

Um Algoritmo de Gerenciamento de Retransmissão em Nível de Aplicação para redes LoRaWAN

João P. Pastório¹, Álvaro R. Castro e Souza¹, Edson T. de Camargo¹

¹ Curso de Engenharia de Computação

Universidade Tecnológica Federal do Paraná (UTFPR) – Toledo, PR – Brasil

joaopastorio@alunos.utfpr.edu.br, {alvarosouza, edson}@utfpr.edu.br

Resumo. A Internet das Coisas (IoT), conceito relacionado à conexão de objetos com a Internet, permite desenvolver diferentes tipos de aplicações em diversas áreas. A tecnologia sem fio LoRa e seu protocolo LoRaWAN formam uma importante infraestrutura de comunicação para aplicações IoT que requerem longo alcance e baixo consumo energético, como sensores localizados em áreas urbanas e rurais. No entanto, o padrão LoRa está sujeito a interferências em seu sinal. Dessa forma, não há qualquer garantia de que os dados transmitidos pelos dispositivos finais IoT serão de fato recebidos pelos pontos de acesso. Esse trabalho propõe uma abordagem em nível de aplicação para gerenciar a retransmissão de pacotes. Em experimentos com envio de 150 pacotes de dados se obteve 100% de sucesso no recebimento dos mesmos após as retransmissões.

Abstract. The Internet of Things (IoT), a concept related to the connection of objects to the Internet, allows the development of different types of applications in different areas. LoRa wireless technology and its LoRaWAN protocol form an important communication infrastructure for IoT applications that require long range and low power consumption, such as sensors located in urban and rural areas. However, the LoRa standard is subject to interference in its signal. Therefore, there is no guarantee that the data transmitted by IoT end devices will actually be received by the gateway. This work proposes an application-level approach to managing packet retransmission. In experiments with sending 150 data packets was achieved 100% success in receiving them after retransmissions.

1. Introdução

A partir dos avanços em diversas áreas de tecnologia, dispositivos eletrônicos que se comunicam com a Internet passaram a estar cada vez mais presentes em nosso cotidiano. Uma das principais características desses dispositivos é a sua capacidade de trocar informações, o que só foi possível após os avanços das tecnologias de informação e comunicação [Khanna and Kaur 2019]. O conceito de Internet das Coisas ou *Internet of Things* (IoT) é utilizado para definir objetos conectados à Internet, capazes de gerar dados úteis e serem representados no mundo virtual [Al-Fuqaha et al. 2015]. A IoT é uma área promissora, com muita pesquisa e desenvolvimento tanto na academia quanto na indústria. Conforme estimado por [Framingham 2019] em 2025 haverá mais de 41 bilhões de dispositivos inteligentes em operação ao redor do mundo.

Os dispositivos finais IoT podem utilizar diferentes infraestruturas de redes sem fio para se comunicar a Internet. Um tipo de rede muito utilizada por sensores é a *Low*

Power Wide Area Network (LPWAN), que tem como característica seu longo alcance - em determinados cenários podendo chegar a um raio de cobertura de até 50 km -, e baixo consumo energético [Mekki et al. 2017]. Criada pela empresa Semtech, a modulação LoRa, derivação para *Long-Range*, é uma das tecnologias LPWAN que, geralmente, utiliza o protocolo de enlace chamado LoRaWAN em sua comunicação. A tecnologia LoRa utiliza uma faixa de frequência não licenciada, o que possibilita a implantação de uma rede de baixo custo, flexível e em grande escala, sendo assim a opção adotada como rede de comunicação por várias cidades [Rossato et al. 2020].

Atualmente, estão sendo desenvolvidas diversas aplicações que utilizam a rede LoRaWAN para comunicação como, por exemplo, na agricultura de precisão, realizando análises de temperatura e umidade do solo [Singh et al. 2020]. Ou ainda no controle de sistemas de irrigação inteligentes [Fraga-Lamas et al. 2020]. Em ambientes urbanos a rede serve de apoio para detecção de níveis de material particulado e qualidade do ar [Johnston et al. 2019], bem como no rastreamento em tempo real de objetos em movimento [Camargo et al. 2021]. Para algumas aplicações, como no monitoramento do nível de gases tóxicos em áreas residenciais, torna-se fundamental a garantia de que os dados dos dispositivos finais sejam entregues aos servidores de rede. Entretanto, um enlace sem fio sofre o impacto de diferentes níveis de obstruções presentes em ambientes urbanos, o que faz com que pacotes de dados possam ser perdidos durante a transmissão [Ferreira et al. 2020].

O protocolo LoRaWAN não atua sobre os pacotes de dados perdidos, mas permite que o desenvolvedor da aplicação utilize mensagens de confirmação do recebimento dos dados [LoRa Alliance 2017]. Passa então a ser papel das aplicações implementar um algoritmo de retransmissão para lidar com casos de instabilidade na rede e perda de pacotes. Entretanto, como mencionado por [Capuzzo et al. 2018], devem ser tomadas medidas de precaução quanto a utilização das mensagens de confirmação e retransmissão de sinal, uma vez que aumentar a transmissão de dados no meio de propagação impacta na performance da rede, podendo levar a congestionamentos e ao aumento na colisão dos pacotes.

O trabalho de [Shanmuga Sundaram et al. 2020] propõe uma política de retransmissão dinâmica pela variação do temporizador de retransmissão das aplicações. O objetivo é reduzir o número de retransmissões desnecessárias pelo atraso no recebimento da mensagem de confirmação. Já [Gomez and Crowcroft 2020] apresentam um algoritmo de retransmissão para redes LPWAN focando em um tempo limite de retransmissão adaptável ao *Round Trip Time* (RTT). O RTT é o tempo entre o envio de dados do emissor e o recebimento da mensagem de confirmação transmitida pelo receptor. Por sua vez o algoritmo CR-LoRaWAN, proposto por [Almeida et al. 2021], realiza o gerenciamento de retransmissões nos dispositivos finais em uma rede LoRaWAN. Isso é realizado através de mensagens de confirmação do tipo ACK presente no próprio protocolo. Destaca-se que o CR-LoRaWAN deixa a cargo do dispositivo final o gerenciamento das retransmissões e não garante retransmitir 100% dos pacotes perdidos.

