

UNIVERSIDADE FEDERAL DE SANTA CATARINA  
CENTRO TECNOLÓGICO  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA E ELETRÔNICA  
CURSO DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Gabriel Germino Martins de Jesus

**Avaliação do Desempenho do Algoritmo Adaptativo de Taxa de Dados de  
Redes LoRaWAN com Flexibilização da Margem do Enlace**

Florianópolis  
2021



Gabriel Germino Martins de Jesus

**Avaliação do Desempenho do Algoritmo Adaptativo de Taxa de Dados de  
Redes LoRaWAN com Flexibilização da Margem do Enlace**

Trabalho de Conclusão de Curso de Graduação em Engenharia Elétrica do Centro Tecnológico da Universidade Federal de Santa Catarina para a obtenção do título de Bacharel em Engenharia Elétrica.  
Orientador: Prof. Richard Demo Souza, Dr.

Florianópolis

2021

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

Jesus, Gabriel Germino Martins de  
Avaliação do desempenho do algoritmo adaptativo de taxa  
de dados de redes LoRaWAN com flexibilização da margem do  
enlace / Gabriel Germino Martins de Jesus ; orientador,  
Richard Demo Souza, 2021.  
56 p.

Trabalho de Conclusão de Curso (graduação) -  
Universidade Federal de Santa Catarina, Centro Tecnológico,  
Graduação em Engenharia Elétrica, Florianópolis, 2021.

Inclui referências.

1. Engenharia Elétrica. 2. Internet das coisas. 3.  
LoRaWAN. 4. Taxa de dados adaptativa. I. Demo Souza,  
Richard. II. Universidade Federal de Santa Catarina.  
Graduação em Engenharia Elétrica. III. Título.

Gabriel Germino Martins de Jesus

**Avaliação do Desempenho do Algoritmo Adaptativo de Taxa de Dados de  
Redes LoRaWAN com Flexibilização da Margem do Enlace**

Este Trabalho Conclusão de Curso foi julgado adequado para obtenção do Título de Bacharel em Engenharia Elétrica e aprovado em sua forma final pelo Curso de Engenharia Elétrica.

Florianópolis, 09 de Abril de 2021.

---

Prof. Jean Viane Leite, Dr.  
Coordenador do Curso

**Banca Examinadora:**

---

Prof. Richard Demo Souza, Dr.  
Orientador  
Universidade Federal de Santa Catarina

---

Prof. Carlos Aurélio Faria da Rocha, Dr.  
Avaliador  
Universidade Federal de Santa Catarina

---

*Eduardo A. Pfeifer*

Eng. Eduardo Augusto Pfeifer  
Avaliador  
Khomp



## **AGRADECIMENTOS**

Agradeço à minha mãe e minha irmã por todo o amor, respeito e apoio incondicionais por toda a minha vida. Sei que não teria chegado aqui se não fosse por assim e que, se ousou sonhar em ir mais longe, é pela certeza que me dão de ter para onde voltar, respirar fundo, e retomar o caminho se as coisas derem errado.

Aos queridos amigos e amigas que encontrei na universidade, da melhor das turmas da Engenharia Elétrica, da turma de calouros que disputou de perto esse título, das turmas anteriores e dos cursos vizinhos, agradeço por estar ao lado de vocês neste tempo pois isso foi o que permitiu que as lembranças desses anos fossem tão boas e que cada momento fosse muito bem aproveitado.

A todos os trabalhadores da Universidade Federal de Santa Catarina, agradeço por garantirem, da manutenção e limpeza à administração, o bom funcionamento da universidade e as estruturas para que eu e meus colegas tivéssemos um ensino de tanta qualidade ao nosso alcance.

Aos excelentes professores que tive ao longo do curso, agradeço por terem me mostrado que “não deixa de ser mágico só porque você sabe como funciona”, assim criando e reavivado em mim o interesse pela Engenharia Elétrica e suas áreas.

Aos professores Carlos Montez e Arliones Hoeller agradeço pela colaboração e orientação na produção dos dois artigos frutos deste trabalho.

Finalmente, ao professor Richard Demo Souza agradeço pelo ótimo exemplo, aulas excelentes, orientações precisas, paciência e atenção singulares.

Adoraria agradecer a cada um pessoalmente com um forte abraço, não fosse o Grande Inconveniente, mas espero que não precise ser adiado por muito mais tempo. Terminei este trabalho com um pouco mais de confiança no futuro, raro nos dias de hoje, por tudo o que aprendi com vocês. A todos e todas: muito obrigado!



## RESUMO

LoRaWAN é um protocolo de rede que permite a implementação de aplicações de redes de longo alcance e baixa potência para a Internet das Coisas. Seu esquema de modulação, LoRa, utiliza diversos parâmetros de transmissão que permitem comunicação bidirecional por longas distâncias, em uma relação de troca entre alcance e tempo de transmissão. As especificações de LoRaWAN sugerem o uso de um algoritmo adaptativo, o *Adaptive Data Rate* (ADR), para atribuir estes parâmetros dinamicamente aos seus usuários. O ADR usado por uma rede LoRaWAN aberta mundial, a *The Things Network*, seleciona os parâmetros para comunicação sem desconexão, diminuindo o tempo de transmissão e consumo energético. No entanto, a suas estimativas frequentemente otimistas resultam na seleção de parâmetros inadequados, especialmente quando os sinais são transmitidos em canais com muitas perdas. Estimativas de canal mais precisas poderiam melhorar a alocação de parâmetros, ao custo do aumento da complexidade do algoritmo. Através de simulações, mostrou-se que o parâmetro de margem de enlace do ADR pode ser selecionado para compensar essa estimativa falha sem adicionar complexidade ao algoritmo, garantindo a qualidade do serviço e ainda um consumo energético menor comparado a uma distribuição aleatória de parâmetros. Neste trabalho, uma modificação ao ADR é proposta, de forma que a seleção da margem de enlace seja feita em tempo de execução, sem necessidade de conhecimento prévio das configurações da rede e sem aumentar a complexidade nas computações feitas no servidor de rede.

**Palavras-chave:** Internet das coisas, LoRaWAN, taxa de dados adaptativa.



## ABSTRACT

LoRaWAN is a network protocol to support the deployment of Low-Power Wide-Area Network applications for the Internet of Things. Its modulation scheme, LoRa, uses different transmission parameters to allow long-range bidirectional communication by trading-off range for time-on-air (ToA). The LoRaWAN specification suggests using an Adaptive Data Rate (ADR) algorithm to assign transmission parameters to nodes dynamically. The ADR algorithm used by The Things Network, a global open network, selects transmission parameters for communication without disconnection while lowering ToA and energy consumption. However, its often optimistic link quality estimate drives the assignment of inadequate parameters, especially in lossy channels. More precise channel quality estimates would improve resource allocation at the cost of increased complexity at the network server. Through simulations, it is shown that the ADR link margin parameter can be used to compensate for inaccurate link quality estimates without increasing complexity, providing better service to nodes and maintaining low levels of energy consumption compared to a random parameter assignment. In this work, a modification to ADR that allows the selection of the link margin at run-time is proposed, increasing the network performance without the need of any prior knowledge and with no increase in the network server computational complexity.

**Keywords:** Internet of things, LoRaWAN, adaptive data rate.



## LISTA DE FIGURAS

Figura 1 – Janelas de recepção de dispositivos Classe A. . . . .	27
Figura 2 – Janelas de recepção de dispositivos Classe B. . . . .	28
Figura 3 – Janelas de recepção de dispositivos Classe C. . . . .	28
Figura 4 – Estrutura dos pacotes LoRaWAN. . . . .	29
Figura 5 – Fluxograma do ADR-TTN no servidor de rede. . . . .	30
Figura 6 – Validação da distribuição Nakagami- $m$ . . . . .	37
Figura 7 – Ilustração do modelo do sistema, com ED, GW e NS. . . . .	39
Figura 8 – DER média utilizando $margin\_db = 10$ dB, variando distância máxima dos EDs até o GW. . . . .	42
Figura 9 – Utilização de SF e $P_t$ quando $margin\_db = 10$ dB para o ADR-TTN, ADR+ e sem ADR, com $R = 1500$ m . . . . .	42
Figura 10 – Impacto de $margin\_db$ na DER média. . . . .	44
Figura 11 – Percentual de dispositivos que precisam de um dado número de <i>uplinks</i> para convergirem ao seu SF final, considerando uma $DER_{ref}$ de 90% e $R = 1500$ m. . . . .	47
Figura 12 – Utilização de SF e $P_t$ considerando uma $DER_{ref}$ de 90% e $R = 1500$ m, com as margens otimizadas para o ADR-TTN e ADR+, e com a margem selecionada dinamicamente para cada ED no ADRx. . . . .	48
Figura 13 – Gráfico de dispersão do valor final de $margin\_db$ no ADRx, considerando uma $DER_{ref}$ de 90% e $R = 1500$ m. . . . .	49
Figura 14 – Percentual de EDs com DER superior à referência de 80% utilizando os três algoritmos. . . . .	50
Figura 15 – DER média dos EDs variando $R$ para diferentes valores de $DER_{ref}$ , utilizando o ADRx. . . . .	51



## LISTA DE TABELAS

Tabela 1 – Taxa de <i>bits</i> , sensibilidade no rádio e SNR mínimo necessário para demodulação ( $SNR_{SF}$ ), de acordo com o Fator de Espalhamento (SF) para valores fixos de CR e BW. . . . .	26
Tabela 2 – Parâmetros de aplicação e perda de propagação. . . . .	41
Tabela 3 – Consumo energético quando $margin\_db = 10$ dB e $R = 1500$ m. . . . .	43
Tabela 4 – Consumo energético com as margens otimizadas para o ADR-TTN e ADR+, e com a margem selecionada dinamicamente para cada ED no ADRx, considerando uma $DER_{ref}$ de 90% e $R = 1500$ m. . . . .	48
Tabela 5 – Consumo energético com as margens otimizadas para o ADR-TTN e ADR+, e com a margem selecionada dinamicamente para cada ED no ADRx, considerando uma $DER_{ref}$ de 80% e $R = 3000$ m. . . . .	50



## LISTA DE ABREVIATURAS E SIGLAS

ADR	<i>Adaptive Data Rate</i>
BW	Largura de Banda
CR	Taxa de Código
CRC	Verificação Cíclica de Redundância
DER	Taxa de Extração de Dados
ED	Dispositivo Final
GW	<i>Gateway</i>
IoT	Internet das Coisas
IP	Protocolo de Internet
LPWAN	Rede de Longo Alcance e Baixa Potência
NS	Servidor de Rede
RSSI	Indicador da Intensidade do Sinal Recebido
SF	Fator de Espalhamento
SNR	Razão Sinal-Ruído
ToA	Tempo de Transmissão
TTN	<i>The Things Network</i>



## LISTA DE SÍMBOLOS

$m$	Parâmetro de desvanecimento do modelo Nakagami-m
$P_t$	Potência de transmissão
$N$	Número de mensagens recebidas necessárias para o cálculo do ADR-TTN, ADR+ e ADRx
$R_b$	Taxa de bits
ADR_ACK_LIMIT	Número máximo de <i>uplinks</i> sem resposta antes da requisição de confirmação
ADR_ACK_DELAY	Número máximo de <i>uplinks</i> sem resposta após a requisição de confirmação
$SNR_{SF}$	Sensibilidade do rádio para cada SF
$margin_{db}$	Margem de erro para considerada no cálculo do ADR-TTN, ADR+ e ADRx
$SNR_m$	Valor máximo de SNR dentre as últimas $N$ transmissões recebidas
$N_{step}$	Número de passos considerado pelos ADR-TTN, ADR+ e ADRx
$P_{tmin}$	Menor valor de $P_t$ permitido nas especificações regionais
$P_{tmax}$	Maior valor de $P_t$ permitido nas especificações regionais
$N_r$	Número de mensagens recebidas
$N_t$	Número de mensagens transmitidas
$PL(d)$	Perda de propagação para dada distância $d$
$\overline{PL}(d_0)$	Perda de propagação média para distância $d_0$
$d_0$	Distância de referência para cálculo da perda de propagação
$\eta$	Expoente de atenuação
$X_\sigma$	Variável aleatória de média $\mu$
$P_r$	Potência recebida
$G_r$	Ganhos na antena do transmissor
$G_r$	Ganhos na antena do receptor
$L$	Outras perdas na recepção
$h$	Variação instantânea da potência recebida
$\Gamma(\cdot)$	Função Gamma completa
$E_t$	Consumo total de energia de um dispositivo
$V_{CC}$	Tensão de alimentação do rádio
$I_{P_i}$	Corrente de alimentação para a $P_t$ utilizada na $i$ -ésima transmissão
$t_{SF_i}$	O tempo de transmissão para o SF utilizado na $i$ -ésima transmissão
$I_r$	Corrente de alimentação quando o rádio está em modo de recepção
$t_{r_i}$	Tempo de duração da recepção
$I_S$	Corrente de alimentação para os outros estados do rádio
$t_S$	Tempo de duração de cada um dos outros estados do rádio

