

AVALIAÇÃO DE ALGORITMOS ADR PARA REDES LONG RANGE**EVALUATION OF ADR ALGORITHMS FOR LONG RANGE NETWORKS**

Juliana Capucho Pereira^{1, i}
Victor Hugo de Carvalho Olmo^{2, ii}
Leandro Poloni Dantas^{3, iii}
Marcones Cleber Brito da Silva^{4, iv}
Luiz Carlos Canno^{5, v}
Fernando Simplicio de Sousa^{6, vi}

Data de submissão: (19/05/2023) Data de aprovação: (24/07/23)

RESUMO

Dispositivos sem fio permitem uma ampla variedade de aplicações e a eficiência energética é requisito chave para maximizar o tempo de vida da bateria dos dispositivos das redes de *Internet of Things* (IoT). Dentre as novas tecnologias de redes IoT, destaca-se a técnica de transmissão *Long Range* (LoRa) cuja característica principal é a transmissão entre dispositivos com eficiência energética em distâncias longas. Um desafio inerente à questão energética do rádio LoRa é determinar qual o conjunto de parâmetros de transmissão mais apropriado e, para contorná-lo, o mecanismo *Adaptive Data Rate* (ADR) pode ser empregado. Neste sentido, este trabalho avaliou experimentalmente o mecanismo ADR proposto por Bor & Roedig (ADR BR) e comparou seu desempenho com o mecanismo proposto pela *Long Range Wide Area Network* (LoRaWAN), com base no número total de *frames*, número de iterações e o consumo energético do conjunto de parâmetros ao final da execução. Os experimentos foram conduzidos utilizando o rádio LoRa SX1276, em ambiente urbano. O projeto utilizou apenas nós do tipo *endpoint* formando uma rede ponto a ponto entre o *source* e o *sync*, sem o uso de equipamentos do tipo *gateway*. Os principais resultados indicaram que o algoritmo ADR BR determinou um conjunto de parâmetros com consumo até aproximadamente 91% menor quando comparado à configuração determinada através do algoritmo LoRaWAN para o mesmo cenário. Utilizando-se o mecanismo de *early termination*, notou-se uma redução de aproximadamente 1/3 do número total de pacotes necessários para o algoritmo determinar o conjunto final.

Palavras-chave: LoRa; LoRaWAN; ADR; consumo de energia.

ABSTRACT

Wireless devices provide a wide variety of applications and energy efficiency is the key

¹ Pós-graduada em Sistemas Embarcados no SENAI Anchieta. E-mail: juliana.capucho@senaisp.edu.br.

² Pós-graduado em Sistemas Embarcados no SENAI Anchieta. E-mail: victor.olmo@senaisp.edu.br.

³ Professor Dr. na Faculdade de Tecnologia SENAI São Paulo – Campus “Anchieta”. E-mail: leandro.poloni@sp.senai.br.

⁴ Professor Me. na Faculdade de Tecnologia SENAI São Paulo – Campus “Anchieta”. E-mail: marcones.silva@sp.senai.br.

⁵ Professor Especialista na Faculdade de Tecnologia SENAI São Paulo – Campus “Anchieta”. E-mail: luis.canno@sp.senai.br.

⁶ Professor Me. na Faculdade de Tecnologia SENAI São Paulo – Campus “Anchieta”. E-mail: fernando.simplicio@sp.senai.br.

to maximizing the battery lifetime of Internet of Things (IoT) network devices. Among the new technologies of IoT networks, Long Range (LoRa) stands out as a transmission technology whose main characteristic is energy-efficient transmission over long distances. An inherent challenge to LoRa's radio energy consumption issue is to determine the most appropriate set of transmission parameters and the Adaptive Data Rate (ADR) mechanism can be employed as a solution. In this regard, this work experimentally evaluated the ADR mechanism proposed by Bor & Roedig (ADR BR) and compared its performance with the mechanism proposed by Long Range Wide Area Network (LoRaWAN), based on the number of frames, number of iterations, and the energy consumption of the set of parameters at the end of the execution. The experiments were conducted using the LoRa SX1276 radio, in an indoor urban environment. The project used only endpoint nodes forming a point-to-point network between source and sink nodes, without the use of a gateway device. The main results indicated that the ADR BR algorithm determined a set of parameters with consumption up to approximately 91% lower when compared to the configuration determined through the LoRaWAN algorithm for the same scenario. It was noticed a reduction of approximately 1/3 of the total number of packets needed for the algorithm to determine the final set using the early termination mechanism.

Keywords: LoRa; LoRaWAN; ADR; power consumption.

1 INTRODUÇÃO

Dispositivos sem fio permitem uma ampla variedade de aplicações coletando informações de fenômenos ou eventos através de sensores, ou executando tarefas por atuadores, e podem ser implantados em ambientes que demandam baixa manutenção e de difícil acesso às fontes de energia. Nesse contexto, a eficiência energética é requisito chave para maximizar o tempo de vida dos dispositivos das redes de *Internet of Things* (IoT), geralmente alimentados por bateria e necessitam operar por longos períodos de forma confiável (BOUGUERA et al., 2018).

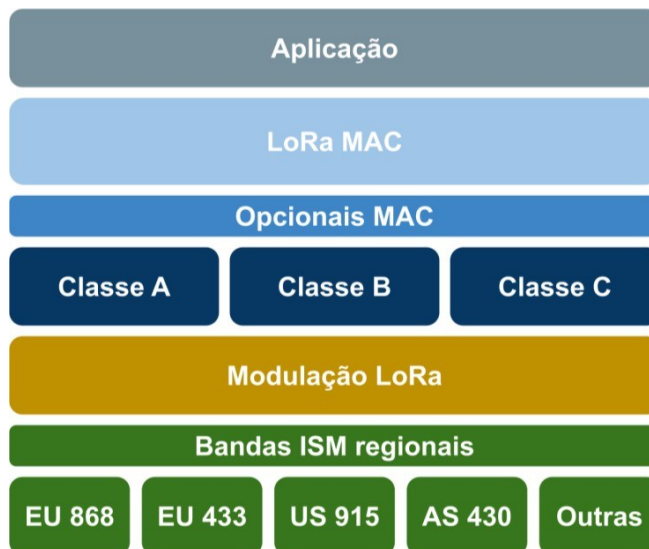
As redes *Low-Power Wide-Area Network* (LPWAN) são redes de telecomunicações sem fio de longo alcance, baixo custo e maior eficiência energética para a conexão dos dispositivos IoT. O panorama das tecnologias LPWAN abrange uma diversidade de tecnologias concorrentes, como *Sigfox*, *Long Range* (LoRa), *Weightless* e a *3rd Generation Partnership Project* (3GPP), originando *Long-Term Evolution Machine* (LTE-M) e *Narrow Band-IoT* (NB-IoT) (PETÄJÄRVI et al., 2017). Em redes em malha ou síncronas, como as redes celulares, os nós frequentemente precisam sair do modo de economia para sincronizar com a rede e verificar a existência de mensagens. Essa sincronização consome energia de forma significativa e é o principal fator na redução da vida útil da bateria, o que torna um limitante em rede LPWAN. Já os dispositivos em uma rede *Long Range Wide Area Network* (LoRaWAN) são assíncronos e se comunicam quando têm dados prontos para enviar (LoRa® ALLIANCE, 2015).

Vários sistemas sem fio usam a modulação *Frequency Shifting Keying* (FSK) em sua camada física, pois é muito eficiente para atingir baixo consumo. As redes LoRa, por sua vez, utilizam a modulação por *Chirp Spread Spectrum* (CSS), a qual mantém as características de baixa potência como a FSK, mas aumenta significativamente o alcance de comunicação. A CSS tem sido utilizada em comunicação militar e espacial por décadas devido à longa distância de comunicação alcançada e robustez à interferência (LoRa® ALLIANCE, 2015).

LoRaWAN define o protocolo de comunicação e arquitetura de sistema para a rede na camada *Medium Access Control* (MAC), enquanto a camada física LoRa possibilita o *link* de

comunicação de longa distância. O LoRaWAN define três classes distintas de dispositivos para otimizar os diferentes perfis de aplicações, atendendo atuadores que necessitam de baixa latência e sensores de baixo consumo de energia. O protocolo LoRaWAN fica logo abaixo da aplicação e é responsável por determinar como um dispositivo deve se comunicar com o outro, já o protocolo LoRa define como a informação será transmitida no meio físico. E, por fim, a camada de rádio frequência (RF) define as frequências que devem ser utilizadas em cada região, conforme ilustrado na Figura 1 (LoRa® ALLIANCE, 2015).

Figura 1 – Pilha de protocolos de dispositivos LoRa.



Fonte: Adaptado de LoRa® Alliance (2015).

Um dispositivo LoRa possui diferentes parâmetros de rádio que podem ser configurados a fim de otimizar o desempenho da rede e o consumo de energia dos dispositivos, são eles: *Transmission Power* (TP), *Carrier Frequency* (CF), *Spreading Factor* (SF), *Bandwidth* (BW) e *Coding Rate* (CR). Os diferentes parâmetros de transmissão do rádio LoRa podem ser configurados de forma a otimizar a taxa de dados, minimizando o consumo de energia dos dispositivos de acordo com a condição atual do canal de comunicação para cenários estáticos.

