



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE AUTOMAÇÃO E
SISTEMAS

Igor de Oliveira Silvestre

**ALGORITMOS DE ADAPTAÇÃO EM REDES LORAWAN COM NODOS
MÓVEIS: ESTUDO DE CASO EM PARQUE EÓLICO**

Florianópolis

2022

Igor de Oliveira Silvestre

**ALGORITMOS DE ADAPTAÇÃO EM REDES LORAWAN COM NODOS
MÓVEIS: ESTUDO DE CASO EM PARQUE EÓLICO**

Dissertação submetida ao Programa de Pós-Graduação em Engenharia de Automação e Sistemas da Universidade Federal de Santa Catarina como parte dos requisitos para obtenção do título de Mestre em Engenharia de Automação e Sistemas.

Orientador: Prof. Marcelo De Lellis Costa de Oliveira, Dr. Eng.

Florianópolis
2022

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Silvestre, Igor

Algoritmos de Adaptação em Redes LoRaWAN com Nodos Móveis
: Estudo de Caso em Parque Eólico / Igor Silvestre ;
orientador, Marcelo De Lellis Costa de Oliveira, 2022.
73 p.

Dissertação (mestrado) - Universidade Federal de Santa
Catarina, Centro Tecnológico, Programa de Pós-Graduação em
Engenharia de Automação e Sistemas, Florianópolis, 2022.

Inclui referências.

1. Engenharia de Automação e Sistemas. 2. Redes de longo
alcance. 3. Redes de baixo consumo. 4. Simulação de redes.
5. Redes sem fio. I. De Lellis Costa de Oliveira,
Marcelo. II. Universidade Federal de Santa Catarina.
Programa de Pós-Graduação em Engenharia de Automação e
Sistemas. III. Título.

Igor de Oliveira Silvestre
Algoritmos de Adaptação em Redes LoRaWAN com Nós Móveis:
Estudo de Caso em Parque Eólico

O presente trabalho em nível de mestrado foi avaliado e aprovado por banca examinadora composta pelos seguintes membros:

Prof. Arliones Stevert Hoeller Jr., Dr.
UFSC

Prof. Carlos Barros Montez, Dr.
UFSC

Prof. Richard Demo Souza, Dr.
UFSC

Certificamos que esta é a **versão original e final** do trabalho de conclusão que foi julgado adequado para obtenção do título de mestre em Engenharia de Automação e Sistemas, na área de concentração Controle, Automação e Sistemas.

Coordenação do Programa de Pós-Graduação

Prof. Marcelo De Lellis Costa de Oliveira, Dr.
Orientador

Florianópolis, 2022.

À minha família e amigos

AGRADECIMENTOS

A esta universidade e seu corpo docente, que me deram as oportunidades e os meios para eu realizar um sonho pessoal.

Ao meu orientador, pela oportunidade, suporte, confiança e sobretudo respeito.

Aos meus pais, pelo incentivo em persistir e apoio incondicional.

A minha vó, pelo acolhimento, ensino e carinho.

Ao meu irmão, pelo auxílio intelectual e emocional, por padecer de minhas frustrações e me incentivar diariamente.

“Those who can imagine anything, can create the impossible. ”
(Alan Turing)

RESUMO

Durante a construção de um parque eólico é comum que os colaboradores permaneçam por longos períodos expostos às intempéries, o que pode levar à fadiga e complicações de saúde. Além disso, os parques eólicos normalmente ocupam vastas áreas, que levam um tempo considerável para serem percorridas a pé ou até mesmo por veículos. Isto costuma ser determinante no limite mínimo de tempo para a chegada de uma equipe de primeiros socorros, no caso de acidentes. Neste contexto, é possível aumentar a segurança dos trabalhadores por meio do uso de dispositivos vestíveis (“crachás”) caracterizados como nodos móveis (*endpoints*) de uma rede sem fio do tipo *long range* (LoRa). Isto não somente permite o rastreamento da posição desses profissionais, obtida por um receptor GPS no crachá, como possibilita a transmissão de um sinal de emergência. Motivado por esse problema, com o objetivo de otimizar o desempenho da rede, nesta dissertação são comparados alguns algoritmos de *adaptive data rate* (ADR) para redes LoRa existentes na literatura utilizando o simulador FLoRa. Além disso, é proposto um novo algoritmo, denominado *distance-ADR* (D-ADR), que com base na distância ao *gateway* mais próximo determina a potência de transmissão e o fator de espalhamento das mensagens a serem transmitidas pelo *endpoint*. Por meio de resultados de simulação com o software FloRa demonstra-se que o algoritmo D-ADR melhora o desempenho de rede, medido em termos do *data extraction rate* (DER), em situações onde os *endpoints* são móveis, alcançando um DER acima de 72% em um cenário de *gateway* único e 300 *endpoints* móveis transmitindo um pacote a cada minuto, enquanto o algoritmo ADR-TTN alcança um DER de 57%. Esta vantagem se mantém no cenário de um parque eólico específico com 4 *gateways* e 1500 *endpoints* móveis transmitindo um pacote a cada minuto, onde o D-ADR alcança um DER de 61%, enquanto os demais algoritmos não ultrapassam 60%. Também são apresentados resultados práticos preliminares do D-ADR implementado em hardware do crachá e testado em ambiente urbano.

Palavras-chave: Redes de longo alcance, redes de baixo consumo, redes sem fio, simulação de redes, protocolos de comunicação.

ABSTRACT

During the construction of wind farms workers may spend large amounts of time exposed to harsh weather conditions, which can lead to fatigue and health complications. Moreover, wind farms usually occupy large areas that may take a relatively long time to be traveled by foot or even in vehicles. This usually determines the minimum response time for the arrival of rescue teams in the case of accidents, for instance. In this context, it is possible to augment the personell's safety by equipping them with wearable devices which are moving nodes (endpoints) of a long range (LoRa) wireless communication network. This not only allows for tracking each worker's position, obtained by means of a GPS receiver, but also for transmitting distress signals. With this motivation, in this dissertation existing adaptive data rate (ADR) algorithms found in the literature are compared to one another by means of the FLoRa simulator. Moreover, a new algorithm, named distance-ADR (D-ADR) is proposed which, based on the endpoints distance to the nearest gateway, determines the transmission power and the spreading factor with which the packets are transmitted. By means of simulation results with the FLoRa software, it is shown that deployment of the D-ADR improves the network performance measured in terms of the data extraction rate (DER) in situations where the endpoints are moving, yielding a DER above 72% in a scenario consisting of a single gateway and 300 endpoints transmitting at every minute, whereas the ADR-TTN algorithm, for instance, reaches a DER of 57%. This advantage is maintained in a specific wind farm case with 4 gateways and 1500 moving endpoints also transmitting at every minute: the D-ADR reaches a DER of 61% while the other algorithms do not surpass the 60% threshold. Also, practical preliminary results are presented regarding the implementation of the D-ADR in hardware and tested in an urban environment.

Keywords: long range networks, low-power networks, wireless networks, network simulation, communication protocols.

LISTA DE ILUSTRAÇÕES

Figura 1 – Acidentes em usinas eólicas ao longo dos anos	15
Figura 2 – Capacidade instalada mundialmente em usinas eólicas. Adaptado de GWEC (2015).	15
Figura 3 – Comparação entre tecnologias IoT. Os vértices mais distantes do centro do polígono indicam melhores valores	16
Figura 4 – Vista aérea do parque eólico Enel, com a distribuição dos <i>gateways</i> LoRa.	17
Figura 5 – Ortogonalidade entre distintos SF	20
Figura 6 – <i>Time on Air</i> (ToA) para diferentes <i>payloads</i> e SF	21
Figura 7 – Transmissões consecutivas de um mesmo símbolo com SFs diferentes	22
Figura 8 – Falha na ortogonalidade entre distintos SF	23
Figura 9 – <i>Frame</i> do pacote LoRa	24
Figura 10 – Estrutura do <i>Preamble</i> LoRa	25
Figura 11 – Exemplo de recebimento de <i>Preamble</i> LoRa	25
Figura 12 – Topologia estrela LoRaWAN entre <i>endpoints</i> e <i>gateways</i>	26
Figura 13 – LoRa/LoRaWAN no modelo OSI	27
Figura 14 – Separação de canais para AU915-928	28
Figura 15 – <i>Time slots</i> de recepção dos <i>endpoints</i> de classe A, B e C	29
Figura 16 – Encapsulamento de um pacote LoRaWAN	30
Figura 17 – <i>Frame</i> LoRaWAN de <i>uplink</i>	31
Figura 18 – Fluxograma do algoritmo ADR-TTN executado no <i>network server</i>	34
Figura 19 – Exemplo de alocação de SF dos algoritmos ExploRa	35
Figura 20 – SNR recebido pelo NS e SNR filtrado pelo algoritmo EMA-ADR	37
Figura 21 – Fluxograma do algoritmo D-ADR no lado do <i>network server</i>	39
Figura 22 – Atribuição de SF e TP em função da distância no algoritmo D-ADR.	40
Figura 23 – Fluxograma do algoritmo D-ADR no lado do <i>endpoint</i>	41
Figura 24 – Ambiente de simulação FLoRa	46
Figura 25 – Comportamento da mobilidade do <i>endpoint</i>	47
Figura 26 – Alcance dos <i>endpoints</i> LoRa para o modelo Okumura-Hata	49
Figura 27 – Perdas de pacotes resultantes em função da potência de transmissão	50
Figura 28 – Perda por colisões e DER resultantes de acordo com a variação de <i>endpoints</i>	51
Figura 29 – Distribuições de <i>gateways</i>	51
Figura 30 – DER em função do número de <i>gateways</i>	52
Figura 31 – DER em função do número de <i>endpoints</i> para diferentes períodos entre transmissões	53
Figura 32 – DER de acordo com a alteração do período entre transmissões	54
Figura 33 – Determinação das áreas críticas	55
Figura 34 – DER em função do período entre transmissões dos <i>endpoints</i>	56

Figura 35 – Cenário de um <i>gateway</i> e 300 <i>endpoints</i> dentro uma área de 35 km ² . . .	57
Figura 36 – DER resultante no cenário de um único <i>gateway</i> com 300 <i>endpoints</i> estáticos.	58
Figura 37 – Alocação de SF e TP nos algoritmos ADR-TTN e ADR+ no cenário de um <i>gateway</i> com 300 <i>endpoints</i> estáticos.	58
Figura 38 – DER resultante no cenário de um único <i>gateway</i> com 300 <i>endpoints</i> móveis.	59
Figura 39 – Cenário de parque eólico com 4 <i>gateways</i> e 1500 <i>endpoints</i> móveis em uma área de 35 km ²	60
Figura 40 – DER resultante em cenário de parque eólico com 4 <i>gateways</i> e 1500 <i>endpoints</i>	60
Figura 41 – Gasto energético total entre distintos algoritmos ADR.	61
Figura 42 – Dinâmica do algoritmo D-ADR.	62
Figura 43 – <i>Hardware</i> utilizado.	63
Figura 44 – Sistema MQTT para coleta de dados.	65
Figura 45 – Trajetória percorrida.	65
Figura 46 – Distância e parâmetros de SF e TP escolhidos.	66

LISTA DE TABELAS

Tabela 1 – <i>Data Rates</i> e seus parâmetros correspondentes	27
Tabela 2 – SNR requerido para recepção correta de sinal	33
Tabela 3 – Parâmetros de simulação	48
Tabela 4 – Gasto energético em função de algoritmo ADR.	62

NOMENCLATURA

SIGLAS

ACK	<i>Acknowledgement.</i>
ABP	<i>Activation By Personalisation.</i>
ADR	<i>Adaptive Data Rate.</i>
BW	<i>Bandwidth.</i>
CR	<i>Code Rate.</i>
CRC	<i>Cyclic Redundancy Check.</i>
DER	<i>Data Extraction Rate.</i>
DR	<i>Data Rate.</i>
D-ADR	<i>Distance-ADR.</i>
EMA	<i>Exponential Moving Average.</i>
FEC	<i>Forward Error Correction.</i>
GW	<i>Gateway.</i>
GNSS	<i>Global Navigation Satellite System .</i>
HMI	<i>Human-Machine Interface.</i>
ISM	<i>Industrial Scientific and Medical.</i>
IMU	<i>Inertial measurement unit.</i>
IP	<i>Internet Protocol.</i>
IoT	<i>Internet of Things.</i>
LoS	<i>Line-of-Sight.</i>
LoRaWAN	<i>Long Range Wide Area Network.</i>
LoRa	<i>Long Range.</i>
M2M	<i>Machine to Machine.</i>
MAC	<i>Medium Access Control.</i>
MIC	<i>Message Integrity Code.</i>
MQTT	<i>Message Queuing Telemetry Transport.</i>
NS	<i>Network Server.</i>
OSI	<i>Open Systems Interconnection.</i>
OTAA	<i>Over-The-Air Activation.</i>
QoS	<i>Quality of Service.</i>
RSSI	<i>Received Signal Strength Indication.</i>
SSL	<i>Secure Sockets Layer.</i>
SNR	<i>Signal-to-Noise Ratio.</i>
SF	<i>Spreading Factor.</i>
SoC	<i>System on a Chip.</i>
ToA	<i>Time on Air.</i>
TCP	<i>Transmission Control Protocol.</i>

TP

Transmission Power.

SUMÁRIO

1	INTRODUÇÃO	15
1.1	OBJETIVOS	17
2	REDES DE LONGO ALCANCE	19
2.1	FATOR DE ESPALHAMENTO	20
2.2	POTÊNCIA DE TRANSMISSÃO	23
2.3	PACOTES LORA	24
2.4	LORAWAN	25
3	ALGORITMOS DE ADR EXISTENTES	32
3.1	RUÍDO NO CANAL	32
3.2	ALGORITMOS PARA <i>ENDPOINTS</i> ESTÁTICOS	33
3.3	ALGORITMOS PARA <i>ENDPOINTS</i> MÓVEIS	35
4	ALGORITMO ADR PROPOSTO: DISTANCE-ADR	38
5	MODELAGEM E AMBIENTE DE SIMULAÇÃO	42
5.1	<i>PATH LOSS</i>	42
5.2	SIMULAÇÃO DE REDES LORAWAN	45
5.3	MOBILIDADE DOS <i>ENDPOINTS</i>	47
6	RESULTADOS DE SIMULAÇÃO	48
6.1	ALCANCE DOS <i>ENDPOINTS</i>	48
6.2	CENÁRIO INICIAL	49
6.3	ÁREAS CRÍTICAS	53
6.4	ALGORITMOS ADR	57
7	TRABALHO EXPERIMENTAL	63
7.1	<i>HARDWARE</i> E PROCEDIMENTOS	63
8	CONCLUSÕES	67
	BIBLIOGRAFIA	69

1 INTRODUÇÃO

A diversificação da matriz energética tem sido uma das principais preocupações dos governos em todo o mundo ultimamente, sendo a energia eólica uma das fontes mais promissoras. No Brasil, a participação da energia eólica na matriz total de eletricidade do país cresceu de 0,4 % no ano de 2010 (2.176,6 GWh) para 8,8 % (57.051 GWh) em 2020 (EPE, 2021). Essa tendência deve continuar: as projeções são de que a participação da energia eólica aumentará ainda mais, para 12 % (aproximadamente 83.000 GWh) até 2024 (EPE, 2019).

Durante a construção de um parque eólico, composto por muitas torres eólicas, os trabalhadores são comumente expostos a condições climáticas adversas por longos períodos de tempo, o que pode levar à fadiga ou outras complicações de saúde. Apesar de todas as medidas de precaução, acidentes também podem acontecer, como os mencionados por Asian et al. (2017). A figura 1 mostra o aumento no número de acidentes relatados em jornais relacionados a construção e manutenção de parques eólicos ao longo dos anos.

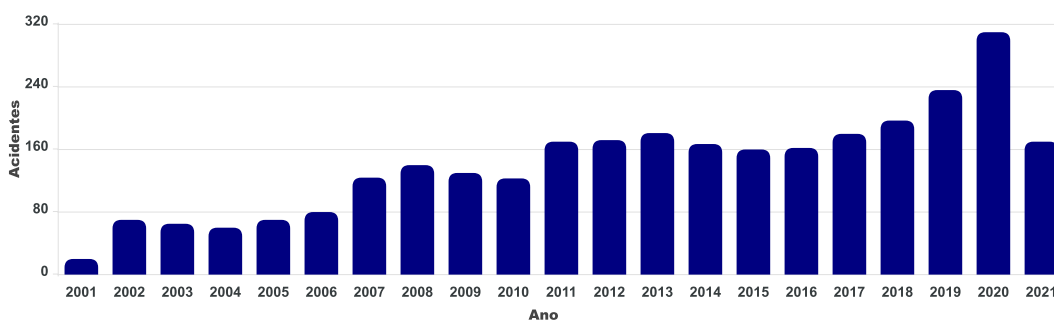


Figura 1 – Acidentes em usinas eólicas ao longo dos anos. Adaptado de Konstantinidis e Botsaris (2016).

Este aumento no número de acidentes está fortemente atrelado a popularização da energia eólica, cuja capacidade instalada mundialmente vem crescendo num passo ainda maior, como mostra a Figura 2.

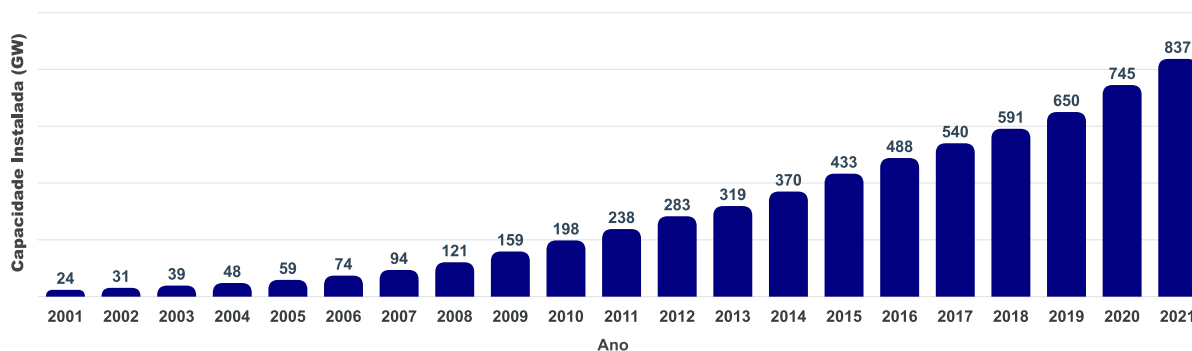


Figura 2 – Capacidade instalada mundialmente em usinas eólicas. Adaptado de GWEC (2015).

Como os parques eólicos podem ocupar grandes áreas, de até dezenas de quilômetros quadrados, em caso de emergência, pode demorar vários minutos até que uma equipe de primeiros socorros chegue a um determinado local do parque, dada a falta de cobertura de rede móvel ou de qualquer outro meio de comunicação confiável e viável. Essa demora deve ser reduzida significativamente, pois pode ser a diferença entre a recuperação total e as sequelas do acidentado, ou mesmo entre a vida e a morte em casos extremos. Mais do que simplesmente reagir a uma emergência, uma rede de comunicação local permitiria, por exemplo, que a equipe de segurança realizasse ações para evitar a ocorrência de algumas situações de emergência.

Entre as alternativas tecnológicas para enfrentar esse desafio, uma única antena de rede de telefonia móvel (3G a 5G) poderia fornecer cobertura suficiente, mas os custos associados são proibitivos, como no caso de um *link* de comunicação via satélite dedicado. Tecnologias de Internet das Coisas (IoT) também devem ser consideradas, e foram comparadas entre si usando o gráfico poligonal da Figura 3. Nela reflete-se o fato de roteadores Wi-Fi terem uma cobertura de campo aberto de apenas algumas centenas de metros, portanto, seriam necessárias centenas de roteadores espalhados pelo parque eólico. O padrão Zigbee tem um alcance menor ainda, entretanto é possível utilizar uma topologia *mesh*, fazendo com que todos os nodos sejam interconectados, estabelecendo possíveis caminhos para roteamento e assim aumentando o seu alcance. Outra possibilidade é o uso da tecnologia IoT de banda estreita (NB-IoT), que apresenta menor latência e melhor qualidade de serviço (QoS). Este QoS leva em consideração a perda de pacotes, *jitter*, *delay* e taxas de transferência (MARTINEZ-CARO; CANO, 2021). Entretanto a tecnologia NB-IoT perde em desempenho para as redes Long Range (LoRa) e SigFox nos quesitos de cobertura, vida útil da bateria dos nodos (*endpoints*) e também em custo. Por fim, entre as opções LoRa e SigFox, a primeira foi escolhida devido à maior acessibilidade (maior disponibilidade e menor custo) do hardware no mercado.

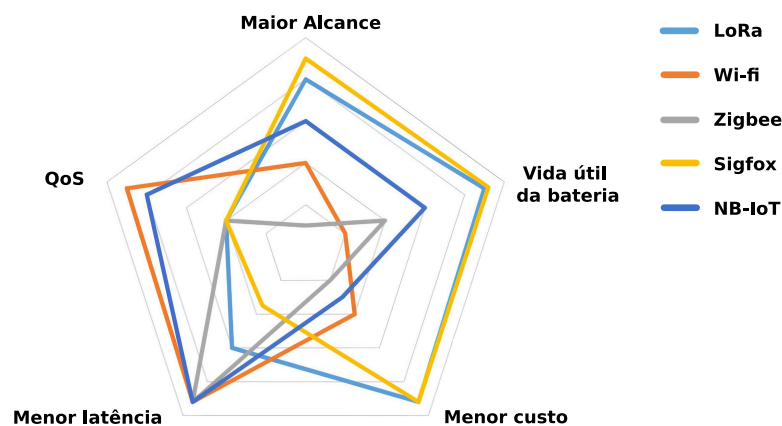


Figura 3 – Comparação entre tecnologias IoT. Os vértices mais distantes do centro do polígono indicam melhores valores. Adaptado de Sendra et al. (2020).

Uma vista aérea do parque eólico que motivou o presente trabalho é mostrada na Figura 4. A usina pertence à empresa Enel e está localizada na região Nordeste do Brasil, estendendo-se por 35 km², com capacidade instalada de 353 MW distribuídos em 84 torres. Na imagem também pode ser vista a localização dos quatro *gateways* LoRa (pontos amarelos), destinados a se comunicar com até 1500 *endpoints* móveis, que são módulos de comunicação compactos (crachás) usados pelos profissionais na planta.

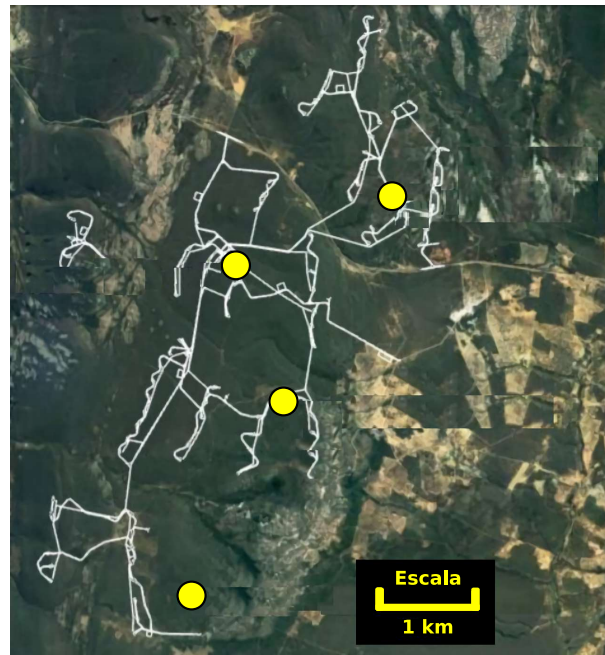


Figura 4 – Vista aérea do parque eólico Enel, com a distribuição dos *gateways* LoRa.

Devido ao número elevado de *endpoints*, existe uma preocupação quanto à viabilidade da utilização do LoRa para a comunicação. Uma análise quanto à configuração da rede (número de *gateways*, limite de alcance, parâmetros de transmissão) necessária para comportar os 1500 *endpoints* é desejada. Além disso, como a natureza dos *endpoints* é móvel e a posição de cada um deles é conhecida graças a um GNSS embarcado, questiona-se sobre a possibilidade de implementar algum algoritmo que melhore o desempenho desta rede LoRa.

