



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE AUTOMAÇÃO E SISTEMAS
CURSO DE GRADUAÇÃO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Valdecir Hoffmann

**Proposta de melhoria para o algoritmo ADR do protocolo LoRaWAN voltado
para aplicações agrícolas de IoT**

Florianópolis
2023

Valdecir Hoffmann

**Proposta de melhoria para o algoritmo ADR do protocolo LoRaWAN voltado
para aplicações agrícolas de IoT**

Relatório final da disciplina DAS5511 (Projeto de Fim de Curso) como Trabalho de Conclusão do Curso de Graduação em Engenharia de Controle e Automação da Universidade Federal de Santa Catarina em Florianópolis.

Orientador: Prof. Carlos Barros Montez, Dr.
Supervisor: Gustavo Gonçalves Ribeiro, Eng.

Florianópolis
2023

Ficha de identificação da obra

A ficha de identificação é elaborada pelo próprio autor.

Orientações em:

<http://portalbu.ufsc.br/ficha>

Valdecir Hoffmann

Proposta de melhoria para o algoritmo ADR do protocolo LoRaWAN voltado para aplicações agrícolas de IoT

Esta monografia foi julgada no contexto da disciplina DAS5511 (Projeto de Fim de Curso) e aprovada em sua forma final pelo Curso de Graduação em Engenharia de Controle e Automação

Florianópolis, 30 de novembro de 2023.

Prof. Marcelo De Lellis Costa de Oliveira, Dr.
Coordenador do Curso

Banca Examinadora:

Prof. Carlos Barros Montez, Dr.
Orientador
UFSC/CTC/DAS

Gustavo Gonçalves Ribeiro, Eng.
Supervisor
Fundação CERTI

Prof. Gabriel Thaler, Me.
Avaliador
UFSC/CTC/DAS

Prof. Eduardo Camponogara, Dr.
Presidente da Banca
UFSC/CTC/DAS

Este trabalho é dedicado à sociedade e aos meus
queridos pais.

AGRADECIMENTOS

Agradeço a Deus em primeiro lugar, pela resiliência e sabedoria e por me acompanhar ao longo das atividades desta jornada. Aos meus pais, Faldi e Valdemar por se esforçarem ao máximo para me proporcionar uma boa educação e serem exemplos de garra, determinação e caráter. Às minhas irmãs que sempre estiveram ao meu lado, para fornecer conselhos e palavras de encorajamento nos momentos difíceis. Aos meus amigos Cristian, Guilherme e Juliano, que sempre estiveram próximos para dividir os desafios e alegrias proporcionados pela graduação. Ao meu orientador Carlos Barros Montez, por dispor do seu tempo e conhecimento, para me guiar no desenvolvimento deste trabalho. Ao meu supervisor Gustavo Gonçalves Ribeiro, por dispor de seu conhecimento e tempo para me auxiliar no desenvolvimento de minhas atividades. À Fundação CERTI, pela oportunidade de integrar a instituição, contribuindo de forma significativa com minha formação pessoal e profissional.

DECLARAÇÃO DE PUBLICIDADE

Florianópolis, 30 de Novembro de 2023.

Na condição de representante da Fundação CERTI na qual o presente trabalho foi realizado, declaro não haver ressalvas quanto ao aspecto de sigilo ou propriedade intelectual sobre as informações contidas neste documento, que impeçam a sua publicação por parte da Universidade Federal de Santa Catarina (UFSC) para acesso pelo público em geral, incluindo a sua disponibilização *online* no Repositório Institucional da Biblioteca Universitária da UFSC. Além disso, declaro ciência de que <o/a> autor<a>, na condição de estudante da UFSC, é obrigado a depositar este documento, por se tratar de um Trabalho de Conclusão de Curso, no referido Repositório Institucional, em atendimento à Resolução Normativa n° 126/2019/CUn.

Por estar de acordo com esses termos, subscrevo-me abaixo.

Gustavo Gonçalves Ribeiro
Fundação CERTI

RESUMO

Este trabalho apresenta o estudo e proposta de melhoria para o algoritmo de adaptação da taxa de dados (*Adaptive Data Rate* - ADR) do protocolo de comunicação de amplo alcance (*Long Range Wide Area Network* - LoRaWAN). Este protocolo é aplicado sobre a tecnologia de comunicação sem fio, de longo alcance (*Long Range-LoRa*), projetada para dispositivos de Internet das Coisas (*Internet of Things* - IoT), particularmente nos casos de baixa taxa de transferência de dados, longo alcance e baixíssimo consumo de energia. Estudos na área apontam que o algoritmo ADR apresenta algumas ineficiências, na definição dos parâmetros de transferência de dados, utilizados pela rede. Estes estudos propõem soluções mais adaptadas e capazes de lidar com um conjunto maior de parâmetros da rede, oferecendo ganhos significativos em desempenho e economia de energia. A utilização de LoRaWAN está associada principalmente a estruturas de sensoriamento e monitoramento remoto, além de controle de sistemas tolerantes à latência de sinal, sendo altamente escalável e com garantias de segurança, através de criptografia nos dados transmitidos. A Fundação CERTI, instituição que está empenhada em oferecer soluções tecnológicas para seus clientes e que já possui bom histórico de atuação na área de tecnologia da comunicação, conhecendo o potencial e as vantagens desta tecnologia, optou por aprofundar sua base de conhecimento sobre a mesma. Vislumbrou-se o potencial da tecnologia para aplicações de agricultura inteligente, com o objetivo de ampliar a maturidade e domínio sobre a mesma, para exploração em projetos futuros. A proposta deste trabalho é apresentar uma melhoria para o algoritmo ADR visando melhorar o desempenho da rede em aplicações voltadas à agricultura, onde os dispositivos sensoriais apresentam-se a distâncias consideráveis do gateway de recepção de dados. Foram elaboradas simulações utilizando um ambiente de simulação de redes (*Objective Modular Network Testbed in C++* - OMNeT++) com uma extensão denominada FLora, voltada para a tecnologia LoRa. Com simulações voltadas para o cenário de aplicação, foram levantados indicadores para avaliar o desempenho do algoritmo convencional e propor melhorias, que foram implementadas em novas simulações e mostraram resultados satisfatórios no desempenho da rede.

Palavras-chave: LoRa. LoRaWAN. Algoritmo ADR.

ABSTRACT

This paper presents a study and proposed improvement to the Adaptive Data Rate (ADR) algorithm of the Long Range Wide Area Network (LoRaWAN) communication protocol. This protocol is applied to the Long Range (LoRa) wireless communication technology designed for Internet of Things (IoT) devices, particularly in cases of low data transfer rate, long range and very low energy consumption. Studies in the field show that the ADR algorithm has some inefficiencies when it comes to defining the data transfer parameters used by the network. These studies propose more adapted solutions capable of dealing with a larger set of network parameters, offering significant gains in performance and energy savings. The use of LoRaWAN is mainly associated with remote sensing and monitoring structures, as well as control systems that are tolerant of signal latency, being highly scalable and with security guarantees, through encryption of the transmitted data. The CERTI Foundation, an institution that is committed to offering technological solutions to its clients and which already has a good track record in the area of communication technology, knowing the potential and advantages of this technology, decided to deepen its knowledge base about it. The technology's potential for intelligent agriculture applications was envisioned, with the aim of increasing its maturity and mastery, for exploitation in future projects. The purpose of this work is to present an improvement to the ADR algorithm in order to improve network performance in agricultural applications, where the sensory devices are at considerable distances from the data reception gateway. Simulations were carried out using a network simulation environment (Objective Modular Network Testbed in C++ - OMNeT++) with an extension called FLora, aimed at LoRa technology. With simulations focused on the application scenario, indicators were collected to evaluate the performance of the conventional algorithm and propose improvements, which were implemented in new simulations and showed satisfactory results in network performance.

Keywords: LoRa. LoRaWAN. ADR algorithm.

LISTA DE FIGURAS

Figura 1 – LoRaWAN na agricultura.	21
Figura 2 – Demonstração do sinal de <i>chirp</i>	22
Figura 3 – Esquema de uma rede LoRaWAN.	24
Figura 4 – Faixa de frequência do padrão AU915	26
Figura 5 – Funcionamento da classe A.	29
Figura 6 – Funcionamento da classe B.	30
Figura 7 – Funcionamento da classe C.	30
Figura 8 – Fluxograma do algoritmo ADR.	33
Figura 9 – Modelagem da rede LoRa na forma gráfica.	38
Figura 10 – Resultados das simulações com ADR-TTN de 25 a 500 nós.	39
Figura 11 – Resultados das simulações com ADR-TTN em áreas de 1 km ² a 25 km ²	40
Figura 12 – Fluxograma do algoritmo ADR-Fuzzy.	43
Figura 13 – Distribuição das faixas de valores de distância e número de nós.	44
Figura 14 – Distribuição das faixas de valores de SF e TP.	44
Figura 15 – Resultados das simulações para a primeira proposta.	47
Figura 16 – Resultado das simulações para a segunda proposta.	49
Figura 17 – Resultados das simulações para o sistema Fuzzy.	51
Figura 18 – Comparação da distribuição de SF.	52

LISTA DE TABELAS

Tabela 1 – Taxa de comunicação para o padrão AU915.	26
Tabela 2 – Parâmetros de simulação.	36

LISTA DE ABREVIATURAS E SIGLAS

ABP	Modo de ativação personalizado (do Inglês Activation by Personalization)
ADR	Algoritmo de ajuste da taxa de dados (do Inglês Adaptive Data Rate)
BW	Largura de banda (do Inglês Bandwidth)
CHIRP	Pulso de Radar de Alta Intensidade Comprimido Compressed (do Inglês High Intensity Radar Pulse)
CR	Taxa de codificação (do Inglês Code Rate)
CSS	Espectro de Dispersão por Chirp (do Inglês Chirp Spread Spectrum)
dBm	Decibél-miliwatt
DR	Taxa de dados (do Inglês Data Rate)
FLoRa	Framework de simulação para LoRa
IoT	Internet das Coisas (do Inglês Internet of Things)
ISM	Faixas de frequência industriais, científicas e medicas (do Inglês Industrial Scientific and Medical)
kHz	Quilo-hertz
LoRa	Longo Alcance (do Inglês Longe Range)
LoRaWAN	Protocolo de comunicação para rede de longo alcance
LPWAN	Redes de baixa potência e longo alcance (do Inglês Low Power Wide Area Network)
mJ	Milijoule
MHz	Megahertz
N_{step}	Número de passos (do Inglês Number of steps)
OTAA	Modo de ativação através da conexão (do Inglês Over-The-Air Activation)
R_b	Taxa de dados em bits (do Inglês bit Rate)
RSSI	Métrica de medição de um sinal recebido (do Inglês Received Signal Strength Indicator)

SF	Fator de Espalhamento (do Inglês Spreading Factor)
SNR	Relação entre ruído e sinal específica de um fator de expansão de sinal (do Inglês Signal-to-Noise Ratio specific to Spreading Factor)
SNR_m	Relação entre ruído e sinal (do Inglês Signal-to-Noise Ratio)
SNR_{SF}	Relação entre ruído e sinal (do Inglês Signal-to-Noise Ratio)
ToA	Tempo de transmissão de um pacote (do Inglês Time on Air)
TP	Potência de transmissão (do Inglês Transmission Power)
TP_{max}	Máxima potência de transmissão (do Inglês Maximal Transmission Power)
TP_{min}	Mínima potência de transmissão (do Inglês Minimal Transmission Power)
TTN	Empresa de IoT (do Inglês The Things Network)

SUMÁRIO

1	INTRODUÇÃO	16
1.1	A INSTITUIÇÃO	17
1.2	MOTIVAÇÃO	17
1.3	OBJETIVOS GERAIS	18
1.4	OBJETIVOS ESPECÍFICOS	18
2	REVISÃO DA LITERATURA	20
2.1	IOT E AGRICULTURA	20
2.2	TECNOLOGIA LORA	21
2.3	PROTOCOLO DE COMUNICAÇÃO LORAWAN	23
2.4	PRINCIPAIS CARACTERÍSTICAS DAS REDES LORAWAN	23
2.5	COMPONENTES DE UMA REDE LORAWAN	24
2.5.1	<i>End-node</i>	24
2.5.2	Gateway	25
2.5.3	Servidor de rede	25
2.5.4	Parâmetros regionais de operação	25
2.5.5	Algoritmo Adaptive Data Rate	26
2.5.6	Segurança	28
2.6	CLASSES DE OPERAÇÃO	28
2.6.1	Classe A	28
2.6.2	Classe B	29
2.6.3	Classe C	30
2.7	CONFIGURAÇÃO DE OPERAÇÃO	30
2.7.1	ABP	31
2.7.2	OTAA	31
2.8	SIMULADOR OMNET++ E <i>FRAMEWORK</i> FLORA	31
3	DESENVOLVIMENTO	33
3.1	INEFICIÊNCIAS RELACIONADAS AO ADR	34
3.2	CENÁRIO RURAL DE APLICAÇÃO	35
3.3	ELABORAÇÃO DO MODELO DE SIMULAÇÃO	35
3.4	PROPOSTAS DE MELHORIA PARA O ALGORITMO ADR	40
3.4.1	Proposta 1	41
3.4.2	Proposta 2	41
3.4.3	Proposta 3	42
4	RESULTADOS	46
4.1	PROPOSTA 1	46
4.2	PROPOSTA 2	48
4.3	PROPOSTA 3	49

5	CONSIDERAÇÕES FINAIS	53
	REFERÊNCIAS	54
	ANEXO A – SIMULAÇÃO COM 10 NÓS	57
	ANEXO B – ALGORITMO DO SISTEMA FUZZY	60

1 INTRODUÇÃO

O aumento significativo no uso de dispositivos IoT tem motivado desenvolvedores e especialistas em sistemas a colaborarem para criar e aprimorar infraestruturas de rede e protocolos de comunicação, capazes de atender às crescentes demandas do mercado. Dentro desse contexto, duas características são desejáveis, dispositivos com consumo de energia reduzido e com conectividade de longo alcance. Assim surge o conceito das redes de longo alcance e baixa potência (LPWAN), criadas para explorar o desenvolvimento destas tecnologias. Uma tecnologia LPWAN que vem se destacando e ganhando muito espaço é o LoRaWAN (ALLIANCE, 2021), um protocolo de rede aberto destinado a tecnologia LoRa. Os elementos que compõem uma rede LoRaWAN são basicamente o dispositivo final (*end-node* ou nó), o *gateway* e o servidor de rede, que realiza grande parte do processamento computacional da rede, fazendo a gestão de seus recursos.

O protocolo LoRaWAN é destinado a tecnologia de modulação de sinal LoRa (SEMTECH, 2023) e, proprietária da empresa Semtech. A base de funcionamento da tecnologia consiste no espalhamento espectral através de *chirps*, realizando a demodulação de mensagens com baixa razão sinal-ruído (Signal-to-Noise Ratio - SNR), mas também com baixa taxa de dados. Na tecnologia LoRa, a taxa de dados é ajustada através da combinação de largura de banda (*bandwidth* - BW), que pode assumir os valores 125 kHz, 250 kHz ou 500 kHz, e do fator de expansão do sinal (*Spreading Factor* - SF), que varia entre 7 e 12. A definição desses parâmetros tem impacto direto sobre o tempo de transmissão dos dados (*Time on Air* - ToA), tendo impacto ainda sobre o consumo de energia dos nós e no número de colisões no *gateway* (RAZA; KULKARNI; SOORIYABANDARA, 2017).

A relação entre SF e BW também afeta diretamente na sensibilidade do rádio, quando se aumenta o valor de SF e diminui-se o valor de BW, a SNR no receptor diminui, resultando em maior sensibilidade. Outro parâmetro que é ajustado no LoRaWAN, além de BW e SF, é a potência de transmissão (*Transmission Power* - TP). A potência de transmissão empregada pelos dispositivos (nós e *gateways*), é composta por valores inteiros (geralmente definidos em dBm - decibél-miliwatt), que são especificados através de legislações regionais (ALLIANCE, 2023). A definição correta dos parâmetros de transmissão é fundamental para assegurar os requisitos de conectividade da rede e garantir a conectividade simultânea de muitos dispositivos, mantendo um longo alcance.

O ajuste dos parâmetros de transmissão no protocolo LoRaWAN é realizado através do algoritmo de ajuste da taxa de dados (Adaptive Data Rate - ADR), que tem como objetivo otimizar a alocação dos recursos possibilitando comunicações de longa distância, com consumo energético reduzido e de forma escalável (ALLIANCE, 2021).

O ADR é executado tanto no servidor de rede quanto nos nós, de maneira assíncrona e atuando de forma distinta em cada uma das partes. O ADR presente nos nós, atua nos casos de perda de conexão com o servidor de rede. Os nós detectam a perda de conexão pela verificação do não recebimento de mensagens de *downlink* (mensagens enviadas do servidor para o nó), que ocorrem periodicamente após o envio de um conjunto de mensagens de *uplink* (mensagens do nó para o servidor). Nos casos de detecção da perda de conexão, o ADR presente nos nós atua de forma robusta, elevando significativamente os valores de SF e TP. Enquanto para os nós existe uma definição do algoritmo ADR, para o Servidor de rede existe maior flexibilidade para implementações customizadas pelo gestor da rede.