O objetivo deste trabalho consiste em desenvolver um algoritmo de retransmissão para redes LoRaWAN em nível de aplicação. Uma das vantagens da abordagem proposta é diminuir a sobrecarga no dispositivo final em lidar com os pacotes perdidos, diminuindo assim o processamento e o consumo energético empregado com a gerencia da retransmissão. O algoritmo proposto gerencia as retransmissões pela comunicação bidirecional

entre dispositivo final e servidor de aplicação. Nesta proposta o servidor de aplicação passa a ser responsável por gerenciar todo o tráfego de mensagens enviadas pelos dispositivos finais. Para realizar tal gerenciamento tem-se a utilização do número de sequência presente no protocolo LoRaWAN. Resultados preliminares realizados em um ambiente real destacam que o algoritmo foi capaz de retransmitir com sucesso todos os pacotes perdidos.

Este artigo está organizado da seguinte forma. A Seção 2 revisa brevemente os padrões LoRa e LoRaWAN e descreve os trabalhos relacionados. A Seção 3 descreve o algoritmo de retransmissão proposto e a metodologia de avaliação. A Seção 4 sumariza e discute os resultados obtidos. A Seção 5 apresenta a conclusão e trabalhos futuros.

2. LoRa, LoRaWAN e a Retransmissão de Pacotes

Concebida pela Semtech, o protocolo LoRa, abreviação para *Long Range*, opera na camada física da comunicação entre os dispositivos IoT, sendo essa uma tecnologia de modulação de rádio frequência desenvolvida para redes LPWAN [Semtech 2019]. Dessa forma, os dispositivos que a utilizam para comunicação conseguem trocar informações em uma ampla área de abrangência, e apresentam um baixo consumo energético, estando associados à baixa transferência de dados - na ordem de milhares de bits por segundo.

Os dispositivos baseados em LoRa convertem seus dados digitais em sinais de rádio frequência utilizando a técnica *Chirp Spred Spectrum* (CSS). A técnica de modulação CSS consiste na utilização do máximo da largura de banda disponível para transmissão do sinal por meio da alteração em frequência, semelhante ao *Frequency-shift Keying* (FSK). A diferença do CSS para o FSK está na forma de modulação baseada em frequências, enquanto o FSK utiliza duas ou mais frequências de modo discreto, uma para cada símbolo, o CSS varia continuamente entre as frequências, com pulsos de *Upchirp*, alteração de uma frequência alta para baixa ou *Downchirp*, mudança de uma frequência baixa para alta [Seneviratne 2019].

Na Figura 1 é apresentada a arquitetura das redes LoRa. A arquitetura segue uma topologia de rede em estrela, onde os nós são os dispositivos finais, compostos por sensores e atuadores interagindo com o ambiente, enquanto o ponto central, que gerencia o redirecionamento das mensagens, é equivalente aos pontos de acesso ou *Gateways*. O processo de comunicação nessas redes é bidirecional, podendo haver transmissão de pacotes dos dispositivos finais para os pontos de acesso ou dos pontos de acesso para os dispositivos finais [Semtech 2019].

Os dispositivos finais, presentes nas redes LoRa, enviam seus dados para os *Gateways* por meio do enlace sem fio utilizando a modulação CSS. Esses dados, após serem recebidos por pelo menos um ponto de acesso, são direcionados para o servidor de rede relativo à aplicação que o dispositivo final está registrado, por meio de uma conexão IP [Semtech 2019]. Como os *Gateways* agem apenas como encaminhadores, os servidores de rede são responsáveis por gerenciar toda a comunicação nas redes LoRaWAN, lidando com a segurança dos dados e garantia de entrega. Então os servidores de rede encaminham os dados recebidos para serem consumidos pelos servidores de aplicação, o que é realizado por meio das chaves de acesso fornecidas pelo dispositivo [Alliance 2017]. Nas aplicações são realizados todos os tratamentos nos dados enviados pelos dispositivos, podendo também ocorrer envios de mensagens em resposta a esses dados. Todo o

Fonte: O Autor

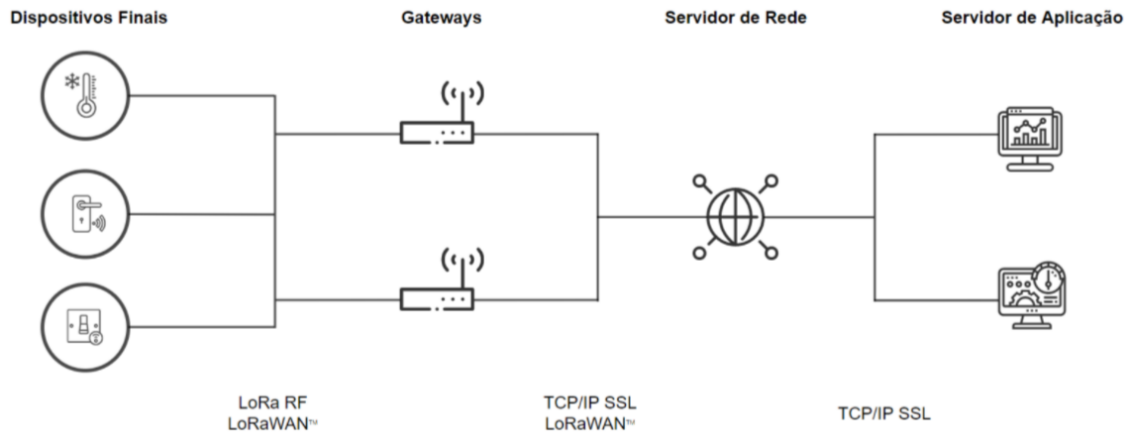


Figura 1. Arquitetura de redes LoRaWAN

processo de comunicação que parte do dispositivo final para as aplicação é referenciado como *Uplink* [Semtech 2019].

Como os dispositivos finais em uma rede LoRa permanecem a maior parte do tempo desativados para uma maior economia de bateria, o protocolo LoRaWAN dispõe três modos de operação com o objetivo de coordenar a comunicação entre dispositivos finais e servidores [Lavric and Popa 2017]: as classes A, B e C.