Essas diferentes configurações de transmissão resultam em mais de 6720 possibilidades de conjuntos de parâmetros em que um dispositivo LoRa pode ser configurado (BOR & ROEDIG, 2017). Portanto, é um desafio determinar qual o conjunto de parâmetros de transmissão mais apropriado dentre todas as possibilidades que a camada física propicia, visto que o canal varia com o tempo. E de modo a contornar a problemática, o mecanismo *Adaptive Data Rate* (ADR) pode ser empregado para determinar o conjunto mais adequado para cada dispositivo com base na qualidade do canal e potência de transmissão, ampliando a capacidade da rede (KUFKUNESU et al., 2020).

Neste sentido, o objetivo principal deste trabalho foi realizar a análise experimental do mecanismo ADR proposto por Bor & Roedig (2017), cuja premissa visa avaliar, a cada iteração, um conjunto de parâmetros que diminua o consumo de energia pela metade, comparar seu desempenho com o mecanismo proposto pela LoRaWAN e propor melhorias com base nos resultados. Os algoritmos foram analisados e comparados com base no número de *frames* para estimar a qualidade do *link*, número de iterações necessárias e o consumo do conjunto de parâmetros ao final da execução.

E para alcançar os objetivos propostos, os experimentos foram conduzidos em ambiente urbano, empregando-se rádio LoRa SX1276 e a ferramenta da Semtech “*LoRa Modem Calculator Tool*”.

O mecanismo ADR possibilita aos rádios com tecnologia LoRa ajustarem o conjunto de parâmetros de transmissão de forma a se adaptarem ao canal de comunicação em cenários com dispositivos estáticos, nos quais o canal tem variação inexpressiva no tempo, justificando a importância do desenvolvimento de estudos atrelados à análise de algoritmos ADR.

2 DESENVOLVIMENTO

Um importante desafio a ser superado no âmbito de dispositivos LoRa é que sua especificação não define a maneira em que o *Network Server* (NS) deve instruir os nós da rede a realizar a adaptação dos parâmetros de transmissão, sendo que muitas vezes a implementação fica a cargo do provedor dos equipamentos que mantêm seus algoritmos de maneira privada. O *framework Long Range Wide Area Network* (LoRaWAN), por sua vez, apresenta uma recomendação de implementação do algoritmo de decisão, porém, considera somente os parâmetros TP e SF, limitando o número de conjunto de parâmetros que os dispositivos podem ser configurados com relação ao que o LoRa possibilita. Essa característica pode ser um limitante para aplicações que necessitam operar com consumo extremamente baixo de energia (KUFKUNESU et al., 2020).

A potência de transmissão do rádio LoRa pode ser ajustada para operar nos valores de -4 até 20 dBm, porém, devido à limitação na implementação do *hardware*, tais valores são geralmente na faixa de 2 até 20 dBm. A frequência central da portadora utilizada para transmitir a mensagem modulada pode ser configurada na faixa de 137 até 1020 MHz (limitado em 860 MHz, dependendo do *hardware*), em passos de 61 Hz. O fator de espalhamento é definido como a relação entre a taxa de símbolo e a taxa de *chirp*, sendo configurável com valores de 6 até 12, sendo a taxa de *chirps* por símbolo definida como $2 * SF$ (BOR & ROEDIG, 2017).

A largura de banda que determina a frequência total utilizada para a transmissão é configurável com os valores de 500, 250 e 125 kHz, os dados são transmitidos com uma taxa de *chirp* que coincide com a largura de banda. Já a taxa de código utilizada pelo algoritmo de código de correção de erros é a relação da quantidade de bits que carregam informação pela quantidade total de bits, a qual pode ser configurada para um dos valores: 4/5, 4/6, 4/7 ou 4/8 (BOR & ROEDIG, 2017).

O consumo de energia da modulação CSS no LoRa é independente do conteúdo transmitido por ela, como ocorre em outras modulações como a *On-Off- Keying* (OOK) e, portanto, dependente apenas de um parâmetro chamado de *Time onAir* (ToA) e do TP. A potência de transmissão influencia diretamente na relação *Signal to Noise* (SNR) do sinal recebido e interferência com os outros nós. O cálculo do ToA é especificado pelo fabricante e depende dos parâmetros BW, SF e CR (BÄUMKER et al., 2019).

Uma largura de banda maior aumenta a taxa de dados (ToA menor), porém, reduz a sensibilidade do receptor devido a maior incidência de ruído, reduzindo o alcance da transmissão. O aumento do valor do SF em uma unidade faz com que o valor do ToA seja dobrado e a taxa de dados diminuída pela metade, aumentando também o consumo de energia, entretanto, com a vantagem de melhorar a sensibilidade do receptor, aumentando a SNR e a distância em que o sinal pode ser transmitido. O código corretor de erros permite corrigir parte dos erros ocorridos na transmissão, mas o aumento da CR incrementa também o ToA e, por consequência, o consumo dos dispositivos. O LoRa possui ainda alguns fatores

constantes que aumentam o ToA, como o preâmbulo, cabeçalho e *Cyclic Redundancy Check* (CRC) (BOR & ROEDIG, 2017; BÄUMKER et al., 2019).

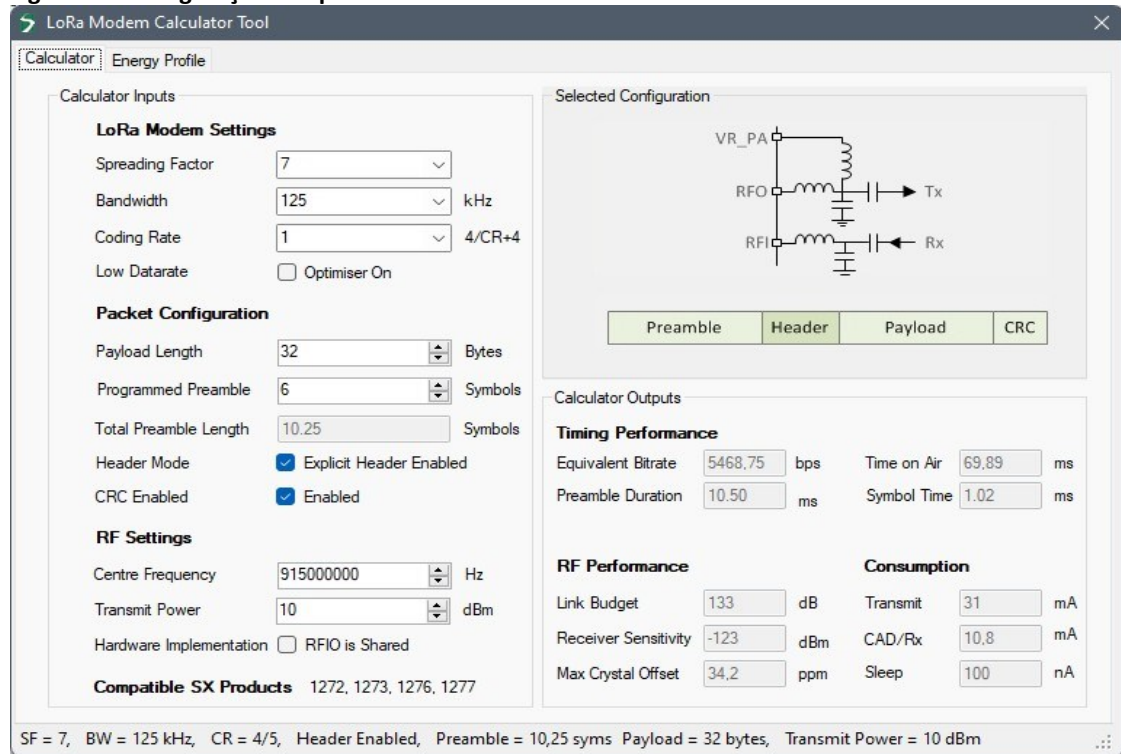
3 METODOLOGIA

Para determinar o consumo de energia de cada uma das possíveis configurações do rádio LoRa SX1276, utilizou-se a ferramenta da *Semtech Corporation* “*LoRa Modem Calculator Tool*” (SEMTECH, 2013b), na qual é possível obter uma estimativa da corrente média consumida pelo dispositivo de acordo com o conjunto de parâmetros de interesse. Os valores da ferramenta devem ser utilizados apenas como um indicativo de consumo, o que é suficiente já que o intuito do trabalho foi explorar os mecanismos ADR e não determinar o consumo exato das configurações.

A ferramenta permite configurar outros parâmetros de transmissão que foram mantidos fixos durante os testes: tamanho da carga útil de 32 *bytes*, preâmbulo programado de 6 símbolos, cabeçalho explícito na mensagem, verificação cíclica de redundância (*Cyclic Redundancy Check* - CRC) habilitada, frequência central de 915 MHz, 1 byte para a mensagem de ACK, *link* bidirecional e transmissor periódico. A Figura 2 demonstra a interface da ferramenta para configuração dos parâmetros.