1.1 OBJETIVOS

O objetivo principal desta dissertação de mestrado é o desenvolvimento de um algoritmo de *adaptive data rate* (ADR) - taxa de dados adaptativa, na língua inglesa - que utilize dados de geolocalização para melhorar o desempenho de uma rede LoRa com nodos (*endpoints*) móveis. Para alcançar este objetivo principal é possível elencar os seguintes objetivos específicos:

- Levantamento dos algoritmos de ADR existentes;

- Configuração de um ambiente de simulação para avaliar o desempenho dos algoritmos;
- Propor um novo algoritmo ADR que apresente desempenho superior utilizando dados de geolocalização;
- Análise por simulação do algoritmo proposto frente aos existentes;

O restante desta dissertação é organizada da seguinte maneira: no capítulo 2 são descritos a tecnologia LoRa e o protocolo LoRaWAN. No capítulo 3 são descritos alguns algoritmos ADR já existentes encontrados na literatura. No capítulo 4 é apresentado o algoritmo ADR proposto nesta dissertação, o *distance-ADR* (D-ADR), que utiliza dados de GNSS para ajustar os parâmetros de transmissão LoRa. No capítulo 5 a modelagem da comunicação LoRa é apresentada assim como o *software* de simulação escolhido. No capítulo 6 os algoritmos ADR, tanto os existentes quanto o novo algoritmo, são comparados através de simulação. No capítulo 7 o trabalho experimental é apresentado, tanto o *hardware* utilizado quanto o algoritmo implementado, assim como resultados preliminares de testes de campo. Finalmente, no capítulo 8 as considerações finais são feitas assim como perspectivas de trabalhos futuros são apresentadas.

2 REDES DE LONGO ALCANCE

LoRa é uma técnica proprietária de modulação de espectro espalhado de propagação de *chirp* (CSS) (SEMTECH, 2015). É comumente usada em aplicações onde um *endpoint* (nodo) de baixa potência deve transmitir pacotes em taxas de dados entre 0,3 kbps e 5,5 kbps em longas distâncias. Portanto, as redes de comunicação que empregam a modulação LoRa são geralmente descritas como redes de área ampla de baixa potência (LPWANs) (BOHN, 2020). A distância máxima entre um *endpoint* e um *gateway* LoRa é determinada principalmente pelas características do terreno: nas áreas rurais essa distância pode chegar a 20 km, enquanto que em um ambiente urbano ela pode cair para 5 km devido à presença de obstáculos maiores e mais altos, como edifícios.

Os *endpoints* e *gateways* LoRa são dispositivos (*hardware*) compostos por pelo menos um módulo de rádio e uma unidade de processamento (por exemplo, microcontrolador), mas também podem acomodar outros componentes. No caso do parque eólico considerado, os *endpoints* são dispositivos compactos e vestíveis que compreendem também um módulo GNSS, uma bateria e uma unidade de medição inercial (IMU). Como os *endpoints* são alimentados por baterias, o consumo de energia é uma preocupação. Isto não se aplica aos *gateways* LoRa por serem comumente conectados à rede elétrica ou serem alimentados por baterias maiores, já que sua operação requer um gasto energético maior.

As redes LoRa operam em uma faixa de largura de banda sem licença ISM (*Industrial, Scientific and Medical*), que está disponibilizada no mundo inteiro. Na Europa, as frequências liberadas para este tipo de uso são as EU433 (433.05–434.79 MHz) e a EU863-870 (863–870 MHz), entretanto essas bandas variam em outros países. Um problema de utilizar estas frequências “abertas” sem licença ISM é que elas costumam ser bastante utilizadas e por isso sofrem frequentemente de interferência, além do fato que as suas taxas de transmissão costumam ser baixas.

Um dos parâmetros de regulamentação das redes LoRa é o fator de ciclo de trabalho, segundo o qual essas redes costumam utilizar o valor de 1%, que significa que tanto os *endpoints* quanto os *gateways* só podem transmitir durante 1% do tempo (LORA ALLIANCE, 2019). Por exemplo, se o tempo que a mensagem leva para alcançar um *gateway* provinda de um *endpoint* é de 500 ms, uma nova mensagem só poderá ser enviada por este *endpoint* depois de 49,5 segundos.

No Brasil o LoRa opera na faixa ISM de 915 MHz a 928 MHz, faixa referenciada como AU915-928, com sua nomenclatura originada da região australiana. Existem algumas restrições para a utilização desta faixa, que é normatizada pela Anatel (ANATEL, 2017).

2.1 FATOR DE ESPALHAMENTO

A modulação LoRa utiliza pulsos de *chirp*¹ para codificar informações. Esses pulsos são basicamente uma onda senoidal que aumenta (*up-chirp*) ou diminui (*down-chirp*) sua frequência com o tempo em um intervalo delimitado por F_{baixa} e F_{alta} , conforme ilustrado na Figura 5. Cada *chirp* corresponde a um símbolo caracterizado por um valor de frequência observado nos instantes inicial e final do *chirp*. Na Figura 5, três símbolos distintos são expressos através de *up-chirps* em um quadro LoRa com fator de espalhamento (SF) 7, denotado pelas linhas pretas, seguido por um símbolo com SF8, na linha laranja. Observe que o fator de espalhamento determina a taxa na qual a frequência da onda senoidal é variada. Por exemplo, quanto maior o fator de espalhamento, menos inclinados serão os segmentos de linha. Isso significa que quanto maior o SF mais tempo levará para que cada pacote seja transmitido, exigindo também mais energia para a transmissão do pacote. De fato, cada incremento no SF reduz a taxa de transmissão pela metade. A modulação LoRa tem um total de seis fatores de espalhamento, variando de SF7 a SF12.

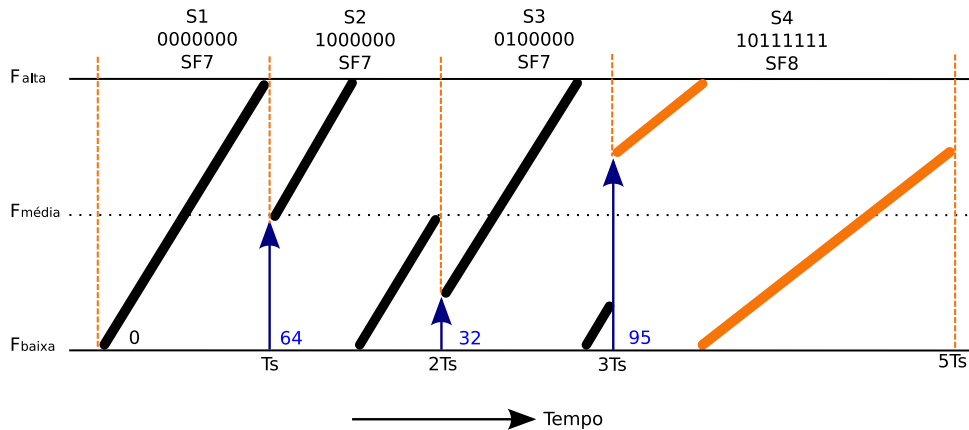


Figura 5 – Símbolos $S1$ até $S4$ representados através de *up-chirps* de LoRa com SF7 e SF8.

A taxa de símbolos R_s pode ser definida como (SEMTECH, 2013):

$$R_s = \frac{BW}{2^{\text{SF}}} \text{ , [Símbolos/s]} \quad (2.1)$$

onde BW se refere à largura de banda. A taxa de *bits* pode ser definida como:

$$R_b = \frac{BW}{2^{\text{SF}}} \frac{4}{4 + \text{CR}} \text{ ,} \quad (2.2)$$

sendo que CR retrata a proporção de *bits* não redundantes envolvidos na técnica de *Forward Error Correction (FEC)*².

¹ *Chirp* é um termo em língua inglesa que faz referência ao gorjeio dos pássaros, cuja frequência está em constante alternância.

² Método de correção de erros onde o transmissor envia dados redundantes e o receptor reconhece somente os dados que aparentemente não contêm erros. O FEC fornece ao receptor a capacidade de

O LoRa permite que utilizemos os seguintes valores para CR:

$$CR = \frac{4}{4+n}, \quad n = 1, 2, 3, 4. \quad (2.3)$$

Portanto os possíveis valores de CR são $4/5$, $4/6$, $4/7$ e $4/8$. O tempo de um bit pode ser calculado como $T_c = 1/R_c$.

Podemos tirar algumas conclusões dessas equações, como por exemplo é perceptível que quanto maior o valor de fator de espalhamento, menor é a taxa de *bits* e maior o tempo de duração de um símbolo. Isto acarreta um acréscimo no tempo que a mensagem leva para alcançar o receptor (ToA) e no consumo de energia para realizar a transmissão dessas mensagens. Entretanto, de acordo com Waret et al. (2019) a elevação do fator de espalhamento SF permite uma zona de abrangência maior devido ao aumento da robustez a ruídos. A Figura 6 mostra os tempos que as mensagens costumam levar com diferentes *payloads* e fatores de espalhamento (SF) para alcançar o destinatário para uma largura de banda de 125kHz em uma frequência na faixa de 863-870 MHz.

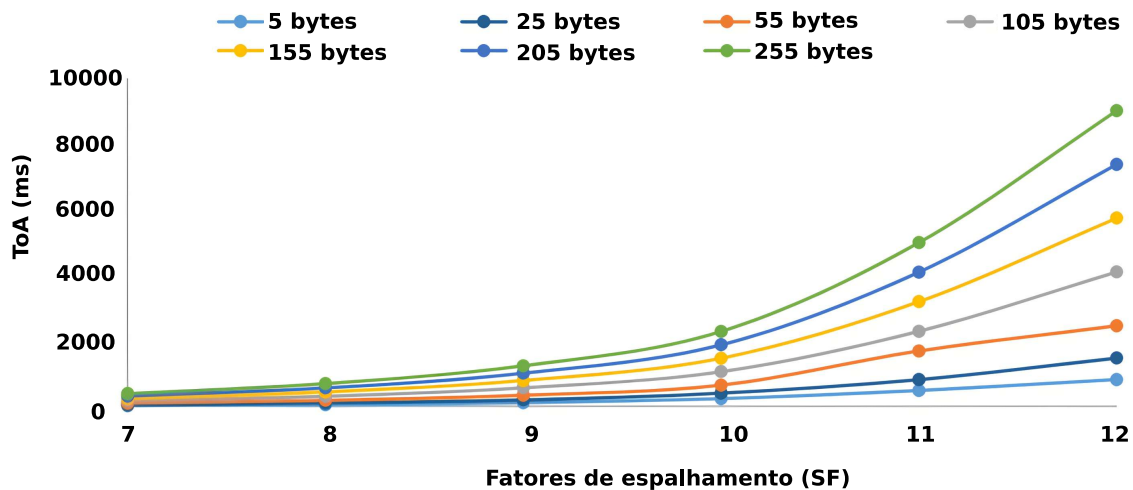


Figura 6 – *Time on Air* (ToA) para diferentes *payloads* e SF. Fonte: Sendra et al. (2020).

Como podemos ver na equação 2.1, a taxa de transmissão do LoRa é inversamente proporcional ao fator de espalhamento. Para ser mais exato, cada incremento no SF reduz pela metade a taxa de transmissão. Isto significa que um pacote transmitido com SF8 irá demorar duas vezes mais para chegar ao receptor do que o mesmo pacote transmitido com SF7. Portanto é possível dizer que a geração de símbolos, ilustrada na Figura 7, é ortogonal entre os fatores de espalhamento. Um símbolo gerado por modulação CSS precisa de 4 parâmetros: o fator de espalhamento S, uma frequência mínima f_{min} e uma corrigir erros sem a necessidade da retransmissão de dados, porém gera um aumento fixo na largura de banda devido à redundância. Em um exemplo simples, se cada *bit* de informação é enviado 9 vezes, o receptor poderá receber uma das 8 (2^3) possíveis variações. Caso receba os 9 bits “101011110” o receptor deverá decidir democraticamente por maioria que o valor correto para o *bit* original é “1”.

máxima f_{max} e os bits a serem transmitidos. A partir destes parâmetros uma frequência inicial f_0 inicial é calculada em função do fator de espalhamento S (REYNDERS; POLLIN, 2016), o símbolo então pode ser transmitido começando com a frequência f_0 até f_{max} e continuando de f_{min} até f_0 , completando a transmissão. Portanto, se um símbolo é gerado pelo fator de espalhamento 7, não existirá nenhum outro símbolo gerado pelos SFs restantes que seja paralelo a ele.

Isto significa que pacotes enviados com diferentes SF não interferem entre si e podem ser recebidos ao mesmo tempo por um *gateway*. A Figura 7 demonstra esta ortogonalidade em um espectrograma (neste caso representada pela derivada dos *chirps*) entre SFs.

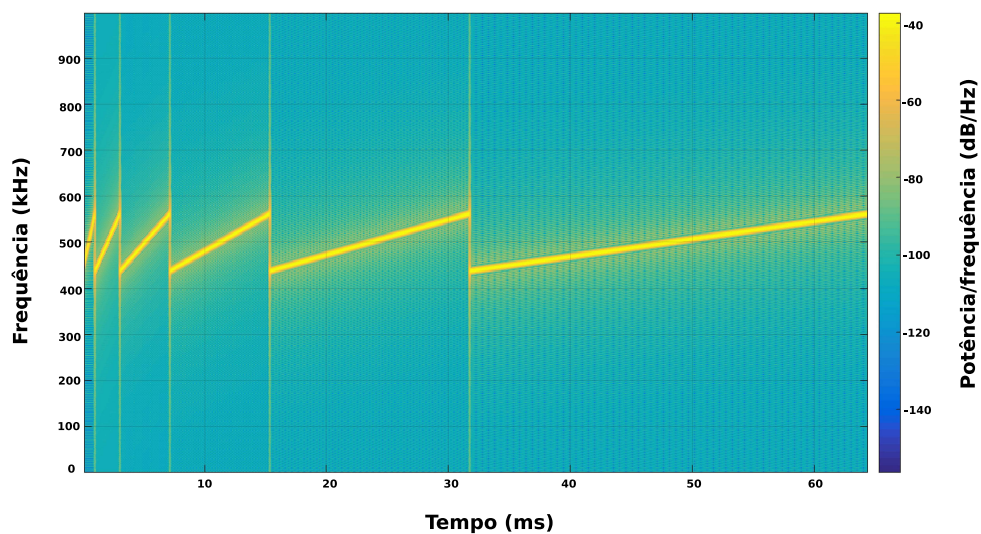


Figura 7 – Transmissões consecutivas de um mesmo símbolo com SFs diferentes. Fonte: Ghoslya (2017).

Porém, esta ortogonalidade na realidade não é perfeita e é possível o acontecimento de interferências e colisões entre mensagens transmitidas com distintos valores de SF. Isto é devido ao fato de que se a largura de banda for alterada, é possível obter sinais não ortogonais com SF distintos. A Figura 8 ilustra esta situação onde temos *chirps* com SFs diferentes e não ortogonais (ortogonalidade representada pelas derivadas dos sinais no espectrograma). Portanto, a ortogonalidade da rede LoRa não pode ser garantida com a utilização de diferentes canais.

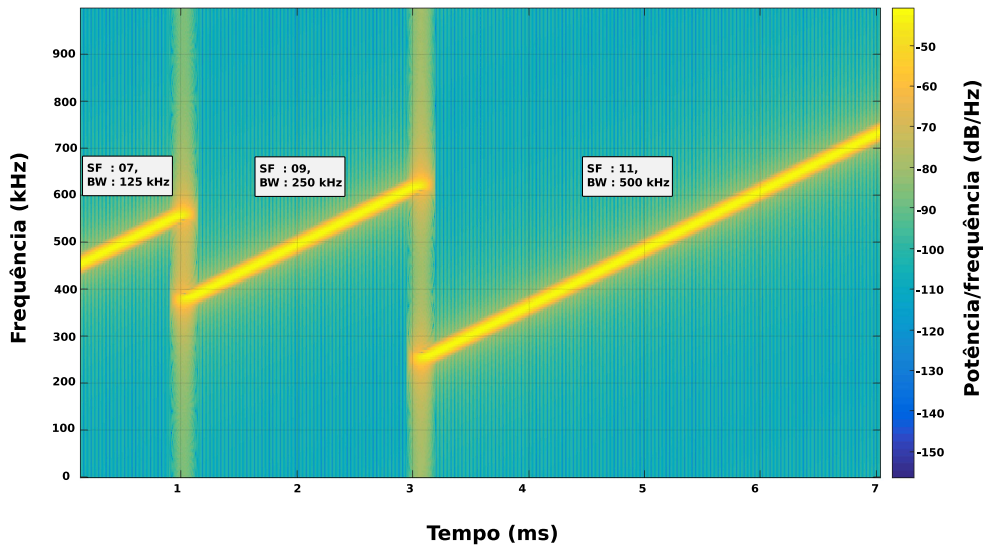


Figura 8 – Falha na ortogonalidade entre distintos SFs. Fonte: Ghoslya (2017).

2.2 POTÊNCIA DE TRANSMISSÃO

Outro parâmetro importante de uma rede LoRa é a potência de transmissão (TP) de cada *endpoint*, que normalmente pode variar entre 2 dBm e 14 dBm. Normalmente a potência de transmissão de um rádio é uma variável medida em Watts, entretanto em comunicações *wireless* é mais comum indicar a potência em decibéis referenciada a um miliwatt (dBm), conforme a seguinte expressão:

$$TP = 10 \cdot \log \frac{\text{Power}}{1\text{mW}} \text{ [dBm]}. \quad (2.4)$$

Esta potência de transmissão está diretamente relacionada ao consumo de energia e ao alcance destas transmissões: uma potência de transmissão maior resulta em um maior consumo de energia e alcance. Entretanto não é somente a potência de transmissão que afeta o alcance de comunicação, o qual também é influenciado pela modulação e pela sensibilidade dos receptores.

A sensibilidade de um receptor é comumente definida como a energia de sinal mínima necessária para a recepção correta de dados. Sinais com energia abaixo deste mínimo são considerados como ruído branco. Portanto, uma energia de sinal mínima detectável menor possibilita a recepção de sinais mais distantes ou transmitidos a uma potência menor. A sensibilidade do receptor é determinada pelo nível mínimo de ruído do receptor e pela relação sinal-ruído SNR, conforme a seguinte expressão:

$$P_{rx} = \text{Nível Mínimo de Ruído do Receptor} + \text{SNR} \text{ [dBm]}. \quad (2.5)$$

Por exemplo, se o nível mínimo de ruído do receptor for = -100 dBm e o SNR = 20 dBm, o sinal que chega neste receptor deve ter no mínimo $P_{rx} = -80$ dBm para ser interpretado corretamente e não como ruído branco.

2.3 PACOTES LORA

O pacote de transmissão LoRa é dividido em algumas partes, a saber: *Preamble*, *Cyclic Redundancy Check (CRC)*, *Header*, *Payload* e CRC do *Payload*. O pacote começa com um *preamble* que é utilizado para o receptor sincronizar antes de receber os dados, após isso existe um campo opcional de *header* onde se encontra as informações sobre o tamanho do pacote em *bytes*, o *code-rate* (proporção entre *bits* não redundantes mencionado na equação 2.3) utilizado para transmissão e se um CRC de 16 *bits* é utilizado para o *payload* ao final do *frame*. Ainda neste campo de *header* existe outro CRC para que o receptor possa descartar pacotes com *headers* inválidos. A Figura 9 representa o formato do pacote LoRa com as partes aqui descritas (SEMTECH, 2013).

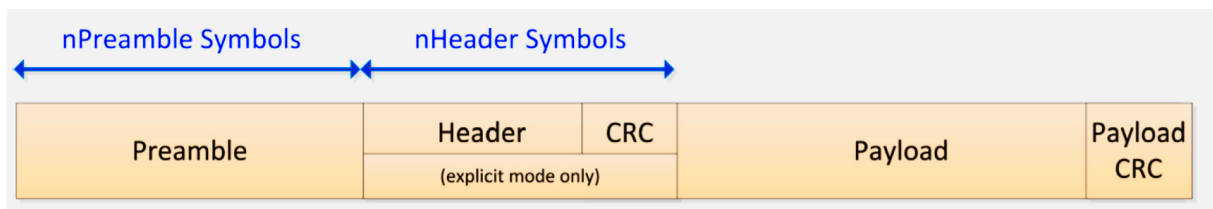


Figura 9 – *Frame* do pacote LoRa. Fonte: Xiao e Xie (2018).

O início da transmissão de um pacote LoRa é marcado com uma sequência pré-definida de símbolos necessários para a sincronização do receptor. Essa sequência de símbolos, chamada de *preamble*, que pode ser configurada no dispositivo, podendo conter desde 10 até 65536 símbolos, sendo 4 deles fixos e os outros de tamanho ajustável. O *preamble* começa sua sequência com esta parte de tamanho ajustável entre 6 e 65532 símbolos, que se trata de *up-chirps* (*chirps* crescentes em frequência) constantes para ajudar na detecção do início do pacote, seguidos por 2 *chirps* equivalentes à palavra de sincronismo (*sync word*). De acordo com Xiao e Xie (2018) a palavra de sincronismo pode ser usada para distinguir diferentes dispositivos de diferentes redes LoRa, usando um valor para cada. Após os 2 símbolos da palavra de sincronismo, o *preamble* finaliza com 2 *down-chirps* (*chirps* decrescentes em frequência) e logo após, um intervalo de tempo de 25% do símbolo representa um silêncio usado para alinhar o tempo no receptor. A Figura 10 representa o *preamble* de forma gráfica.

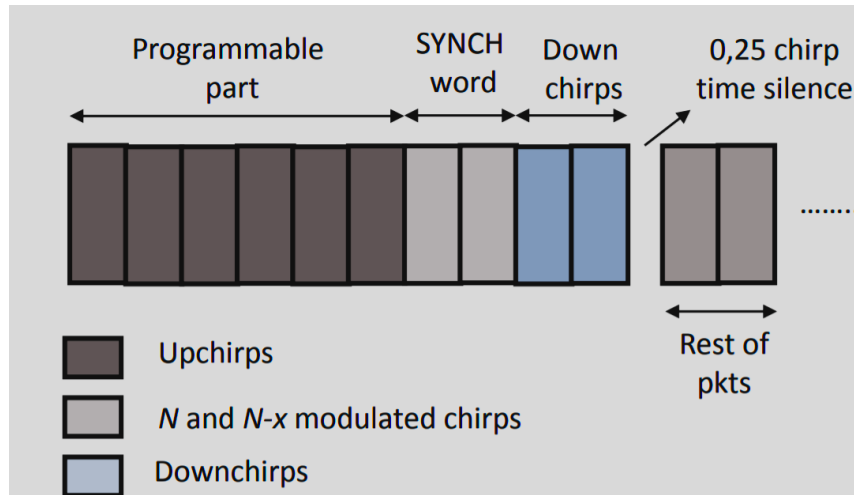


Figura 10 – Estrutura do *Preamble* LoRa. Fonte: Haxhibeqiri et al. (2018).

Em pacotes de *uplink*, este *preamble* consiste de *up-chirps* e o *payload* de *down-chirps*. Já em pacotes de *downlink* temos o contrário, onde o *preamble* consiste de *down-chirps* e o *payload* de *up-chirps*. A primeira janela de *chirps* deste *preamble* ilustrado na Figura 10 ajuda o receptor, mesmo que dessincronizado, a detectar e começar a receber este *preamble*. O receptor detecta a mudança brusca de frequência quando um *chirp* alcança a f_{max} (no caso de *up-chirps*) e a partir desta detecção o receptor se sincroniza ao transmissor e pode receber o pacote, como ilustra a Figura 11.

