à Internet para a UE, solucionando os problemas relacionados ao roteamento e à permissões de tráfego de dados.

O *setup* para a realização dos testes previamente descritos está ilustrado na Figura 9. Para tal, foi considerado um cenário com um computador configurado para operar como BS e 3 computadores configurados como UEs. O código *StubPHY* foi utilizado para emular a camada física e permitir os testes de desempenho da camada MAC em uma rede cabeada conectada através de um *switch*.



Figura 9: *Setup* para cenário de testes de desempenho da Camada MAC.

Para medir a largura de banda e avaliar a qualidade de conexão da rede, foi utilizada a ferramenta Linux chamada *iperf3*, cujo objetivo é realizar testes de velocidade de transferência de dados entre dois ou mais dispositivos. Para os testes de *uplink*, cada UE foi configurada de modo a obter um quarto da alocação máxima de recursos. Tendo em vista o cenário de referência adotado para os testes, que considera o suporte a uma taxa total de aproximadamente de 100 Mbps, os resultados obtidos a partir desta configuração deveriam ser próximos à uma taxa de 25 Mbps para cada UE. Através da Figura 10 ilustrada abaixo, podemos verificar que o teste de *uplink* alcançou resultados próximos ao esperado.

```

[CoreL1] Valor de socketout: 0
[CoreL1] Valor de numberSent: 58
[CoreL1] Pdu sent:58 bytes.
[CoreL1] Valor de socketout: 1
[CoreL1] Valor de numberSent: 58
[CoreL1] Pdu sent:58 bytes.
[CoreL1] Valor de socketout: 2
[CoreL1] Valor de numberSent: 58
[CoreL1] Pdu sent:58 bytes.
[CoreL1] Received SubFrameTx.End message.
[CoreL1] PDU with size 607 received.
[CoreL1] Valor de sizeDas recebido: 597
[CoreL1] Received BSSubFrameTx.Start message. Receiving PDU from L2...
[CoreL1] Valor do tamanho do buffer recebido de L2: 133
[CoreL1] Allocation: First RB: 0 | Number RB: 1
[CoreL1] Valor de socketout: 0
[CoreL1] Valor de numberSent: 58
[CoreL1] Pdu sent:58 bytes.
[CoreL1] Valor de socketout: 1
[CoreL1] Valor de numberSent: 58
[CoreL1] Pdu sent:58 bytes.

[Scheduler] #5$S rbsAllocation[ueOffset]: 0
[Scheduler] #5$S ueOffset: 0 target_ue_id 2 all actions-back() First RB: 0
[Scheduler] Scheduled 398 bytes for PDU 0 destined to UE 2
[SduBuffers] Got SDU from L3 queue.
[Multiplexer] Multiplexed 1 SDUs into PDU.
[Multiplexer] MAC Header inserted.
[MACCTRL] #5# currentParameters->getNumberUEs(): 3
[MACCTRL] #5# macPdu.size(): 1
[MACCTRL] #5# currentParameters->getFLUTMatrix(): 15
[MACCTRL] #5# currentParameters->getNumLogLog(): 0
[MACCTRL] #5# currentParameters->getRWMetricsPeriodicity(): 7
[TransmissionProtocol] Sending packet to L1.
[L1L2Interface] Pdu sent:133 bytes.
[ProtocolControl] Receiving PDU from L1...
[ReceptionProtocol] Receiving packet from L1.
[MacController] Decoding MAC Address 2: in progress...
[MacController] #5# getSnrConvertToMcs: 22
[MacController] #5# getMcsUplink(macAddress): 22
[Multiplexer] MAC Header removed successfully.
[Multiplexer] Demultiplexed SDU 1
[ProtocolControl] RWMetrics from UE 2 received: FL

root@LCRR-048637-R:/home/inatel-crr# iperf3 -s -p 5201
Server listening on 5201
-----
Accepted connection from 10.0.0.11, port 52658
[ ID] Interval      Transfer      Bitrate
[ 5] 0.00-1.00    sec  2.59 MBytes  21.7 Mbts/sec
[ 5] 1.00-2.00    sec  2.70 MBytes  22.6 Mbts/sec
[ 5] 2.00-3.00    sec  2.70 MBytes  22.6 Mbts/sec
[ 5] 3.00-4.00    sec  2.71 MBytes  22.7 Mbts/sec
[ 5] 4.00-5.00    sec  2.70 MBytes  22.6 Mbts/sec
[ 5] 5.00-6.00    sec  2.70 MBytes  22.6 Mbts/sec
[ 5] 6.00-7.00    sec  2.70 MBytes  22.6 Mbts/sec
[ 5] 7.00-8.00    sec  2.71 MBytes  22.7 Mbts/sec
[ 5] 8.00-9.00    sec  2.70 MBytes  22.6 Mbts/sec
[ 5] 9.00-10.00   sec  2.70 MBytes  22.6 Mbts/sec
[ 5] 10.00-10.57  sec  1.55 MBytes  22.7 Mbts/sec
-----
[ ID] Interval      Transfer      Bitrate
[ 5] 0.00-10.57   sec  28.4 MBytes  22.6 Mbts/sec
-----
receiver

root@LCRR-048637-R:/home/inatel-crr# iperf3 -s -p 5202
Server listening on 5202
-----
Accepted connection from 10.0.0.12, port 37718
[ ID] Interval      Transfer      Bitrate
[ 5] 0.00-1.00    sec  2.60 MBytes  21.8 Mbts/sec
[ 5] 1.00-2.00    sec  2.70 MBytes  22.6 Mbts/sec
[ 5] 2.00-3.00    sec  2.70 MBytes  22.6 Mbts/sec
[ 5] 3.00-4.00    sec  2.70 MBytes  22.6 Mbts/sec
[ 5] 4.00-5.00    sec  2.70 MBytes  22.6 Mbts/sec
[ 5] 5.00-6.00    sec  2.71 MBytes  22.7 Mbts/sec
[ 5] 6.00-7.00    sec  2.70 MBytes  22.6 Mbts/sec
[ 5] 7.00-8.00    sec  2.70 MBytes  22.6 Mbts/sec
[ 5] 8.00-9.00    sec  2.70 MBytes  22.6 Mbts/sec
[ 5] 9.00-10.00   sec  2.71 MBytes  22.7 Mbts/sec
[ 5] 10.00-10.56  sec  1.51 MBytes  22.5 Mbts/sec
-----
[ ID] Interval      Transfer      Bitrate
[ 5] 0.00-10.56   sec  28.4 MBytes  22.6 Mbts/sec
-----
receiver

root@LCRR-048637-R:/home/inatel-crr# iperf3 -s -p 5203
Server listening on 5203
-----
Accepted connection from 10.0.0.13, port 43262
[ ID] Interval      Transfer      Bitrate
[ 5] 0.00-1.00    sec  2.59 MBytes  21.7 Mbts/sec
[ 5] 1.00-2.00    sec  2.70 MBytes  22.6 Mbts/sec
[ 5] 2.00-3.00    sec  2.70 MBytes  22.6 Mbts/sec
[ 5] 3.00-4.00    sec  2.71 MBytes  22.7 Mbts/sec
[ 5] 4.00-5.00    sec  2.70 MBytes  22.6 Mbts/sec
[ 5] 5.00-6.00    sec  2.70 MBytes  22.6 Mbts/sec
[ 5] 6.00-7.00    sec  2.70 MBytes  22.6 Mbts/sec
[ 5] 7.00-8.00    sec  2.71 MBytes  22.7 Mbts/sec
[ 5] 8.00-9.00    sec  2.70 MBytes  22.6 Mbts/sec
[ 5] 9.00-10.00   sec  2.70 MBytes  22.6 Mbts/sec
[ 5] 10.00-10.57  sec  1.55 MBytes  22.8 Mbts/sec
-----
[ ID] Interval      Transfer      Bitrate
[ 5] 0.00-10.57   sec  28.4 MBytes  22.6 Mbts/sec
-----
receiver

```

Figura 10: Testes de *uplink* para avaliação de desempenho da taxa de transferência da rede.

Já para os testes de *downlink*, a BS deve alocar seus recursos dinamicamente de acordo com a quantidade de UEs funcionais na rede. Inicialmente, foi solicitada a conexão apenas com a UE 1, conforme ilustrado pela Figura 11.

```

[CoreL1] Valor de socketout: 0
[CoreL1] Valor de numberSent: 58582
[CoreL1] Pdu sent:58582 bytes.
[CoreL1] Valor de socketout: 1
[CoreL1] Valor de numberSent: 58582
[CoreL1] Pdu sent:58582 bytes.
[CoreL1] Valor de socketout: 2
[CoreL1] Valor de numberSent: 58582
[CoreL1] Pdu sent:58582 bytes.
[CoreL1] Received SubframeTx.End message.
[CoreL1] PDU with size 1033 received.
[CoreL1] Valor de sizeRxas recebido: 1585
[CoreL1] Received BSSubframeTx.Start message. Receiving PDU from L2...
[CoreL1] Valor do tamanho do buffer recebido de L2 : 58657
[CoreL1] Allocation: First RB: 0 | Number RB: 132
[CoreL1] Valor de socketout: 0
[CoreL1] Valor de numberSent: 58582
[CoreL1] Pdu sent:58582 bytes.
[CoreL1] Valor de socketout: 1
[CoreL1] Valor de numberSent: 58582
[CoreL1] Pdu sent:58582 bytes.

[Multiplexer] Demultiplexed SDU 14
[MacController] Data SDU received. Forwarding to L3.
[TransmssionProtocol] Sending packet to L3.
[Multiplexer] Demultiplexed SDU 15
[MacController] Data SDU received. Forwarding to L3.
[TransmssionProtocol] Sending packet to L3.
[Multiplexer] Demultiplexed SDU 16
[MacController] Data SDU received. Forwarding to L3.
[TransmssionProtocol] Sending packet to L3.
[Multiplexer] Demultiplexed SDU 17
[MacController] Data SDU received. Forwarding to L3.
[TransmssionProtocol] Sending packet to L3.
[Multiplexer] Demultiplexed SDU 18
[MacController] Data SDU received. Forwarding to L3.
[TransmssionProtocol] Sending packet to L3.
[Multiplexer] Demultiplexed SDU 19
[MacController] Data SDU received. Forwarding to L3.
[TransmssionProtocol] Sending packet to L3.
[Multiplexer] Demultiplexed SDU 20
[MacController] Data SDU received. Forwarding to L3.
[TransmssionProtocol] Sending packet to L3.

root@LCRR-048637-R:/home/lnatel-crr# iperf3 -c 10.0.0.11 -p 5201 -t 0
Connecting to host 10.0.0.11, port 5201
[ 5] Local 10.0.0.10 port 60478 connected to 10.0.0.11 port 5201
[ ID] Interval      Transfer      Bitrate      Retr      Cwnd
[ 5] 0.00-1.00    sec  14.3 MBytes  120 Mbts/sec  0      881 KBytes
[ 5] 1.00-2.00    sec  12.5 MBytes  105 Mbts/sec  0      1.45 MBytes
[ 5] 2.00-3.00    sec  11.2 MBytes  94.4 Mbts/sec  0      2.03 MBytes
[ 5] 3.00-4.00    sec  12.2 MBytes  94.4 Mbts/sec  0      2.61 MBytes
[ 5] 4.00-5.00    sec  12.5 MBytes  105 Mbts/sec  0      3.03 MBytes
[ 5] 5.00-6.00    sec  11.2 MBytes  94.4 Mbts/sec  0      3.03 MBytes
[ 5] 6.00-7.00    sec  11.2 MBytes  94.4 Mbts/sec  0      3.03 MBytes
[ 5] 7.00-8.00    sec  12.5 MBytes  105 Mbts/sec  0      3.03 MBytes
[ 5] 8.00-9.00    sec  11.2 MBytes  94.4 Mbts/sec  0      3.03 MBytes
[ 5] 9.00-10.00   sec  11.2 MBytes  94.4 Mbts/sec  0      3.03 MBytes
[ 5] 10.00-11.00  sec  12.5 MBytes  105 Mbts/sec  0      3.03 MBytes
[ 5] 11.00-12.00  sec  11.2 MBytes  94.4 Mbts/sec  0      3.03 MBytes
[ 5] 12.00-13.00  sec  11.2 MBytes  94.4 Mbts/sec  0      3.03 MBytes
[ 5] 13.00-14.00  sec  12.5 MBytes  105 Mbts/sec  0      3.03 MBytes
[ 5] 14.00-15.00  sec  11.2 MBytes  94.4 Mbts/sec  0      3.03 MBytes
[ 5] 15.00-16.00  sec  11.2 MBytes  94.4 Mbts/sec  0      3.03 MBytes
[ 5] 16.00-17.00  sec  12.5 MBytes  105 Mbts/sec  0      3.03 MBytes
[ 5] 17.00-18.00  sec  11.2 MBytes  94.4 Mbts/sec  0      3.03 MBytes
[ 5] 18.00-19.00  sec  11.2 MBytes  94.4 Mbts/sec  0      3.03 MBytes
[ 5] 19.00-20.00  sec  12.5 MBytes  105 Mbts/sec  0      3.03 MBytes
[ 5] 20.00-21.00  sec  11.2 MBytes  94.4 Mbts/sec  0      3.03 MBytes
[ 5] 21.00-22.00  sec  12.5 MBytes  105 Mbts/sec  0      3.03 MBytes