Na interface da ferramenta (Figura 2) também foram configurados os parâmetros de transmissão, variados e escolhidos pelo mecanismo ADR. O fator de espalhamento variando de 7 a 12, largura de banda podendo ser 125, 250 ou 500 MHz, taxa de codificação entre 4/5 e 4/8 e potência de transmissão entre 2 e 17 dBm, totalizando 1152 combinações possíveis para o rádio SX1276.

Figura 2 – Configuração dos parâmetros na interface da “LoRa Modem Calculator Tool”.



Fonte: Semtech (2013a).

O consumo do rádio para cada um dos parâmetros configurados é apresentado na aba *Energy Profile* dentro da seção *Energy per Day*. A comparação entre cada conjunto de configurações foi realizada considerando o valor de corrente média em μA .

3.1 Topologia

A topologia utilizada para os testes consistiu em dois tipos de nós: o *source* que é responsável por prover informação de forma periódica e possui restrições de consumo; e o *sync* que fica aguardando as mensagens dos nós do tipo *source* e não possui restrições de alimentação. O *sync* é responsável também por definir o novo conjunto de parâmetros a ser utilizado pelos nós com rádio LoRa, de acordo com os algoritmos ADR escolhidos para comparação.

O projeto utilizou apenas nós do tipo *endpoint* formando uma rede ponto a ponto entre o *source* e o *sync*, sem o uso de equipamentos do tipo *gateway* que permitem comunicar em mais de um canal e com diversos parâmetros de rádio ao mesmo tempo, sem restrições de consumo de energia e com maior potência, visto que esses dispositivos estão sempre conectados a uma fonte de energia.

Dado que os *endpoints* podem se comunicar com apenas um conjunto de parâmetros por vez, torna-se necessário manter o sincronismo dos parâmetros utilizados entre os nós, especialmente o fator de espalhamento e a largura de banda, de modo que os rádios não fiquem com parâmetros diferentes, impossibilitando a comunicação entre eles.

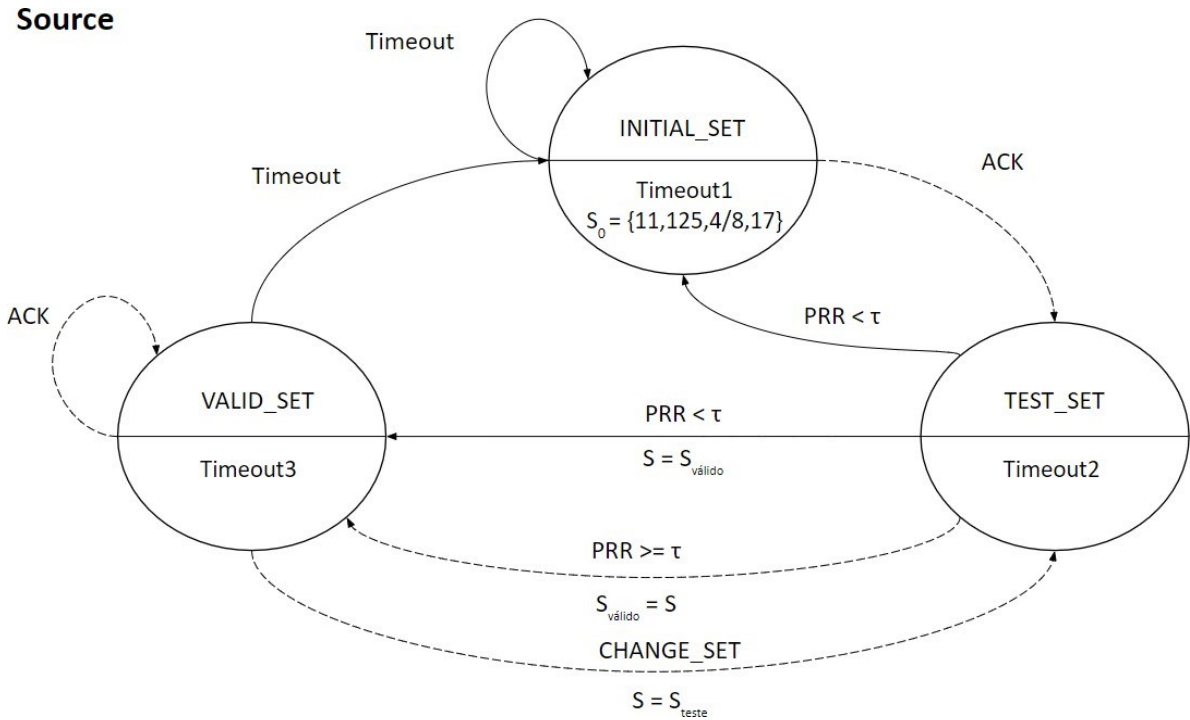
Destaca-se que essa limitação trouxe a necessidade de implementar um conjunto de regras para a transição do conjunto de parâmetros nos dispositivos *source* e *sync*, de forma que os dispositivos pudessem voltar a uma configuração válida à comunicação, após realizar a troca para um conjunto de parâmetros que impossibilitasse a comunicação entre eles no

cenário atual.

Portanto, foram criados três tipos de comandos que são utilizados na comunicação entre os nós: PING, ACK e CHANGE_SET. O comando PING é utilizado pelo nó *source* em todos os pacotes enviados, com os quais também é encaminhado um número sequencial utilizado para diferenciar cada pacote. O comando ACK é enviado pelo nó *sync* como confirmação do recebimento de um comando PING. Para iniciar a troca de parâmetros, o nó *sync* envia um comando do tipo CHANGE_SET juntamente com o novo conjunto de parâmetros determinado pelo algoritmo ADR.

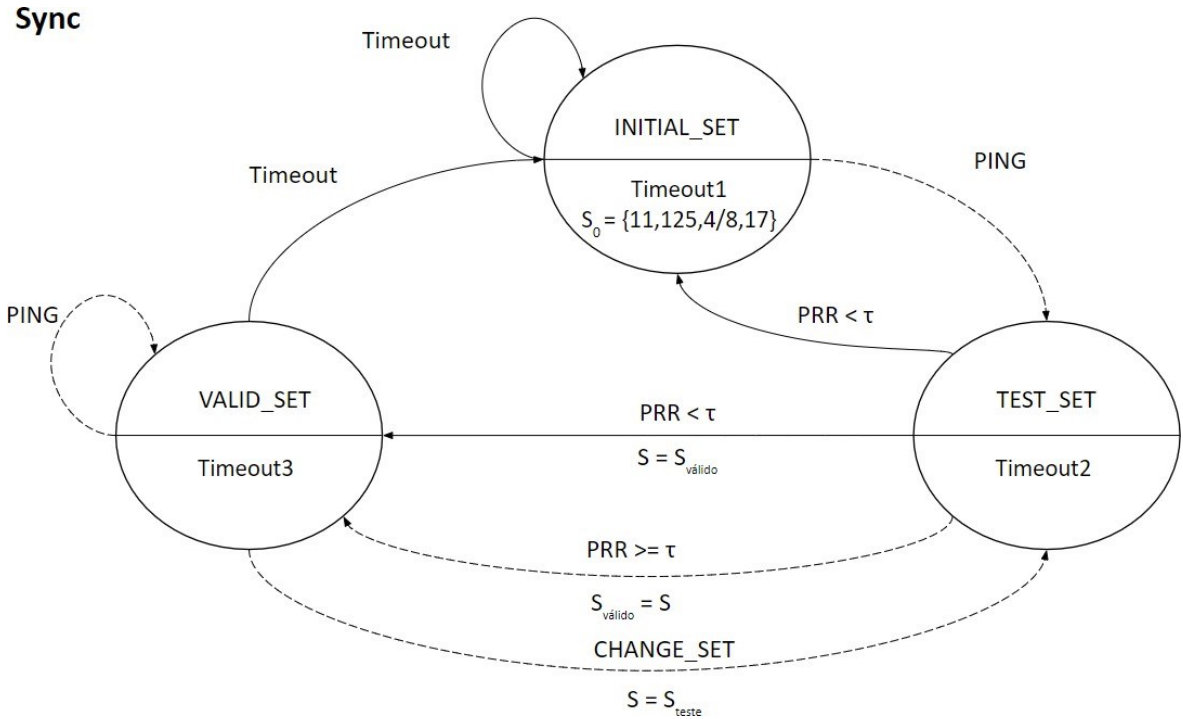
O envio e recebimento desses comandos servem como eventos para a máquina de estados que determinam qual novo estado o nó deve executar. Foram definidos três estados possíveis para os dispositivos com suas próprias ações para cada tipo de nó: INITIAL_SET, TEST_SET e VALID_SET. As Figuras 3 e 4 apresentam os diagramas de estados para o nó do tipo *source* e *sync*, respectivamente, com os eventos que definem a troca de cada estado. E na Tabela 1 estão descritos os estados e os eventos da máquina de estados para os nós do tipo *source* e *sync*.