$n$	Número de EDs simulados
$R$	Raio da área circular em que os EDs são dispostos
$1/\lambda_0$	Média da distribuição exponencial que determina o período de espera até que o ED faça sua primeira transmissão
$1/\lambda_1$	Média da distribuição exponencial que determina o período de espera até que o ED faça sua próxima transmissão
$p_a$	Número de símbolos nos preâmbulos dos pacotes
$p_l$	Número de <i>bytes</i> nos <i>payloads</i> dos pacotes
$DER_{ref}$	DER de referência para os cálculos no ADRx
$DER_{inst}$	DER alcançada entre cálculos do ADRx
$FCnt$	Número do último pacote recebido no NS
$FCntUp$	Vetor que armazena o número dos pacotes recebidos entre execuções do ADR
$I$	Número atual de pacotes recebidos no NS desde a última execução do ADR

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>21</b>
1.1	OBJETIVOS	22
<b>1.1.1</b>	<b>Objetivo Geral</b>	<b>22</b>
<b>1.1.2</b>	<b>Objetivos Específicos</b>	<b>22</b>
1.2	PUBLICAÇÕES	23
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>25</b>
2.1	LORA E LORAWAN	25
<b>2.1.1</b>	<b>LoRa</b>	<b>25</b>
<b>2.1.2</b>	<b>LoRaWAN</b>	<b>26</b>
2.1.2.1	Classes de Dispositivos	26
2.1.2.2	Pacotes LoRaWAN	27
2.2	ADAPTIVE DATA RATE	29
<b>2.2.1</b>	<b>ADR-TTN</b>	<b>29</b>
<b>2.2.2</b>	<b>Trabalhos Relacionados ao ADR no Servidor de Rede</b>	<b>31</b>
<b>3</b>	<b>PREPARAÇÃO DAS SIMULAÇÕES</b>	<b>35</b>
3.1	<i>FRAMEWORK FOR LORA</i>	35
<b>3.1.1</b>	<b>Simulações</b>	<b>35</b>
<b>3.1.2</b>	<b>Aquisição de Métricas</b>	<b>35</b>
<b>3.1.3</b>	<b>Modelo do Canal</b>	<b>36</b>
<b>3.1.4</b>	<b>Modelo de Consumo de Energia</b>	<b>37</b>
3.2	CENÁRIOS DE SIMULAÇÃO	38
<b>4</b>	<b>RESULTADOS DE SIMULAÇÃO</b>	<b>41</b>
4.1	SIMULAÇÕES COM MARGEM PADRÃO	41
4.2	SIMULAÇÕES COM MARGEM VARIÁVEL	43
<b>5</b>	<b>PROPOSTA DO ADRx</b>	<b>45</b>
5.1	DESCRIÇÃO DO ALGORITMO	45
5.2	RESULTADOS DE SIMULAÇÕES COM ADRx	46
<b>6</b>	<b>CONCLUSÃO</b>	<b>53</b>
	<b>REFERÊNCIAS</b>	<b>55</b>



## 1 INTRODUÇÃO

A Internet das Coisas (IoT), na qual aparelhos eletrônicos, máquinas e objetos do dia a dia fazem medições de seus arredores e compartilham informação com outros dispositivos ou seres humanos através da Internet, torna-se realidade com o número crescente de objetos conectados. Era previsto que estes dispositivos tivessem superado o número de dispositivos móveis de interface humana (*smartphones* e *tablets*) até o final de 2020, tornando possível o surgimento de uma indústria de IoT com receita de mais de 4 trilhões de dólares até 2024 (RAZA; KULKARNI; SOORIYABANDARA, 2017).

Um dos desafios na IoT é possibilitar aplicações que demandam um número massivo de dispositivos de baixo custo, alimentados por baterias de longa vida útil, comunicando-se a longa distâncias, em aplicações como medição inteligente e monitoramento ambiental, por exemplo (RAZA; KULKARNI; SOORIYABANDARA, 2017). Apesar de ser possível, utilizando tecnologias estabelecidas, realizar comunicações a longas distâncias (p.ex. comunicação celular), operar com baixo consumo de potência (p.ex. *Bluetooth low energy*), e até mesmo assegurar o funcionamento de números massivos de dispositivos (p.ex. *Multiple Input Multiple Output* massivo), algumas aplicações demandam que todas essas características sejam atendidas pela mesma tecnologia. O conceito de uma Rede de Longo Alcance e Baixa Potência (LPWAN) foi criado para guiar o desenvolvimento destas tecnologias. Uma tecnologia de LPWAN proeminente é LoRaWAN (LORA ALLIANCE, 2018a), um protocolo de rede aberto. Uma rede LoRaWAN é composta de três elementos principais: o nó, ou Dispositivo Final (ED); o *Gateway* (GW); e o Servidor de Rede (NS), onde reside a maior parte da complexidade computacional da rede.

LoRaWAN é baseado em LoRa (SEMTECH, 2015), uma tecnologia de camada física proprietária da empresa Semtech, que utiliza espalhamento espectral por *chirps*, permitindo a demodulação de mensagens com baixa Razão Sinal-Ruído (SNR), ao custo de baixas taxas de dados. Em LoRa, a taxa de dados é modificada selecionando uma das combinações de Largura de Banda (BW) (125-, 250- ou 500 kHz) e SF (que varia entre 7 e 12 (SF7-SF12)) disponíveis. A escolha por SF maiores e BW menores tem efeitos significativos no Tempo de Transmissão (ToA) das transmissões (RAZA; KULKARNI; SOORIYABANDARA, 2017), impactando tanto no consumo de energia dos nós, quanto no número de colisões no receptor (BOR *et al.*, 2016). Por outro lado, conforme o SF aumenta e a BW diminui, a SNR necessária no receptor diminui, resultando numa melhoria de sensibilidade do rádio. Além da escolha de BW e SF, em LoRaWAN a potência de transmissão ( $P_t$ ) dos dispositivos também pode ser modificada, assumindo valores discretos que dependem das especificações da região (LORA ALLIANCE, 2018b).

A atribuição correta dos parâmetros de transmissão é fundamental para atender os requisitos de LPWAN de conectividade massiva e longo alcance. Se por um lado a escolha por atribuir SF12 e  $P_t$  máximo a todos os ED resulte numa alta probabilidade de conexão

até mesmo em longas distâncias, o número de colisões nos GW aumentaria muito devido ao ToA excessivo das transmissões (BOR *et al.*, 2016). Além disso, o número de transmissões realizadas por cada ED diminuiria devido a restrições de *duty-cycle* (proporção a qual o dispositivo permanece ativo) imposta por regulamentações regionais (LORA ALLIANCE, 2018b). Por fim, também causado pelo ToA excessivo, os nós consumiriam muita energia, reduzindo o tempo de vida de suas baterias.

Para a seleção adequada destes parâmetros de transmissão, o protocolo LoRaWAN inclui o mecanismo de *Adaptive Data Rate* (ADR), visando otimizar esta alocação de recursos e alcançar comunicação de longa distância, baixo consumo energético e escalabilidade (LORA ALLIANCE, 2018a). O ADR é executado assincronamente no NS e nos ED, funcionando de maneira diferente em cada caso. No lado dos nós, o ADR seleciona parâmetros de transmissão mais conservadores ( $P_t$  ou SF maiores) caso aparente ter perdido conexão com o NS, isto é, se não tiver recebido nenhum *downlink* (transmissão do NS para o ED) após um certo número de *uplinks* (transmissão do ED para o NS). Embora o algoritmo seja definido para os nós, o lado do NS do ADR não é explicitado, delegando-o ao operador da rede.

Apesar de o lado do NS do ADR não ser padronizado, a *The Things Network* (TTN) (THE THINGS NETWORK, 2019), uma rede LoRaWAN aberta, mantida pela comunidade e com presença mundial, implementa uma versão atualizada de um algoritmo sugerido pela Semtech (SEMTECH, 2016), aqui referido por ADR-TTN. Este algoritmo atribui novos parâmetros de transmissão aos nós baseado em estimativas das condições do canal obtidas pelas medidas de SNR dos  $N$  pacotes recebidos mais recentes. O ADR-TTN é capaz de atribuir dinamicamente aos nós parâmetros de transmissão razoáveis, mas diversos trabalhos de pesquisa recentes demonstram que ainda há melhorias consideráveis a serem feitas.

## 1.1 OBJETIVOS

Foram identificados o objetivo geral e os objetivos específicos do trabalho, apresentados nas subseções seguintes.

### 1.1.1 Objetivo Geral

O objetivo deste trabalho é avaliar o desempenho de propostas existentes para o algoritmo do ADR no NS, e identificar possíveis melhorias na implementação do algoritmo usado pela TTN.

### 1.1.2 Objetivos Específicos

Os objetivos específicos identificados que guiam a elaboração deste trabalho são:

- a) Fazer a revisão bibliográfica de LoRa e LoRaWAN e compreender suas características principais;
- b) Fazer a revisão bibliográfica das propostas para o algoritmo do ADR no NS, identificando suas vantagens e desvantagens;
- c) Estudar um simulador computacional para redes LoRaWAN já validado e modificá-lo para se adequar aos requisitos do trabalho;
- d) Avaliar o algoritmo da TTN no ambiente de simulação e identificar suas vantagens e desvantagens;
- e) Propor melhorias ao algoritmo da TTN e avaliá-las comparando à versão anterior.

## 1.2 PUBLICAÇÕES

O desenvolvimento deste trabalho resultou na publicação de dois artigos científicos:

O primeiro deles, com título “*Improving LoRaWAN Performance Through Adaptive Data Rate Parameter Selection*”<sup>1</sup>, foi apresentado no XXXVIII Simpósio Brasileiro de Telecomunicações, realizado pela Sociedade Brasileira de Telecomunicações em Novembro de 2020. Trata-se de um artigo de iniciação científica, de duas páginas, onde são apresentados e discutidos os resultados obtidos no Capítulo 4 deste trabalho de conclusão de curso.

O segundo artigo, com título “*LoRaWAN Adaptive Data Rate with Flexible Link Margin*”<sup>2</sup>, foi aceito para publicação no periódico *Internet of Things Journal*, do Instituto de Engenheiros Eletricistas e Eletrônicos. O artigo está disponível no repositório do instituto, em acesso antecipado, e será publicado em uma edição futura do periódico. Este artigo dá continuidade ao anterior, abordando os resultados apresentados e discutidos nos Capítulos 4 e 5 deste trabalho.

---

<sup>1</sup> Disponível em: <http://www.sbrt.org.br/sbrt2020/papers/1570655507.pdf>.

<sup>2</sup> Disponível em: <https://ieeexplore.ieee.org/document/9240995>.



## 2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentadas as principais características das tecnologias LoRa e LoRaWAN, e do algoritmo ADR-TTN.

### 2.1 LORA E LORAWAN

LoRaWAN é um protocolo de rede desenvolvido pela LoRa Alliance para ser usado com LoRa em sua camada física. As características de LoRa são exploradas pelo protocolo para alcançar os requisitos estabelecidos de LPWAN (RAZA; KULKARNI; SOORIYABANDARA, 2017). Nesta seção são apresentados os principais aspectos de LoRa e uma descrição do protocolo LoRaWAN.

#### 2.1.1 LoRa

LoRaWAN utiliza a modulação LoRa na banda industrial, científica e médica sub-GHz, com valores de BW de 125-, 250- e 500 kHz. A modulação disponibiliza códigos corretores de erro com Taxa de Código (CR) variando de 4/5 a 4/8 (SEMTECH, 2015). Os rádios LoRa apresentam, ainda, valores de  $P_t$  discretos, que dependem da região<sup>1</sup> (LORA ALLIANCE, 2018b).

