

**INSTITUTO FEDERAL DE EDUCAÇÃO,
CIÊNCIA E TECNOLOGIA DO CEARÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE
TELECOMUNICAÇÕES**

Leonardo Estevão da Mota

**FUOTA 4.0 - UMA SOLUÇÃO DE ATUALIZAÇÃO RÁPIDA DE
FIRMWARE UTILIZANDO LORA E LORAWAN EM APLICAÇÕES DE
MEDIÇÃO INTELIGENTE DE ENERGIA.**

Fortaleza, Ceará

2023

Leonardo Estevão da Mota

FUOTA 4.0 - UMA SOLUÇÃO DE ATUALIZAÇÃO RÁPIDA DE FIRMWARE
UTILIZANDO LORA E LORAWAN EM APLICAÇÕES DE MEDIÇÃO INTELIGENTE
DE ENERGIA.

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Telecomunicações do Instituto Federal de Educação, Ciência e Tecnologia do Ceará – IFCE – Campus Fortaleza, como requisito para obtenção do Título de Mestre.

Orientador: Dr. Francisco José Alves de Aquino

Fortaleza, Ceará

2023

Dados Internacionais de Catalogação na Publicação
Instituto Federal do Ceará - IFCE
Sistema de Bibliotecas - SIBI
Ficha catalográfica elaborada pelo SIBI/IFCE, com os dados fornecidos pelo(a) autor(a)

MOTA, LEONARDO.

FUOTA 4.0 - UMA SOLUÇÃO DE ATUALIZAÇÃO RÁPIDA DE FIRMWARE UTILIZANDO LORA E LORAWAN EM APLICAÇÕES DE MEDIÇÃO INTELIGENTE DE ENERGIA. / LEONARDO MOTA. - 2023.

73 f. : il. color.

Dissertação (Mestrado) - Instituto Federal do Ceará, Mestrado em Engenharia de Telecomunicações, Campus Fortaleza, 2023.

Orientação: Prof. Dr. Francisco José Alves de Aquino.

1. FUOTA. 2. LoRa. 3. IoT. 4. Redes. 5. Telecomunicações. I. Título.

CDD 621.382



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO CEARÁ
Av. 13 de maio, 2081 , - Bairro Benfica - CEP 60040-531 - Fortaleza - CE - www.ifce.edu.br

AVALIAÇÃO

Processo: 23256.006103/2023-81

Interessado: Leonardo Estevao da Mota

LEONARDO ESTEVÃO DA MOTA

FUOTA 4.0 - UMA SOLUÇÃO DE ATUALIZAÇÃO RÁPIDA DE FIRMWARE UTILIZANDO LORA E LORAWAN EM APLICAÇÕES DE MEDIÇÃO INTELIGENTE DE ENERGIA

Dissertação apresentada ao curso de Pós-Graduação em Engenharia de Telecomunicações do Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE) - *Campus Fortaleza*, como requisito parcial para obtenção do Título de Mestre em Engenharia de Telecomunicações.

Área de concentração: Sistemas de Telecomunicações

Aprovado em: **19 / 05 / 2023**

BANCA EXAMINADORA

Prof. Dr. **Francisco José Alves de Aquino** - orientador
Instituto Federal de Educação, Ciência e Tecnologia do Ceará - IFCE

Prof. Dr. **Auzuir Ripardo de Alexandria**
Instituto Federal de Educação, Ciência e Tecnologia do Ceará - IFCE

Prof. Dr. **Regis Cristiano Pinheiro Marques**
Instituto Federal de Educação, Ciência e Tecnologia do Ceará - IFCE

Prof. Dr. **Kléber Zuza Nóbrega**
Universidade Federal do Ceará - UFC

Dedico este trabalho a minha mãe Francisca, meu pai José e minha namorada Josy, sem os quais eu não teria chegado até aqui.

Agradecimentos

Primeiramente a Deus, pela saúde e força para trilhar todas as oportunidades que me foram dadas.

Aos meus pais, Francisca Estevão da Silva (in memoriam) e José Estevão da Mota, por todo suporte durante todos os anos de minha vida.

A minha namorada Josy Quintela, pela ajuda e força durante todo o período do projeto.

Ao meu orientador Francisco José Alves de Aquino, pela orientação e conselhos, além das excelentes aulas de Processamento Digital de Sinais.

Aos professores do PPGET que participaram da minha formação durante o curso, em especial aos professores Glendo de Freitas Guimarães, Auzuir Ripardo de Alexandria, Regis Cristiano Pinheiro Marques, Daniel Xavier Gouveia e Tarique da Silveira Cavalcante.

A Fundação Cearense de Apoio ao Desenvolvimento Científico e Tecnológico (FUN-CAP), pelo apoio financeiro, possibilitando a realização desse trabalho.

Ao Núcleo de Estudos e Pesquisas do Norte e Nordeste (NEPEN) e a Equatorial Energia, pelo apoio com instalações para a realização das provas de conceito do projeto.

Ao Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE), por sempre apoiar a melhoria da educação tecnológica no nosso estado.

Aos membros da banca.

A todos que direta ou indiretamente contribuíram para a realização deste trabalho.

“Inteligência é a habilidade de se adaptar às mudanças”

Stephen Hawking

Resumo

A cada ano mais dispositivos IoT são instalados mundialmente, previsões para 2025 são que teremos cerca de 75 bilhões de dispositivos em todo mundo. Cada vez mais a atualização de firmware se torna necessária para manter a vida útil dessas aplicações. Neste trabalho propomos um modelo de FUOTA via LoRa. Tal modelo é composto de quatro etapas: configuração, atualização, ativação de sementes futuras e ativação de dispositivos comuns. Como prova-de-conceito para o modelo proposto, foi implementado um código-fonte em C e embarcado em uma PCI contendo um Arm Cortex M4 e um rádio SEMTECH SX126x. Em uma rede hipotética realizou-se os ensaios de laboratório e de campo. Foram realizados testes em uma rede LoRaWAN real, dos medidores de energia elétrica da Equatorial Energia na cidade de São Luís. Com base nos resultados obtidos, em ambos os cenários hipotético e real, concluiu-se que o modelo de FUOTA proposto pode vir a ser utilizado em redes LoRaWAN como meio de atualização de imagens de *firmware* ou transfência de grandes quantidades de dados, pois o mesmo obteve um tempo de transferência inferior aos relatados na literatura.

Palavras-chaves: FUOTA, LoRa, IoT, Redes, Telecomunicações.

Abstract

Every year more IoT devices are installed worldwide, predictions for 2025 are that we will have about 75 billion devices worldwide, so more and more firmware updates become necessary to maintain the useful life of these applications. In this work we propose a model of FUOTA via LoRa, such model is composed of four steps: configuration, update, activation of future seeds and activation of common devices. As a proof-of-concept for the proposed model, a source code in C was implemented and embedded in a PCI containing an Arm Cortex M4 and a SEMTECH SX126x radio. Next, the laboratory and field tests that were carried out on a hypothetical network were detailed. The next step was to carry out tests on a real LoRaWAN network, on Equatorial Energia's electricity meters in the city of São Luís. Based on the results obtained in both the hypothetical and real scenarios, it was concluded that the proposed FUOTA model can be used in LoRaWAN networks as a means of updating *firmware* images or transferring large amounts of data, because it obtained a transfer time lower than those reported in the literature.

Key-words: FUOTA, LoRa, IoT, Nets, Telecommunications

Lista de ilustrações

Figura 1 – Estimação de dispositivos IoT x Ano. Fonte: Poursafar, Alahi e Mukhopadhyay (2018)	15
Figura 2 – Alcance e Taxa de Transferência de tecnologias Wireless. Fonte: Poursafar, Alahi e Mukhopadhyay (2018)	19
Figura 3 – Representação de um CHIRP LoRa. Fonte: Montagny (2021)	20
Figura 4 – Estrutura da Mensagem LoRa. Fonte: Alliance (2017)	21
Figura 5 – Pilha do protocolo LoRaWAN. Fonte: Alliance (2017)	23
Figura 6 – Slots de recepção para um dispositivo Classe A. Fonte: Montagny (2021)	23
Figura 7 – Slots de recepção para um dispositivo Classe B. Fonte: Montagny (2021)	24
Figura 8 – Slots de recepção para um dispositivo Classe C. Fonte: Montagny (2021)	25
Figura 9 – Potência consumida x capacidade de downlink. Fonte: Montagny (2021)	25
Figura 10 – Visão geral de uma rede LoRaWAN. Fonte: Montagny (2021)	26
Figura 11 – Visão geral de um Gateway LoRaWAN. Fonte: Montagny (2021)	27
Figura 12 – Arquitetura de um FUOTA LoRaWAN. Fonte: Abdelfadeel et al. (2020)	28
Figura 13 – Diagrama de Sequência de um FUOTA LoRaWAN Multicast Classe C. Fonte: Abdelfadeel et al. (2020)	29
Figura 14 – Tempo de Atualização x Tamanho do Firmware. Fonte: Abdelfadeel et al. (2020)	30
Figura 15 – Eficiência de Atualização Teórica. Fonte: Abdelfadeel et al. (2020) . . .	30
Figura 16 – Modelo Proposto para uso de BC. Fonte: Anastasiou et al. (2020) . . .	31
Figura 17 – Diagrama de Sequência com BC. Fonte: Anastasiou et al. (2020)	32
Figura 18 – Tempo de Atualização x Número de Dispositivos. Fonte: Anastasiou et al. (2020)	33
Figura 19 – Tela de informações sobre a IDE Atollic utilizada no desenvolvimento. Fonte: STMicroelectronics	34
Figura 20 – Janela de trabalho da IDE Atollic. Fonte: STMicroelectronics	35
Figura 21 – Segger J-link. Fonte: SEGGER	35
Figura 22 – Placa de circuito impresso com o sistema de hardware IoT LoRa. Fonte: Autor	36
Figura 23 – Medidores de energia monofásicos Eletra (à esquerda) e Nansen (à direita). Fonte: Autor	37
Figura 24 – Antena Omnidirecional Miot. Fonte: Miot Wireless Solutions.	37
Figura 25 – Comparação entre as pilhas LFR e LoRaWAN. Fonte: Autor	39
Figura 26 – Modelo de Fuota Proposto - LFR. Fonte: Autor	39
Figura 27 – Estrutura da mensagem LFR. Fonte: Autor	40
Figura 28 – Diagrama de Sequência Unicast para FUOTA LFR. Fonte: Autor	42

Figura 29 – Diagrama de Sequência Multi-layer para FUOTA LFR. Fonte: Autor	44
Figura 30 – Máquina de Estado do Modelo. Fonte: Autor	45
Figura 31 – Locações dos dispositivos para os testes em Fortaleza/CE. Fonte: Autor.	47
Figura 32 – Perfil de elevação entre Master-Seed e Common-device 4. Fonte: Autor.	48
Figura 33 – Perfil de elevação entre Master-Seed e Common-device 6. Fonte: Autor.	48
Figura 34 – Perfil de elevação entre Master-Seed e Common-device 7. Fonte: Autor.	48
Figura 35 – Locações dos dispositivos para os testes na CEASA em São Luís/MA. Fonte: Autor.	50
Figura 36 – Locações dos dispositivos para os testes na CEASA em São Luís/MA. Fonte: Autor.	50
Figura 37 – Atualização do <i>firmware</i> de campo para suporte ao protocolo LFR. Fonte: Autor.	51
Figura 38 – Frequência dos Canais US902-928. Fonte: Alliance (2020).	52
Figura 39 – Sistema Supervisório da Equatorial. Fonte: Autor.	53
Figura 40 – Locações dos dispositivos para os testes em Areinha São Luís/MA. Fonte: Autor.	54
Figura 41 – Perfil de Elevação entre Master-Seed 2 e Common-Device 4. Fonte: Autor.	55
Figura 42 – Sistema de agendamento do LFR - versão beta. Fonte: Autor.	55
Figura 43 – Sistema Supervisório da Equatorial. Fonte: Autor.	56
Figura 44 – Tempos de atualização (minutos) em diferentes cenários/processos FU- OTA. Fonte: Autor.	58

Lista de tabelas

Tabela 1 – Tempo de transferência da imagem via LFR	46
Tabela 2 – Medidas de Desempenho e distância entre os sítios	47
Tabela 3 – Cenários de validação do FUOTA LFR em Campo	49
Tabela 4 – Parâmetros de Modulação LoRa dos Cenários de validação	52
Tabela 5 – Comparação entre tempos de atualização em diferentes processos/cenários de FUOTA	57

Lista de abreviaturas e siglas

BC	Block Chain
BT	Bandwidth Time Product
CHIRP	Compressed High Intensity Radar Pulse
CR	Code Rate
CSS	Chirp Spread Spectrum
DR	Data Rate
End-point	End-Point Device
FSK	Frequency Shift Keying
FHSS	Frequency-hopping spread spectrum
FUOTA	Firmware Update over the Air
IDE	Ambiente de Desenvolvimento Integrado
IoT	Internet of Things
LNS	LoRaWAN Network Server
LPWAN	Low Power Wide Area Network
LoRa	Long Range
LoRaWAN	Long Range Wide Area Network
MAC	Media Access Control
PA	Power Amplifier
PCI	Placa de Circuito Impresso
RSSI	Received Signal Strength Indicator
SNR	Signal to Noise
SF	Spreading Factor

Sumário

1	INTRODUÇÃO	15
1.1	Justificativa	15
1.2	Objetivos	17
1.2.1	Objetivos específicos	17
1.3	Organização do texto	17
2	FUNDAMENTOS TEÓRICOS	19
2.1	Tecnologias <i>Low-Power Wide Area Networks (LPWAN)</i>	19
2.2	Modulação LoRa	20
2.2.1	Largura de banda escalável	21
2.2.2	Baixo consumo	21
2.2.3	Alta robustez	21
2.2.4	Resistência a Multi-percurso / Desvanecimento	22
2.2.5	Capacidade de rede aprimorada	22
2.3	Protocolo LoRaWAN	22
2.3.1	Classe A	23
2.3.2	Classe B	24
2.3.3	Classe C	24
2.3.4	Gateway e Topologia de Rede LoRaWAN	26
2.4	FUOTA - <i>Firmware Upgrade On the Air</i>	27
2.5	Trabalhos Relacionados	28
3	MATERIAIS E MÉTODOS	34
3.1	Materiais	34
3.1.1	Linguagem de Programação e IDE utilizadas	34
3.1.2	Hardware utilizado	36
3.2	Metodologia	38
3.2.1	Topologia de Rede Proposta	38
3.2.2	Modelo de FUOTA Proposto	38
3.2.2.1	Estrutura de Payload Proposto para o Protocolo	40
3.2.2.2	Etapa de Configuração - <i>Setup</i>	40
3.2.2.3	Etapa de Transmissão e Atualização - <i>Update</i>	41
3.2.2.4	Etapa de ativação da <i>Future Seed</i>	43
3.2.2.5	Etapa de atualização do <i>Common End-node Device</i>	43
3.2.2.6	Máquina de Estado do Modelo	44
3.2.3	Avaliação do Modelo	46

4	RESULTADOS E DISCUSSÕES	49
5	CONCLUSÕES	59
5.1	Trabalhos futuros	60
	REFERÊNCIAS	61
	APÊNDICES	63
	APÊNDICE A – DIAGRAMA DE SEQUÊNCIAS	64
	APÊNDICE B – DIAGRAMA DE SEQUÊNCIAS	65
	APÊNDICE C – TESTES DE CAMPO CEASA SÃO LUÍS/MA	66
	APÊNDICE D – TRECHO DO CÓDIGO LFR	68
	APÊNDICE E – TRECHO DO LOG DE UMA TRANSFE- RÊNCIA DE FIRMWARE VIA FUOTA LFR	72

1 Introdução

1.1 Justificativa

A Internet das Coisas (em inglês *Internet of Things*) é um conceito que devido ao avanço das tecnologias embarcadas, sobretudo as de rádio, torna-se possível conectar objetos, eletrônicos, sensores e vários outros dispositivos que enviam dados remotos e sem fio para aplicações e pessoas. Espera-se que o IoT produza uma quantidade maciça de informações a serem utilizadas para otimizar consumo de todos os tipos de recursos, melhorar eficiência e agregar valor a soluções já existentes.

De acordo com várias projeções, das quais podemos citar *Machina Research*, *Analysis Manson*, *Gartner*, estima-se números de dispositivos conectados na ordem de dezenas de bilhões, com crescimento exponencial até 2026. Segundo [Poursafar, Alahi e Mukhopadhyay \(2018\)](#) até 2025, cerca de 75 bilhões de dispositivos inteligentes estarão instalados e operacionais em diversos sistemas e ambientes em todo o mundo. A Figura 1 mostra como será essa progressão ao longo dos anos.

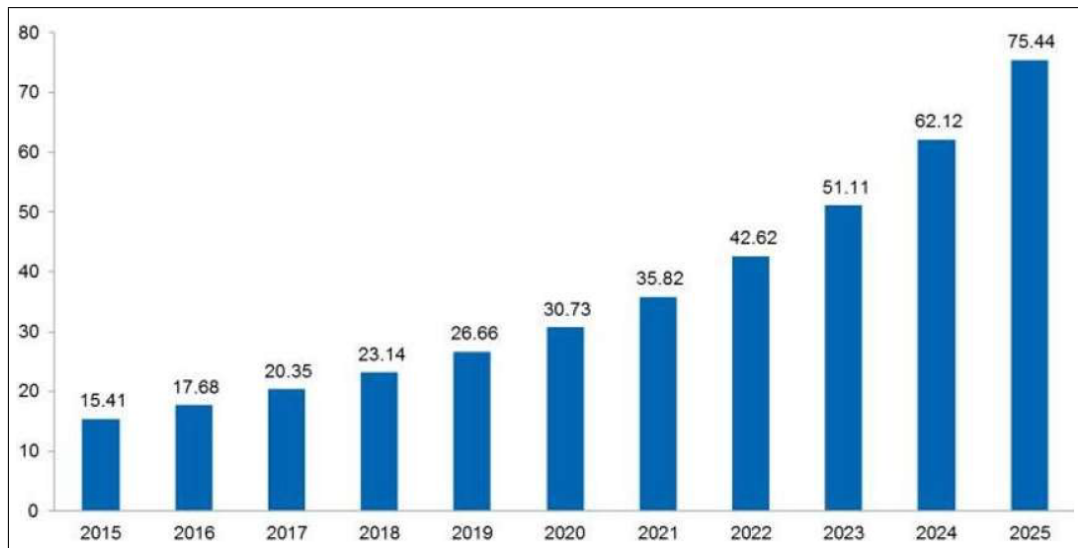


Figura 1 – Estimação de dispositivos IoT x Ano. Fonte: [Poursafar, Alahi e Mukhopadhyay \(2018\)](#)

Para viabilizar essa demanda cada vez maior de tornar e manter esses dispositivos conectados, diversas tecnologias de comunicação via rádio para aplicações *IoT* vêm sendo cada mais utilizadas. Dentre essas inúmeras tecnologias, estão as redes LPWAN, do inglês *Low-Power Wide Area Network*, que surgem como a opção mais adequada para aplicações que requerem uma ampla cobertura, sem a necessidade de uma alta taxa de

transmissão de dados. Redes LoRaWAN possuem uma alta escalabilidade, baixo consumo de energia e baixo custo de implantação (TURCINOVIC et al., 2020; POURSAFAR; ALAHI; MUKHOPADHYAY, 2018; FERREIRA et al., 2020; STAIKOPOULOS; KANAKARIS; PAPAKOSTAS, 2020; ZOURMAND et al., 2019). Dentre as diversas tecnologias LPWAN, destaca-se a tecnologia de modulação LoRa (*Long Range*) e seu protocolo de rede LoRaWAN (ALLIANCE, 2017), como sendo a mais popular e utilizada (MTETWA et al., 2020). As redes LoRaWAN possuem uma topologia em estrela, onde os dispositivos se comunicam com um nó central chamado *Gateway*, que faz a ponte entre os dispositivos *LoRa/LoRaWAN* e um servidor externo, funcionando como um encaminhador de pacotes para uma entidade de software chamado *LoRa Network Server (LNS)* (ALLIANCE, 2017).