A classe A foi projetada para atingir o maior desempenho energético dos dispositivos. Nessa classe o dispositivo final permanece desativado até precisar enviar alguma informação para a rede, como mostra a Figura 2. Assim que a informação é completamente encaminhada o mesmo aguarda um certo tempo para abrir a primeira janela de recebimento de dados, onde, em sincronia com o servidor de rede, pode receber mensagens de *Downlink*. Caso nenhuma mensagem seja recebida pelo dispositivo em sua primeira janela de recepção, o mesmo aguarda um segundo intervalo e abre uma segunda oportunidade de recebimento de dados. Após as duas tentativas, o dispositivo é desativado até precisar enviar um novo pacote de dados [Semtech 2019].

Fonte: O Autor

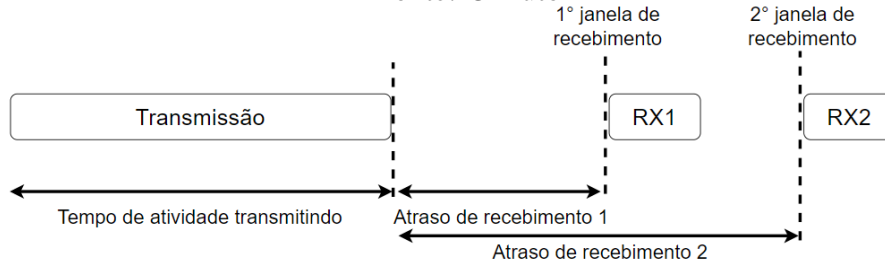


Figura 2. Comunicação Classe A

Diferente da classe A, a classe B utiliza mensagens *Beacon* para sincronizar com o servidor intervalos periódicos de abertura de janela de recepção. Por necessitar de ativações periódicas, a utilização dessa classe implica em um maior consumo energético. Já os dispositivos de classe C permanecem a todo o momento ativos, com uma janela de recepção aberta. Esse caso é utilizado para aplicações que independem de eficiência

energética [Semtech 2019].

2.1. Mecanismo de Retransmissão

Como mencionado por [Almeida et al. 2021], o protocolo LoRaWAN não fornece um mecanismo para gerenciamento de retransmissões e tolerância a falhas durante a comunicação, apresentando apenas a possibilidade de se solicitar mensagens de confirmação *Acknowledgement* (ACK) em cada pacote de dados enviado. Com o ACK é possível saber apenas se o pacote de dados atual foi recebido pelo ponto de acesso em caso de *Uplink*, ou pelo dispositivo final em caso de *Downlink*.

Como documentado em [Alliance 2017], no protocolo LoRaWAN as mensagens trocadas entre dispositivo final e ponto de acesso apresentam em sua estrutura de cabeçalho um número de sequência *Frame Counter* (Fcnt). O Fcnt é um número incremental para mensagens *Downlink* e *Uplink*, sendo utilizado como identificação particular de cada mensagem. Os números de sequência são muito utilizados em protocolos de redes clássicos, como no protocolo TCP, onde o mesmo serve para gerenciar a troca de mensagem entre emissor e receptor. No protocolo TCP, o número é utilizado para ordenar as mensagens no destinatário, uma vez que os pacotes de dados enviados podem assumir diferentes caminhos durante a sua propagação pela rede [Forouzan 2009].

O gerenciamento dos números de sequência *Downlink* pode ser realizado pelo dispositivo final. Já as mensagens *Uplink* são gerenciadas pelos servidores de rede e aplicação. Neste trabalho, as mensagens *Downlink* são empregadas para informar ao dispositivo final se pacotes foram perdidos durante a comunicação.

2.2. Trabalhos Relacionados

Relacionado ao tema de retransmissão em redes LoRaWAN, o trabalho de [Capuzzo et al. 2018] fornece uma visão do impacto da utilização de mensagens de confirmação, e consequentemente retransmissões, por dispositivos finais. Os resultados do estudo foram obtidos a partir da utilização do simulador de código aberto NS-3. O cenário simulado consistia em um *Gateway* atuando como receptor de diferentes números de dispositivos finais, esses com capacidade de enviar mensagens com e sem confirmação, assim como retransmissões quando os pacotes com confirmação forem perdidos. Para as interferências do meio de propagação o estudo utilizou perdas por distância logarítmica, sombreamento e construções.

Os resultados da pesquisa realizada por [Capuzzo et al. 2018] mostram que o aumento no número de dispositivos finais utilizando mensagens de confirmação impacta negativamente o desempenho da rede. Com isso, expõem o risco de se utilizar mensagem de confirmação excessivamente. Para contornar tal problema, algumas soluções são elencadas pelos autores, como a utilização de estratégias para adaptar parâmetros do sistema e o aumento no número de *Gateways* para balancear a carga de mensagens necessárias e o tempo que os mesmos passam transmitindo.

Já o trabalho de [Marais et al. 2020] analisa diferentes cenários que utilizam mensagem de confirmação. O objetivo é verificar a viabilidade da sua utilização e propor, com base em demais pesquisas científicas no segmento, alternativas para o melhor desempenho da rede quando essas mensagens forem necessárias. Uma das propostas é a utilização da estratégia de retransmissão por *back-off*, como abordado em [Alliance 2017]. Nessa

estratégia é utilizado um intervalo de tempo aleatório para a tentativa de retransmissão dos dispositivos finais, evitando que ocorra transmissões simultâneas de mensagens de confirmação e sobrecarga dos *Gateways*. Nessa estratégia, caso a retransmissão não seja realizada, passa a ser papel do servidor de rede gerenciar a retransmissão.

Outra abordagem, realizada por [Abdelfadeel et al. 2020] é a utilização de um esquema de agendamento. Nesse caso os dados são armazenados no dispositivo final e apenas transmitidos de forma agrupada em um mesmo *payload*, enviando assim pacotes de dados maiores, porém com uma menor frequência. Com essa ação as mensagens de ACK são válidas para todo o grupo de dados enviado, reduzindo o número de mensagens de confirmação solicitadas pelos dispositivos finais.