Um elemento chave da modulação LoRa são os valores de SF, que têm impacto direto no ToA do sinal transmitido. Quanto maior o SF, mais robusto é o sinal, podendo ser demodulado bem abaixo do piso de ruído (RAZA; KULKARNI; SOORIYABANDARA, 2017). São especificados sete SF, mas apenas seis deles são utilizados por LoRaWAN: SF7-SF12 (SEMTECH, 2015). Os SF são semi-ortogonais entre si, permitindo que múltiplos pacotes sejam recebidos ao mesmo tempo, praticamente sem colisões, desde que sejam transmitidos utilizando SF diferentes (BOR *et al.*, 2016). A escolha do SF tem impacto significativo no desempenho das redes, pois a SNR necessária no GW diminui com o aumento do SF, permitindo que os sinais sejam sujeitos a atenuações mais fortes, possibilitando comunicação a distâncias maiores. Por outro lado, o ToA aumenta com o SF, resultando em uma relação de troca entre alcance e ToA, impactando diretamente no consumo energético, taxa de dados, e até mesmo em recepções bem sucedidas, já que transmissões mais longas têm maior probabilidade de colisão (BOR *et al.*, 2016).

A escolha de SF, BW e CR definem a taxa de *bits*  $R_b$  (SEMTECH, 2015) de acordo com

$$R_b = SF \cdot \frac{BW}{2^{SF}} \cdot CR. \quad (1)$$

Os valores de  $R_b$  resultantes da escolha de SF para um dado CR de 4/5 e BW de 125 kHz, com valores de sensibilidade específicos para o rádio Semtech SX1272, são apresentados

<sup>1</sup> Nas simulações computacionais deste trabalho foram considerados os valores da banda EU868, com  $P_t$  variando entre 2 e 14 dBm, em passos de 2 dB.

na Tabela 1.

Tabela 1 – Taxa de *bits*, sensibilidade no rádio e SNR mínimo necessário para demodulação ( $SNR_{SF}$ ), de acordo com o SF para valores fixos de CR e BW.

SF	$R_b$ (kbps)	Sensibilidade no receptor (dBm)	$SNR_{SF}$ (dB)
7	5,47	-124	-7,5
8	3,12	-127	-10
9	1,76	-130	-12,5
10	0,98	-133	-15
11	0,44	-135	-17,5
12	0,25	-137	-20

Fonte – Semtech (2015, 2017)

### 2.1.2 LoRaWAN

O protocolo LoRaWAN (LORA ALLIANCE, 2018a) tem uma topologia estrela de estrelas, de forma que cada ED é conectado a tantos GW quanto consiga alcançar, sem a necessidade de *handover*. Os GW encaminham as mensagens decodificadas, por uma conexão Protocolo de Internet (IP) padrão, ao NS, que é responsável por remover os pacotes replicados, agendar *downlinks*, e selecionar para o nó, se assim requisitado pelo mesmo, novos parâmetros de transmissão para futuros *uplinks*. Todos os dados transmitidos pelos nós são criptados, e o conteúdo útil (p. ex. uma medição de temperatura realizada pelo dispositivo) é visível apenas no servidor de aplicação.

Seguindo o protocolo de acesso ao meio ALOHA (GOLDSMITH, 2005), os nós transmitem sempre que têm dados disponíveis, a não ser que estejam restritos por *duty-cycle*. Os nós funcionam, na maioria das aplicações, de forma assíncrona ao NS, alternando pseudo-aleatoriamente entre os canais de frequência disponíveis. LoRaWAN ainda define três classes de dispositivos, para diferentes requisitos de aplicações (LORA ALLIANCE, 2018a).

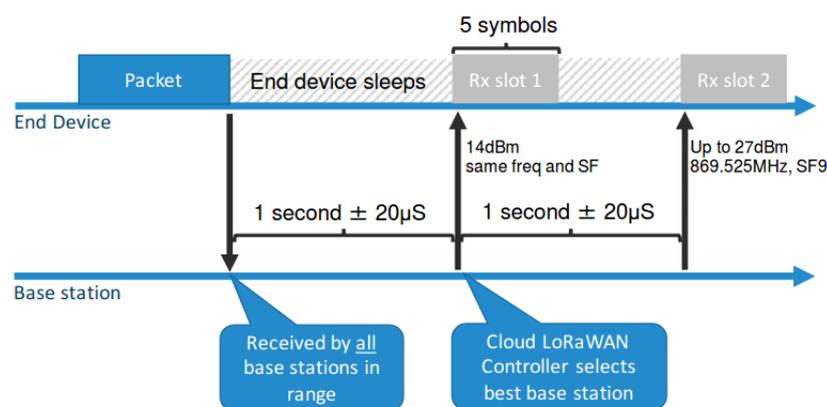
#### 2.1.2.1 Classes de Dispositivos

Em LoRaWAN são definidas classes A, B e C de dispositivos para utilização em aplicações específicas (LORA ALLIANCE, 2018a). Apesar de todos os dispositivos disponibilizarem a configuração classe A por padrão, a implementação das classes B e C é opcional, sendo que estas outras classes permitem mais oportunidades de *downlink*, reduzindo latência ao custo de aumentar o consumo energético.

Na Figura 1 são apresentadas as janelas de recepção dos dispositivos **Classe A**. Estes dispositivos só recebem *downlinks* em duas janelas de tempo especificadas logo após um *uplink*. O *downlink* na primeira janela de recepção (RX1) é transmitido em um canal

de frequência e SF que são função dos parâmetros do *uplink* inicial. Caso o ED não receba um *downlink* em RX1, é aberta uma segunda janela (RX2), em que são utilizados canais de frequência, BW e SF pré-definidos (na banda AUS915, na qual LoRaWAN opera no Brasil, utiliza-se o canal 923.3 MHz, BW= 500 kHz e SF12 (LORA ALLIANCE, 2018b)), mas que podem ser alterados a partir de comandos próprios para isto. O servidor precisa esperar um *uplink* futuro caso deseje realizar um *downlink*. Os dispositivos da classe A são os com menor consumo de energia e também os mais comuns pois todos os dispositivos fabricados precisam oferecer esta configuração.

Figura 1 – Janelas de recepção de dispositivos Classe A.



Fonte – Orange Connected Objects & Solutions (2016)

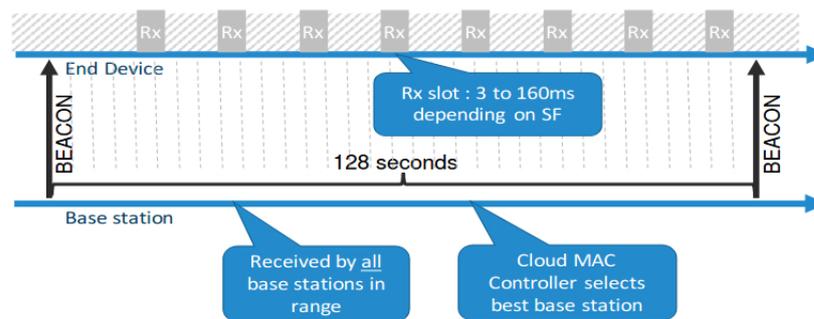
Caso a aplicação demande baixa latência, mas ainda há preocupação com o consumo de energia, é recomendado que se utilize os dispositivos da **Classe B**. Nesta configuração os dispositivos abrem janelas de recepção em intervalos de tempo fixos, ilustrados na Figura 2, dessa forma garantindo que o servidor possa decidir quando realizar *downlinks*. A sincronização entre servidor e nós é feita por meio de um *beacon* transmitido frequentemente a partir do(s) GW(s).

Por último, para aplicações em que a otimização do consumo de energia não é levado em consideração, podem ser utilizados dispositivos da **Classe C**. São abertas janelas de recepção iguais às RX2 definidas na Classe A a todo momento, exceto quando o dispositivo está transmitindo ou em uma janela de recepção RX1 subsequente a um *uplink*, como é ilustrado na Figura 3.

### 2.1.2.2 Pacotes LoRaWAN

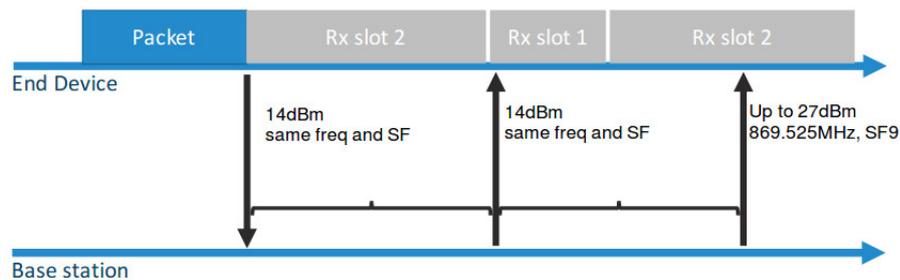
Os pacotes LoRaWAN são compostos de duas partes: o preâmbulo e o *payload*. Opcionalmente pode-se adicionar um cabeçalho que indicará o tamanho do *payload*, a taxa do código corretor de erros e a presença de uma Verificação Cíclica de Redundância (CRC).

Figura 2 – Janelas de recepção de dispositivos Classe B.



Fonte – Orange Connected Objects & Solutions (2016)

Figura 3 – Janelas de recepção de dispositivos Classe C.

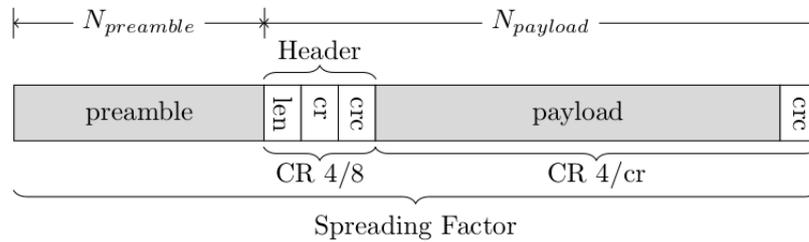


Fonte – Orange Connected Objects & Solutions (2016)

O cabeçalho, por sua vez, possui sua própria CRC que indicará sua integridade. O tamanho do preâmbulo pode variar de 6 a 65535 símbolos (com 4,25 símbolos extras adicionados pelo rádio), e o *payload* contém de 1 a 255 *bytes* (BOR *et al.*, 2016). Opcionalmente, os últimos 16 *bits* do *payload* contém o valor da CRC. Na Figura 4 esta estrutura é ilustrada, sendo em cinza as partes obrigatórias do pacote, e, em branco, as opcionais.

Em LoRaWAN, um pacote colide quando todas as condições seguintes são verdadeiras: a recepção de um pacote novo começa enquanto o pacote anterior ainda está sendo recebido (colisão no tempo); os pacotes utilizam o mesmo SF (colisão em SF); e os pacotes são transmitidos no mesmo canal de frequência (colisão em frequência). Entretanto, mesmo que ocorra uma colisão, um dos pacotes ainda pode ser decodificado devido ao efeito de captura presente em modulações em frequência (como é o caso de LoRa), possibilitando que o pacote com sinal mais forte (com SNR pelo menos 6 dB maior que o outro) seja recuperado, desde que o pacote mais fraco termine de ser transmitido antes dos últimos

Figura 4 – Estrutura dos pacotes LoRaWAN.



Fonte – Bor *et al.* (2016)

símbolos do pré-amplio do pacote mais forte (BOR *et al.*, 2016).

## 2.2 ADAPTIVE DATA RATE

Para alocar dinamicamente o SF e a  $P_t$  entre os nós, uma rede LoRaWAN pode utilizar o algoritmo ADR, que é executado tanto no NS quando nos nós. O algoritmo no lado dos ED é especificado pela LoRa Alliance, mas o ADR no lado do NS fica a cargo do operador de rede (LORA ALLIANCE, 2018a).

No lado do ED, um contador monitora a conectividade do dispositivo. Para cada nova transmissão, o contador é incrementado, retornando a 0 sempre que um *downlink* é recebido. Caso o contador chegue a um limiar (ADR\_ACK\_LIMIT), os pacotes subsequentes fazem a requisição de uma resposta por parte do NS. Após um número de transmissões com a requisição sem resposta (ADR\_ACK\_DELAY), o dispositivo aumenta sua  $P_t$  para o próximo valor discreto disponível, ou aumenta SF se o valor de  $P_t$  for máximo. Este processo é repetido a cada ADR\_ACK\_DELAY até que o dispositivo receba um *downlink*, ou o valor de SF seja máximo. Tanto ADR\_ACK\_LIMIT quanto ADR\_ACK\_DELAY têm valores configuráveis, e são iguais a 32 por padrão.

### 2.2.1 ADR-TTN

A Semtech, empresa proprietária da modulação LoRa, apresenta um algoritmo de referência, cuja versão atualizada é o ADR-TTN, implementado pela TTN, e de forma muito similar pelas redes LoRa Server (FINNEGAN; FARRELL; BROWN, 2020). Devido a ampla cobertura das redes TTN, o ADR-TTN tem sido considerado como algoritmo padrão em trabalhos de pesquisa.