Uma das características mais críticas para instalações de *IoT* é o *Firmware Update Over The Air (FUOTA)*. O *FUOTA* é um processo de atualização do *firmware* de dispositivos de forma remota em redes sem fio. Tal característica permite aos dispositivos *IoT* expandir o seu tempo de vida e melhorar sua performance em campo, pois novas funcionalidades podem ser instaladas sem a necessidade de intervenção humana *in loco*. O *FUOTA* representa um meio de manter os dispositivos *IoT* atualizados durante todo o seu tempo de vida, mantendo assim a confiabilidade, segurança (BEDI et al., 2018) e operação à longo prazo (ABDELFADEEL et al., 2020).

Atualmente os *firmwares* de dispositivos embarcados como aqueles supracitados, são da ordem de algumas centenas de *kilobytes*. Por exemplo, o *firmware* implementado para os testes realizados do protocolo proposto neste trabalho possui cerca de 196kB. Atualizar dispositivos com esse tamanho de *firmware* é um desafio para redes LoRaWAN (STAIKOPOULOS; KANAKARIS; PAPAKOSTAS, 2020; ABDELFADEEL et al., 2020), pois uma das características desse protocolo é sua baixa *data rate*. Outra característica do protocolo LoRaWAN é o fato dele operar em uma banda sub-GHz não licenciada (ALLIANCE, 2017; ALLIANCE, 2020), o que significa que o seu *Gateway* deve operar com uma restrição de *duty-cycle*. Por exemplo, na Europa essa restrição é de 1%, em outras palavras, o *Gateway* precisa ficar em silêncio um período igual à 99 vezes o tempo da última transmissão (ALLIANCE, 2017; ABDELFADEEL et al., 2020). Para entender melhor sobre como essas limitações impactam em um possível *FUOTA* LoRaWAN, pode-se tomar como exemplo a transferência de uma imagem de *firmware* de cerca de 50kB de tamanho. Esta levaria cerca de 17 horas em uma transmissão com confirmação de recebimento (ABDELFADEEL et al., 2020), ou seja, para cada mensagem do servidor→dispositivo (*downlink*) teria-se uma mensagem do dispositivo→servidor (*uplink*). Estima-se que para uma transferência sem confirmação, esse tempo cairia pela metade, de toda forma ainda seriam cerca de 8,5 horas para se transmitir uma imagem de apenas 50kB.

Diante do exposto, este trabalho propõe o desenvolvimento de um protocolo de rede que seria utilizado para a implementação de um mecanismo de *FUOTA* LoRa, não

LoRaWAN, mais rápido e eficiente que seu análogo que utiliza *LoRaWAN*. Ao longo desse trabalho estão descritas as especificações de um protocolo de rede para viabilizar um FUOTA LoRa, além disso, no Capítulo 4 são analisados os resultados iniciais de transmissões de imagens de *firmware* por meio do protocolo proposto nesse trabalho.

1.2 Objetivos

Desenvolver um protocolo de comunicação para a camada de transporte e rede da tecnologia LoRa que possibilite atualizar firmwares de dispositivos com velocidade e confiabilidade.

1.2.1 Objetivos específicos

- Estabelecer comunicação peer-to-peer entre os dispositivos permitindo mudança entre o protocolo LoRaWAN e o LFR.
- Criar uma rede com arquitetura em árvore, implementando algoritmos de roteamento que permitam atualização escalonada entre os dispositivos.
- Realizar um processo de FUOTA em uma rede LoRa em um tempo inferior ao descrito pela literatura.
- Realizar um processo de FUOTA em uma rede LoRa sem tornar a rede LoRaWAN nativa inoperante.

1.3 Organização do texto

Este trabalho está organizado em cinco capítulos, onde no Capítulo 2 são abordados os conceitos básicos teóricos a respeito das tecnologias utilizadas no desenvolvimento do protocolo de rede proposto. Também estão descritas as características das tecnologias de transmissão *LPWAN*, dando ênfase na modulação *LoRa* e em seu protocolo de rede comercial *LoRaWAN*. Ainda neste capítulo estão as características da arquitetura em árvore, os conceitos de FUOTA e trabalhos relacionados ao desenvolvimento de FUOTA.

No Capítulo 3 são apresentados os Materiais e Métodos, com uma breve descrição da linguagem de programação e da IDE utilizadas para a implementação da biblioteca do protocolo. Neste capítulo também está uma descrição do hardware utilizado. Após isso é descrito o modelo do protocolo proposto e as medidas de desempenho utilizadas para avaliação do modelo.

No Capítulo 4 são apresentados os resultados obtidos nos testes iniciais do FUOTA proposto.

E por último, no Capítulo 5 é apresentada a conclusão do trabalho de acordo com os resultados apresentados no Capítulo 4 além de apresentar sugestões para trabalhos futuros.

2 Fundamentos Teóricos

2.1 Tecnologias *Low-Power Wide Area Networks (LPWAN)*

Nos dias atuais, a demanda por coleta de informações se torna mais presente e necessária, seja em cidades inteligentes, setor agropecuário, setor energético (TURCINOVIC et al., 2020). Assim o número de dispositivos conectados para realizar a aquisição desses dados tem aumentado significativamente, segundo Poursafar, Alahi e Mukhopadhyay (2018) até 2025 estima-se que cerca de 75 bilhões de dispositivos inteligentes estejam instalados e operacionais em diversos sistemas e ambientes em todo o mundo. Com isso diversas tecnologias de comunicações via rádio para aplicações de *Internet of Things (IoT)* e redes de sensores sem fio são utilizadas dependendo dos requisitos de cada ambiente ou aplicação. As tecnologias LPWAN, surgem como uma melhor opção para as aplicações que requerem uma ampla cobertura, sem a necessidade de uma alta taxa transmissão de dados, alta escalabilidade, baixo consumo de energia e baixo custo de implantação (FERREIRA et al., 2020; STAIKOPOULOS; KANAKARIS; PAPAKOSTAS, 2020; NTSEANE; ISONG, 2019; ZOURMAND et al., 2019; POURSAFAR; ALAHI; MUKHOPADHYAY, 2018).

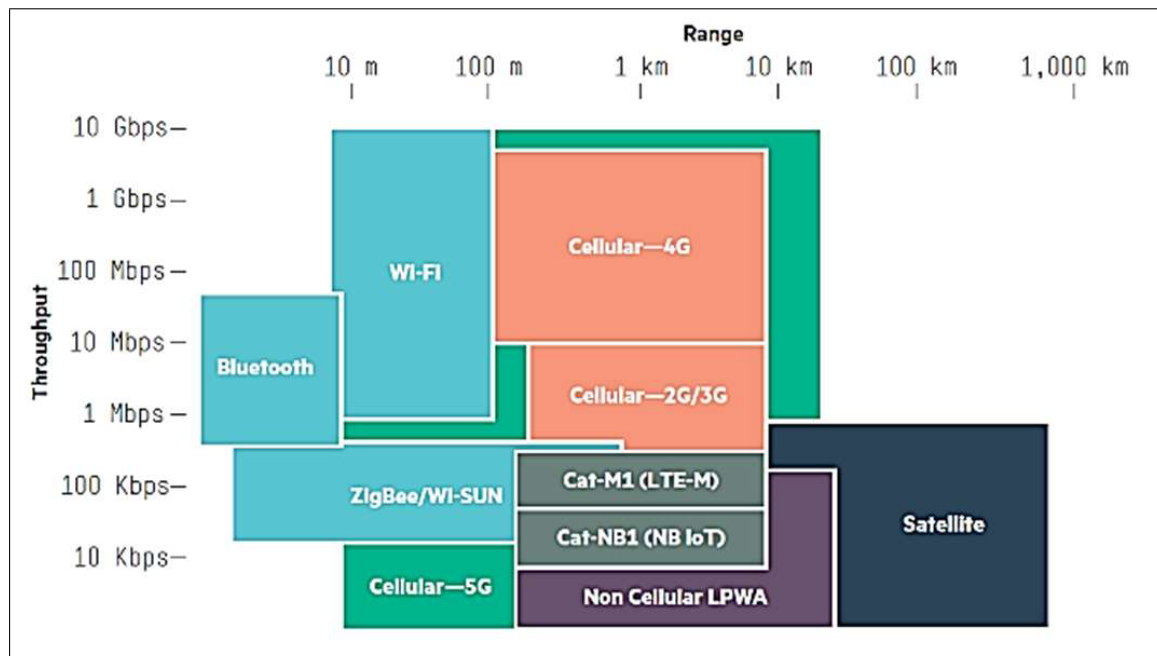


Figura 2 – Alcance e Taxa de Transferência de tecnologias Wireless. Fonte: Poursafar, Alahi e Mukhopadhyay (2018)

A Figura 2, apresenta os valores de taxa de transferência de dados versus alcance de transmissão. Pode-se observar que as tecnologias LPWAN possuem uma taxa de

transferência de dados baixa, contudo seu alcance de transmissão pode chegar a mais de 10km. Além dessas características, o consumo de potência de tecnologias LPWAN é consideravelmente baixo quando comparado com outras tecnologias de longo alcance como 2G, 3G, 4G, LTE e 5G ([POURSAFAR; ALAHI; MUKHOPADHYAY, 2018](#); [QUERALTA et al., 2019](#)).

2.2 Modulação LoRa

A tecnologia LoRa é um padrão de modulação por espalhamento espectral proprietária desenvolvido pela Semtech. LoRa deriva da modulação por espalhamento espectral CHIRP (CSS) ([SPRINGER et al., 2000](#)), tendo como uma das principais características a possibilidade de trocar taxa de dados por sensibilidade dentro de uma largura de banda fixa ([SEMTECH, 2015](#)). Na Figura 3 tem-se a representação de um CHIRP LoRa em relação a sua magnitude x tempo. Segundo [Turcinovic et al. \(2020\)](#), a tecnologia LoRa opera em cerca de 157 países tornando ela a tecnologia *LP-WAN* mais comum em todo o mundo.

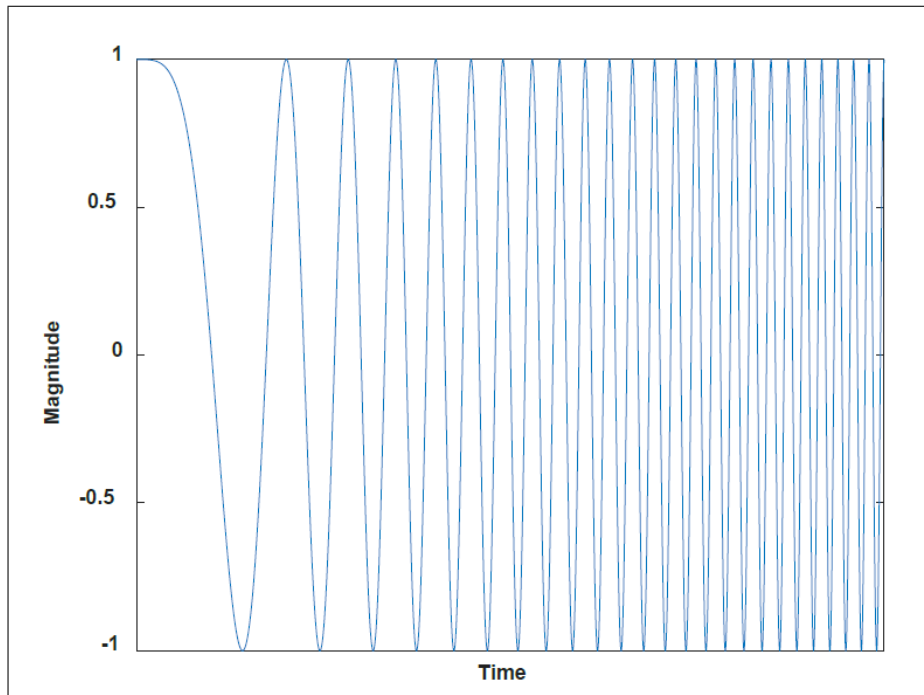


Figura 3 – Representação de um CHIRP LoRa. Fonte: [Montagny \(2021\)](#)

A Figura 4 mostra como é a estrutura da mensagem LoRa, onde pode-se observar um campo inicial chamado Preambulo (*preamble*) que contém alguns símbolos (normalmente oito), indicando atividade no canal para que o receptor se prepare para receber uma mensagem. Em seguida tem-se os campos PHDR e PHDR-CRC que contêm, respectivamente, as informações de cabeçalho da mensagem LoRa e seu CRC para indicar

a integridade da informação transmitida. Após eles, tem-se o *PHYPayload* que de fato contém os dados úteis transmitidos e em seguida tem-se o CRC que garante a integridade do *PHYPayload* transmitido ([ALLIANCE, 2017](#); [MONTAGNY, 2021](#); [SAĞIR et al., 2019](#)).

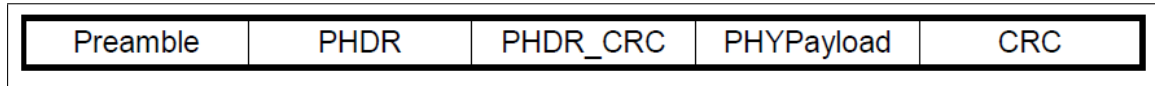


Figura 4 – Estrutura da Mensagem LoRa. Fonte: [Alliance \(2017\)](#)

Segundo [Semtech \(2015\)](#), LoRa é uma implementação em camada física independente de outras camadas mais superiores, assim podendo coexistir com outras aplicações e arquiteturas de rede, como por exemplo o modelo de protocolo proposto por este trabalho.

2.2.1 Largura de banda escalável

A modulação LoRa pode ser escalável tanto em frequência quanto em largura de banda, dessa forma podendo ser utilizada em aplicações de salto em salto de frequência de banda estreita (*narrowband frequency hopping*) ou sequência direta de banda larga (*wideband direct sequence*) ([SEMTECH, 2015](#)).

2.2.2 Baixo consumo

Outra característica importante da modulação LoRa, é seu baixo consumo que se dá por conta de seu esquema de modulação e de envelope constante, ou seja, estágios de PA de alta eficiência, baixo consumo e baixo custo podem ser reutilizados sem modificações, de forma semelhante ao FSK. Contudo, devido ao ganho de processamento associado a modulação LoRa, a potência de saída do transmissor pode ser reduzida em comparação ao FSK, mantendo ou até melhorando a qualidade do enlace ([SEMTECH, 2015](#); [MARAIS; MALEKIAN; ABU-MAHFOUZ, 2017](#)).

2.2.3 Alta robustez

Segundo [Faber et al. \(2020\)](#), a modulação LoRa apresenta alta robustez a presença de altos níveis de ruído. Tal característica se deve pelo alto BT ($BT > 1$) e sua natureza assíncrona, tornando assim um sinal LoRa muito resistente a mecanismos de interferência, dentro e fora da banda, como exposto por [Semtech \(2015\)](#), [Turcinovic et al. \(2020\)](#).

Como o período do símbolo LoRa pode ser mais longo que o de sistemas FHSS de salto rápido, este oferece uma excelente imunidade a mecanismos de interferência de AM pulsado; pode-se obter valores típicos de seletividade fora do canal do receptor na ordem

de 90 dB e rejeição do co-canal melhor que 20 dB, valores bem melhores se comparados com a modulação FSK, que respectivamente, 50 dB e -6 dB ([SEMTECH, 2015](#)).

2.2.4 Resistência a Multi-percurso / Desvanecimento

O pulso CHIRP é de banda larga, assim, a modulação LoRa oferece imunidade a multi-percurso e desvanecimento, tornando essa modulação ideal para uso em ambientes urbanos e suburbanos, onde ambos os mecanismos dominam ([SEMTECH, 2015](#)).

2.2.5 Capacidade de rede aprimorada

A modulação LoRa emprega diversos fatores de espalhamento (*spreading factor*) ortogonais, permitindo que múltiplos sinais de espalhamento sejam transmitidos ao mesmo tempo e no mesmo canal sem degradar minimamente a sensibilidade do receptor, pois sinais modulados em diferentes fatores de espalhamento aparecem como ruído para o receptor alvo, assim sendo tratados como tal.

2.3 Protocolo LoRaWAN

LoRaWAN é um protocolo de rede e sistema de arquitetura desenvolvido pela *LoRa Alliance* para ser utilizado sobre a modulação LoRa ([ZOURMAND et al., 2019](#); [ALLIANCE, 2017](#)). Além disso, LoRaWAN também possui uma camada de controle de acesso de mídia (MAC) que define as regras de acesso a rede, arquitetura, parâmetros de modulação LoRa e frequência de operação LoRaWAN, além de implementar segurança dos quadros transmitidos por meio de um algoritmo AES-128 ([ALLIANCE, 2017](#)). Quanto as frequências de operação, uma rede LoRaWAN, são bem variadas, dependendo da região onde o dispositivo está sendo utilizado. As mais comuns são 433 MHz para Ásia, 868 MHz para Europa e 915 MHz para as Américas. Até a presente data LoRaWAN opera em redes sub-GHz ([ZOURMAND et al., 2019](#); [ALLIANCE, 2020](#)).

Segundo [Sağır et al. \(2019\)](#) e [Montagny \(2021\)](#), LoRaWAN possui três diferentes classes de dispositivos *end-point*, classificados de acordo com o consumo de energia e capacidade de *downlinks*, que é a facilidade com que o usuário pode transmitir um quadro para o dispositivo.