Figura 3 – Máquina de estados para o nó do tipo *source*.



Fonte: Elaborado pelos autores.

Figura 4 – Máquina de estados para o nó do tipo sync.



Fonte: Elaborado pelos autores.

Tabela 1 – Descritivo dos estados para os nós do tipo source e sync.

Estado do nó	Descrição
INITIAL_SET	<p>Source. Estado inicial ao ligar o dispositivo, ao entrar nesse estado o dispositivo é configurado com um conjunto de parâmetros fixo que garanta a comunicação inicial entre os nós, sendo o conjunto $S_0 = \{SF11, BW125, CR4/8, 17 \text{ dBm}\}$ de consumo $C(S_0) = 977 \mu\text{A}$ escolhido. O nó transmite periodicamente pacotes com o comando PING até que sejam transmitidos corretamente três pacotes consecutivos. A transmissão é confirmada através do recebimento de um pacote com o comando ACK contendo o mesmo número que foi enviado no comando PING. Após o envio consecutivo dos três pacotes, o dispositivo passa ao estado TEST_SET.</p> <hr/> <p>Sync. Da mesma forma que ocorre para o nó do tipo <i>source</i>, esse é o estado inicial para o nó <i>sync</i> ao ligar o dispositivo com o conjunto de parâmetros inicial $S_0 = \{SF11, BW125, 4/8, 17 \text{ dBm}\}$ de consumo $C(S_0) = 977 \mu\text{A}$. O nó aguarda o recebimento de pacotes com o comando PING dentro da janela de recebimento até que sejam recebidos corretamente três pacotes consecutivos e, nesse momento, o dispositivo muda de estado para TEST_SET.</p>

TEST_SET

Source. Nesse estado é determinado se o conjunto de parâmetros em teste é um conjunto válido ou inválido. Para isso são transmitidos vinte pacotes consecutivos como comando PING para o cálculo da taxa de recepção de pacotes, *Packet Reception Rate* (PRR). Se a PRR for menor que o limite r configurado como 0.8, o conjunto de parâmetros é considerado inválido, o nó é configurado com o último conjunto de parâmetros válido e passa para o estado VALID_SET. Um caso especial é quando o conjunto de parâmetros inválido é o conjunto inicial e, nesse cenário, o dispositivo volta ao estado inicial uma vez que não existe nenhum outro conjunto válido para retornar. No caso de a PRR possuir valor maior ou igual ao limite, o conjunto de parâmetros é considerado válido e o dispositivo passa ao estado VALID_SET com o conjunto atual de parâmetros.

Sync. O dispositivo *sync* no estado TEST_SET aguarda o recebimento de vinte pacotes para o cálculo da PRR. Caso a PRR seja menor que o limite r , o conjunto de parâmetros é invalidado, alterando os parâmetros de transmissão para o último estado válido e passando para o estado VALID_SET (caso o conjunto de parâmetros não seja o inicial). Se a PRR for maior ou igual ao limite r , o estado é alterado para VALID_SET, validando o conjunto de parâmetros atual.

VALID_SET

Source. Ao entrar no estado VALID_SET o nó *source* envia um pacote com comando PING e aguarda a resposta. Caso a resposta seja um pacote com o comando ACK, o dispositivo envia um novo comando PING repetindo esse procedimento de forma cíclica. Caso seja recebido um comando CHANGE_SET, o dispositivo altera seus parâmetros de transmissão para os valores recebidos juntamente com o comando e passa ao estado TEST_SET. Se o comando PING falhar por três vezes consecutivas, o dispositivo volta ao estado inicial. Destaca-se que esse comportamento foi necessário para o cenário em que a transmissão venha a falhar no estado com um conjunto de parâmetros que anteriormente era válido.

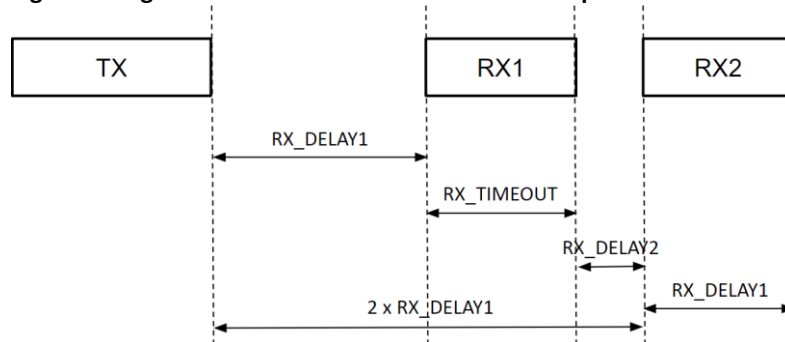
Sync. No estado VALID_SET o nó *sync* aguarda o recebimento de um comando PING que, ao recebê-lo, utiliza o algoritmo ADR selecionado para calcular o novo conjunto de parâmetros a ser validado. O nó *sync* envia o comando CHANGE_SET para o nó *source* juntamente com o novo conjunto de parâmetros que será validado no estado seguinte, TEST_SET. Caso o algoritmo tenha finalizado ao atingir o número máximo de iterações ou ao encontrar o conjunto de parâmetros final para o *link* de comunicação atual, o nó *sync* envia um comando ACK e o dispositivo fica aguardando um novo comando PING. Se porventura o dispositivo não receber nenhum pacote com comando PING por três janelas consecutivas, o nó volta ao estado inicial de forma análoga ao nó *source*.

Fonte: Elaborado pelos autores.

3.1.1 Agendamento de transmissão

O agendamento da transmissão para o nó *source* foi baseado na implementação dos dispositivos LoRaWAN de classe A. Após o nó *source* realizar a transmissão do pacote com comando PING, o dispositivo aguarda por um intervalo RX_DELAY1 em modo ocioso de baixo consumo até iniciar a janela de recepção pelo intervalo RX_TIMEOUT. O dispositivo entra novamente em modo ocioso pelo intervalo RX_DELAY2 até completar o intervalo de duas vezes o tempo RX_DELAY1 desde o fim da transmissão para, então, abrir uma nova janela de recepção pelo intervalo RX_TIMEOUT. A Figura 5 apresenta os intervalos de cada etapa do agendamento de transmissão para o nó *source*. O dispositivo fica em estado ocioso após a segunda janela de recepção até a transmissão de um novo pacote.

Figura 5 – Agendamento de transmissão do nó do tipo source.



Fonte: Elaborado pelos autores.

Para a realização do experimento foram escolhidos os seguintes valores para os intervalos descritos anteriormente: $RX_DELAY1 = 3$ s, $RX_TIMEOUT = 2$ s e $RX_DELAY2 = 1$ s. O nó do tipo *sync* mantém a janela de recepção aberta pelo intervalo total de 8 s até que um pacote seja recebido. Caso nenhum pacote seja recebido neste intervalo, o pacote é considerado perdido e uma nova janela de recebimento é aberta.

3.1.2 ADR LoRaWAN

O primeiro algoritmo ADR implementado e utilizado como referência para as comparações foi o mecanismo recomendado pela *Semtech Corporation* para o protocolo LoRaWAN (SEMTECH, 2016; LI et al., 2018), o qual foi projetado apenas para dispositivos estáticos, uma vez que o canal de transmissão muda drasticamente a cada pacote para dispositivos móveis.

O nó do tipo *sync* armazena o valor de SNR de cada novo pacote recebido, armazenando sempre o valor de SNR dos últimos vinte pacotes. Ao atingir o número mínimo de pacotes recebidos, o dispositivo *sync* calcula os parâmetros necessários para determinar o novo conjunto de parâmetros que será utilizado para a transmissão entre os dispositivos. O parâmetro SNR_{max} é o maior valor de SNR recebido nos últimos vinte pacotes. O parâmetro SNR_{req} é o valor de SNR requerido para a transmissão de acordo com o valor do fator de espalhamento do último pacote recebido, conforme indicado na Tabela 2.

Tabela 2 – Valor de SNR requerido (SNR_{req}) por fator de espalhamento.

Fator de espalhamento	12	11	10	9	8	7
SNR_{req} (dB)	-20,0	-17,5	-15,0	-12,5	-10,0	-7,5

Fonte: Elaborado pelos autores.