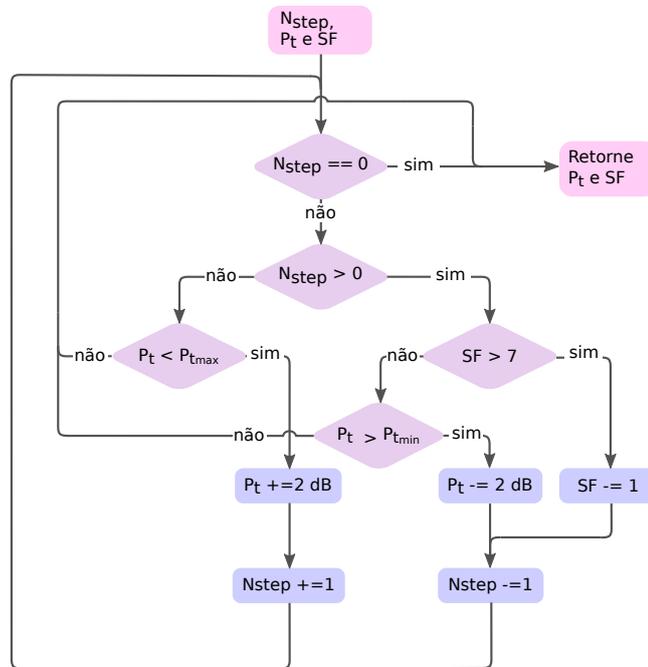
Para executar o ADR-TTN, o NS avalia a qualidade do enlace a partir das medições de SNR feitas nos GW. O algoritmo calcula o número de passos para ajustar  $P_t$  e SF levando em conta, além da estimativa da qualidade do enlace, a sensibilidade do rádio para

cada SF ( $SNR_{SF}$ ) (SEMTECH, 2017), e uma margem de erro ( $margin\_db$ ). Para cada passo calculado, o NS instrui o ED a aumentar ou diminuir SF e/ou  $P_t$ , ou aumentar  $P_t$ . A estimativa da qualidade do enlace é feita considerando o maior valor de SNR ( $SNR_m$ ) dentre as últimas  $N$  transmissões recebidas. O número de passos  $N_{step}$  é dado por

$$N_{step} = \lfloor \frac{SNR_m - SNR_{SF} - margin\_db}{3} \rfloor. \quad (2)$$

Com este resultado, SF e  $P_t$  são modificados de acordo com o fluxograma apresentado na Figura 5, em que  $P_{t_{min}}$  e  $P_{t_{max}}$  são os valores mínimos e máximos de  $P_t$  permitidos nas especificações regionais.

Figura 5 – Fluxograma do ADR-TTN no servidor de rede.



Fonte – Do autor, baseado em The Things Network (2019).

O valor padrão para  $N$  é 20 transmissões, enquanto que os valores típicos para  $margin\_db$  variam entre 5 e 15 dB (THE THINGS NETWORK, 2019; SEMTECH, 2016; SLABICKI; PREMSANKAR; DI FRANCESCO, 2018). Embora  $SNR_m$  varie de acordo com as condições do canal e a distância do ED ao(s) GW(s), os valores de  $SNR_{SF}$  são predeterminados de acordo com o receptor. Para uma BW fixa de 125 kHz, na Tabela 1 estão apresentados os valores de sensibilidade do Semtech SX1272, o rádio que é modelado no simulador LoRaWAN que será utilizado neste trabalho. Com cada aumento do SF, a sensibilidade do rádio diminui em aproximadamente 3 dB. Por fim, o tempo de convergência de cada lado do algoritmo é diferente, pois no lado dos ED são necessárias pelo menos  $ADR\_ACK\_LIMIT + ADR\_ACK\_DELAY$  transmissões para que o algoritmo comece a

modificar os parâmetros, enquanto que no NS, no melhor dos casos, são necessárias apenas  $N$  mensagens recebidas (RAZA; KULKARNI; SOORIYABANDARA, 2017).

Uma característica interessante do ADR-TTN é que não é permitido que o NS aumente o SF dos dispositivos, de forma a prevenir o mau uso de taxas de dados menores, que resultam em ToA maiores (SEMTECH, 2016). Caso o ED tenha uma conexão ruim, o NS pode apenas aumentar sua  $P_t$  nas próximas transmissões, sendo permitido apenas ao ED decidir por aumentar seu SF. Além disso, o ADR-TTN é projetado de forma que os nós explorem todos os valores de  $P_t$  disponíveis quando utilizam SF7, e utilizem apenas  $P_{t_{max}}$  em SF maiores, uma vez que, utilizando o menor SF, aumentar  $P_t$  é mais eficiente energeticamente, tendo praticamente o mesmo efeito em termos de conectividade que aumentar o SF.

### 2.2.2 Trabalhos Relacionados ao ADR no Servidor de Rede

A literatura apresenta diversas propostas para a alocação de recursos em redes LoRaWAN, desde pequenas mudanças no modo como o ADR-TTN estima as condições do canal, até abordagens radicalmente diferentes nos algoritmos. Nesta seção alguns destes trabalhos são apresentados, destacando as características únicas das propostas.

Para distribuir os SF entre os nós, Cuomo *et al.* (2017) exploram o Indicador da Intensidade do Sinal Recebido (RSSI) no GW de pacotes transmitidos, atribuindo valores maiores de SF a nós com valores de RSSI menores. Para determinar a proporção de dispositivos operando com cada SF, os autores apresentam dois métodos distintos: EXPLoRa-SF, em que a distribuição de SF é uniforme entre os dispositivos; e o EXPLoRa-AT, em que a alocação de SF é feita de modo a equalizar o ToA total de cada SF, resultando num número menor de dispositivos utilizando SF mais altos. Comparado ao ADR-TTN, EXPLoRa-SF é capaz de atender um número maior de ED conectados, ao custo de colisões mais frequentes, enquanto que o EXPLoRa-AT compensa o aumento significativo de complexidade nos cálculos no NS ao garantir menos colisões e melhor desempenho.

De forma similar ao algoritmo EXPLoRa-AT, Abdelfadeel, Cionca e Pesch (2018) propõem o FADR, que distribui os SF de modo a equalizar a probabilidade de colisões de pacotes. Além disso, os autores também propõem um método para distribuir os valores discretos de  $P_t$  entre os nós de modo a garantir uma probabilidade aproximadamente igual a todos os dispositivos de terem suas mensagens recebidas, independentemente de sua distância ao GW, atingindo o critério de equidade<sup>2</sup> estipulado. As simulações apresentadas pelos autores mostram que o FADR supera o desempenho do EXPLoRa-SF, em termos de equidade, em quase todos os cenários.

Park *et al.* (2020) propõem o algoritmo EARN, que também inclui a seleção dinâmica entre os CR disponíveis na modulação LoRa. O método leva em conta a probabilidade

<sup>2</sup> Em inglês, *fairness*, daí o “F” em FADR.

de um pacote ser perdido ao considerar a qualidade do enlace e a utilização dos SF pelos outros dispositivos da rede, atribuindo ao ED uma combinação de SF,  $P_t$  e CR com maior probabilidade de obter comunicação confiável com o NS. Opcionalmente, uma margem de erro variável, baseada no desvio padrão das medidas de SNR, pode ser adicionada à estimativa de qualidade de enlace, compensando pela variação de potência recebida. Quando um canal com pouca variação é considerado, EARN tem desempenho similar ao FADR em termos de Taxa de Extração de Dados (DER) (razão entre o número de pacotes transmitidos e recebidos) e eficiência energética, com ambos superando o desempenho do ADR-TTN consideravelmente nestas métricas.

Kim e Yoo (2018) propõem um algoritmo para otimizar o *throughput* ao explorar a semi-ortogonalidade dos SF para criar novos canais. Para a avaliação do método, apenas os SF7-SF9 são utilizados, pois os autores consideram que, em uma situação prática, os nós raramente utilizam SF mais altos quando há múltiplos GW. O algoritmo proposto é capaz de fornecer maior *throughput* comparado com métodos similares ao EXPLoRa-SF e EXPLoRa-AT, por alocar os nós principalmente em SF menores. Apesar do aumento em *throughput*, a confiabilidade das transmissões é menor devido a desconexões, já que a sensibilidade nos GW degrada com SF menores. Além disso, à medida que mais dispositivos são adicionados à rede, o tempo de convergência do algoritmo aumenta significativamente para dar conta de realocar todos os ED.

Marais, Malekian e Abu-Mahfouz (2019) avaliam o algoritmo proprietário da empresa Multitech em uma instalação física experimental, mostrando que o desempenho do algoritmo é pior do que a alocação manual dos parâmetros de transmissão. Há pouca informação disponível sobre o algoritmo, mas ele se comporta de forma diferente ao ADR-TTN, distribuindo os ED quase exclusivamente entre o maior e o menor SF disponível.

Finnegan, Farrell e Brown (2020) avaliam o desempenho do ADR-TTN utilizando o simulador NS-3, e propõem modificações ao algoritmo. Primeiro, o NS realiza uma estimativa da DER e instrui aos ED que aumentem seus SF caso a estimativa esteja abaixo de 80%. No lado do NS, depois de uma mudança de SF, caso haja medidas de SNR suficientes, os comandos do ADR podem ser antecipados. Estas modificações resultam em uma melhora considerável de desempenho tanto em termos de DER, quanto de tempo de convergência, quando comparado ao ADR-TTN padrão.

Li, Raza e Khan (2018) avaliam o ADR-TTN com foco no tempo de convergência do algoritmo, que difere no NS e no ED, com o primeiro convergindo mais rapidamente. Os autores concluem que reduzir os valores de `ADR_ACK_LIMIT` e `ADR_ACK_DELAY` resulta em uma convergência mais rápida, especialmente quando a qualidade do enlace é baixa. Por outro lado, modificar o valor de  $N$  para executar o ADR-TTN não traz nenhuma melhoria significativa.

Slabicki, Preamsankar e Di Francesco (2018) apresentam FLoRa, um simulador de LoRaWAN baseado em OMNeT++, juntamente com a proposta do ADR+, uma modificação

simples e eficiente ao ADR-TTN. No lugar de utilizar a maior medição de SNR dos últimos  $N$  pacotes recebidos para estimar a qualidade do canal, ADR+ adota uma abordagem mais conservadora considerando a média destas medições, superando o desempenho do ADR-TTN tanto em DER quanto em eficiência energética.

Garg (2019) realizou uma investigação experimental extensa do algoritmo ADR-TTN, propondo reduzir o número de colisões ao restringir o número de canais de frequência que um ED pode utilizar, e evitar desconexão entre ED e GW ao definir um limiar para a variável  $N_{step}$  utilizada nos cálculos do ADR-TTN. Este limiar depende do fato de o ED em questão ter seus pacotes recebidos por múltiplos GW ou apenas um. Estas modificações aumentam a DER significativamente, especialmente porque o NS se torna mais conservador na alocação de recursos, como ocorre com o ADR+. Por outro lado, caso a qualidade do enlace seja ruim, as modificações não apresentam oportunidades de melhoria na alocação de recursos, delegando ao lado lento do algoritmo a responsabilidade de melhorar o desempenho, pois aproximadamente em 99% das vezes o NS não trabalha ativamente para selecionar valores de  $P_t$  maiores.



### 3 PREPARAÇÃO DAS SIMULAÇÕES

Para avaliar o funcionamento do ADR-TTN, o simulador de redes LoRaWAN FLoRa foi escolhido (SLABICKI; PREMSANKAR; DI FRANCESCO, 2018). No trabalho científico que apresenta o simulador também é apresentado o ADR+, uma modificação simples ao ADR-TTN, em que  $SNR_m$  assume o valor médio de SNR dentre as  $N$  medições mais recentes. Neste capítulo o simulador é apresentado brevemente, as modificações feitas ao programa são destacadas, e é feita a descrição do cenário considerado nas simulações.

#### 3.1 FRAMEWORK FOR LORA

O *Framework for LoRa*, ou FLoRa, é um simulador de redes LoRaWAN desenvolvido por Slabicki, Preamsankar e Di Francesco (2018). O simulador utiliza, no OMNeT++, um conjunto de bibliotecas para simulação de redes, e o *framework* INET para o *backhaul* da rede. O simulador é de código aberto e simula aspectos de LoRaWAN, como a camada física, o protocolo de controle de acesso ao meio, bem como elementos da rede como os GW, nós e NS, além de apresentar um modelo de consumo de energia de cada ED.