A Figura 5 mostra o modelo de como a pilha LoRaWAN está definida, onde existe uma camada que define os parâmetros de modulação utilizados em cada região do mundo e que são definidos pela LoRa Alliance ([ALLIANCE, 2020](#)) e sobre eles estão os demais parâmetros de modulação LoRa, tais como SF, DR e CR; acima da modulação, têm-se a camada MAC (*Media Access Control*) que define as regras de acesso a rede LoRaWAN e o tipo do dispositivo *end-point* ([ALLIANCE, 2017](#)).

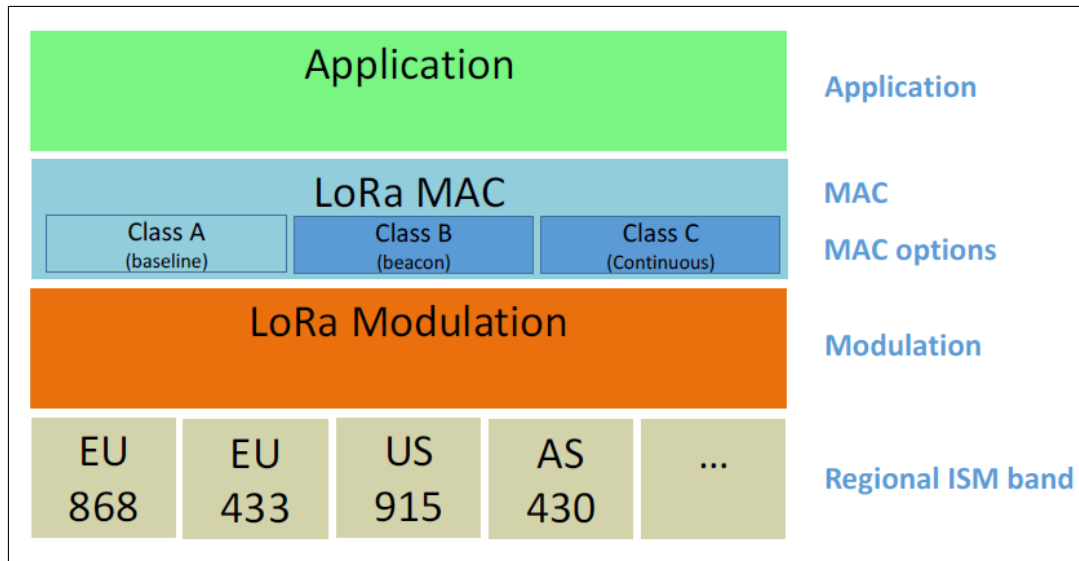


Figura 5 – Pilha do protocolo LoRaWAN. Fonte: Alliance (2017)

2.3.1 Classe A

Por padrão todos os dispositivos LoRaWAN pertencem a classe A (ALLIANCE, 2017; MONTAGNY, 2021), dessa forma todos os dispositivos *end-point* podem transmitir um UPLINK para o Gateway sem a necessidade de verificar a disponibilidade do mesmo (MONTAGNY, 2021; ABDELFADEEL et al., 2020; ALLIANCE, 2017). Esse UPLINK enviado para o LNS por meio do Gateway é seguido pela abertura de duas janelas de recepção por parte do dispositivo *end-point*, como pode ser observado na Figura 6. Importante ressaltar que o LNS só pode enviar um DOWNLINK em uma das duas janelas, não em ambas. As janelas de recepção seguem os padrões regionais definidos pela LoRa Alliance (ALLIANCE, 2020).

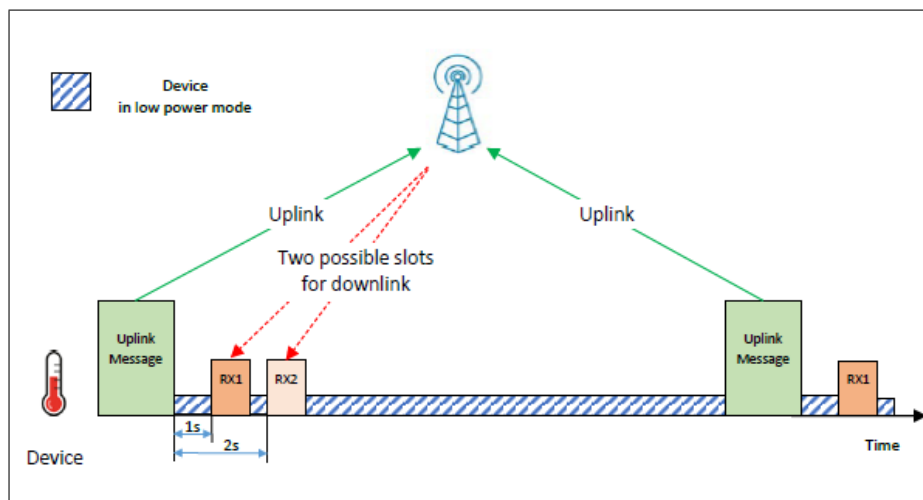


Figura 6 – Slots de recepção para um dispositivo Classe A. Fonte: Montagny (2021)

2.3.2 Classe B

Os dispositivos classe-B se comportam de forma similar aos classe-A ([MONTAGNY, 2021](#)), contudo os dispositivos dessa classes abrem janelas de recepção adicionais de forma periódica, onde esse tempo acordado previa com o gateway, essas janelas são chamadas de *slot de ping* ([ALLIANCE, 2017](#)). Como pode ser observado na Figura 7, o gateway LoRaWAN envia de forma periódica um beacon que sinaliza para os dispositivos classe-B da rede se haverá ou não um DOWNLINK, em caso afirmativo, os dispositivos classe-B abrem janelas de recepção extras para aguardar a chegada do DOWNLINK ([MONTAGNY, 2021](#); [MONTAGNY; AUGAGNEUR, 2022](#); [ALLIANCE, 2017](#)).

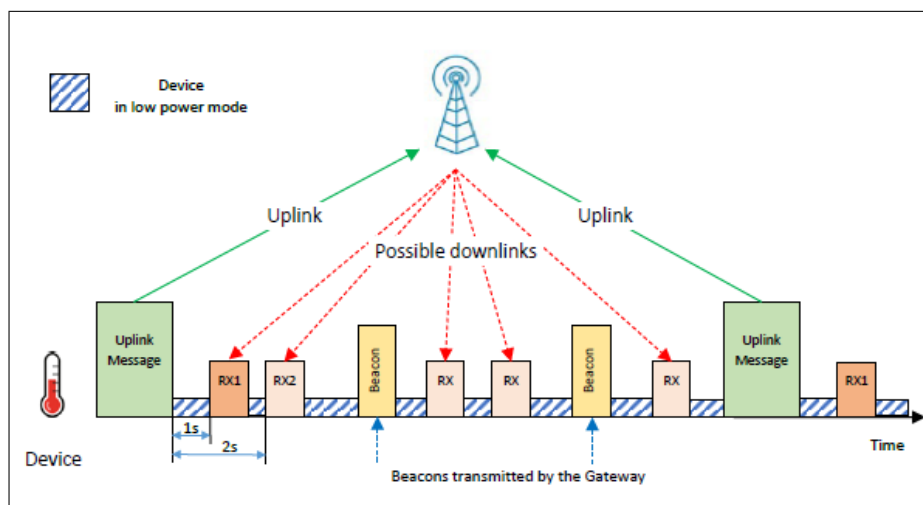


Figura 7 – Slots de recepção para um dispositivo Classe B. Fonte: [Montagny \(2021\)](#)

2.3.3 Classe C

Os dispositivos classe-C herdam as características dos dispositivos Classe-A, tais como abrir a janela de recepção após o envio de um UPLINK para o LNS, contudo a diferença principal nesse tipo de dispositivos é que entre essa janela RX1, como descrito na Figura 8, e o próximo UPLINK para o LNS, os dispositivos da Classe-C deixam uma janela de recepção sempre aberta, tal como pode ser observado na Figura 8, assim esse tipo de dispositivo é normalmente utilizado em aplicações onde não é necessário reduzir o consumo de energia visando autonomia de baterias por meio da minimização do tempo de recepção ([ALLIANCE, 2017](#); [MONTAGNY, 2021](#); [ABDELFADEEL et al., 2020](#); [BEDI et al., 2018](#); [POURSAFAR; ALAHI; MUKHOPADHYAY, 2018](#))

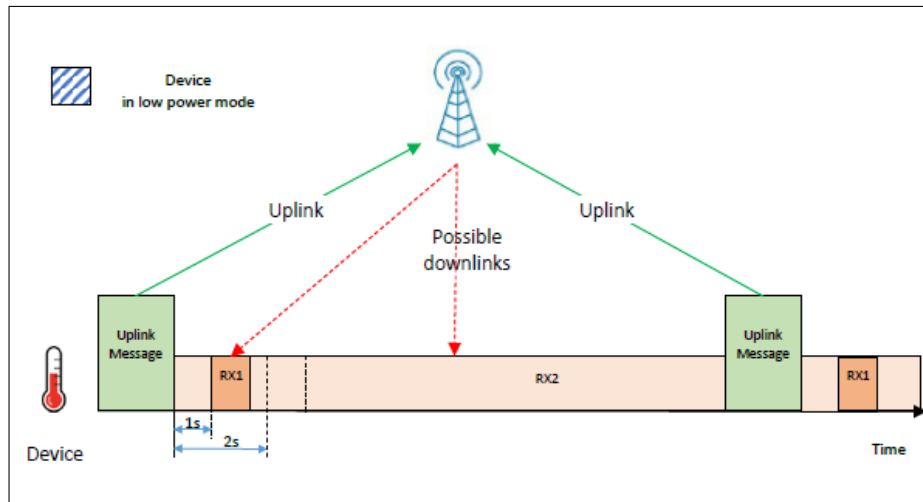


Figura 8 – Slots de recepção para um dispositivo Classe C. Fonte: [Montagny \(2021\)](#)

Na Figura 9 tem-se uma representação de como o consumo de energia aumenta de acordo com a classe do dispositivo *end-point*, onde a Classe-A é o que menos consome energia, pois seus sistemas passam a maior parte do tempo em estados de baixo consumo de energia e com a recepção do radio desligada; já os dispositivos da classe-B consomem mais que os da classe anterior, pois abrem a janela de recepção em períodos pré-acordados com o gateway; por sua vez os dispositivos classe-C são os que mais consomem energia, pois em aplicações que normalmente essa classe é utilizada, o requisito de economia de consumo, normalmente não é um dos principais ou sequer chega a ser considerado ([ALLIANCE, 2017](#); [MONTAGNY, 2021](#); [MONTAGNY; AUGAGNEUR, 2022](#); [BEDI et al., 2018](#)).

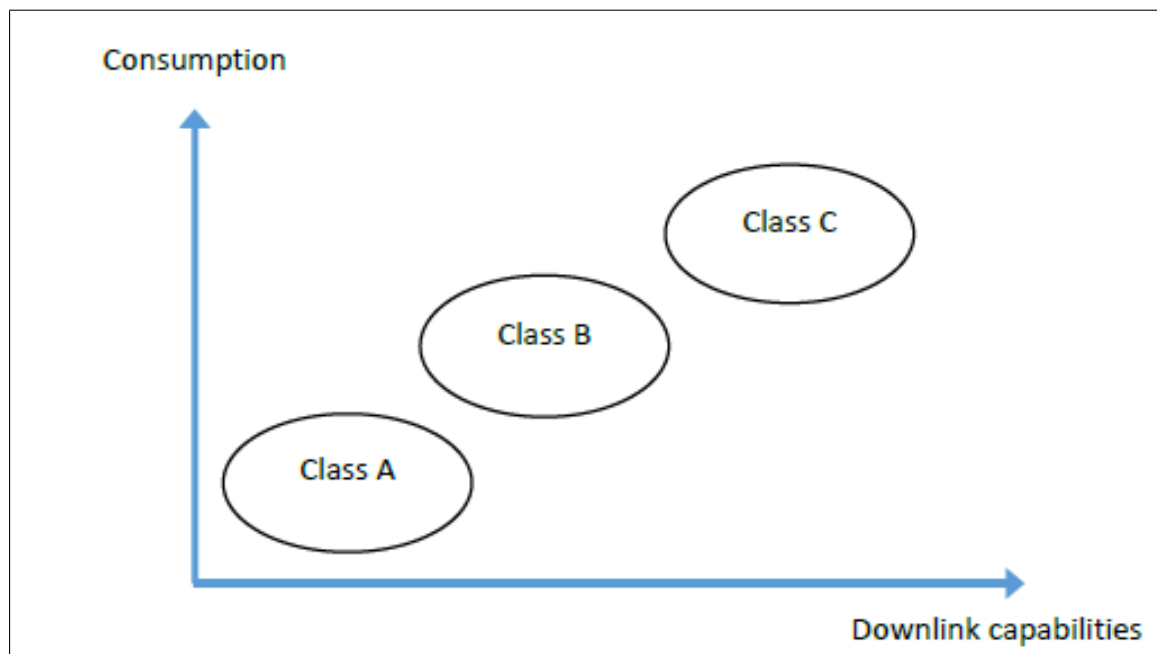


Figura 9 – Potência consumida x capacidade de downlink. Fonte: [Montagny \(2021\)](#)

2.3.4 Gateway e Topologia de Rede LoRaWAN

Na Figura 10 está descrita a topologia de uma rede LoRaWAN, onde têm-se os dispositivos *end-node* transmitindo em um dos canais LoRaWAN pré-determinados conforme a região, esse *UPLINK* chega até um *gateway* LoRaWAN que repassa a mensagem para o LNS que tem como função fazer a primeira interpretação da mensagem LoRaWAN e encaminha-la para o servidor de aplicação correspondente e por último essa informação alcança o usuário por meio de plataformas de IoT ou similares.

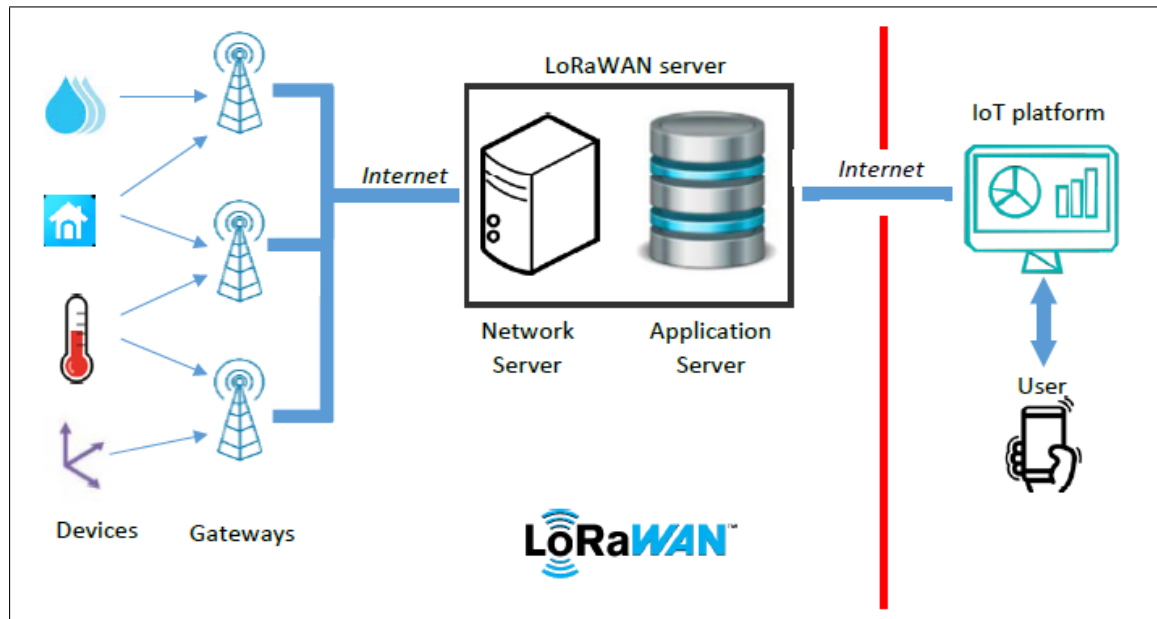


Figura 10 – Visão geral de uma rede LoRaWAN. Fonte: [Montagny \(2021\)](#)

Na Figura 11 está descrita uma visão geral das pilhas que compõem uma rede LoRaWAN padrão. Em destaque tem-se as pilhas que compõem um *gateway* LoRaWAN, essa entidade tem um papel importante nas redes LoRaWAN, pois a comunicação com os servidores de aplicação ocorrem sempre passando por um ele, pois redes LoRaWAN possuem essencialmente a topologia estrela, onde os *gateways* atuam como entidades centralizadoras para um conjunto de dispositivos *end-node* ([ALLIANCE, 2017](#)). Os *gateways* LoRaWAN escutam em todos os canais definidos para a região onde estiverem instalados e também em todos os fatores de espalhamento da modulação LoRa ao mesmo tempo ([MONTAGNY, 2021](#)).

Ainda segundo [Montagny \(2021\)](#), os *gateways* LoRaWAN por um lado recebem mensagens na modulação LoRa, contudo em seu outro lado ele pode retransmitir a mensagem recebida por qualquer meio de internet, tais como Wi-Fi, 3G, LTE, 5G, Ethernet dentre outros. Contudo uma característica importante é a natureza *half-duplex* dos *gateways*, ou seja, enquanto estiverem transmitindo *DOWNLINKS* eles não permanecem aptos a escutar *UPLINKS*, o contrário também é válido, outra característica também importante é

que de acordo com a região onde esteja instalado, o *gateway* deve seguir algumas regras de *duty-cycle* específicas, onde por exemplo na Inglaterra, para cada *DOWNLINK* enviado, o *gateway* precisa ficar em modo de escuta um tempo igual a 99 vezes o tempo gasto para enviar o *DOWNLINK*, sem poder enviar nenhum outro *DOWNLINK* nesse período de escuta ([ALLIANCE, 2020](#)).