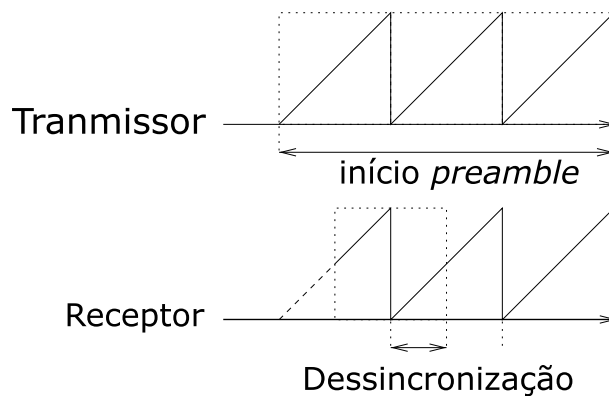


Figura 11 – Exemplo de recebimento de *Preamble* LoRa. Fonte: Haxhibeqiri et al. (2018).

2.4 LORAWAN

Para as camadas superiores da rede, o protocolo LoRaWAN se destaca como uma alternativa de código aberto apresentada pela LoRa Alliance, organização sem fins lucrativos fundada em 2015 por mais de 500 empresas parceiras com o objetivo de incentivar o desenvolvimento de LPWANs (LORA ALLIANCE, 2020). LoRaWANs geralmente implementam uma topologia em estrela entre *endpoints* e *gateways*, conforme ilustrado na Figura 12. Os *gateways* encaminham as informações, geralmente por meio de um link 3G/4G ou mesmo Ethernet, de/para um servidor de rede (NS) na nuvem, que é responsável

por retransmitir os dados de/para um servidor de aplicativos usando o protocolo TCP/IP e protocolos SSL. Nesse contexto, dois fluxos de informação podem ser distinguidos do ponto de vista do *endpoint*: (a) o fluxo *downlink*, com informações transmitidas do servidor de rede para o endpoint; e (b) o fluxo *uplink*, na direção oposta (MICROCHIP TECHNOLOGY, 2021).

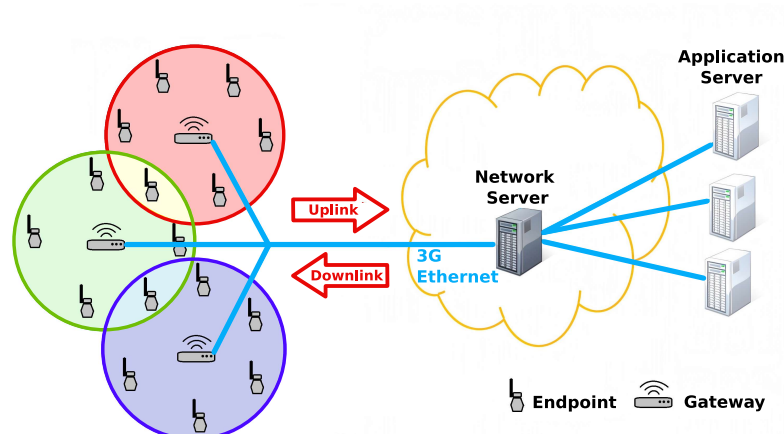


Figura 12 – Topologia estrela LoRaWAN entre *endpoints* e *gateways*. Adaptado de Sendra et al. (2020).

O protocolo de comunicação LoRaWAN é utilizado principalmente em soluções de *Internet of Things* (IoT) que implementam comunicação *Machine to Machine* (M2M), principalmente com sensores e atuadores de baixa complexidade. Foram definidos diversos padrões e métodos para facilitar o desenvolvimento e a escalabilidade de novas redes LoRaWAN, como classes de *endpoints* e definições de *data rate* de *uplink* e *downlink*. O *Data rate* (taxa de transmissão) de uma rede LoRa depende de parâmetros da rede como SF, largura de banda *BW*, taxa de bits e o tamanho do *payload* fixados para o envio. O *data rate* de envio de *uplink* pode ser fixo ou variável, dependendo da configuração no *endpoint*. Caso o *data rate* seja variável, é possível implementar alguns algoritmos de adaptação de *data rates*. Estes algoritmos são chamados de *Adaptive Data Rate* (ADR). Existe também um limite estabelecido para que o *payload* possa ser transportado pelo pacote LoRaWAN, o qual depende da região do mundo onde o protocolo LoRaWAN é utilizado. Isto ocorre pois cada região do mundo estabelece um tempo limite para o ToA permitido por regulamentação. Logo, caso ocorra uma alteração no SF (que aumenta o ToA) é esperado uma redução na taxa de transferência de *bits* máxima. Apesar da similaridade na denominação, o *data rate* e *bitrate* não são sinônimos. Enquanto o *bitrate* se trata de uma medida referente à taxa de quantidade de dados transmitidos durante um período de tempo, o *data rate* se trata de uma classificação, por exemplo: se um *endpoint* está transmitindo pacotes LoRa com SF9 e BW de 125 kHz, de acordo com a documentação da faixa de banda AU915-928 utilizada no Brasil, está utilizando o *data rate* 3. A Tabela 1 demonstra um exemplo de distintos *data rates* (DRs) e seus parâmetros

para a regulamentação brasileira “AU915-928”.

DR	SF	Freq. Canal	Up/Down	Bitrate (Bits/s)
0	SF12	125 kHz	Uplink	250
1	SF11	125 kHz	Uplink	440
2	SF10	125 kHz	Uplink	980
3	SF9	125 kHz	Uplink	1760
4	SF8	125 kHz	Uplink	3125
5	SF7	125 kHz	Uplink	5470
6	SF8	500 kHz	Uplink	12500
7			RFU	
8	SF12	500 kHz	Downlink	980
9	SF11	500 kHz	Downlink	1760
10	SF10	500 kHz	Downlink	3900
11	SF9	500 kHz	Downlink	7000
12	SF8	500 kHz	Downlink	12500
13	SF7	500 kHz	Downlink	21900
14..15			RFU	

Tabela 1 – *Data Rates* e seus parâmetros correspondentes da faixa AU915-928. Fonte: Semtech (2020).

A Figura 13 ilustra um possível mapeamento da tecnologia LoRa/LoRaWAN em um modelo OSI (*Open Systems Interconnection*) de camadas. A modulação LoRa está inserida na primeira camada do modelo OSI, a camada física, responsável pela transmissão física dos dados através de um meio de transporte. Neste caso esta transmissão é realizada pela conversão dos *bits* de informação em sinais de rádio. Já o protocolo LoRaWAN está inserido principalmente na segunda camada do modelo, a camada de enlace (*data link*), realizando o controle de acesso ao meio.

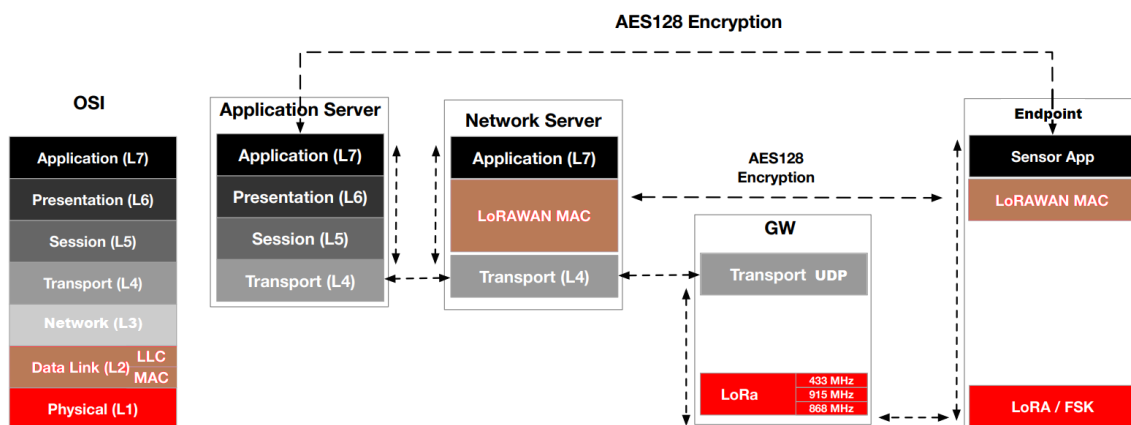


Figura 13 – LoRa/LoRaWAN no modelo OSI. Fonte: adaptado de Ertürk et al. (2019).

A modulação LoRa pode usar largura de banda de 125 kHz, 250 kHz e 500 kHz, resultando em diversos sub-canais dentro da frequência de operação de cada região. No

Brasil (AU915-928) é possível trabalhar em 64 canais diferentes (numerados de 0 a 63) para *uplink*, com largura de banda 125 kHz (separados incrementando 200 kHz de forma linear), iniciando em 915,2 MHz até 927,8 MHz, outros 8 canais (numerados de 64 a 71) para *uplink* com largura de banda 500 kHz (separados linearmente incrementando 1,6 MHz de forma linear), iniciando em 915,9 MHz até 927,1 MHz, e terminando com 8 canais (numerados de 0 a 7) para *downlink* com largura de banda 500 kHz (separados incrementando 600 kHz de forma linear), iniciando em 923,3 MHz até 927,5 MHz. Esta divisão de canais evidencia que o fluxo preferencial de comunicação da tecnologia LoRa/LoRaWAN é o fluxo de dados provindos de *endpoints* para um *network server*. A Figura 14 demonstra como são separados os canais dentro da faixa de frequência de operação da região AU915-928.

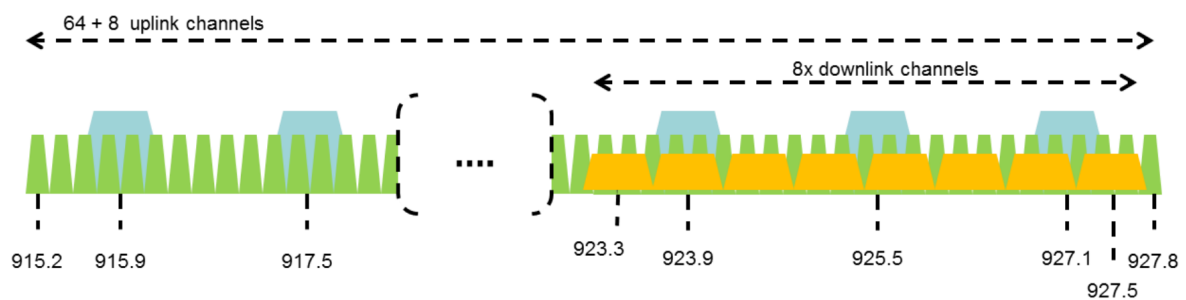


Figura 14 – Separação de canais para AU915-928. Fonte: LoRa Alliance (2019).

A separação em diversos canais, juntamente com a aleatoriedade de seleção de canal da transmissão de um *uplink* e também com a utilização de diferentes SFs é muito eficiente. Assim como dois pacotes enviados no mesmo canal com diferentes SFs não sofrem interferência mútua (exceto, por exemplo, quando dois pacotes de largura de banda distintos podem não ser ortogonais entre si), dois pacotes são enviados com o mesmo SF em canais diferentes também são livres de interferência mútua.

Classes pré-definidas são atribuídas aos *endpoints* para facilitar e padronizar uma nova configuração e ativação destes em uma rede LoRaWAN. As classes são definidas a partir da disponibilidade de recepção de janelas de *downlinks* vindos dos *gateways*, e são definidas juntamente com os padrões de cada região e de cada rede LoRaWAN. Existem 3 classes de *endpoints* configuráveis no protocolo, as classes A, B e C:

- Classe A: nesta classe os *endpoints* podem enviar pacotes livremente. Após cada transmissão o *endpoint* deve abrir uma janela de recepção (*downlink*) RX1. Caso o *endpoint* não receba um pacote na primeira janela de recepção, uma segunda janela de recepção RX2 é aberta. Vale ressaltar que, para esta classe, os *gateways* nunca iniciam a comunicação, a iniciativa é sempre dos *endpoints*, com uma mensagem de *uplink*. Então, caso haja a necessidade de realizar uma transmissão de *downlink*, seja por conta de um atuador em campo ou apenas a troca de uma configuração, deve-se esperar o *endpoint* iniciar a comunicação para realizar o *downlink*. Por conta disso,

esta classe é limitada a aplicações de baixíssima taxa de transmissão de dados, onde não há a necessidade de atualizações com alta periodicidade no *endpoint* em campo, em troca de economia de energia. A maior parte dos *endpoints* de classes A tem sua alimentação limitada por uma bateria, e quando não estão coletando informações ou transmitindo pacotes, estão em um estado de *sleep* para economizar energia. Todos os *endpoints* devem ser inseridos em uma rede como *endpoints* de classe A e podem ter suas classes alteradas posteriormente;

- Classe B: nela os *endpoints* podem abrir uma janela de recepção periódica curta chamada de *ping slot*. Para a realização da sincronização entre *gateways* e *endpoints* é realizada uma transmissão de um *beacon*, que praticamente se trata de um pacote LoRa, ilustrado na Figura 9, entretanto sem a presença do *Physical Header* e do CRC. O *beacon* então contém somente o *preamble* e o *payload* denominado de “*BCNPayload*”. Esta classe é raramente utilizada na indústria e nela os *endpoints* são também em maior parte alimentados por baterias;
- Classe C: nela os *endpoints* abrem as duas janelas de recepção RX1 e RX2 e mantêm a segunda delas aberta até o próximo *uplink*. Este comportamento surgiu da necessidade de dispositivos que estivessem o tempo todo disponíveis para receber pacotes de *downlink* de *gateways*. Com esta configuração de classe, é possível realizar o envio de *multicast downlinks* para diversos *endpoints*.

O comportamento temporal dos *endpoints* de classe A, B e C estão ilustrados na Figura 15.

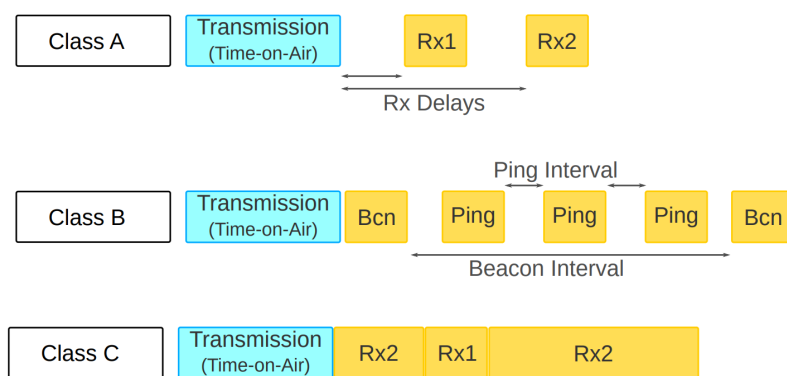


Figura 15 – *Time slots* de recepção dos *endpoints* de classe A, B e C. Fonte: Cotrim e Kleinschmidt (2020).

Como citado anteriormente, a estrutura de mensagem LoRa é composta por *preamble*, *PHY Header*, *PHY Header CRC*, *PHY Payload*, CRC (apenas para *uplink*), sendo o *PHY Header* e *PHY Header CRC* opcionais caso seja pré-definido na configuração da rede. O formato da mensagem de LoRaWAN é definido dentro do campo *PHY Payload* da

estrutura de mensagem de LoRa. Dentro dele existem 3 campos, o *MAC Header*, o *MAC Payload* ou *Join-Request* ou *Join-Response* e o *Message Integrity Code* (MIC).

As estruturas de *Join-request* e *Join-response* são estruturas usadas para realizar a ativação pelo ar dos *endpoints* (*Over the air activation*). O MAC Header contém o tipo de mensagem que está sendo transmitido e qual a versão de LoRaWAN usada. Os tipos de mensagem podem ser de *confirmed uplink*, *confirmed downlink*, *unconfirmed uplink*, *unconfirmed downlink*, além das *Join-request* e *Join-response* e outros reservados para desenvolvimentos futuros.

Para um *endpoint* iniciar uma comunicação em uma rede LoRaWAN, é necessário que o dispositivo seja ativado, tanto por questões de gestão de rede como por questões de segurança. Existem dois métodos de ativação em LoRaWAN, *Activation By Personalisation* (ABP) e *Over The Air Activation* (OTAA), o primeiro usado preferencialmente para desenvolvimento e o segundo para implantação da solução. Todas as classes de *endpoint* necessitam passar pelo processo de ativação. A Figura 16 mostra o encapsulamento de um pacote LoRaWAN, na qual destacado em vermelho está o campo “*FRMPayload*”, que se trata da limitação em *bytes* que depende da regulamentação local.

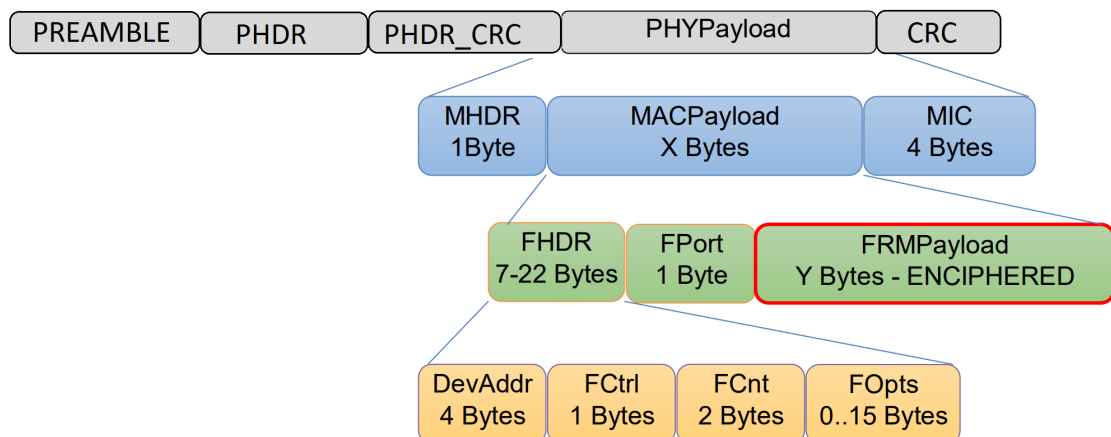


Figura 16 – Encapsulamento de um pacote LoRaWAN. Fonte: adaptado de Sisinni et al. (2019).

Uplink Frame DE APLICAÇÃO

Como mencionado anteriormente, o projeto que motivou o presente estudo visa estabelecer uma comunicação entre *wearable endpoints* LoRa e *gateways* espalhados em um canteiro de obras de um parque eólico. O *payload* de um pacote LoRaWAN de *uplink*, que corresponde à cama de aplicação, pode ser visto na Figura 17, contendo as seguintes informações: endereço MAC, *timestamp*, um bit para o tipo de pacote, informações GNSS (latitude, longitude e altitude), nível de bateria, um bit de emergência, versões de *firmware*

e *hardware* e dados da unidade de medição inercial (IMU). O pacote de *downlink*, neste momento, ainda não está definido.

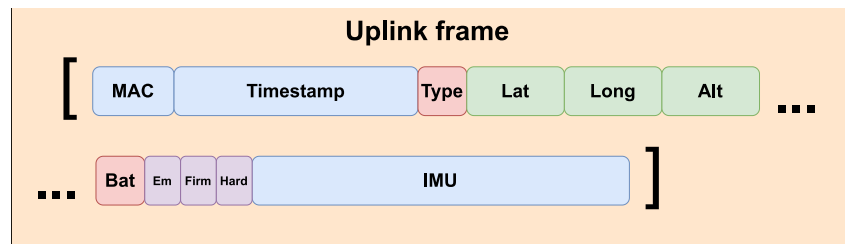


Figura 17 – *Frame* LoRaWAN de *uplink*.

RESUMO

Ao longo deste capítulo foram expostos os principais conceitos envolvidos em uma transmissão LoRa, desde a sua modulação até a utilização de protocolos como LoRaWAN para disponibilizar as informações transmitidas para aplicações em camadas superiores de rede. Com isto feito, podemos passar para a análise dos algoritmos de ADR existentes na literatura e a proposição de um novo algoritmo que utilize dados localização, enfoque desta dissertação de mestrado.

3 ALGORITMOS DE ADR EXISTENTES

A performance de redes de comunicação LoRaWAN pode ser mensurada através da taxa de extração de dados, do inglês *Data Extraction Rate* (DER), qual é definida como a razão entre o número de pacotes recebidos e de pacotes transmitidos durante uma janela de tempo (BOR et al., 2016). O DER ideal é 1, que é mais comum em redes de densidade baixa (e.g. 10 *endpoints* estáticos e 1 *gateway*). Entretanto, em redes com alta densidade ou com *endpoints* móveis (FONSECA et al., s.d.) o DER costuma decrescer.

Com o intuito de aumentar o DER ou ao menos tornar o consumo energético menor nos *endpoints*, é possível implementar em uma rede LoRaWAN um algoritmo ADR, que pode ser computado no *network server* e/ou no *endpoint*. O intuito com tais algoritmos é ajustar os parâmetros de transmissão LoRa, mais comumente alterando o SF e o TP (KUFKUNESU et al., 2020). Antes, porém, de discorrer sobre os algoritmos de ADR comumente encontrados na literatura, convém explicar brevemente o conceito de ruído no canal, que afeta a relação sinal-ruído (SNR) a qual, por sua vez, é uma medição fundamental para muitos desses algoritmos.

3.1 RUÍDO NO CANAL

Qualquer objeto com temperatura maior que zero absoluto emite radiação na forma de um ruído gaussiano aleatório, usualmente chamado de ruído térmico. Em dispositivos eletrônicos, a energia dos elétrons em materiais usados para se fazer resistores aumenta à medida que a temperatura aumenta. Este ruído térmico atua tanto nos *endpoints* como nos *gateways* e sua potência pode ser calculada como:

$$N = k.T.B \quad , \quad (3.1)$$

onde:

- N - Potência do Ruído (Watts);
- k - Constante de Boltzman ($1,38 \times 10^{-23}$ J/K);
- T - Temperatura (Kelvin);
- B - Largura de banda medida (Hz).

Além do ruído térmico, o receptor também emite um ruído próprio. Esta qualidade do ruído de um receptor pode ser expressada como *Noise Figure*, representando a diferença em dB entre o ruído de saída de um receptor real e um “ideal” (que não emite ruído). Esta diferença é principalmente causada pela amplificação na entrada do receptor. Um receptor de rádio é considerado excelente caso este valor seja menor do que 2 dB. O cálculo desta diferença pode ser feito da seguinte maneira (FRENZEL, 2016):

$$\text{Noise Figure} = SNR_{in} - SNR_{out} \text{ [dB]}, \quad (3.2)$$

onde SNR_{in} se trata da relação sinal-ruído (SNR, do termo inglês, *signal-to-noise ratio*) do sinal de entrada do receptor, enquanto o SNR_{out} refere-se ao SNR do sinal de saída do receptor.

3.2 ALGORITMOS PARA *ENDPOINTS* ESTÁTICOS

O algoritmo ADR-TTN, provido pela associação colaborativa *Internet-of-Things* chamada de *The Things Network*, é provavelmente o algoritmo mais popularmente conhecido (SEMTECH, 2016). No lado do *endpoint*, um “contador” monitora a conectividade do dispositivo, de forma que a cada transmissão o contador é incrementado e só retorna a zero quando um pacote de *downlink* (provindo do *gateway*) é recebido. Se este contador alcançar um valor previamente estipulado como limite (ADR_ACK_LIMIT), os próximos *uplinks* enviam pacotes requisitando um *acknowledgement* (ACK) do *gateway*. Caso este ACK não for recebido pelo *endpoint* após algumas novas transmissões (ADR_ACK_DELAY), o contador volta ao valor de ADR_ACK_LIMIT e o parâmetro de TP é aumentado se já não estiver em seu valor máximo. Caso o valor de TP já tenha alcançado seu limite, o valor de SF é aumentado. A ideia do ADR-TTN é de lidar com o desvanecimento do sinal preferencialmente aumentando o TP, e se isso não for o suficiente, aumentando o SF. Este processo é repetido até que um ACK seja recebido ou que o *endpoint* alcance o SF máximo. Os valores de ADR_ACK_LIMIT e ADR_ACK_DELAY são configuráveis e iguais respectivamente a 64 e 32 como padrão.