```

Figura 11: Testes de *downlink* para avaliação de desempenho da taxa de transferência considerando 1 UE conectada à rede.

Conforme pode ser visto pela Figura 11, podemos observar que a taxa de transferência atingiu o valor próximo à taxa máxima esperada de 100 Mbps.

Ao conectar a UE 2, a taxa de transferência para cada uma das UEs foi reajustada pela metade da taxa máxima aproximada de 100 Mbps, resultando em uma vazão de 50Mbps para as UEs 1 e 2. A Figura 12 ilustra o ajuste de vazão considerando 2 UEs conectadas à rede.

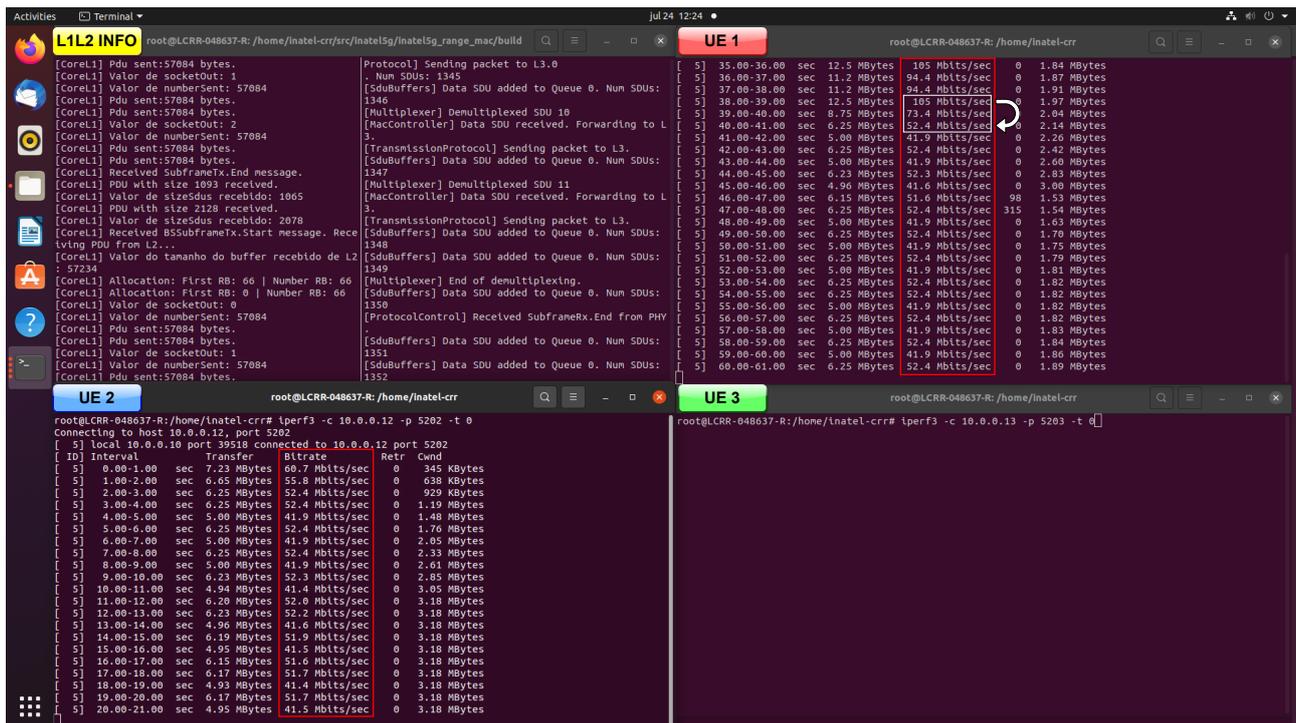


Figura 12: Testes de *downlink* para avaliação de desempenho da taxa de transferência considerando 2 UEs conectadas à rede.

Por fim, ao conectar a UE 3 na rede, podemos observar na Figura 13 que, através dos testes realizados usando a ferramenta *iperf3*, a vazão máxima esperada de 100 Mbps foi uniformemente distribuída entre as 3 UEs, resultando em uma taxa de transferência aproximada de 33 Mbps para cada equipamento de usuário.

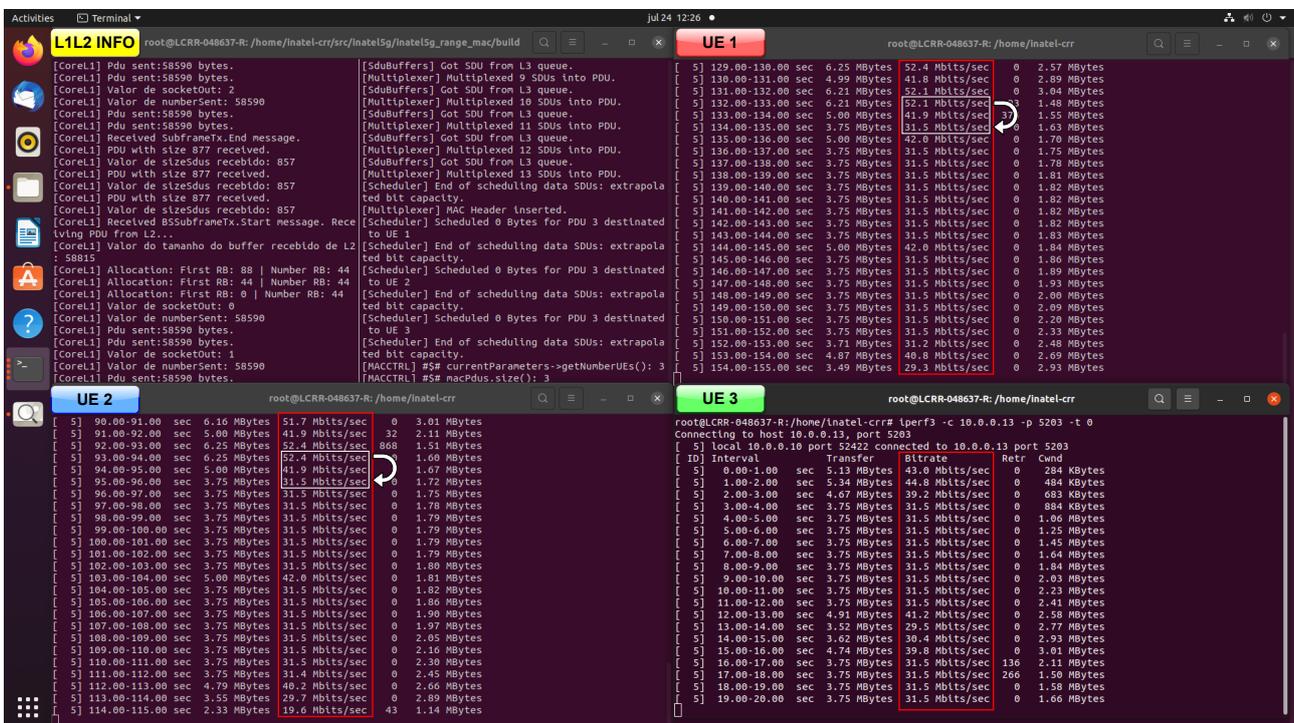


Figura 13: Testes de *downlink* para avaliação de desempenho da taxa de transferência considerando 3 UEs conectadas à rede.

3 Integração de ferramentas de I.A. na camada física

João Henrique Silva Delfino

O projeto Brasil 6G tem como um de seus objetivos a concepção, desenvolvimento e otimização de uma rede de acesso, conforme já mencionado. Foi destacado anteriormente que as tecnologias associadas à IA são elencadas como habilitadoras para a composição da rede 6G. A presente seção apresenta o resultado de pesquisas, desenvolvimento e testes considerando a integração de IA na camada PHY do transceptor Brasil 6G, que é o dispositivo principal da rede de acesso Brasil 6G.

Ao longo da execução das atividades do projeto, foram realizadas otimizações na versão já desenvolvida do Transceptor Brasil 6G. O objetivos destas otimizações visam, dentre outros aspectos, proporcionar um melhor suporte à diferentes aplicações, bem como prover uma melhor experiência para os usuários. Para isso, o monitoramento e análise dos parâmetros do canal de comunicação torna-se primordial para a oferta de serviços confiáveis e precisos. Por conseguinte, os estudos acerca dos chamados enlaces adaptativos se potencializam ainda mais, tornando-se cruciais para se assegurar a *Quality of Service* (QoS) fornecida. Estes parâmetros contribuem diretamente para a definição das ações adaptativas, fornecendo alguma métrica relacionada a qualidade do canal e possibilitam estimar a influência do mesmo na transmissão e recepção dos dados. Dito isso, o MCS define condições de operação do enlace de comunicação relacionados à modulação e codificação, que são definidos visando otimizar taxa de transmissão de dados e minimizar a *Bit Error Rate* (BER). Para a definição do MCS, algum critério deve ser utilizado como base, sendo os parâmetros de qualidade de transmissão necessários para tal questão.