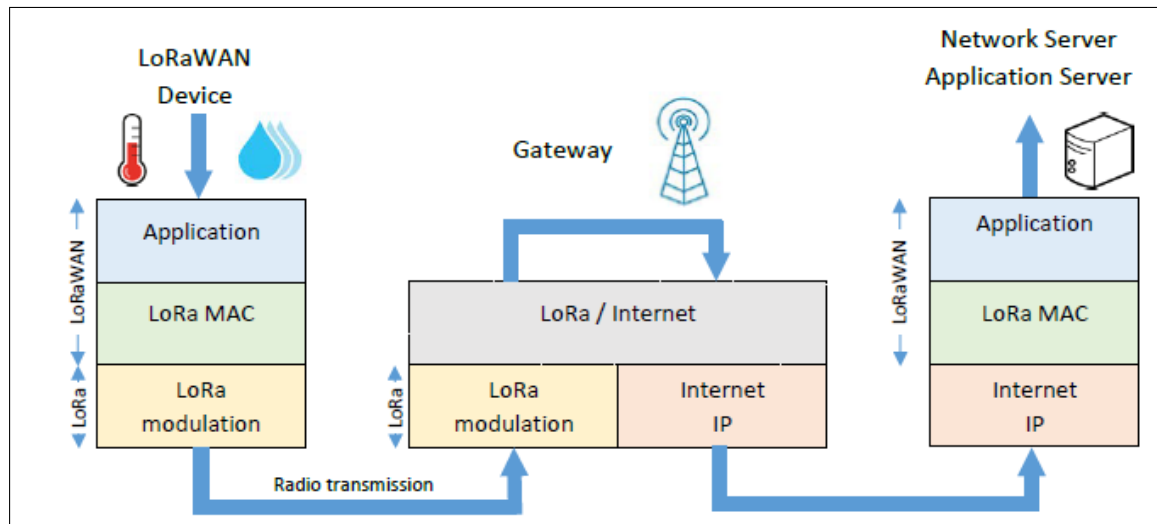


Figura 11 – Visão geral de um Gateway LoRaWAN. Fonte: [Montagny \(2021\)](#)

2.4 FUOTA - *Firmware Upgrade On the Air*

O processo de atualização do *firmware* de um dispositivo por um meio sem fio é chamado de FUOTA, segundo [Abdelfadeel et al. \(2020\)](#) e [Sornin \(2020\)](#) esse processo se tornou uma requerimento chave para a maioria dos dispositivos IoT, sendo uma funcionalidade importante para manter a segurança cibernética, otimização da aplicação instalada e instalação de novas funcionalidades, estendendo desta forma o tempo de vida e melhorando o desempenho de tais dispositivos.

De acordo com [Sornin \(2020\)](#), o processo de atualização de firmware é uma tarefa complexa, que normalmente segue alguns passos como os citados abaixo:

1. Identificação dos dispositivos que precisam ser atualizados e coloca-los em grupos *multicast*.
2. Geração de um arquivo binário contendo a imagem específica dos dispositivos que serão atualizados.
3. Assinar o binário com uma chave-privada para garantir integridade e autenticidade.
4. Publicar o arquivo binário para o grupo de dispositivos que será atualizado.

5. Cada dispositivo deve verificar a autenticidade da assinatura do arquivo binário.
6. Cada dispositivo então deve instalar a atualização de *firmware*.
7. Cada dispositivo deve reportar o estado do processo de FUOTA como sucesso ou falha.

2.5 Trabalhos Relacionados

Nessa sessão está descrito de forma sumarizada dois trabalhos relacionados com o desenvolvimento de métodos de atualização de *firmware* utilizando tecnologias LoRa e LoRaWAN, os trabalhos são respectivamente *How to make Firmware Updates over LoRaWAN Possible* de [Abdelfadeel et al. \(2020\)](#) e *IoT Device Firmware Update over LoRa: The Blockchain Solution* de [Anastasiou et al. \(2020\)](#).

No trabalho de [Abdelfadeel et al. \(2020\)](#), os autores desenvolveram uma ferramenta de simulação chamada *FUOTASim* que tem como objetivo avaliar os melhores parâmetros LoRaWAN para determinar uma melhor estratégia de FUOTA. Ainda no artigo de [Abdelfadeel et al. \(2020\)](#), os autores avaliam as novas especificações de FUOTA propostas pela *LoRa Alliance* ([SORNIN, 2020](#)). Nas Figuras 12 e 13 estão descritos, respectivamente, a arquitetura base de um FUOTA LoRaWAN e o diagrama de sequência para uma sessão multicast de atualização de dispositivos LoRaWAN classe C, ambos utilizados por [Abdelfadeel et al. \(2020\)](#) em seu trabalho.

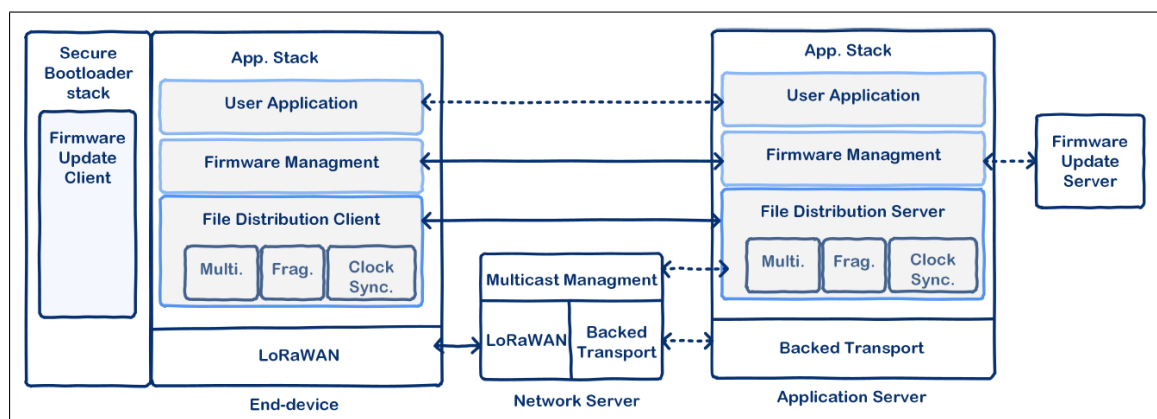


Figura 12 – Arquitetura de um FUOTA LoRaWAN. Fonte: [Abdelfadeel et al. \(2020\)](#)

Na Figura 12 observa-se que a arquitetura de um FUOTA LoRaWAN necessita da presença de um Gateway, um servidor de rede, um servidor de aplicação e um servidor de controle do processo de FUOTA, além da necessidade de mecanismos de fragmentação dos dados, sincronização e controle de sessão multicast.

Na Figura 13 está descrito o processo de uma sessão multicast para dispositivos classe C, conforme avaliado no trabalho de [Abdelfadeel et al. \(2020\)](#). Nota-se que durante o processo de FUOTA são definidas cinco etapas distintas, onde na primeira se faz uma configuração dos dispositivos que pertencerão ao grupo multicast alvo da atualização, seguida da etapa de definição dos parâmetros de fragmentação da imagem do firmware, depois sincronização do tempo dos dispositivos-alvo, pré-etapa de inicialização e por último etapa de transmissão da imagem. Importante notar que em todas as etapas os dispositivos estão em uma rede LoRaWAN, assim utilizando todas as entidades de rede descritas nas Figuras 10 e 12.

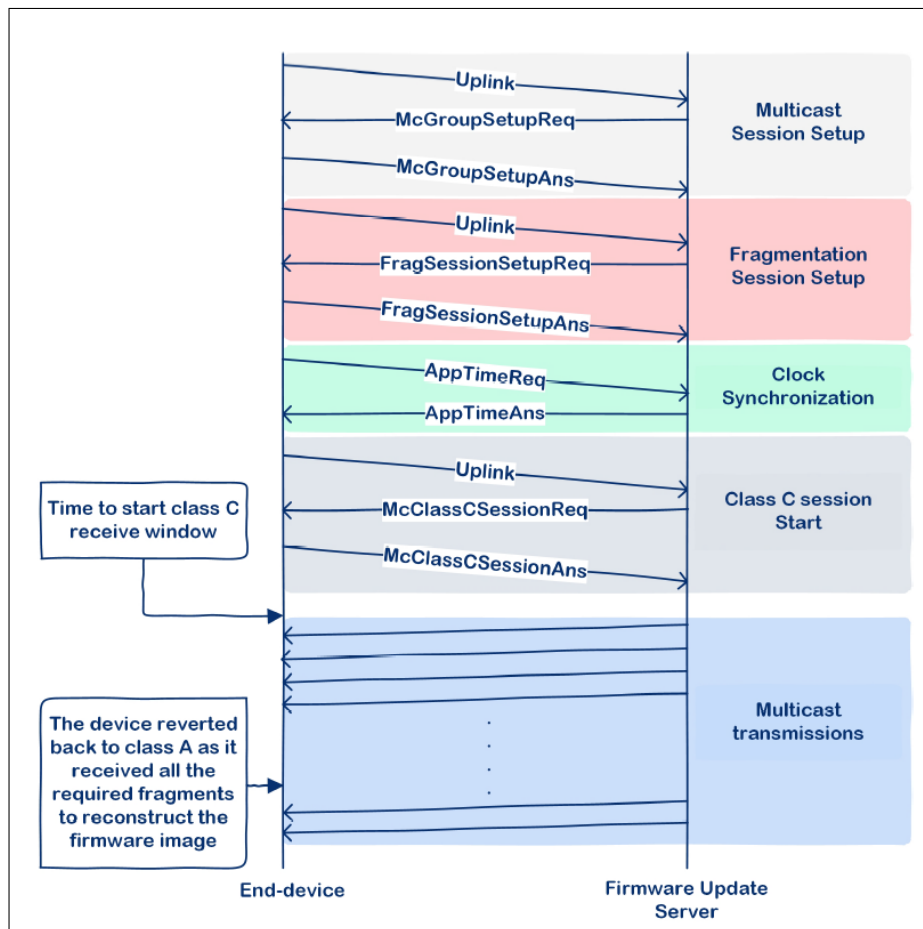


Figura 13 – Diagrama de Sequência de um FUOTA LoRaWAN Multicast Classe C. Fonte: [Abdelfadeel et al. \(2020\)](#)

No trabalho supracitado, os autores relatam que o tempo de transmissão de imagens de firmware da ordem de 50 kBytes levaram cerca de 17 horas, tomando como premissa que o envio de mensagens seria confirmado, além de considerar um cenário ideal onde não houvessem perdas e somente um dispositivo estivesse sendo alvo da atualização.

Nas Figuras 14 e 15 têm-se, respectivamente, o tempo de atualização e eficiência de atualização para um cenário de 100 dispositivos LoRaWAN Classe C, método de envio

de mensagens sem confirmação de recebimento, ambos os resultados obtidos por meio da ferramenta FUOTASim desenvolvida pelo autores ([ABDELFADEEL, 2020](#)).

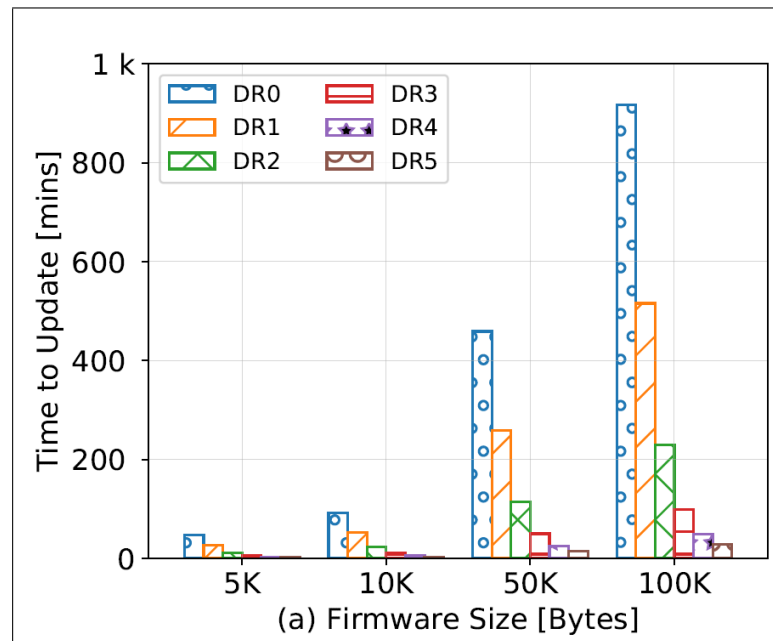


Figura 14 – Tempo de Atualização x Tamanho do Firmware. Fonte: [Abdelfadeel et al. \(2020\)](#)

Importante observar que segundo [Abdelfadeel et al. \(2020\)](#), não foram consideradas perdas de pacote em suas simulações, mesmo neste cenário que não possui confirmação de entrega/recebimento dos pacotes, assim esse gráfico de eficiência mostra somente se o dispositivo-alvo estava dentro do alcance de transmissão e não se a imagem de firmware foi completamente recebida.

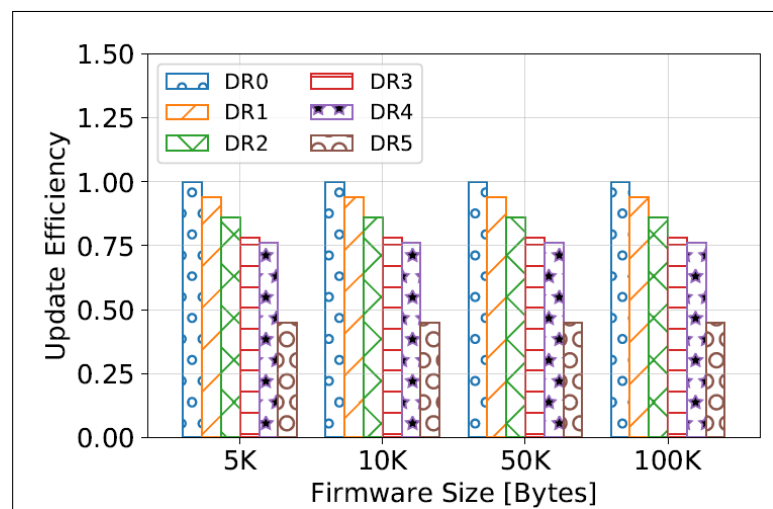


Figura 15 – Eficiência de Atualização Teórica. Fonte: [Abdelfadeel et al. \(2020\)](#)

Observando o diagrama de sequência descrito na figura 13, nota-se que não foi previsto nenhum mecanismo de envio de pacotes pendentes, ou seja, os dispositivos *end-device* não possuem um meio de informar ao servidor de atualização que alguma parte da imagem está faltando, para que assim a mesma seja reenviada.

Como um meio mitigar o tempo elevado de atualização de imagens de firmware, Abdelfadeel et al. (2020) propões que diversos gateways sejam utilizados em uma mesma rede, assim minimizando os impactos relacionados pelo tempo de *duty cycle*, descritos na seção 2.3.4, os autores ainda propõe que estes gateways possam ser móveis, assim sendo deslocados sempre que houver a necessidade de atualização de um determinado grupo de dispositivos.

No trabalho de Anastasiou et al. (2020), os autores propõe o uso de um mecanismo baseado em *block-chain* para incrementar a segurança de um FUOTA. Ainda no artigo são avaliados alguns cenários de FUOTA por meio da ferramenta de atualização FUOTASim, desenvolvida por Abdelfadeel (2020).

Na Figura 16 está descrito o modelo proposto para utilização de BC em um processo de FUOTA, pode-se observar que nesse modelo além das entidades de rede citadas anteriormente na seção 2.3.4, existe um elemento *BlockChain* que é responsável pela criptografia das mensagens durante o processo de atualização.

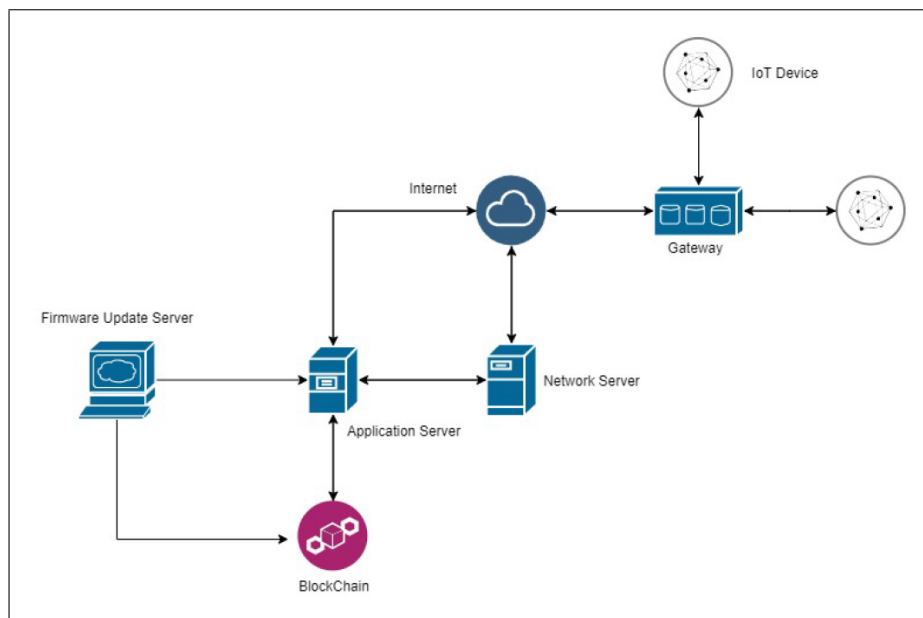


Figura 16 – Modelo Proposto para uso de BC. Fonte: Anastasiou et al. (2020)

Na Figura 17 está o diagrama de sequência proposto e segundo Anastasiou et al. (2020) o mesmo foi baseado no processo de FUOTA LoRaWAN proposto pela LoRa Alliance (SORNIN, 2020). Importante observar que nesse processo de FUOTA, existe uma

etapa de confirmação de recebimento de pacotes, algo que não estava descrito no trabalho de [Abdelfadeel et al. \(2020\)](#), além da etapa de verificação de integridade dos dados por meio do uso de BC.