##### 3.1.1 Simulações

O OMNeT++ é um simulador de eventos discretos, ou seja, o sistema é considerado imutável entre eventos. As simulações são configuradas em arquivos de extensão `ini`, onde são escolhidos todos os parâmetros variáveis nos ED, GW e NS, como posição dos dispositivos, SF e  $P_t$  iniciais, modelo de perda de propagação, entre outros. Costuma-se definir uma configuração geral para as simulações, onde são definidos aspectos comuns aos diversos cenários avaliados. Para diferenças menores nos cenários, são definidas configurações adicionais que copiam todos os parâmetros da configuração geral, exceto as alterações. As simulações podem rodar tanto em um ambiente gráfico, quanto em linha de comando diretamente no terminal. Além de as simulações serem mais rápidas na linha de comando, é possível rodar configurações diferentes paralelamente.

##### 3.1.2 Aquisição de Métricas

O OMNeT++ entrega os resultados das simulações de três maneiras diferentes: vetores, escalares e histogramas. Os componentes dos vetores são atualizados a cada evento relacionado a ele; os escalares a qualquer momento da simulação definido na programação (geralmente no início ou no fim); e os histogramas sempre ao final da simulação. As variáveis que são armazenadas como vetores são aquelas relevantes para a análise dinâmica do sistema simulado. No caso do FLoRa, quando o ADR é ativado, existe um vetor para SF para cada um dos nós, assim como  $P_t$ , RSSI e SNR, por exemplo. Pode-se exportar

os resultados para arquivos em formato `csv` e analisá-los com outras ferramentas, como MATLAB e Python.

Os escalares costumam ser aqueles em que há interesse nos valores ao fim da simulação. Um escalar importante para este trabalho é a DER, definido como  $N_r/N_t$ , sendo  $N_r$  o número de mensagens recebidas (sem considerar réplicas), e  $N_t$ , mensagens transmitidas (sem considerar retransmissões). Nativamente, são coletadas DER para cada SF e a DER média da rede.

Como incremento ao simulador, foram adicionados vetores e escalares que permitem a extração direta da DER de cada ED individualmente. Para isso, a parte do código que modelava o NS foi modificada de forma que o servidor acompanhe a DER individual de cada ED a partir do *framecounter* contido no cabeçalho dos pacotes LoRaWAN. Além de ser mais conveniente para a extração e análise dos resultados ao final da simulação, o conhecimento da DER por parte do NS tornou-se útil para a modificação ao ADR-TTN proposta no trabalho.

### 3.1.3 Modelo do Canal

O modelo de perda de propagação normalmente utilizado no FLoRa é o log-normal com sombreamento, mas os desenvolvedores também implementaram outros modelos, como o Hata-Okumura. No caso do log-normal com sombreamento, Para calcular a perda de potência do transmissor ao receptor utiliza-se, em dB, a equação

$$PL(d) = \overline{PL}(d_0) + 10\eta \log\left(\frac{d}{d_0}\right) + X_\sigma, \quad (3)$$

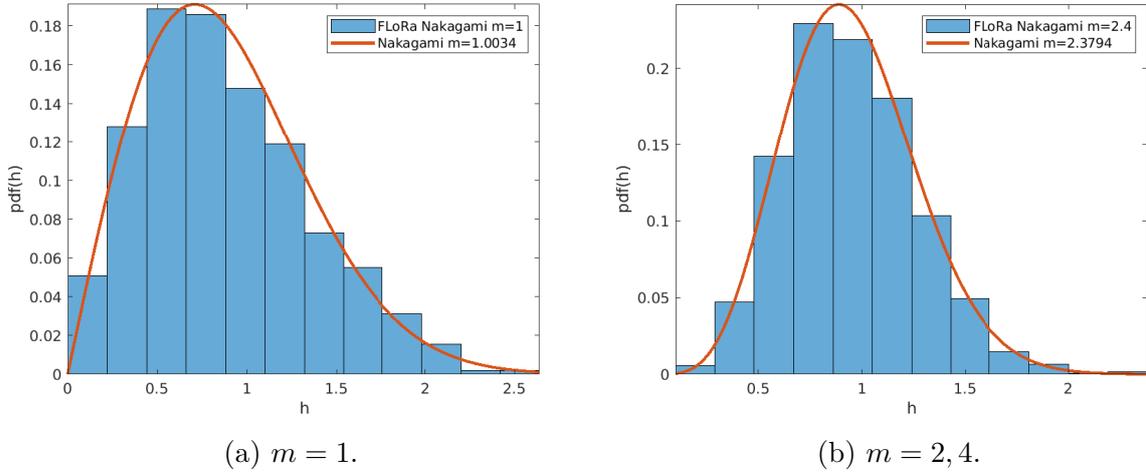
onde  $PL(d)$  é a perda de percurso para uma certa distância  $d$ ,  $\overline{PL}(d_0)$  a perda de percurso para uma distância de referência  $d_0$  e  $\eta$  um expoente obtido experimentalmente.  $X_\sigma$  é uma variável aleatória de média  $\mu = 0$  e desvio padrão  $\sigma_x$  também obtido experimentalmente (GOLDSMITH, 2005).

A potência recebida ( $P_r$ ) é obtida pela multiplicação da potência de transmissão, no domínio linear, com o produto dos ganhos das antenas, dividido pelas perdas, de acordo com

$$P_r = P_t \frac{G_t G_r}{PL(d)L}, \quad (4)$$

em que o termo que multiplica  $P_t$  tem valor máximo de 1.  $G_t$  e  $G_r$  são os ganhos nas antenas de transmissão e recepção, respectivamente, e  $L$  representa perdas provenientes de outros fatores.

Caso a potência do pacote recebido seja menor do que a sensibilidade definida em (SEMTECH, 2015), este é contabilizado como estando fora de alcance, sendo ignorado no receptor. Assim, se um sinal que poderia causar uma colisão for recebido abaixo da sensibilidade, este não tem efeito algum. Para modelar o ruído adicionado ao sinal, são definidas potências de ruído de acordo com o SF e BW utilizados na transmissão.

Figura 6 – Validação da distribuição Nakagami- $m$ .

Fonte – Do autor.

O simulador foi estendido para adicionar um modelo estatístico de perdas de potência por desvanecimento, onde é considerada uma variação instantânea de potência em cada transmissão, de forma que  $P_r = h^2 \overline{P_r}$  (GOLDSMITH, 2005). O cálculo da potência média recebida  $\overline{P_r}$  é similar ao de (4) com (3), exceto pela variável aleatória  $X_\sigma$ , que é desconsiderada. O valor esperado de  $h^2$ ,  $E[h^2]$ , é 1, enquanto que  $h$  é modelado pela distribuição de probabilidade Nakagami- $m$ , conforme

$$p(h) = \frac{2m^m h^{2m-1} e^{-mh^2}}{\Gamma(m)}, \quad (5)$$

onde  $\Gamma(\cdot)$  é a função Gamma completa. O parâmetro de desvanecimento  $m$  é alterado para modelar os diferentes cenários, podendo variar de  $0,5 \leq m \leq 1$  para situações sem linha de visada, a valores mais altos de  $m$ , indicando maior existência de uma parcela de linha de visada, sendo que quando  $m \rightarrow \infty$  tem-se equivalência com o canal adicionado de ruído gaussiano branco (GOLDSMITH, 2005).

Para validar a implementação do modelo Nakagami- $m$  foi feita uma simulação com apenas um GW e um ED a uma distância fixa. A sensibilidade de recepção foi alterada para  $-180$  dBm (apenas para que os pacotes com potência de transmissão abaixo da sensibilidade pudessem ser registrados). A partir do vetor de potência recebida, obteve-se um fator  $m = 1,0034$  quando o valor definido para a simulação era  $m = 1$  e  $m = 2,3794$  quando definido em  $m = 2, 4$ , mostrados na Figura 6, que são resultados satisfatórios.

### 3.1.4 Modelo de Consumo de Energia

O modelo original para o consumo de energia no FLoRa é baseado em uma máquina de estados finita, sendo o consumo de energia dependente do tempo em que um rádio LoRa passa em determinado estado. São modelados 3 estados: `idle`, quando o rádio não

transmite nem recebe; **transmit**, quando o rádio está transmitindo; e **receiving**, quando o rádio está recebendo um *downlink* ou “escutando”. Para estes estados, são definidas correntes de alimentação, tendo o estado **transmit** correntes de alimentação específicas que aumentam com cada potência de transmissão utilizada. Sempre que uma mudança de estados ocorre em um ED, o consumo de energia daquele ED no estado anterior é calculado (tensão e corrente de alimentação, e tempo em que o ED permaneceu no estado anterior são multiplicados) e somado ao consumo total (SLABICKI; PREMSANKAR; DI FRANCESCO, 2018).

O modelo de consumo energético modelado originalmente no simulador é um tanto simplista e desconsidera alguns estados secundários pelos quais o rádio do dispositivo passa, como por exemplo o processamento dos dados na transmissão e na recepção. Por causa disso, um modelo mais detalhado, baseado em Casals *et al.* (2017) foi implementado. Neste modelo, o consumo energético é calculado após o fim das simulações a partir dos vetores de SF,  $P_t$  e número de *downlinks* recebidos. Uma aproximação razoável foi feita, em que os dispositivos só abrem a primeira janela de recepção. Os valores das correntes são baseados em Casals *et al.* (2017) e também em Petäjäljärvi *et al.* (2017).

O consumo total de energia  $E_t$  de um dispositivo é calculado ao final das simulações de acordo com

$$E_t = V_{CC} \sum_{i=1}^{N_t} \left( I_{P_i} t_{SF_i} + I_r t_{r_i} + \sum_S I_S t_S \right), \quad (6)$$

onde  $V_{CC}$  é a tensão de alimentação do rádio,  $I_{P_i}$  e  $t_{SF_i}$  são a corrente de alimentação para a  $P_t$  e o tempo de transmissão para o SF utilizados na  $i$ -ésima transmissão;  $I_r$  é a corrente de alimentação quando o rádio está em modo de recepção,  $t_{r_i}$  é o tempo de duração da recepção (5 períodos de símbolo de acordo com o SF utilizado quando não há *downlink*, ou o tempo de duração do *downlink*, caso contrário);  $I_S$  e  $t_S$  são a corrente de alimentação e o tempo de duração de cada estado  $S$  do rádio identificado em Casals *et al.* (2017), fixos independentemente dos parâmetros de transmissão; e  $N_t$  é o número total de transmissões realizadas.

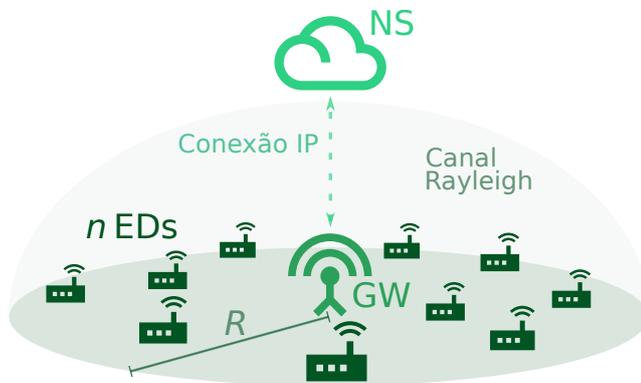
## 3.2 CENÁRIOS DE SIMULAÇÃO

São consideradas três situações principais: as redes com parâmetros de transmissão aleatórios, utilizando ADR-TTN e utilizando ADR+. Para avaliá-las, é considerada uma aplicação com  $n$  dispositivos Classe A dispostos em uma região circular de raio  $R$ , com um único GW LoRaWAN posicionado no centro desta região. O GW é conectado ao NS por uma conexão IP padrão. Todos os ED estão sujeitos ao mesmo *duty-cycle*, e começam a transmitir após períodos determinados por distribuições exponenciais de média  $1/\lambda_0$ . Além do período de *duty-cycle*, os ED determinam o momento de suas transmissões seguintes a partir de distribuições exponenciais com média  $1/\lambda_1$ . Todos os dispositivos rodam a mesma aplicação, com pacotes com preâmbulos de  $p_a$  símbolos e *payloads* de  $p_l$  bytes.

Quando nenhum ADR é utilizado, cada dispositivo transmite utilizando apenas uma combinação de SF e  $P_t$  aleatória, uniformemente distribuída entre os nós. Por outro lado, quando qualquer um dos ADR é utilizado, todos os dispositivos começam transmitindo com os valores máximos, ou seja, SF12 e  $P_{t_{max}}$ . Desta forma, os algoritmos são inteiramente responsáveis pela seleção de parâmetros de transmissão.