O algoritmo ADR-TTN também pode ser computado no *network server* (NS), o qual avalia a qualidade da comunicação entre cada *endpoint* e *gateway* com base nos dados de SNR medido em cada *gateway*. O algoritmo calcula a quantidade de passos de ajuste (N_{steps}) em TP (incremento e decremento em passos de 3 dBm) e SF (somente decrementos em passos de 1) são necessários para alcançar uma comunicação estável (SNR suficiente). Este cálculo é realizado levando em conta os valores tabelados de mínimo SNR de cada SF (SNR_{SF}) para que o sinal seja corretamente demodulado, conforme Tabela 2.

SF	SNR
6	-5 dB
7	-7,5 dB
8	-10 dB
9	-12,5 dB
10	-15 dB
11	-17,5 dB
12	-20 dB

Tabela 2 – SNR requerido para recepção correta de sinal. Adaptado de Semtech (2019).

Além disso também é considerado no cálculo de N_{steps} uma margem de erro ($margin_{db}$) utilizada para compensar o erro na estimação da condição do canal, que é normalmente configurada como 10 dB. Por fim, também é considerado no cálculo o valor máximo (SNR_{max}) entre as medições de SNR das últimas N transmissões, resultando no seguinte cálculo:

$$N_{steps} = \lfloor \frac{SNR_{max} - SNR_{SF} - margin_{db}}{3} \rfloor. \quad (3.3)$$

Por exemplo, o *network server* está recebendo pacotes de um *endpoint* com SF10 (cujo valor tabelado mínimo de SNR é -20 dB) e o valor máximo das últimas 20 transmissões foi de -7 dB. Portanto, se a margem de erro for considerado como 10 dB o valor calculado de N_{steps} será 1. De forma geral, o valor de N_{steps} pode ser tanto negativo como positivo. Quando positivo expressa que os valores de SNR medidos estão mais altos que o necessário e que tanto o SF como o TP podem ser diminuídos de forma a reduzir o consumo energético. Entretanto, quando negativo representa que os valores de SNR medidos estão abaixo do esperado e que portanto o TP pode ser amplificado para melhorar o SNR.

O parâmetro de SF só pode ser decrementado no algoritmo computado no NS para que seja evitado o uso indevido de SF maiores (possivelmente escolhidos pelo algoritmo computado no *endpoint*), resultando em *bitrates* menores. Portanto, se o NS detectar uma conexão com SNR abaixo do esperado, ele só tem a liberdade de alterar o TP. Desta forma os *endpoints* são compelidos a explorar todos os valores de TP enquanto usam o SF7, e só podem usar SF maiores caso já estejam com valor máximo de TP. Isto acontece pois, do ponto de vista de conectividade, utilizar um TP menor com um SF maior é menos eficiente energeticamente do que utilizar um TP maior com um SF menor (JESUS et al., 2021). O comportamento do algoritmo ADR-TTN está ilustrado na Figura 18, onde o NS acumula os valores de SNR referentes a cada *endpoint*, realiza o cálculo de N_{steps} e realiza a alteração dos parâmetros de SF e TP.

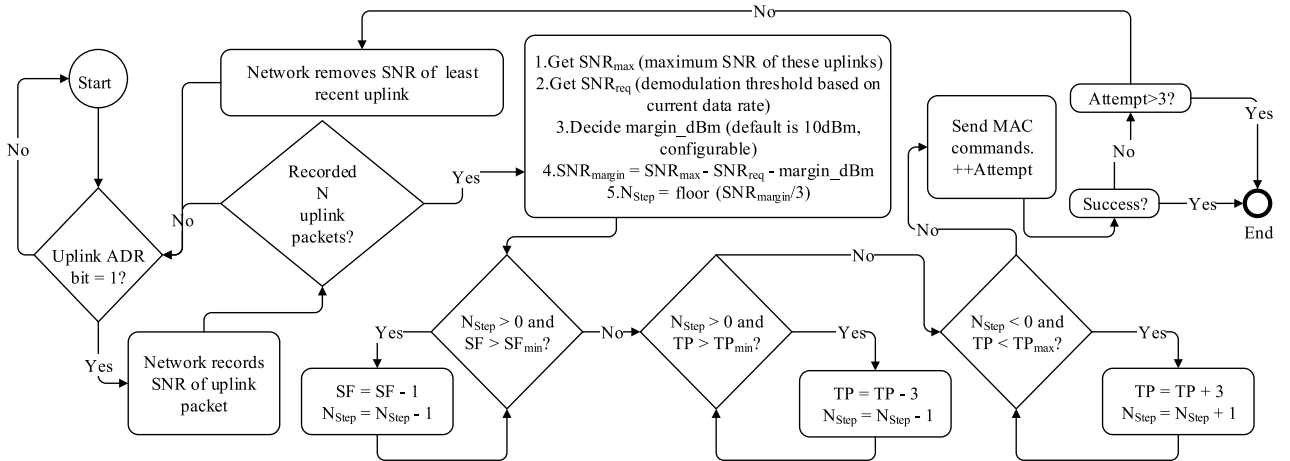


Figura 18 – Fluxograma do algoritmo ADR-TTN executado no *network server*. Adaptado de Li et al. (2018).

O algoritmo ADR_+ é similar ao ADR-TTN, exceto pelo seu método de estimação de canal. Enquanto o algoritmo ADR-TTN considera SNR_{\max} como o valor máximo de SNR das últimas N transmissões, o algoritmo ADR_+ utiliza a média do SNR das últimas N transmissões (SLABICKI et al., 2018). O ADR_+ tem uma performance melhor do que o ADR-TTN na maioria das situações em termos de eficiência energética e DER devido ao fato de o primeiro ser mais conservador na alocação de recursos, utilizando o pior SNR entre os últimos pacotes recebidos como base para calcular novos parâmetros de SF e TP.

Com o propósito de aumentar o DER de uma rede LoRaWAN, em Cuomo et al. (2017) propuseram dois algoritmos: ExploRa-SF e ExploRa-AT. O primeiro utiliza uma heurística com o intuito de alocar igualmente todos os SFs aos *endpoints* no alcance de um *gateway*. Isso funciona da seguinte forma: ao grupo de *endpoints* mais próximo do *gateway* (GW), o que é determinado em termos de *received signal strength indication* (RSSI), é alocado um SF menor, enquanto *endpoints* mais distantes operam com SF maiores. Esta divisão leva a uma distribuição que maximiza a ortogonalidade entre os possíveis SFs, entretanto não é uma alocação justa de recursos, tendo em vista que, por exemplo, um pacote de SF7 é transmitido na metade do tempo de um com SF8. Para uma alocação mais justa em relação ao ToA, o ExploRa-AT tenta equalizar o ToA médio entre os distintos SFs. Como o mesmo pacote transmitido em SF8 leva o dobro do tempo para ser transmitido do que em SF7, o algoritmo ExploRa-AT leva a uma distribuição com um número menor de *endpoints* operando em SF maiores. Um exemplo de alocação de SF de ambos algoritmos pode ser visto na Figura 19.

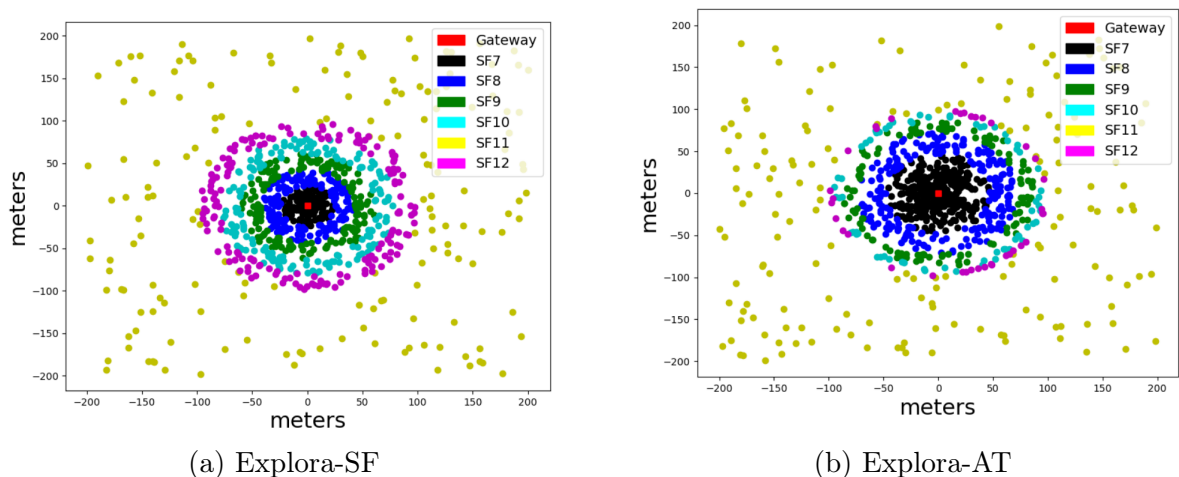


Figura 19 – Exemplo de alocação de SF dos algoritmos ExploRa. Fonte: Cuomo et al. (2017).

3.3 ALGORITMOS PARA *ENDPOINTS* MÓVEIS

Além dos algoritmos previamente mencionados, que foram projetados para *endpoints* estáticos, existem algoritmos que foram concebidos para cenários mais universais, considerando a mobilidade dos *endpoints*. Por exemplo, dois algoritmos foram apresentados por

Farhad et al. (2020): o algoritmo G-ADR e o EMA-ADR. No primeiro, um filtro gaussiano é implementado para amenizar os valores de SNR recebidos no NS de forma a reduzir o período de convergência do algoritmo ADR. O algoritmo G-ADR consiste basicamente nos seguintes passos:

- Quando o NS recebe um pacote de *uplink* com o bit *ADRACKReq* ele começa a guardar os valores de SNR dos pacotes, de forma semelhante ao algoritmo ADR-TTN mencionado na seção 3.2. O algoritmo G-ADR é então iniciado e computa as seguintes funções de média (μ) e variância (σ):

$$\mu = \frac{1}{M} \sum_{i=1}^M SNR_i, \quad (3.4)$$

$$\sigma^2 = \frac{1}{M-1} \sum_{i=1}^M (SNR_i - \mu)^2, \quad (3.5)$$

onde i corresponde ao número do pacote dentro de um total de $M = 20$ pacotes.

- São selecionados os valores de SNR que estão centralizados na distribuição dentro do intervalo $\mu + \sigma$ e $\mu - \sigma$. O valor de SNR é computado como a média entre estes valores selecionados.
- Por fim, é obtido o SNR requerido (SNR_{SF}), conforme a Tabela 2, e computado o valor de N_{steps} utilizando a equação 3.3. Com este valor, é realizada a alteração dos parâmetros de SF e TP de forma semelhante ao ADR-TTN.

Já o algoritmo EMA-ADR implementa um filtro de média móvel exponencial para reduzir os picos dos valores medidos de SNR. Este filtro, do inglês *exponential moving average* (EMA), realiza o cálculo de média móvel com pesos para cada valor de SNR dentro dos M pacotes recebidos pelo NS. É comum que valores de SNR mudem ao longo do tempo, mesmo em um ambiente fixo, devido a fatores de ruído, desvanecimento do sinal e interferência. Desta forma, uma média móvel exponencial pode ser computada e expressa de forma:

$$S(k) = \begin{cases} Y(k), & k = 1; \\ \beta \cdot Y(k) + (1 - \beta) \cdot S(k - 1), & k > 1, \end{cases} \quad (3.6)$$

com k pertencente ao conjunto dos números naturais sendo a amostra de tempo, $Y(k)$ se refere ao valor atual de SNR na amostra de tempo k , $S(k)$ denota o valor da média móvel exponencial também em qualquer amostra de tempo k e o parâmetro β é um fator de suavização ($0 < \beta < 1$) que pode ser ajustado.

O uso do algoritmo EMA-ADR resulta em uma suavização nos valores de SNR recebidos pelo NS, como demonstrado na Figura 20.

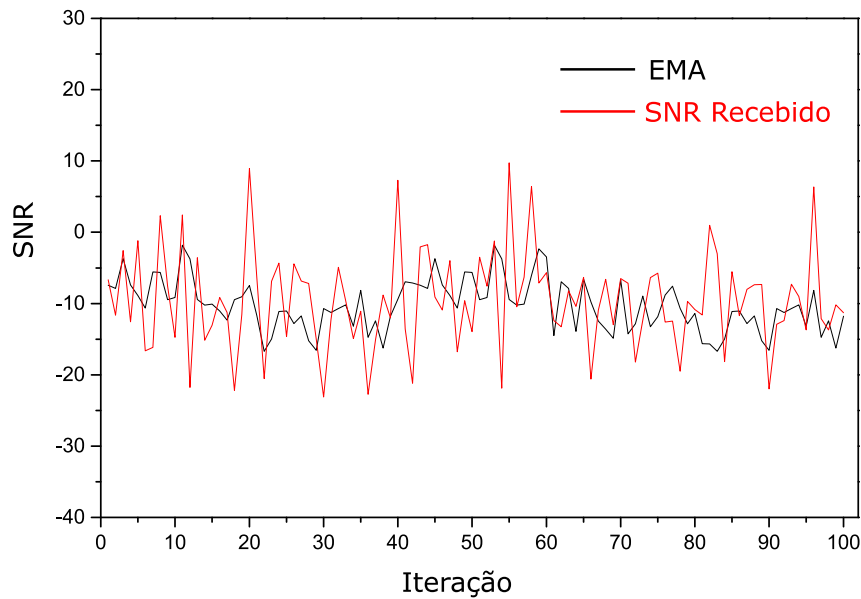


Figura 20 – SNR recebido pelo NS e SNR filtrado pelo algoritmo EMA-ADR. Fonte: Farhad et al. (2020).

Além destes algoritmos, há também o E-ADR, introduzido por BENKAHLA et al. (2021), que utiliza um modelo oculto de Markov para prever a próxima posição de um *endpoint* móvel. O algoritmo E-ADR propõe que cada *endpoint* deve especificar o seu padrão de mobilidade quando se conecta ao NS. A partir disso, o NS utiliza um método de trilateração para estimar a posição do *endpoint* e as próximas n posições baseado em um algoritmo de regressão linear. Com a posição estimada, o NS calcula a distância entre o *endpoint* e os *gateways* para escolher os valores de SF, CR, BW, entre outros parâmetros.

Duas distintas abordagens podem ser vistas nos algoritmos apresentados previamente. O algoritmo G-ADR e o EMA-ADR tentam ajustar o SNR medido, estratégia previamente utilizada nos algoritmos de *endpoints* estáticos ADR-TTN e ADR+. Entretanto o algoritmo E-ADR estima a posição do *endpoint* para então alocar os parâmetros de transmissão (SF, BW, CR, etc.).

4 ALGORITMO ADR PROPOSTO: DISTANCE-ADR

Como será demonstrado no capítulo 6, os valores de DER para os algoritmos de ADR avaliados caem significativamente quando a mobilidade dos *endpoints* é considerada. Na tentativa de mitigar esse problema, um novo algoritmo será proposto a seguir, denominado *Distance-ADR* (ou simplesmente D-ADR).

Como previamente mencionado, no cenário do parque eólico os *endpoints* são equipados com um receptor GNSS que fornece dados de geolocalização, os quais são enviados nos pacotes LoRa para o NS. Baseado nesta geolocalização, um algoritmo de ADR foi desenvolvido, capaz de ser computado tanto no NS quanto no *endpoint*. Assim como o algoritmo G-ADR mencionado na seção 3.3, que amplia o algoritmo ADR-TTN de modo a não ser suscetível a bruscas variações no SNR, a principal ideia do algoritmo D-ADR que é computado no NS é a de, no lugar de guardar os valores de SNR, armazenar os valores de posição dos *endpoints*. O comportamento do algoritmo é simples, como demonstra o fluxograma na Figura 21. Caso o NS receba um *uplink* com o bit de *Uplink ADR* em 1, significando que o *endpoint* espera que o NS envie parâmetros de ADR periodicamente, os valores de posição começam a ser armazenados pelo NS até que um certo número N de amostras seja alcançado. Após isso, o algoritmo D-ADR prediz onde o *endpoint* vai estar na próxima janela de N amostras e calcula a distância entre esta posição predita e os *gateways* (cuja posição já é conhecida). Com estas distâncias calculadas, o NS pode decidir se uma alteração nos parâmetros de SF e TP é necessária e envia um *downlink* para o *endpoint* com os parâmetros calculados.

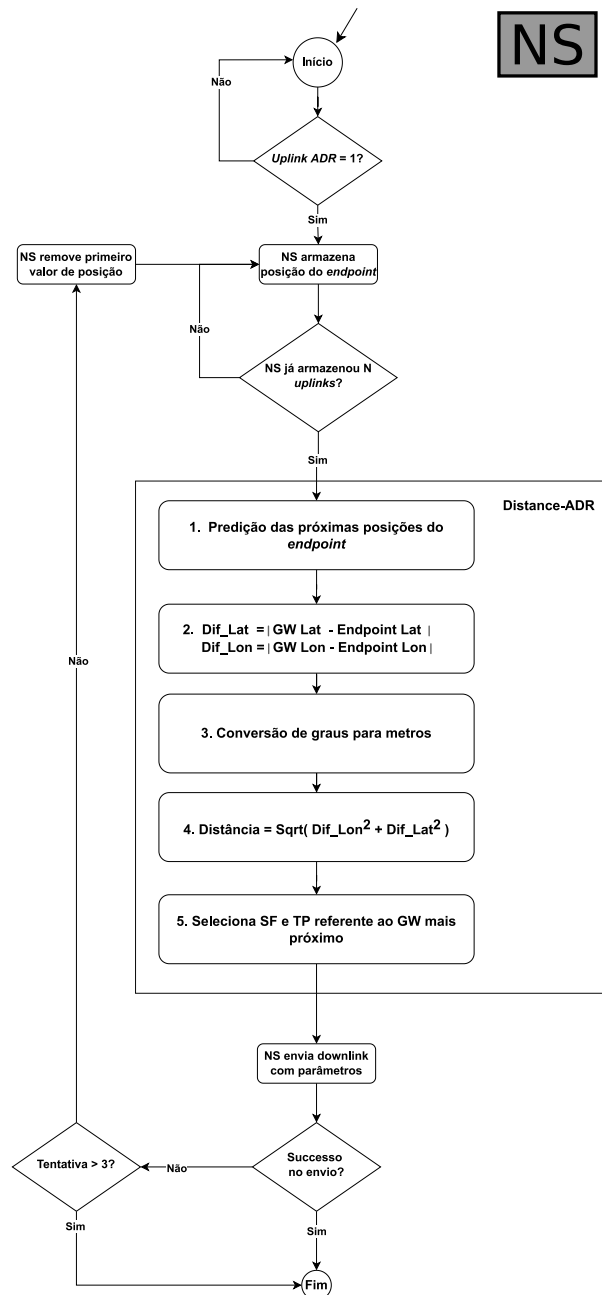


Figura 21 – Fluxograma do algoritmo D-ADR no lado do *network server*.

A seleção de SF e TP elucidada no passo 5 do fluxograma na Figura 21 é realizada baseada no valor de distância entre *endpoint* e *gateway* e respeita as funções presentes na Figura 22. Estas funções foram levantadas de maneira empírica, considerando um ambiente com poucos obstáculos entre os nodos da rede, a modo de completamente explorar os parâmetros de SF7 até SF12 e potência de transmissão de 4 dBm até 14 dBm. Entretanto, caso o ambiente fosse, por exemplo, uma área com a presença de prédios e outros obstáculos, ajustes teriam que ser realizados nas funções, como por exemplo aumentar o piso de TP de 4 para um valor maior, tendo em vista que um pacote transmitido com baixa potência em uma zona com muitos obstáculos pode apresentar dificuldades em alcançar o *gateway*.

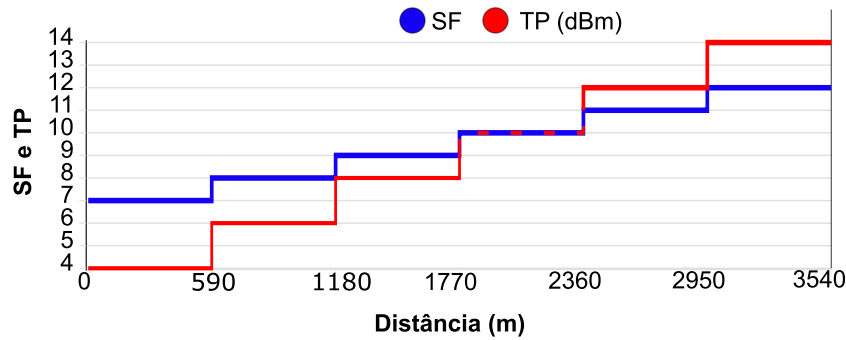


Figura 22 – Atribuição de SF e TP em função da distância no algoritmo D-ADR.

Existem alguns benefícios na implementação centralizada do algoritmo D-ADR no lado do NS. Talvez o mais eminente deles seja que esta centralização permite observar um critério de justiça na alocação de recursos. Por exemplo, uma regra simples de justiça pode ser implementada pelo NS de modo a limitar a quantidade de *endpoints* que podem operar com um determinado SF, de maneira a evitar o congestionamento do mesmo (colisão de pacotes).

Entretanto, algumas desvantagens também podem ser elencadas com relação à implementação centralizada do algoritmo D-ADR no lado do NS. Primeiramente, ela resulta em uma maior utilização do meio de comunicação tendo em vista que periodicamente o NS deve enviar um *downlink* para todos os *endpoints*. Em uma rede de baixa densidade (com poucos *endpoints*) isto não seria um problema, entretanto em uma rede com alta densidade, como no caso do problema motivador desta dissertação, isso pode gerar colisões de pacotes, piorando o desempenho da rede. Outro ponto que pode ser levantado é que o valor N de amostras necessárias deve ser ajustado para que o algoritmo D-ADR tenha uma dinâmica capaz de acompanhar o movimento dos *endpoints*. Por exemplo, se N for considerado como 20 e o *endpoint* transmitir um pacote LoRa a cada 60 segundos, o NS tomará no mínimo 20 minutos para alterar os parâmetros do *endpoint*. Considerando a velocidade de uma caminhada de 5 km/h é possível que um *endpoint* se desloque do exterior do parque eólico ilustrado na Figura 4 até o centro do mesmo ao longo desses 20 minutos. Para tornar o algoritmo mais dinâmico, uma estratégia de janela deslizante pode ser utilizada. Por exemplo, o algoritmo pode armazenar as últimas 20 posições de um *endpoint* e realizar um cálculo de novos valores de SF e TP a cada 5 pacotes recebidos pelo NS. Outra estratégia seria a de reduzir o valor de N , entretanto ambas opções gerariam uma ainda maior utilização do meio de comunicação, resultante do envio mais frequente de *downlinks* por parte do NS.

Por estes motivos, uma abordagem distribuída foi desenvolvida com o algoritmo D-ADR sendo computado no *endpoint*, cujo funcionamento segue de acordo com o fluxograma na Figura 23. A funcionalidade é simples: a cada nova requisição de *uplink* o algoritmo D-ADR calcula a diferença em latitude e longitude entre *endpoint* e cada *gateway*, considerando

que as posições de todos os *gateways* são programadas previamente no *software* que é computado nos *endpoints* e podem ser alteradas através de atualizações de *firmware*. Após isso o algoritmo converte essas diferenças para metros e calcula as distâncias utilizando o teorema de Pitágoras. Por fim, os parâmetros de SF e TP são escolhidos baseados na distância ao *gateway* mais próximo, no passo 4, de acordo com as funções estabelecidas na Figura 22.