1.1 A INSTITUIÇÃO

A Fundação CERTI é uma instituição de pesquisa e desenvolvimento, que se destaca na oferta de serviços tecnológicos especializados para o setor privado, governamental e terceiro setor. Operando de maneira independente e sem fins lucrativos, a instituição foi criada em 1984, acumulando conhecimento e experiência no cenário de Ciência, Tecnologia e Inovação no Brasil, aliando suas Competências em áreas segmentadas a partir de seus Centros de Referência em Tecnologias Inovadoras (CRITs), que trabalham de forma cooperativa com entidades parceiras do país e do exterior.

Um dos Centros de Referência da Fundação CERTI, é o Centro de Convergência Digital e Mecatrônica (CDM). Este Centro atua no desenvolvimento de produtos e serviços inovadores nas áreas de software, internet, sistemas embarcados, televisão digital, telecomunicações, redes, comunicação móvel, inclusão digital e na área de estratégia de negócios e análise de mercado e tecnologia. O Centro apresenta uma visão multidisciplinar e trabalha, de maneira colaborativa, atuando na fronteira entre as áreas de tecnologia e de negócios, com análises de mercado e de produto para dar suporte à tomada de decisão.

1.2 MOTIVAÇÃO

A constante evolução dos sistemas de comunicação e o crescimento acelerado da IoT fazem com que as empresas busquem continuamente soluções mais sofisticadas para atender às suas necessidades e permanecerem competitivas no mercado. Atuando no segmento de tecnologia, a Fundação CERTI é uma organização de pesquisa e desenvolvimento que presta serviços especializados para a iniciativa privada, governo e terceiro setor. A instituição visa entregar soluções inovadoras para atender às necessidades de seus clientes, auxiliando na manutenção de sua competitividade no mercado.

Visando ampliar sua gama de soluções voltadas para IoT, a Fundação CERTI identificou na tecnologia de comunicação LoRa uma boa oportunidade para expandir seu horizonte na criação de redes autônomas. Assim como em outras áreas de atuação, o primeiro passo para o desenvolvimento de produtos e softwares começa com a construção de uma base de conhecimento (prova de conceito) sobre as novas tecnologias. A partir de uma prova de conceito já realizada, propôs-se o desenvolvimento de melhorias no protocolo de comunicação LoRaWAN, com o objetivo de aplicar a tecnologia no ambiente agropecuário. Constatou-se que, no cenário agropecuário, a disposição de nós de uma rede LoRaWAN se dá em áreas bastante amplas, onde múltiplos nós apresentam-se ligeiramente distantes de um único *gateway* ao qual estão conectados. Isso torna a comunicação suscetível a falhas, como perdas de pacotes, colisão de mensagens e consumo elevado de energia, relacionado à retransmissão de mensagens. O algoritmo ADR, responsável pela gestão dos recursos e parâmetros da rede, é visto como um dos pilares para mitigar a ocorrência destes problemas, através de uma boa gestão dos recursos da rede.

A proposta deste trabalho consiste na sugestão de soluções e na validação por meio de simulações de melhorias no algoritmo ADR. Além disso, propõe-se a elaboração de documentação detalhada das informações para compor uma base de conhecimento para a Fundação.

1.3 OBJETIVOS GERAIS

O objetivo geral deste projeto é a proposição de melhorias para o algoritmo ADR do protocolo LoRaWAN, visando melhorar o desempenho de comunicação em redes destinadas ao ambiente agropecuário.

1.4 OBJETIVOS ESPECÍFICOS

Os objetivos específicos previstos para o trabalho são:

- Elencar os principais problemas relacionados ao algoritmo ADR tradicional, em termos de desempenho de comunicação.
- Propor melhorias que podem ser implementadas no algoritmo ADR, que possibilitem a atenuar ou eliminar alguns de seus principais problemas de desempenho.
- Elaborar um modelo fiel para simulação do comportamento de redes LoRaWAN.
- Realizar simulações sobre as propostas de melhoria, analisar os resultados e reavaliar as propostas.

ESTRUTURA DO DOCUMENTO

Este trabalho está estruturado em 5 capítulos. O primeiro capítulo é voltado para a apresentação geral do projeto. O capítulo 2 contém uma explanação sobre os conceitos mais importantes abordados no desenvolvimento desta monografia. No capítulo 3 são apresentados os métodos e as soluções propostas em torno da problemática principal, o modelo para simulação e detalhes sobre as implementações realizadas para a validação das propostas de melhoria. O capítulo 4 contempla os resultados obtidos e suas respectivas análises. No capítulo 5 são apresentadas as considerações finais e as sugestões para trabalhos futuros feitos pelo autor.

2 REVISÃO DA LITERATURA

Este capítulo traz uma breve introdução sobre a utilização de IoT na agricultura, e o potencial da tecnologia LoRa neste contexto. Em seguida, é apresentada uma base teórica para facilitar a compreensão sobre a tecnologia LoRa e os itens correlacionados a esta, que formaram a base deste projeto. Constatam informações sobre características básicas da camada física da rede e os itens que compõem a estrutura de comunicação desta. Já na camada de controle da rede, são abordados aspectos sobre o protocolo LoRaWAN, as classes de operação e o algoritmo ADR, foco do projeto.

2.1 IOT E AGRICULTURA

A utilização de IoT no meio rural tem se tornado uma realidade a cada dia, isto porque uma variedade de atividades é beneficiada com a adoção de tecnologia. Como resultado, são apontados ganhos de eficiência na produtividade e na otimização de custos com a produção e insumos. Tais fatores contribuem para o aumento da competitividade dos produtores agropecuários diante do mercado nacional e internacional, fortalecendo a segurança alimentar mundial.

Estudos sobre a utilização da IoT apontam como possibilidades de utilização a sua aplicação no monitoramento, controle, logística e predição dentro do setor do agronegócio e em seus segmentos e atividades diversos. O monitoramento de plantas, condições climáticas, solo, água, e animais são algumas das possibilidades de monitoramento no meio agrícola (ELANGO VAN; LAKSHMANAN, 2018).

O emprego de IoT como ferramenta de controle pode ser observado no controle de sistemas de irrigação, distribuição de fertilizantes e pesticidas, iluminação e acesso, conforme demonstrado na Figura 1. Em termos de predição, os dispositivos de IoT fornecem informações que possibilitam o uso de ferramentas para tomada de decisões sobre condições ambientais, estimativa de produção e crescimento da colheita. Na parte final da cadeia, a IoT permite aplicações para a comercialização, transporte de produtos e na gestão da cadeia de suprimentos agroalimentar (ELANGO VAN; LAKSHMANAN, 2018).

A tecnologia LoRa apresenta um potencial enorme para a agricultura, no âmbito da IoT, possibilitando o desenvolvimento de soluções eficientes para enfrentar os desafios do setor. Sua capacidade de prover comunicação sem fio de longo alcance, baixo consumo de energia e conectividade confiável, destacam-se para sua escolha para implementações neste ambiente. Essa tecnologia viabiliza a criação de redes de sensores com a capacidade de cobrir extensas áreas rurais de forma eficiente, possibilitando no futuro a transformação digital da agricultura, impulsionando a produtividade, otimizando recursos e promovendo práticas sustentáveis.

Figura 1 – LoRaWAN na agricultura.



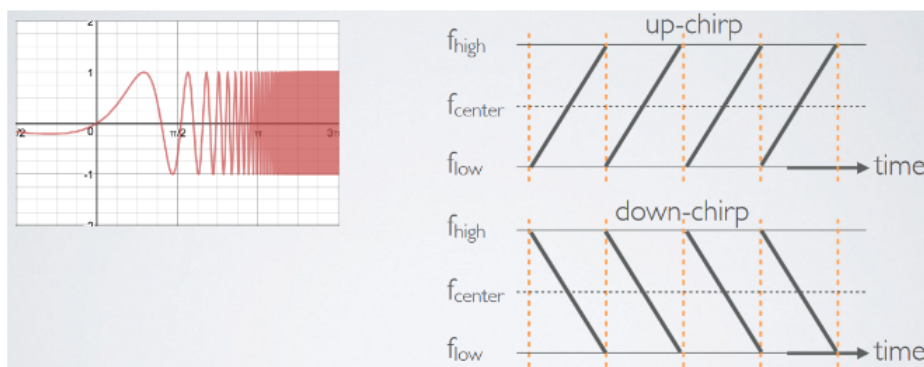
Fonte: adaptado de (SEMTECH, 2023b).

2.2 TECNOLOGIA LORA

LoRa é uma tecnologia de comunicação sem fio de longo alcance, baixo consumo de energia e baixa taxa de transmissão de dados, sendo projetada para conectar dispositivos IoT em grandes áreas geográficas. Este tipo de comunicação emprega uma técnica de modulação de espectro espalhado baseado na modulação *Chirp Spread Spectrum* (CSS). O método CSS é uma técnica de espalhamento espectral que usa pulsos *chirp* modulados em frequência linear de banda larga para codificar informações. Nesta técnica de modulação, como a largura de banda utilizada para a transmissão é maior que a banda mínima necessária para transmitir a informação, a energia do sinal transmitido passa a ocupar uma banda muito maior do que a da própria informação. Isso possibilita que a comunicação LoRa alcance distâncias de alguns quilômetros em áreas urbanas e até dezenas de quilômetros em áreas rurais. Tudo isso aliado a um baixo consumo de energia, o que é importante para dispositivos de IoT que são alimentados por baterias.

Na Figura 2 é demonstrado o comportamento do sinal e a formação do *chirp*, com o aumento da frequência que ocorre em virtude dos pulsos aplicados.

Do ponto de vista de funcionamento, a partir de suas características de modulação de frequência e espalhamento espectral de sinal, o tamanho das mensagens, alcance e o consumo de energia são definidos a partir dos seguintes parâmetros de configuração:

Figura 2 – Demonstração do sinal de *chirp*.

Fonte: (LORA, 2018).

- a taxa de bits (*Bit Rate*);
- o tamanho máximo dos pacotes de dados (*Payload Size*);
- o tempo entre envio e recepção de um sinal (ToA).

Esta parametrização possibilita ao LoRa o gerenciamento da taxa de dados e a potência de saída do sinal de radiofrequência em cada nó que compõem uma rede. Isso se dá através do algoritmo ADR, que busca a otimização das taxas de transmissão, aumento de capacidade da rede e diminuição no consumo de energia (GARCIA; KLEINSCHMIDT, 2017). A definição formal destes parâmetros é apresentada a seguir:

- SF: define o número de bits que podem ser codificados a cada símbolo, logo, quanto maior o SF, maior a quantidade de informações transmitidas a cada modulação do sinal e maior a otimização de processamento.
- BW: define a faixa de frequência em que a rede irá operar, sendo assim, o uso de uma banda de modulação BW mais estreita assegura maior sensibilidade de recepção além de incrementar o ToA, sendo que para um dado SF, a largura de banda pode ser de 125 kHz, 250 kHz ou 500 kHz (WERNER, 2018).

Em relação ao desempenho da tecnologia LoRa, se destacam duas vantagens da técnica de CSS, a robustez com relação ao desvanecimento e o efeito Doppler (WERNER, 2018). A modulação do Lora possibilita a utilização de códigos corretores de erro, que possuem Taxa de Código (CR) entre de 4/5 a 4/8 (SEMTECH, 2023).

Os valores de SF possuem impacto direto sobre a modulação LoRa, que por consequência, têm impacto direto no ToA do sinal transmitido. Valores mais elevados de SF apresentam maior robustez de sinal, possibilitando a demodulação mesmo com a presença de muito ruído (RAZA; KULKARNI; SOORIYABANDARA, 2017). Seis

valores de SF são definidos e empregados pelo LoRaWAN, variando do SF7 à SF12. Sem dúvidas, o aspecto mais importante, diz respeito à semi-ortogonalidade entre os SF, isso possibilita que múltiplos pacotes sejam recebidos pelo *gateway* de forma simultânea e praticamente sem ocorrência de colisões, quando transmitidos utilizando SF diferentes (SEMTECH, 2023).

A definição do SF tem impacto direto no desempenho da rede, já que a relação ruído-sinal (SNR) empregada no *gateway* reduz com o aumento do SF e possibilita que os sinais sofram atenuações maiores, o que permite a comunicação em distâncias superiores. Como consequência do aumento de SF, o ToA também aumenta, o que tem impacto direto no consumo de energia, taxa de dados, e no sucesso das recepções, já que transmissões mais longas possuem maior probabilidade de colisão. A escolha dos parâmetros SF, BW e CR define a taxa de bits (R_b) conforme a Equação 1 (SEMTECH, 2023).

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

2.3 PROTOCOLO DE COMUNICAÇÃO LORAWAN

LoRaWAN é um protocolo de comunicação desenvolvido para a tecnologia LoRa, e o único suportado por esta. O que levou ao seu desenvolvimento foi a utilização em aplicações IoT, de grande área de cobertura e baixo consumo de energia (*Low Power Wide Area Network* - LPWAN), (CARVALHO SILVA; RODRIGUES, 2017).

Nas redes LoRaWAN a comunicação é bidirecional, ou seja, os dispositivos finais fazem transmissões para o *gateway* (*uplink*) e o *gateway* transmite informações para os dispositivos finais (*downlink*). A comunicação é realizada através de frequências determinadas, sendo elas: 169 MHz, 433 MHz, 868 MHz e 915 MHz (WERNER, 2018). Essas frequências, voltadas para comunicação de longo alcance, costumam ser denominadas de sub-gigahertz (sub-GHz).

Para ser capaz de atender os diferentes tipos de aplicações, o LoRaWAN é dividido em três tipos de operação para os dispositivos finais, definidos através das classes A, B e C (ALLIANCE, 2023). Estas classes de operação são apresentadas na Seção 2.6.

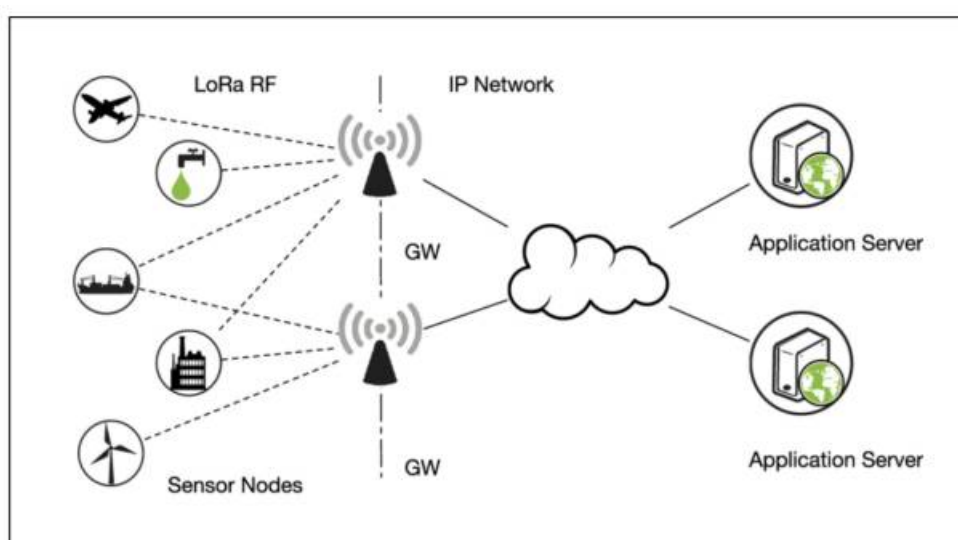
2.4 PRINCIPAIS CARACTERÍSTICAS DAS REDES LORAWAN

As redes LoRaWAN normalmente apresentam uma topologia que utiliza um número reduzido de equipamentos. Enquanto os nós realizam a aquisição e transmissão de dados, o *gateway* que é o segundo componente do sistema, recebe as mensagens e as encaminha ao servidor de rede, utilizando outro tipo de comunicação. Para que o *gateway* receba dados dos nós, o mesmo também precisa ser dotado da tecnologia

LoRa. O servidor de rede, terceiro componente do sistema, realiza o tratamento dos dados e retorna uma mensagem sempre que houver necessidade, realizando desta forma o gerenciamento da rede.

Em uma rede LoRa, os *gateways* são capazes de lidar com centenas de dispositivos ao mesmo tempo, fazendo a recepção em múltiplas frequências simultaneamente. Vale ressaltar que comunicações diretas entre os nós da rede não são suportadas, as mensagens precisam obrigatoriamente transitar pelo *gateway*. Na Figura 2 é demonstrada a estrutura de uma rede LoRaWAN.

Figura 3 – Esquema de uma rede LoRaWAN.