Em [Gomez and Crowcroft 2020] outro aspecto é levado em consideração ao lidar com retransmissão em redes LPWAN. Esse trabalho propõe a utilização de tempo limite de retransmissão (RTO) de forma multimodal, considerando diferentes cenários para o *Round Trip Time* (RTT), que calcula o tempo entre ida e volta de uma mensagem entre o emissor e receptor. Nos resultados do trabalho são levantadas as diferenças entre a utilização de um algoritmo padrão RTO e de sua versão multimodal. Verifica-se ainda como a utilização multimodal reduz a latência das mensagens de confirmação e a entrega de pacotes em momentos de perda de sinal.

Como concluído no trabalho de [Pastório et al. 2021], apesar do acréscimo no número de *Gateways* auxiliar na entrega de pacotes em redes LoRa, as obstruções presentes no meio sem fio impactam diretamente na redução da porcentagem de pacotes recebidos, restringindo a confiabilidade da entrega. Surge com essa limitação a necessidade de se utilizar outros mecanismos para auxiliar a entrega de mensagens em redes LoRa, como pelo emprego de algoritmos para gerenciamento de retransmissões.

Em [Almeida et al. 2021] é proposto um algoritmo de retransmissão para redes LoRaWAN, o CR-LoRaWAN. No algoritmo as mensagens de confirmação são interpretadas para grupos de cinco pacotes. Dessa forma, caso o quinto pacote enviado com a mensagem não seja recebido, o dispositivo final retransmite todos os cinco pacotes de dados referenciados pela mesma. O mesmo ocorre quando a mensagem de confirmação é recebida, podendo ocorrer a interpretação errônea de pacotes de dados como confirmados, quando os mesmos podem ter sofrido alguma interferência e não chegaram a ser recebidos pelos pontos de acesso. O CR-LoRaWAN ainda utiliza mensagens ACK para gerenciamento de retransmissão. Como levantado por [Capuzzo et al. 2018] a utilização excessiva dessa abordagem pode afetar negativamente o desempenho de uma rede LoRaWAN.

3. Materiais e Métodos

Esta seção apresenta a proposta do algoritmo para gerenciamento de retransmissão. Primeiramente, os materiais utilizados, como dispositivos, bibliotecas e protocolos são apresentados. Em seguida a abordagem proposta é descrita, bem como sua implementação no dispositivo final e servidor de aplicação.

3.1. Materiais Utilizados

O algoritmo para gerenciamento de retransmissão proposto consiste em uma comunicação bidirecional entre dispositivo final e servidor de aplicação, como mostra a Figura 3.

Utiliza-se como dispositivo final, durante a obtenção de resultados, a placa de desenvolvimento TTGO TBeam. A placa TTGO TBeam apresenta um *Chip* ESP32 com modulação LoRa e é voltada ao desempenho energético das aplicações que a utilizam. Esse dispositivo é configurado com a biblioteca LMIC, que implementa a camada de controle de acesso ao meio do protocolo LoRaWAN versão 1.0.2. A biblioteca é responsável por realizar a transmissão e recebimento de dados pelo transceptor SX1276 presente na placa, assim como codificar e decodificar os pacotes de dados. Para comunicação com a aplicação, os dados são recebidos por um ponto de acesso Radioenge RD43 [Radioenge 2018] e encaminhados para o servidor de redes *The Things Stack*. O servidor é integrado com o MQTT *Server*, possibilitando a comunicação com um MQTT *Client*. O MQTT *Client* é programado com a linguagem Python através da biblioteca *Eclipse Paho*.

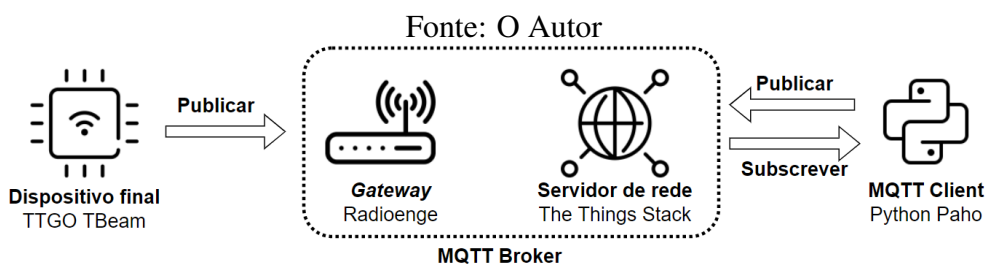


Figura 3. Diagrama de troca de mensagens com MQTT

3.2. Algoritmo de Retransmissão

Como no dispositivo final ocorre a transmissão de dados *uplink* e o recebimento e manipulação das mensagens *downlink* enviadas pelo servidor, o fluxo de dados referente a programação realizada no dispositivos é mostrado na Figura 4. Uma vez recebida a mensagem de *downlink*, o dispositivo final a decodifica e extrai o pacote de dados enviado pelo servidor de aplicação. Esses dados possuem quais são as mensagens que precisam ser retransmitidos pelo dispositivo final. Então, o dispositivo final retransmite apenas os pacotes de dados necessários, não sobrecarregando os pontos de acesso com mensagens de confirmação ou enviando mensagens duplicadas. Toda a configuração mencionada é realizada em C++ com Arduino IDE, utilizando também a biblioteca LMIC para implementação do protocolo LoRaWAN.

No servidor de aplicação, onde está sendo executado o MQTT *Client*, os dados são recebidos com um número de sequência *Fcnt*. Como o *Fcnt* é sequencial e incrementado em uma unidade para cada pacote enviado pelo dispositivo, é possível gerenciar quais pacotes de dados não foram recebidos corretamente. Desse modo, o servidor de aplicação armazena o número de sequência dos pacotes recebidos e perdidos separadamente. Os pacotes não recebidos irão compor uma mensagem *downlink* a ser enviada ao dispositivo solicitando a retransmissão dos dados faltantes. Essa mensagem de *downlink* só será agendada para transmissão quando houver *N* pacotes de dados necessário para serem retransmitidos, sendo *N* um valor ajustável buscando o melhor desempenho da rede. Após o dispositivo final reenviar os pacotes faltantes, eles são removidos da lista de pacotes perdidos do servidor de aplicação. Todo esse processo é demonstrado na Figura 5.