A Equação 1 apresenta o cálculo do parâmetro SNR_{marg} , onde $Marg_{DB}$ é um valor constante de margem de instalação da rede específico por dispositivo, sendo o valor padrão 5 dB recomendado e utilizado nos experimentos deste trabalho.

$$SNR_{marg} = SNR_{max} - SNR_{req} - Marg_{DB} \quad (1)$$

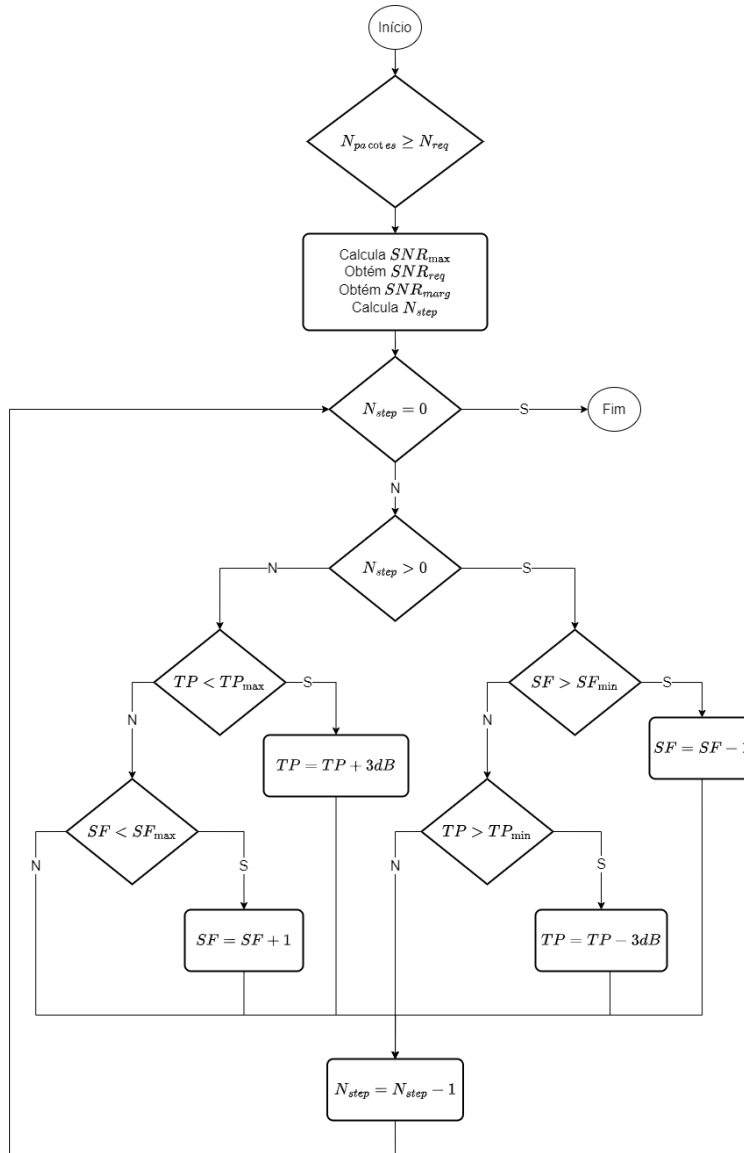
O cálculo do parâmetro N_{step} é realizado através da Equação 2, onde $int()$ é a parte inteira do resultado da divisão:

$$N_{step} = int\left(\frac{SNR_{marg}}{3}\right) \quad (2)$$

O valor de N_{step} indica o número de iterações utilizado pelo algoritmo ADR LoRaWAN para determinar o novo conjunto de parâmetros, de acordo com o fluxograma da Figura 6. Quando o valor de N_{step} é maior que zero, a qualidade atual do *link* de transmissão é boa, indicando que o dispositivo pode aumentar a taxa de transmissão (diminuir o fator de espalhamento) e, portanto, diminuir a potência de transmissão para reduzir o consumo de energia. Caso o valor de N_{step} seja menor que zero, a qualidade atual do *link* de transmissão é insatisfatória, de modo que os dispositivos devem aumentar a potência de transmissão e diminuir a taxa de transmissão (aumentar o fator de espalhamento) para tentar melhorar a qualidade da transmissão. O algoritmo termina quando o valor de N_{step} é igual a zero e, nesse caso, o novo conjunto de parâmetros é determinado e enviado para o nó *source*.

Os testes com os algoritmos ADR foram realizados em ambiente urbano interno, em um prédio residencial separados por uma distância de aproximadamente 12 m, sendo o dispositivo *sync* alocado no terceiro andar (dentro de um apartamento) e o dispositivo *source*, em um estacionamento no segundo subsolo. Os resultados foram coletados em cinco dias distintos em horários próximos, estimando o consumo a cada pacote recebido até que a execução do algoritmo fosse encerrada ao atingir o valor ótimo.

Figura 6 – Fluxograma algoritmo ADR LoRaWAN.



Fonte: Elaborado pelos autores.

3.2 ADR Bor & Roedig

O segundo algoritmo adaptativo implementado foi o algoritmo apresentado por Bor & Roedig no artigo “LoRa Transmission Parameter Selection” (BOR & ROEDIG, 2017), referenciado por “ADR BR”, o qual considera todas as combinações de parâmetros disponíveis para configurar o rádio LoRa, no caso as 1152 configurações para o hardware escolhido, enquanto o algoritmo ADR do LoRaWAN permite selecionar um subconjunto de apenas seis configurações de potência de transmissão e seis configurações de fator de espalhamento (BOR & ROEDIG, 2017).

O fluxograma da Figura 7 apresenta os passos do algoritmo ADR BR, onde é utilizado o valor de corrente média consumida pelo rádio LoRa para determinar o novo conjunto de parâmetros de transmissão a ser validado pelos nós.

O primeiro passo é inicializar os arranjos e conjuntos de parâmetros utilizados pelo

algoritmo: o arranjo U que armazena os conjuntos de parâmetros que já foram utilizados é inicializado como vazio; o conjunto de teste S_t é inicializado com o conjunto de parâmetros iniciais S_i ; o conjunto de parâmetros válido S_v é inicializado com os parâmetros do conjunto de teste S_t ; e o arranjo de conjuntos de parâmetros possíveis P é inicializado com o arranjo de todos os possíveis conjuntos de parâmetros S .

O passo seguinte é adicionar o conjunto S_t ao arranjo U para que este não seja utilizado novamente. Após a transmissão de vinte pacotes, verifica-se o valor de PRR para o conjunto S_t , caso o valor seja maior ou igual ao limite r , o conjunto S_t é considerado válido e atribuído ao conjunto S_v . A corrente média desejada c para o próximo conjunto de parâmetros a ser validado é calculada como sendo metade da corrente média do conjunto S_v . Caso a PRR seja menor que o limite r , o conjunto S_v permanece o mesmo e a variável c é calculada como sendo metade da soma da corrente média de S_v e S_t .

O arranjo P dos conjuntos de parâmetros possíveis é determinado como sendo os conjuntos de parâmetros que ainda não foram utilizados (todos os conjuntos que pertencem a S e que não pertencem a U) com corrente média menor que c . Caso o arranjo P seja vazio indicando que todos os conjuntos de parâmetros possíveis com corrente média menor que c já foram validados, o algoritmo termina atribuindo o último conjunto válido S_v ao conjunto de parâmetros final S_f .

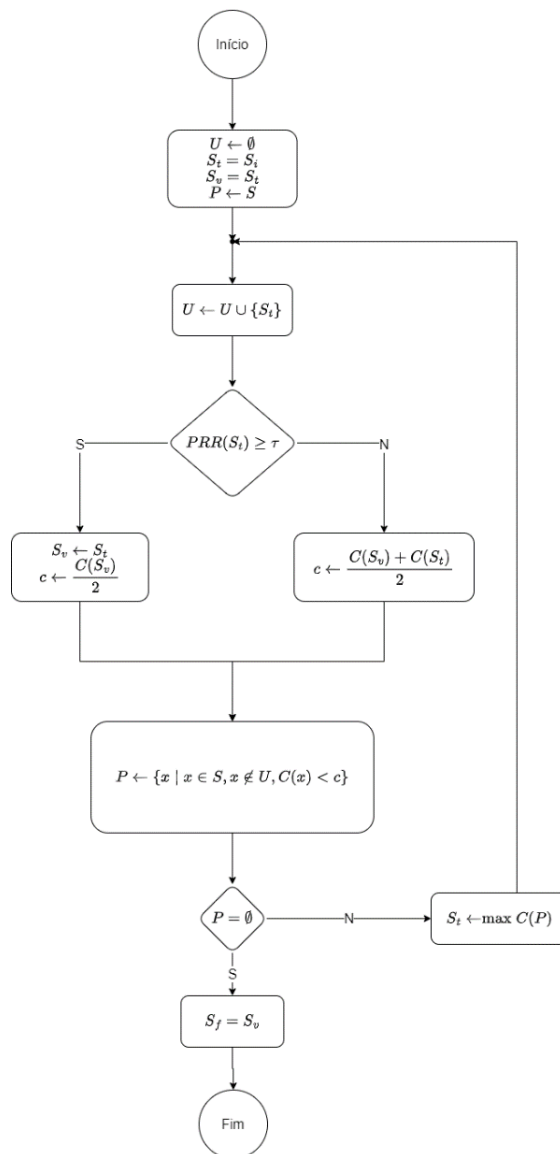
Caso o arranjo P não seja vazio, o próximo conjunto de parâmetros S_t é determinado como sendo o conjunto de maior corrente média dentro do arranjo P (todos os conjuntos de parâmetros não utilizados e que possuem corrente média menor que o limiar). O algoritmo retorna então ao segundo passo onde o novo S_t é adicionado ao arranjo U . O processo se repete até que o arranjo P seja vazio ou que o número máximo de iterações pré-determinado seja atingido.