Fonte: (ALGHAMDI; KHAIRULLAH; AL MOJAMED, 2022).

2.5 COMPONENTES DE UMA REDE LORAWAN

As redes LoRaWAN são compostas basicamente por nós, *gateway(s)* e servidor, para assegurar que os dados adquiridos a campo estejam disponíveis através da interface de um sistema. A seguir estes componentes serão abordados detalhadamente.

2.5.1 *End-node*

Um *end-node*, *end-device* ou mesmo nó de uma rede LoRaWAN, é um dispositivo de baixo consumo de energia projetado para se comunicar com o *gateway* desta rede. Os nós podem contar com sensores, medidores, atuadores, entre outros, utilizando LoRaWAN para se comunicar com o *gateway* e enviar as medidas realizadas ou informações pertinentes à aplicação desenvolvida (SANTOS, 2021).

Os nós LoRaWAN são projetados para enviar pequenas quantidades de dados em intervalos programados ou em resposta a eventos específicos. Estes dispositivos

podem atuar nas diferentes classes de operação supracitadas, com diferentes níveis de consumo de energia e capacidade de comunicação.

2.5.2 Gateway

O *gateway* LoRa é o dispositivo responsável por estabelecer uma conexão entre os dispositivos e o servidor de rede, recebendo os dados dos diversos nós, interpretando, armazenando e repassando estes ao servidor de rede (SANTOS, 2021). O *gateway* pode ser conectado ao servidor de rede via conexão física ou sem fio e empregar diferentes protocolos de comunicação. Normalmente o *gateway* utiliza uma comunicação com taxas de transmissão maiores, já que concentra uma grande quantidade de dados vinda de múltiplos nós. O LoRaWAN permite que vários *gateways*, recebam e encaminhem um mesmo pacote para um servidor ou nó (AUGUSTIN *et al.*, 2016), para posterior tratamento da informação e remoção da redundância no servidor de rede.

2.5.3 Servidor de rede

Um servidor de rede LoRaWAN é um componente essencial para uma rede LoRaWAN. O servidor de rede é responsável por validar as informações e a integridade dos pacotes recebidos, e pela estruturação dos pacotes que retornam aos nós, além de realizar o encaminhamento dos dados ao servidor de aplicação (SANTOS, 2021).

O servidor de rede é responsável ainda por gerenciar a rede, coordenar as comunicações entre os dispositivos finais e os *gateways* LoRaWAN, garantindo acima de tudo a segurança e a integridade dos dados transmitidos, e fornecendo serviços como provisionamento de dispositivos e gerenciamento de chaves de criptografia. Além disso, o servidor de rede LoRaWAN pode oferecer recursos adicionais, como a integração com outras redes e sistemas, e a análise de dados para fins particulares de cada projeto.

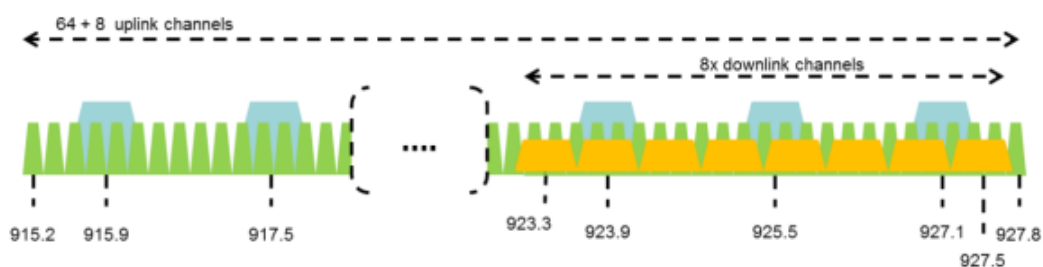
2.5.4 Parâmetros regionais de operação

A Agência Nacional de Telecomunicações (ANATEL) é o órgão brasileiro responsável pela fiscalização e regulamentação dos serviços de telecomunicações, tendo definido os parâmetros regionais utilizados no território nacional, como sendo o padrão australiano AU915, por meio da Resolução 680 do Ato 114448, que diz respeito à restrição da radiação de equipamentos.

Sistemas de modulação de sequência direta, possuem limitação de potência máxima definida em 30 dBm, na faixa de frequência ISM de 915 a 928 MHz. A Figura 4 apresenta os possíveis canais de comunicação para esta faixa de frequência.

O padrão AU915 é composto das seguintes configurações:

Figura 4 – Faixa de frequência do padrão AU915



Fonte: (ALLIANCE, 2023).

- *Upstream*: é composto por 64 canais (de 0 a 63), que utilizam 125 KHz de BW, CR = 4/5, DR0 à DR5 e iniciam em 915,2 MHz chegando até 927,8 MHz;
- *Upstream*: este composto por 8 canais (de 64 a 71), utilizam 500 KHz de BW, DR6 e iniciam com frequência de 915,9 MHz chegando até 927,1 MHz;
- *Downstream*: apresenta 8 canais (de 0 a 7), com 500 KHz de BW, DR8 a DR13 e iniciando em 923,3 MHz chegando até 927,5 MHz.

De acordo com a regulamentação, a taxa de comunicação permitida para o padrão AU915 é dada de acordo com a Tabela 1.

Tabela 1 – Taxa de comunicação para o padrão AU915.

DR	Configuração	Bit rate [bps]
0	SF12 - BW125	250
1	SF11 - BW125	440
2	SF10 - BW125	980
3	SF9 - BW125	1760
4	SF8 - BW125	3125
5	SF7 - BW125	5470
8	SF12 - BW500	980
9	SF11 - BW500	1760
10	SF10 - BW500	3900
11	SF9 - BW500	7000
12	SF8 - BW500	12500
13	SF7 - BW500	21900

Fonte: adaptado de (ALLIANCE, 2023).

2.5.5 Algoritmo Adaptive Data Rate

O algoritmo ADR é parte fundamental do protocolo LoRaWAN, seu objetivo é otimizar a comunicação entre dispositivos finais (nós) e *gateways* em uma rede

LoRaWAN, atuando no ajuste de forma dinâmica, da taxa de dados e da potência de transmissão, para melhorar a eficiência da rede e otimizar o consumo de energia da bateria dos dispositivos. Para definir os valores de SF e TP de uma rede LoRaWAN o algoritmo ADR pode ser utilizado, sendo executado no servidor de rede, nos nós, ou em ambos. O algoritmo presente nos nós é definido pela LoRa Alliance, enquanto o ADR do servidor de rede é de responsabilidade do gestor da rede (ALLIANCE, 2021). A respeito do seu funcionamento, são elencados os seguintes aspectos:

- Coleta de informações: o ADR começa coletando informações sobre o ambiente da rede. Isso inclui a qualidade do sinal, a relação sinal-ruído (SNR), as taxas de perda de pacotes e outros parâmetros de qualidade de link. Essas informações são obtidas a partir das mensagens de *uplink* e das mensagens de *downlink*.
- Cálculo da métrica de link: com base nas informações coletadas, o ADR calcula uma métrica de link que reflete a qualidade da comunicação entre um dispositivo final e o *gateway*. Essa métrica é geralmente expressa em um valor que pode variar de acordo com a implementação, mas o objetivo é medir a confiabilidade da conexão.
- Decisões de ajuste: com a métrica de link em mãos, o ADR decide como ajustar as configurações de transmissão do dispositivo final. Tais decisões incluem:
 - Taxa de dados: o ADR pode aumentar ou diminuir a taxa de dados com base na métrica de link. Se a qualidade do link for boa, pode aumentar a taxa de dados para transmitir mais informações em menos tempo. Se a qualidade for ruim, pode diminuir a taxa de dados para melhorar a confiabilidade da comunicação.
 - Potência de transmissão: o ADR também pode ajustar a potência de transmissão. Em condições ideais, a potência pode ser reduzida para economizar energia. Em condições adversas, a potência pode ser aumentada para estender o alcance ou melhorar a confiabilidade.
- Mensagens de *downlink*: as decisões do ADR são enviadas aos dispositivos finais por meio de mensagens de *downlink*. Essas mensagens contêm as instruções para ajustar as configurações de transmissão. Os dispositivos finais devem então aplicar essas instruções em suas próximas transmissões.
- Feedback: os dispositivos finais enviam feedback sobre o sucesso ou fracasso das novas configurações para os *gateways* por meio de mensagens de *uplink*. Isso permite que o ADR continue a otimizar as configurações com base no desempenho real da rede.

- Ciclo de otimização contínua: o processo de coleta de informações, cálculo da métrica de link, tomada de decisões, envio de mensagens de *downlink* e feedback é contínuo. O ADR adapta as configurações à medida que as condições da rede mudam ao longo do tempo.

O algoritmo ADR do LoRaWAN é projetado para otimizar a comunicação de dispositivos finais em uma rede IoT, por exemplo, garantindo que a taxa de dados e a potência de transmissão sejam ajustadas dinamicamente com base na qualidade do link e nas condições da rede, o que contribui para uma melhor eficiência e vida útil da bateria dos dispositivos.

2.5.6 Segurança

A segurança é um dos quesitos mais importantes quando se trata de dispositivos IoT, sendo responsável pelo funcionamento ininterrupto da rede, assegurando o sistema contra intervenções externas. O protocolo LoRaWAN faz utilização de duas camadas de segurança: uma responsável pela rede e outra pela aplicação. Enquanto a camada de rede garante a autenticidade dos dispositivos conectados à rede, a camada de segurança da aplicação garante que o operador da rede não tenha acesso aos dados das aplicações finais (ALLIANCE, 2023).

A solução de rede LoRaWAN possui um sistema de autenticação baseado em chaves de criptografia, além de um quadro de segurança baseado no sistema de criptografia (*Advanced Encryption Standard* - AES-128). Para garantir a confidencialidade, o AES-128 criptografa o quadro de segurança e em seguida, para garantir a integridade, gera um *Message Integrity Code* (MIC). Cada dispositivo final possui sua atribuição de chaves fornecidas pelos fabricantes de dispositivos ou pelos proprietários das aplicações.

2.6 CLASSES DE OPERAÇÃO

As classes de operação LoRaWAN diferenciam as diversas funcionalidades disponíveis para dispositivos nesta rede de comunicação de longo alcance. Uma característica distintiva notável entre essas classes é a sincronicidade da troca de mensagens. Neste contexto, cada classe possui particularidades que atendem a requisitos específicos de aplicação. A seguir, são apresentados mais detalhes sobre cada uma delas.

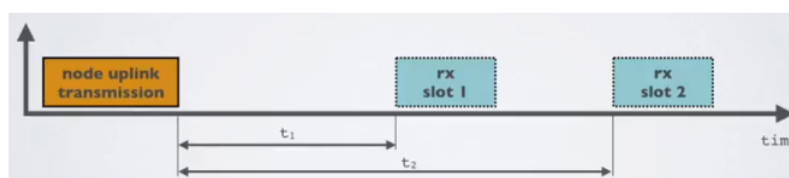
2.6.1 Classe A

A classe de comunicação A é assíncrona, já que a comunicação é iniciada apenas pelo dispositivo final. O dispositivo final pode permanecer em estado de suspensão por longos períodos, de acordo com a aplicação, fazendo com que o mesmo

consoma baixa quantidade de energia (ALLIANCE, 2023). Devido a característica de baixo consumo de energia, esta classe é a mais indicada para dispositivos que operam alimentados por bateria.

Na classe A de comunicação, sempre após a transmissão de uma mensagem de *uplink*, são abertas duas janelas curtas de *downlink*, em que o nó aguarda pelo recebimento de dados. Este comportamento é apresentado na Figura 5, onde o retângulo na cor amarela representa uma janela de *uplink*, enquanto os retângulos azuis representam as janelas de *downlink*, seguindo a linha temporal. O *gateway* pode responder na primeira janela de recebimento ou na segunda, mas não em ambas (WERNER, 2018). Os dispositivos de classe B e C também devem oferecer suporte à funcionalidade de classe A.

Figura 5 – Funcionamento da classe A.



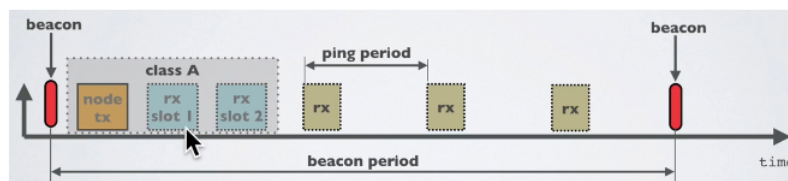
Fonte: (LORA, 2018).

2.6.2 Classe B

Na classe B, os dispositivos apresentam um modo de operação, de forma que em horários programados ocorre um processo de sincronização, a partir do envio de *beacons* pelo *gateway*. Com esta sincronização, é aberta uma janela de transmissão com o dispositivo final, chamada *ping-slot*, que permite que o *gateway* saiba quando o nó está escutando. Neste momento acontece o *uplink* e logo após o *gateway* pode realizar a operação de *downlink* (WERNER, 2018). Este comportamento é observado na Figura 6, em que se pode observar as janelas de comunicação de *uplink* e *downlink* idênticas ao caso da classe A, mas com a inclusão de novas janelas de *downlink*, com um período definido.

As transmissões podem ser programadas pela aplicação com uma latência de até 128 segundos para serem enviadas em horários específicos. Apesar desta configuração ter um consumo adicional de energia, ela ainda continua sendo viável para aplicações alimentadas por bateria (ALLIANCE, 2023). É importante reafirmar que um dispositivo de classe B oferece suporte a classe A, mas não oferece suporte à funcionalidade do dispositivo C.

Figura 6 – Funcionamento da classe B.



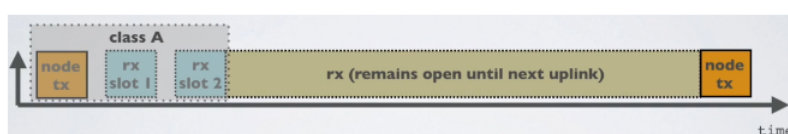
Fonte: (LORA, 2018).

2.6.3 Classe C

Nesta classe, os dispositivos finais permanecem continuamente aptos para a recepção de dados por parte dos *gateways*, a não ser que o mesmo esteja realizando uma operação de *uplink*, momento em que a janela de recepção é fechada. Este modo é o mais custoso em relação ao consumo de energia, por isso é recomendado para aplicações em que o nó está conectado a uma fonte de energia contínua e não a uma bateria (WERNER, 2018).

Na Figura 7 é demonstrada a estrutura de *uplink* da classe C, também com duas janelas de *downlink* como na classe A (na coloração azul). Observa-se ainda janelas de *uplink* em amarelo, e um retângulo bege representando o período em que o dispositivo permanece ininterruptamente apto a receber mensagens de *downlink*. Este é o modo que oferece menor latência para *downlink* (ALLIANCE, 2023). Vale frisar que um dispositivo de classe C não oferece suporte à funcionalidade do dispositivo classe B.

Figura 7 – Funcionamento da classe C.



Fonte: (LORA, 2018).

2.7 CONFIGURAÇÃO DE OPERAÇÃO

Uma configuração que precisa ser definida para estabelecer a comunicação entre os dispositivos finais e o servidor de rede é a especificação do método de ativação. A mesma é realizada no momento da adição de um novo nó à rede, sendo possível a escolha entre os métodos: *Activation by Personalization (ABP)* e *Over-the-Air-Activation*

(OTAA) (SEMTECH, 2023). A seguir serão passados mais detalhes acerca de cada um destes métodos.

2.7.1 ABP

O método ABP permite uma ativação mais simplificada e rápida. Em contrapartida, é menos seguro. No método ABP, os identificadores e chaves são personalizados, com a realização da ativação de forma manual. As chaves de criptografia (NwkSKey e AppSKey) são geradas no servidor da aplicação e inseridas manualmente no dispositivo, juntamente com endereços DevAddr, DevEUI e AppEUI ao qual estão vinculadas. Embora mais simples, este método acaba sendo mais trabalhoso devido à necessidade de configuração manual de cada dispositivo, principalmente quando se trata de redes compostas por centenas ou milhares de *end-nodes* (SEMTECH, 2023).

2.7.2 OTAA

O método de ativação OTAA ocorre com a geração automática de todos os parâmetros de provisionamento do dispositivo. Este método é considerado mais seguro, realizando a troca dinâmica e segura das chaves de configuração entre redes e operadoras, enquanto a rede está ativa (SEMTECH, 2023). No método OTAA, semelhante ao que é feito no método ABP, são gerados os identificadores DevEUI e AppEUI e a chave AppKey, que são idênticos entre si e devem ser configurados no dispositivo. Assim que ocorre a adesão do dispositivo à rede, as chaves de criptografia (NwkSKey e AppSKey) são geradas em um processo de ativação remota do dispositivo, o que faz com que as chaves de segurança sejam renovadas regularmente a cada nova ativação.