3.3. Validação do Algoritmo

Testes de desempenho são realizados com o propósito de validar o aumento na entrega de pacotes de dados. Nesse caso, é examinada a porcentagem de pacotes não recebidos

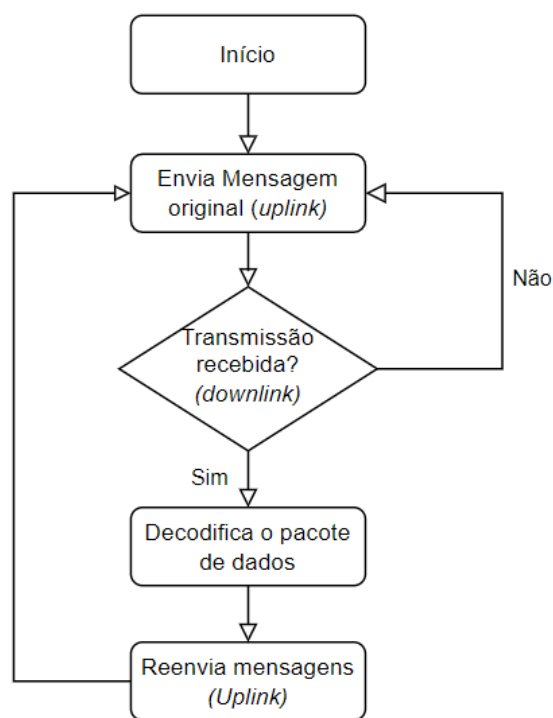


Figura 4. Fluxograma da troca de mensagens do dispositivo final

e a porcentagem de pacotes recebidos após as retransmissões solicitadas pelo algoritmo proposto. Verifica-se, para cada pacote de dados, quantas solicitações de retransmissão são necessárias para que o pacote seja de fato recebido pelo ponto de acesso. Ao saber qual é o tempo de recebimento de cada pacote de dados, calcula-se o atraso para recepção de cada um dos pacotes, sendo essa a diferença em segundos entre o tempo de recepção do Fcnt em análise e o de seu anterior.

Essa abordagem é adotada para investigar se ocorre uma redução na perda de pacotes transmitidos pelo dispositivo. Esses pacotes que são perdidos pelas interferências presentes no meio de propagação, sendo que esses pacotes de dados nunca chegariam a ser recebidos se não houvesse a utilização de um mecanismo para retransmissão. Investiga-se também como o atraso de tempo entre pacotes retransmitidos varia para diferentes números de Fcnt em uma mesma solicitação, valor N da Figura 5.

4. Resultados Preliminares

Esta seção descreve a coleta dos resultados, incluindo o número de execuções realizadas e os parâmetros utilizados em cada uma dessas execuções. Em seguida são expostos os resultados obtidos, como a relação entre o número de pacotes de dados perdidos e as solicitações de retransmissões realizadas. Também é mostrado quanto tempo se levou para que os pacotes de fato fossem recebidos, calculando a diferença de tempo, em segundos, entre o momento de recebimento do pacote atual e de seu antecessor. Foram adotados diferentes valores associados ao fator de retransmissão N, a fim de verificar o seu impacto quanto ao tempo que levaria para um pacote perdido ser retransmitido e recebido corretamente.

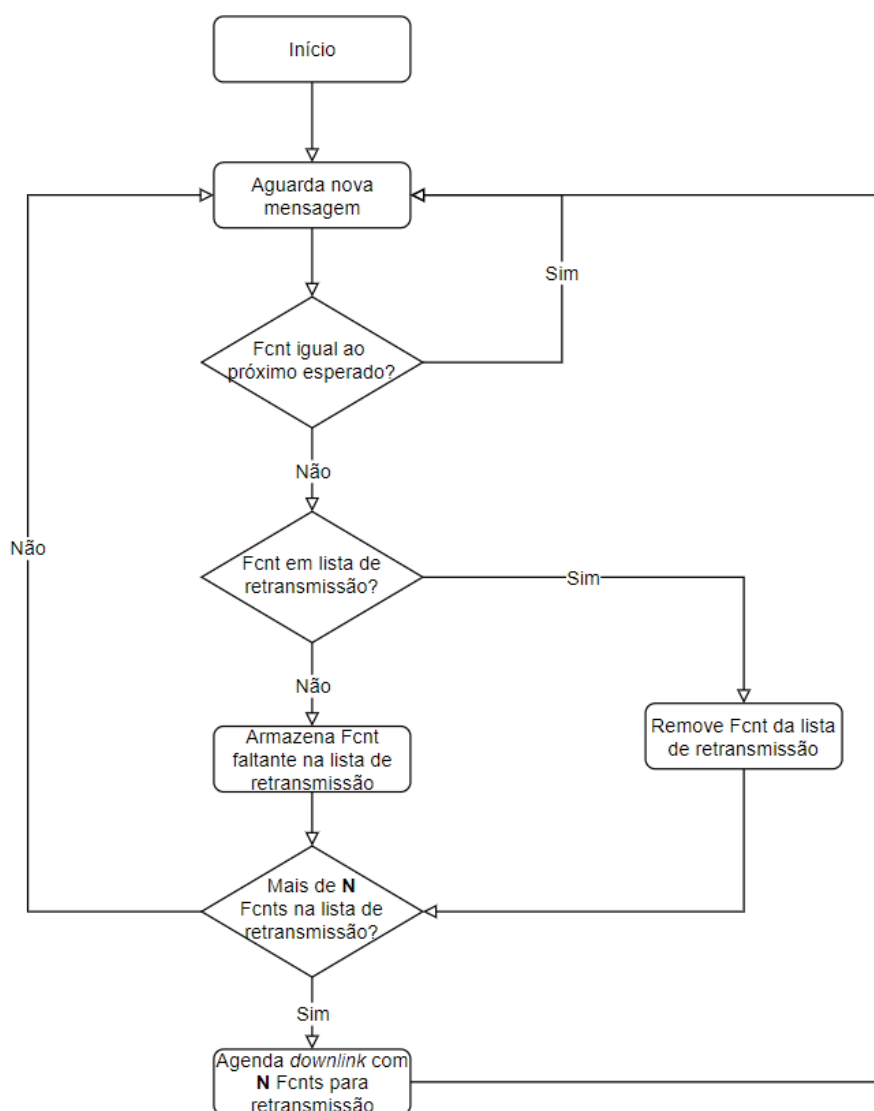


Figura 5. Fluxograma do mecanismo de retransmissão

4.1. Ambiente de Realização das Execuções

Em todas as execuções foi utilizado apenas um dispositivo final, esse estando localizado em uma residência a 5 Km de distância do ponto de acesso, localizado na UTFPR Campus Toledo. Por conta do relevo local e, principalmente, das obstruções presentes no meio, ocorreram eventuais interrupções de sinal durante o período de transmissão de dados entre o dispositivo final e ponto de acesso. Este cenário é o mesmo encontrado em um ambiente real, onde sensores localizados em áreas remotas se comunicam com o ponto de acesso.