O experimento foi conduzido da mesma forma que o ADR LoRaWAN, isto é, em cinco dias distintos em horários próximos, estimando o consumo a cada pacote recebido até que a execução do algoritmo e suas variações fosse encerrada ao atingido valor ótimo ou o número máximo de quinze iterações.

3.2.1 ADR BR – *Early Termination*

Com o intuito de reduzir o número de pacotes transmitidos por iteração, o algoritmo ADR BR sugere o uso de um mecanismo de *early termination* para encerrar iteração atual logo que a PRR atingir um valor menor que o limite r ou quando o número de pacotes recebidos corretamente for igual a $r \cdot 20$, de modo que se sabe que a PRR não pode mais ser menor que r .

Figura 7 – Fluxograma algoritmo de seleção de parâmetros ADR BR.



Fonte: Elaborado pelos autores.

3.2.2 ADR BR – RSSI

Outra adição ao algoritmo é a validação da potência do sinal recebido pelo nó *sync* através da *Received Signal Strength Indication* (RSSI). Ao receber um pacote com valor de RSSI superior a um limiar $\sigma = -105$ dBm, o nó encerra a iteração imediatamente considerando o conjunto de parâmetros como válido e envia essa informação juntamente com o comando ACK.

3.2.3 ADR BR – SNR

O último experimento consistiu em uma proposta de melhoria ao ADR BR a fim de diminuir o número de iterações iniciais necessárias para atingir mais rapidamente os conjuntos de parâmetros com menor consumo quando o *link* de comunicação entre os dispositivos apresenta boa qualidade.

Com base no algoritmo ADR LoRaWAN o nó *sync* armazena o valor de SNR dos pacotes

recebidos e ao final de cada iteração calcula o valor de SNR médio dos pacotes recebidos para o conjunto de parâmetros em teste. O valor médio é então comparado ao valor SNR_{req} da Tabela 2 para determinar qual o menor valor possível para o fator de espalhamento do próximo conjunto de parâmetros a ser testado para o *link* de comunicação atual. Reduziu-se também o número máximo de iterações do algoritmo para dez, uma vez que o algoritmo possibilita encontrar conjuntos com menor consumo mais rapidamente.

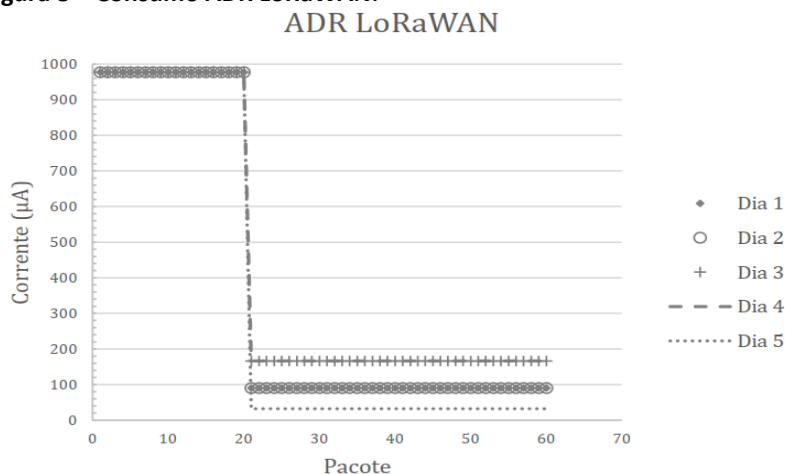
4 RESULTADOS E DISCUSSÕES

Nas subseções a seguir, são apresentados e discutidos os resultados obtidos nos testes realizados.

4.1 ADR LoRaWAN

A Figura 8 demonstra o gráfico do consumo de corrente média para os dispositivos LoRa determinado por meio do conjunto de parâmetros utilizados. Os dispositivos iniciam com o conjunto de parâmetros iniciais $S_0 = \{SF11, BW125, CR4/8, 17 \text{ dBm}\}$ e corrente média de 977 μA , após vinte pacotes o dispositivo *sync* executa o algoritmo ADR e determina o novo conjunto de parâmetros no dia de pior resultado como sendo $S_1 = \{SF8, BW125, CR4/8, 17 \text{ dBm}\}$ com $C(S_1) = 166,46 \mu\text{A}$. E para o dia de melhor resultado como sendo $S_1 = \{SF7, BW125, CR4/8, 17 \text{ dBm}\}$ com $C(S_1) = 90,65 \mu\text{A}$. O novo conjunto de parâmetros é enviado para o dispositivo *source* e os dispositivos vão para o estado *TESTE_SET* onde mais vinte pacotes são enviados para validar o conjunto de parâmetros com $PRR = 1$. O conjunto de parâmetros é o final e os dispositivos permanecem no estado *VALID_SET*.

Figura 8 – Consumo ADR LoRaWAN.

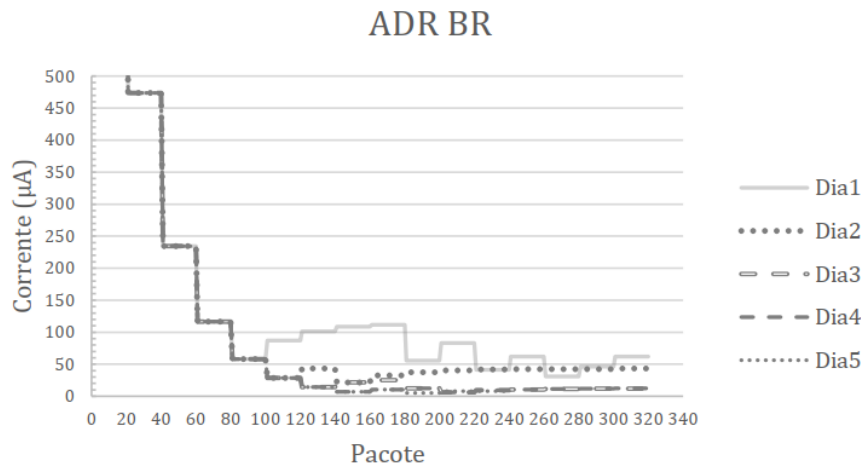


Fonte: Elaborado pelos autores.

4.2 ADR Bor & Roedig

O resultado do experimento do ADR BR é apresentado na Figura 9, onde os valores do gráfico foram concentrados na região na qual o consumo fica entre 0 e 500 μA dado que após a primeira iteração o consumo cai para esse intervalo, facilitando a visualização das curvas para os valores menores nas últimas iterações.

Figura 9 – Consumo ADR BR.



Fonte: Elaborado pelos autores.

A primeira iteração inicia com o conjunto inicial S_0 de consumo $C(S_0) = 977 \mu\text{A}$. Para o dia com o pior resultado o algoritmo finaliza ao atingir o número máximo de iterações, sendo o último conjunto de parâmetros válido determinado na décima segunda iteração com o conjunto $S_{12} = \{\text{SF9, BW125, CR4/5, 9 dBm}\}$ e consumo $C(S_{12}) = 62,14 \mu\text{A}$. Para o dia com o melhor resultado o algoritmo finaliza na décima iteração após não encontrar mais nenhum conjunto de parâmetros possível sendo $S_{10} = \{\text{SF7, BW500, CR4/5, 14 dBm}\}$ com $C(S_{10}) = 7,79 \mu\text{A}$.

Os dados apresentam uma redução de aproximadamente 62% no pior resultado, e de aproximadamente 91% para o melhor resultado comparando o consumo médio de corrente da configuração obtida através do ADR LoRaWAN com o ADR BR. Em contrapartida, nota-se que para o pior caso, após a décima segunda iteração, o algoritmo de seleção de parâmetros não encontra mais nenhum conjunto de parâmetros válidos com corrente média menor que $C(S_{12})$, desperdiçando energia na transmissão até atingir o limite máximo de iterações.

4.3 ADR BR – Early Termination

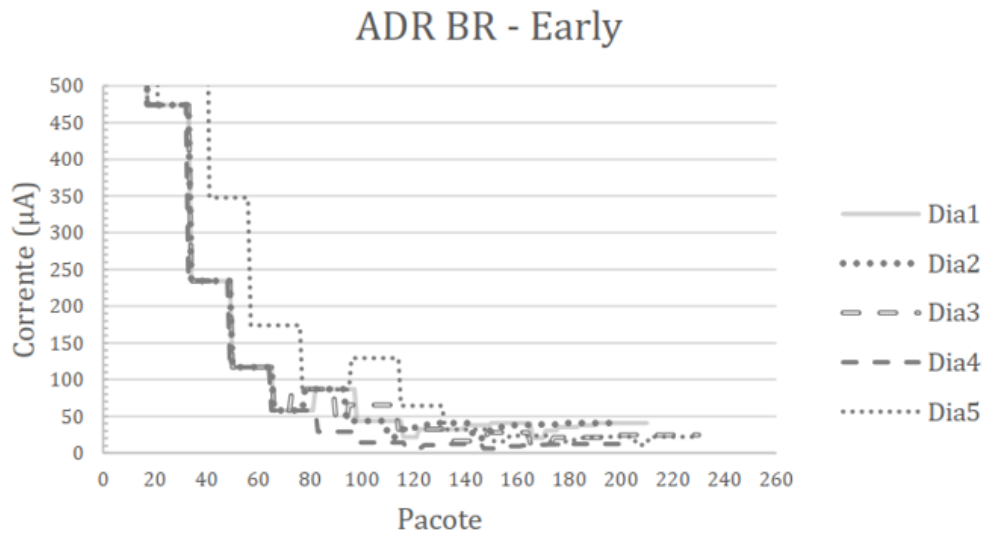
Os resultados para os cinco dias de testes para o algoritmo ADR BR com o mecanismo *early termination* encontram-se na Figura 10. O conjunto de parâmetros final determinado para o melhor resultado foi $S_9 = \{\text{SF7, BW250, CR4/5, 13 dBm}\}$ com consumo $C(S_9) = 12,34 \mu\text{A}$.