2.8 SIMULADOR OMNET++ E FRAMEWORK FLORA

OMNeT++ (Objective Modular Network Testbed in C++) é um ambiente de simulação de código aberto e extensível amplamente utilizado para modelagem e simulação de redes de computadores, sistemas de comunicação e protocolos de comunicação. A ferramenta fornece uma estrutura flexível que permite aos usuários criar modelos detalhados de redes, dispositivos e sistemas de comunicação para fins de pesquisa e experimentação (OMNET++, 2019).

A extensão FLoRa é uma extensão específica do OMNeT++ que se concentra na simulação de redes LoRa e LoRaWAN. A extensão FLoRa foi desenvolvida para suportar a modelagem e simulação de redes LoRaWAN usando o OMNeT++ (FLORA, 2022). Os principais recursos e funcionalidades dessa extensão incluem:

- Modelagem de todos os dispositivos que compõem uma rede LoRa, incluindo as configurações de taxa de dados, potência de transmissão e outros parâmetros relevantes;

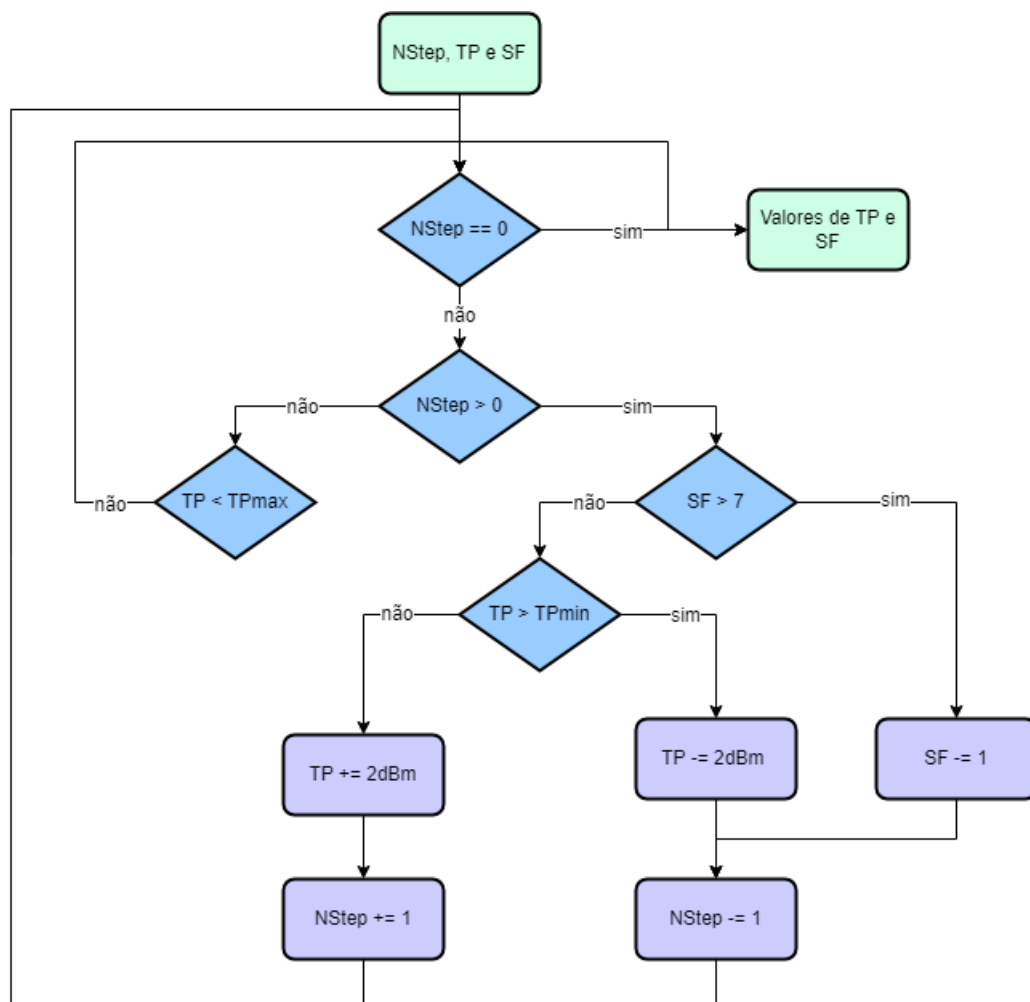
- Modelagem de comportamento de rede, como o ajuste da taxa de dados e potência de transmissão com base nas condições de link, simulando o algoritmo ADR;
- Simulação da cobertura e propagação de sinal, levando em consideração o alcance estendido do LoRa;
- Análise de desempenho, que se dá através da simulação de diferentes cenários, considerando fatores como interferência, colisões e escalabilidade.

A extensão FLoRa é uma ferramenta muito importante, já que permite a reprodução de um ambiente de simulação controlado, o que possibilita a avaliação de desempenho, otimização de configurações e a realização de experimentos sem a necessidade de hardware físico, economizando tempo e recursos (FLORA, 2022).

3 DESENVOLVIMENTO

O algoritmo ADR base é fornecido pela Semtech, sendo adaptado por diversas empresas e instituições que utilizam LoRaWAN. Este é o caso do algoritmo ADR-TTN, adaptado pela *The Things Network* e um dos mais utilizados no mundo, sendo considerado como referência em trabalhos de pesquisa (THE THINGS NETWORK, 2023). O ADR-TTN apresenta relativa simplicidade, atuando através de de passos para ajustar os valores de SF e TP que serão aplicados nos nós de uma rede. Na Figura 8 é apresentado um fluxograma do processo realizado pelo algoritmo.

Figura 8 – Fluxograma do algoritmo ADR.



Fonte: adaptado de (THE THINGS NETWORK, 2023).

O servidor de rede LoRaWAN é responsável por executar o algoritmo ADR para ajustar os parâmetros da rede. O processo começa com a avaliação da qualidade do enlace, através de medições da relação entre ruído e sinal (*Signal-to-Noise Ratio* - SNR) recebidos no *gateway*. Em seguida, o algoritmo calcula o número de passos

para ajustar TP e SF, considerando também a sensibilidade do rádio para cada SF (*Signal-to-Noise Ratio specific to Spreading Factor* - SNR_{SF}) e a margem de erro ($margin_db$), (SEMTECH, 2023a). A cada passo, o servidor de rede comanda o nó a aumentar ou diminuir SF ou TP. A estimativa de qualidade do enlace é obtida a partir do maior valor de SNR para um conjunto das últimas 20 transmissões recebidas. O número de passos (N_{step}) é dado por

$$N_{step} = \frac{SNR_m - SNR_{SF} - margin_db}{3}. \quad (2)$$

Finalmente, os valores de SF e TP são ajustados conforme apresentado no fluxograma da Figura 8. Os valores de TP_{min} e TP_{max} são os valores mínimo e máximo respectivamente, que são estabelecidos pelas entidades regulamentadoras através de especificações regionais, conforme a Seção 2.5.4.

3.1 INEFICIÊNCIAS RELACIONADAS AO ADR

O algoritmo ADR foi projetado para otimizar a comunicação entre dispositivos finais e os *gateways*, porém de acordo com vários estudos, o mesmo apresenta alguns problemas de eficiência. Em alguns trabalhos com o de (KUFKUNES; HANCKE; ABU-MAHFOUZ, 2020), são mencionados diversos aspectos relacionados ao desempenho do algoritmo ADR sendo os principais, listados a seguir:

- Taxa de dados inadequada: em alguns casos, o ADR realiza a redução da taxa de dados quando a qualidade do link não é ideal, o que pode fazer com que as taxas de transferência de dados fiquem muito baixas. Em outros casos, o oposto também pode ocorrer, com o aumento da taxa de dados em condições de boa qualidade de link, resultando em um consumo elevado de energia ou em uma limitação no alcance de transmissão, já que estas variáveis estão diretamente ligadas.
- Consumo elevado de energia: em caso de ajuste da potência de transmissão ou da taxa de dados de forma inadequada, o consumo de energia dos nós pode ser elevado. Se as configurações não forem otimizadas corretamente, os dispositivos podem consumir energia de forma inadequada, reduzindo o intervalo entre trocas da bateria.
- Latência de sinal variável: ocorre como reflexo de constantes mudanças nas configurações de taxa de dados e potência de transmissão. Este aspecto pode ser potencialmente problemático em aplicações que exigem latência consistente.
- Problemas de retransmissão: está relacionada à redução da taxa de dados a ponto das mensagens apresentarem uma alta probabilidade de colisões ou erros,

levando a mais retransmissões de pacotes, aumento do tráfego na rede e piora gradual de desempenho.

- Adaptação lenta: em dadas situações, o ADR pode ser lento em se adaptar às mudanças nas condições de link, resultando em subutilização de recursos, quando a qualidade do link melhora ou, em comunicações problemáticas, quando a qualidade do link piora.
- Sincronização de ADR: em alguns casos pode não haver uma coordenação adequada, resultando no não recebimento das instruções de ajuste de taxa de dados e potência de transmissão, pelos nós.
- Interferências ambientais: obstáculos no ambiente podem afetar negativamente o desempenho do ADR, tornando a otimização das configurações mais desafiadora, principalmente em casos onde essas interferências apresentam-se bastante dinâmicas.
- Seleção de parâmetros incorretos, em alguns casos, podem ocorrer equívocos na interpretação dos parâmetros de qualidade do link e suas métricas, o que pode levar a decisões sub-ótimas na adaptação das configurações de transmissão.

Dados os comportamentos apresentados, fica evidente o espaço existente para melhorias, principalmente, percebe-se que é possível ter ganhos expressivos em desempenho, quando as otimizações são direcionadas para aplicações específicas.

3.2 CENÁRIO RURAL DE APLICAÇÃO

Segundo estudos, o cenário rural é o mais recomendado para a instalação e utilização de LoRaWAN, isso devido a menor quantidade de obstáculos físicos e de interferência de sinal, que impactam na qualidade de transmissão e recepção do sinal de rádio.

Diante dessa circunstância, o cenário rural considerado para realizar as simulações, consiste em uma área ampla, o que implica em distâncias significativas para as transmissões. O relevo considerado não apresenta variações de nível, cenário comum em muitas áreas rurais do Brasil. O objetivo das simulações é registrar o comportamento da rede variando o número de dispositivos conectados e a área de cobertura das redes, avaliando como pequenos ajustes nos parâmetros da rede personalizados para este ambiente, podem contribuir para a melhora no desempenho.

3.3 ELABORAÇÃO DO MODELO DE SIMULAÇÃO

A partir da definição de um cenário base, foram elaboradas simulações para explorar e conhecer o comportamento e desempenho da rede. As simulações foram

elaboradas no *Framework* FLora (OMNet++) empregando as configurações apresentadas na Tabela 2.

Tabela 2 – Parâmetros de simulação.

Parâmetros	Valor
Tempo total simulado	3 dias
Fator de propagação	7-12
Frequência	915 MHz
Largura de banda	125 KHz
Taxa de codificação	4/8
Potência máxima de transmissão	22 dBm
Área de simulação	1 - 25 km ²
Número de nós	25, 50, 100, 250, 500
Intervalo de transmissão	600 s
Repetição	5 vezes

Fonte: autor.

No modelo criado para as simulações, foram adotados apenas um *gateway* e um servidor de rede, enquanto o número de nós e a área foram variados, conforme apresentado na Tabela 2. Uma série de ajustes se fez necessária quanto a configuração da simulação, um destes foi a troca do rádio utilizado, que passou do modelo SX1272, para o modelo SX1262. O principal motivo foi a possibilidade de explorar potências de transmissão mais elevadas, passando de 14 dBm para 22 dBm. Com essa troca, as configurações de sensibilidade e consumo de energia precisaram ser atualizados, cujos dados foram obtidos através das especificações técnicas do componente, vistas em (SEMTECH, 2023c).

No *Framework* FLora um parâmetro que mereceu atenção foi o valor de σ . Este parâmetro refere-se ao desvio padrão logarítmico das atenuações lentas, que indica a variação do sinal e seu respectivo desvio da média. O parâmetro é dependente da localização, enquanto em ambiente urbano seu valor é de 3,57 dB, em ambiente rural o mesmo vale 7,08 dB (MIRÓN, 2023).

As simulações utilizaram como base o método de algoritmo ADR-TTN, presente no *Framework*. Este algoritmo foi usado como base para a aplicação das propostas de melhoria. O Pseudocódigo 1 representa a implementação do algoritmo ADR-TTN para o servidor de rede do *Framework* FLora.

O algoritmo ADR empregado pelos nós no *Framework* FLora, segue a especificação feita pela Semtech. O Pseudocódigo 2 apresenta a forma de implementação conforme (SEMTECH, 2023a).

Realizadas as configurações e a adição dos parâmetros contidos na Tabela 2, obteve-se o primeiro modelo representado de forma gráfica na Figura 9, em uma rede composta por 25 nós.

Algorithm 1 Pseudocódigo do ADR do servidor de rede

```

 $SNR_m \leftarrow (SNR\_of\_last\_20\_frames)$ 
 $SNR_{req} \leftarrow demodulation\_floor(current\_data\_rate)$ 
 $deviceMargin \leftarrow 10$ 
 $SNR_{margin} \leftarrow (SNR_m - SNR_{req} - deviceMargin)$ 
 $steps \leftarrow floor(SNR_{margin}/3)$ 

while  $steps > 0$  and  $SF > SF_{min}$  do
     $SF \leftarrow SF - 1$ 
     $steps \leftarrow steps - 1$ 
end while
while  $steps > 0$  and  $TP > TP_{min}$  do
     $TP \leftarrow TP - 3$ 
     $steps \leftarrow steps - 1$ 
end while
while  $steps > 0$  and  $TP > TP_{max}$  do
     $TP \leftarrow TP + 3$ 
     $steps \leftarrow steps - 1$ 
end while

```

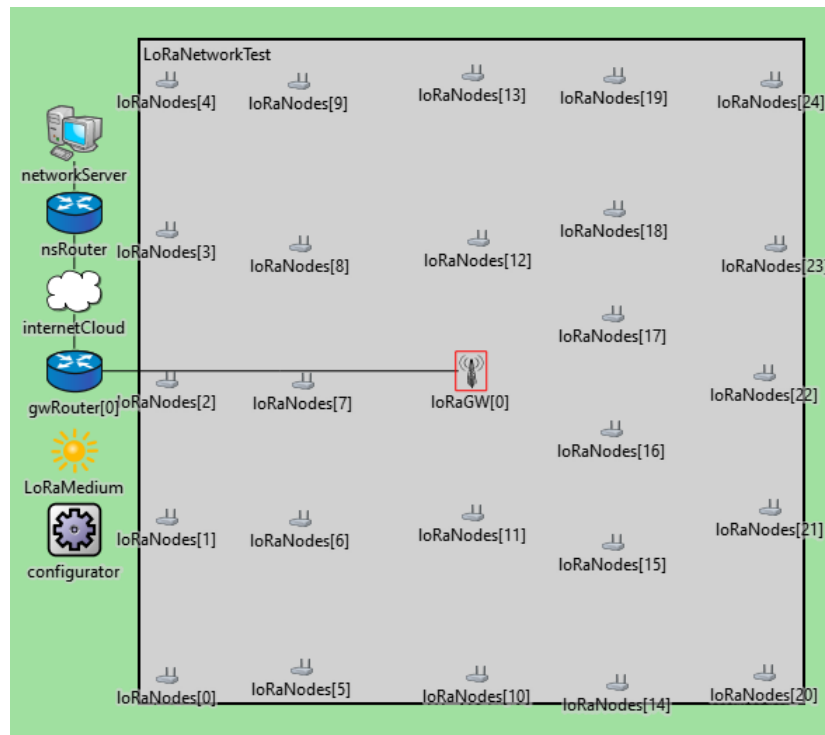
Algorithm 2 Pseudocódigo do ADR do nó

```

 $ADR\_ACK\_LIMIT \leftarrow 64$ 
 $ADR\_ACK\_DELAY \leftarrow 32$ 
 $ADR\_ACK\_CNT \leftarrow 0$ 
if  $uplink\_transmission$  then
     $ADR\_ACK\_CNT \leftarrow ADR\_ACK\_CNT + 1$ 
end if
if  $ADR\_ACK\_CNT == ADR\_ACK\_CNT\_LIMIT$  then
     $Request\_response\_from\_network\_server$ 
end if
if  $ADR\_ACK\_CNT \geq ADR\_ACK\_CNT\_LIMIT + ADR\_ACK\_DELAY$  then
     $Increase\_SF$ 
end if
if  $downlink\_transmission\_received$  then
     $ADR\_ACK\_CNT \leftarrow 0$ 
end if

```

Figura 9 – Modelagem da rede LoRa na forma gráfica.



Fonte: autor.