A relação sinal-ruído, SNR, é um exemplo, o qual relaciona de forma direta a potência do sinal e a do ruído, determinando a influência desta interferência no canal de comunicação. Logo, a estimativa de tal parâmetro torna-se fundamental no contexto de enlaces adaptativos, podendo ser utilizado na escolha do MCS. Diversas abordagens referente a estimativa de SNR podem ser encontradas na literatura, como é o caso da *Normalized Mean Squared Error* (NMSE)[5]. Além de algoritmos básicos como a NMSE, a aplicação de IA tem se tornado uma opção cada vez mais plausível. As aplicações de IA são viabilizadas, principalmente, em contextos cuja a geração de conjuntos de dados é possível, os quais podem ser utilizados para o treinamento de modelos de *Machine Learning* (ML) e poderão atuar na inferência de parâmetros, alocação de recursos, entre outros casos de aplicação.

Visando a refinar a estimativa da SNR, foi feita a integração de uma técnica mais precisa e efetiva na camada física do transceptor Brasil 6G, a qual utiliza *Deep Learning* (DL), sub-área de ML. A técnica inovadora, concebida ao longo da execução do projeto Brasil 6G, apresentada com detalhes em [6], é intitulada de LDNN-SNRE. A implementação e integração do referido algoritmo na camada PHY do transceptor Brasil 6G ocorreu em três etapas: verificação dos recursos de computacionais demandados para sua execução, implementação e integração do algoritmo na camada física do transceptor Brasil 6G, sendo empregada inicialmente a versão de complexidade reduzida do transceptor, e a realização de testes e análise de desempenho da técnica já integrada no transceptor. O desenvolvimento das etapas mencionadas, bem como os resultados alcançados, encontram-se detalhadas a seguir.

3.1 Recursos computacionais utilizados

Os recursos computacionais inicialmente previstos para o desenvolvimento e testes de aplicações de IA integradas ao transceptor Brasil 6G foram apresentados no relatório 2.1[7]. O

ambiente de *software* utilizado em tal desenvolvimento considera o uso do sistema operacional Ubuntu 20.04 (Focal Fossa) com *kernel* na versão 5.15 (necessário 5.13+), em virtude da compatibilidade necessária para com a placa de vídeo RTX 3070 TI. Sua utilização como uma unidade gráfica de processamento geral garante que o treinamento e inferência de modelos de ML sejam feitos com alta performance. Sendo assim, a aplicação de IA integrada ao transceptor introduz a necessidade de utilização de computadores com recursos de *hardware* que possibilitam executar aplicações de alto desempenho. A Tabela 5 listados os recursos de *hardware* empregados para a realização do desenvolvimento:

Tabela 5: Descrição de *Hardware*

Dispositivo de <i>Hardware</i>	Descrição
CPU	Processador Intel® Xeon® Gold 5220
GPU	Nvidia 3070 TI
RAM	64 GB

Além dos recursos acima citados, a fim de integrar o algoritmo de IA ao transceptor, são necessários recursos de *software* que habilitam o treinamento e inferência do modelo por meio da placa de vídeo como recurso de aceleração. Dito isso, para uso da *Graphics Processing Unit* (GPU), alguns *softwares* foram necessários conforme mostrado na Tabela 6.

Tabela 6: Descrição de *Software*

Recursos de <i>software</i>	Versão	Aquisição
Nvidia Driver	v510.47	https://www.nvidia.com/Download/driverResults.aspx/186156
Cuda	v11.7	https://developer.nvidia.com/cuda-downloads
Cudnn	v8.5	https://developer.nvidia.com/cudnn

Ainda no contexto de ferramentas de *softwares* necessárias para a realização das simulações e treinamentos do modelo de DL utilizando a linguagem Python, pode-se destacar que foram utilizadas algumas bibliotecas. A Tabela 7 apresenta tais bibliotecas, bem como a versão e uma breve descrição de cada uma delas.

Tabela 7: Bibliotecas

Bibliotecas	Versão	Descrição
Numpy	v1.21.5	Operação com <i>arrays</i>
Matplotlib	v3.1.2	Criação e visualização de gráficos
Numba	v0.55.1	Compilação <i>Just-In-Time</i> (JIT) de funções Python
Tensorflow	v2.8.0	Criação e treinamento de modelos de DL

3.2 Técnicas de Estimação de SNR

Como dito previamente, os parâmetros empregados nos enlaces da rede de acesso pode ser adaptados a partir de informações da condição do canal de comunicação. Nessa seção será apresentada a técnica inicialmente utilizada no transceptor, bem como uma abordagem inovadora de implementação que é baseada no uso de IA.

3.2.1 Estimador baseado em NMSE

O transceptor Brasil 6G de complexidade reduzida faz uso da técnica NMSE, a qual possibilita estimar a SNR do canal de comunicação e definir o modo de operação do enlace. Para isso, durante a etapa de demodulação QAM, tal métrica é aplicada por meio de um cálculo iterativo a todos os os k símbolos recebidos. A equação abaixo apresenta o funcionamento deste algoritmo:

$$\chi(m) = -10 \cdot \log \left[\frac{\sum_{i=0}^{k-1} (\hat{m}_i - m_i)^2}{\sum_{i=0}^{k-1} \hat{m}_i^2} \right] \text{ dB} \quad (1)$$

onde χ é o valor da SNR estimada com base no NMSE, sendo este calculado, por sua vez, em função de m , conjunto dos símbolos recebidos, e de \hat{m} , os símbolos estimados após a demodulação QAM com decisão abrupta (*hard decision*).

3.2.2 Estimador baseado em LDNN-SNRE

O algoritmo inovador denominado de LDNN-SNRE implementa um modelo de DL cujo objetivo é inferir a correção necessária de modo a estimar o valor de SNR com maior precisão. Dado o valor computado por uma função *Mean Squared Error* (MSE) modificada, o modelo estima tal correção. A fim de integrá-lo no transceptor Brasil 6G, realizou-se todo o processo de criação, treinamento e validação do modelo, além da implementação da função MSE proposta. A Figura 14 ilustra o diagrama com as etapas principais de processamento previstos na técnica, indicando a soma entre a saída da rede neural e o valor fornecido pela função, de modo a estimar a SNR do canal de comunicação.

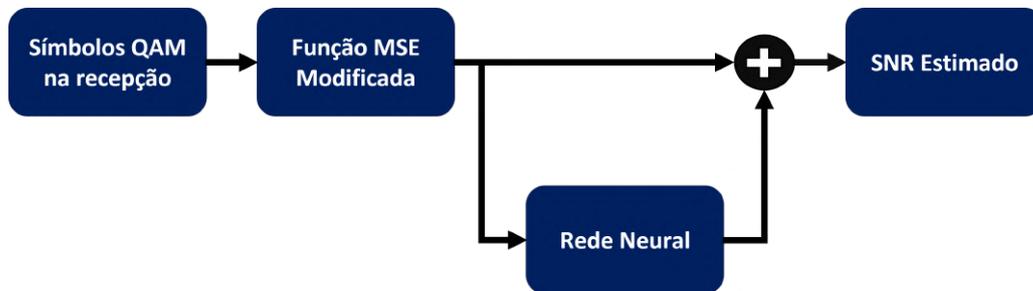


Figura 14: Diagrama do algoritmo LDNN-SNRE.

3.3 Implementação da técnica baseada em IA

Com intuito de viabilizar a integração no transceptor Brasil 6G da técnica do estimador baseado em LDNN-SNRE, foi utilizada, inicialmente, a versão de complexidade reduzida do transceptor. Isto se deve ao fato desta versão em específico facilitar as análises e testes iniciais, já que a mesma possui um número menor de etapas de processamento, funcionalidades e variáveis. De qualquer forma, os resultados alcançados em tal versão podem servir de base para a posterior adaptação e integração à versão de alta capacidade do transceptor. A seguir, estão apresentados maiores detalhes sobre o processo de integração e testes da técnica de estimação de SNR no transceptor Brasil 6G.

3.3.1 Etapas principais da Camada física do transceptor utilizadas para integração e testes

As etapas principais que compõe a camada física do transceptor de complexidade reduzida, cujas etapas foram detalhadas no relatório 2.1 [7], foi utilizada como referência para integração e verificação de desempenho da técnica estimação de SNR baseada em IA. A Figura 15 apresenta as etapas principais de processamento necessárias para compôr tal versão e que são necessárias para a comunicação entre dois dispositivos, como uma BS e uma UE.

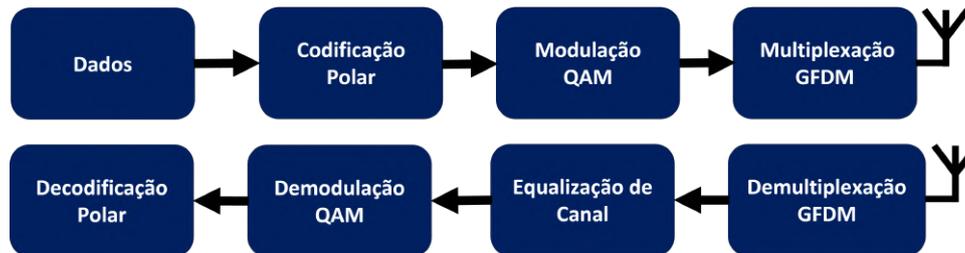


Figura 15: Cadeia de comunicação Ponto-a-Ponto.

3.3.2 Função MSE modificada

A função MSE modificada, apresentada e detalhada em [6], utiliza como base o cálculo de MSE dos símbolos QAM considerando a demodulação com decisão abrupta. A função emprega fatores complementares a fim de mitigar os erros que ocorrem quando as componentes reais e imaginárias dos símbolos recebidos ultrapassam os limites de decisão. Este limite é dado por $d_{min}/2$, onde d_{min} é a menor distância entre dois símbolos adjacentes no diagrama da constelação. A ultrapassagem desses limites somente será detectada quando o módulo das coordenadas de um símbolo recebido exceder o limite máximo dado por $S_{max} + d_{min}/2$, em que S_{max} é o máximo valor, em módulo, que as componentes reais e imaginárias podem assumir na constelação em questão.

A ultrapassagem do limite de decisão só é explicitamente identificável quando um símbolo recebido assume valores além da extremidade da constelação, ou seja, valores superiores a $S_{max} + d_{min}/2$. Dessa maneira, um fator ponderador é introduzido ao processo de tal forma a incluir, amostral e proporcionalmente, as referidas ultrapassagens internas não detectáveis, ou seja, que transcorrem dentro do limite máximo da constelação, mas que não demonstram alguma certeza sobre sua ocorrência. Tal situação é compreendida, assumindo-se que a ocorrência de um evento externo determina uma alta variância do ruído.