Considerando um cenário pessimista, todos os nós disputam um único canal de frequência, sendo o modelo de perda de propagação utilizado nas simulações é o Nakagami- $m$  com  $m = 1$ , equivalente ao canal Rayleigh (GOLDSMITH, 2005). Uma ilustração do cenário é apresentada na Figura 7.

Figura 7 – Ilustração do modelo do sistema, com ED, GW e NS.



Fonte – Do autor.



## 4 RESULTADOS DE SIMULAÇÃO

Neste Capítulo são apresentados os resultados iniciais de simulação no simulador FLoRa. O modelo de atenuação utiliza dados de medições reais obtidos por Petäjäljärvi *et al.* (2016). Na Tabela 2 os parâmetros de perda de propagação e da aplicação são apresentados, seguindo as notações apresentadas nos capítulos anteriores. Para cada cenário foram executadas dez simulações independentes. Cada simulação dura 20 dias simulados, mas os dados só são coletados após o décimo dia, para que o sistema possa passar por um período de “aquecimento” e os algoritmos sejam avaliados pós-convergência. Nos gráficos de linha, as barras indicam os intervalos de confiança de 95%.

Tabela 2 – Parâmetros de aplicação e perda de propagação.

Parâmetro	Valor
$n$	200
$R$	1500, 2500, 3500, 4500, 5500 m
<i>Duty-cycle</i>	0.1% referido ao SF12
CR	4/8
BW	125 kHz
$1/\lambda_0$	100 s
$1/\lambda_1$	1200 s
$p_a$	8 símbolos
$p_l$	20 bytes
$\overline{PL}(d_0)$	128.95 dB
$\eta$	2.32
$d_0$	1000 m

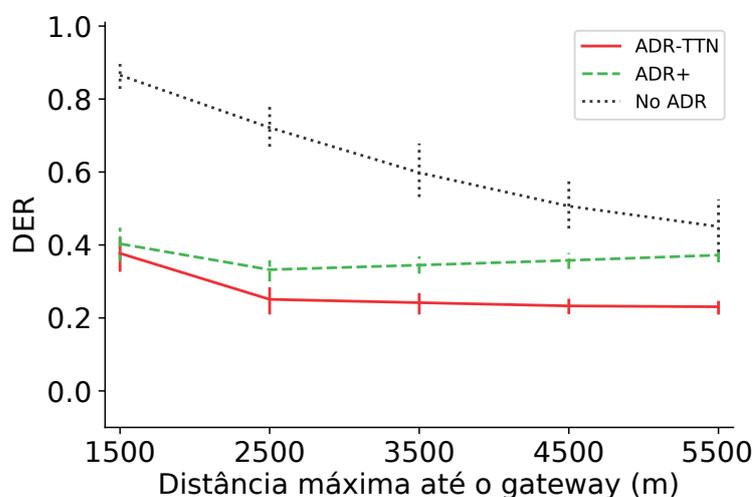
Fonte – Do autor.

### 4.1 SIMULAÇÕES COM MARGEM PADRÃO

Inicialmente, o parâmetro *margin\_db* é selecionado em 10 dB, um valor típico de acordo com (SEMTECH, 2016). A Figura 8 mostra a DER média das redes quando  $R$  varia de 1500 a 5500 m, a passos de 1000 m, considerando os algoritmos ADR-TTN, ADR+ e o caso sem ADR (No ADR). Na configuração “No ADR”, tanto SF quanto  $P_t$  são distribuídos uniformemente entre os nós.

Em distâncias mais curtas, nota-se que os algoritmos ADR-TTN e ADR+ têm um desempenho similar, ambos tendo um desempenho inferior ao caso sem ADR. É interessante notar que a DER no caso com ADR+ acaba tendo um ganho de desempenho em distâncias maiores, já que a abordagem menos agressiva deste algoritmo permite que dispositivos utilizem SF mais altos (o que não é o caso para o ADR-TTN) sem que haja um aumento expressivo do número de colisões, já que no cenário considerado poucos nós

Figura 8 – DER média utilizando  $margin\_db = 10$  dB, variando distância máxima dos EDs até o GW.

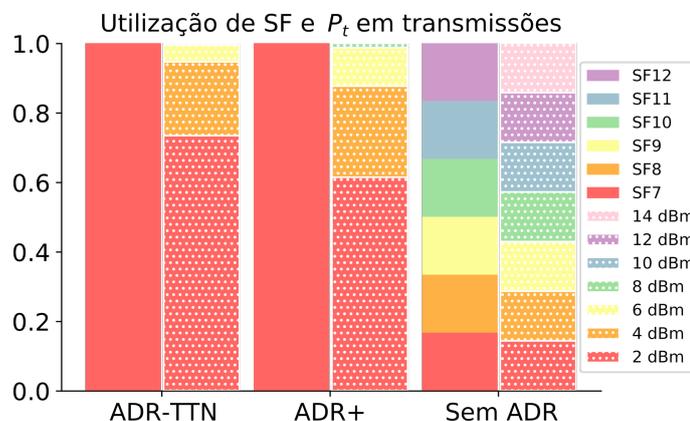


Fonte – Do autor.

disputam o canal. Apesar disso, o desempenho final é insuficiente para garantir um serviço confiável.

O motivo de o ADR-TTN e o ADR+ apresentarem uma DER inferior ao caso sem ADR é a alocação agressiva de SF e  $P_t$  feita por estes algoritmos. Como indica a Figura 9, num cenário com  $R = 1500$  m, praticamente todos os nós transmitem utilizando SF7 e os menores valores de  $P_t$  disponíveis. A escolha destes parâmetros por estes algoritmos é esperada porque ambos, mas especialmente o ADR-TTN, foram projetados com o objetivo de minimizar a utilização do canal e aumentar o tempo de vida da bateria.

Figura 9 – Utilização de SF e  $P_t$  quando  $margin\_db = 10$  dB para o ADR-TTN, ADR+ e sem ADR, com  $R = 1500$  m



Fonte – Do autor.

Tabela 3 – Consumo energético quando  $margin\_db = 10$  dB e  $R = 1500$  m.

Algoritmo	LI (J)	Q1 (J)	Mediana (J)	Q3 (J)	LS (J)	DER (%)
ADR-TTN	57,07	60,17	61,24	62,28	65,39	37,70
ADR+	56,89	60,17	61,27	62,39	65,61	40,32
Sem ADR	57,06	66,16	78,82	120,01	200,71	86,53

Fonte – Do autor.

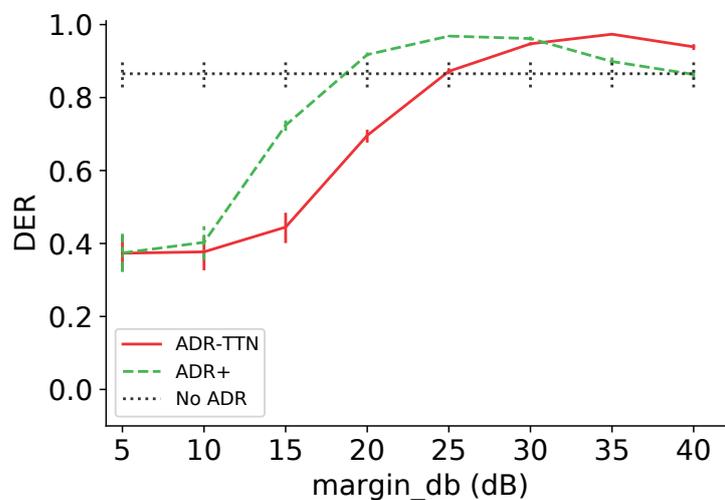
De fato, utilizar os menores SF e  $P_t$  resultam num consumo de energia significativamente menor pelos nós, como indica a Tabela 3. Nela estão quantizados os elementos de um diagrama do tipo *box-plot*, que ilustra uma distribuição de valores, apresentando o limite inferior (LI), o primeiro quartil (Q1), a mediana, o terceiro quartil (Q3) e o limite superior (LS), além da DER atingida em cada caso. No entanto, o consumo menor de energia não necessariamente é traduzido em eficiência energética, já que a DER nestes casos é substancialmente inferior. Apesar disso, nota-se que o ADR+ tem um desempenho minimamente superior ao caso do ADR-TTN, sem nenhuma alteração significativa na distribuição do consumo energético. Como o ADR+ adota uma abordagem mais conservadora, o algoritmo tem uma margem de SNR extra em seus cálculos, comparado ao ADR-TTN. Essa margem extra é a motivação para modificar o parâmetro  $margin\_db$  nos algoritmos, visando uma melhoria na DER e mantendo um consumo baixo de energia.

## 4.2 SIMULAÇÕES COM MARGEM VARIÁVEL

A variável  $margin\_db$  no ADR impacta na alocação de parâmetros de transmissão. Na Figura 10 é apresentada a variação da DER média nos ED ao variar o valor de  $margin\_db$  (a mesma para cada ED no mesmo cenário) entre 5 e 40 dB, num cenário com  $R = 1500$  m. Ao otimizar este parâmetro, é possível atingir uma DER superior ao caso sem ADR, enquanto o consumo energético é mantido baixo. Por exemplo, com uma  $margin\_db$  ótima que garante uma DER de 90%, a distribuição de consumo de energia dos nós é similar à apresentada anteriormente, com exceção de poucos *outliers*.

Comparado ao ADR-TTN, o ADR+ alcança uma DER alvo de 90% com um valor menor de  $margin\_db$ . Como o ADR+ considera, naturalmente, valores menores de SNR para suas estimativas, a margem adicional necessária pelo algoritmo para estimar as condições do canal de forma confiável é menor. Além disso, apesar de a escolha de valores arbitrariamente altos para a  $margin\_db$  para o algoritmo terem chance de causar bons resultados de DER, esta estratégia pode ter um resultado indesejado, pois os nós terão mais dificuldade em utilizar valores menores de SF, aumentando o consumo energético e o número de colisões. Este efeito fica claro para valores de  $margin\_db$  maiores que 25 dB para ADR+ e 35 dB para o ADR-TTN.

A escolha ideal do parâmetro  $margin\_db$  é fundamental para o bom desempenho de

Figura 10 – Impacto de  $margin\_db$  na DER média.

Fonte – Do autor.

uma rede LoRaWAN, mas é necessário que essa escolha seja feita tendo um conhecimento prévio da rede, pois configurações diferentes teriam resultados diferentes. Para que essa seleção seja feita de forma automática, neste trabalho é proposta uma modificação ao ADR-TTN, baseada no ADR+, que é apresentada no próximo Capítulo.

## 5 PROPOSTA DO ADRx

Após uma longa campanha de simulações com os algoritmos ADR-TTN e ADR+, foi percebido que o parâmetro  $margin\_db$  pode ser selecionado de forma a atingir um nível de desempenho desejado, uma vez que este parâmetro compensa pela imprecisão das estimativas de qualidade do canal. Se um ED é incapaz de entregar um certo número de transmissões devido a desconexões, o NS pode utilizar valores de SNR que podem ser viesados, já que transmissões com valores de RSSI menores do que a sensibilidade do rádio não serão recebidas e não serão consideradas na estimativa. É possível, mesmo com falhas de recepção, estimar com precisão a distribuição de RSSI de um conjunto de medidas viesado através de ajuste de curva (KIELGAST *et al.*, 2017), podendo ser feita uma escolha adequada de  $margin\_db$  para o ED, mas isso pode aumentar muito a complexidade do sistema, além de ser dependente do modelo selecionado (i.e., o modelo estatístico do canal deveria ser conhecido a priori, o que nem sempre é o caso). Por outro lado, escolher o mesmo valor de  $margin\_db$  para todos os dispositivos pode resultar em parâmetros de transmissão muito elevados para dispositivos próximos aos GW. Para resolver ambos os problemas, é proposto o ADRx.

### 5.1 DESCRIÇÃO DO ALGORITMO

Ao invés de selecionar manualmente o valor de  $margin\_db$  para cada dispositivo durante a instalação, no ADRx todos os dispositivos começam com o mesmo valor de  $margin\_db$  e o adaptam toda vez que o NS calcula o ADR (i.e., a cada  $N$  *uplinks* recebidos com sucesso). Uma DER de referência,  $DER_{ref}$  é definida e, se a DER alcançada entre cálculos de ADR ( $DER_{inst}$ ) for inferior a  $DER_{ref}$ , então uma nova margem é calculada somando 5 dB à margem atual, com um limite máximo de 30 dB. Como a  $DER_{inst}$  não representa fielmente a DER de um dispositivo, é considerada uma margem de segurança antes de diminuir o valor de  $margin\_db$  se aparentar ser necessário. Então, para diminuir  $margin\_db$  e evitar que os dispositivos saturem em valores de  $margin\_db$  muito altos, mas ao mesmo tempo garantir um desempenho satisfatório, se a  $DER_{inst}$  for maior do que  $1.15 \times DER_{ref}$ , ADRx subtrai 2,5 dB da margem atual. O limite inferior para a  $margin\_db$  é definido em 5 dB, e o resto do algoritmo é idêntico ao ADR+.