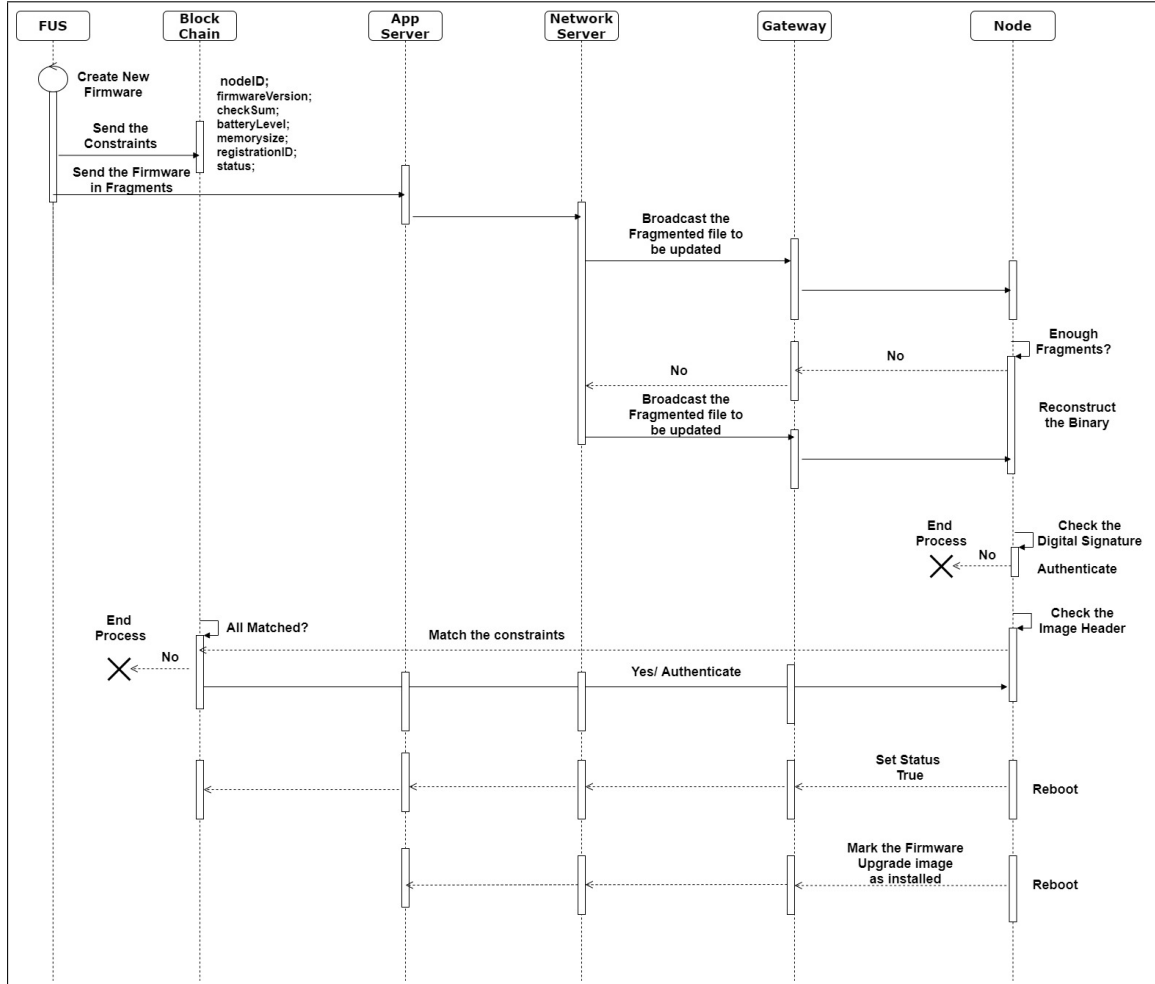


Figura 17 – Diagrama de Sequência com BC. Fonte: [Anastasiou et al. \(2020\)](#)

O cenário definido por [Anastasiou et al. \(2020\)](#) para avaliação do modelo proposto foi composto por 5 sub-cenários alterando o número de *end-device*/nós entre 10, 20, 50, 100 e 200 e todos com uma imagem de firmware com 50 kBytes de tamanho, conforme descrito na Figura 18. Pode-se observar que mesmo para cenários com poucos dispositivos como 10, o tempo de atualização utilizando DR0 chegou a ser aproximadamente de 11,5 dias e em cenários com 100 e 200 dispositivos esse tempo chegou, respectivamente, à 31 dias e 62 dias. Já em DR5, o cenário de 100 dispositivos levou cerca de 15 horas para finalizar o processo e o cenário de 200 dispositivos levou 1,5 dias, ressaltando que em todos esses cenários o gateway fica exclusivamente funcionando somente para o processo de atualização, conforme descrito na Seção 2.3.4. De forma similar ao [Abdelfadeel et al. \(2020\)](#), [Anastasiou et al. \(2020\)](#) conclui que a utilização de mais gateways podem minimizar o tempo de FUOTA.

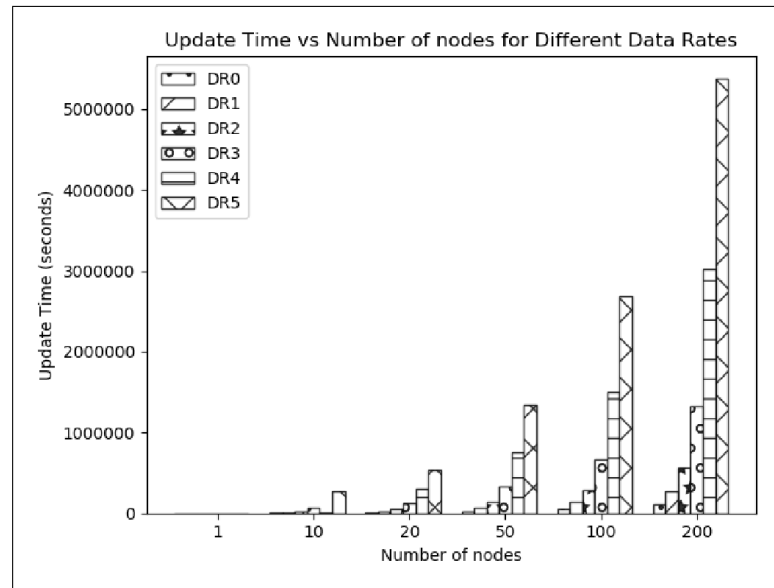


Figura 18 – Tempo de Atualização x Número de Dispositivos. Fonte: [Anastasiou et al. \(2020\)](#)

Outro trabalho que também utiliza BC para o processo de FUOTA é o de [Mtetwa et al. \(2020\)](#), contudo em seu trabalho foram avaliados cenários de transferências de imagem de até 5 kbytes de tamanho. O modelo proposto por [Mtetwa et al. \(2020\)](#) é muito parecido com o modelo de [Anastasiou et al. \(2020\)](#).

3 Materiais e Métodos

3.1 Materiais

3.1.1 Linguagem de Programação e IDE utilizadas

Para o desenvolvimento do código-fonte do FUOTA proposto foi utilizado a linguagem de programação C, criada em 1970 por Dennis Ritchie (SCHILDT, 1990) e como ambiente de programação foi utilizado a IDE *Atollic TrueSTUDIO*, que foi desenvolvida sobre a *Eclipse IDE* pela *ST Microelectronics* e suporta códigos nas linguagens C/C++ (STMICROELECTRONICS, 2023). Nas Figuras 19 e 20 está demonstrado as informações sobre a IDE *Atollic TrueSTUDIO* e a tela de trabalho onde são codificados os arquivos de código-fonte.

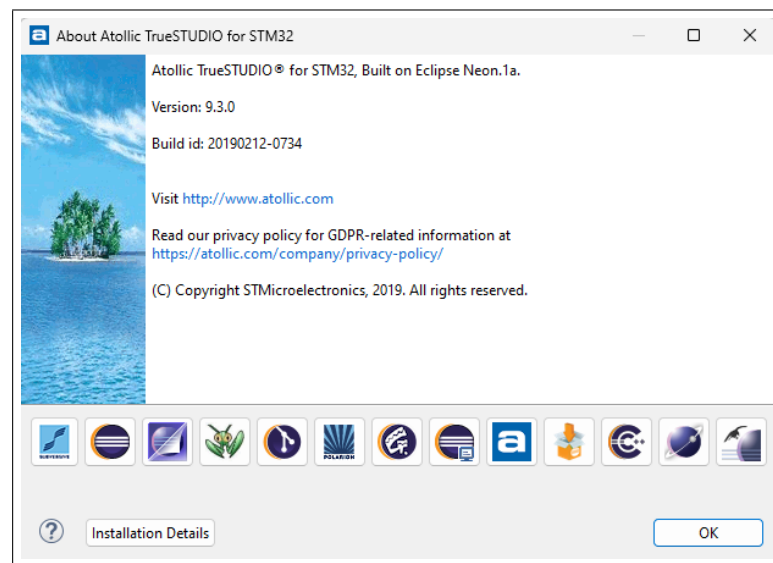


Figura 19 – Tela de informações sobre a IDE Atollic utilizada no desenvolvimento. Fonte: STMicroelectronics

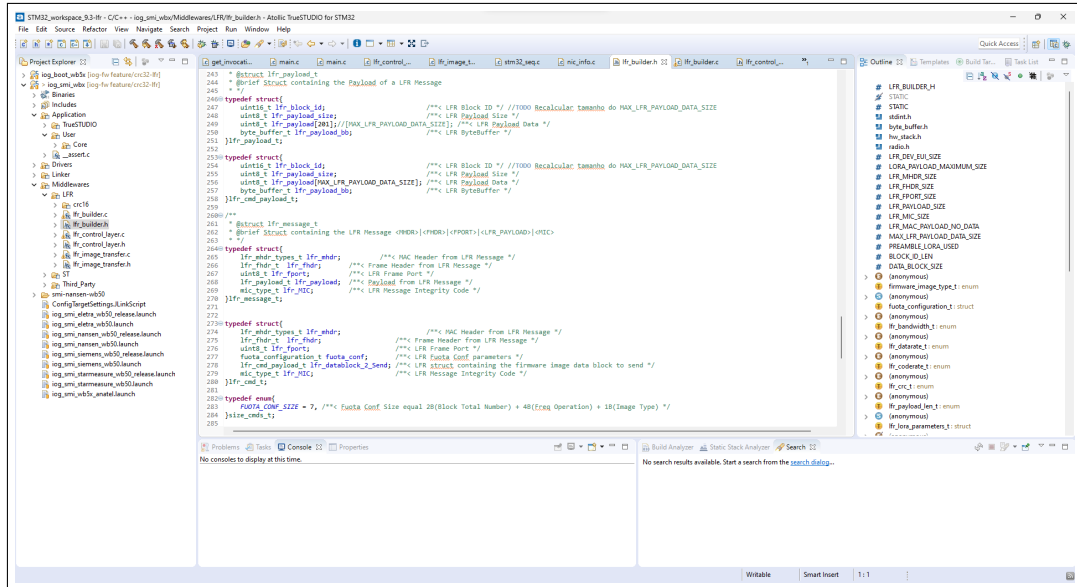


Figura 20 – Janela de trabalho da IDE Atollic. Fonte: STMicroelectronics

Para gravação do código-fonte desenvolvido foi utilizada o J-link, que é uma ferramenta de gravação e debug criada pela Segger (SEGGER, 2023). Na Figura 21 podemos ver a ferramenta de gravação J-link.



Figura 21 – Segger J-link. Fonte: SEGGER

3.1.2 Hardware utilizado

Para o desenvolvimento do FUOTA proposto neste trabalho foi utilizado um hardware composto basicamente de um microcontrolador Arm Cortex-M4 ST32WB55 (STMICROELECTRONICS, 2022) e um transceptor LoRa SX1262 (SEMTECH, 2021). Na Figura 22 observa-se a placa de circuito impresso (PCI) utilizada para testar a implementação do modelo. Essa placa foi desenvolvida em parceria com o Núcleo de Pesquisas do Norte e Nordeste - NEPEN.

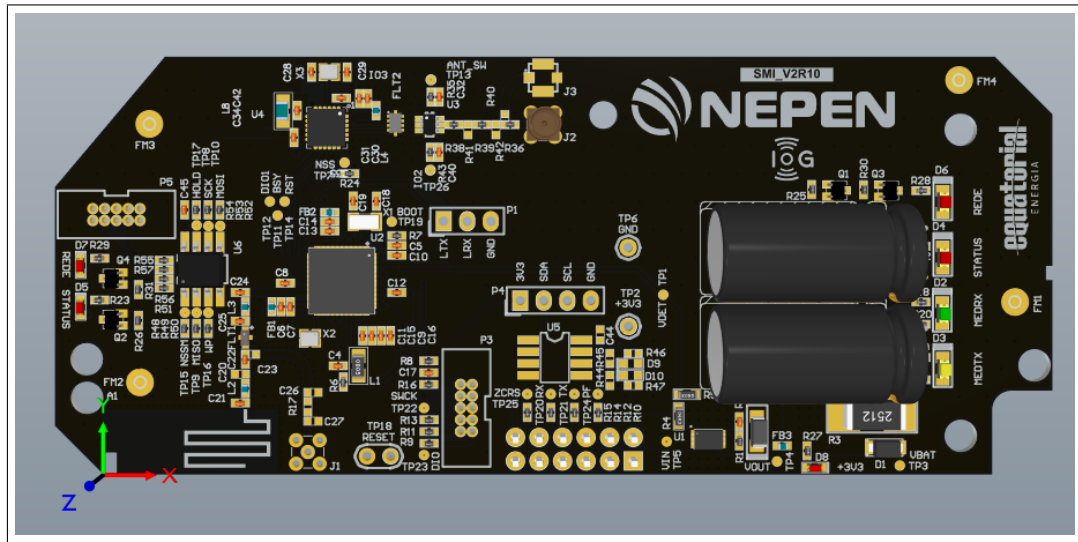


Figura 22 – Placa de circuito impresso com o sistema de hardware IoT LoRa. Fonte: Autor

A comunicação entre o MCU ST32WB55 e o rádio SX1262 foi feita por meio de uma interface SPI e a comunicação entre medidor e PCI foi feita por meio de uma interface UART.



Figura 23 – Medidores de energia monofásicos Eletra (à esquerda) e Nansen (à direita).
Fonte: Autor

A antena utilizada pela PCI foi uma antena externa omnidirecional que opera na faixa de 900 a 930 MHz, possui ganho de 2,2dBi e impedância de 50Ω . Na Figura 24 está disposta uma imagem da antena utilizada.



Figura 24 – Antena Omnidirecional Miot. Fonte: Miot Wireless Solutions.

3.2 Metodologia

3.2.1 Topologia de Rede Proposta

Nesta secção está descrita a arquitetura de rede para o protocolo proposto, doravante, o modelo e o protocolo serão chamados de LFR.

As redes LFR são normalmente dispostas na topologia em árvore, ou hierárquica, onde uma semente-mestre (*master-seed*) transmite as mensagens para os dispositivos localizados nos galhos (níveis hierárquicos) abaixo, sendo que alguns destes dispositivos podem vir a ser sementes-futuras (*future-seeds*) assim dando continuidade e criando sub-redes (ver Secção 3.2.2). Uma característica das redes LFR é que embora dispostas em árvore, elas são dinâmicas, permitindo que os dispositivos mudem de nível hierárquico e/ou mudem de galhos (*branches*). Importante ressaltar que diferente do protocolo LoRaWAN, o protocolo LFR dispensa a necessidade de *Gateways*, uma vez que sua topologia é em árvore (TANENBAUM, 2002) e não estrela de estrelas como no caso do LoRaWAN (MARAIS; MALEKIAN; ABU-MAHFOUZ, 2017).

Os parâmetros de modulação *LoRa* podem ser configurados conforme o padrão de parâmetros regionais para o lugar onde os dispositivos estiverem instalados, dessa forma a frequência do canal utilizado pode ser escolhida conforme a documentação definida pela *LoRa Alliance* (ALLIANCE, 2020). Nesse trabalho foram escolhidos os canais que compreendem a faixa de 902MHz à 928MHz, que possui frequência nominal de 915MHz, frequência essa utilizada pelos dispositivos *LoRa/LoRaWAN* nas Américas

3.2.2 Modelo de FUOTA Proposto

Nesta secção está descrito as especificações do modelo LFR incluindo formato das mensagens e diagramas de sequência. Na Figura 25 estão ilustradas as pilhas LoRaWAN e LFR. Comparando ambas pode-se observar que a proposta do protocolo LFR é substituir a camada MAC LoRaWAN durante o processo de FUOTA, assim redefinindo os parâmetros de comunicação e modulação do rádio SX para o padrão LFR pre-determinado, mas por outro lado ainda mantendo as especificações regionais definidas pela *LoRa Alliance*.

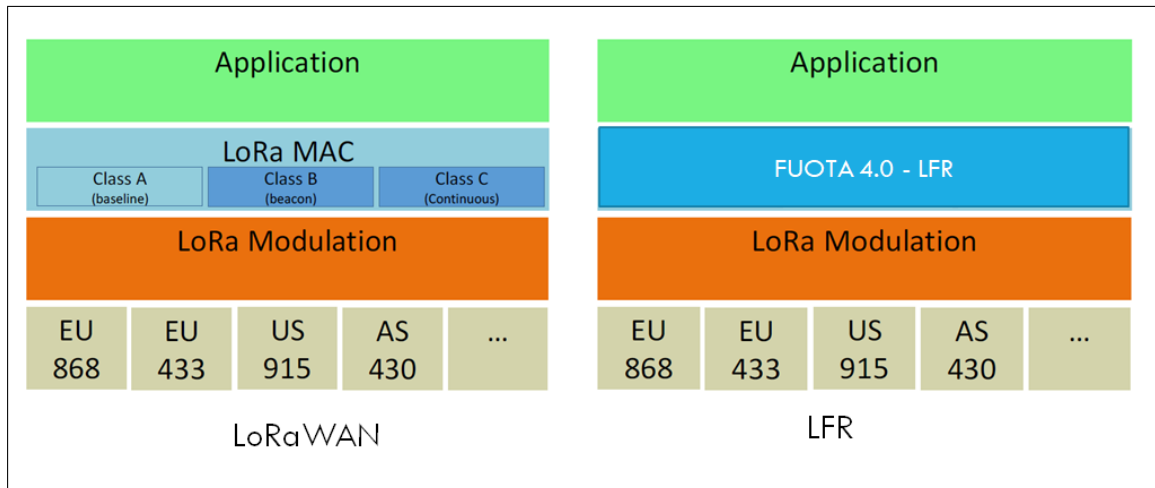


Figura 25 – Comparação entre as pilhas LFR e LoRaWAN. Fonte: Autor

Na Figura 26 está descrito o processo de FUOTA proposto por este artigo. O modelo consiste de quatro etapas: *Setup*, *Update*, *Future Seeds* e por último *Common end-node*.