3.3.3 Amostragem, pré-processamento e geração do *Dataset*

As técnicas de IA baseadas em DL necessitam de conjuntos de dados representativos do problema a ser solucionado. Dessa maneira, as etapas de processamento da camada física apresentadas na subseção 3.3.1 foram utilizadas para geração dos dados necessários para o treinamento da rede neural. A fim de coletar tais informações, realizou-se uma simulação no próprio *software* GNU Radio [4], cujo objetivo era mapear o valores medidos pela função MSE modificada para cada valor de SNR configurada para o canal de comunicação. A coleta dos dados ocorre em um processo iterativo após a etapa de equalização para cada uma das modulações: *Quadrature Phase Shift Keying* (QPSK), 16-QAM, 64-QAM e 256-QAM. A

Figura 16 apresenta o diagrama do processo de amostragem, a qual foi feita em uma faixa de -40 a 40 dB em passos de $0,1$ dB.

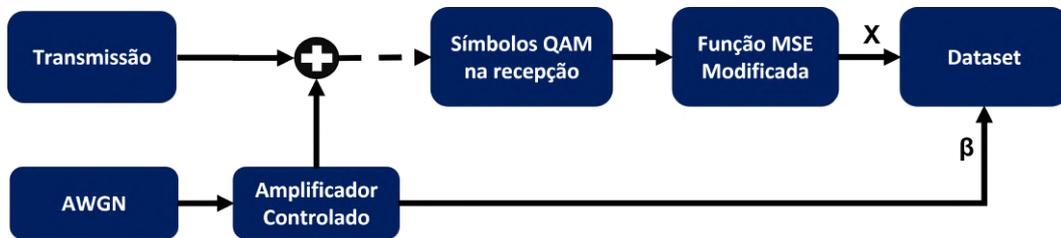


Figura 16: Amostragem e geração do *Dataset*.

Finalizado o processo de amostragem, os dados coletados são registrados em seus respectivos arquivos em formato *Comma-Separated Values* (CSV). Por conseguinte, a etapa de pré-processamento das amostras previamente coletadas é feita a fim de gerar a componente y do conjunto de dados, que é a componente a ser estimada pelo modelo. O vetor y é obtido pela equação 2, possibilitando computar a correção necessária para estimar precisamente a SNR do canal de comunicação, gerando assim, um problema de regressão que pode ser solucionado por meio de uma rede neural.

$$\mathbf{y} = \beta - \mathbf{X} \quad (2)$$

onde β é o vetor que contém os valores de SNR atuante no canal de comunicação e \mathbf{X} é o vetor com os valores de SNR estimados pela função MSE modificada.

3.3.4 Treinamento e validação do modelo de DL

A arquitetura da rede neural é definida como sendo de baixa complexidade em virtude de possuir apenas quatro camadas, sendo duas delas ocultas. A utilização da topologia de tal forma, tem como objetivo identificar a relação não-linear entre as variáveis. Além disso, a escolha pela complexidade baixa torna-se necessária dado o interesse de implementar tal modelo a camada de física do transceptor, a qual deve introduzir a menor latência possível e consumindo a menor quantidade de recursos computacionais. A Figura 17 apresenta a topologia *feedforward* utilizada e suas respectivas funções de ativação.

A criação do modelo foi feita, utilizando a biblioteca Keras, uma *Application Programming Interfaces* (API) de alto nível, que facilita a implementação e treinamento do modelo em questão. A rede neural foi treinada utilizando a MSE como função de custo, além do *Adaptative Momentum* (ADAM) como algoritmo de otimização. Para validação dos modelos treinados para suas respectivas modulações, apresentou-se novamente as amostras a fim de verificar se o modelo adequou-se corretamente aos dados de treinamento. A Figura 18 apresenta a resposta obtida por cada modelo (QPSK, 16-QAM, 64-QAM e 256-QAM), evidenciando a capacidade dos mesmos em estimar a relação não linear subjacente entre as variáveis em questão (Medida da função MSE modificada e a correção necessária).

3.3.5 Integração a camada física

A integração da técnica de IA a camada física do transceptor é feita por meio da criação de um bloco de processamento no próprio *software* GNU Radio. Tal ferramenta oferece um

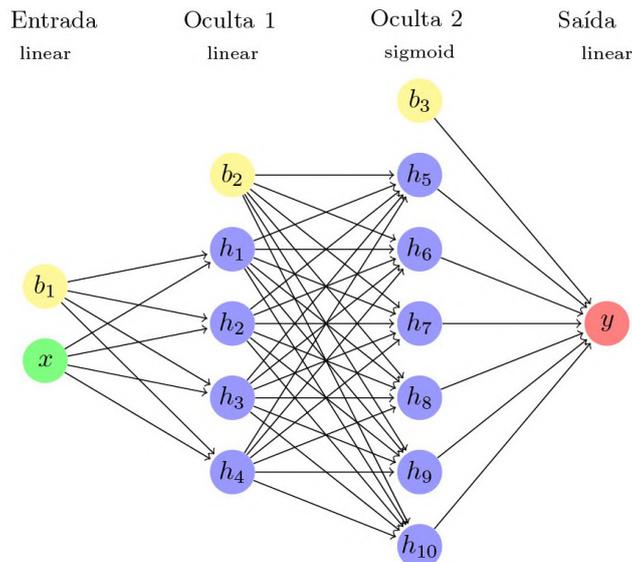


Figura 17: Estrutura da rede neural criada.

framework que possibilita a implementação de etapas de processamento digital de sinais desenvolvidas utilizando as linguagens C++ e Python. A criação do bloco através da linguagem Python é feita a fim de processar os símbolos QAM e estimar a SNR do canal por meio da função MSE modificada e da rede neural previamente treinada. Como o treinamento foi realizado de maneira *off-line*, ou seja, após o treinamento do modelo, este foi exportado para um arquivo em formato *JavaScript Object Notation* (JSON), bem como seus respectivos pesos em um segundo arquivo, a fim de serem utilizados em quaisquer aplicações posteriormente. No processo de inicialização do bloco, ou seja, seu construtor, os modelos são carregados com suas topologias e pesos, utilizando os recursos da biblioteca Keras. Aliado aos modelos, a implementação da função MSE modificada foi realizada utilizando a diretiva *Just-in-Time* (JIT) da biblioteca Numba para aceleração do processamento dos símbolos recebidos. Como a linguagem Python é interpretada, seu desempenho quando comparada a linguagem C++, a qual é uma linguagem compilada, é notavelmente menor. Ao utilizar tal diretiva, o processo iterativo de estimar e computar o erro do símbolo torna-se viável de ser realizado, visto que esta diretiva permite compilar funções Python em tempo de execução para melhorar seu desempenho.

Vale destacar que as etapas desenvolvidas e apresentadas proporcionam maior precisão no valor estimado de SNR. Vale destacar também que a partir da SNR estimada é definido o MCS a ser empregado no enlace de comunicação. A Tabela 8 apresenta os valores de MCS previstos no transceptor de complexidade reduzida, bem como os valores mínimos de SNR previstos para o seu respectivo emprego e o tipo de modulação associada.

3.4 Análise e resultados

A fim de verificar o desempenho da técnica LDNN-SNRE, tomou-se como base a estimativa obtida com base na NMSE, a qual é utilizada no transceptor Brasil 6G de complexidade reduzida. Para isso, realizou-se uma simulação juntamente com a cadeia de comunicação de maneira a coletar amostras das estimativas de SNR em uma faixa de valores entre -10 dB e 40 dB em passos de $0,1$ dB. O controle da SNR do canal de comunicação é feito por meio do ajuste do ganho aplicado ao ruído gaussiano inserido na recepção. O diagrama da simulação é

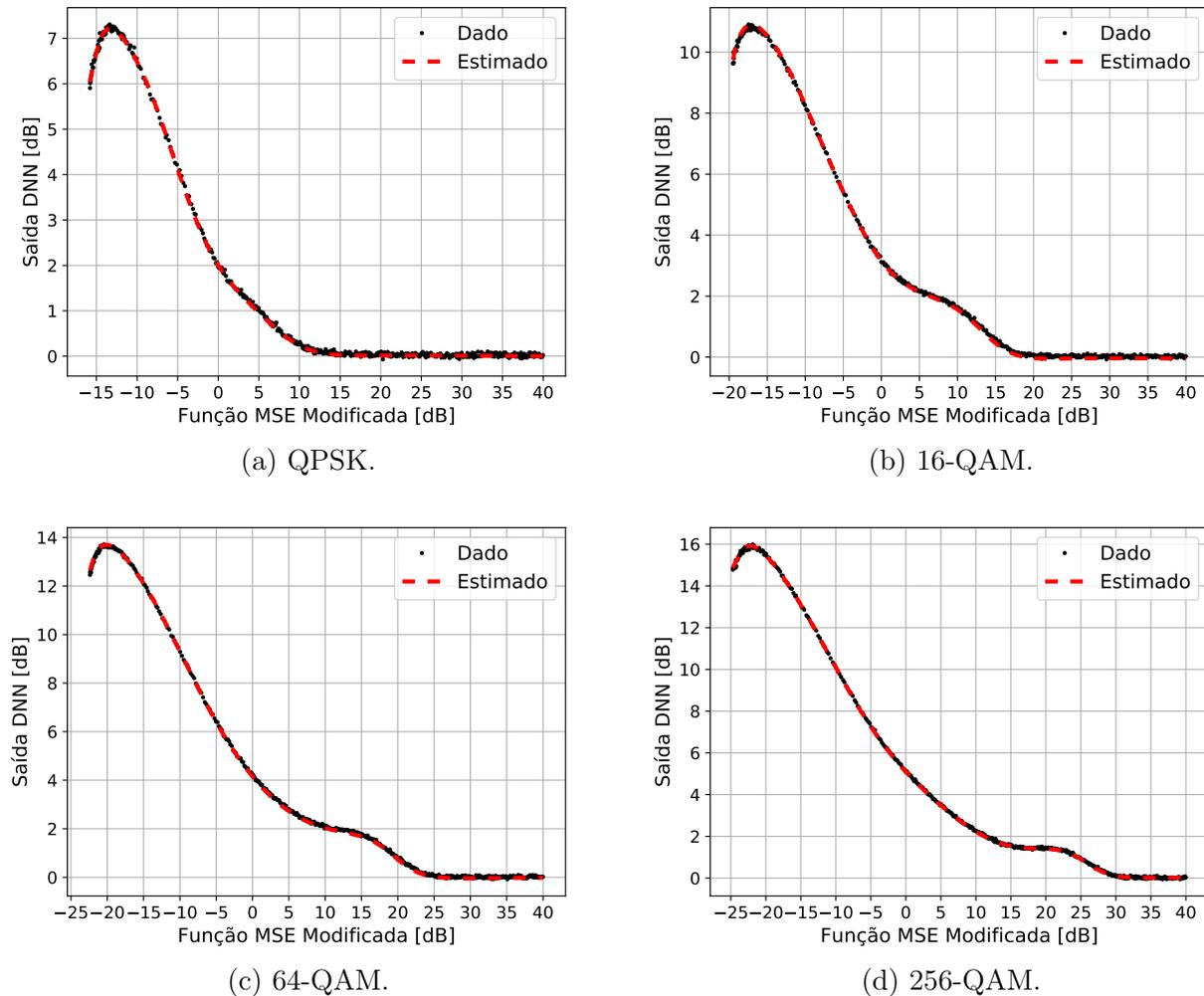


Figura 18: Saída dos modelos para as modulações: QPSK, 16-QAM, 64-QAM e 256-QAM.

apresentado na Figura 19, omitindo alguns processos da cadeia de comunicação, a fim de dar ênfase a implementação da técnica de IA.