O Algoritmo 1 detalha em pseudocódigo a parte do ADR específica ao ADRx. Este algoritmo tem como entradas a  $margin\_db$  atual do dispositivo e o número do último pacote recebido no NS ( $FCnt$ ). Inicialmente (linha 1), o algoritmo armazena o número do último pacote em  $FCntUp$ , na sua posição correspondente  $I$  (entre 0 e  $N$ , sendo a posição 0 se o pacote anterior a este causou a execução do ADR, e  $N$  se o pacote atual causará a execução). A variável  $I$  é incrementada na linha 2, e, se  $I$  for menor que  $N$  (linha 3), a margem atual se mantém (linha 4). Se não,  $N$  pacotes foram recebidos desde a última

**Algoritmo 1:** ADRx.

---

**Variáveis de Entrada:**  $FCnt, margin\_db$

**Inicialização:**  $I \leftarrow 0$

- 1:  $FCntUp[I] \leftarrow FCnt$
- 2:  $I = I + 1$
- 3: **if**  $I < N$  **then**
- 4:    $margin\_db \leftarrow margin\_db$
- 5: **else**
- 6:    $I = 0$
- 7:    $DER_{inst} \leftarrow N / (FCntUp[N - 1] - FCntUp[0])$
- 8:   **if**  $DER_{inst} < DER_{ref}$  **and**  $margin\_db < 30$  **then**
- 9:      $margin\_db \leftarrow margin\_db + 5$
- 10:   **else if**  $DER_{inst} > 1,15 \times DER_{ref}$  **and**  $margin\_db > 5$  **then**
- 11:      $margin\_db \leftarrow margin\_db - 2,5$
- 12:   **else**
- 13:      $margin\_db \leftarrow margin\_db$
- 14:   **end if**
- 15: **end if**

---

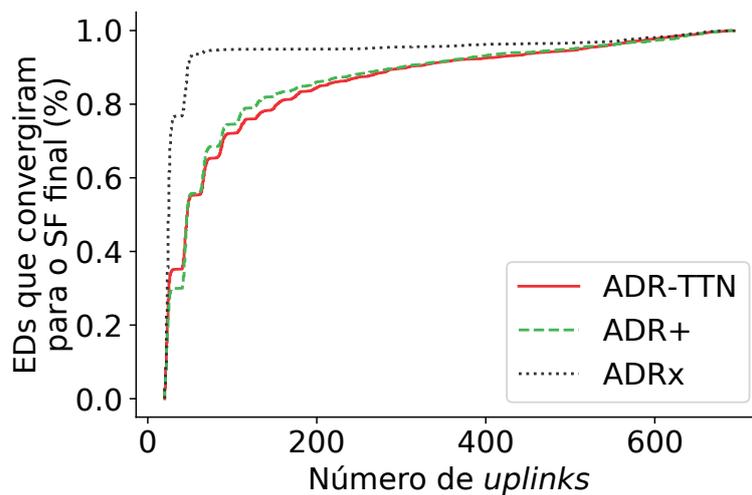
execução do ADRx, resultando em uma nova execução do algoritmo que selecionará a nova margem (e, posteriormente, os possíveis novos valores de transmissão) da seguinte forma: primeiramente,  $I$  volta ao valor de 0 (linha 6); a estimativa da  $DER_{inst}$  é feita a partir da razão do número de mensagens recebidas pela diferença entre o número do primeiro pacote recebido desde a última execução e o número do pacote mais recente (linha 7); se esta estimativa for inferior à referência estabelecida e a margem atual for inferior ao limite máximo (linha 8), então a margem é somada de 5 dB (linha 9); por outro lado, se a estimativa for 15% maior do que a referência e a margem estiver acima do limite mínimo (linha 10), então a margem atual é decrescida de 2,5 dB; ainda, se a estimativa estiver entre o valor de referência e a tolerância de 15%, a margem atual é mantida.

## 5.2 RESULTADOS DE SIMULAÇÕES COM ADRx

Os resultados das simulações na Seção 4.2 demonstram que é possível melhorar o desempenho de redes LoRaWAN que utilizam o ADR ao selecionar o parâmetro  $margin\_db$  de forma adequada. Para o mesmo cenário com  $R = 1500$  m, é selecionado o valor de  $DER_{ref} = 90\%$  para o ADRx. Para os outros algoritmos, após um grande número de simulações, os valores de  $margin\_db$  que resultavam em um desempenho médio suficientemente próximo da referência foram 26 e 19 dB para o ADR-TTN e ADR+, respectivamente. As simulações com os algoritmos foram feitas com duração de 20 dias simulados. Para determinar o período de aquecimento, foram feitas simulações para descobrir quantas transmissões, em média, cada algoritmo precisa para chegar ao seu SF definitivo. Como ilustra a Figura 11, após aproximadamente 350 *uplinks*, os três algoritmos já têm pelo

menos 80% de seus nós utilizando o SF adequado de acordo com os critérios dos algoritmos. Este número de *uplinks* significa um período de aproximadamente 10 dias simulados, como foi o período de aquecimento utilizado nas simulações anteriores. Neste caso, o ADRx tem um período de convergência menor do que os outros algoritmos, pois começa com *margin\_db* consideravelmente menor, atribuindo aos nós o valor mínimo de SF logo nas primeiras iterações, reservando mudanças posteriores aos nós mais distantes do GW, que realmente precisam utilizar outros valores de SF. Por outro lado, ADR-TTN e ADR+ iniciam com margens maiores, tendendo a manter os nós em valores maiores de SF desde o início das simulações, sendo necessário estimativas de qualidade de enlace muito boas para a escolha adequada dos parâmetros de transmissão.

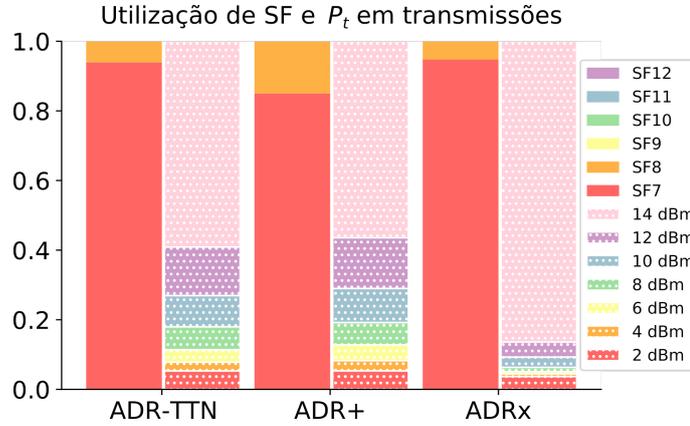
Figura 11 – Percentual de dispositivos que precisam de um dado número de *uplinks* para convergirem ao seu SF final, considerando uma  $DER_{ref}$  de 90% e  $R = 1500$  m.



Fonte – Do autor.

Na Figura 12 estão apresentadas a utilização de SF e  $P_t$  ao longo das simulações por cada algoritmo, enquanto que a Tabela 4 apresenta a distribuição do consumo energético total dos dispositivos. Todos os algoritmos têm consumo energético similar, já que a principal diferença na alocação de recursos reside na utilização de  $P_t$ , que tem menos impacto no consumo, comparado à utilização de SF. Os três algoritmos apresentam uma vantagem considerável em relação ao caso com alocação aleatória de parâmetros, tanto por terem um consumo energético médio muito inferior, quanto por assegurar uma DER superior. Além disso o ADRx é capaz de se adequar, em tempo de execução, a um nível de desempenho pré-definido sem precisar de conhecimento prévio da rede, enquanto que o ADR-TTN e ADR+ só alcançariam o mesmo nível de desempenho após uma otimização prévia.

Figura 12 – Utilização de SF e  $P_t$  considerando uma  $DER_{ref}$  de 90% e  $R = 1500$  m, com as margens otimizadas para o ADR-TTN e ADR+, e com a margem selecionada dinamicamente para cada ED no ADRx.



Fonte – Do autor.

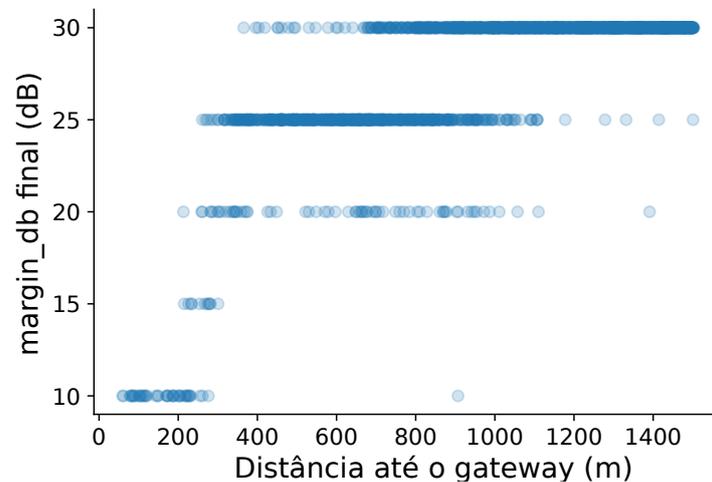
Tabela 4 – Consumo energético com as margens otimizadas para o ADR-TTN e ADR+, e com a margem selecionada dinamicamente para cada ED no ADRx, considerando uma  $DER_{ref}$  de 90% e  $R = 1500$  m.

Algoritmo	LI (J)	Q1 (J)	Mediana (J)	Q3 (J)	LS (J)	DER (%)
ADR-TTN	57,92	61,89	63,22	64,55	68,53	89,31
ADR+	58,35	62,18	63,49	65,01	69,17	89,82
ADRx	58,61	62,22	63,38	64,64	68,22	90,61

Fonte – Do autor.

Apesar de o ADRx utilizar o valor médio das medições de SNR das  $N$  transmissões mais recentes, alguns ED terminam com valores para  $margin\_db$  diferentes de 19 dB, a margem otimizada para o ADR+, como mostra a Figura 13. Além disso, a Figura 13 também mostra que o valor de  $margin\_db$  aumenta com a distância até o GW. No entanto, uma vez que o modelo de desvanecimento é o mesmo para todos os ED da simulação, a variação estatística ao redor da média de RSSI deveria ser a mesma para todos os ED, então todos deveriam convergir para o mesmo valor de  $margin\_db$ . Como foi discutido no início deste Capítulo, as medições de RSSI são enviesadas devido à sensibilidade do rádio. Com isso, a SNR média calculada para dispositivos mais distantes do GW tende a ser mais próxima do valor máximo alcançado, pois valores abaixo da sensibilidade não são contabilizados, diferentemente do que acontece em dispositivos mais próximos. Assim, se faz necessário aumentar o valor de  $margin\_db$  com o aumento da distância até o GW. Este aumento na  $margin\_db$  garante não apenas que o ED use melhores parâmetros de transmissão, mas também que ele os mantenha mesmo que a estimativa de qualidade do enlace melhore.

Figura 13 – Gráfico de dispersão do valor final de  $margin\_db$  no ADRx, considerando uma  $DER_{ref}$  de 90% e  $R = 1500$  m.



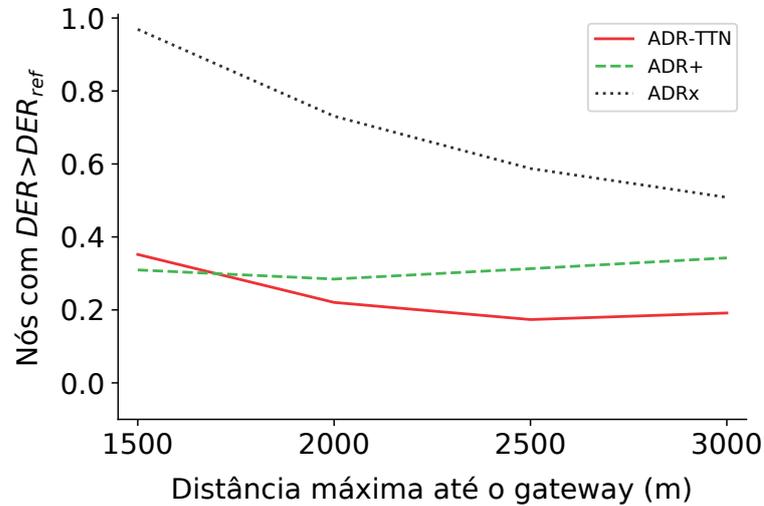
Fonte – Do autor.