Durante os testes foram realizadas 6 execuções. Cada execução consistiu em enviar coordenadas geográficas pelo dispositivo final, simulando o rastreamento de objetos. Foram enviados 150 pacotes de dados com Fcnt distintos por execução. A cada execução os pacotes foram enviados periodicamente a cada 30 segundos em transmissões normais ou 10 segundos durante a realização de retransmissões, esse período foi adotado para satisfazer as necessidades de transmissão em tempo real da aplicação de rastreamento executada.

4.2. Resultados

A Tabela 1 apresenta as 6 execuções realizadas, considerando diferentes valores para o fator de retransmissão N . Para um valor de N igual a 1 temos a retransmissão de pacotes de dados sendo solicitadas logo após a identificação de sua ausência, que ocorre quando se recebe um F_{cnt} fora de sequência. Quando o valor N aumenta para 2, torna-se necessário detectar a ausência de dois pacotes de dados que irão compor uma única solicitação de retransmissão *downlink*. O dispositivo final, por sua vez, após decodificar o pacote de dados, enviará as duas mensagens solicitadas, apresentando um espaçamento de 10 segundos entre cada uma das duas transmissões. A mesma lógica se estende para as execuções 5 e 6, com o valor N igual a 3.

Tabela 1. Resultado das execuções.

	N	Enviados	Perdidos	Solicitações	Retransmissões
1° Execução	1	150	6	7	6
2° Execução	1	150	7	9	7
3° Execução	2	150	17	14	17
4° Execução	2	150	14	11	14
5° Execução	3	150	8	3	8
6° Execução	3	150	9	6	9

A Tabela 1 apresenta também quantas mensagens, no total, foram enviadas pelos dispositivos, lembrando que todas as execuções se limitaram a 150 mensagens. A coluna *Perdidos* descreve o número de mensagens que não chegaram a ser recebidas pelo ponto de acesso em sua primeira transmissão. A coluna *solicitações* informa quantas mensagens foram enviadas pelo servidor de aplicação. Dessa forma, observa-se que nem todos os pacotes solicitados são de fato retransmitidos, podendo haver interferência na mensagem *downlink* enviada pelo ponto de acesso ou *uplink* reenviada pelo dispositivo final. Por exemplo, se uma dada mensagem for perdida durante a sua transmissão, o servidor de aplicação vai realizar uma solicitação para retransmissão, porém, como essa mensagem utiliza o mesmo enlace para comunicação, ela pode ser perdida durante a sua transmissão, sendo necessário o envio de uma nova solicitação pelo servidor de aplicação. Já a coluna *Retransmissões* descreve quantos dos pacotes de dados perdidos foram retransmitidos e recebidos com sucesso. Dessa forma, percebe-se que as colunas *Perdidos* e *Retransmissões* apresentam os mesmos valores, representando que todos os pacotes de dados que foram perdidos chegaram a ser recebidos através de retransmissões. Percebe-se que em alguns casos é necessário realizar mais de uma solicitação de retransmissão para o mesmo pacote de dados, como na primeira e segunda execuções.

A Tabela 2 apresenta quantas solicitações foram necessárias em cada uma das execuções para o recebimento dos dados após as retransmissões. Dessa forma, na execução de número 1 foram recebidos 5 pacotes de dados após a primeira solicitação de retransmissão e 1 durante a segunda solicitação, sendo esse, interferido em sua primeira tentativa de retransmissão. Já, na segunda execução, foram recebidos 5 pacotes de dados na primeira tentativa e 2 durante a segunda. A lógica se estende para todas as outras execuções.

Com isso, é possível verificar que de fato a maioria das retransmissões estão ocor-

	Execuções					
	Execução 1	Execução 2	Execução 3	Execução 4	Execução 5	Execução 6
1° Solicitação	5	5	13	9	7	5
2° Solicitação	1	2	2	5	1	1
3° Solicitação	-	-	-	-	-	3
4° Solicitação	-	-	-	-	-	-
5° Solicitação	-	-	2	-	-	-

Tabela 2. Número de pacotes recebidos por solicitações realizadas.

rendo logo após a sua solicitação de retransmissão. Porém, nas execuções 3 e 6 houve alguns pacotes sendo recebidos durante a terceira e quinta solicitação. Já durante a execução 4 vários pacotes foram recebidos apenas durante a segunda solicitação.

A seguir são apresentadas as sequências de recebimento de dados para cada uma das execuções. Na Figura 6 temos a sequência de pacotes referentes à primeira e à segunda execuções. Para um fator de retransmissão N igual a 1, a maioria dos pacotes retransmitidos são recebidos em até 90 segundos de diferença. Mas, como nem todos os pacotes de dados são recebidos na primeira tentativa de retransmissão, alguns apresentam um intervalo maior. O pacote com o maior intervalo entre transmissão e recepção atingiu 276 segundos.