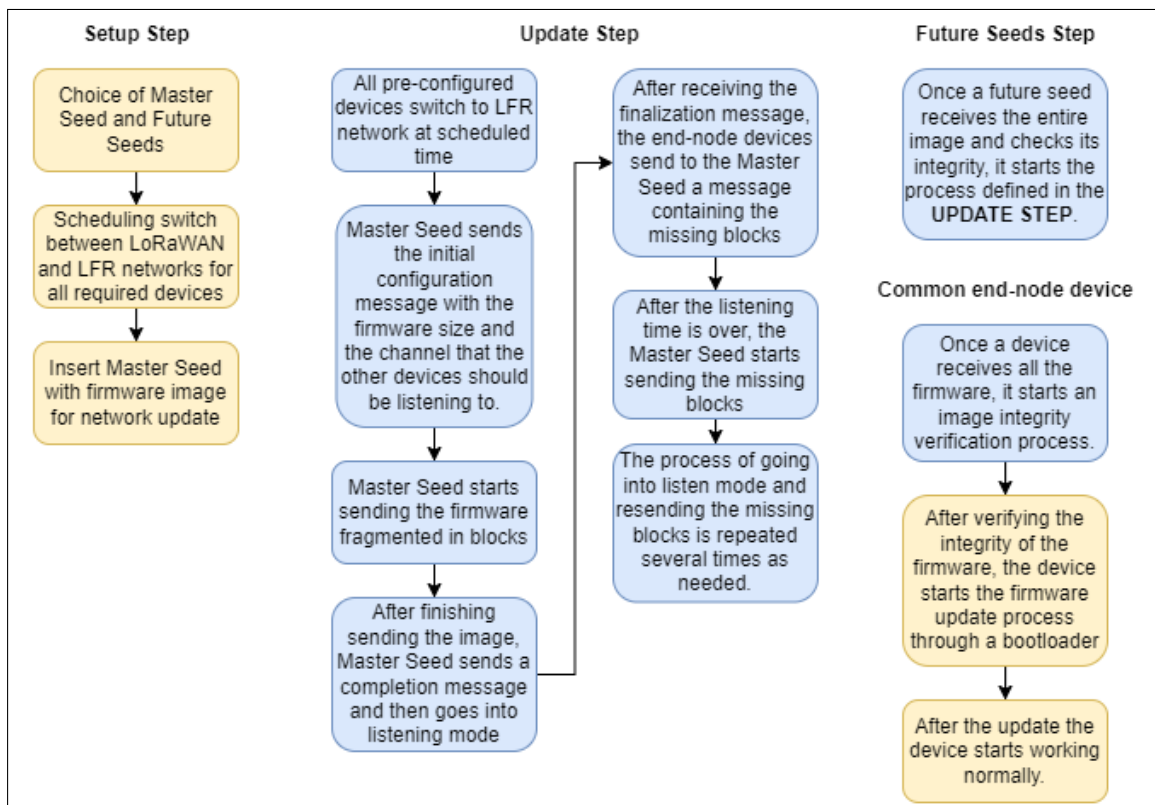


Figura 26 – Modelo de Fuota Proposto - LFR. Fonte: Autor

3.2.2.1 Estrutura de Payload Proposto para o Protocolo

A Figura 27 apresenta a estrutura de dados de uma mensagem LFR, toda esse quadro está contido dentro do PHYPayload, na Figura 4.

MHDR	MACPayload			MIC
	FHDR	Fport	LFRPayload	
< 1 byte >	< 16 bytes >	< 1 byte >	< 200 bytes >	< 4 bytes >

Figura 27 – Estrutura da mensagem LFR. Fonte: Autor

- **MHDR:** Este é o campo inicial da mensagem LFR e sua função é indicar o tipo da mensagem LFR: configuração, blocos do *firmware* alvo, finalização de atualização, pacotes pendentes. Esses tipos são melhor explicados na Seção 3.2.2.3.
- **MACPayload:**
 - **FHDR:** Este campo contém os campos de identificação dos dispositivos na rede, onde os primeiros 8 bytes indicam o endereço da origem e os 8 bytes finais o endereço do destino.
 - **Fport:** Este campo contém um byte para indicar em qual porta a mensagem está sendo enviada, portas diferentes podem ser utilizada para funções diferentes do protocolo.
 - **LFRPayload:** Este campo contém os bytes úteis da mensagem, que podem ser informações de configuração dos receptores como frequência e largura de banda, tamanho total da imagem do *firmware* alvo, ou os blocos da imagem.
- **MIC:** Este campo contém o mecanismo de integridade das mensagens, que é calculado para todos os campos da mensagem LFR e toma como base a RFC4493.

3.2.2.2 Etapa de Configuração - Setup

Nesta etapa são configuradas as especificações que a rede LFR irá operar. Para esta etapa são necessários o envio de dois comandos via *LoRaWAN* para todos os dispositivos que forem operar na rede LFR. Os comandos são os seguintes:

- **SetLfrRole:** comando responsável por definir qual o papel do dispositivo LoRa na rede LFR.

- Common-Device: papel atribuído ao dispositivo da rede LFR que somente recebe a imagem de *firmware* e se atualiza quando a mesma é enviada por completo.
 - Future-seed: papel atribuído ao dispositivo da rede LFR que após receber uma imagem de *firmware* válida, inicia o processo de envio da mesma para os dispositivos de sua sub-rede LFR.
 - Master-Seed: Papel atribuído ao dispositivo da rede LFR que inicia o envio da imagem de *firmware* tão logo o mesmo comute para a rede LFR.
- **SwitchSchedulingForLfr**: Comando responsável por definir o horário de agendamento para a mudança de rede dos dispositivos, entre LoRaWAN e LFR. Esse comando contém o horário inicial de mudança para a rede LFR e o horário final de retorno para a rede LoRaWAN pré-definida no dispositivo. Importante notar que todos os dispositivos retornam para a rede LoRaWAN, ao final do agendamento, independente de terem ou não concluído sua transmissão e/ou recepção da imagem de *firmware*.

3.2.2.3 Etapa de Transmissão e Atualização - *Update*

Nesta etapa todos os dispositivos pré-configurados para operarem na rede LFR, mudam para a rede LFR no horário de agendamento definido. Seguindo o fluxo definido na Figura 26, a *Master-seed* inicia o envio do comando `FuotaConfiguration`, contendo a frequência de operação da transmissão da imagem e o tamanho da imagem em blocos. Posteriormente são enviados os comandos `FirmwareUpdateMessage` contendo os blocos com a imagem de *firmware* a ser transferida.

Após o envio de toda a imagem, a *Master-seed* envia o comando `DownloadedFirmware` e entra no modo de escuta por um tempo pré-determinado. Uma vez que os dispositivos da rede que estavam recebendo a imagem recebem o comando de conclusão do envio, os mesmos analisam se toda a imagem foi transferida, caso algum bloco esteja faltando os dispositivos enviam para a *Master-seed* um comando `FirmwarePendingBlocks` contendo um mapa de *bits* que representa todos os blocos da mensagem e indica quais desses estão pendentes.

Uma vez que o tempo de escuta termine a *Master-seed*, analisa todas as mensagens de blocos pendentes que foram recebidas e inicia um novo processo de envio das mensagens contendo os blocos da imagem de *firmware*, contudo, somente para os blocos que foram sinalizados como pendentes.

Esse processo se repete por todo o horário de agendamento da rede LFR que foi definido durante a etapa de configuração. A Figura 28 exhibe um diagrama do processo explicado.

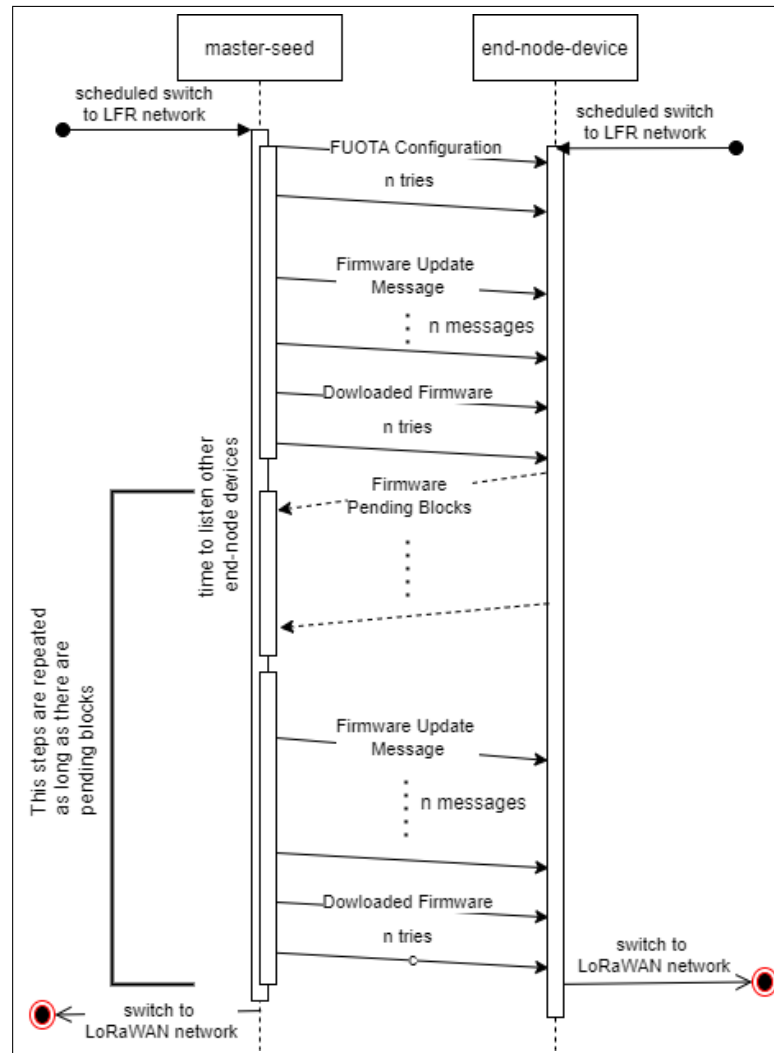


Figura 28 – Diagrama de Sequência Unicast para FUOTA LFR. Fonte: Autor

Abaixo estão descritos os comandos LFR utilizados durante a etapa de transmissão e atualização.

- **FuotaConfiguration:** Comando responsável por realizar a configuração inicial da transmissão da imagem de *firmware*. Ele configura a frequência de operação da rede LFR durante o processo de FUOTA e informa aos dispositivos comuns (*Common-Device*) qual o tamanho da imagem que será transferida em números de blocos. Importante ressaltar que o tamanho do bloco padrão é definido durante a etapa de projeto e desenvolvimento do código-fonte.
- **FirmwareUpdateMessage:** Comando que contém os blocos com os fragmentos da imagem do *firmware*, além de conter também um identificador do bloco, para que o código-fonte da aplicação do *Common-device* possa identificar qual o bloco está sendo recebido.

- **DownloadedFirmware:** Comando que indica que a transferência de toda a imagem do *firmware* foi concluída.
- **FirmwarePendingBlocks:** Comando enviado pelo *Common-device* para a *Master-seed* ou *Future-seed* da rede LFR. Esse comando indica quais blocos da imagem do *firmware* estão pendentes de serem recebidos pelo *Common-device*.

3.2.2.4 Etapa de ativação da *Future Seed*

Nesta etapa todos os dispositivos que foram configurados com o papel *Future-seed* verificam a integridade da imagem de *firmware* enviada, de forma análoga aos *Common-device*, contudo em vez de inicializar o processo de atualização, estes por sua vez iniciam o processo de Transmissão, conforme descrito na secção 3.2.2.3.

Observação importante, nessa nova etapa de transmissão o papel da *Master-seed* será realizado pela *Future-seed* para a nova sub-rede LFR. Na Figura 29 pode-se observar o diagrama de sequência para um processo de FUOTA *Multi-layer*, onde existem *Future-seeds* na rede LFR.

3.2.2.5 Etapa de atualização do *Common End-node Device*

Nesta etapa todos os dispositivos que foram configurados com o papel *Common-device* verificam a integridade da imagem de *firmware* enviada e uma vez que esta imagem esteja íntegra o dispositivo inicia o processo de atualização, que consiste basicamente de reiniciar o dispositivo para que o código-fonte do *bootloader*, sub-programa responsável por instalar novas imagens no processador, possa atualizar o dispositivo com a nova imagem recém recebida. Após essa atualização o dispositivo retorna a rede LoRaWAN e volta ao seu funcionamento normal de acordo com a aplicação de cada dispositivo.

A diferença básica entre os processos *uni-layer* e *multi-layer* é a inclusão de dispositivos que possuem o papel de *future-seed* na rede, pois a partir desses novos dispositivos que são incluídas as novas camadas na rede, como pode ser observado nas raias *end-node-device* 2, 3 e 4 na Figura 29. No processo de FUOTA LFR assume-se que todos os processos são *multicast*, tendo em vista que uma transmissão por rádio-frequência é por sua natureza uma transmissão *multicast*, embora no LFR os dispositivos ainda podem ser agregado ou separados por grupos de agendamento como pode ser visto na secção 3.2.2.2.

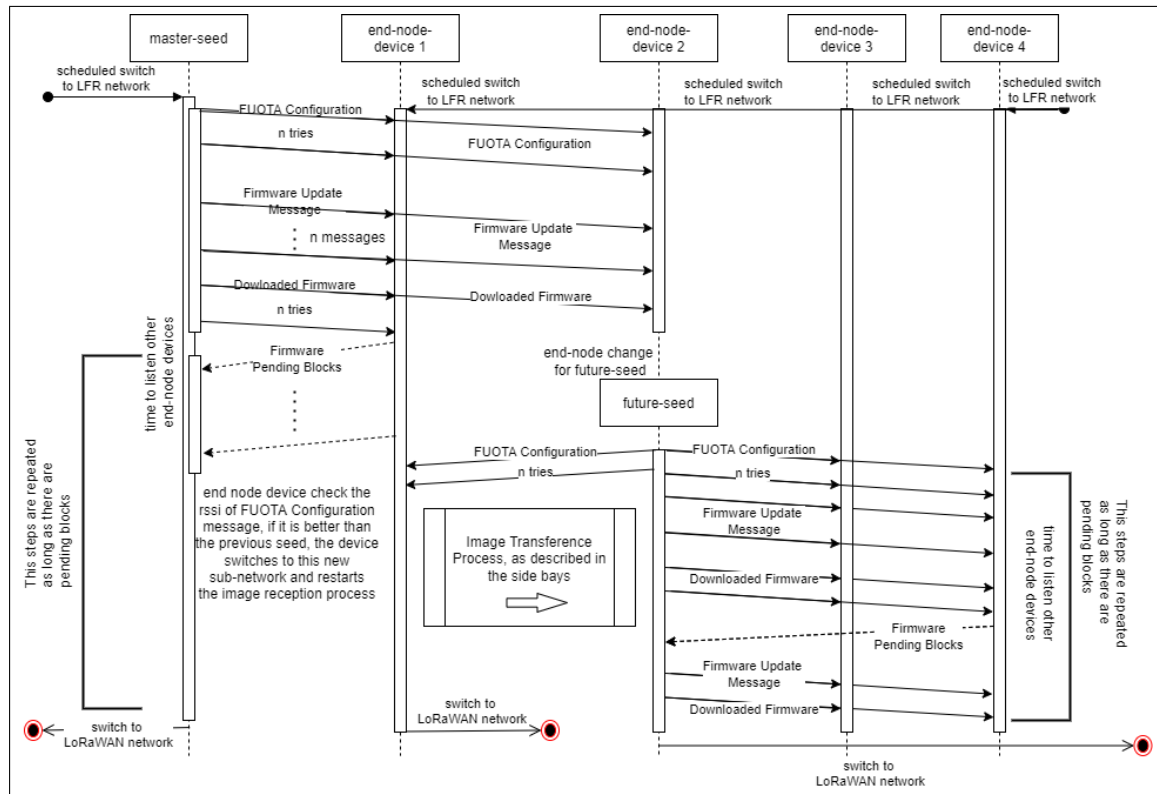


Figura 29 – Diagrama de Sequência Multi-layer para FUOTA LFR. Fonte: Autor

3.2.2.6 Máquina de Estado do Modelo

Nesta secção está descrito o fluxo de estados que os dispositivos podem assumir durante o processo de FUOTA LFR, considerando o papel de cada *end-node device*, conforme descrito em 3.2.2.2.

Na Figura 30 está um fluxograma com os estados que os dispositivos podem assumir durante um processo de FUOTA LFR. Como pode ser observado, os dispositivos inicialmente assumem o estado de *LoRaWAN Device*, independente de seu papel; nesse estado os dispositivos ainda operam na rede LoRaWAN e aguardam o horário pré-definido do agendamento do processo de FUOTA LFR para mudarem de estado. Uma vez que o horário de agendamento seja alcançado, o estado dos dispositivos pode mudar conforme seu papel.

- Se **Master-Seed**: o dispositivo assume o estado de **Master-Seed Ready** e inicia o envio das mensagens de configuração da rede LFR, como descrito na secção 3.2.2.3. Uma vez concluído o envio das mensagens de configuração, o dispositivo muda para o estado **Master-Seed Sender**, nesse estado o dispositivo inicia o envio da imagem de *firmware* definida para o processo de FUOTA atual, ao finalizar o envio de todos os blocos da imagem, o dispositivo envia uma mensagem de finalização como descrito na secção 3.2.2.3 e depois altera seu estado para **Master-Seed Listener**.

No estado de *listener* o dispositivo passa a escutar as mensagens de blocos pendentes de *firmware* que cheguem no canal definido para o processo de FUOTA, como descrito na secção 3.2.2.2, após receber a primeira mensagem de blocos pendentes, o dispositivo aguarda por um tempo pré-determinado para então voltar para o estado *Master-Seed Sender* e assim repetir todo o processo. Importante observar que caso nenhuma mensagem de blocos pendentes chegue ao *Master-Seed*, este por sua vez vai aguardar no estado de *listener* até o fim do agendamento do processo de FUOTA LFR.