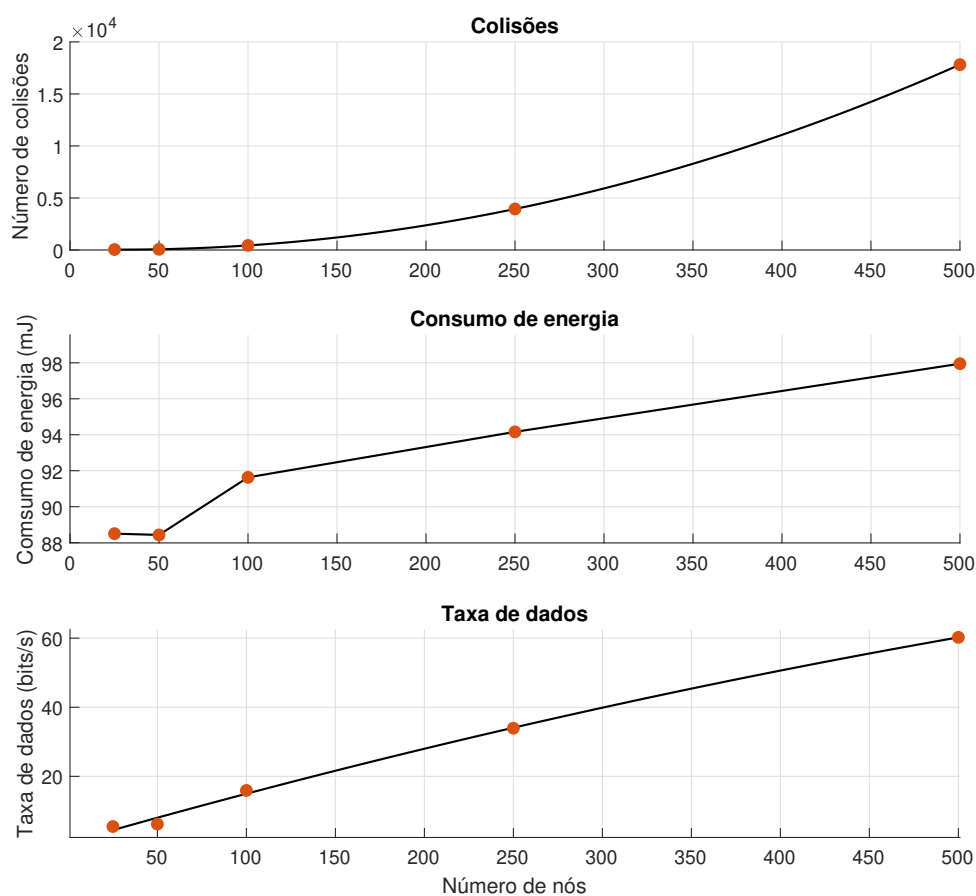
A partir das simulações, os principais parâmetros/indicadores utilizados para analisar o desempenho da rede foram o número de colisões, o consumo médio de consumo de energia (valor médio de consumo por nó em Milijoule - mJ), dado pela Equação 3 e a taxa média de transferência de dados ($bits/s$), encontrada a partir da Equação 4. Outros indicadores também foram empregados para análises mais detalhadas dos resultados das simulações.

$$\text{Consumo médio de energia} = \frac{\sum \text{Consumo de energia dos nós}}{\text{Número de nós}}. \quad (3)$$

$$\text{Taxa de transferência} = \frac{\sum \text{Número de bits de } \textit{uplink} \textit{ e } \textit{downlink}}{\text{Tempo (s)}}. \quad (4)$$

A partir da definição dos cenários e configuração do modelo, um primeiro conjunto de simulações foi realizado. Foi definida uma área de 25 km^2 e em cada simulação o número de nós presente na rede foi alterado. Foram realizadas 5 simulações, com 25, 50, 100, 250 e 500 nós respectivamente. Os valores das variáveis de interesse foram armazenados em vetores, exportados em formato `csv` e analisados através da ferramenta MATLAB. Como resultado do primeiro cenário simulado, obteve-se o gráfico da Figura 10.

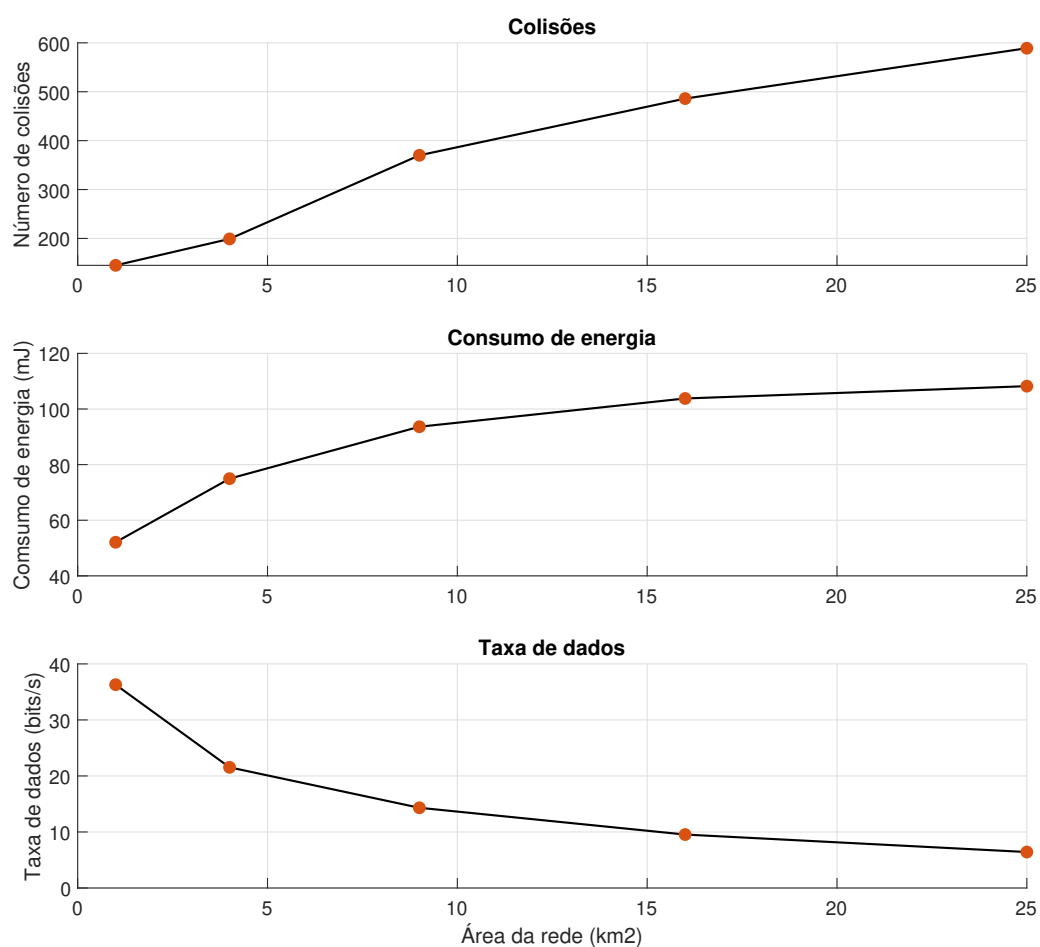
Figura 10 – Resultados das simulações com ADR-TTN de 25 a 500 nós.



Fonte: autor.

O resultado das simulações deste primeiro cenário indicou um incremento de cerca de 400 vezes no número de colisões, quando a rede passou de 25 para 500 nós. O consumo médio de energia dos nós aumentou cerca 11% e a taxa de dados trocados entre o *gateway* e os nós cresceu aproximadamente 1000%.

Um segundo cenário foi elaborado para avaliar o comportamento dos mesmos indicadores do caso anterior, mas considerando a variação da área de cobertura da rede. Neste caso também foram realizadas 5 simulações, em todos os casos foram alocados 100 nós na rede, mas adotando os valores 1 km², 4 km², 9 km², 16 km² e 25 km² de área ocupada pelos nós da rede. A análise dos dados obtidos resultou no gráfico da Figura 11.

Figura 11 – Resultados das simulações com ADR-TTN em áreas de 1 km² a 25 km².

Fonte: autor.

O resultado das simulações do segundo cenário indicou um incremento de cerca de 400% no número de colisões, considerando o aumento da área da rede de 1 km² para 25 km². O consumo médio de energia dos nós cresceu cerca 108% e a taxa de dados trocados entre o *gateway* e os nós diminuiu aproximadamente 82%.

Com as informações obtidas através das simulações e do referencial bibliográfico, algumas estratégias foram traçadas para mitigar os problemas de desempenho da rede LoRaWAN, para os cenários em estudo.

3.4 PROPOSTAS DE MELHORIA PARA O ALGORITMO ADR

Esta seção é destinada a descrever 3 propostas elaboradas pelo autor, de melhoria para o algoritmo ADR. As duas primeiras são voltadas para o ajuste de tempo entre as transmissões de mensagens, considerando o número de nós e a área de

cobertura de uma rede. Atualmente este parâmetro não é considerado pelo algoritmo ADR, podendo ser integrado e utilizado com a finalidade de melhorar o desempenho da rede. Já a terceira proposta prevê a utilização de um sistema de controle Fuzzy para auxiliar no ajuste da taxa de dados e da potência de transmissão. Para isso, a quantidade de nós e a distância dos nós serão considerados nessa abordagem que visa otimizar a distribuição dos nós em diferentes faixas de SF.

3.4.1 Proposta 1

Dois dos pontos relevantes apontados como deficiências do algoritmo ADR são: a ocorrência de colisões e o aumento do consumo de energia, com o aumento do número de nós de uma rede. Neste sentido, a primeira proposta consiste na adição de um fator variável de ajuste da periodicidade no envio de mensagens. A periodicidade é elevada à medida que o número de nós aumenta em uma rede. O Pseudocódigo 3 demonstra a lógica de implementação do algoritmo.

Algorithm 3 Pseudocódigo da proposta 1

```
if Num_nodes > 50 and Num_nodes < 100 then
    Time_to_next_message ← Time_to_next_message * 1.05
end if
if Num_nodes > 100 and Num_nodes < 400 then
    Time_to_next_message ← Time_to_next_message * 1.2
end if
if Num_nodes > 400 then
    Time_to_next_message ← Time_to_next_message * 1.4
end if
```

3.4.2 Proposta 2

A segunda proposta visa amenizar os problemas associados às redes que cobrem áreas amplas, apresentando distâncias elevadas entre o *gateway* e os nós. Nestas condições de aplicação, os valores de SF adotados são elevados (SF10, SF11 e SF12), elevando também o ToA. Este fato tem impacto direto sobre a taxa de utilização destes canais. Seguindo o exemplo da primeira proposta, é possível mitigar o número de colisões e reduzir o consumo de energia com retransmissões, atuando sobre a periodicidade de envio das mensagens. Neste caso, com o aumento da área da rede (ou distância dos nós), é adicionado um fator variável que aumenta a periodicidade de transmissão das mensagens. A proposta a ser integrada ao algoritmo ADR é apresentada no Pseudocódigo 4.

Algorithm 4 Pseudocódigo da proposta 2

```
if  $Dist\_nodes > 1500$  and  $Dist\_nodes < 2000$  then
     $Time\_to\_next\_message \leftarrow Time\_to\_next\_message * 1.05$ 
end if
if  $Dist\_nodes > 2000$  and  $Dist\_nodes < 2500$  then
     $Time\_to\_next\_message \leftarrow Time\_to\_next\_message * 1.1$ 
end if
if  $Dist\_nodes > 2500$  then
     $Time\_to\_next\_message \leftarrow Time\_to\_next\_message * 1.15$ 
end if
```

3.4.3 Proposta 3

A terceira proposta elaborada tem como objetivo oferecer uma solução mais completa, em relação às propostas já apresentadas. Para atingir este objetivo, as informações do número de nós na rede e suas respectivas distâncias até o *gateway*, serão utilizadas para realizar o ajuste dos parâmetros de SF e TP. Propõem-se a utilização de um sistema Fuzzy integrado ao ADR, este sistema recebe os dados da quantidade de nós existente na rede, a posição geográfica de cada nó e a posição do *gateway*. O sistema calcula a distância do nó até o *gateway* e em seguida faz a computação destas informações, entregando o valor de SF, TP adequado para cada nó. Estes valores são informados aos nós e em seguida ocorre o refinamento e ajuste destes parâmetros com o cálculo de N_{steps} (considerando as condições do canal de comunicação). O fluxograma que indica o funcionamento do sistema é apresentado na Figura 12.

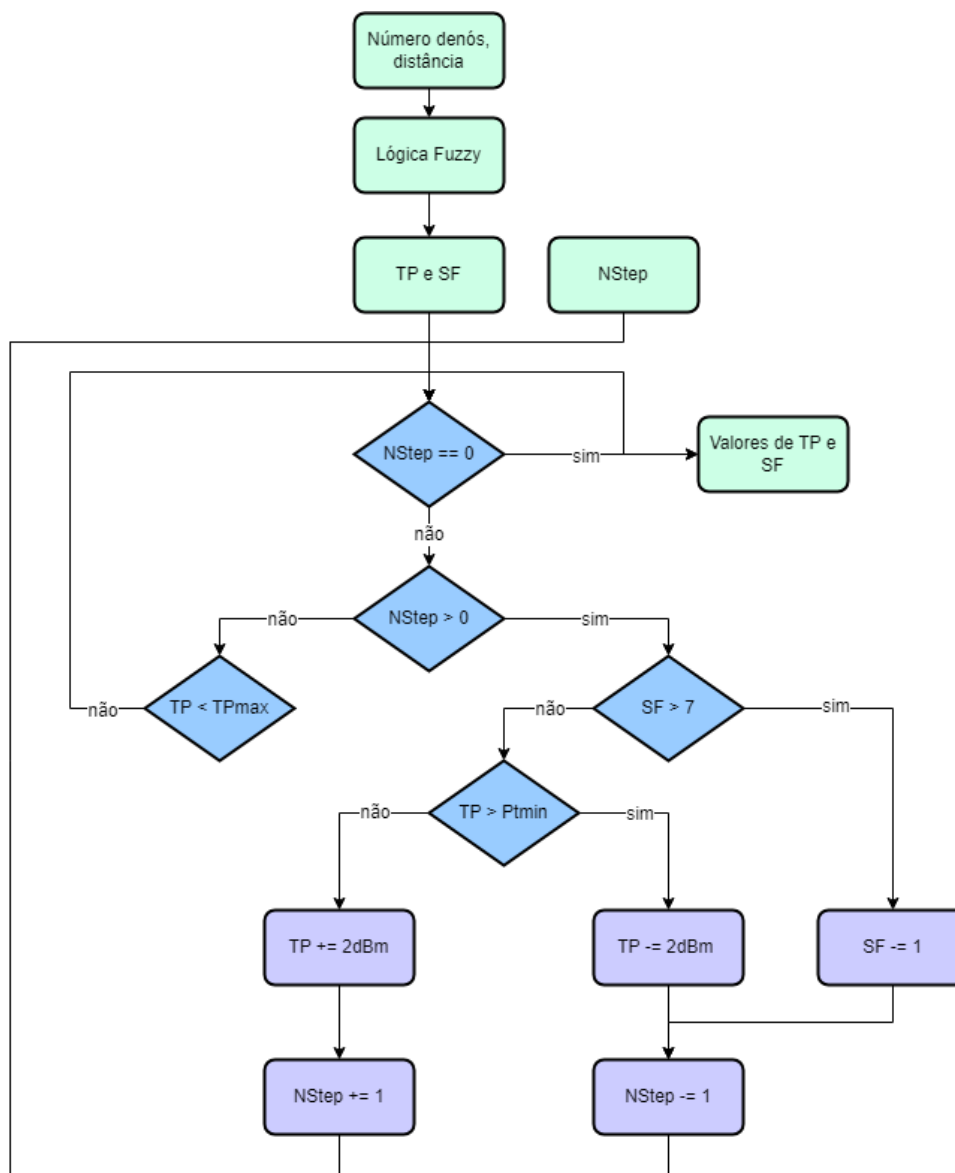
O algoritmo do sistema de controle Fuzzy foi implementado na linguagem de programação Python, utilizando a biblioteca Skufuzzy (SCIKIT-FUZZY, 2023), que fornece um conjunto de ferramentas para facilitar o processo de implementação, principalmente a base de cálculos e estatísticas. O código é disponibilizado no Anexo B.

Os sistemas de controle Fuzzy permitem a inclusão de uma base de conhecimento que compreende o conhecimento do projetista a respeito do processo, aplicando este em controles computadorizados e automação de processos como sistemas industriais e agrícolas (SHEN; ZHU; MAO, 2007). Neste sentido, a análise prévia do comportamento do sistema, foi crucial para o levantamento de uma base de dados e, o desenvolvimento de um conjunto de regras de controle Fuzzy. A base de dados provê as definições necessárias para estabelecer as regras Fuzzy e a manipulação das informações, enquanto o conjunto de regras caracteriza os objetivos de controle e a política de controle. Os parâmetros considerados no sistema Fuzzy foram:

- Número de nós - entrada;
- Posição em x de cada nó - entrada;

- Posição em y de cada nó - entrada;
- SF - saída;
- TP - saída;

Figura 12 – Fluxograma do algoritmo ADR-Fuzzy.