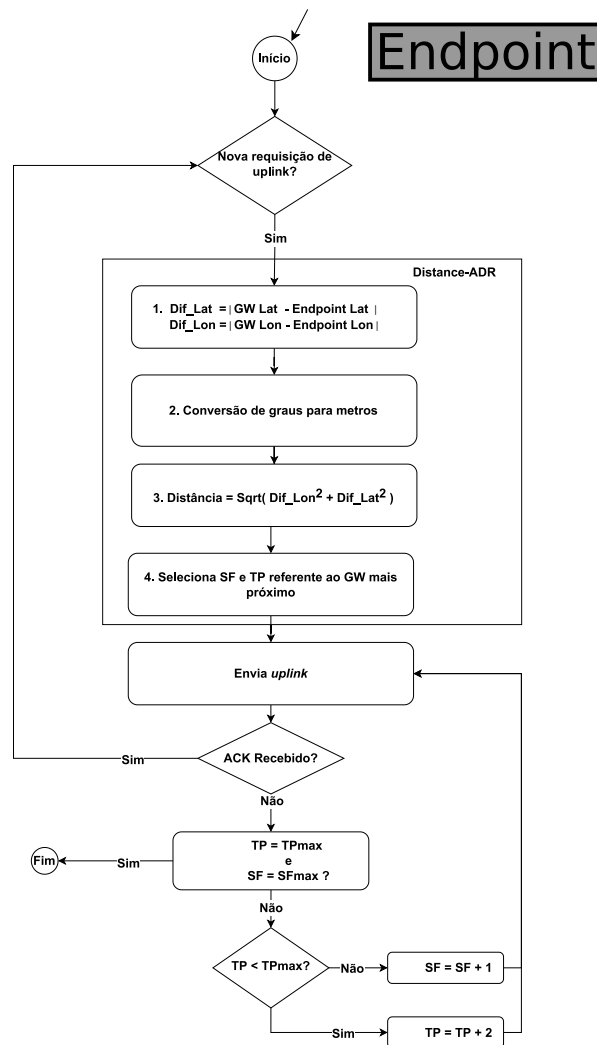


Figura 23 – Fluxograma do algoritmo D-ADR no lado do *endpoint*.

O restante desta dissertação considera a implementação distribuída do algoritmo D-ADR no lado do *endpoint* devido aos motivos citados e também devido ao fato que a implementação em código para a realização de testes em campo só seria possível desta maneira, tendo em vista que o acesso ao *firmware* do NS seria de difícil acesso, em contraste à facilidade de programação do *firmware* dos *endpoints*.

5 MODELAGEM E AMBIENTE DE SIMULAÇÃO

Para possibilitar a realização de uma simulação de rede LoRa é necessário modelar corretamente alguns fenômenos importantes, como o desvanecimento de sinal (*Path loss*).

5.1 PATH LOSS

A atenuação do sinal é um fenômeno de alta importância em telecomunicações, sendo um dos fatores preponderantes na determinação da potência necessária de um sinal. Modelos de propagação de sinais podem ser classificados como empíricos ou estocásticos. Os modelos empíricos são baseados em medições práticas, como os modelos Okumura, HATA, COST-231 HATA (OKUMURA, 1968; HATA, 1980; DAMOSSO; CORREIA et al., 1999) e muitos outros. Já os modelos estocásticos necessitam de uma quantidade massiva de informação geométrica sobre o ambiente e também de um poder computacional elevado para a realização de cálculos. Alguns destes modelos estocásticos são os modelos de Ray-Tracing, Ikegami e outros.

A atenuação de um sinal se dá pela redução na sua potência durante a transmissão. Transmissões por radiofrequência são atenuadas por três fenômenos físicos:

- Reflexão;
- Difração;
- Espalhamento.

O expoente de *path loss* “*n*”, visto na Eq. 5.2, indica o quanto um sinal é enfraquecido com o aumento da distância percorrida por ele. Este expoente normalmente se encontra entre o valor de 2 e 6 (onde 2 se refere a propagação em espaço livre e 6 a ambientes com muitos obstáculos). Este expoente deve ser único e diferente para cada tipo de ambiente, englobando os três fenômenos físicos listados acima.

5.1.1 MODELO EM ESPAÇO LIVRE

A atenuação de sinal em espaço livre ideal pode ser calculada como:

$$L = 20 \cdot \log \frac{4\pi D}{\lambda} , \quad (5.1)$$

onde:

- $L = Path\ loss$ (dB);
- $\lambda =$ Comprimento da onda portadora de frequência (metros);
- $D =$ Distância entre transmissor e receptor (metros).

Entretanto, a equação 5.1 descreve somente a perda de propagação em linha reta (*line-of-sight*), sem obstáculos. Em grande parte das aplicações existem obstáculos como prédios, muros, árvores, entre outros. Nessas situações a perda por *path loss* é consideravelmente maior. De acordo com Zyren e Petrick (1998), as perdas por propagação em ambientes com número elevado de obstáculos pode chegar a 1 dB por metro, enquanto que em campo aberto é em torno de 0,2 dB por metro.

Além do *path loss* existe outro fator, denominado *multipath*, que também causa degradação no sinal. Este fenômeno faz com que o sinal transmitido percorra um trajeto diferente e interfira com si mesmo, prejudicando a sua adequada recepção. Este problema pode ser amenizado aumentando a potência de transmissão e garantindo uma sensibilidade do receptor suficiente. O enfraquecimento por *multipath* pode diminuir o sinal consideravelmente e a quantidade necessária de potência de transmissão para superar este enfraquecimento é chamada de margem de desvanecimento. O enfraquecimento do sinal também pode ser causado pelo efeito Doppler e pelo desvio de frequência.

5.1.2 MODELO LOG-DISTANCE

O modelo *Log-distance* de *path loss* é uma extensão ao modelo em espaço livre. Este modelo é utilizado para prever a perda por propagação em uma grande amplitude de ambientes enquanto o modelo em espaço livre necessita de um percurso completamente livre de obstáculos entre transmissor e receptor. Mesmo em um cenário de *line-of-sight* (LoS) irá existir perdas por reflexão em obstáculos de grande porte como prédios por exemplo. O modelo *Log-distance* leva essas perdas em consideração e é representado pela expressão:

$$L_p(d) = L_p(d_0) + 10n \log \frac{d}{d_0} + X, \quad (5.2)$$

onde:

- d = Distância entre transmissor e receptor (metros);
- d_0 = Distância referência (metros), usualmente 1 m para transmissões curtas e 1 km para longas;
- X - Variável de distribuição média Gaussiana (dB), refletindo a atenuação causada por *flat fading*;
- n = Expoente de *path loss* representando a velocidade em que o sinal é enfraquecido com o aumento da distância (número racional positivo).

5.1.3 MODELO OKUMURA

Trata-se de um dos modelos mais populares, muito utilizado para áreas urbanas como modelo de propagação. Este modelo foi estabelecido empiricamente com a coleta de dados na cidade de Tokyo e pode ser expressado da seguinte maneira:

$$L_m = L_f(d) + A_{mu}(f, d) - H_{MG} - H_{BG} - K_{correction} , \quad (5.3)$$

onde:

- d = distância entre emissor e receptor (metros);
- f = frequência do sinal (MHz)
- $L_m = Path\ loss$ (dB);
- $L_f = Path\ loss$ em espaço livre (dB) dada pela Eq. 5.1;
- A_{mu} = Atenuação mediana relativa ao espaço livre (dB), levantada empiricamente em curvas como função de f e d ;
- H_{MG} = Fator de ganho da altura da estação móvel (dB), que neste caso é um *endpoint*;
- H_{BG} = Fator de ganho da altura da estação de base (dB), que neste caso é um *gateway*;
- $K_{correction}$ = Fator de correção dependente do tipo de ambiente (dB).

5.1.4 MODELO HATA

O modelo Hata é uma formulação empírica baseada nos resultados obtidos do modelo Okumura (OKUMURA, 1968), portanto também é conhecido como modelo Okumura-Hata. Entretanto o modelo Hata não pode ser aplicado com todas as frequências cobertas pelo modelo Okumura, que suporta frequências de até 1920 MHz. O modelo Hata é válido quando:

- Frequência de portadora (f_c) : $150\text{ MHz} \leq f_c \leq 1500\text{ MHz}$;
- Altura da antena da estação de base (h_b) : $30\text{ m} \leq h_b \leq 200\text{ m}$;
- Altura da antena da estação móvel (h_m) : $1\text{ m} \leq h_m \leq 10\text{ m}$;

Com essas condições respeitadas o modelo elenca três equações de *path loss*, uma para cada tipo de ambiente:

$$L = A + B \log(d - E) \quad \text{Presença de muitos obstáculos,} \quad (5.4)$$

$$L = A + B \log(d - C) \quad \text{Presença menor de obstáculos,} \quad (5.5)$$

$$L = A + B \log(d - D) \quad \text{Sem a presença de obstáculos,} \quad (5.6)$$

onde A, B, C, D e E são as seguintes funções matemáticas que podem considerar como entrada os parâmetros de frequência portadora e a altura das estações de base e móvel:

$$A = 69.55 + 26.16 \log(f_c) - 13.82 \log(h_b) , \quad (5.7)$$

$$B = 44.9 - 6.55 \log(h_b) , \quad (5.8)$$

$$C = 5.4 + 2 \left[\log \left(\frac{f_c}{28} \right) \right]^2 , \quad (5.9)$$

$$D = 40.94 + 4.78 [\log(f_c)]^2 - 18.33 \log(f_c) , \quad (5.10)$$

$$E = 3.2 [\log(11.75 h_m)]^2 - 4.97 , \quad (5.11)$$

5.2 SIMULAÇÃO DE REDES LORAWAN

Para qualificar a utilização da tecnologia LoRa em um ambiente extenso e com um número elevado de *endpoints* os seguintes simuladores foram considerados: FLoRa (SLABICKI; PREMSANKAR, 2018), ns-3 (MAGRIN et al., 2017), LoRaSim (BOR et al., 2016).

O simulador FLoRa trata-se de uma expansão ao simulador orientado a eventos discretos amplamente utilizado OMNeT++. Além disto o simulador utiliza o *framework* INET que possibilita, entre outras coisas, a programação de mobilidade dos *endpoints*. A interface gráfica do simulador pode ser visualizada na Figura 24, onde temos presentes, a título de exemplo, quatro *gateways*, que recebem informações de dois mil *endpoints*.

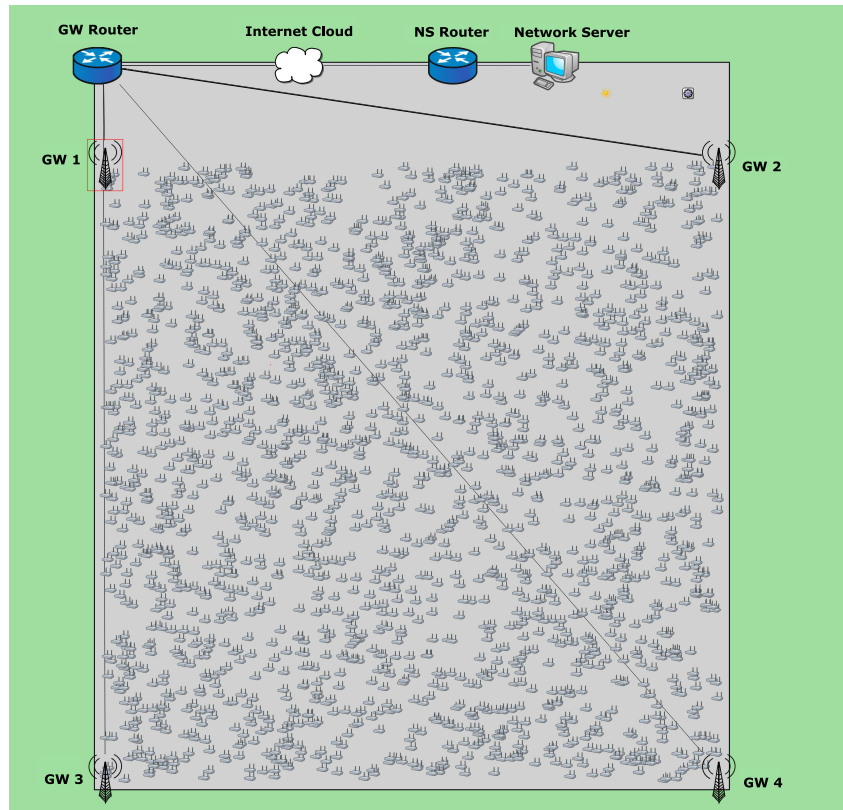


Figura 24 – Interface gráfica do ambiente de simulação FLoRa.

Com o *software* FLoRa é possível configurar o modelo de *path loss*, alterar livremente o número de *endpoints* e *gateways* assim como o período para envio de pacotes e diversos outros parâmetros do sistema de comunicação LoRa/LoRaWAN. O simulador também realiza o cálculo de gasto energético dos *endpoints* e os emite em um relatório após o término das simulações. Com o simulador FLoRa é possível realizar testes de carga (e.g aumentar gradualmente a frequência de envio de pacotes) e testes de sensibilidade para avaliar como o sistema desempenha com a variação de determinados parâmetros.

Similar ao OMNeT++, o ns-3 é um simulador orientado a eventos discretos que possibilita a utilização de módulos adicionais referentes a distintas redes de comunicação. Um destes módulos se refere ao de redes LoRaWAN, este possibilita a simulação de dispositivos LoRa de classe A e de múltiplos *gateways*. Além disso este módulo possibilita a implementação de algoritmos ADR e assim como o simulador FLoRa realiza o cálculo de gasto energético dos *endpoints*.

Por último, o simulador LoRaSim é desenvolvido em python com o intuito de analisar a escalabilidade de redes LoRa. Este simulador possibilita a simulação de *endpoints* LoRa se comunicando com até quatro *gateways*. Diferente de ambos os simuladores mencionados previamente, o LoRaSim não prevê o cálculo de gasto energético dos *endpoints*.

Devido a natureza móvel dos *endpoints* e a facilidade de implementar esta mobilidade utilizando o *framework* INET, o simulador FLoRa foi escolhido para realizar as simulações de análise de escalabilidade e referentes ao desempenho de diferentes algoritmos ADR.

MODELO *path loss* IMPLEMENTADO

O modelo Hata foi escolhido para simulação do parque eólico uma vez que este modelo possui equações específicas para a situação em campo aberto, que se aproxima à de um parque eólico. Entretanto o cenário de parque eólico foi modelado, devido a limitações no simulador, como um plano 2D desconsiderando a altura das estações móveis e de base.

5.3 MOBILIDADE DOS *ENDPOINTS*

Como padrão, o simulador FLoRa considera os *endpoints* como estáticos, uma suposição aceitável para grande parte das aplicações LoRa. Entretanto, em cenários como o que motivou o desenvolvimento desta tese, nos quais os *endpoints* se movem é necessário a implementação desta mobilidade no simulador. Para isto foi utilizado o *framework* INET e atribuída três variáveis para cada *endpoint*: velocidade, ângulo e aceleração. A velocidade foi configurada como 1 m/s para representar a cadência de uma caminhada, o ângulo é gerado aleatoriamente no início da simulação e a aceleração é considerada como 0 m/s². Além disso, é estabelecida uma área de operação de modo que, quando os *endpoints* alcançam os limites dessa área, eles invertem a sua direção de movimento e portanto permanecem dentro dela. Por exemplo, se um *endpoint* está se movendo com um ângulo de 45° e alcança o limite superior da área de operação, o ângulo se tornará -45° (ou 315°) e o *endpoint* se manterá dentro dos limites da área. Este comportamento está representado na Figura 25.

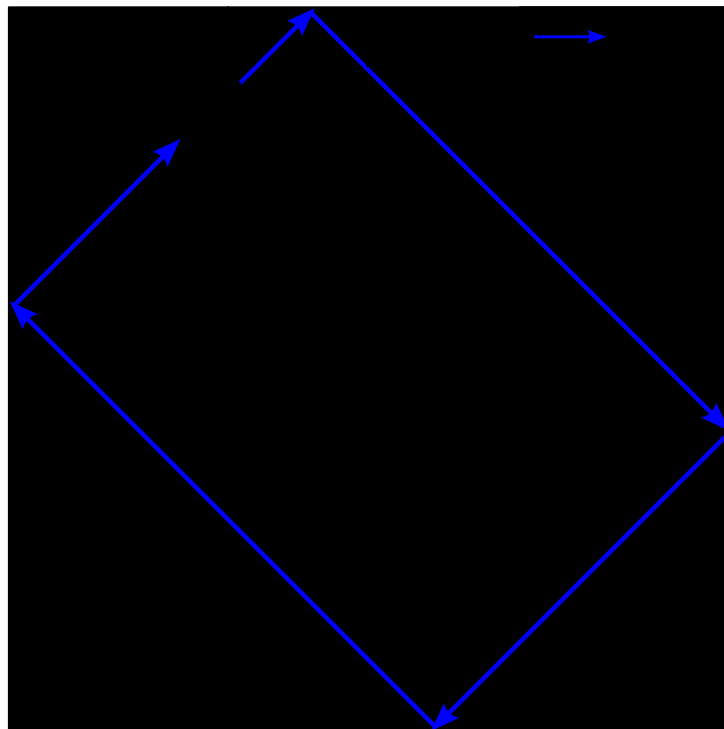


Figura 25 – Comportamento da mobilidade do *endpoint*.

6 RESULTADOS DE SIMULAÇÃO

Antes de avaliar o desempenho dos algoritmos ADR, é de interesse analisar os limites da comunicação via LoRa no ambiente de simulação modelado para avaliar a viabilidade no uso do LoRa como mencionado no capítulo 1. Após esta análise prévia (onde nenhum algoritmo ADR é implementado), os algoritmos serão então comparados tanto em cenários com *endpoints* estáticos quanto móveis. Durante todas as simulações realizadas alguns parâmetros foram mantidos, como ilustra a Tabela 3.

Parâmetro	Descrição	Valor
CR	<i>Coding Rate</i>	4/8
BW	Largura de banda	125 kHz
PL(d_0)	Desvanecimento a uma distância d_0	109,21 dB
d_0	Distância d_0	1000 m
pa	Tamanho do <i>preamble</i>	8 bytes
pl	Tamanho do <i>payload</i>	20 bytes
<i>Shadowing</i>	Variância de <i>Shadowing</i>	0

Tabela 3 – Parâmetros de simulação

6.1 ALCANCE DOS *ENDPOINTS*

Possivelmente a primeira métrica que surge em mente para avaliar o desempenho de um sistema LoRa é o alcance de comunicação entre *endpoints* e *gateways*, o qual está fortemente atrelado a algumas variáveis do simulador. Dentre elas é possível citar principalmente a potência de transmissão dos *endpoints* LoRa e o modelo de *Path Loss* utilizado no simulador. O modelo Okumura-Hata, mencionado na seção 5.1, foi escolhido devido ao fato de ele possibilitar a configuração de um cenário “rural”, que melhor representa a situação de um parque eólico.

Realizando a codificação deste modelo no simulador FLoRa obtemos os resultados representados na Figura 26. Nela, os *endpoints* configurados para transmitir com uma potência de 14 dBm conseguem alcançar o *gateway* dentro de um raio de 4,6 km; já se a potência de transmissão for elevada para 22 dBm (limite por especificação de *hardware*) a distância máxima de comunicação entre *endpoint* e *gateway* passar a ser de 7,8 km.

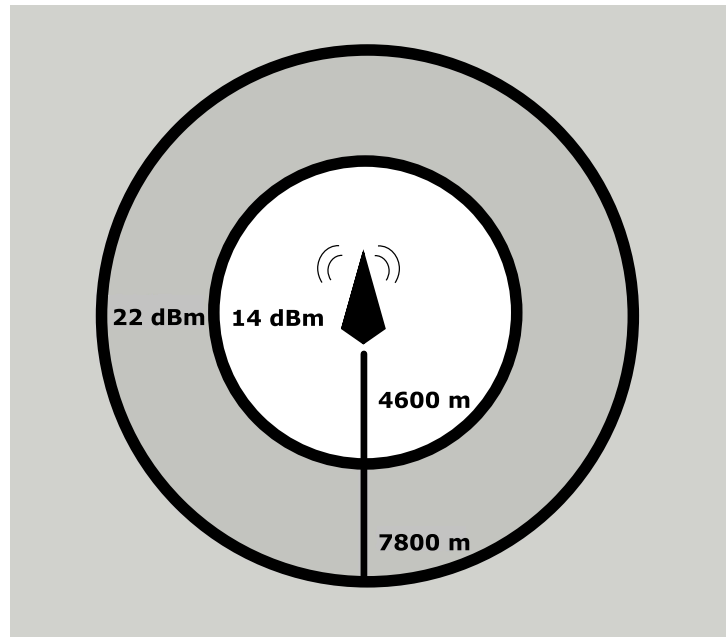


Figura 26 – Alcance dos *endpoints* LoRa para o modelo Okumura-Hata.

Esta informação configura o primeiro cenário de *trade-off*: embora o alcance de 7,8 km seja desejável ele traz junto consigo um aumento considerável no consumo de energia para os *endpoints*. Em princípio foi decidido manter a potência de transmissão dos *endpoints* em 14 dBm e, caso necessário e por meio de um algoritmo de ADR, modificar esta potência.

6.2 CENÁRIO INICIAL

Com o alcance dos *endpoints* verificado, foi realizada uma breve avaliação de como a rede LoRa se comporta com a variação dos seguintes parâmetros:

- Potência de transmissão dos *endpoints*;
- Número de *endpoints*;
- Número de *gateways*;
- Período entre transmissões dos *endpoints*.

Esta breve avaliação tem como intuito entender o comportamento de escalabilidade de uma rede LoRa e a partir disto construir um cenário inicial, a partir do qual seja possível verificar o efeito de modificações na rede na tentativa de se estabelecer uma configuração que atenda da melhor forma os requisitos de projeto, que são os seguintes:

- Período entre transmissões dos *endpoints* ajustável para permitir prioridade para *endpoints* em “área crítica”, onde um trabalho de maior risco é realizado e portanto o período entre transmissões deve ser reduzido;
- Atender 1500 *endpoints*;

- Manter o *Data Extraction Rate* DER da rede de 60%;

O primeiro parâmetro a ser alterado foi a potência de transmissão dos *endpoints*. O gráfico demonstrado na Figura 27 representa a perda total de pacotes e a perda somente por colisões enquanto a potência é alterada. Este gráfico está situado em um cenário onde temos somente um *gateway*, quinhentos *endpoints* estáticos com um período entre transmissões de 60 segundos e distribuídos em um círculo de raio de 2 km ao redor do *gateway*. O gráfico demonstra que a partir de um determinado valor, neste caso 14 dBm, não existe mais benefício em aumentar a potência de transmissão. A partir deste valor, quase não existem mais perdas de pacotes por *path loss* e todas as perdas são aproximadamente por colisões de pacotes. É importante notar que se o cenário de simulação fosse alterado de forma que os *endpoints* estivessem distribuídos em um raio de 5 km esta curva seria alterada para a perda total de pacotes se tornar principalmente perda por colisões aos 18 dBm.

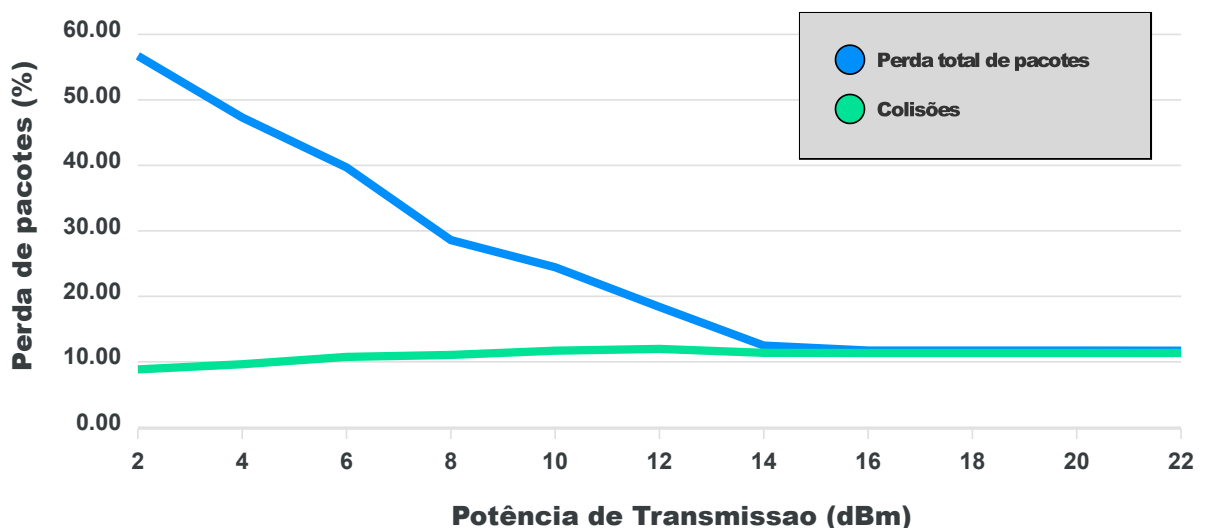


Figura 27 – Perdas de pacotes resultantes de acordo com a variação de potência de transmissão. Cenário: 1 *gateway*, 500 *endpoints* estáticos, 60 s de intervalo entre transmissões, 2 km de raio.