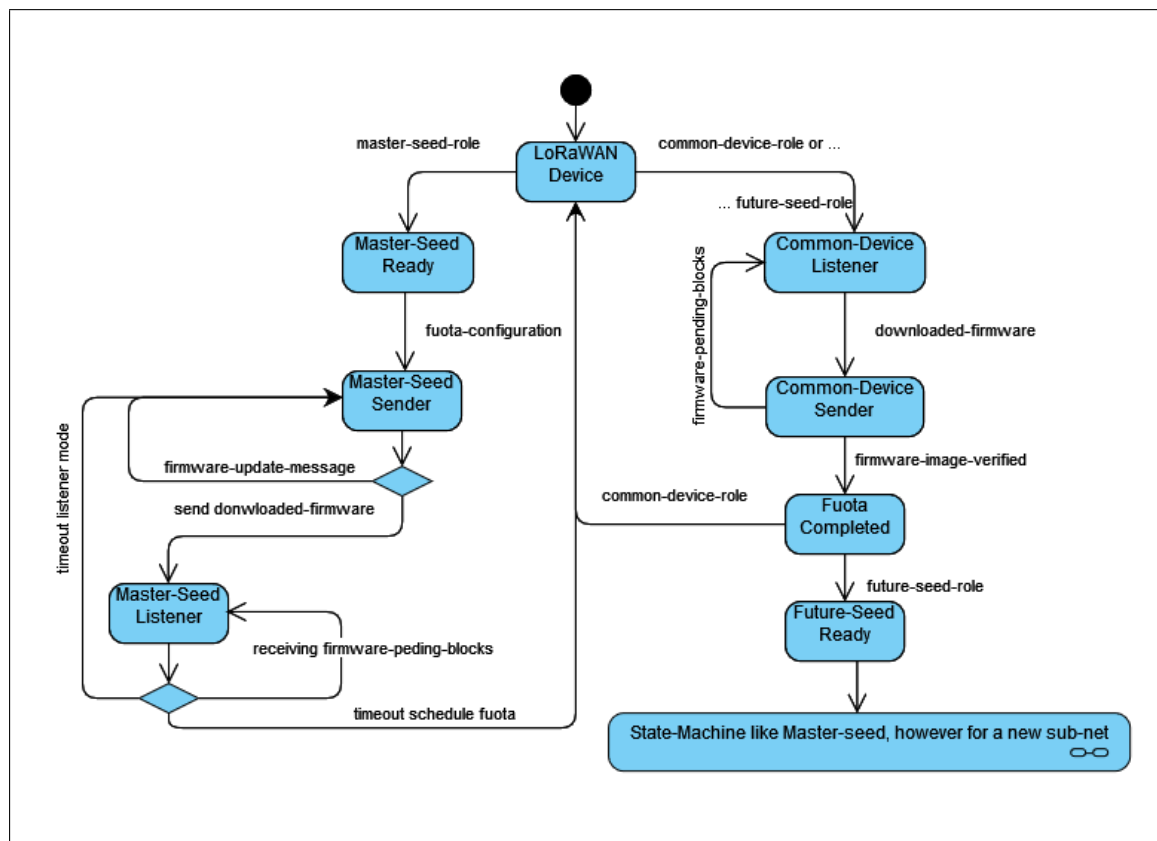


Figura 30 – Máquina de Estado do Modelo. Fonte: Autor

- Se **Common-Device** ou **Future-Seed**: o dispositivo assume o estado *Common-Device Listener* e inicia o processo de escutar as mensagens que cheguem no canal definido para o processo de FUOTA, o dispositivo permanece nesse estado até receber a mensagem finalizadora, que indica que o dispositivo *Master-Seed* ou *Future-Seed* da rede finalizou o envio de todos os blocos do *firmware*. Após receber a mensagem finalizadora do envio de blocos o dispositivo muda para o estado *Common-Device Sender*, nesse estado o dispositivo verifica se existe algum bloco faltante no *firmware* e informa ao *Master-Seed*, no caso de não existir blocos faltantes, o dispositivo verifica a integridade da imagem *firmware* transferida, caso a imagem esteja íntegra,

o dispositivo muda seu estado para *Fuota Completed*, caso não esteja integra, o dispositivo apaga todos os blocos recebidos e envia a mensagem de blocos faltantes para o *Master-Seed* indicando que todos os blocos estão faltantes, para que assim todos os blocos sejam reenviados e após isso volta para o estado de *Common-Device Listener*. A partir do estado de *Fuota Completed*, o dispositivo finaliza o processo de FUOTA LFR instalando a atualização recebida e retornando para a rede LoRaWAN no caso do seu papel ser um *Common-Device*, agora no caso de seu papel ser *Future-Seed*, o dispositivo muda para uma máquina de estado similar ao do processo da *Master-Seed*.

3.2.3 Avaliação do Modelo

Para a avaliação preliminar do modelo proposto, foram realizadas duas etapas de testes. A primeira em laboratório e a segunda com os dispositivos localizados em campo em locais próximos entre si. A Figura 31 mostra a locação dos dispositivos do teste de campo. Ambos os testes consistiram do envio de duas imagens de *firmware*, uma com o tamanho de 196kB e outra com o tamanho de 260kB. A variável utilizada para validar o modelo proposto nesse trabalho foi o tempo de transferência da imagem, mesma variável utilizada nos trabalhos relacionados citado na Secção 2.5. Contudo também foram utilizadas as seguintes variáveis auxiliares para medir o desempenho da transferência: *RSSI* (*Received Signal Strength Indicator*) que mede a força do sinal recebido, o *SNR* (*Signal-to-Noise Ratio*) que indica a qualidade do sinal recebido por meio de uma taxa calculada entre a potência do sinal e do ruído de fundo.

Tabela 1 – Tempo de transferência da imagem via LFR

Tamanho da Imagem	SF10	SF12
100kB (indoor)	31 min	41 min
100kB (outdoor)	35.5 min	47 min
260kB (indoor)	41.2 min	54.5 min
260kB (outdoor)	47.5 min	63.2 min

Os testes de laboratório foram realizados com o seguinte cenário: foram colocados dois dispositivos com apenas alguns metros de distância entre si, onde um deles assume o papel de *Master-seed* e o de *Common-device*, em seguida foi realizado o processo de *FUOTA* conforme definido na Secção 3.2.2.

Tabela 2 – Medidas de Desempenho e distância entre os sítios

Sítio	RSSI Médio	SNR Médio	SF	Distância
1	-98dBm	-8dB	10	143m
2	-99dBm	-4dB	10	97m
3	-91dBm	5db	10	53m
4	-114dBm	-14db	10	140m
5	-79dBm	3db	12	131m
6	-85dBm	3db	12	422m
7	-111dBm	-13db	12	290m

Por sua vez, os testes de campo foram realizados também em dois dispositivos, contudo, a distância entre eles foi maior e variável se comparada com a distância dos testes de laboratório em que o dispositivo com função de *Common-device* foi alocado em diferentes distâncias, modificadas entre cada um dos testes. Na Tabela 2 estão as distâncias entre os *Common-devices* e a *Master-seed*. Ambos os testes foram realizados para *spreading factors* 10 e 12. Na Figura 31 é mostrada a localização de todos os sítios utilizados durante os testes de campo.



Figura 31 – Locações dos dispositivos para os testes em Fortaleza/CE. Fonte: Autor.

Na Tabela 1 pode-se observar que o tempo de transferência para SF 10 são menores que para SF 12, contudo observando a Tabela 2 pode-se ver que os dispositivos que estavam operando em SF 12 conseguem valores melhores de $RSSI$ e SNR , consequentemente operando em distâncias maiores para o dispositivo *Master-seed*.

Importante notar que observando as Figuras 32, 33 e 34, onde tem-se respectivamente os perfis de elevação para os sítios 4,6 e 7, pode-se ver que os sítios 6 e 7 apesar de possuírem perfis de elevação piores do que o sítio 4, conseguem operar de forma similar no caso do sítio 7 e melhor no caso do sítio 6.

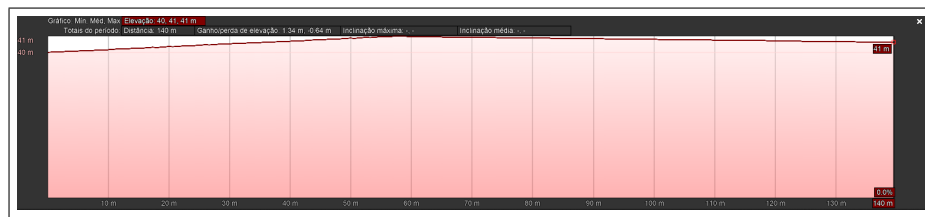


Figura 32 – Perfil de elevação entre Master-Seed e Common-device 4. Fonte: Autor.

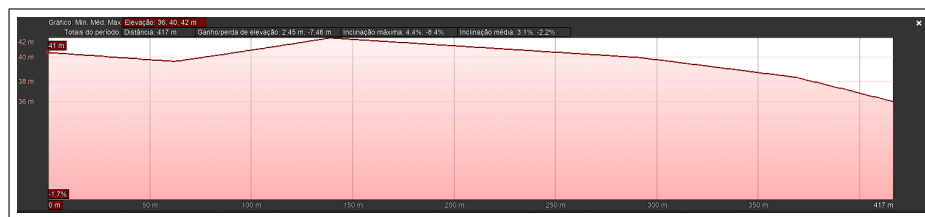


Figura 33 – Perfil de elevação entre Master-Seed e Common-device 6. Fonte: Autor.

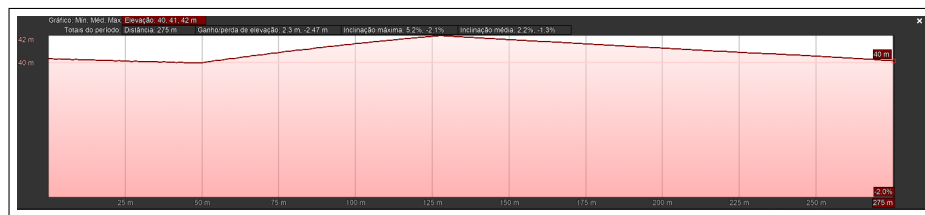


Figura 34 – Perfil de elevação entre Master-Seed e Common-device 7. Fonte: Autor.

Analisando o tempo de atualização de *firmware* apresentado na Tabela 1, observa-se que a velocidade do processo de FUOTA LFR é superior as velocidades alcançadas por Abdelfadeel e Anastasiou (Secção 2.5), pois para imagens de tamanho de 100 kB o tempo de atualização previsto por eles foi, respectivamente, 13,3 horas e 62 dias, em contrapartida o tempo obtido pelo FUOTA LFR foi de 41 minutos para DR0 (SF12) e 31 minutos para DR2 (SF10).

4 Resultados e Discussões

Para realizar a validação do modelo de FUOTA proposto em um ambiente real, foi o utilizado o parque de equipamentos da empresa Equatorial Energia no estado do Maranhão. Segundo dados do NEPEN, até a presente data de desenvolvimento desse trabalho, o parque de medidores inteligentes que possuem interface LoRaWAN no Maranhão é de 1000 unidades de medição individual e 550 unidades de medição coletiva. A Tabela 3 descreve os cenários de teste de campo que foram utilizados para validar o FUOTA LFR.

Tabela 3 – Cenários de validação do FUOTA LFR em Campo

ID Cenário	Quantidade de Unidades	Mês	Tamanho do <i>Firmware</i>
1	4 dispositivos	JUN/2022	260kB
2	5 dispositivos	JAN/2023	260kB

Em ambos os cenários a imagem transferida tinha o tamanho aproximado de 260kB, e todos os dispositivos tratavam-se de conjuntos reais de medidores inteligentes e NICs, como mostrado na Seção 3.1.2. No caso do primeiro cenário os dispositivos estavam instalados na CEASA de São Luís e realizavam a medição real de consumo de energia elétrica nos estabelecimentos. As Figuras 35 e 36 mostram a localização dos medidores. Nesse cenário foram utilizados 4 dispositivos, onde um deles foi definido como *Master-Seed* e os demais foram definidos como *Common-Device*.

No Cenário 1, o processo de validação do FUOTA LFR, ocorreu seguindo as seguintes etapas:

1. Instalação manual da versão de *firmware* **2.2.2-7** para que os dispositivos de campo suportassem o FUOTA LFR;
2. Envio dos comandos de agendamento do FUOTA LFR de forma manual por meio do LNS da Equatorial Energia;
3. Processo de FUOTA LFR ocorrendo conforme agendamento, versão de *firmware* alvo após o processo é **2.2.4-4**.
4. Verificação no sistema supervisor da Equatorial Energia que os dispositivos foram atualizados via FUOTA LFR.



Figura 35 – Locações dos dispositivos para os testes na CEASA em São Luís/MA. Fonte: Autor.



Figura 36 – Locações dos dispositivos para os testes na CEASA em São Luís/MA. Fonte: Autor.

O processo de instalação da versão de *firmware* com suporte ao FUOTA LFR foi realizada *in loco* com o apoio da equipe operacional da Equatorial, conforme pode ser visto na Figura 37 e no Apêndice C. Durante esse processo foi instalada a versão de testes

2.2.2-7, essa versão como mencionado anteriormente já dava suporte ao protocolo LFR e ao processo de FUOTA LFR.

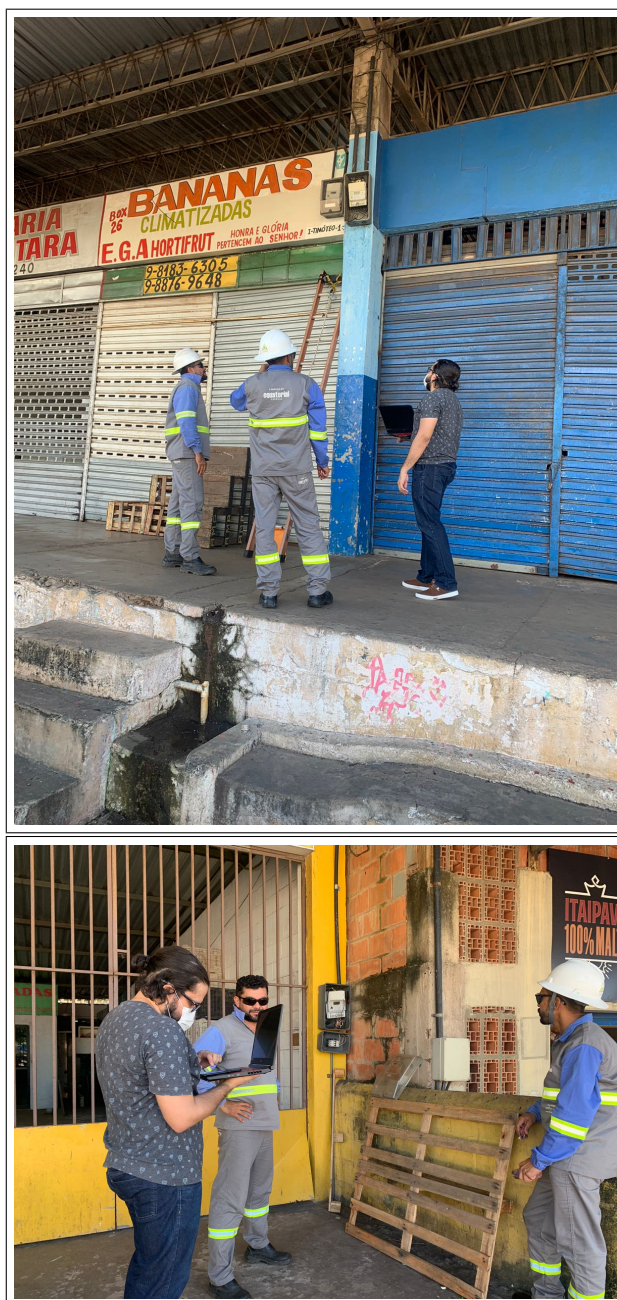


Figura 37 – Atualização do *firmware* de campo para suporte ao protocolo LFR. Fonte: Autor.

O envio dos comandos de configuração do agendamento do LFR, como descritos na Seção 3.2.2.2, foi realizado diretamente na interface do LNS da Equatorial Energia, pois até a data de realização do teste do Cenário 1 ainda não estava implementada a interface de suporte ao LFR, como será mostrado no ensaios do Cenário 2.

Uma vez configurado o agendamento, o processo de FUOTA LFR ocorreu no

horário definido e de forma automática e independente da rede LoRaWAN, vale ressaltar que durante o processo de FUOTA LFR, os demais dispositivos instalados na CEASA para medição de energia dos outros estabelecimentos continuaram o seu funcionamento normal na rede LoRaWAN, sem sofrer nenhuma interferência da rede LFR, a Tabela 4 mostra os parâmetros de modulação definidos para os testes de ambos os cenários.

Tabela 4 – Parâmetros de Modulação LoRa dos Cenários de validação

Parâmetro	Valor
<i>Spreading Factor</i>	12
<i>Data Rate</i>	0
<i>Code Rate</i>	4/7
Potência de Transmissão	27dBm
Preambulo LoRa	8 símbolos
<i>Bandwidth</i>	500kHz
Frequência de Transmissão	905MHz

Observa-se que a frequência de transmissão escolhida está contida dentro da faixa determinada para a frequência nominal dos Estados Unidos que é de 915MHz e compreende a faixa entre 902-928 MHz ([ALLIANCE, 2020](#)). A Figura 38 mostra a faixa de frequência que os equipamentos podem operar, essa faixa de frequência é a mesma utilizada no Brasil e por consequência nos equipamentos da Equatorial Energia.

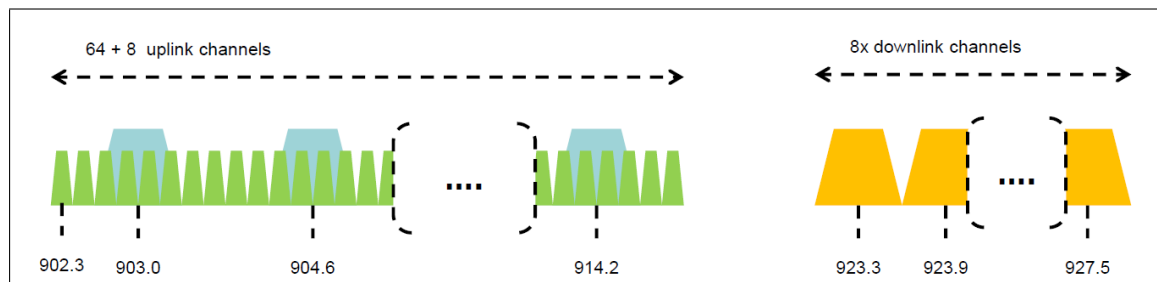


Figura 38 – Frequência dos Canais US902-928. Fonte: [Alliance \(2020\)](#).

A validação do processo de FUOTA LFR, foi constatada por meio da observação da versão de *firmware* descrita no sistema supervisor da Equatorial Energia, essa informação é enviada diretamente para o sistema supervisor pelo dispositivo de campo assim que o mesmo se junta a rede LoRaWAN da empresa. Na Figura 39, pode-se observar que no primeiro quadro a versão do dispositivo era **2.2.2-7** e após o o processo de FUOTA LFR a versão reportada foi **2.2.4-4**, dessa forma validando em uma rede LoRa real, o modelo de FUOTA proposto neste trabalho. O tempo de atualização do Cenário 1 foi de cerca de 73 minutos, superior ao obtido nos testes de bancada e campo em Fortaleza/CE

(Secção 3.2.3). Isso dá indícios que em ambiente real possivelmente foi necessário o reenvio de alguns blocos pendentes (Secção 3.2.2.3), dessa forma validando os mecanismos de integridade da imagem de *firmware* transmitida.

The figure displays two screenshots of the 'Sistema Supervisório da Equatorial' web interface. Both screenshots show the 'Informações do Medidor' (Meter Information) and 'Modem' sections for a specific meter (ID: 63000142).

Screenshot 1 (Top):

- Medidor Section:**
 - Serial: 63000142
 - Fase: Trifásico
 - Modelo: ZEUS 8023
 - Fabricante: ELETRA
 - Versão de Firmware: 23Z.A1.F0.BR02.V7011
 - Código de Instalação: 2000142760
 - Versão de Firmware: 0
 - Período Uplink de Leitura: 24H:0M
 - Estado de Rádio: **Ativado**
 - Estado de comunicação do medidor: **Online**
 - Tags: **Etapa 11 TESTE LFR**
 - CRC memória de massa: ZEUS 8023 Com Status do Registro e Energia Incremental
 - CRC Energia: Registro do Status Eletra
 - Última comunicação com MDC: 30/6/2022 15:08:32
 - Última comunicação: 30/6/2022 15:09:42
 - Último comando executado: system
 - Comando: **Atualizar Data e Hora**
- Modem Section:**
 - EUI: 0080e126004e93aa
 - Versão do firmware: 2.2.2-7
 - Qualidade do Sinal: **Excelente (5 Barras)**
 - SNR: -10
 - RSSI: -95
 - Última atualização: 30/6/2022 15:47:57
- Alarms Críticos:**
 - Descrição: LAST GASP
 - Horário de Leitura: 30/6/2022 15:07:14
 - Última Atualização: 30/6/2022 15:08:32

Screenshot 2 (Bottom):

- Medidor Section:** (Identical to Screenshot 1)
- Modem Section:**
 - EUI: 0080e126004e93aa
 - Versão do firmware: 2.2.4-4
 - Qualidade do Sinal: **Excelente (5 Barras)**
 - SNR: 4.19999980926514
 - RSSI: -95
 - Última atualização: 30/6/2022 17:01:34
- Alarms Críticos:**
 - Descrição: LAST GASP
 - Horário de Leitura: 30/6/2022 15:07:14
 - Última Atualização: 30/6/2022 15:08:32

Figura 39 – Sistema Supervisório da Equatorial. Fonte: Autor.