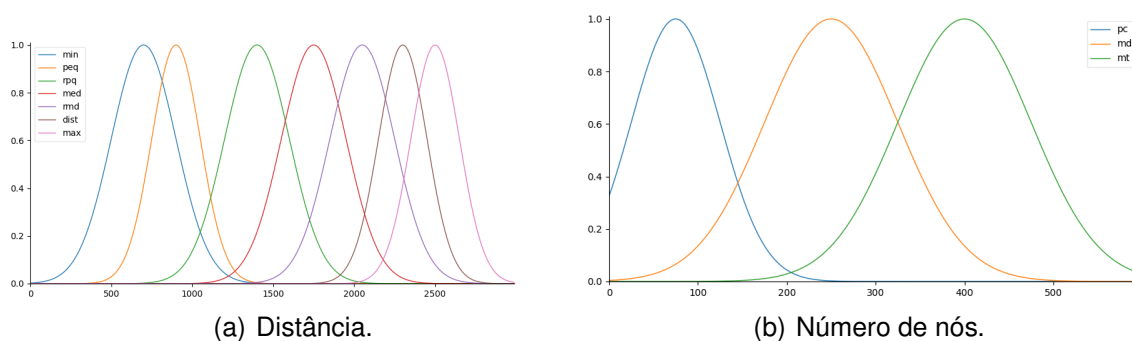


Fonte: autor.

Em relação às variáveis de entrada, estas foram organizadas através de distribuições estatísticas, em que o número de nós foi dividido em poucos ("pc"), quantidade média ("md") e muitos ("mt"), variando de 0 a 600 dispositivos. A variável de distância foi subdividida em 7 faixas, dentro do intervalo de 0 à 3000 metros. Para ambas as

variáveis, adotou-se a distribuição Normal Gaussiana, sendo a representação gráfica apresentada na Figura 13.

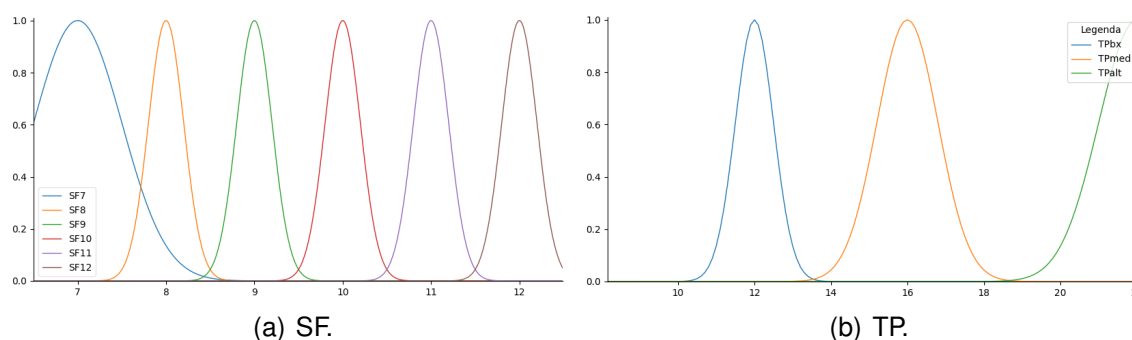
Figura 13 – Distribuição das faixas de valores de distância e número de nós.



Fonte: autor.

As variáveis de saída também foram dispostas em distribuições estatísticas, utilizando a distribuição Normal Gaussiana. Os valores de SF foram distribuídos uniformemente de 7 à 12. Os valores de TP variaram de 10 dBm à 22 dBm e foram agrupados em três faixas, baixa potência de transmissão ("TPbx"), média potência de transmissão ("TPmed") e alta potência de transmissão ("TPalt"). O Gráfico com as distribuições é apresentado na Figura 14.

Figura 14 – Distribuição das faixas de valores de SF e TP.



Fonte: autor.

A obtenção dos valores de saída do sistema depende da inclusão de um conjunto de relações entre as variáveis de entrada e o resultado esperado para cada uma destas relações. Cada relação é denominada como uma regra do sistema Fuzzy. Para o sistema projetado, um conjunto de 42 regras foi implementado, sendo 21 regras que relacionam as variáveis de entrada e resultam no parâmetro de saída SF e, 21 regras que resultam no parâmetro de saída TP. Abaixo são representadas quatro destas regras implementadas.

```
rule1 = ctrl.Rule(dist['min'] & nod['pc'], sf['SF7'])
```

```
rule21 = ctrl.Rule(dist['max'] & nod['mt'], sf['SF12'])
```

```
rule22 = ctrl.Rule(dist['min'] & nod['pc'], tp['TPbx'])
```

```
rule42 = ctrl.Rule(dist['max'] & nod['mt'], tp['TPalt'])
```

No caso da primeira regra listada, define-se que quando a distância for pequena (mínima) e número de nós na rede for reduzido (pouco), o nó deve receber SF7. Para as mesmas condições da regra 1, a regra 22 indica que este nó deve realizar transmissões com TP baixa, na ordem de 12 dBm. Na regra 21, as informações de distância elevada (máxima - na ordem de 3000 m) e muitos nós na rede deve resultar em um SF12. A regra 42 recebe as mesmas condições de entrada da regra 21 e, aponta que o nó deve transmitir com TP alta (que corresponde a 22 dBm).

O conjunto completo de regras é apresentado no algoritmo implementado, no Anexo deste documento. As regras foram definidas de tal forma que redes que contenham muitos nós e distâncias elevadas entre nós e o *gateway*, ocorra uma distribuição destes nós mais distantes entre o SF11 e o SF12, para reduzir a ocorrência de colisões. Alternativamente, valores de TP foram ajustados para que mesmo com SF11, a comunicação possa ocorrer sem perda de dados.

As regras são computadas pelo sistema e dado um conjunto de valores de entrada, o sistema Fuzzy definirá quais valores de saída (TP e SF) melhor atendem à condição imposta.

4 RESULTADOS

4.1 PROPOSTA 1

Com a finalidade de avaliar o desempenho da primeira proposta de melhoria, as simulações foram realizadas com o algoritmo ADR-TTN e em seguida considerando a alteração proposta. Foram considerados os casos por redes compostas por 25, 50, 100, 250 e 500 nós, conforme apresentado previamente. Neste cenário, os valores das variáveis de interesse encontrados para a simulação com o ADR-TTN são apresentados no Quadro 1.

Quadro 1 – Simulações com ADR-TTN.

Número de dispositivos	25	50	100	250	500
N° de colisões	44	68	437	3945	17817
Consumo médio (mJ)	92,6	88,43	91,63	94,16	97,95
Taxa de transferência (bps)	5,48	6,18	15,92	33,94	60,23

Fonte: autor.

Os resultados encontrados a partir das simulações, considerando a melhoria proposta, são apresentados no Quadro 2.

Quadro 2 – Simulações com ADR ativado e ajuste da taxa de dados.

Número de dispositivos	25	50	100	250	500
N° de colisões	44	68	426	2404	6487
Consumo médio (mJ)	92,16	88,43	83,14	72,43	53,36
Taxa de transferência (bps)	5,48	6,18	14,79	27,45	36,29

Fonte: autor.

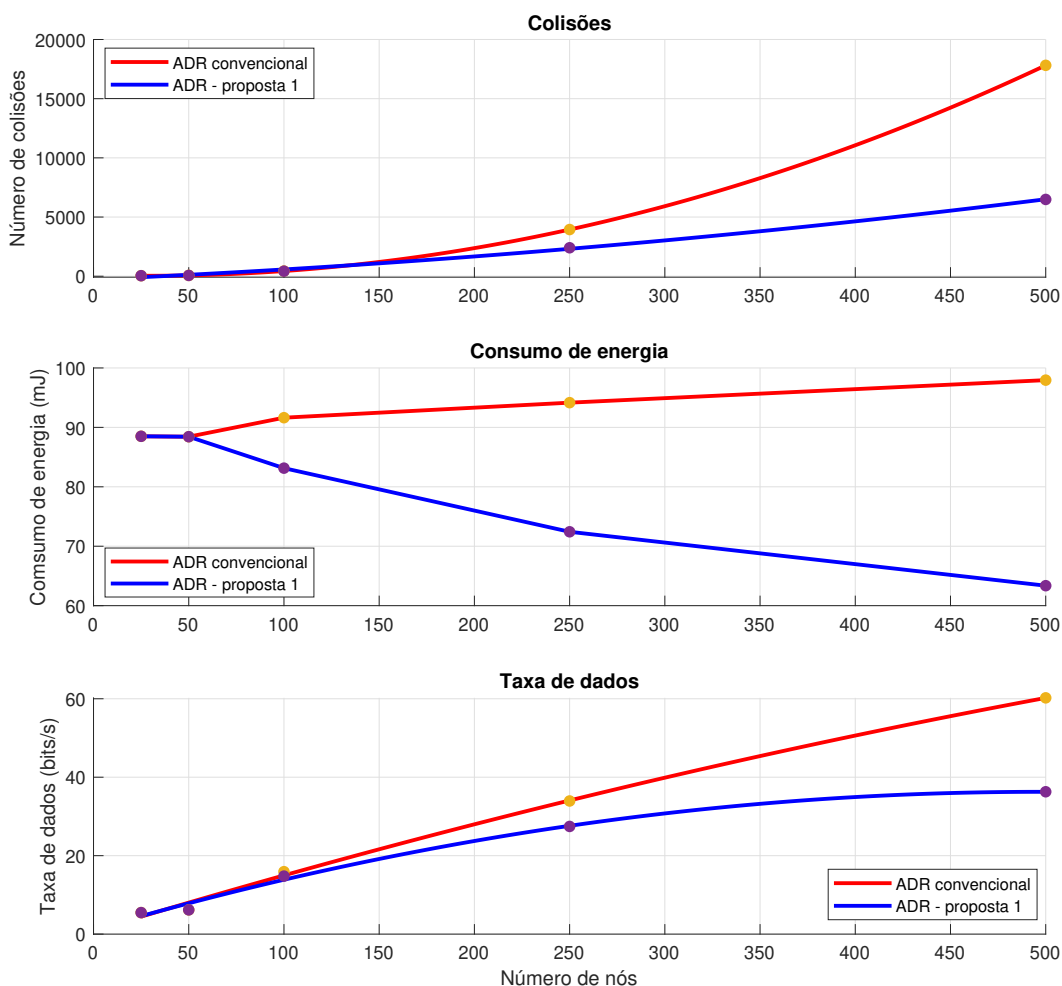
A partir dos dados é possível verificar uma redução considerável do número de colisões, na ordem de 63,5% para 500 nós conectados. A comparação do número de colisões para diferentes quantidades de nós na rede é apresentada no gráfico de colisões da Figura 15, onde estão os valores encontrados, juntamente com as curvas geradas a partir da regressão polinomial que evidenciam o comportamento na faixa de valores analisada.

O consumo médio de energia, também apresentou valores positivos. O gráfico do consumo de energia da Figura 15, demonstra o comparativo. A redução atingiu o valor de 45,5% para 500 nós, ficando evidente que a redução de consumo é superior a redução da quantidade de mensagens enviadas, evidenciando a redução de consumo em retransmissões, ocasionadas principalmente pelas colisões.

A taxa média de transmissão de dados sofreu uma redução que se equipara aos valores percentuais adotados na ampliação do tempo entre transmissões. Essa

taxa é dada pelo total de bits enviados por *downlink* e *uplink* sobre o tempo total de simulação, a comparação é apresentada no gráfico da taxa de dados da Figura 15.

Figura 15 – Resultados das simulações para a primeira proposta.



Fonte: autor.

Os valores obtidos a partir das simulações evidenciaram que os ajustes na periodicidade de transmissão de mensagens, de acordo com o crescimento do número de nós em uma rede LoRa, apresenta resultados positivos na melhoria de desempenho da rede (redução de colisões) e principalmente do consumo energético dos nós. Por outro lado, como consequência ocorre uma redução significativa da transmissão de dados, o que pode não ser interessante para alguns sistemas de monitoramento, que precisam de menor periodicidade dos dados adquiridos.

4.2 PROPOSTA 2

Com a finalidade de avaliar o desempenho da segunda proposta, as simulações foram realizadas com o algoritmo ADR do *framework* e em seguida considerando a proposta, para todo o conjunto de número de nós estipulado inicialmente. Neste cenário, os valores encontrados a partir da simulação com o ADR-TTN são apresentados no Quadro 3.

Quadro 3 – Simulações com ADR-TTN.

Área da rede (km ²)	1	4	9	16	25
N° de colisões	145	199	370	486	589
Consumo médio (mJ)	52,09	74,95	93,62	103,78	108,21
Taxa de transferência (bps)	36,29	21,54	14,32	9,54	6,42

Fonte: autor.

Os resultados encontrados a partir das simulações, considerando a proposta de incremento no intervalo entre transmissões, dado o aumento da área da rede, é apresentado no Quadro 4.

Quadro 4 – Simulações com ADR e ajuste da taxa de dados.

Área da rede (km ²)	1	4	9	16	25
N° de colisões	145	199	314	355	412
Consumo médio (mJ)	52,09	74,95	83,09	85,22	87,93
Taxa de transferência (bps)	36,29	21,54	13,51	8,21	5,10

Fonte: autor.

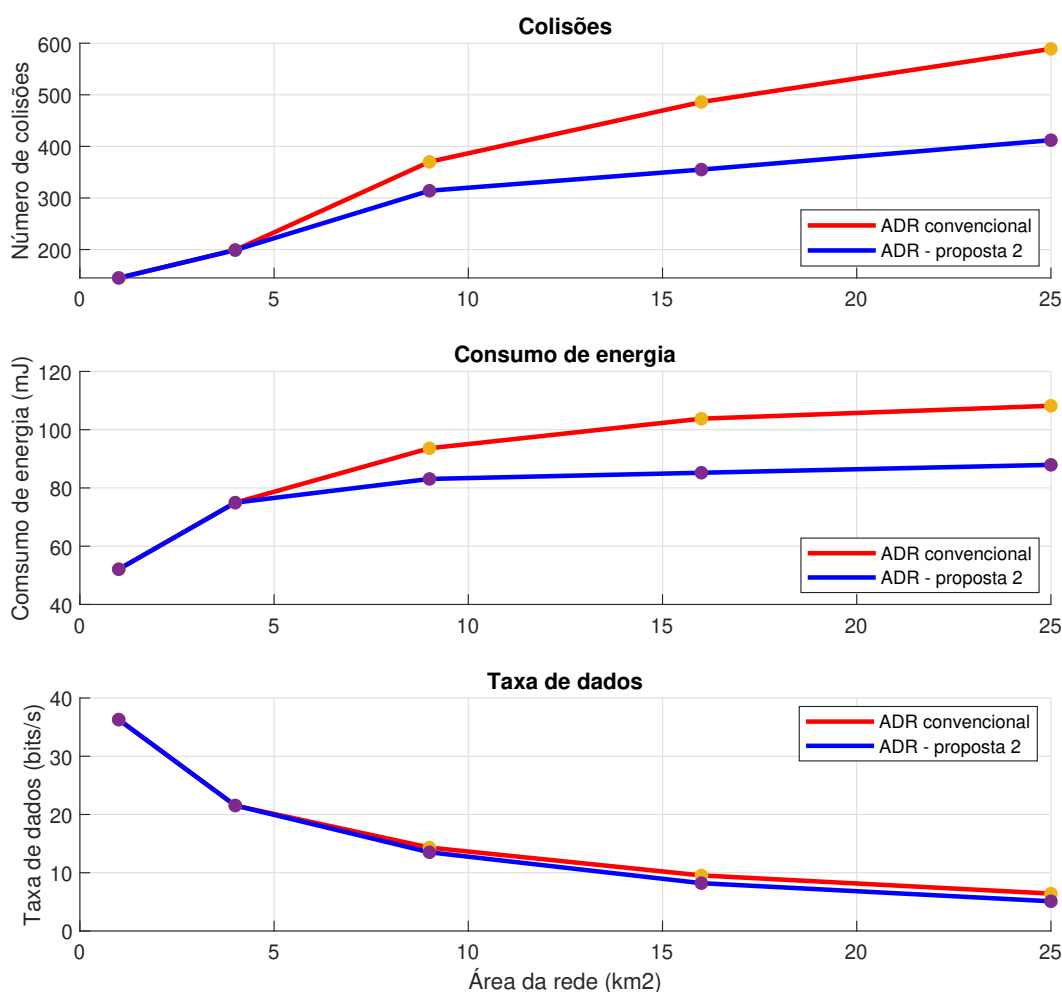
A partir dos dados é possível verificar uma redução considerável do número de colisões, atingindo a marca de 30,0% para uma rede de 25 km². A comparação para as diferentes áreas de cobertura da rede é apresentada no gráfico de colisões da Figura 16, onde estão os valores encontrados, juntamente com as curvas geradas para evidenciar o comportamento na faixa de valores analisada.