O amostrador apresentado é responsável por amostrar as estimativas realizadas por cada uma das técnicas, bem como controlar a SNR aplicada ao canal de comunicação. Seu funcionamento é definido a partir da quantidade de amostras coletadas n e a frequência de amostragem f_s , sendo estes valores parametrizáveis. A Equação 3 apresenta como é computado o k -ésimo valor médio de SNR para as n amostras coletadas a cada $1/f_s$ em cada uma das métricas. Vale ressaltar que tal processo é realizado para todos os valores de SNR atuantes no canal de comunicação dado a faixa estabelecida anteriormente.

$$\gamma_k = \frac{1}{n} \sum_{i=0}^{n-1} \beta_i \quad (3)$$

onde β_i é o i -ésimo valor de SNR estimado pela métrica e coletado a cada $1/f_s$ segundos.

A simulação foi feita a fim de analisar o desempenho das métricas com o enlace operando com código MCS fixo e em seu maior valor, ou seja, o enlace operando com a modulação 256-QAM e taxa de codificação mínima. A Figura 20 apresenta os resultados obtidos, além da constelação de recepção com SNR em 27 dB. Os valores estimados por cada uma das métricas

Tabela 8: Tabela de MCS

MCS	Mínimo valor de SNR	Modulação
0	7.0	QPSK
1	8.0	QPSK
2	8.5	QPSK
3	9.5	QPSK
4	13.5	16-QAM
5	14.5	16-QAM
6	15.5	16-QAM
7	16.0	16-QAM
8	20.0	64-QAM
9	21.0	64-QAM
10	21.5	64-QAM
11	27.0	256-QAM
12	28.0	256-QAM
13	29.0	256-QAM

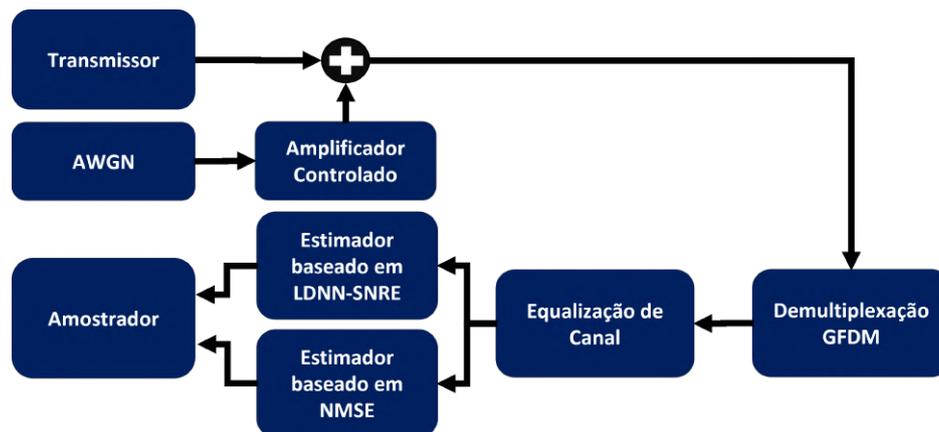


Figura 19: Diagrama da simulação.

estão identificados no gráfico.

Dados os resultados apresentados no gráfico, observa-se que a NMSE apresenta estimativas imprecisas para baixos valores de SNR, visto que sua convergência ao valor real da SNR configurada para o canal de comunicação só ocorre a partir de, aproximadamente, 27 dB. Dessa maneira, as estimativas realizadas por essa técnica para valores abaixo deste citado superestimam os valores reais, interpretando que a qualidade do canal de comunicação é melhor que a real situação. Por outro lado, ao analisar a técnica, que utiliza a função MSE modificada, observa-se que suas estimativas dentro da faixa positiva da SNR são melhores quando comparadas a NMSE. Logo, a utilização de um fator ponderador, que compensa os eventos internos não detectáveis em virtude de uma alta variância do ruído gaussiano, permite que as estimativas obtidas sejam mais precisas.

Por fim, a utilização de uma técnica baseada em inteligência artificial, a LDNN-SNRE, possibilita obter estimativas convergentes ao valor real da SNR configurada para o canal de comunicação. Logo, a aplicação da função MSE modificada juntamente com a correção fornecida pela rede neural mostra-se eficaz mesmo para baixos valores de SNR. Dito isso, a estimativa

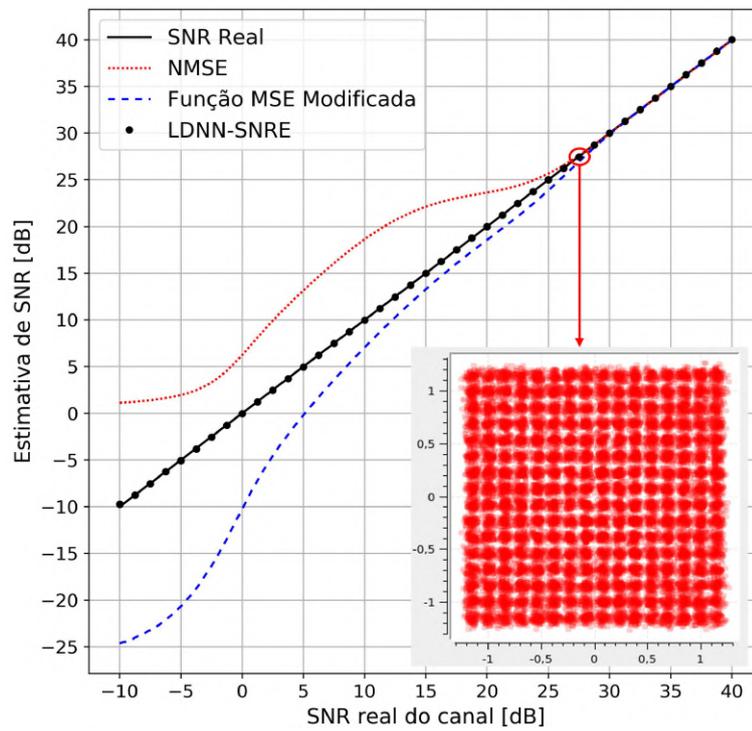


Figura 20: Resultado da análise de desempenho para 256-QAM.

obtida por essa técnica permite uma melhor definição do código MCS que, por sua vez, define a modulação e codificação a serem utilizadas em cada enlace de comunicação. Desta forma, é otimizada a definição do MCS do enlace de comunicação e, conseqüentemente, do QoS provido pela rede.

4 Forma de onda: adequações do GFDM para redução da PAPR

Anderson Reis Rufino Marins
anderson@inatel.br

As pesquisas visando a otimização e evolução da rede de acesso Brasil 6G contemplaram a forma de onda adotada pelo transceptor Brasil 6G que é o GFDM. As pesquisas realizadas com relação a forma de onda tiveram como foco técnicas de redução de PAPR, bem como análise da integração de uma da técnica, eleita como promissora, no transceptor Brasil 6G. Nas subseções seguintes estão apresentados os embasamento de técnicas principais estudadas, bem como o processo de integração e avaliação de uma delas no transceptor Brasil 6G.

4.1 Estudo de técnicas para Redução de PAPR

A PAPR é a razão entre a potência de pico máxima e a potência média de um sinal e é associada à eficiência energética do transmissor em sistemas de comunicação sem fio. Nativamente as formas de onda multiplexadas apresentam PAPR alta, devido a soma de subportadoras adjacentes em fase, e quanto maior o número de subportadoras, maior o valor da PAPR. Várias técnicas são propostas para reduzir a PAPR, dentre elas destaca-se o método envolvendo pré-codificação *Discrete Fourier Transform* (DFT), *Selective Mapping* (SLM), *Partial Transmission Sequence* (PTS) e IA. Esta seção tem por objetivo apresentar o estado da arte de cada técnica de redução de PAPR mencionada.

4.1.1 Pré-codificação DFT

A técnica de pré-codificação já vem sendo utilizada em sistemas de telecomunicações sem fio para redução da PAPR. A técnica consiste em, a partir de uma matriz de transformação, codificar um determinado conjunto de bits [8]. Esse método é aplicado nos símbolos modulados antecedendo o mapeamento nas subportadoras e atua de forma a espalhar toda a informação pelas subportadoras que serão transmitidas. Dessa forma, além de atuar na redução da PAPR, atua na melhora da eficiência em relação a taxa de erro pela manutenção da ortogonalidade entre subportadoras e reduzindo a interferência entre usuários.

A DFT é muito difundida como meio de pré-codificação como a forma de onda DFT-*spread-OFDM* (DFT-s-OFDM) ou *Single Carrier Frequency Division Multiple Access* (SC-FDMA). Com a implementação da DFT em OFDM, a DFT-s-OFDM, foi possível reduzir a PAPR em 6 dB em relação ao OFDM original. Com esta redução na PAPR, reduziu também o consumo energético do sistema o que proporcionou o seu uso nos enlaces de subida dos dispositivos móveis *Long Term Evolution* (LTE) e *5G New Radio* (5G-NR).

Devido a seu desempenho, a pré-codificação DFT tem sido considerada em diversos estudos e associada à várias formas de onda. Recentemente em OFDM, foi proposto um *framework* chamado *Generalized DFT-s-OFDM* onde se unifica a forma de onda OFDM com DFT sem a inserção de prefixo cíclico [9]. Assim o prefixo cíclico é substituído por um intervalo de guarda, sendo sua largura determinada dinamicamente de acordo com o espalhamento de atraso do canal. Com isso, é mantido a flexibilidade do canal juntamente com a contenção espectral do sistema, reduzindo a PAPR sem necessitar de processos adicionais.

Para *Filtered OFDM* (F-OFDM), foi feita uma pré-codificação com DFT conforme apresentado em [10], e assim como a DFT-s-OFDM, o princípio de transformar o F-OFDM em um

sinal com portadora única é mantido. O resultado foi uma redução de 2,5 dB na PAPR sem degradar o desempenho da taxa de erro de bits (BER).

Já na forma de onda GFDM com pré-codificação DFT [11, 12], houve uma redução bem significativa na PAPR, com uma redução próxima de 4 dB com relação a curva de PAPR do OFDM.