Outro comportamento de interesse é relacionado à variação no número de *endpoints*, representado na Figura 28. O cenário utilizado para colher estes dados é semelhante ao anterior, ou seja, um *gateway*, número variante de *endpoints* com um período entre transmissões de 60 s e distribuídos em um círculo de raio de 2 km ao redor do *gateway*. Entre os resultados é apresentado o gráfico da *Data Extraction Rate* (DER), que representa a porcentagem de mensagens recebidas pelo *gateway* com sucesso. É visível que existe um limite para a escalabilidade de uma rede LoRa com somente um *gateway* e que a perda por colisões vai aumentando gradativamente, ultrapassando 50% quando quatro mil *endpoints* estão se comunicando com o *gateway*.

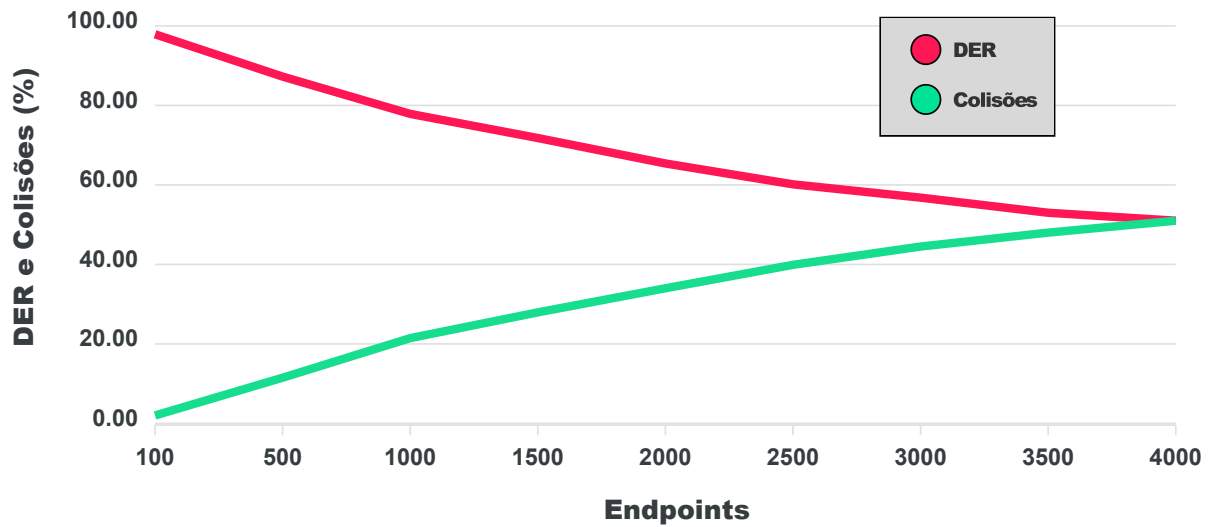


Figura 28 – Perda por colisões e DER resultantes em função do número de *endpoints*. Cenário: 1 *gateway*, 60 s de intervalo entre transmissões, 2 km de raio.

Para possibilitar a escalabilidade da rede LoRa para um número elevado de *endpoints* se torna necessário a adição de *gateways*, que foi o próximo parâmetro a ser alterado. Na Figura 29 está representado as diferentes configurações utilizadas para a distribuição dos *gateways*.

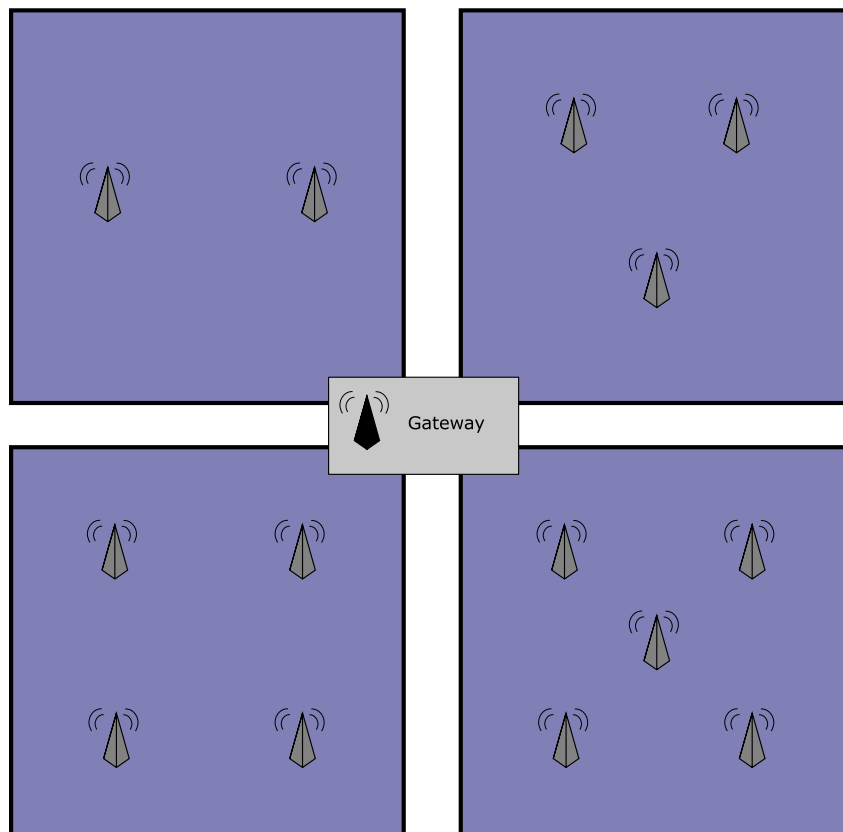


Figura 29 – Distribuições de *gateways* analisados no estudo de sensibilidade.

Distribuindo os *gateways* desta maneira os resultados representados na Figura 30 são obtidos. O cenário considerado para obter estes resultados é o seguinte: quinhentos *endpoints* com período entre transmissões de 60 s distribuídos em um círculo de raio de 5 km (para que o benefício de aumentar o número de *gateways* possa ser percebido). O número de *gateways* é alterado de um até cinco *gateways*. É possível observar o impacto positivo da adição de *gateways*, aumentando consideravelmente o DER. Caso um mesmo pacote seja recebido por mais de um *gateway*, ambos são enviados para *Network Server*, que irá considerar apenas o primeiro pacote recebido.

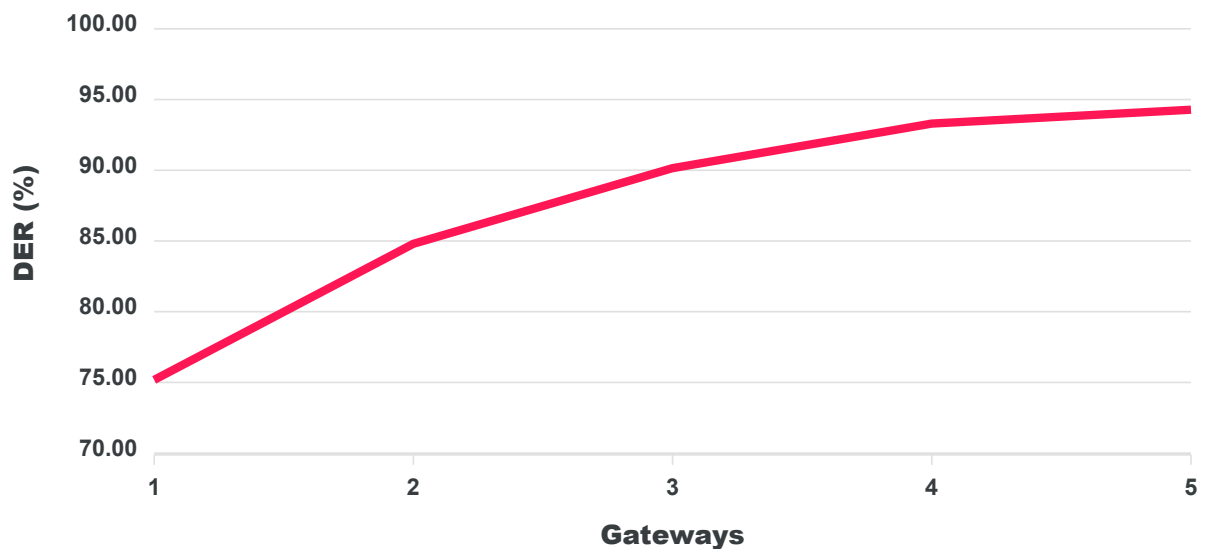


Figura 30 – DER resultante em função da variação do n^o de *gateways*. Cenário: 500 *endpoints*, 60 s de intervalo entre transmissões, 5 km de raio.

Por fim, antes de ser levantado um cenário inicial, é verificado o comportamento da rede LoRa com a alteração do período entre transmissões. Na Figura 31 são apresentados gráficos de DER para diferentes períodos entre transmissões dos *endpoints*, cuja quantidade é variada. É visível que a medida que o período entre transmissões cresce, o DER também aumenta. Entretanto, esta razão tem um comportamento de *diminishing returns*, tendo em vista que o aumento do período entre transmissões de 60 s para 120 s resulta em um aumento de DER consideravelmente maior de que quando o período passa de 180 s para 240 s. Isto ocorre pois a alteração de período de 60 s para 120 s reduz a utilização da rede em 50%, enquanto a de 180 s para 240 s reduz apenas em 25%.

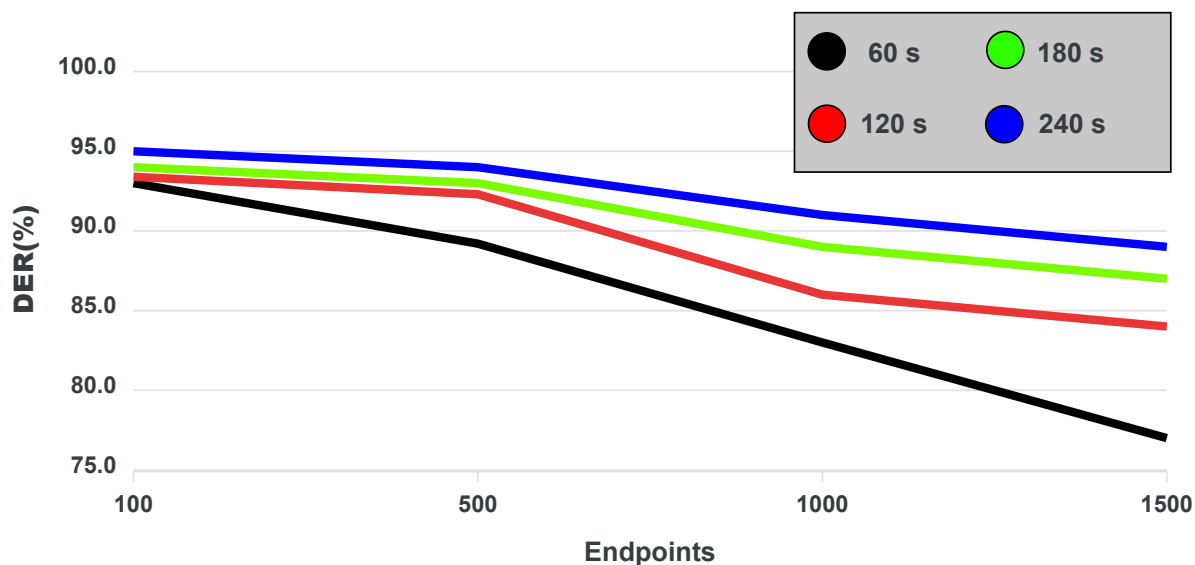


Figura 31 – DER resultante de acordo com a variação do período entre transmissões. Cenário: 5 *gateways*, 10 km de raio.

De maneira similar à escolha de potência de transmissão dos *endpoints*, a princípio foi decidido manter o período entre transmissões em 60 s e, se necessário (por exemplo, com base no critério de área crítica), alterá-lo em tempo de execução, conforme será tratado na seção seguinte.

6.3 ÁREAS CRÍTICAS

Durante a construção dos parques eólicos existem áreas onde um trabalho de maior risco está sendo realizado. Portanto, seria interessante uma análise de como o sistema LoRa se comporta caso seja necessário aumentar a frequência de comunicação *endpoint-gateway* para uma determinada quantidade de *endpoints* que estão inseridos em uma área “crítica”.

Para implementar esta alteração no código do simulador foi necessário a geração de variáveis auxiliares, que correspondem aos períodos entre transmissões que se deseja avaliar. Esta abordagem permite, por exemplo, atribuir a alguns um período entre transmissões de 60 s e a outros 100 *endpoints* um período de 120 s.

Para melhor entender como é impactado o desempenho de um *gateway* quando ele está recebendo pacotes com períodos entre transmissões distintos, um conjunto de simulações foi realizado. Nestas simulações, uma rede LoRa foi configurada com somente um *gateway* recebendo pacotes de 500 *endpoints*, na qual metade dos *endpoints* se encontram em uma área crítica e portanto transmitem pacotes com maior frequência. Ao longo das simulações o período entre transmissões dos *endpoints* em área crítica foi reduzido para avaliar como o *gateway* se comporta. Os resultados compilados deste conjunto de simulações está

representado na Figura 32, onde foi reduzido o período entre transmissões de metade dos *endpoints* em decrementos de 10 s.

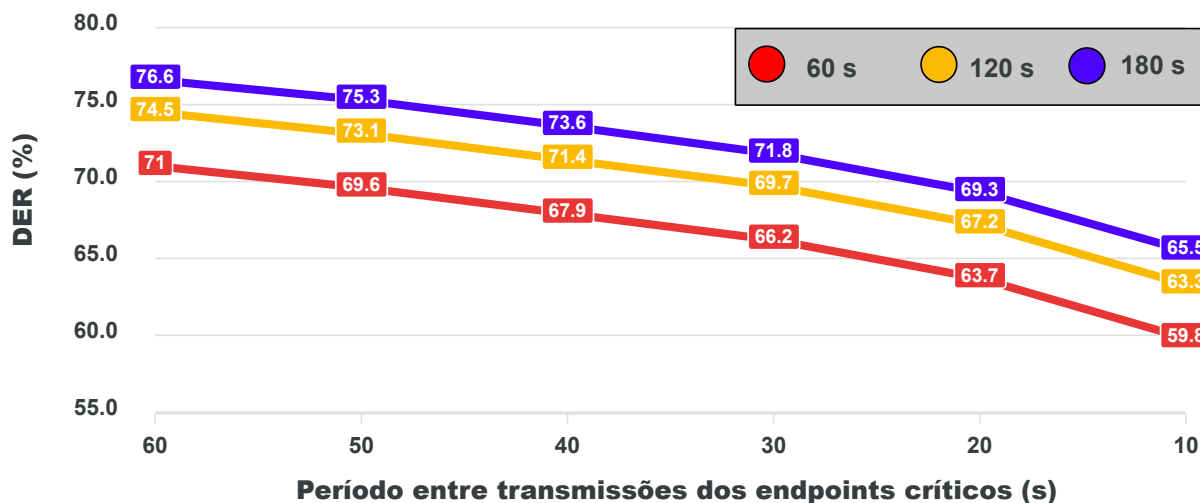


Figura 32 – DER de acordo com a alteração do período entre transmissões. Cenário: 1 *gateway*, 500 *endpoints* (250 críticos e 250 não críticos), 60 s de período para *endpoints* não críticos, 10-60 s de período para *endpoints* críticos, 3,34 km de raio.

Com a intenção de compensar pelo impacto negativo da diminuição no desempenho da rede é possível aumentar o período entre transmissões dos *endpoints* que não estão em área crítica. Entretanto é bom ressaltar que este aumento gera uma defasagem no monitoramento destes *endpoints* “não críticos”. A Figura 32 mostra os resultados desta avaliação, onde três cenários são representados em quais os *endpoints* não críticos foram configurados para transmitir pacotes a cada 60, 120 e 180 segundos. Nestes cenários, de forma semelhante ao que foi feito acima, o período entre transmissões dos *endpoints* críticos é reduzido em decrementos de 10 s para avaliar o DER resultante desta combinação de período entre transmissões de *endpoints* não críticos/críticos.

Como é possível observar na Figura 32, o aumento do período entre transmissões dos *endpoints* não críticos recupera o impacto negativo da redução do período entre transmissões dos *endpoints* críticos. Por exemplo, quando temos todos os *endpoints*, tanto os em área crítica quanto os fora dela, transmitindo a cada 60 s temos um DER de 71%. É possível obter um DER semelhante dando maior prioridade para os pacotes “críticos” utilizando uma combinação de períodos entre transmissões de forma que os pacotes críticos sejam transmitidos a cada 40 s e os pacotes não críticos a cada 120 s.

Com este conjunto de simulações realizado pode se perceber os impactos na recepção de um *gateway* quando o período entre transmissões dos pacotes são alterados. A partir disto é possível inserir esta abordagem de áreas críticas no cenário inicial, onde existem 4 *gateways* recebendo mensagens de 1500 *endpoints*.

6.3.1 DETERMINAÇÃO DAS ÁREAS CRÍTICAS

Após a avaliação de desempenho de um *gateway* em relação à alteração na periodicidade entre transmissões dos *endpoints*, uma alteração no código do simulador foi realizada para estabelecer as áreas críticas ao longo do parque apresentado no capítulo 1. Foram estabelecidas a princípio duas áreas críticas, nas quais os *endpoints* inseridos nelas são considerados como críticos e transmitem com uma periodicidade menor. Estas duas áreas estão representadas na Figura 33 por quadrados vermelhos 1 km de lado. Estas duas áreas foram dimensionadas e posicionadas livremente, sendo o principal motivo desta implementação a geração e posicionamento de áreas críticas para a avaliação de desempenho da rede LoRa.

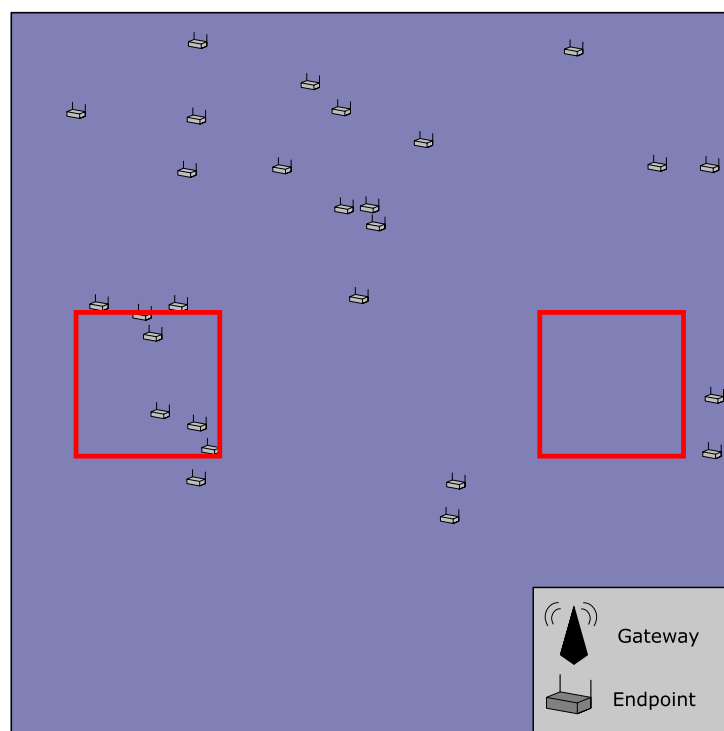


Figura 33 – Determinação das áreas críticas.

6.3.2 IMPACTO NO CENÁRIO INICIAL

Para avaliar o impacto da existência de *endpoints* “críticos”, que transmitem com uma maior frequência por estarem dentro de áreas críticas do parque eólico, foi realizado um conjunto de simulações com 4 *gateways* recebendo pacotes LoRa de 1500 *endpoints*. Destes 1500 *endpoints*, foram considerados como “críticos” 300 deles, ou seja, 20% do total de *endpoints*, o que pode ser considerado como um *worst-case-scenario*.

Estes 300 *endpoints* críticos foram distribuídos igualmente dentro das duas áreas críticas definidas na Figura 6.3.1. Na Figura 32 são apresentados os resultados das simulações realizadas para avaliar os efeitos da alteração na periodicidade de transmissão dos *endpoints*. Estas simulações foram realizadas de forma semelhante ao apresentado na Figura 32, ou

seja, foi analisado o cenário em que o período entre transmissões dos *endpoints* não críticos é mantido em 60 s e é realizada a redução progressiva do período entre transmissões dos *endpoints* críticos. Além deste cenário foram analisados outros dois de forma semelhante ao explicado previamente porém realizando a alteração da periodicidade de transmissão dos *endpoints* não críticos para 120 s e 180 s.

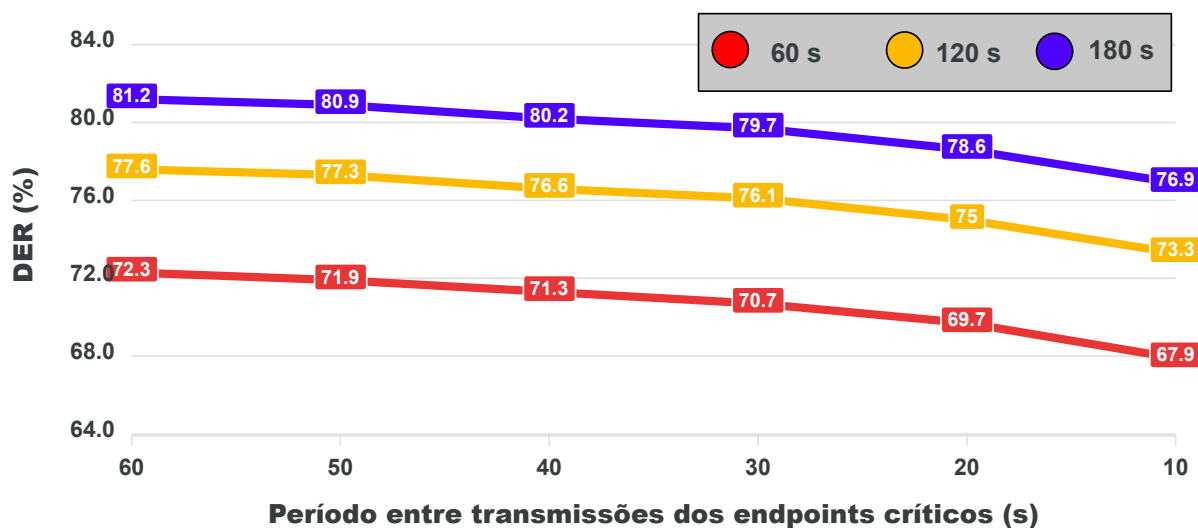


Figura 34 – DER de acordo com a combinação do período entre transmissões dos *endpoints*. Cenário: 4 *gateways*, 1500 *endpoints* (1200 não críticos e 300 críticos), 60-180 s de período para *endpoints* não críticos, 10-60 s de período para *endpoints* críticos, 3,34 km de raio.

Distintamente do que é apresentado ao longo da seção 6.3, aqui as simulações são realizadas considerando o conjunto de quatro *gateways* e a quantidade relativa de *endpoints* críticos é reduzida de 50% para 20%. Neste caso é perceptível que o impacto negativo no DER com a diminuição da periodicidade entre transmissões dos *endpoints* críticos é consideravelmente menor do que quando a simulação foi realizada com só um *gateway*, na seção 6.3. Por exemplo, no cenário em que consideramos a periodicidade dos *endpoints* não críticos de 60 s temos uma redução no DER de 2,6 pp. quando a periodicidade dos *endpoints* críticos é reduzida de 60 s para 20 s. Porém, considerando a utilização de somente um *gateway* e a relação de *endpoints* críticos de 50%, como foi ilustrado na seção 6.3, a redução no DER da rede quando a periodicidade dos *endpoints* críticos é reduzida de 60 s para 20 s é maior, em 7,3 pp..