O consumo médio de energia, também apresentou valores positivos, o gráfico de consumo de energia da Figura 16, demonstra o comparativo. A redução chegou na ordem de 18,7% para uma área de 25 km², demonstrando uma redução de consumo superior à redução da quantidade de mensagens enviadas, consequência da redução do número de colisões e retransmissões.

A taxa média de transmissão de dados, sofreu uma redução levemente superior aos valores percentuais adotados para ampliar o tempo entre transmissões, a comparação é apresentada no gráfico da taxa de dados da Figura 16.

Os valores obtidos a partir das simulações evidenciaram que ajustes na periodicidade de transmissão de mensagens, de acordo com o crescimento do número de

Figura 16 – Resultado das simulações para a segunda proposta.



Fonte: autor.

nós em uma rede LoRa, apresenta resultados positivos na melhoria de desempenho da rede (redução de colisões) e principalmente do consumo energético dos nós. Por outro lado, como consequência ocorreu uma redução significativa da transmissão de dados, o que pode não ser interessante para alguns sistemas de monitoramento, que precisam de menor periodicidade dos dados adquiridos.

4.3 PROPOSTA 3

Com a finalidade de avaliar o desempenho da proposta com o sistema fuzzy integrado ao ADR, novas simulações foram realizadas com o algoritmo ADR do *framework* e em seguida a proposta estabelecida. Neste cenário, os valores encontrados a partir da simulação com o ADR-TTN são apresentados no Quadro 5.

Quadro 5 – Simulações com ADR-TTN.

Número de dispositivos	50	100	250	500
Nº de colisões	68	437	3945	17817
Consumo médio (mJ)	88,43	91,63	94,16	97,95
Taxa de transferência (bps)	6,18	15,92	33,94	60,23

Fonte: autor.

Já os resultados encontrados a partir das simulações, considerando integração do sistema Fuzzy ao algoritmo ADR são apresentados no Quadro 6.

Quadro 6 – Simulações com ADR - Fuzzy.

Número de dispositivos	50	100	250	500
Nº de colisões	55	379	2153	9035
Consumo médio (mJ)	81,43	82,48	84,27	85,75
Taxa de transferência (bps)	3,65	9,12	30,34	58,26

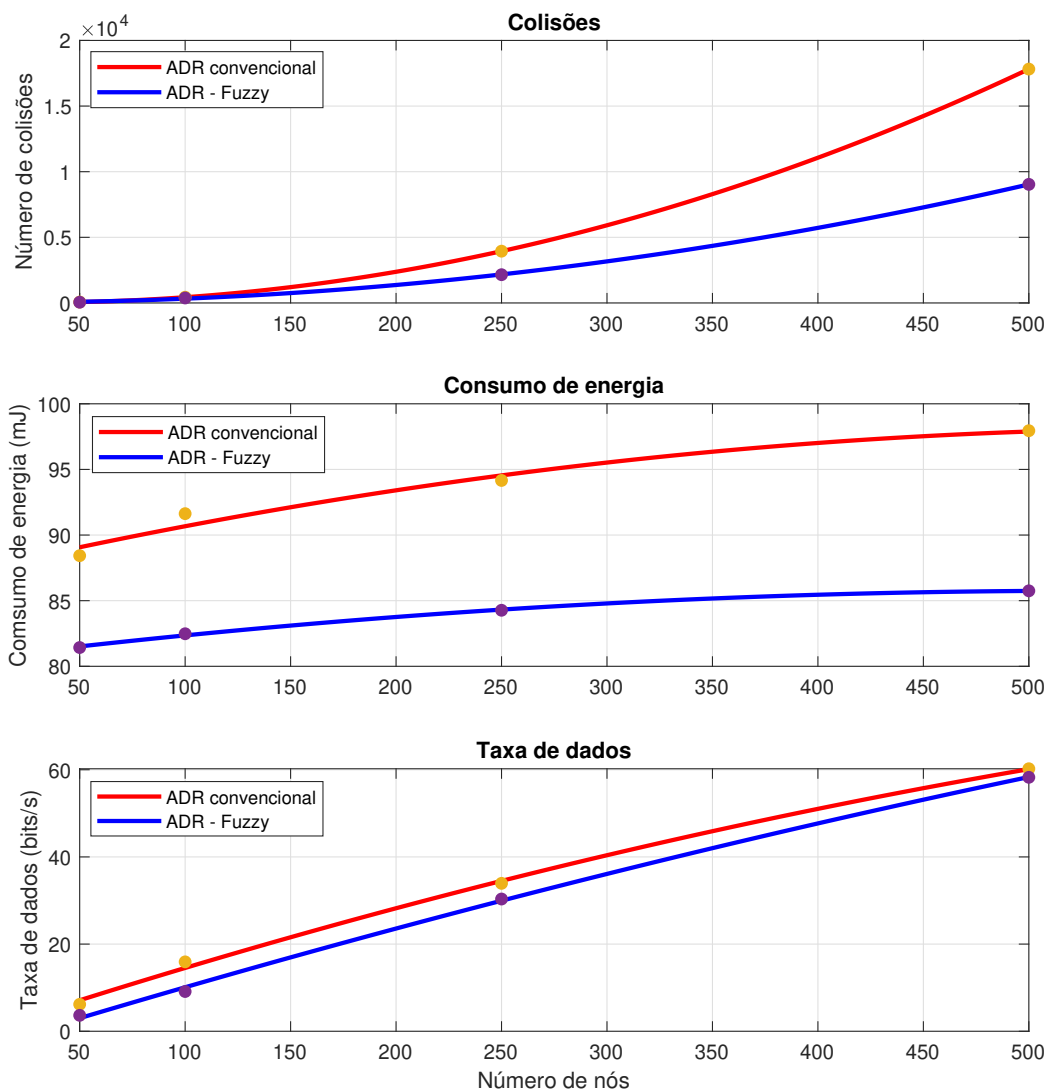
Fonte: autor.

A partir dos dados é possível verificar uma redução considerável do número de colisões, atingindo a marca de 49,3% para uma rede 500 nós. No gráfico de colisões da Figura 17, é possível observar o comportamento para diferentes quantidades de nós na rede. É possível constatar que o aumento do número de nós tem um impacto enorme sobre o número de colisões e, que a adição do sistema Fuzzy ao algoritmo ADR, consegue amenizar de maneira expressiva este comportamento.

O consumo médio de energia dos nós, apresentou uma redução significativa com a utilização do sistema Fuzzy. No gráfico de consumo de energia da Figura 17 é demonstrado o comparativo em relação ao ADR tradicional. É possível verificar que a diferença é ampliada com o aumento do número de nós e, chega a ser 12,5% inferior, para 500 dispositivos conectados na rede.

A taxa média de transmissão de dados sofreu uma redução da ordem de 8,0% para as distintas quantidades de nós. A comparação é apresentada no gráfico da taxa de dados da Figura 17.

Figura 17 – Resultados das simulações para o sistema Fuzzy.

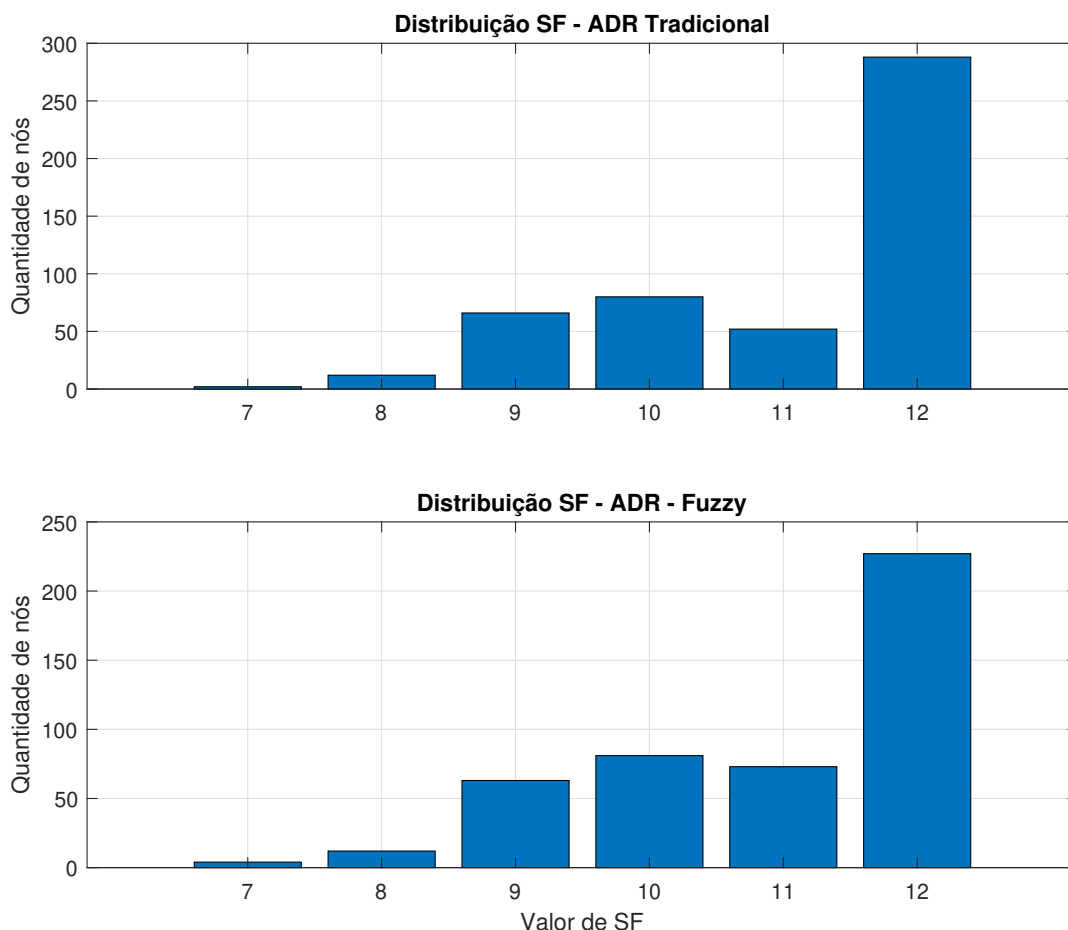


Fonte: autor.

Com os resultados encontrados, fica evidente a relação entre as variáveis, número de colisões, consumo de energia e taxa de dados transmitida. A inclusão do sistema Fuzzy, teve impacto direto na alocação do SF entre os nós, realizando uma melhor ponderação entre o ajuste de TP e SF. Tal comportamento pode ser observado no gráfico da Figura 18, em que para o ADR - Fuzzy é maior a quantidade de nós utilizando SF7, SF8, SF9, SF10 e SF11 e menor quantidade usando SF12. A redução do número de nós utilizando o SF12 também tem impacto sobre o número de colisões e o consumo de energia, já que este apresenta maior ToA. A redução na taxa de dados pode ser explicada pela perda de algumas mensagens devido a utilização de valores

de SF menores, com menor sensibilidade de recepção.

Figura 18 – Comparação da distribuição de SF.



Fonte: autor.

Os valores obtidos a partir das simulações evidenciam um ganho substancial em termos de desempenho da rede. Impactos expressivos foram observados na redução do número de colisões e principalmente no consumo de energia. Certamente a minimização da ocorrência de colisões teve impacto direto sobre o número de retransmissões de mensagens e por consequência no consumo de energia dos nós. Além disso, essa proposta apresentou o menor impacto na redução da taxa de transmissão de dados, podendo ser considerada inexpressiva diante dos ganhos alcançados.

5 CONSIDERAÇÕES FINAIS

Os resultados encontrados foram bastante satisfatórios. Do ponto de vista de desempenho energético, as propostas possuem grande potencial e podem ser empregadas para aumentar a autonomia dos sistemas que operam remotamente e apresentam desafios quanto à manutenção de baterias, já que a redução de consumo foi significativa.

A proposta com o sistema Fuzzy foi sem dúvidas, a que apresentou melhor resultado. Foram oferecidos ganhos significativos em relação ao consumo de energia e na redução da colisão de mensagens, com uma redução inferior na taxa de transferência de dados, quando comparado às outras propostas. Outro aspecto importante, é que com esta proposta, também é possível manter a periodicidade do envio de mensagens fixa, permitindo mais determinismo na aquisição de informações. Mais um ponto que tem grande peso para a terceira proposta diz respeito à sua capacidade de lidar com os dois cenários tratados, aumento do número de nós e aumento da área da rede. Logo, o potencial desta proposta é significativamente maior quando comparado às duas primeiras.

Com relação às duas primeiras propostas, embora o desempenho da rede apresente ganho significativo, é importante observar que o aumento da periodicidade na transferência de mensagens, pode inviabilizar a solução para aplicações que demandam taxas mais elevadas de aquisições.

O presente trabalho atingiu os objetivos previstos, conseguindo validar em ambiente simulado propostas que tem potencial expressivo para melhorar o desempenho de redes LoRaWAN voltadas para o ambiente agropecuário.

Sugestões para trabalhos futuros:

- Realizar a implementação e estudo do comportamento das propostas sugeridas a campo.
- Explorar a proposta com controle Fuzzy incluindo mais variáveis de entrada, regras e formas de dispersão estatística dos dados.
- Utilização de um rádio com maior potência de transmissão para avaliar também casos de maior distância entre *gateway* e nós.
- Considerar casos com nós móveis, analisando o desempenho da terceira proposta em atualizar os parâmetros da rede a partir das informações de localização dos nós.
- Realizar um estudo detalhado para avaliar o impacto do aumento do custo computacional adotando a sistema Fuzzy no ADR.

REFERÊNCIAS

ALGHAMDI, Atheer M.; KHAIRULLAH, Enas F.; AL MOJAMED, Mohammad M. LoRaWAN Performance Analysis for a Water Monitoring and Leakage Detection System in a Housing Complex. **Sensors**, 2022. Disponível em: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9573328/#>.

ALLIANCE, LoRa. **LoRaWAN Specification v1.1**. [S.l.: s.n.], 2021. <https://resources.lora-alliance.org/technical-specifications/lorawan-specification-v1-1>. acessado em 014/10/2023.

ALLIANCE, LoRa. **What is LoRaWAN Specification**. [S.l.: s.n.], 2023. <https://lora-alliance.org/about-lorawan/>. acessado em 04/08/2023.

AUGUSTIN, Aloÿs; YI, Jiazi; CLAUSEN, Thomas; TOWNSLEY, William Mark. **A Study of LoRa: Long Range & Low Power Networks for the Internet of Things**. [S.l.: s.n.], 2016. <https://doi.org/10.3390/s16091466>. acessado em 07/08/2023.

CARVALHO SILVA, Jonathan de; RODRIGUES, Joel J. P. C. **LoRaWAN - A Low Power WAN Protocol for Internet of Things: a Review and Opportunities**. [S.l.: s.n.], 2017. <https://ieeexplore.ieee.org/abstract/document/8019271>. acessado em 02/08/2023.

ELANGO VAN, Dinesh; LAKSHMANAN, Ramesh. Analysis of cloud services in agriculture through internet of things and big data. **Indian Journal of Science and Technology**, 2018.

FLORA. **Flora**. [S.l.: s.n.], 2022. <https://flora.aalto.fi/>. acessado em 05/10/2023.

GARCIA, Paulo Sérgio Rangel; KLEINSCHMIDT, João Henrique. **Tecnologias Emergentes de Conectividade na IoT: Estudo de Redes LPWAN**. [S.l.: s.n.], 2017. <https://biblioteca.sbvt.org.br/articles/618>. acessado em 01/08/2023.

KUFAKUNES, Rachel; HANCKE, Gerhard Petrus; ABU-MAHFOUZ, Adnan. A Survey on Adaptive Data Rate Optimization in LoRaWAN: Recent Solutions and Major Challenges. **Sensors**, 2020.

LORA. **LoRa**. [S.l.: s.n.], 2018.

<https://lora.readthedocs.io/en/latest/#modulation-types-and-chirp-spread-spectrum>. acessado em 05/10/2023.

MIRÓN, Cintia Maroto. **Aplicacions de LoRaWAN en xarxes IoT**. [S.l.: s.n.], 2023.

<https://openaccess.uoc.edu/handle/10609/126226>. acessado em 07/08/2023.

OMNET++. **What is OMNeT++?** [S.l.: s.n.], 2019. <https://omnetpp.org/intro/>. acessado em 05/10/2023.

RAZA, Usman; KULKARNI, Parag; SOORIYABANDARA, Mahesh. Low Power Wide Area Networks: An Overview. **IEEE Communications Surveys Tutorials**, v. 19, n. 2, p. 855–873, 2017.

SANTOS, Alexandro Vanderley dos. **Integração entre Dispositivos LoRa e Servidores de Aplicação Utilizando o Protocolo LoRaWAN**. [S.l.: s.n.], 2021.

<https://repositorio.ufsc.br/handle/123456789/223220>. acessado em 20/07/2023.

SCIKIT-FUZZY. **Skfuzzy 0.3 Documentation**. [S.l.: s.n.], 2023.