Em *Orthogonal Time Frequency Space* (OTFS), é proposta a forma de onda DFT-*spread*-OTFS (DFT-s-OTFS), sendo a pré-codificação inserida antes da operação de *Inverse Symplectic Finite Fourier Transform* (ISFFT) no transmissor sem aumento de complexidade no mesmo. Houve uma redução na PAPR próximo de 2 dB em relação ao OTFS original.

Com o exposto, é possível observar como a DFT é uma solução eficaz na redução da PAPR. Sua fácil implementação e complexidade baixa e a possibilidade de se associar a outras técnicas para melhorar ainda mais o seu desempenho na redução da PAPR são suas vantagens de destaque.

4.1.2 *Partial Time Sequence*

Dentre as técnicas de redução de acPAPR destaca-se a PTS, que tem sido utilizada de forma extensa por proporcionar um redução significativa da PAPR sem distorcer o sinal. O esquema da técnica consiste em dividir o vetor de dados de N elementos em X sub-blocos distintos [13]. Cada sub-bloco é submetido a operação de *Inverse Fast Fourier Transformer* (IFFT) de N pontos. Depois, as fases resultantes de cada sub-bloco é modificada por um conjunto de coeficientes de rotação de fase. Assim calcula-se a PAPR de cada subsequência e o sinal que apresentar o menor índice é o que será transmitida junto com a configuração de fase para decodificá-la na recepção. Porém o processo de busca exaustiva de uma PAPR ótima aumenta a complexidade computacional do sistema e a inserção de redundância na transmissão reduz a eficiência espectral da forma de onda usada no sistema.

Por ser uma técnica bastante explorada como técnica de redução de PAPR, é utilizada em várias formas de onda multiportadoras. Em OFDM, é encontrada na literatura várias soluções, sendo algumas delas com proposta para redução da complexidade do PTS. Uma das soluções importantes está em [14], onde é proposto um novo algoritmo para particionamento da IFFT de N pontos em duas partes. Na primeira parte, é feita uma sequência intermediária onde o sinal de entrada é submetido aos primeiro L estágios da IFFT. Então, esta sequência é dividida em V subsequências e depois é aplicado os demais $N - L$ estágios da IFFT. Finalizando, são multiplicados os parâmetros de fase e as subsequências são somadas e enviadas ao sistema de seleção que determinará o sinal com a menor PAPR. O novo algoritmo apresenta um resultado de 48% de redução da complexidade computacional em relação ao PTS convencional e diminuiu o desempenho de redução de PAPR em apenas 0,2 dB.

Em F-OFDM também é implementada a técnica PTS convencional, sendo introduzida em cada sub-banda do sinal [15]. Os resultados demonstram uma redução de aproximadamente 3 dB na PAPR e de 10 dB na emissão fora de faixa do sinal. Porém, mesmo não afetando o desempenho da taxa de erro de bits, houve um aumento na complexidade computacional do sistema o que se faz necessário planejar a implementação da técnica de forma a balancear o desempenho e a complexidade.

Para GFDM, também é proposta a implementação do PTS convencional [16]. Com a aplicação do PTS após a modulação do GFDM, houve uma redução de 5 dB na PAPR em relação ao GFDM convencional. Já em [17], é proposto um novo esquema de PTS exclusivo para GFDM com baixa complexidade *Low Complexity* GFDM (LC-GFDM). Neste caso, os autores

indicam que o método convencional não seria eficiente pois o método atua individualmente em cada símbolo, e a proposta do LC-GFDM seria fazer a soma deles no final do processamento. Para tal, foi desenvolvido o *Cumulative Symbol Optimization* PTS (CSO-PTS) pelos autores, que faz a otimização da sequência de dados após a soma de cada símbolo. Isso é feito com o intuito de eliminar os efeitos negativos do processo de adição de símbolo na PAPR do sinal. Foi demonstrado pelos resultados obtidos uma redução na PAPR de 6,5 dB com esse método novo e uma melhora também na redução da emissão fora de faixa e no desempenho da taxa de erro de bits.

4.1.3 *Selective Mapping*

A técnica *Selective Mapping* consiste em multiplicar o vetor de dados de entrada por um conjunto U de sequências pseudo-aleatórias (PN), onde se obtém um conjunto de sinais diferentes porém com a mesma informação. Após isso, um seletor determinará qual é o sinal que apresenta a menor PAPR entre eles e o transmite. Porém, para que o receptor possa decodificá-lo adequadamente, a informação de qual sequência PN foi utilizada deve ser também transmitida. Desta forma, esta técnica tem a desvantagem de reduzir a eficiência espectral, assim como ocorre na técnica PTS. Outra desvantagem seria a necessidade que o transmissor faça o cálculo de n IFFT, ou seja, quanto maior o número de n maior a complexidade computacional da mesma, contudo a probabilidade de se obter uma PAPR mais baixa aumenta. Sendo assim, uma relação de compromisso deve ser estabelecida.

Na literatura encontramos várias implementações em OFDM, como em [18] onde os autores propõe um novo esquema para redução da complexidade que divide a IFFT em dois estágios. No primeiro estágio é aplicada a IFFT com k estágios nos dados de entrada. No segundo estágio, após o sinal ser multiplicado pela sequência PN é aplicada os $n - k$ estágios restantes. Com este esquema conseguiu-se uma redução de 51% da complexidade computacional com 2048 subportadoras para um $n - k = 5$ mantendo o mesmo desempenho da redução de PAPR do que o SLM convencional.

Em [19] os autores avaliam a técnica SLM em diferentes números de subportadoras variando o valor de sequências de fase U .

4.1.4 *Inteligência Artificial*

Com a evolução dos sistemas de telecomunicações e o advento do Quinta Geração de Rede Móvel Celular (5G) e 6G, IA tem-se tornado a solução fundamental para melhoria de desempenho dos sistemas de comunicação. Com análise de dados e aprendizado de máquina, esta integração propõe solução cada vez mais eficiente na modelagem e otimização dos múltiplos cenários de comunicações sem fio. Assim com o uso de IA será possível o reconhecimento das características do ambiente fazendo as adequações da rede e o uso do espectro de acordo com o que cada usuário necessita. A questão da gestão de energia também é beneficiada pela integração de IA no transmissor. E com esse mesmo sentido, muitos estudos estão sendo feitos para aplicar IA em técnicas de redução de PAPR em sinais multiportadoras. Porém, ainda devido a alta complexidade computacional exigidas para o treinamento de novos modelos, gera uma restrição no uso de IA.

Na literatura, é proposto em [20] a implementação de Rede Neural para a redução da PAPR. Essa rede é treinada com um banco de dados obtido através de resultados das técnicas de filtragem e *clipping* simplificado com baixa PAPR. Para se ter uma melhoria no desempenho

da taxa de erro de bits, tanto o módulo de transmissão quanto o módulo de recepção são treinados baseados na otimização objetiva.

O sistema de processamento da transmissão é realizada em três partes. Na primeira, divide-se o sinal de entrada em duas partes: componentes de fase e componentes de quadratura. Na segunda, é extraída pelo sistema a informação efetiva à partir da análise de pesos e vieses dos dados de entrada. E na terceira e última, todas essas informações obtidas modelam a saída do sistema. No receptor, o sinal que foi transmitido é processado por um módulo semelhante ao do transmissor. Ambos os módulos do transmissor e receptor atualizam seus coeficientes simultaneamente e o treinamento é realizado com a realimentação a partir dos dados de saída desses módulos para reduzir a PAPR e minimizar a taxa de erro de bits. Nos resultados apresentados, o esquema proposto reduziu a PAPR em 4,5 dB para sinais OFDM.

Em [21] é proposto um novo esquema de redução de PAPR que foi denominado como PAPR *Reducing Network* (PRNet). O sistema emprega DL baseado na estrutura de *autoencoder*. Nesse método a rede neural profunda será treinada para o mapeamento dos símbolos de cada subportadora OFDM, garantindo a redução da PAPR e otimizando o desempenho da taxa de erro de bits. O treinamento do *autoencoder* é feito de forma a codificar e decodificar um símbolo de forma a distinguir o sinal de entrada do sinal de saída. Esse treinamento é realizado pelo método de otimização gradiente descendente estocástico, no qual os pesos e vieses são atualizados a cada iteração. Nos resultados das simulações, o PRNet superou o PTS em 2 dB, o *clipping* em 5 dB e o OFDM convencional em 7 dB aproximadamente. E uma constatação importante foi a que o PRNet não sobrecarregou o sistema de comunicação, o que permite a sua implementação em tempo real.

Em outro trabalho [22], é implementado DL para otimizar um esquema com *Tone Reservation* (TR). A técnica TR é utilizada para redução da PAPR a partir do cancelamento de picos inserindo um sinal em subportadoras reservadas para este fim. A proposta deste trabalho foi implementada em sinais OFDM sendo a estrutura de cancelamento de picos realizada após a IFFT no transmissor. Essa rede neural foi projetada para abrigar T camadas e reduzir a cada iteração de L picos do sinal. Para o treinamento, um conjunto de dados de treinamento é inserido para otimização do nível de cancelamento e os pesos da função *kernel* no domínio do tempo. O sistema se objetiva em minimizar a PAPR e a distorção do sinal sem aumentar a potência média do sinal de transmissão. Os resultados apresentaram valores de PAPR menores do que o sistema OFDM tradicional e de técnicas como SLM, *Least Square Approximation* TR (LSA-TR), *Scaling Signal to Clipping noise Ratio* (S-SCR) e *Extreme Learning Machine-based* TR (ELM-TR), porém, se comparada a PRNet, o seu desempenho é menor a um custo de redução considerável na complexidade do sistema.

Em outras literaturas também consideram aplicar os algoritmos de IA em técnicas de busca exaustiva como PTS e SLM. Em [23] é proposto uma implementação de uma versão modificada do algoritmo *Colônia Artificial de Abelhas* (CAB) com baixa complexidade que determinam os parâmetros de fase empregados pelo PTS na redução da PAPR no sinal OFDM. O algoritmo CAB é referenciado no sistema de busca por alimento das abelhas, sendo dividido em 3 partes. Na primeira parte, as abelhas trabalhadoras são enviadas para a fonte de néctar para determinarem a quantidade disponibilizada. Na segunda, as abelhas assistentes são treinadas para selecionar as fontes baseadas nas informações das abelhas trabalhadoras e extraírem o néctar. E por fim, as abelhas extratoras são enviadas aleatoriamente nos arredores para encontrar outras fontes de néctar. Para o contexto do PTS, a localização da fonte correspondente a localização da matriz de fatores de fase que possivelmente solucionarão o problema e a quantidade de néctar é a quantidade destas soluções. Os resultados das simulações mostraram que o algoritmo

proposto apresentou um desempenho bem próximo ao PTS ótimo (diferença de 0,35 dB) com uma redução expressiva da complexidade do sistema de aproximadamente 97%. Outros estudos semelhantes ao CAB demonstraram o mesmo comportamento na redução da PAPR para sistemas OFDM [24, 25].