Com estes resultados é possível concluir que a implementação de áreas críticas com a diminuição da periodicidade dos *endpoints* críticos é aceitável (graças ao maior número de *gateways*) e, embora traga impactos negativos no DER da rede LoRa, tal redução é tolerável tendo em vista o benefício da diminuição da periodicidade entre transmissões dos *endpoints* críticos, que necessitam ser monitorados com uma frequência maior.

6.4 ALGORITMOS ADR

Os resultados de simulação que comparam o desempenho dos algoritmos ADR serão divididos em dois cenários. Primeiramente, um cenário genérico com somente um *gateway* é investigado. As conclusões levantadas deste cenário são então verificadas em um cenário específico de parque eólico, com 4 *gateways* e 1500 *endpoints*. Cabe destacar que o G-ADR é executado somente no NS, segundo Farhad et al. (2020). No entanto, esse algoritmo será comparado com o ADR-TTN e ADR+, os quais podem ser computados tanto no NS como no *endpoint*.

6.4.1 ÚNICO *gateway*

Consideremos uma rede LoRaWAN caracterizada por somente um *gateway* cercado por 300 *endpoints* em uma área de 35 km² e com os *endpoints* transmitindo pacotes a cada 60 segundos com um TP inicial de 14 dBm, como ilustrado na Figura 35. Entre as implementações de algoritmos ADR está uma abordagem onde nenhum algoritmo é implementado. Neste caso, o SF é alocado aleatoriamente para os *endpoints* no início da simulação. O desempenho desta estratégia e dos algoritmos ADR previamente mencionados é comparado com base no critério de DER, que deve ser maximizado.

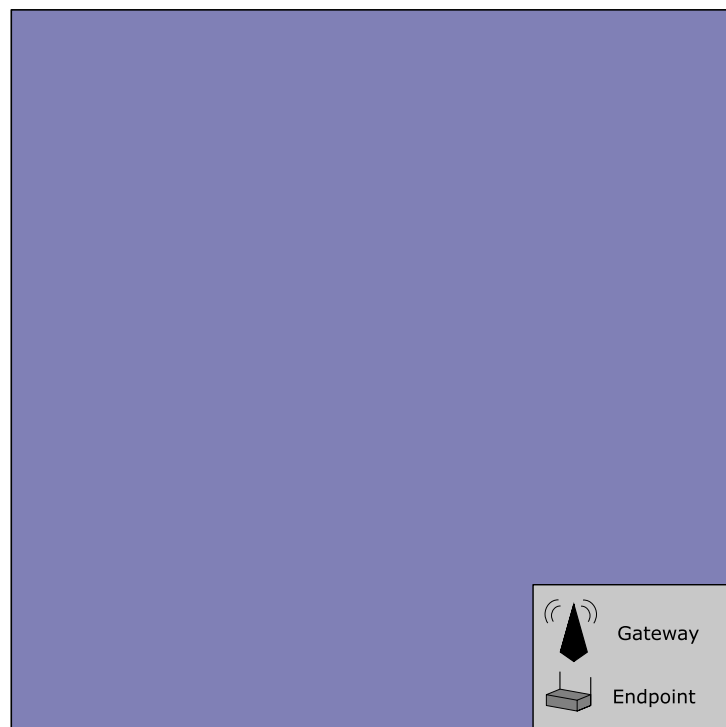


Figura 35 – Cenário de um *gateway* e 300 *endpoints* dentro uma área de 35 km².

Na Figura 36 é apresentado como o DER evolui ao longo do tempo para os diferentes algoritmos ADR. Além do fato de o algoritmo G-ADR ter um desempenho melhor, é interessante observar que tanto o ADR-TTN quanto o ADR+ deveriam apresentar o

mesmo desempenho. No entanto, isso não se observa devido à aleatoriedade nas posições dos *endpoints* no começo de cada simulação, o que causou uma leve disparidade nos valores de DER até se atingir o equilíbrio.. Além disso, ambos ADR-TTN quanto ADR+ têm um desempenho pior do que quando nenhum algoritmo é implementado. Isto ocorre principalmente devido ao método de alocação “agressivo” de SF e TP destes algoritmos, com resultados que podem ser visualizados na Figura 37, alocando muitos *endpoints* ao mesmo SF, e dessa forma permitindo um número excessivo de colisões.

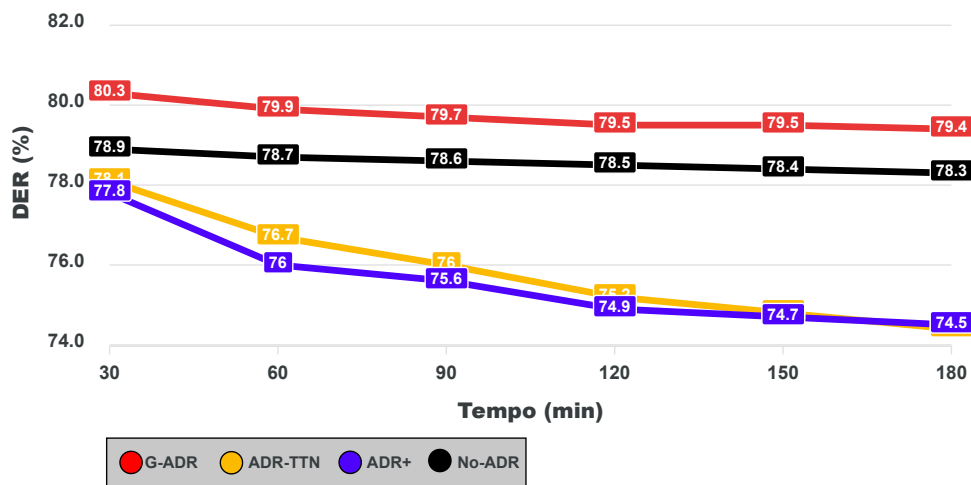


Figura 36 – DER resultante no cenário de um único *gateway* com 300 *endpoints* estáticos.

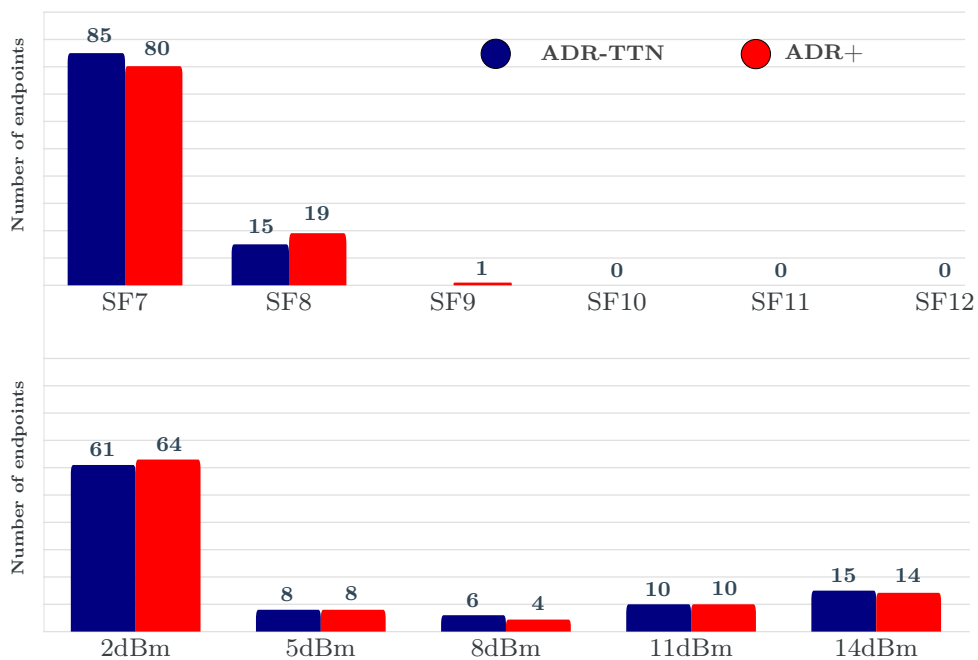


Figura 37 – Alocação de SF e TP nos algoritmos ADR-TTN e ADR+ no cenário de um *gateway* com 300 *endpoints* estáticos.

Quando a mobilidade dos *endpoints* é considerada, os resultados de DER são os presentes

na Figura 38. Pode ser visto que continua ser desvantajosa a implementação do algoritmo ADR-TTN quando comparado à abordagem onde nenhum algoritmo é implementado. Isto acontece pois o algoritmo ADR-TTN foi desenvolvido para *endpoints* estáticos e procura minimizar o custo energético utilizando valores baixos de TP, enquanto a estratégia “No-ADR” consiste em distribuição aleatória de SF e TP sempre a 14 dBm, aumentando a média de RSSI nos *gateways*. O algoritmo G-ADR neste cenário se apresenta como a segunda melhor opção, entretanto, assim como os outros algoritmos, tem desempenho pior do que quando os *endpoints* eram estáticos. Nota-se que o algoritmo D-ADR, proposto nesta dissertação, alcança o melhor desempenho, com um DER de 72,4% ao final da simulação.

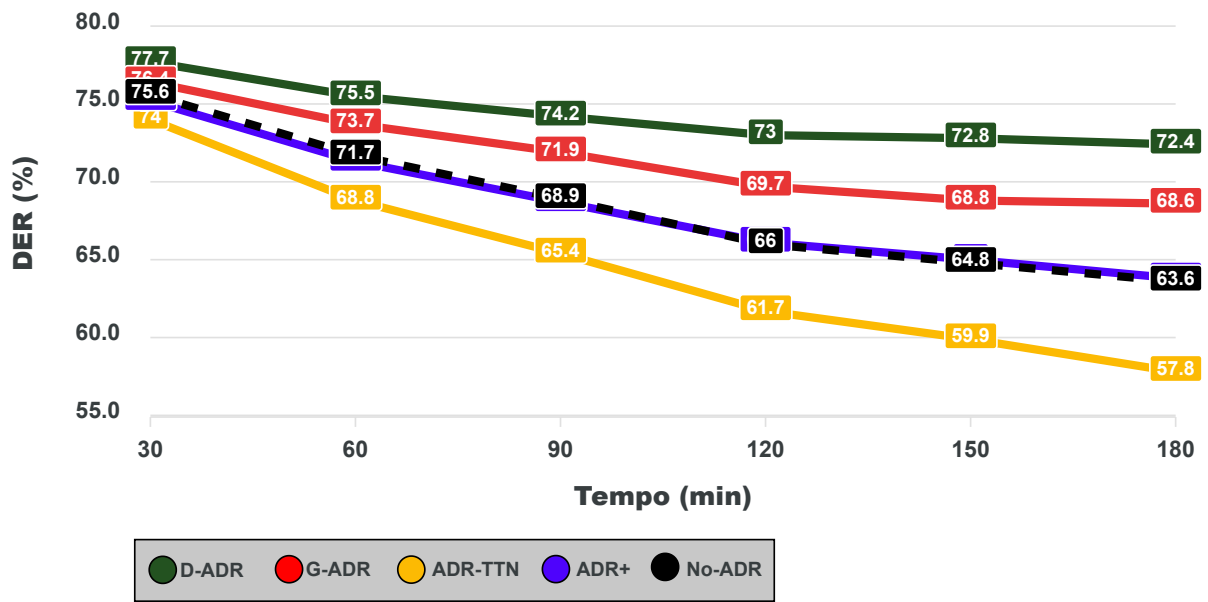


Figura 38 – DER resultante no cenário de um único *gateway* com 300 *endpoints* móveis.

6.4.2 CENÁRIO DE PARQUE EÓLICO

Com o melhor desempenho do algoritmo D-ADR no cenário de único *gateway* apresentado, é de interesse verificar se este resultado se mantém no cenário do parque eólico ENEL. Portanto, o simulador FLoRa foi reconfigurado para 4 *gateways*, posicionados de forma a respeitar a estrutura do parque apresentada na Figura 4, comunicando com 1500 *endpoints* dentro de uma área de 35 km², configurando o cenário ilustrado na Figura 39. O TP inicial é mantido como 14 dBm e a periodicidade de envio de *uplinks* é de 60 s, com um SF inicial de 7.

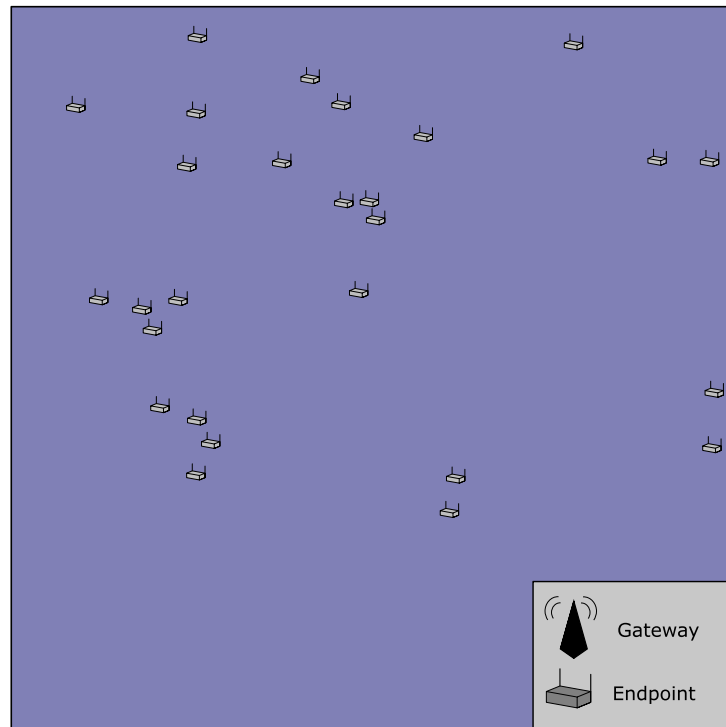


Figura 39 – Cenário de parque eólico com 4 gateways e 1500 endpoints móveis em uma área de 35 km².

Na Figura 40 pode ser visualizado o DER resultante do uso de algoritmos ADR no cenário do parque eólico. Nota-se que o algoritmo D-ADR se mantém como o mais eficiente, alcançando um DER de 61% no final da simulação. Estes resultados são piores que os correspondentes quando no cenário de único gateway (Figura 38) devido à maior densidade de endpoints: enquanto o número de gateways aumentou 4 vezes, o número de endpoints aumentou 5 vezes.

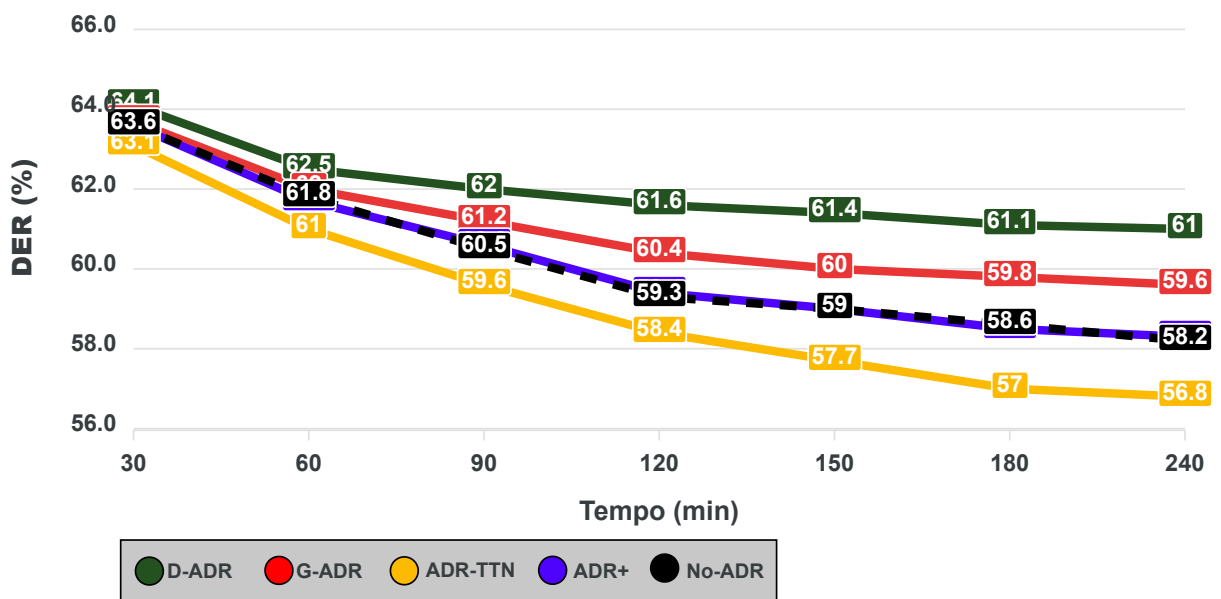


Figura 40 – DER resultante em cenário de parque eólico com 4 gateways e 1500 endpoints.

GASTO ENERGÉTICO

Além do DER, é importante comparar os algoritmos de ADR com base no consumo energético, uma vez que um desempenho melhor em DER pode ser obtido graças a um consumo energético muito maior, tal que um equilíbrio entre os dois critérios deva ser buscado. Na Figura 41 o número de *endpoints* é variado e os algoritmos ADR são comparados através do gasto energético total dos *endpoints*.

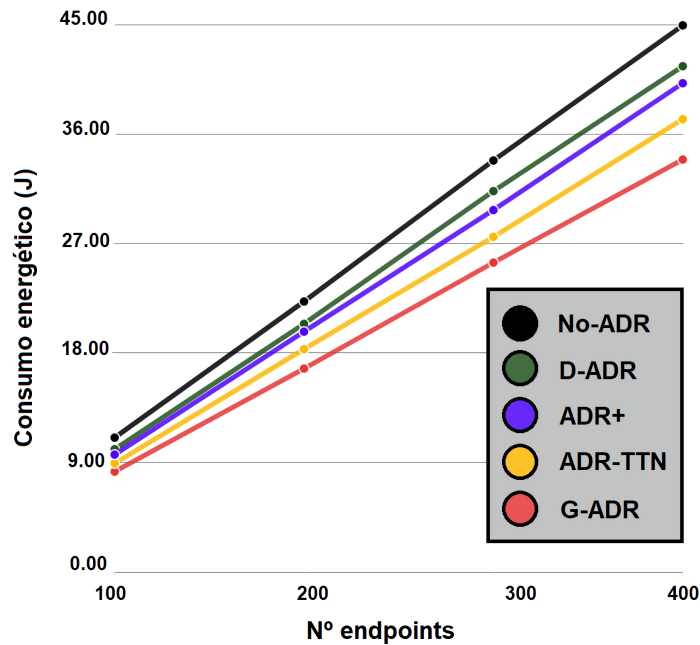


Figura 41 – Gasto energético total entre distintos algoritmos ADR.

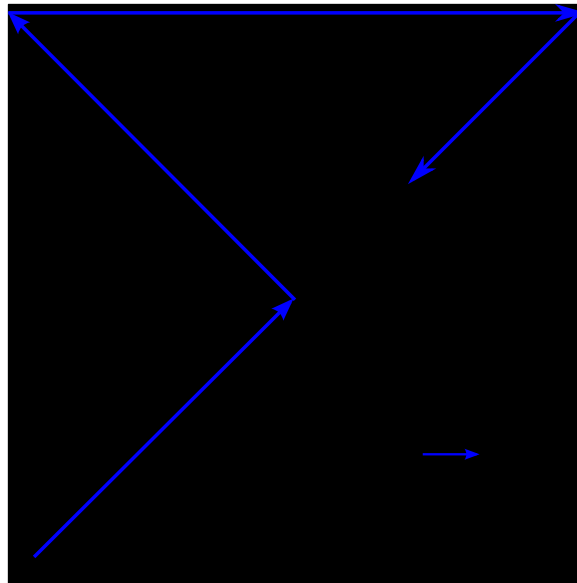
O algoritmo D-ADR proposto tem o gasto energético similar, porém pior, ao do algoritmo ADR+, enquanto os algoritmos ADR-TTN e G-ADR tem gastos energéticos consideravelmente menores, sendo o G-ADR o algoritmo com melhor desempenho, neste caso menor gasto energético. A situação onde nenhum algoritmo ADR é implementado e todos *endpoints* transmitem com TP de 14 dBm tem o pior desempenho, que já era esperado. Com estes dados é possível levantar o gasto energético para a transmissão de um pacote LoRa em função do algoritmo ADR utilizado, conforme ilustrado na Tabela 4. Nesta tabela está informado também a melhora em % em referência ao pior caso, quando nenhum algoritmo ADR é implementado denominado de “No-ADR”.

DINÂMICA DO ALGORITMO D-ADR

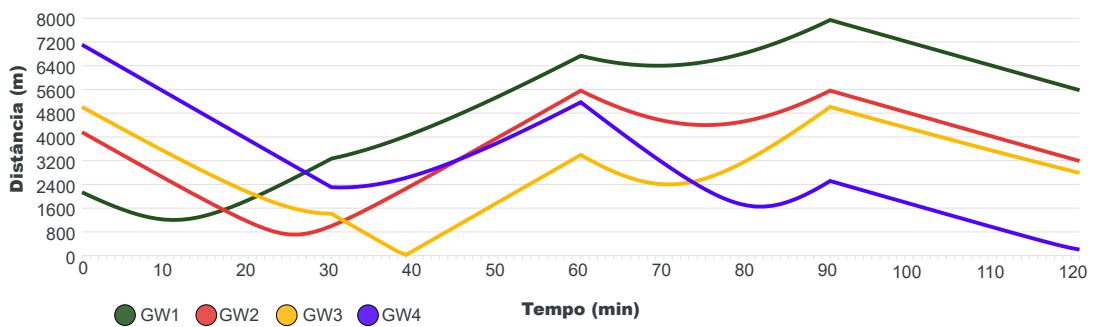
Para visualizar a dinâmica do algoritmo proposto nesta dissertação, o seguinte experimento pode ser realizado: estipula-se um padrão de mobilidade para um *endpoint* singular e se registra os parâmetros de ADR utilizados ao longo desta trajetória. Na Figura 42 o comportamento dinâmico do algoritmo D-ADR para um *endpoint* (com um padrão de mobilidade *Ad Hoc*) é apresentado.

Algoritmo ADR	Gasto energético por pacote (mJ)	Melhora em % ref. pior caso
No-ADR	31,2	0%
D-ADR	28,9	7,9%
ADR+	27,9	11,8%
ADR-TTN	25,8	20,9%
G-ADR	23,6	32,2%

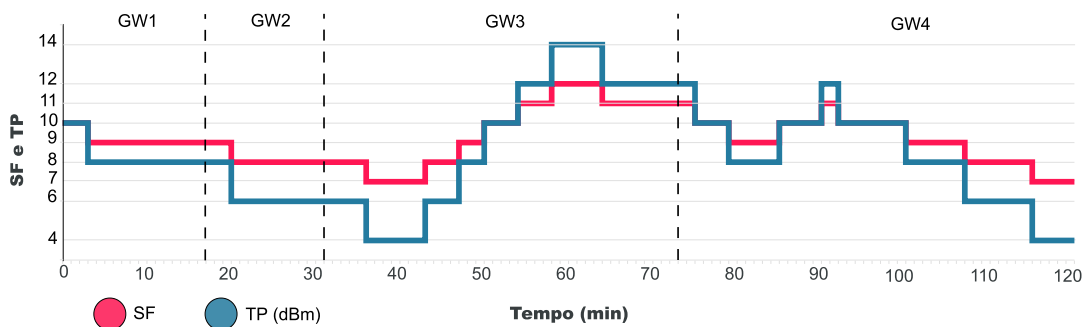
Tabela 4 – Gasto energético em função de algoritmo ADR.



(a) Trajetória *Ad Hoc* do *endpoint*



(b) Distância dos *gateways*



(c) Alocação de SF e TP com indicação do *gateway* mais próximo

Figura 42 – Dinâmica do algoritmo D-ADR.