<https://scikit-fuzzy.readthedocs.io/en/latest/index.html>. acessado em 02/10/2023.

SEMTECH. **In-Depth: LoRaWAN End Device Activation**. [S.l.: s.n.], 2023.

<https://lora-developers.semtech.com/documentation/tech-papers-and-guides/lorawan-device-activation/device-activation>. acessado em 20/07/2023.

SEMTECH. **LoRaWAN – simple rate adaptation recommended algorithm**.

[S.l.: s.n.], 2023a.

<https://lora-developers.semtech.com/documentation/product-documents/>. acessado em 10/10/2023.

SEMTECH. **Semtech Smart Agriculture**. [S.l.: s.n.], 2023b.

<https://www.semtech.com/lora/lora-applications/smart-agriculture>. acessado em 12/10/2023.

SEMTECH. **Semtech SX1262**. [S.l.: s.n.], 2023c.

<https://www.semtech.com/products/wireless-rf/lora-connect/sx1262>. acessado em 05/08/2023.

SHEN, Ying; ZHU, Zhongxiang; MAO, Enrong. Double-Fuzzy Kalman Filter Based on GPS/IMU/MV Sensor Fusion for Tractor Autonomous Guidance. **IEEE International Conference on Automation and Logistics**, p. 61–65, 2007.

THE THINGS NETWORK. **Adaptive Data Rate**. [S.l.: s.n.], 2023.

<https://www.thethingsindustries.com/docs/reference/adr/>. acessado em 07/10/2023.

WERNER, Thiago. **Uso da tecnologia LoRaWAN para o monitoramento de dados ambientais**. [S.l.: s.n.], 2018.

https://wiki.sj.ifsc.edu.br/images/7/77/TCC290_Thiago_Werner.pdf. acessado em 02/08/2023.

ANEXO A – SIMULAÇÃO COM 10 NÓS

```
[General]
network = flora.simulations.LoRaNetworkTest
output-scalar-file = ../results/n100-gw1-s${runnumber}.ini.sca
**.maxTransmissionDuration = 7s
**.energyDetection = -110dBm
**.vector-recording = false
rng-class = "cMersenneTwister"
**.loRaGW[*].numUdpApps = 1
**.loRaGW[0].packetForwarder.localPort = 2000
**.loRaGW[0].packetForwarder.destPort = 1000
**.loRaGW[0].packetForwarder.destAddresses = "networkServer"
**.loRaGW[0].packetForwarder.indexNumber = 0

**.networkServer.numApps = 1
**.networkServer.**.evaluateADRinServer = true
**.networkServer.app[0].typename = "NetworkServerApp"
**.networkServer.app[0].destAddresses = "loRaGW[0]"
**.networkServer.app[0].destPort = 2000
**.networkServer.app[0].localPort = 1000
**.networkServer.app[0].adrMethod = ${"avg"}

**.numberOfPacketsToSend = 0 # 0 means infinite number of packets
sim-time-limit = 3d
simtime-resolution = -11
repeat = 5

**.timeToFirstPacket = exponential(1000s)
**.timeToNextPacket = 840s
**.alohaChannelModel = false

#nodes features
**.loRaNodes[*]**.initFromDisplayString = false
**.loRaNodes[*]**.evaluateADRinNode = true
**.loRaNodes[*]**.initialLoRaBW = 125 kHz
**.loRaNodes[*]**.initialLoRaCR = 4
**.loRaNodes[*].numApps = 1
```

```
** .loRaNodes[*].app[0].typename = "SimpleLoRaApp"

#gateway features
** .LoRaGWNic.radio.iAmGateway = true
** .loRaGW[*]**.initFromDisplayString = false
** .gateway.modulationMode = "LoRa125k"

#power consumption features
** .loRaNodes[*].LoRaNic.radio.energyConsumer.typename =
    "LoRaEnergyConsumer"
** .loRaNodes[*]**.energySourceModule =
    "^ .IdealEpEnergyStorage"
** .loRaNodes[*].LoRaNic.radio.energyConsumer.
configFile = xmldoc("../energyConsumptionParameters.xml")

#general features
** .sigma = 7.08
** .constraintAreaMinX = 0m
** .constraintAreaMinY = 0m
** .constraintAreaMinZ = 0m
** .constraintAreaMaxZ = 0m

LoRaNetworkTest.**.radio.separateTransmissionParts = false
LoRaNetworkTest.**.radio.separateReceptionParts = false

** .ipv4Delayer.config = xmldoc("../cloudDelays.xml")
** .radio.radioMediumModule = "LoRaMedium"
** .LoRaMedium.pathLossType = "LoRaLogNormalShadowing"
** .minInterferenceTime = 0s
** .displayAddresses = false

** .numberOfNodes = 10
** .numberOfGateways = 1
** .loRaGW[*]**.initialLoRaBW = 125 kHz
** .constraintAreaMaxX = 4000.0m
** .constraintAreaMaxY = 4000.0m
** .loRaGW[0]**.initialX = 2000.00m
** .loRaGW[0]**.initialY = 2000.00m
** .loRaNodes[0]**.initialX = 300.63m
```

```
** .loRaNodes [0] .** .initialY = 3800.08m
** .loRaNodes [1] .** .initialX = 200.29m
** .loRaNodes [1] .** .initialY = 3004.24m
** .loRaNodes [2] .** .initialX = 200.60m
** .loRaNodes [2] .** .initialY = 2070.25m
** .loRaNodes [3] .** .initialX = 200.86m
** .loRaNodes [3] .** .initialY = 1029.63m
** .loRaNodes [4] .** .initialX = 200.85m
** .loRaNodes [4] .** .initialY = 300.11m
** .loRaNodes [5] .** .initialX = 1018.27m
** .loRaNodes [5] .** .initialY = 3701.84m
** .loRaNodes [6] .** .initialX = 1004.11m
** .loRaNodes [6] .** .initialY = 2914.96m
** .loRaNodes [7] .** .initialX = 1030.86m
** .loRaNodes [7] .** .initialY = 2075.17m
** .loRaNodes [8] .** .initialX = 1010.41m
** .loRaNodes [8] .** .initialY = 937.16m
** .loRaNodes [9] .** .initialX = 1000.53m
** .loRaNodes [9] .** .initialY = 304.82m
** .loRaNodes [0] .** initialLoRaSF = 12
** .loRaNodes [0] .** initialLoRaTP = 18dBm
** .loRaNodes [1] .** initialLoRaSF = 12
** .loRaNodes [1] .** initialLoRaTP = 20dBm
** .loRaNodes [2] .** initialLoRaSF = 12
** .loRaNodes [2] .** initialLoRaTP = 20dBm
** .loRaNodes [3] .** initialLoRaSF = 12
** .loRaNodes [3] .** initialLoRaTP = 20dBm
** .loRaNodes [4] .** initialLoRaSF = 12
** .loRaNodes [4] .** initialLoRaTP = 20dBm
** .loRaNodes [5] .** initialLoRaSF = 12
** .loRaNodes [5] .** initialLoRaTP = 18dBm
** .loRaNodes [6] .** initialLoRaSF = 11
** .loRaNodes [6] .** initialLoRaTP = 17dBm
** .loRaNodes [7] .** initialLoRaSF = 12
** .loRaNodes [7] .** initialLoRaTP = 20dBm
** .loRaNodes [8] .** initialLoRaSF = 12
** .loRaNodes [8] .** initialLoRaTP = 20dBm
** .loRaNodes [9] .** initialLoRaSF = 12
** .loRaNodes [9] .** initialLoRaTP = 20dBm
```

ANEXO B – ALGORITMO DO SISTEMA FUZZY

```

"""
PFC
Autor:
Valdecir Hoffmann
"""

import math
import matplotlib.pyplot as plt
import numpy as np
import skfuzzy as fuzz
from skfuzzy import control as ctrl
plt.rcParams["figure.figsize"] = (12, 5)

# Localizacao dos dispositivos em coordenadas cartesianas

pos_nodes_x = [0.63, 0.29, 0.60, 100.86, 0.85, 18.27, 4.11]
pos_nodes_y = [500.10, 10.20, 900.74, 300.12, 02.95, 187.12, 764.11]
pos_gateway = (2000, 2000)

# Numero de nos
n_nodes = 7

# Calculo da distancia a partir da posicao dos nos

def calc_dist(xg, yg, xn, yn):
    return math.sqrt((xn - xg)**2 + (yn - yg)**2)

"""
Definicao das variaveis linguisticas:

SF -> Spreading Factor (saida)
TP -> Transmission Power (saida)

X -> Posicao em x (entrada)
Y -> Posicao em y (entrada)
N -> Numero de end-nodes (entrada)

```

*Alocacao do Spreading Factor:**SF7 -> Maxima taxa de dados**SF8 -> Elevada taxa de dados**SF9 -> Taxa razoavelmente elevada de dados**SF10 -> Taxa intermediaria de dados**SF11 -> Baixa taxa de dados**SF12 -> Minima taxa de dados**Alocacao de Transmission Power:**TPbx -> Potencia baixa**TPmed -> Potencia media**TPalt -> Potencia elevada*

"""

*# Distribuicao das faixas de distancia (alcance)**dist = ctrl.Antecedent(np.arange(0, 3000, 100), 'dist')**dist['min'] = fuzz.gaussmf(dist.universe, 700, 200)**dist['peq'] = fuzz.gaussmf(dist.universe, 900, 150)**dist['rpq'] = fuzz.gaussmf(dist.universe, 1400, 200)**dist['med'] = fuzz.gaussmf(dist.universe, 1750, 200)**dist['rmd'] = fuzz.gaussmf(dist.universe, 2050, 200)**dist['dist'] = fuzz.gaussmf(dist.universe, 2300, 150)**dist['max'] = fuzz.gaussmf(dist.universe, 2500, 150)**# Distribuicao das quantidades de nos**nod = ctrl.Antecedent(np.arange(0, 600, 5), 'nod')**nod['pc'] = fuzz.gaussmf(nod.universe, 75, 50)**nod['md'] = fuzz.gaussmf(nod.universe, 250, 75)**nod['mt'] = fuzz.gaussmf(nod.universe, 400, 75)**sf = ctrl.Consequent(np.arange(6.5, 12.5, 0.1), 'sf')**sf['SF7'] = fuzz.gaussmf(sf.universe, 7, 0.5)**sf['SF8'] = fuzz.gaussmf(sf.universe, 8, 0.2)**sf['SF9'] = fuzz.gaussmf(sf.universe, 9, 0.2)**sf['SF10'] = fuzz.gaussmf(sf.universe, 10, 0.2)*

```
sf['SF11'] = fuzz.gaussmf(sf.universe, 11, 0.2)
```

```
sf['SF12'] = fuzz.gaussmf(sf.universe, 12, 0.2)
```

```
tp = ctrl.Consequent(np.arange(8, 22, 0.1), 'tp')
```

```
tp['TPbx'] = fuzz.gaussmf(tp.universe, 12, 0.5)
```

```
tp['TPmed'] = fuzz.gaussmf(tp.universe, 16, 0.8)
```

```
tp['TPalt'] = fuzz.gaussmf(tp.universe, 22, 1)
```

Regras Fuzzy

```
rule1 = ctrl.Rule(dist['min'] & nod['pc'], sf['SF7'])
```

```
rule2 = ctrl.Rule(dist['min'] & nod['md'], sf['SF7'])
```

```
rule3 = ctrl.Rule(dist['min'] & nod['mt'], sf['SF7'])
```

```
rule4 = ctrl.Rule(dist['peq'] & nod['pc'], sf['SF8'])
```

```
rule5 = ctrl.Rule(dist['peq'] & nod['md'], sf['SF8'])
```

```
rule6 = ctrl.Rule(dist['peq'] & nod['mt'], sf['SF7'])
```

```
rule7 = ctrl.Rule(dist['rpq'] & nod['pc'], sf['SF9'])
```

```
rule8 = ctrl.Rule(dist['rpq'] & nod['md'], sf['SF9'])
```

```
rule9 = ctrl.Rule(dist['rpq'] & nod['mt'], sf['SF8'])
```

```
rule10 = ctrl.Rule(dist['med'] & nod['pc'], sf['SF10'])
```

```
rule11 = ctrl.Rule(dist['med'] & nod['md'], sf['SF10'])
```

```
rule12 = ctrl.Rule(dist['med'] & nod['mt'], sf['SF9'])
```

```
rule13 = ctrl.Rule(dist['rmd'] & nod['pc'], sf['SF11'])
```

```
rule14 = ctrl.Rule(dist['rmd'] & nod['md'], sf['SF11'])
```

```
rule15 = ctrl.Rule(dist['rmd'] & nod['mt'], sf['SF11'])
```

```
rule16 = ctrl.Rule(dist['dist'] & nod['pc'], sf['SF12'])
```

```
rule17 = ctrl.Rule(dist['dist'] & nod['md'], sf['SF11'])
```

```
rule18 = ctrl.Rule(dist['dist'] & nod['mt'], sf['SF12'])
```

```
rule19 = ctrl.Rule(dist['max'] & nod['pc'], sf['SF12'])
```

```
rule20 = ctrl.Rule(dist['max'] & nod['md'], sf['SF12'])
```

```
rule21 = ctrl.Rule(dist['max'] & nod['mt'], sf['SF12'])
```

```
rule22 = ctrl.Rule(dist['min'] & nod['pc'], tp['TPbx'])
```

```
rule23 = ctrl.Rule(dist['min'] & nod['md'], tp['TPmed'])
```

```
rule24 = ctrl.Rule(dist['min'] & nod['mt'], tp['TPalt'])
```

```
rule25 = ctrl.Rule(dist['peq'] & nod['pc'], tp['TPbx'])
```

```
rule26 = ctrl.Rule(dist['peq'] & nod['md'], tp['TPalt'])
```

```
rule27 = ctrl.Rule(dist['peq'] & nod['mt'], tp['TPalt'])
```

```
rule28 = ctrl.Rule(dist['rpq'] & nod['pc'], tp['TPmed'])
```

```
rule29 = ctrl.Rule(dist['rpq'] & nod['md'], tp['TPalt'])
rule30 = ctrl.Rule(dist['rpq'] & nod['mt'], tp['TPalt'])
rule31 = ctrl.Rule(dist['med'] & nod['pc'], tp['TPmed'])
rule32 = ctrl.Rule(dist['med'] & nod['md'], tp['TPalt'])
rule33 = ctrl.Rule(dist['med'] & nod['mt'], tp['TPalt'])
rule34 = ctrl.Rule(dist['rmd'] & nod['pc'], tp['TPmed'])
rule35 = ctrl.Rule(dist['rmd'] & nod['md'], tp['TPalt'])
rule36 = ctrl.Rule(dist['rmd'] & nod['mt'], tp['TPalt'])
rule37 = ctrl.Rule(dist['dist'] & nod['pc'], tp['TPmed'])
rule38 = ctrl.Rule(dist['dist'] & nod['md'], tp['TPalt'])
rule39 = ctrl.Rule(dist['dist'] & nod['mt'], tp['TPalt'])
rule40 = ctrl.Rule(dist['max'] & nod['pc'], tp['TPalt'])
rule41 = ctrl.Rule(dist['max'] & nod['md'], tp['TPalt'])
rule42 = ctrl.Rule(dist['max'] & nod['mt'], tp['TPalt'])
```

```
# Processamento das regras
```

```
agreg = ctrl.ControlSystem(  
    [rule1, rule2, rule3, rule4, rule5, rule6, rule7, rule8,  
     rule9, rule10, rule11, rule12, rule13, rule14, rule15,  
     rule16, rule17, rule18, rule19, rule20, rule21, rule22,  
     rule23, rule24, rule25, rule26, rule27, rule28, rule29,  
     rule30, rule31, rule32, rule33, rule34, rule35, rule36,  
     rule37, rule38, rule39, rule40, rule41, rule42])
```

```
controle = ctrl.ControlSystemSimulation(agreg)
```

```
# Geracao dos resultados
```

```
def fuzzy_control(distance, n_nodes):  
    controle.input['dist'] = distance  
    controle.input['nod'] = n_nodes  
    controle.compute()  
    sf_val = controle.output['sf']  
    tp_val = controle.output['tp']  
    return sf_val, tp_val
```

```
# Execucao para 7 nos
```



```
for i in range(n_nodes):
    distancia = calc_dist(pos_gat[0], pos_gat[1],
    pos_nodes_x[i], pos_nodes_y[i])
    sf_value , tp_value = fuzzy_control(distancia , 100)

    if (sf_value - int(sf_value)) > 0.5:
        sf_value = int(sf_value) + 1

    if (tp_value - int(tp_value)) > 0.5:
        tp_value = int(tp_value) + 1

    if sf_value > 12:
        sf_value = 12

    if tp_value > 22:
        tp_value = 22

    # Resultados

    print("Node[ ", i, " ]_SF_=", sf_value , sep=' ')
    print("Node[ ", i, " ]_TP_=", tp_value , sep=' ')
```