Nota-se que o algoritmo finaliza com um número consideravelmente menor de pacotes transmitidos quando o mecanismo *early termination* é utilizado, reduzindo o número total de pacotes em aproximadamente 1/3. A diminuição de pacotes é especialmente útil pois permite reduzir o número de pacotes desperdiçados ao utilizar um conjunto de parâmetros inválido já que são necessários apenas cinco pacotes para detectar o conjunto inadequado.

4.4 ADR BR – RSSI

Para o experimento de validação da potência do sinal recebido pelo nó *sync* através de RSSI, observou-se que para o ambiente onde os testes foram realizados, o nível do sinal recebido estava sempre abaixo do limiar σ , de forma que não houve vantagem em utilizar esse mecanismo, onde o número de pacotes necessários para finalizar o algoritmo (Figura 11) foi similar ao obtido no resultado anterior (Figura 10).

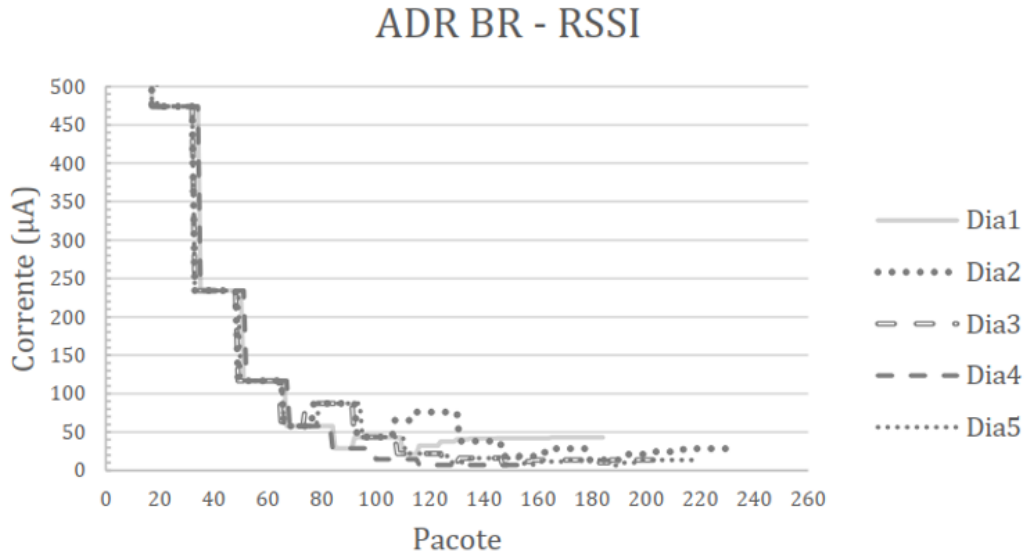
Figura 10 – Consumo ADR BR + Early.



Fonte: Elaborado pelos autores.

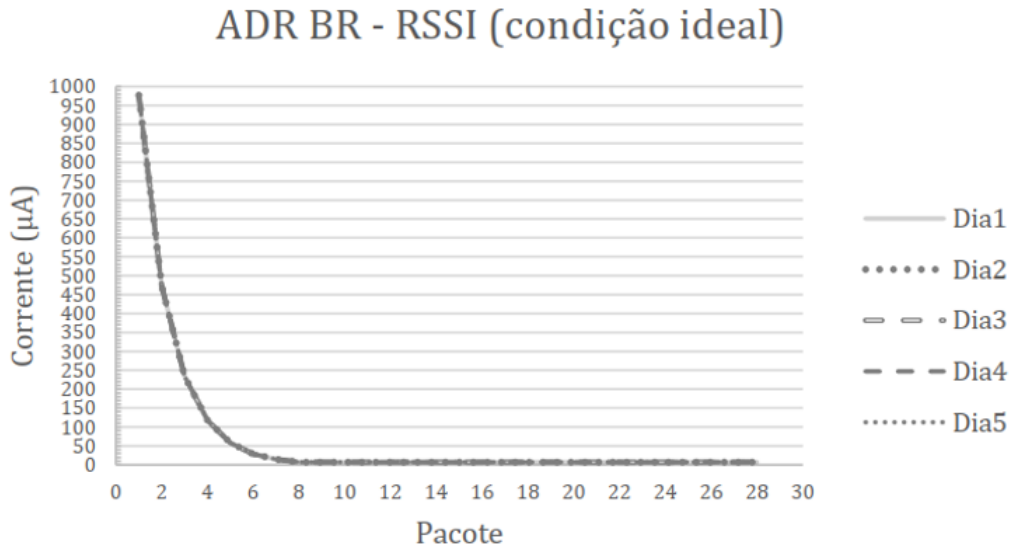
Com o intuito de validar o mecanismo em outro cenário, o teste foi repetido posicionando os dois nós no mesmo andar e mantendo uma distância de poucos metros um do outro. Nessa condição ideal todos os pacotes recebidos possuem valor de RSSI $> \sigma$, de forma que a cada iteração apenas um pacote é necessário para validar o conjunto de parâmetros e todo conjunto verificado é válido para o *link* de comunicação. Logo, com poucos pacotes o dispositivo é capaz de determinar o conjunto ideal para minimizar o consumo de energia, conforme mostra a curva desse experimento na Figura 12.

Figura 11 – Consumo ADR BR + RSSI.



Fonte: Elaborado pelos autores.

Figura 12 – Consumo ADR BR + RSSI (condição ideal).

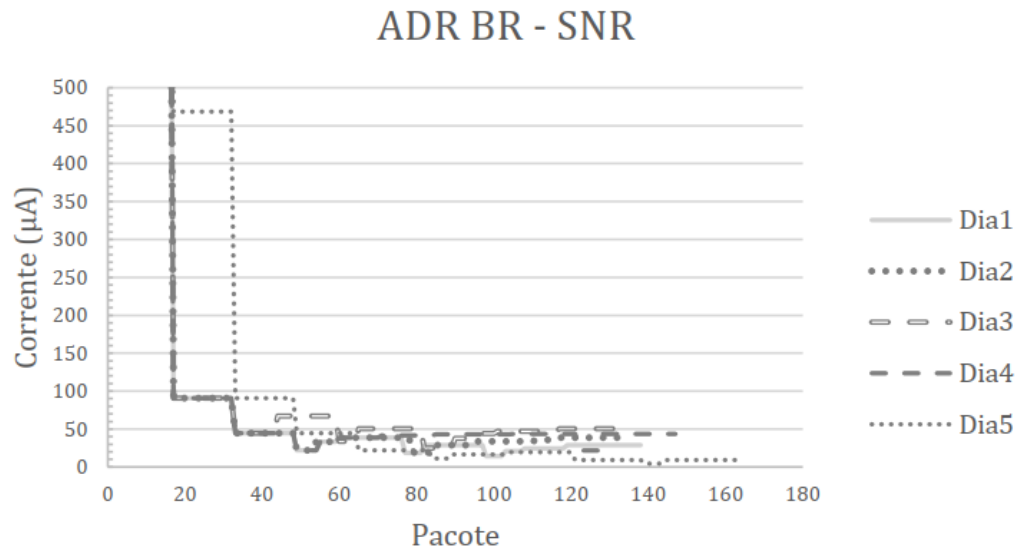


Fonte: Elaborado pelos autores.

4.5 ADR BR – SNR

A Figura 13 apresenta os resultados obtidos para os testes relacionados ao uso do valor SNR para reduzir o número de iterações necessárias para atingir mais rapidamente os conjuntos de parâmetros com menor consumo, considerando-se cinco dias distintos conforme realizado em todos os experimentos. Destaca-se que em apenas duas iterações o algoritmo foi capaz de determinar um conjunto de parâmetros com consumo inferior a 50 µA.

Figura 13 – Consumo ADR BR + SNR.



Fonte: Elaborado pelos autores.