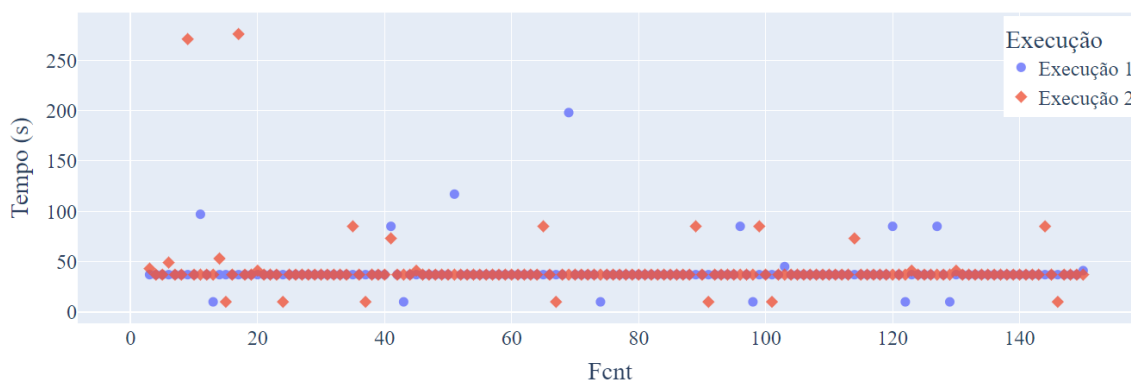


Figura 6. Sequencia de recebimento de dados para N igual a 1

Já para um fator de retransmissão igual a 2 percebe-se que o atraso de recebimento se apresenta mais elevado, o que é esperado, uma vez que, para se realizar a solicitação de retransmissão é necessário que pelo menos dois pacotes de dados tenham sido perdidos. Como demonstra a Figura 7, a maioria dos dados são recebidos em até 500 segundos de diferença, enquanto o pior caso para a execução 3 chega a 1290 segundos e para a execução 4 atinge 1050 segundos.

Na última execução mostrada na Figura 8, considerando N igual a 3, a maioria das retransmissões ocorrem em até 1500 segundos, porém podendo chegar, no pior dos casos, a 3697 segundos. O elevado tempo observado nessas execuções se deve aos mesmos motivos explicados quando se tinha um valor N igual a 2.

Com isso, verifica-se que em todas as execuções os pacotes de dados são recebidos com um intervalo de 30 a 40 segundos. Após a solicitação de retransmissão o próximo

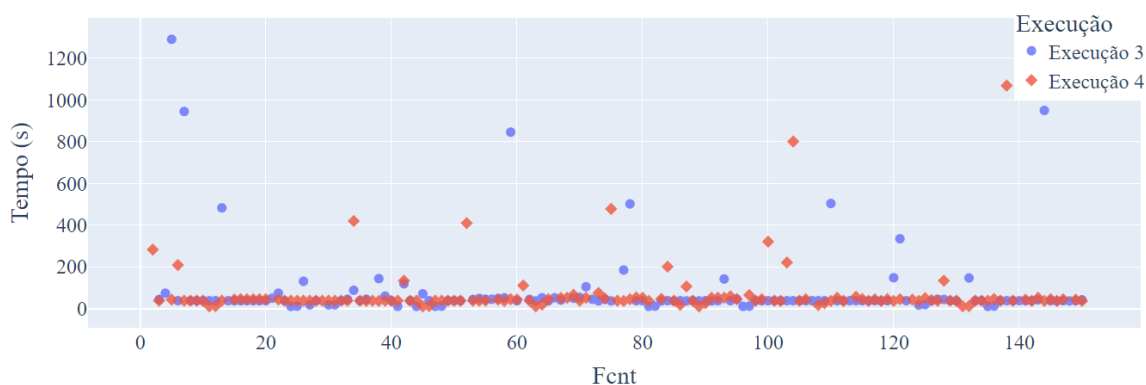


Figura 7. Sequencia de recebimento de dados para N igual a 2

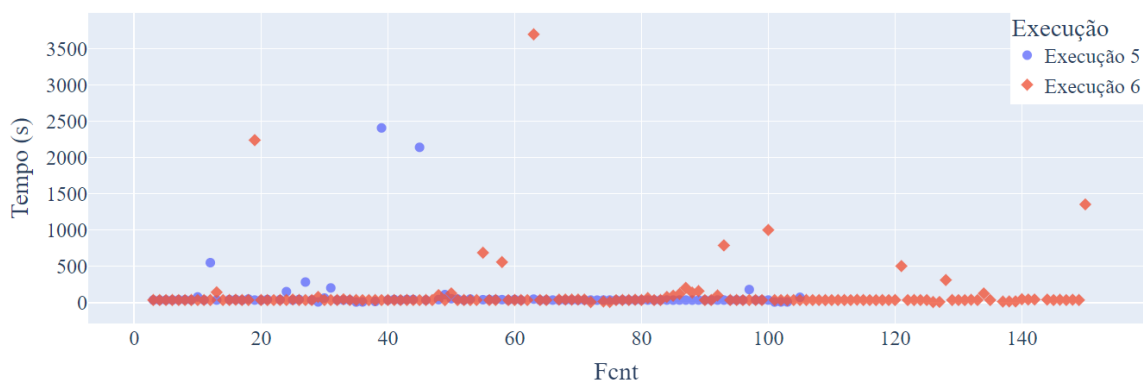


Figura 8. Sequencia de recebimento de dados para N igual a 3

pacote de dados, que utiliza a janela de recebimento do dispositivo final, é recebido com uma diferença de 10 segundos, como mencionado anteriormente pela abordagem adotada.

A partir das execuções realizadas, verifica-se como a utilização de diferentes fatores de retransmissão N podem influenciar no tempo para recebimento de um pacote de dados inicialmente perdido. Para um fator N igual a 1, o pacote com maior atraso para recebimento foi de 276 segundos; já para N igual a 2 o pior caso foi de 1290 segundos, e para o seu valor em 3 esse tempo aumentou para 3697 segundos. Assim, fica evidente que em aplicações que necessitam de uma troca de mensagens em tempo real um valor de N igual a 1 passa a ser mais adequado. Enquanto que, ao lidar com aplicações em que o momento de recebimento não é prioridade, mas sim o fato de que todos os dados gerados pelo dispositivo sejam recebidos, um valor de N igual a 2 ou 3 pode ser mais adequado.