Para avaliar melhor o desempenho do ADRx, também foram selecionados valores diferentes de  $DER_{ref}$ , para distâncias maiores de  $R$ . Já que a distância agora é superior a  $R = 1500$  m, um valor menos rigoroso de  $DER_{ref} = 80\%$  é selecionado. Novamente, para os outros algoritmos, os valores para  $margin\_db$  foram selecionados manualmente após várias simulações, enquanto que o ADRx inicializava cada ED com  $margin\_db = 10$  dB. Para o ADR-TTN, a margem selecionada foi de 22 dB, e, para o ADR+, 16 dB, resultando, em média, uma DER de 80% para o caso base de  $R = 1500$  m.

Na Figura 14, é apresentado um gráfico de linhas que indica o percentual de nós que terminaram as simulações com uma DER média superior à referência, quando a distância máxima até o GW varia entre 1500 e 3000 m. Para distâncias curtas como  $R = 1500$  m, tanto o ADR+ quanto o ADR-TTN falham em garantir um serviço confiável a um número razoável de ED. À medida que a distância aumenta, de forma similar aos primeiros resultados obtidos neste trabalho, o ADR+ tende a uma certa melhoria no desempenho ao permitir que os dispositivos utilizem parâmetros de transmissão maiores, mas ainda falha em garantir que a maioria dos ED tenha bom desempenho. Diferentemente, com o ADRx, quase todos os nós têm um bom desempenho na menor distância, com esse número diminuindo, naturalmente, à medida que  $R$  aumenta. Apesar disso, o algoritmo ADRx ainda supera os outros dois, tendo aproximadamente 55% dos dispositivos com DER acima da referência, enquanto que para o ADR+ esta porção é aproximadamente 40%, e para o ADR-TTN, apenas aproximadamente 20%.

Na Tabela 5 é apresentado o consumo energético e DER média ao final das simulações para o caso com ADR-TTN, ADR+ e ADRx quando  $DER_{ref} = 80\%$  e  $R = 3000$  m.

Figura 14 – Percentual de EDs com DER superior à referência de 80% utilizando os três algoritmos.



Fonte – Do autor.

Tabela 5 – Consumo energético com as margens otimizadas para o ADR-TTN e ADR+, e com a margem selecionada dinamicamente para cada ED no ADRx, considerando uma  $DER_{ref}$  de 80% e  $R = 3000$  m.

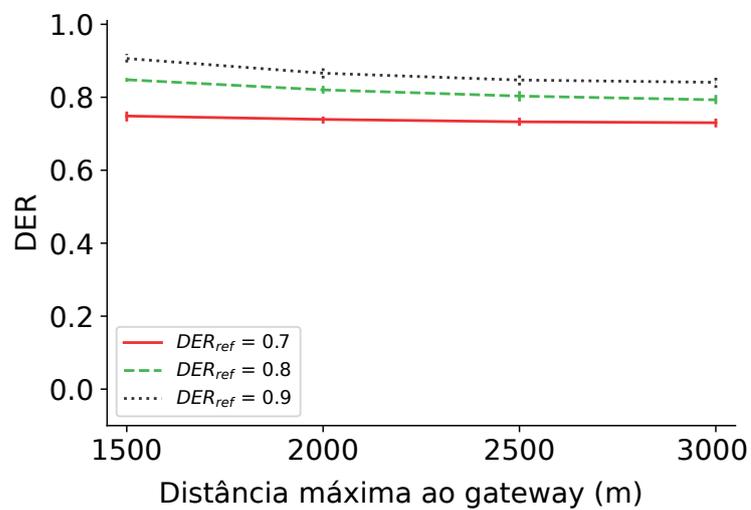
Algoritmo	LI (J)	Q1 (J)	Mediana (J)	Q3 (J)	LS (J)	DER (%)
ADR-TTN	57,35	63,32	66,50	71,28	83,10	75,61
ADR+	57,35	63,38	67,95	75,93	94,13	78,73
ADRx	57,88	63,84	68,03	74,68	90,90	79,29

Fonte – Do autor.

Naturalmente o consumo energético é superior aos casos anteriores, já que os algoritmos precisam selecionar valores maiores de SF para os dispositivos devido ao aumento de distância entre eles e o GW. ADRx e ADR+ alcançam uma DER satisfatória e têm consumo energético similar, com o ADRx superando-o levemente, enquanto que o ADR-TTN falha em alcançar a DER esperada, apesar de ter um consumo energético um pouco menor.

Por fim, são selecionados valores de  $DER_{ref}$  de 70%, 80% e 90%, variando a distância da mesma forma que foi feita no experimento anterior. A DER média dos dispositivos é apresentada na Figura 15. Para referências menos exigentes, o ADRx é capaz de garantir DER consistentes até para distâncias maiores. Até mesmo quando considerada uma referência mais restritiva de 90%, o desempenho médio da rede é de 84,72% e 84,58% para os casos com  $R = 2500$  e  $3000$  m, respectivamente, indicando que o ADRx é robusto à variação de distância, desde que a DER desejada seja viável.

Figura 15 – DER média dos EDs variando  $R$  para diferentes valores de  $DER_{ref}$ , utilizando o ADRx.



Fonte – Do autor.



## 6 CONCLUSÃO

Neste trabalho a principal implementação para o ADR em redes LoRaWAN (ADR-TTN) e uma proposta de alteração (ADR+) foram estudados, e foi feita uma investigação do impacto de um parâmetro normalmente negligenciado nestes algoritmos, a *margin\_db*. Este parâmetro serve como uma margem de segurança na estimativa de qualidade do enlace que o algoritmo realiza. Foi mostrado que os valores tipicamente escolhidos para *margin\_db* não são adequados em situações em que o meio tem muitas perdas, pois esta margem não compensaria as estimativas otimistas, fazendo com que o ADR-TTN (e, em menor grau, o ADR+) selecionasse parâmetros de transmissão incapazes de garantir comunicação confiável na maioria dos casos. Ao selecionar adequadamente o valor de *margin\_db*, é possível melhorar consideravelmente o desempenho dos dispositivos, ao mesmo tempo mantendo um consumo energético baixo, garantindo um tempo de vida maior para as baterias. No entanto, esta otimização pode não ser possível em diversas aplicações, pois ela demanda um conhecimento prévio das configurações da rede para alcançar o desempenho desejado.

Para resolver esse problema, é proposto o ADRx, no qual o parâmetro *margin\_db* é selecionado em tempo de execução pelo NS, individualmente para cada ED, considerando o número de pacotes perdidos pelo GW entre execuções do ADR. Através de experimentos utilizando o simulador FLoRa, foi verificado que o ADRx alcança um desempenho tão bom quanto o que o ADR-TTN e o ADR+ alcançam após uma otimização prévia do parâmetro *margin\_db*, indicando que o algoritmo proposto é mais vantajoso por ser capaz de selecionar este parâmetro em tempo de execução.

Em trabalhos futuros, é possível avaliar o funcionamento do ADRx em outros cenários que não foram abordados neste trabalho. Por exemplo, considerando modelos com desvanecimento menos severos, seria interessante identificar se, ou em que ponto, a comparação de desempenho entre os algoritmos sofre alguma alteração qualitativa. Além disso, cenários com múltiplos GW e com a inclusão de replicações de mensagens podem ser estudados.



## REFERÊNCIAS

ABDELFADEEL, Khaled; CIONCA, Victor; PESCH, Dirk. Fair Adaptive Data Rate Allocation and Power Control in LoRaWAN. *In: 2018 IEEE 19th International Symposium on "A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)*. [S.l.: s.n.], jun. 2018. P. 14–15. DOI: 10.1109/WoWMoM.2018.8449737.

BOR, Martin; ROEDIG, Utz; VOIGT, Thiemo; ALONSO, Juan M. Do LoRa Low-Power Wide-Area Networks Scale? *In: PROCEEDINGS of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*. Malta: [s.n.], 2016. P. 59–67. ISBN 978-1-4503-4502-6. DOI: 10.1145/2988287.2989163.

CASALS, Lluís; MIR, Bernat; VIDAL, Rafael; GOMEZ, Carles. Modeling the Energy Performance of LoRaWAN. **Sensors**, v. 17, n. 10, 2017. ISSN 1424-8220. DOI: 10.3390/s17102364.

CUOMO, Francesca; CAMPO, Manuel; CAPONI, Alberto; BIANCHI, Giuseppe; ROSSINI, Giampaolo; PISANI, Patrizio. 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)*. [S.l.: s.n.], 2017. P. 1–8.

FINNEGAN, Joseph; FARRELL, Ronan; BROWN, Stephen. Analysis and Enhancement of the LoRaWAN Adaptive Data Rate Scheme. **IEEE Internet of Things Journal**, v. 7, n. 8, p. 7171–7180, 2020.

GARG, Priyanka. **Performance Evaluation of a LoRaWAN Towards Development of an Optimised ADR (Adaptive Data Rate) Model**. 2019. Diss. (Mestrado) – Delft University of Technology, the Netherlands.

GOLDSMITH, Andrea. **Wireless Communications**. New York, USA: Cambridge University Press, 2005. ISBN 0521837162.

KIELGAST, Mathias; RASMUSSEN, Anders; LAURSEN, Mathias; NIELSEN, Jimmy; POPOVSKI, Petar; KRIGSLUND, Rasmus. Estimation of Received Signal Strength Distribution for Smart Meters With Biased Measurement Data Set. **IEEE Wireless Communications Letters**, v. 6, n. 1, p. 2–5, 2017.

KIM, Sungryul; YOO, Younghwan. Contention-Aware Adaptive Data Rate for Throughput Optimization in LoRaWAN. **Sensors**, MDPI AG, v. 18, n. 6, p. 1716, mai. 2018. ISSN 1424-8220. DOI: 10.3390/s18061716.

LI, Shengyang; RAZA, Usman; KHAN, Aftab. How Agile is the Adaptive Data Rate Mechanism of LoRaWAN? *In*: 2018 IEEE Global Communications Conference (GLOBECOM). [*S.l.*: *s.n.*], 2018. P. 206–212. DOI: 10.1109/GLOCOM.2018.8647469.

LORA ALLIANCE. **LoRaWAN 1.0.3 Specifications**. [*S.l.*], jul. 2018a.

LORA ALLIANCE. **RP002-1.0.0 LoRaWAN Regional Parameters**. [*S.l.*], 2018b.

MARAIS, Jaco Morné; MALEKIAN, Reza; ABU-MAHFOUZ, Adnan M. Evaluating the LoRaWAN Protocol Using a Permanent Outdoor Testbed. **IEEE Sensors Journal**, v. 19, n. 12, p. 4726–4733, 2019.

ORANGE CONNECTED OBJECTS & SOLUTIONS. **LoRa Device Developer Guide**. [*S.l.*], abr. 2016.

PARK, Junhyun; PARK, Kunho; BAE, Hyeongho; KIM, Chong-Kwon. EARN: Enhanced ADR With Coding Rate Adaptation in LoRaWAN. **IEEE Internet of Things Journal**, v. 7, n. 12, p. 11873–11883, 2020. DOI: 10.1109/JIOT.2020.3005881.

PETÄJÄJÄRVI, Juha; MIKHAYLOV, Konstantin; HÄMÄLÄ, Matti; IINATTI, Jari. Evaluation of LoRa LPWAN technology for remote health and wellbeing monitoring. *In*: 2016 10th International Symposium on Medical Information and Communication Technology (ISMICT). [*S.l.*: *s.n.*], mar. 2016. P. 1–5. DOI: 10.1109/ISMICT.2016.7498898.

PETÄJÄJÄRVI, Juha; MIKHAYLOV, Konstantin; YASMIN, Rumana; HÄMÄLÄINEN, Matti; IINATTI, Jari. Evaluation of LoRa LPWAN Technology for Indoor Remote Health and Wellbeing Monitoring. **International Journal of Wireless Information Networks**, Springer Science e Business Media LLC, v. 24, n. 2, p. 153–165, fev. 2017. DOI: 10.1007/s10776-017-0341-8. Disponível em: <https://doi.org/10.1007/s10776-017-0341-8>.

RAZA, Usman; KULKARNI, Parag; SOORIYABANDARA, Mahesh. Low Power Wide Area Networks: An Overview. **IEEE Communications Surveys and Tutorials**, v. 19, n. 2, p. 855–873, abr. 2017. ISSN 1553-877X. DOI: 10.1109/COMST.2017.2652320.