5 CONCLUSÃO

A partir dos testes propostos e desenvolvidos no trabalho, para o caso do ADRBR, obteve-se um conjunto de parâmetros que minimiza o consumo de energia para o rádio LoRa, mantendo a qualidade da transmissão, porém, com o custo de ser necessário um número maior de iterações e, por consequência, um número maior de pacotes transmitidos quando comparado ao LoRaWAN cujo algoritmo busca ajustar a potência de transmissão e o fator de espalhamento de acordo com a condição do canal para melhorar a confiabilidade do *link* ou diminuir o consumo quando a qualidade do canal é boa, necessitando de um número menor de pacotes, de modo a determinar o conjunto de parâmetros mais adequado sem que um conjunto que impossibilita a comunicação seja selecionado.

Apesar de o algoritmo apresentar uma forma conveniente de evitar o uso de conjuntos inválidos e de ser capaz de determinar um conjunto de parâmetros com consumo consideravelmente menor que o LoRaWAN, o ADR BR não faz uso da qualidade do canal para tomar a decisão de qual o conjunto de parâmetros a ser avaliado na iteração seguinte. Com o uso da SNR para determinar os valores possíveis de SF para o *link* de comunicação, pôde-se determinar com maior assertividade um conjunto de parâmetros com menor consumo em poucas iterações, de modo que o número de pacotes transmitidos com alto consumo nas primeiras iterações foi reduzido.

Considerando os mecanismos que podem auxiliar na diminuição do número de pacotes para validar cada conjunto de parâmetros, reduzindo, assim, os pacotes não aproveitáveis, destaca-se o emprego do *early termination*, entretanto, este mecanismo promove maior complexidade na implementação do sincronismo da comunicação ponto a ponto e sua utilização deve ser avaliada de acordo com a robustez e as exigências técnicas de cada projeto.

E para o experimento de validação da potência do sinal recebido pelo nó *sync* através de RSSI, o número de pacotes necessários para finalizar o algoritmo foi similar ao obtido para o algoritmo ADR BR com o mecanismo *early termination*. Logo, sua utilização no ambiente urbano proposto neste trabalho não foi atrativa, porém, sugere-se uma avaliação do mecanismo RSSI em cenários distintos para validar sua aplicabilidade no âmbito do algoritmo ADR BR.

Estudos da implementação de algoritmos com vistas à redução do consumo de energia devem ser encorajados, visto que há uma consonância global para a proposição de projetos de eficiência energética de dispositivos de comunicação e de acesso à rede.

REFERÊNCIAS

- BÄUMKER, E.; GARCIA, A.; WOIAS, P. Minimizing power consumption of LoRa and LoRaWAN for low-power wireless sensor nodes. **Journal of Physics: Conference Series**, v. 1407, id 012092, 2019. Disponível em: <https://iopscience.iop.org/article/10.1088/1742-6596/1407/1/012092/pdf> . Acesso em: 20 mai. 2023.
- BOR, M.; ROEDIG, U. LoRa transmission parameter selection. *In*: INTERNATIONAL CONFERENCE ON DISTRIBUTED COMPUTING IN SENSOR SYSTEMS (DCOSS), 13., 2017, Ottawa, Canada. **Proceedings** [...]. RosewoodDrive, Danvers, MA: The Institute of Electrical and Electronics Engineers, 2017. Disponível em: https://eprints.lancs.ac.uk/id/eprint/85515/4/lora_tps_r1342.pdf . Acesso em: 20 mai. 2023.
- BOUGUERA, T. et al. Energy consumption model for sensor nodes based on LoRa and LoRaWAN. **Sensors**, v. 18, n. 7, ID. 2104, 2018. Disponível em: <https://www.mdpi.com/1424-8220/18/7/2104/pdf> . Acesso em: 20 mai. 2023.
- KUFAKUNESU, R.; HANCKE, G. P.; ABU-MAHFOUZ, A. M. A survey on adaptive data rate optimization in LoRaWAN: Recent solutions and major challenges. **Sensors**, v. 20, n. 18, id 5044, 2020. Disponível em: <https://www.mdpi.com/1424-8220/20/18/5044/pdf>. Acesso em: 20 mai. 2023.
- LI, S.; RAZA, U.; KAHN, A. How agile is the adaptive data rate mechanism of LoRaWAN? *In*: IEEE Global Communications Conference (GLOBECOM), 2018, Abu Dhabi, United Arab Emirates. **Proceedings** [...]. (Electronic ISSN: 2576-6813). RosewoodDrive, Danvers, MA: The Institute of Electrical and Electronics Engineers, 2018. Disponível em: <https://arxiv.org/pdf/1808.09286> . Acesso em: 20 mai. 2023.
- LoRa® ALLIANCE. LoRa® Alliance Technical Marketing Workgroup. A technical overview of LoRa and LoRaWAN. [San Ramon, CA]: LoRa® Alliance, 2015. Disponível em: <https://loro-alliance.org/wp-content/uploads/2020/11/what-is-lorawan.pdf> . Acesso: 10 nov. 2022.
- PETÄJÄJÄRVI, J. et al. Evaluation of LoRa LPWAN technology for indoor remote health and wellbeing monitoring. **International Journal of Wireless Information Networks**, v. 24, p. 153–165, 2017. Disponível em: <http://jultika.oulu.fi/files/nbnfi-fe201704076006.pdf>. Acesso em: 20 mai. 2023.
- SEMTECH. SX1272/3/6/7/8: LoRa Modem – Low Energy Consumption Design – AN1200.17.

[Camarillo, CA]: Semtech, 2013a. Disponível em: <https://www.semtech.com/products/wireless-rf/lora-connect/sx1272>. Acesso em: 20 maio 2023.

SEMTECH. SX1272/3/6/7/8: LoRa Modem – Designer’sGuide – AN1200.17. [Camarillo, CA]: Semtech, 2013b. Disponível em: <https://www.semtech.com/products/wireless-rf/lora-connect/sx1272>. Acesso em: 20 mai. 2023.

SEMTECH. LoRaWAN – Simple rateadaptation recommended algorithm. [S.I.]: Semtech, 2016. Disponível em <https://www.thethingsnetwork.org/forum/uploads/default/original/2X/7/7480e044aa93a54a910dab8ef0adfb5f515d14a1.pdf> . Acesso em: 10 nov. 2022.

SOBRE OS AUTORES

i JULIANA CAPUCHO PEREIRA



Bacharel em Ciência e Tecnologia (2014) e Engenheira de Informação (2019) pela Universidade Federal do ABC (UFABC). Atualmente é colaboradora da empresa Turing Inc., desenvolvendo atividades nos seguintes temas: qualidade de software, microsserviços e automação. Possui experiência em desenvolvimento e qualidade de software, já atuou em testes de interface de usuário do sistema Android, consumo de bateria de smartphones e rede móvel.

ii VICTOR HUGO DE CARVALHO OLMO



Bacharel em Ciência e Tecnologia (2014) e Engenheiro de Informação (2019) pela Universidade Federal do ABC (UFABC). Atualmente é colaborador da empresa Instituto de Pesquisas Eldorado. Possui experiência em desenvolvimento de sistemas embarcados com conectividade móvel (2G, 3G, 4G), TCP/IP, WiFi, IMS, VoLTE, VoWiFi e suporte nos processos de homologação de novas tecnologias.

iii LEANDRO POLONI DANTAS



Engenheiro (2004) e Doutor (2018) em Engenharia Elétrica pelo Centro Universitário FEI. Atuou por 15 anos na indústria eletrônica no desenvolvimento de novos produtos. Desde 2009, vem lecionando em cursos de pós-graduação, graduação e de nível técnico em diferentes instituições paulistanas. Atualmente é professor na Faculdade de Tecnologia SENAI e no Insper. <https://orcid.org/0000-0003-3674-336X>

iv **MARCONES CLEBER BRITO DA SILVA**



Tecnólogo em Mecatrônica Industrial (2011), Engenheiro Mecatrônico (2013) e Especialista em Engenharia de Manutenção Industrial pela Centro universitário Eniac (2013). Mestre em Tecnologia Nuclear (2020) pela Universidade de São Paulo. Desde 2011, vem lecionando em cursos de nível técnicos e de graduação. Atualmente é professor da Faculdade de Tecnologia SENAI e na FESA. <https://orcid.org/0000-0002-3690-1682>

v **LUIZ CARLOS CANNO**



Graduado em Tecnologia de Automação Industrial (2009) com Especialização em Gestão Empresarial pela Universidade Nove de Julho (2012), e Especialização em Docência na Educação Profissional e Tecnológica pelo SENAI CETIQT (2015). Professor na Faculdade de Tecnologia SENAI do curso de Tecnologia em Eletrônica Industrial e Pós-graduação em Sistemas Embarcados. <https://orcid.org/0000-0001-9331-9309>

vi **FERNANDO SIMPLICIO DE SOUSA**



Professor da Faculdade SENAI no curso de Pós-Graduação em Sistemas Embarcados. Mestre em Engenharia Elétrica pela Universidade Federal do ABC (UFABC) e Pós-Graduado (Lato Sensu) pela Universidade Mackenzie. Graduado em Gestão de Pequenas e Médias Empresas pela UNIP e em Projetos Mecânicos pela Faculdade de Tecnologia de São Paulo (UNESP/FATEC-SP). <https://orcid.org/0009-0009-5760-4845>