Da mesma metodologia, CAB também é implementada no esquema SLM para redução da complexidade [26]. Foi obtido uma redução de complexidade da técnica SLM de 75% com 128 subportadoras e 60,9% com 256 subportadoras.

Como é possível observar, a implementação dos algoritmos de IA juntamente as técnicas de redução de PAPR, consegue-se otimizar a complexidade computacional e os ganhos energéticos do sistema, o que torna esses métodos viáveis a sua implementação nos sistemas modernos de comunicação.

4.2 Integração da Técnica de Redução de PAPR no Transceptor Brasil 6G

Dentre as técnicas apresentadas na seção 4.1, foi eleita a técnica de pré-codificação DFT para ser implementada no transceptor Brasil 6G. Essa definição se deu por algumas vantagens apresentadas pela pré-codificação DFT: seu bom desempenho na redução da PAPR, sua baixa complexidade computacional e sua fácil implementação.

O conceito da DFT aplicada a um vetor de dados é fazer o espalhamento de toda informação em todas as subportadoras de forma que cada subportadora carrega uma combinação linear dos elementos do vetor de entrada ponderados por cada elemento correspondente na matriz de transformação de *Fourier*.

Assim, para implementar a pré-codificação DFT em um vetor, basta multiplicá-lo pela matriz de transformação de Fourier. Porém, no caso do transceptor Brasil 6G, nem todo elemento da matriz de dados que serão transmitidos são informações úteis. Existem também dados de controle e pilotos que servem como referências entre transmissor e receptor. Então neste caso, esses dados de controle não podem ser modificados pela DFT. A Figura 21 apresenta como foi a implementação da pré-codificação DFT no transceptor Brasil 6G.

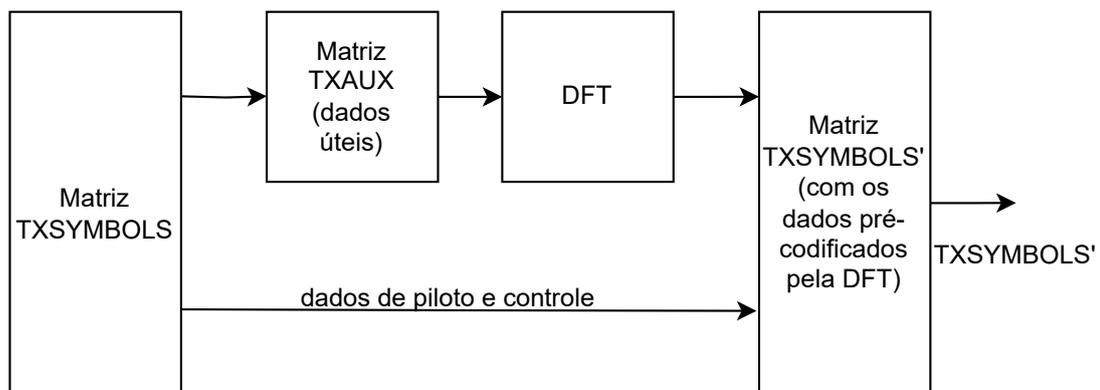


Figura 21: Diagrama em blocos do processo de DFT na matriz de entrada do modulador GFDM do transceptor Brasil 6G.

Assim como apresentado no diagrama da Figura 21, temos a matriz **TXSYMBOLS** que armazena os elementos modulados QAM e os dados de controle e piloto que serão transmitidos. Assim, para que se possa aplicar a pré-codificação DFT, as informações úteis devem ser

separadas para que somente estes sejam modificados pelos processos seguintes. Desta forma, emprega-se uma matriz auxiliar denominada **TXAUX** para armazenamento dos dados de informação útil. No passo seguinte, multiplica-se a matriz **TXAUX** pela matriz de transformação DFT para a realização da pré-codificação. Após esse processo, os elementos da matriz **TXAUX** são reposicionados na mesma posição que tinham na matriz **TXSYMBOLS** porém agora na nova matriz **TXSYMBOLS'** conforme Figura 21.

Na recepção, para recuperação dos dados de informação úteis, é feito o processo inverso da transmissão. Conforme apresentado no diagrama em blocos da Figura 22, uma vez feito o processo de recepção dos dados, são armazenados na matriz **RXSYMBOLS'**. Nessa matriz são armazenados os dados de informação pré-codificados e dos dados de controle. Da mesma forma que foi feito no transmissor, é separado os dados de informação úteis pré-codificados na matriz auxiliar **RXAUX**. Assim, a matriz **RXAUX** é multiplicada pela inversa da matriz de transformação, ou seja, pela matriz *Inverse Discrete Fourier Transformer* (IDFT). Com isso é desfeito o processo de pré-codificação. Com os dados de informação recuperados, os mesmos são reposicionados na posição original que assumiam na matriz **RXSYMBOLS'**, mas compondo agora a matriz de dados recuperados **RXSYMBOLS**. Desta forma, os dados de informação recuperados a matriz **RXSYMBOLS** estão prontos para a demodulação QAM da recepção.

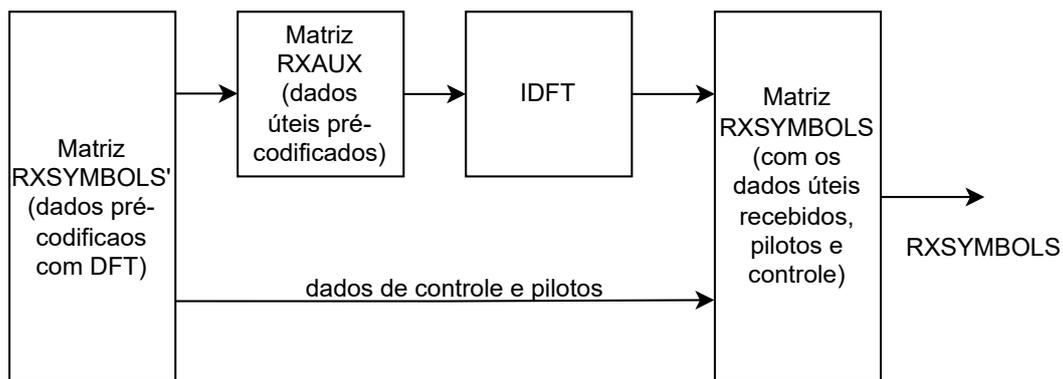


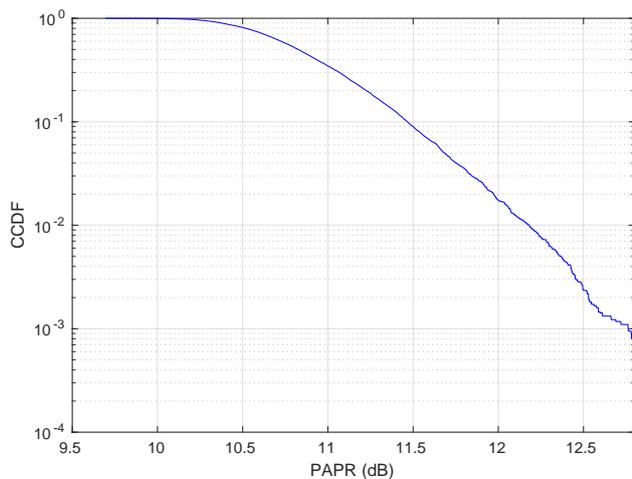
Figura 22: Diagrama em blocos do processo de IDFT na matriz de saída do modulador GFDM do transceptor Brasil 6G.

4.3 Testes e Resultados alcançados

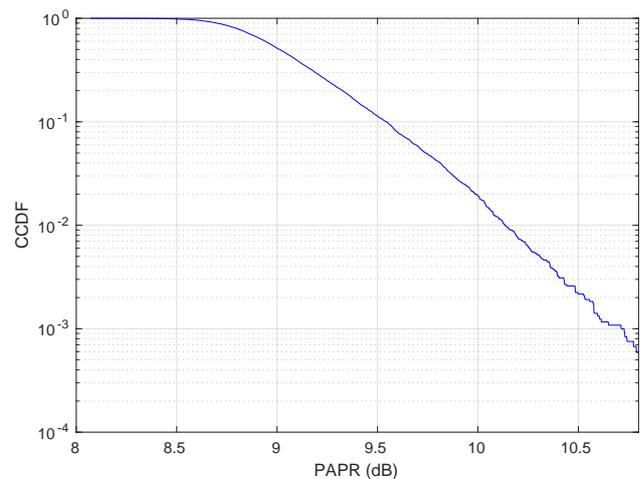
A partir do desenvolvimento das etapas de processamento descritas na seção 4.2, foi possível integrar, via simulação computacional, a técnica de pré-codificação DFT no transceptor Brasil 6G. A integração via simulação foi considerada a fim de facilitar o processo de avaliação de desempenho da técnica selecionada. Desta forma, foram realizada a integração e testes em um ambiente de simulação do desenvolvido em *MATLAB*. Através de tal simulação, foi possível analisar e constatar a eficácia da redução da PAPR pela pré-codificação DFT conforme apresentado a seguir.

As Figura 23 e 24 mostram, respectivamente, o resultado da análise da função de distribuição cumulativa complementar ou *Complementary Cumulative Distribution Function* (CCDF) da PAPR e a curva de *Symbol Error Rate* (SER) obtida na simulação do transceptor sem e com o uso da técnica pré-codificação DFT. Vale ressaltar que os resultados foram obtidos considerando o transceptor Brasil 6G configurado para operar com os seguintes parâmetros principais:

- Forma de onda: GFDM;
- Modulação: 4-QAM;
- Número de subportadoras (K): 8192;
- Número de subsímbolos (M): 4;
- Fator de *roll-off* (α): 0.

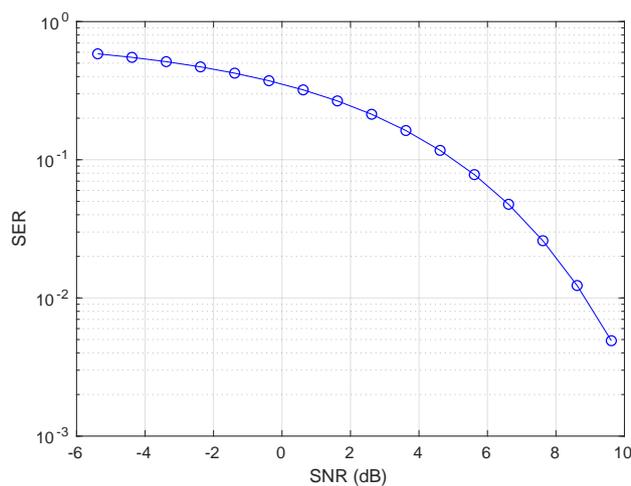


(a) PAPR GFDM.