7 TRABALHO EXPERIMENTAL

Para validar o funcionamento do algoritmo D-ADR uma implementação em *hardware* foi desenvolvida. Neste capítulo são detalhadas as características principais deste *hardware*, bem como os procedimentos experimentais adotados. Testes de campo foram realizados com um *endpoint* móvel para demonstrar o funcionamento esperado do algoritmo.

7.1 HARDWARE E PROCEDIMENTOS

Como mencionado no capítulo 2, *endpoints* são dispositivos compostos por, pelo menos, um módulo de rádio e uma unidade de processamento. Estes dispositivos podem acomodar outros componentes, como no caso deste trabalho, um módulo GNSS, uma bateria, IMU e uma *Human-machine interface* (HMI). Entretanto, como o enfoque desta dissertação é o desenvolvimento de um algoritmo ADR utilizando a tecnologia LoRa e como o *hardware* consiste em um protótipo desenvolvido por uma empresa privada, estes componentes adicionais (GNSS, bateria, IMU e HMI) não serão abordados. Para realizar as transmissões e recepções por LoRa, o *endpoint* utiliza um *chip* “SX1262” da Semtech e o processamento destes pacotes é realizado pelo *System on a Chip* (SoC), dispostos em uma placa eletrônica como mostra a Figura 43.

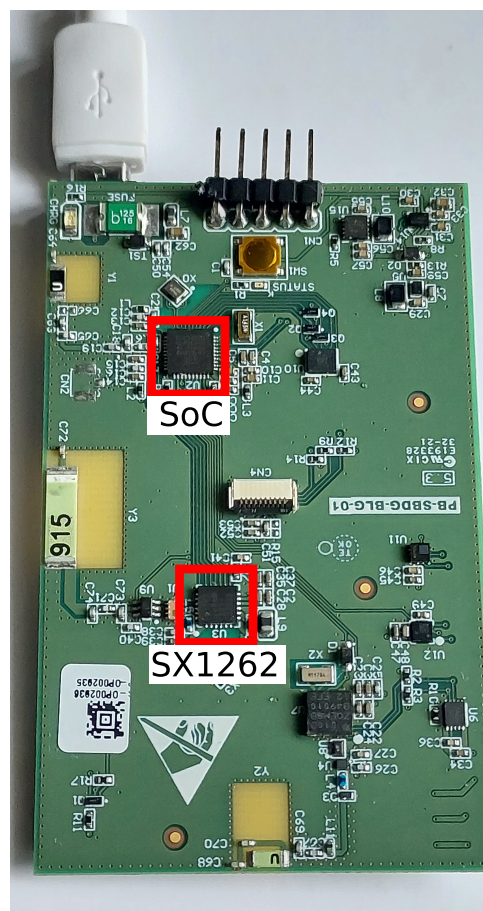


Figura 43 – *Hardware* utilizado.

CÓDIGO IMPLEMENTADO

Em conjunto ao *hardware*, uma versão em código do algoritmo D-ADR apresentado no capítulo 4 foi implementada e está demonstrada no “pseudo-código” a seguir:

```

Início: Requisição de uplink;
foreach Gateway i do
    | Dif_lati = Lat - Lat_Gwi;
    | Dif_loni = Lon - Lon_Gwi;
    | Dist_Gwi calculado utilizando Teorema de Pitágoras
end
if Menor distância < Lim1 then
    | SF = 7;
    | TP = 4;
    ...
else if Menor distância < Lim6 then
    | SF = 12;
    | TP = 14;
else
    | Mantém parâmetros de SF e TP;
end

```

Algoritmo 1: D-ADR.

O algoritmo 1 começa o seu funcionamento após uma requisição de novo *uplink*, quando as distâncias referentes aos *gateways* existentes são calculadas rapidamente utilizando o teorema de Pitágoras. Com base nesta distância, os parâmetros SF e TP são escolhidos dentro de 6 intervalos conforme mostra a Figura 22. Por fim, caso a distância calculada seja maior do que o limite do último intervalo, seja por uma má leitura dos dados provindos do GNSS ou caso o *endpoint* esteja fora dos limites do parque, os parâmetros de SF e TP são mantidos. Desta forma, caso uma má leitura provinda do GNSS ocorra resultando em um cálculo de distância errôneo (fora dos limites do parque), os parâmetros da última iteração são mantidos.

MQTT

Os dados são enviados dos *endpoints* para *gateways* e, após, para um *network server*. É comum que este NS publique os dados recebidos para um *broker* MQTT, para a disponibilização destes dados. A Figura 44 demonstra o sistema montado para o registro dos pacotes LoRa, onde o NS, após receber um *uplink* provindo de um *endpoint*, realiza a publicação dos pacotes para um *broker* MQTT. A coleta dos dados é então realizada através de um agente externo (no caso um computador) que realiza a “inscrição” nestes pacotes, conforme o padrão de troca de mensagens *Publisher/Subscriber*.

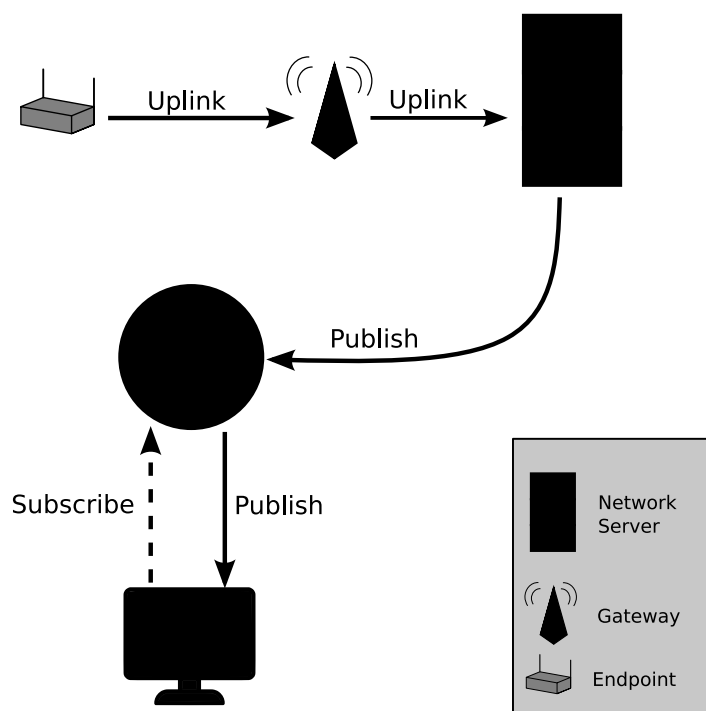


Figura 44 – Sistema MQTT para coleta de dados.

TESTES DE CAMPO

Com as preparações feitas, um teste de campo foi realizado para validar o comportamento do algoritmo D-ADR. Este teste foi realizado levando em consideração apenas um *endpoint* móvel e um *gateway*. A Figura 45 mostra a trajetória percorrida durante este teste, de aproximadamente 6 km ao longo de 66 minutos. Na Figura também estão sinalizados o início e fim da trajetória e a sede da empresa, que também é considerada como a posição do *gateway*.

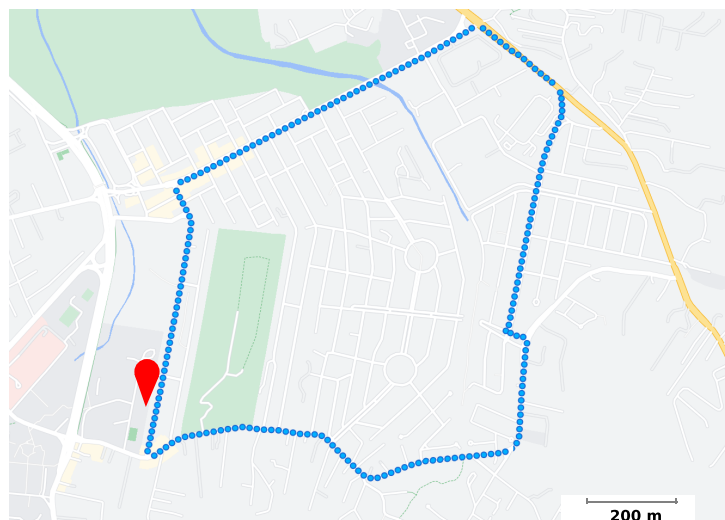
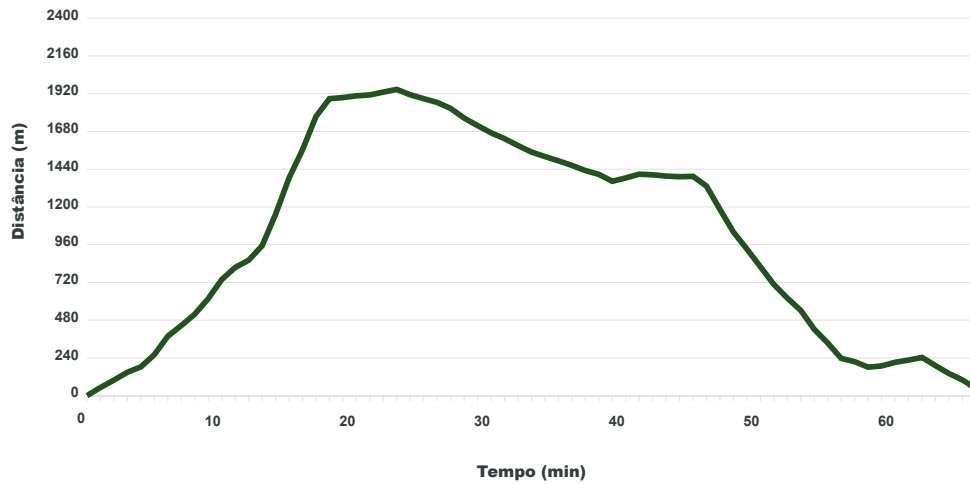


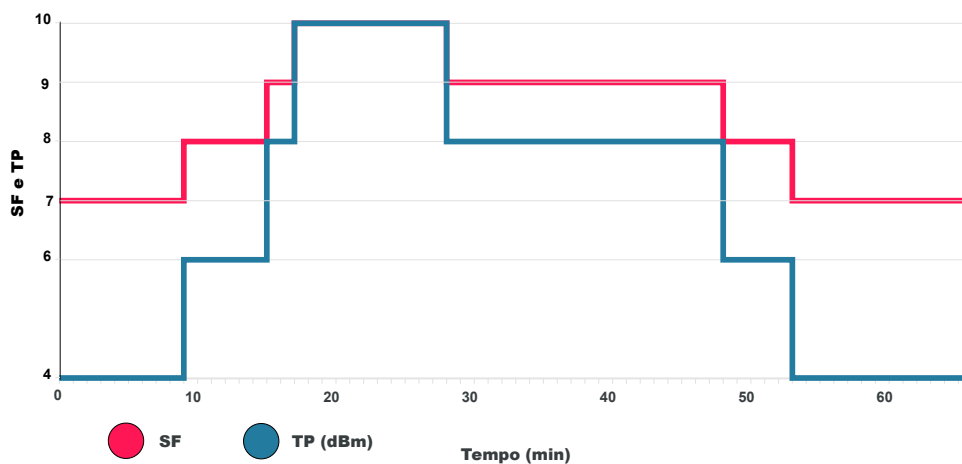
Figura 45 – Trajetória percorrida.

Após a realização dos testes, os dados foram coletados e estão presentes na Figura

46, com a curva de distância entre *endpoint* e *gateway* e os parâmetros de SF e TP escolhidos para transmissão com base nesta distância. A dinâmica do algoritmo é similar à apresentada na Figura 42 para o cenário do parque eólico: a medida que o *endpoint* se afasta do *gateway*, os parâmetros de SF e TP são ajustados conforme as funções estabelecidas na Figura 22.



(a) Distância do *gateway*



(b) Alocação de SF e TP

Figura 46 – Distância e parâmetros de SF e TP escolhidos.

8 CONCLUSÕES

A tarefa desafiadora de gerenciar a logística e garantir a segurança dos trabalhadores em áreas vastas, remotas e potencialmente perigosas, como parques eólicos, pode ser abordada adequadamente com a adoção de tecnologias de rastreamento, como os dispositivos vestíveis LoRa desenvolvidos e discutidos nesta dissertação. Por meio de uma análise no ambiente de simulação FLoRa, o emprego da tecnologia LoRa/LoRaWAN se provou adequada para utilização em ambientes com número elevado de *endpoints*, mesmo quando o período entre transmissões é reduzido em situação de “área crítica”. Entretanto, a natureza móvel dos *endpoints*, como esperado, prejudica o desempenho da rede.

As maiores dificuldades encontradas ao longo do desenvolvimento desta dissertação de mestrado foram a ambientação ao simulador FLoRa e a programação de algoritmos ADR no mesmo. Além disso, a implementação do algoritmo D-ADR na estrutura de software descrita no capítulo 7 para a realização do trabalho experimental também gerou dificuldades, sendo o principal fator a familiarização com uma nova estrutura de *software* e a implementação de novas funções para estruturar o algoritmo D-ADR.

O algoritmo de taxa de dados adaptável D-ADR, proposto e validado com simulações nesta dissertação, provou ser a melhor opção em comparação aos algoritmos ADR disponíveis para otimizar o fator de espalhamento e a potência de transmissão dos terminais LoRa, a fim de maximizar o DER. O D-ADR contribuiu significativamente, dessa forma, para a melhoria da logística de rastreamento de campo e segurança do trabalho, por meio de dispositivos LoRa.

No entanto, melhorias podem ser feitas no algoritmo, como a utilização de critérios para melhorar a alocação de recursos. Por exemplo, na versão atual do algoritmo D-ADR, se todos os terminais estiverem próximos uns dos outros, eles tendem a operar com os mesmos parâmetros, levando a colisões e, conseqüentemente, à perda de pacotes. Isso pode ser evitado com a alocação justa já mencionada, que será discutida em trabalhos futuros.

TRABALHOS FUTUROS

Como mencionado, melhorias e algumas análises ainda podem ser feitas, de forma a aperfeiçoar o algoritmo D-ADR, entre as quais podem ser citadas:

- Implementação em *software* do algoritmo D-ADR no NS, possibilitando a implementação de critérios de justiça na alocação de recursos;
- Realização de testes de campo com número elevado de *endpoints*;
- Recalibragem das funções de SF e TP, vistas na Figura 22, para ambientes distintos;
- Verificação da utilização do *duty-cycle* em cenários com zonas críticas, onde temos períodos entre transmissões menores do que 60 s, que podem exceder a utilização regulamentar;

-
- Análise da utilização de replicações para *endpoints* situados em áreas críticas, para que os dados transmitidos pelos mesmos tenham uma maior chance de alcançarem e serem recebidos pelos *gateways*;
 - Análise da utilização de trilateração dos dados recebidos por múltiplos *gateways* para estimar a posição dos *endpoints*, assim sendo possível a remoção do módulo de GNSS que é o maior consumidor de energia da eletrônica embarcada.

BIBLIOGRAFIA

ANATEL. **Ato nº 14448**. 2017. Disponível em:

<<https://www.anatel.gov.br/legislacao/atos-de-certificacao-de-produtos/2017/1139-ato-14448>>. Acesso em: 4 jan. 2021.

ASIAN, S.; ERTEK, G.; HAKSOZ, C.; PAKTER, S.; ULUN, S. Wind Turbine Accidents: A Data Mining Study. **IEEE Systems Journal**, v. 11, n. 3, p. 1567–1578, 2017.

BENKAHLA, N.; TOUNSI, H.; FRIKHA, M.; SONG, Y.-Q. VHMM-based E-ADR for LoRaWAN networks with unknown mobility patterns. In: IWCMC 2021 - 17TH INT. WIRELESS COMMUNICATIONS & MOBILE COMPUTING CONFERENCE. Herbin, China, jun. 2021. Disponível em:

<<https://hal.archives-ouvertes.fr/hal-03283215>>.

BOHN, E. **LoRa Documentation**. 2020. Disponível em:

<<https://lora.readthedocs.io/en/latest/>>. Acesso em: 4 jan. 2021.

BOR, M. C.; ROEDIG, U.; VOIGT, T.; ALONSO, J. M. Do LoRa Low-Power Wide-Area Networks Scale? **Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems**, nov. 2016. Acesso em: 8 nov. 2021.

COTRIM, J. R.; KLEINSCHMIDT, J. H. LoRaWAN Mesh Networks: A Review and Classification of Multihop Communication. **Sensors**, v. 20, p. 4273, jul. 2020. Acesso em: 7 jun. 2021.

CUOMO, F.; CAMPO, M.; CAPONI, A.; BIANCHI, G.; ROSSINI, G.; PISANI, P. EXPLoRa: Extending the performance of LoRa by suitable spreading factor allocations. In: 2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob). 2017. P. 1–8.

DAMOSSO, E.; CORREIA, L. M. et al. COST action 231: Digital mobile radio towards future generation systems: Final report. **European commission**, 1999.

EPE. **Balanco Energético Nacional 2021**. Empresa de Pesquisa Energética (EPE), 2021. Disponível em: <<https://www.epe.gov.br/pt/publicacoes-dados-abertos/publicacoes/balanco-energetico-nacional-2021>>. Acesso em: 20 dez. 2021.

- EPE. **Plano Decenal de Expansão de Energia 2024**. Empresa de Pesquisa Energética (EPE), 2019. Disponível em: <<https://www.epe.gov.br/pt/publicacoes-dados-abertos/publicacoes/Plano-Decenal-de-Expansao-de-Energia-2024>>. Acesso em: 20 dez. 2021.
- ERTÜRK, M. A.; AYDIN, M. A.; BÜYÜKAKKAŞLAR, M. T.; EVIRGEN, H. A Survey on LoRaWAN Architecture, Protocol and Technologies. **Future Internet**, v. 11, p. 216, out. 2019.
- FARHAD, A.; KIM, D.-H.; SUBEDI, S.; PYUN, J.-Y. Enhanced LoRaWAN Adaptive Data Rate for Mobile Internet of Things Devices. **Sensors**, v. 20, p. 6466, nov. 2020. Acesso em: 3 jul. 2021.
- FONSECA, D. F.; LAMARE, R. C. de; MADRUGA, E. L. Performance Evaluation of LoRaWAN and RS-LoRa in Mobile Scenarios.
- FRENZEL, L. E. **Principles of electronic communication systems**. Mcgraw-Hill Education, 2016.
- GHOSLYA, S. **All About LoRa and LoRaWAN: LoRa: Symbol Generation**. 2017. Disponível em: <<http://www.sghoslya.com/p/lora-is-chirp-spread-spectrum.html>>. Acesso em: 7 abr. 2021.
- GWEC. **Global Wind Energy Council**. Disponível em: <<https://gwec.net/>>. Acesso em: 10 jun. 2015.
- HATA, M. Empirical formula for propagation loss in land mobile radio services. **IEEE Transactions on Vehicular Technology**, v. 29, p. 317–325, ago. 1980. Disponível em: <<https://ieeexplore.ieee.org/document/1622772>>. Acesso em: 15 jun. 2021.
- HAXHIBEQIRI, J.; DE POORTER, E.; MOERMAN, I.; HOEBEKE, J. A Survey of LoRaWAN for IoT: From Technology to Application. **Sensors**, v. 18, n. 11, 2018. Disponível em: <<https://www.mdpi.com/1424-8220/18/11/3995>>.
- JESUS, G. G. M. de; SOUZA, R. D.; MONTEZ, C.; HOELLER, A. LoRaWAN Adaptive Data Rate With Flexible Link Margin. **IEEE Internet of Things Journal**, v. 8, n. 7, p. 6053–6061, 2021.

KONSTANTINIDIS, E.; BOTSARIS, P. Wind turbines: current status, obstacles, trends and technologies. **IOP Conference Series: Materials Science and Engineering**, v. 161, p. 012079, nov. 2016.

KUFAKUNESU, R.; HANCKE, G. P.; ABU-MAHFOUZ, A. M. A Survey on Adaptive Data Rate Optimization in LoRaWAN: Recent Solutions and Major Challenges. **Sensors**, v. 20, n. 18, 2020.

LI, S.; RAZA, U.; KHAN, A. How Agile is the Adaptive Data Rate Mechanism of LoRaWAN? **2018 IEEE Global Communications Conference (GLOBECOM)**, dez. 2018. Disponível em: <<https://ieeexplore.ieee.org/document/8647469>>. Acesso em: 18 out. 2021.

LORA ALLIANCE. **LoRaWAN® Specification v1.1**. 2020.

LORA ALLIANCE. **RP002-1.0.0 LoRaWAN Regional Parameters**. 2019. Disponível em: <https://lora-alliance.org/wp-content/uploads/2019/11/rp_2-1.0.0_final_release.pdf>. Acesso em: 4 jan. 2021.

MAGRIN, D.; CENTENARO, M.; VANGELISTA, L. Performance evaluation of LoRa networks in a smart city scenario. In: 2017 IEEE International Conference on Communications (ICC). 2017. P. 1–7.

MARTINEZ-CARO, J.-M.; CANO, M.-D. A novel holistic approach for performance evaluation in Internet of Things. **International Journal of Communication Systems**, v. 34, e4454, jan. 2021.

MICROCHIP TECHNOLOGY. **LoRaWAN™ Architecture**. 2021. Disponível em: <<https://microchipdeveloper.com/lora:lorawan-architecture>>. Acesso em: 4 jan. 2021.

OKUMURA, Y. Field strength and its variability in VHF and UHF land-mobile radio service. **Rev. Electr. Commun. Lab.**, v. 16, p. 825–873, 1968.

REYNDERS, B.; POLLIN, S. Chirp spread spectrum as a modulation technique for long range communication. In: p. 1–5.

SEMTECH. **LoRa and LoRaWAN: Technical overview | DEVELOPER PORTAL**. 2020. Disponível em: <<https://lora-developers.semtech.com/library/tech-papers-and-guides/lora-and-lorawan/>>.

SEMTECH. **LORAWAN - simple rate adaptation recommended algorithm**. Semtech, 2016.

SEMTECH. **Semtech Application Note AN1200.13. 2013. LoRa Modem Designer's Guide**. 2013. Disponível em: <<https://semtech.my.salesforce.com/sfc/p/#E0000000JelG/a/2R00000010K4/K1xBJSCPf1EbqU03CfABAjL29tRKA9KsdAdTIsWBA8s>>. Acesso em: 4 jan. 2021.

SEMTECH. **Semtech: AN1200.22 - LoRa Modulation Basics**. 2015.

SEMTECH. **SX1272 Datasheet**. 2019.

SENDRA, S.; GARCÍA, L.; LLORET, J.; BOSCH, I.; VEGA-RODRÍGUEZ, R. LoRaWAN Network for Fire Monitoring in Rural Environments. **Electronics**, v. 9, p. 531, mar. 2020.

SISINNI, E.; FERRARI, P.; FERNANDES CARVALHO, D.; RINALDI, S.; PASETTI, M.; FLAMMINI, A.; DEPARI, A. A LoRaWAN range extender for Industrial IoT. **IEEE Transactions on Industrial Informatics**, p. 1–1, 2019. Acesso em: 24 dez. 2019.

SLABICKI, M.; PREMSANKAR, G. **FLoRa Documentation**. 2018. Disponível em: <<https://flora.aalto.fi/>>. Acesso em: 4 jan. 2021.

SLABICKI, M.; PREMSANKAR, G.; DI FRANCESCO, M. Adaptive configuration of lora networks for dense IoT deployments. In: NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium. 2018. P. 1–9.

WARET, A.; KANEKO, M.; GUITTON, A.; EL RACHKIDY, N. LoRa Throughput Analysis With Imperfect Spreading Factor Orthogonality. **IEEE Wireless Communications Letters**, v. 8, n. 2, p. 408–411, 2019.

XIAO, L.; XIE, S. Device of Port Shore Power of Envelope Voltage of Wireless Seamless Connection. **IOP Conference Series: Earth and Environmental Science**, v. 189, p. 052018, nov. 2018. Acesso em: 7 jun. 2021.

ZYREN, J.; PETRICK, A. **Tutorial on Basic Link Budget Analysis**. 1998. Disponível em: <<https://www.semanticscholar.org/paper/Tutorial-on-Basic-Link-Budget-Analysis-Zyren-Petrick/16eeb0165db27a1b62981648f40282a94db05f90>>. Acesso em: 7 mai. 2021.