5. Conclusões

Com este trabalho foi possível verificar como a utilização de um algoritmo para gerenciamento de retransmissões auxilia na tolerância a falhas presentes em redes sem fio LoRaWAN. A abordagem proposta operou em nível de aplicação das redes LoRaWAN. Foi desenvolvido todo o gerenciamento das mensagens em um servidor de aplicação com integração ao protocolo de comunicação MQTT. Os experimentos trouxeram diferentes visões para validação da proposta utilizada. Em um primeiro, realizou-se uma análise geral quanto a eficácia do algoritmo proposto. Com emprego do algoritmo houve aumento significativo na confiabilidade de entrega das mensagens, uma vez que, após as

retransmissões, em todas as execuções 100% dos dados gerados pelo dispositivo foram recebidos. Pelos resultados se torna perceptível como a utilização de diferentes fatores de retransmissão podem impactar na tolerância do intervalo de recebimento das retransmissões. Enquanto para um valor de N menor são realizadas mais solicitações de retransmissões, o atraso de recebimento das mensagens perdidas se torna menor. Ou seja, um valor de N menor é mais adequado para aplicações que necessitam de troca de mensagens em tempo real, como é o caso de rastreamentos de objetos móveis.

Como trabalhos futuros, é importante realizar a avaliação do desempenho energético da abordagem adotada, uma vez que, ao ser solicitadas retransmissões em um maior número de pacotes de dados, são necessárias decodificações em um maior número de janelas de recebimentos de dados dos dispositivos. Assim como, torna-se necessário realizar um estudo referente ao impacto dos diferentes valores de N no congestionamento dos pontos de acesso, como mencionado por [Marais et al. 2020] a utilização excessiva de mensagens de *downlink* pode afetar negativamente o desempenho da rede. Outro trabalho a ser desenvolvido é comparar a abordagem proposta com o algoritmo CR-LoRaWAN para verificar de fato quais são os ganhos operacionais do dispositivo final. E, ainda, verificar como o algoritmo se comporta frente ao aumento de pontos de acesso em uma aplicação real para cidades inteligentes.

Referências

- Abdelfadeel, K. Q., Zorbas, D., Cionca, V., and Pesch, D. (2020). Fine-grained scheduling for reliable and energy-efficient data collection in lorawan. *IEEE Internet of Things Journal*, 7(1):669–683.
- Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., and Ayyash, M. (2015). Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys Tutorials*, 17(4):2347–2376.
- Alliance, L. (2017). *LoRaWAN 1.1 specification*. LoRa Alliance, Beaverton, Oregon, Estados Unidos.
- Almeida, T. V. O., Pastório, J. P., Álvaro R. C. Souza, Spanhol, F. A., and de Camargo, E. T. (2021). Em busca de um algoritmo de retransmissão em redeslorawan. *V Workshop de Computação Urbana*.
- Camargo, E., Spanhol, F. A., and e Souza, Á. R. C. (2021). Deployment of a lorawan network and evaluation of tracking devices in the context of smart cities. *J. Internet Serv. Appl.*, 12(1):8.
- Capuzzo, M., Magrin, D., and Zanella, A. (2018). Confirmed traffic in lorawan: Pitfalls and countermeasures. *17th Annual Mediterranean Ad Hoc Networking Workshop*, pages 87–93.
- Ferreira, A. E., Ortiz, F. M., Costa, L. H. M. K., Foubert, B., Amadou, I., and Mitton, N. (2020). A study of the lora signal propagation in forest, urban, and suburban environments. *Annals of Telecommunications*.
- Forouzan, B. A. (2009). *Comunicação de dados e redes de computadores*. AMGH Editora.

- Fraga-Lamas, P., Celaya-Echarri, M., Azpilicueta, L., Lopez-Iturri, P., Falcone, F., and Fernández-Caramés, T. M. (2020). Design and empirical validation of a lorawan iot smart irrigation system. *Proceedings*.
- Framingham, M. (2019). The growth in connected iot devices is expected to generate 79.4zb of data in 2025.
- Gomez, C. and Crowcroft, J. (2020). Multimodal retransmission timer for lpwan. *IEEE Internet of Things Journal*, 7(6):4827–4838.
- Johnston, S. J., Basford, P. J., Bulot, F. M. J., Apetroaie-Cristea, M., Easton, N. H. C., Davenport, C., Foster, G. L., Loxham, M., Morris, A. K. R., and Cox, S. J. (2019). City scale particulate matter monitoring using lorawan based air quality iot devices. *Sensors*.
- Khanna, A. and Kaur, S. (2019). Evolution of internet of things (iot) and its significant impact in the field of precision agriculture. *Computers and Electronics in Agriculture*, pages 218–321.
- Lavric, A. and Popa, V. (2017). Internet of things and lora™ low-power wide-area networks: A survey. In *2017 International Symposium on Signals, Circuits and Systems (ISSCS)*, pages 1–5.
- LoRa Alliance (2017). *LoRaWAN™ 1.1 specification*. LoRa Alliance, Beaverton, Oregon, Estados Unidos.
- Marais, J. M., Abu-Mahfouz, A. M., and Hancke, G. P. (2020). A survey on the viability of confirmed traffic in a lorawan. *IEEE Access*, 8:9296–9311.
- Mekki, K., Bajic, E., Chaxel, F., and Meyer, F. (2017). A comparative study of lpwan technologies for large-scale iot deployment. *ICT Express*.
- Pastório, J. P., Álvaro R. C. Souza, and de Camargo, E. T. (2021). Simulação do monitoramento de veículos em uma rede lorawan usando ns-3. *XIX Escola Regional de Redes de Computadores*.
- Radioenge (2018). *LoRa Gateway Radioenge*. Radioenge.
- Rossato, J., Spanhol, F., and Camargo, E. (2020). Implantação e avaliação de uma rede sem-fio de longo alcance e baixa potência para cidades inteligentes. In *Anais do IV Workshop de Computação Urbana*, pages 192–205, Porto Alegre, RS, Brasil. SBC.
- Semtech (2019). *LoRa and LoRaWAN: A Technical Overview*.
- Seneviratne, P. (2019). *Beginning LoRa Radio Networks with Arduino: Build Long Range, Low Power Wireless IoT Networks*. Apress.
- Shanmuga Sundaram, J. P., Du, W., and Zhao, Z. (2020). A survey on lora networking: Research problems, current solutions, and open issues. *IEEE Communications Surveys Tutorials*, 22(1):371–388.
- Singh, R. K., Aernouts, M., Meyer, M. D., Weyn, M., and Berkvens, R. (2020). Leveraging lorawan technology for precision agriculture in greenhouses. *Sensors (Basel, Switzerland)*.