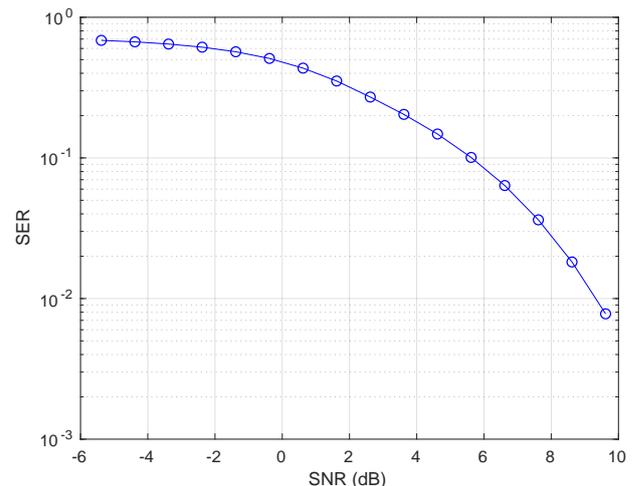


(b) PAPR DFT-s-GFDM.

Figura 23: PAPR 4-QAM para $K = 8192$, $M = 4$, (*roll-off*) $\alpha = 0$.



(a) SER GFDM.



(b) SER DFT-s-GFDM.

Figura 24: SER 4-QAM para $K = 8192$, $M = 4$, (*roll-off*) $\alpha = 0$.

A partir dos resultados apresentados na Figura 23 é possível observar que há um ganho de aproximadamente 2 dB na PAPR. Na Figura 23(a), curva PAPR GFDM sem DFT, pode-se observar que, para uma PAPR de 12 dB tem-se uma CCDF de 2×10^{-2} . Já para a Figura 23(b),

observando-se a curva PAPR DFT-*spread*-GFDM (DFT-s-GFDM) tem-se uma PAPR igual a 10 dB com a mesma CCDF de 2×10^{-2} .

A partir dos resultados apresentados na Figura 24 é possível observar uma redução no desempenho do sistema em termos de SER quando empregada a técnica baseada em DFT. Como apresentado na Figura 24(a), SER para a forma de onda GFDM sem DFT, para uma SNR de 8 dB tem-se a SER de 2×10^{-2} ; na Figura 23, parte (b), SER quando empregada a técnica DFT-s-GFDM, tem-se para a SNR de 8 dB a SER de 3×10^{-2} .

Desta forma, pode-se concluir com o uso da pré-codificação DFT na configuração citada, foi possível obter um ganho de 2 dB na PAPR, porém, a um custo de uma redução no desempenho do sistema em relação a SER.

5 Conclusão

O presente relatório apresentou diversas melhorias nos transceptores da rede de acesso do Projeto Brasil 6G, abrangendo otimizações nas camadas física e de enlace, implementação de técnicas IA na camada física do transceptor para estimativa de SNR. Além disso, foram apresentadas pesquisas relacionadas à técnicas de redução de PAPR, bem como otimizações de PAPR integradas à forma de onda GFDM. Os resultados obtidos durante a implementação e avaliação dessas melhorias demonstraram benefícios significativos para o futuro das redes de comunicações móveis.

As melhorias nas camadas física e de enlace, por meio da implementação de novos algoritmos e técnicas, além de otimizações, viabilizam uma rede de acesso mais robusta e efetiva. Isso se traduz em uma maior taxa de transferência de dados, menor taxa de erro de transmissão e maior qualidade de serviço para os usuários da rede.

A utilização de algoritmos de IA para estimar de forma precisa e dinâmica a SNR em tempo real permite um melhor gerenciamento de recursos, como por exemplo, a alocação dinâmica de largura de banda, potência e modulação. Isso resulta na utilização mais eficiente do espectro, proporcionando maior capacidade de transmissão e uma comunicação mais confiável.

Por fim, a redução de PAPR é crucial para mitigar distorções no sinal de RF e melhorar a eficiência de transmissão. Ao aplicar técnicas avançadas de redução de PAPR, é possível minimizar a amplificação não linear do sinal, garantindo uma transmissão mais estável e confiável. Isso permite um uso mais eficiente dos amplificadores de potência e ajuda a evitar a degradação do sinal causada por picos de potência elevados.

Em um cenário em que o 6G se aproxima e suas respectivas demandas continuam a crescer exponencialmente, é fundamental enfatizar a importância do desenvolvimento tecnológico contínuo das redes de acesso. Essas melhorias são essenciais para impulsionar o avanço das telecomunicações e da próxima geração de comunicações móveis e garantir uma infraestrutura sólida e eficiente capaz de suportar as altas taxas de dados, a conectividade ubíqua, bem como o melhor suporte à aplicações inovadoras que podem advir com a rede 6G.

Referências

- [1] Centro de Referência em Radiocomunicações (CRR) Inatel, “Atividade 2.1 - Definição das arquiteturas física e lógica da rede protótipo,” *Projeto Brasil 6G*, Jul. 2022.
- [2] J. R. Hampton, *Introduction to MIMO Communications*. New York, NY, USA: Cambridge University Press, 2014.
- [3] Centro de Referência em Radiocomunicações (CRR) Inatel, “Atividade 3.4 - Técnicas MIMO,” *Projeto Brasil 6G*, Agosto 2023.
- [4] GNU Radio, “GNU Radio Web Site,” 2023, [Online]. Disponível em: <http://www.gnuradio.org>. Acessado em: 28 de abril de 2023.
- [5] A. M. Khan, V. Jeoti, M. Rehman, e M. Jilani, “Noise power estimation for broadcasting ofdm systems,” in *2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*, 2017, p. 1–6.
- [6] R. Kagami e L. Mendes, “A low-complexity deep neural network for signal-to-interference-plus-noise ratio estimation,” in *Anais do I Workshop de Redes 6G*. Porto Alegre, RS, Brasil: SBC, 2021, p. 1–6. [Online]. Disponível em: <https://sol.sbc.org.br/index.php/w6g/article/view/17227>
- [7] Projeto Brasil 6G, “Definição das arquiteturas física e lógica da Plataforma Brasil 6G,” 2022, [Online]. Disponível em: <https://inatel.br/brasil6g/brasil-6g/338-brasil-6g-meta-2-atividade-2-1-arquiteturas-de-redes/file>. Acessado em: 10 de julho de 2023.
- [8] P. Kumar, L. Kansal, G. S. Gaba, M. Mounir, A. Sharma, e P. K. Singh, “Impact of peak to average power ratio reduction techniques on generalized frequency division multiplexing for 5th generation systems,” *Computers and Electrical Engineering*, v. 95, p. 107386, 2021. [Online]. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0045790621003542>
- [9] G. Berardinelli, “Generalized DFT-s-OFDM waveforms without cyclic prefix,” *IEEE Access*, v. 6, p. 4677–4689, 2018.
- [10] M. Ben Mabrouk, M. Chaffi, Y. Louet, e F. Bader, “A precoding-based PAPR reduction technique for UF-OFDM and filtered-OFDM modulations in 5G systems,” in *European Wireless 2017; 23th European Wireless Conference*, 2017, p. 1–6.
- [11] S. Tiwari, S. S. Das, e K. K. Bandyopadhyay, “Precoded GFDM system to combat inter carrier interference: Performance analysis,” 2015. [Online]. Disponível em: <https://arxiv.org/abs/1506.03744>
- [12] S. S. Das e S. Tiwari, “Discrete Fourier transform spreading-based generalised frequency division multiplexing,” *Electronics Letters*, v. 51, n. 10, p. 789–791, 2015. [Online]. Disponível em: <https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/el.2014.3833>

- [13] G. P. Aquino e L. L. Mendes, “PAPR reduction and performance improvement in OFDM systems using WHT-based schemes,” *Journal of Communication and Information Systems*, v. 32, n. 1, Nov. 2017. [Online]. Disponível em: <https://jcis.sbvt.org.br/jcis/article/view/490>
- [14] D.-W. Lim, S.-J. Heo, J.-S. No, e H. Chung, “A new PTS OFDM scheme with low complexity for PAPR reduction,” *IEEE Transactions on Broadcasting*, v. 52, n. 1, p. 77–82, 2006.
- [15] Y. A. Al-Jawhar, K. N. Ramli, M. A. Taher, N. S. M. Shah, S. A. Mostafa, e B. A. Khalaf, “Improving PAPR performance of filtered OFDM for 5G communications using PTS,” *ETRI Journal*, v. 43, n. 2, p. 209–220, 2021. [Online]. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.4218/etrij.2019-0358>
- [16] A. Endang Jayati, I. Wirawan, T. Suryani, e E. Endroyono, “Partial transmit sequence and selected mapping schemes for PAPR reduction in GFDM systems,” *International Journal of Intelligent Engineering and Systems*, v. 12, p. 114–122, 12 2019.
- [17] Ş. Şimçir e N. Taşpınar, “Cumulative symbol optimization–based partial transmit sequence technique for PAPR reduction in low complexity GFDM system,” *Transactions on Emerging Telecommunications Technologies*, v. 31, n. 6, p. e3801, 2020. [Online]. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1002/ett.3801>
- [18] D.-W. Lim, J.-S. No, C.-W. Lim, e H. Chung, “A new slm ofdm scheme with low complexity for papr reduction,” *IEEE Signal Processing Letters*, v. 12, n. 2, p. 93–96, 2005.
- [19] K. Mhatre e U. P. Khot, “Efficient selective mapping papr reduction technique,” *Procedia Computer Science*, v. 45, p. 620–627, 2015, international Conference on Advanced Computing Technologies and Applications (ICACTA). [Online]. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1877050915003531>
- [20] F. Zou, Z. Liu, X. Hu, e G. Wang, “A novel PAPR reduction scheme for OFDM systems based on neural networks,” *Wireless Communications and Mobile Computing*, v. 2021, p. 5574807, 2021. [Online]. Disponível em: <https://doi.org/10.1155/2021/5574807>
- [21] M. Kim, W. Lee, e D.-H. Cho, “A novel PAPR reduction scheme for OFDM system based on deep learning,” *IEEE Communications Letters*, v. 22, n. 3, p. 510–513, 2018.
- [22] X. Wang, N. Jin, e J. Wei, “A model-driven DL algorithm for PAPR reduction in OFDM system,” *IEEE Communications Letters*, v. 25, n. 7, p. 2270–2274, 2021.
- [23] Y. Wang, W. Chen, e C. Tellambura, “A papr reduction method based on artificial bee colony algorithm for ofdm signals,” *IEEE Transactions on Wireless Communications*, v. 9, n. 10, p. 2994–2999, 2010.
- [24] D. Ramesh, K. M. Sharma, e K. R. Kiran, “Papr reduction by partial transmit sequence and complexity reduction by abc algorithm,” in *2014 Fourth International Conference on Advances in Computing and Communications*, 2014, p. 25–28.

- [25] T. Mata, P. Boonsrimuang, e P. Boontra, “A papr reduction scheme based on improved pts with abc algorithm for ofdm signal,” in *2018 15th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, 2018, p. 469–472.
- [26] X. Cheng, D. Liu, S. Feng, H. Fang, e D. Liu, “An artificial bee colony-based slm scheme for papr reduction in ofdm systems,” in *2017 2nd IEEE International Conference on Computational Intelligence and Applications (ICCIA)*, 2017, p. 449–453.