No Cenário 2, diferentemente do Cenário 1, em vez medir o consumo de estabelecimentos comerciais, os dispositivos mediram o consumo em residências da localidade de Areinha em São Luís/MA. Nesse cenário foram utilizados 5 dispositivos, onde um deles foi definido como *Master-Seed* e os demais foram definidos como *Common-Device*. No Cenário 2, o processo de validação do FUOTA LFR ocorreu seguindo as seguintes etapas:

1. Instalação manual da versão de *firmware* **2.2.8-5** para que os dispositivos de campo suportassem o FUOTA LFR;

2. Envio dos comandos de agendamento do FUOTA LFR de forma automatizada por meio do sistema de agendamento do LFR instalado nos servidores da Equatorial Energia;
3. Processo de FUOTA LFR ocorrendo conforme agendamento, versão de *firmware* alvo após o processo é **2.2.8-6**.
4. Verificação no sistema supervisor da Equatorial Energia que os dispositivos foram atualizados via FUOTA LFR.



Figura 40 – Locações dos dispositivos para os testes em Areinha São Luís/MA. Fonte: Autor.

O processo de instalação da versão de *firmware* que desse suporte ao FUOTA LFR foi realizada *in loco* pela equipe operacional da Equatorial. Durante esse processo foi instalada a versão de testes **2.2.8-5**, essa versão como mencionado anteriormente já dava suporte ao protocolo LFR e ao processo de FUOTA LFR. A Figura 40 mostra a localização dos dispositivos que participaram do processo de FUOTA LFR. A distância do dispositivo mais distante da *Master-Seed 2* era de cerca de 280m e observando na Figura 41 pode-se notar que não existe uma visada direta entre esses dispositivos.

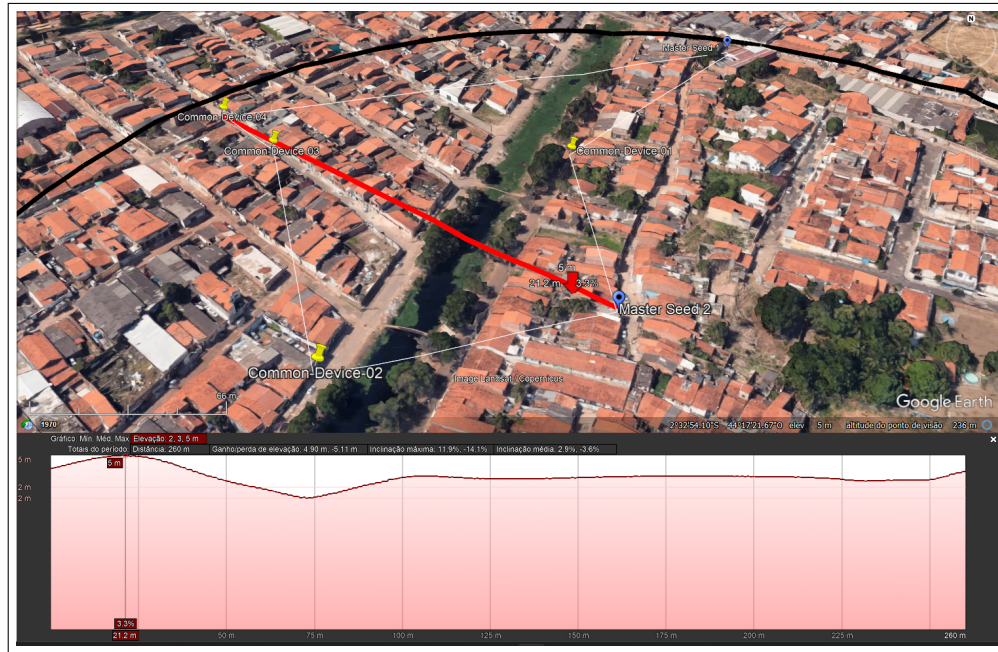


Figura 41 – Perfil de Elevação entre Master-Seed 2 e Common-Device 4. Fonte: Autor.

Os parâmetros de modulação LoRa definidos para esse cenário foram os mesmos descritos na Tabela 3. Na Figura 42 está uma imagem do sistema de agendamento do LFR, que foi desenvolvido em parceria com o NEPEN. O objetivo desse sistema é realizar a etapa de configuração ainda em rede LoRaWAN, descrita na Seção 3.2.2.2, em destaque dessa figura pode-se observar os dispositivos selecionados para o processo de validação do Cenário 2.

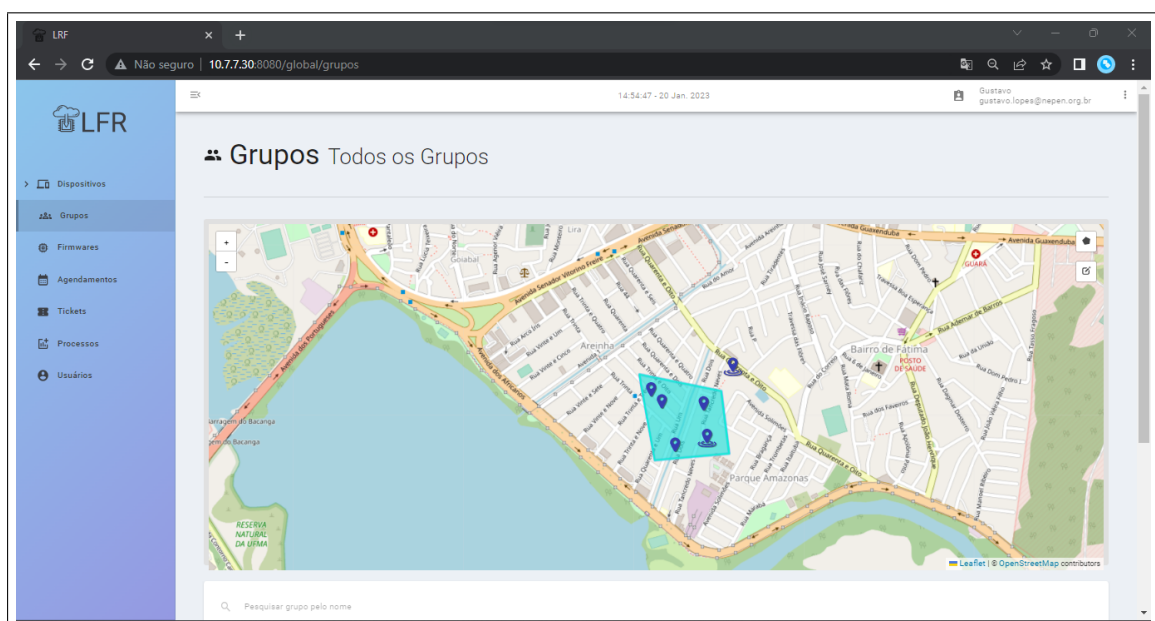


Figura 42 – Sistema de agendamento do LFR - versão beta. Fonte: Autor.

De forma similar ao Cenário 1, a validação do processo de FUOTA LFR no cenário 2 foi constatada por meio da observação da versão de *firmware* descrita no sistema supervisor da Equatorial Energia. Pode-se observar na Figura 43 que a versão de *firmware* reportada por um dos dispositivos foi **2.2.8-6** e o tempo de atualização médio do Cenário 2 foi de cerca de 70 minutos. Assim como o Cenário 1, o tempo de atualização obtido foi superior aos dos testes de Fortaleza/CE (3.2.3).

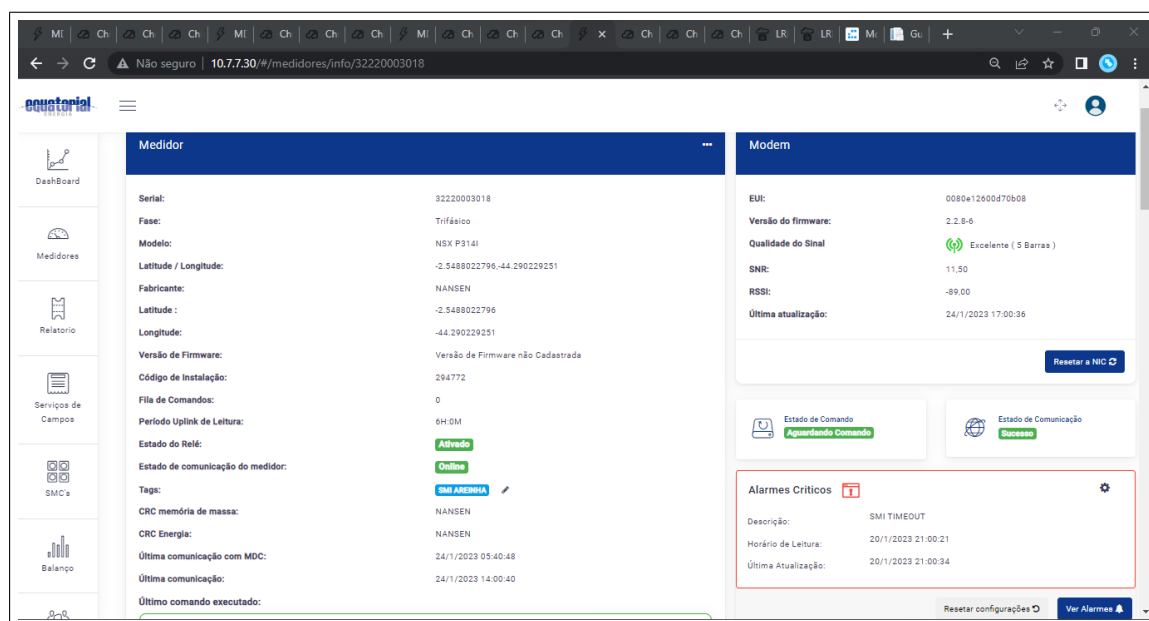


Figura 43 – Sistema Supervisor da Equatorial. Fonte: Autor.

Analisando os tempos de transferência alcançados pelo FUOTA LFR em ambos os testes, seja em laboratório ou em campo, pode-se observar que o modelo de FUOTA proposto neste trabalho foi mais rápido em transferir imagens do que os modelos propostos por Anastasiou et al. (2020) e Abdelfadeel et al. (2020).

Tabela 5 – Comparação entre tempos de atualização em diferentes processos/cenários de FUOTA

FUOTA	Spreading Factor	Tamanho da Imagem	Tempo de Atualização
Abdelfadeel	12	50kB	480 min
Abdelfadeel	10	50kB	120 min
Abdelfadeel	12	100kB	920 min
Abdelfadeel	10	100kB	250 min
Anastasiou	12	50kB	11,5 dias
Anastasiou	10	50kB	9,8 dias
LFR Bancada	12	260kB	54,5 min
LFR Bancada	10	260kB	41,2 min
LFR Campo	12	260kB	63,2 min
LFR Campo	10	260kB	47,5 min
LFR CEASA	12	260kB	73 min
LFR Areinhada	12	260kB	70 min

Na Tabela 5 estão descritos os tempos de atualização, tamanho da imagem do *firmware* transferida e SF para os modelos propostos por Anastasiou e Abdelfadeel, bem como diferentes cenários onde LFR foi aplicado desde os testes de bancada como os testes em campo nas cidades de Fortaleza e São Luís. A Figura 44 mostra graficamente uma comparação entre o modelo proposto por Abdelfadeel et al. (2020) e o LFR. Nesse gráfico não foram inseridos os resultados do modelo de Anastasiou, pois os mesmos possuem um tempo de atualização na ordem de dias e ficaria graficamente difícil de representá-los em uma escala adequada com o tempo de atualização dos outros processos que estão na ordem de dezenas e/ou centenas de minutos.

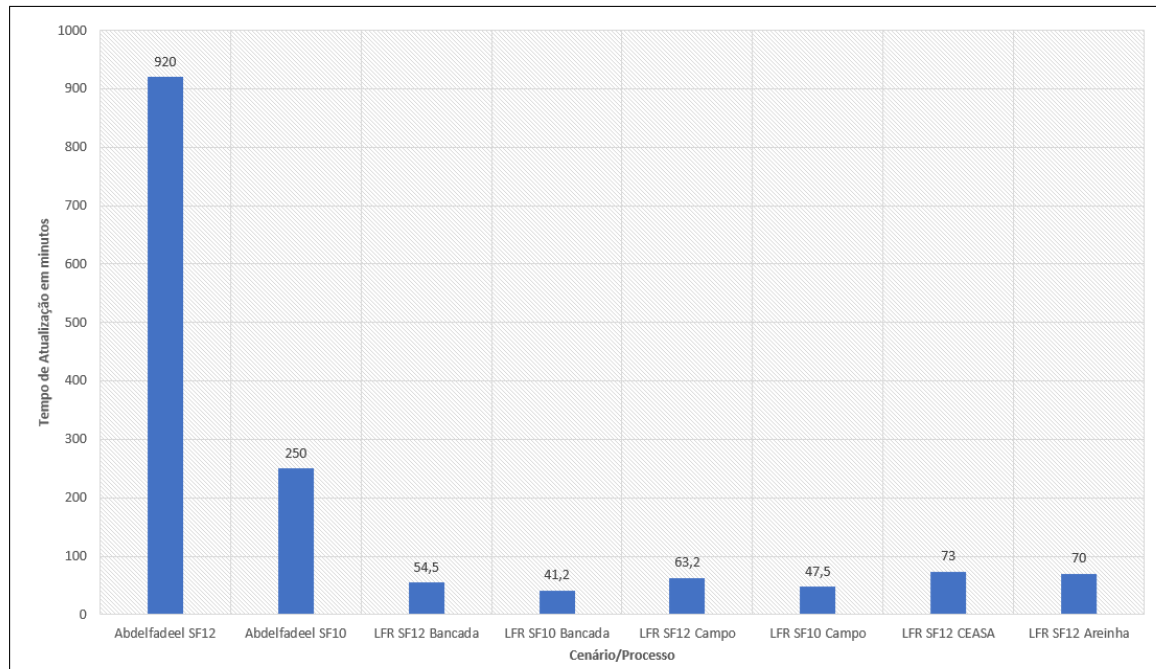


Figura 44 – Tempos de atualização (minutos) em diferentes cenários/processos FUOTA.
Fonte: Autor.

Analisando os resultados mostrados na Tabela 5, observa-se que o LFR foi mais rápido em transferir as imagens em todos os cenários nos quais ele foi utilizado. Em contrapartida os modelos de Abdelfadeel e Anastasiou levaram, respectivamente, várias horas e até dias para realizar o processo de atualização. Vale ressaltar que ambos os modelos utilizam uma rede LoRaWAN, logo possuem limitações por parte das regulações de cada região, como descrito por Alliance (2020), e além disso necessitam da ocupação do *gateway* LoRaWAN durante todo o processo de atualização, como descrito por Montagny (2021) e Sornin (2020). No caso do LFR essas limitações não existem, pois os dispositivos transferem as imagens de *firmware* entre si obedecendo as regras e topologia propostas por esse trabalho. Outra característica muito importante do LFR é a não necessidade de reenvio de toda a imagem em caso de blocos faltantes, pois somente os blocos pendentes serão reenviados. Além disso a ativação da imagem é feita por cada dispositivo final (*end-node device*), sem a necessidade que esse comando de ativação venha de um servidor centralizado como é o caso do modelo proposto por Anastasiou, dessa forma cada dispositivo se atualiza e se ativa, tão logo seja possível e em seguida já retorna para seu funcionamento normal na rede LoRaWAN.

5 Conclusões

A atualização de *firmware* é algo extremamente importante em dispositivos IoT, pois aumenta a vida útil da aplicação, garantindo a segurança, confiabilidade e otimização de processos e funções. Neste trabalho foi apresentado um modelo de protocolo de comunicação para ser utilizado em dispositivos LoRa, de forma a implementar uma camada de rede que possa vir a ser utilizada no lugar de LoRaWAN durante processo de atualização ou envio de imagens muito grandes, da ordem de *kilobytes*. As principais características do modelo proposto são:

- Operar de forma independente do protocolo LoRaWAN, não gerando nenhuma interferência em redes LoRaWAN próximas e permitindo comutação do dispositivo entre ambos os protocolos.
- Tempo de transferência de imagens de inferior aos modelos propostos para redes LoRaWAN.
- Implementação de mecanismos de segurança que garantem o reenvio de fragmentos faltantes da imagem do *firmware* para dispositivos solicitantes, sem interferir no processo de atualização dos demais dispositivos da rede.
- Verificação de integridade e ativação da imagem de *firmware* sem a necessidade de intervenção de um servidor externo, assim garantindo o retorno do dispositivo ao seu funcionamento normal em LoRaWAN, tão logo seja atualizado.

Para realizar uma prova de conceito, o modelo foi implementado em linguagem C e portado para uma PCI contendo um microcontrolador Arm Cortex M4 e um rádio da SEMTECH SX126x e foram definidos 2 cenários iniciais de validação em Fortaleza, um em bancada e outro em campo em uma rede hipotética. Além desses cenários de Fortaleza, o modelo foi integrado a dispositivos reais de medição de energia instalados na cidade de São Luís e pertencentes a empresa Equatorial Energia. Em São Luís foram definidos dois cenários para validação do modelo.

Com base nos resultados experimentais obtidos em Fortaleza e nos resultados dos testes de campo obtidos em São Luís, conclui-se que o modelo de FUOTA proposto por este trabalho permite uma atualização de *firmware* ou transferência de dados com uma velocidade superior aos modelos propostos pela literatura, garantindo confiabilidade no envio dos dados e garantindo que a rede LoRaWAN nativa não seja afetada pelo FUOTA, seja por ruídos ou inoperância do gateway devido ao envio de downlinks.

5.1 Trabalhos futuros

A seguir estão algumas sugestões de atividades para a continuidade da pesquisa e o aprimoramento do protocolo proposto

- Aprimorar o conceito de sub-redes já implementado no protocolo por meio do cabeçalho FHDR, dessa forma explorando um conceito de auto-roteamento da rede e escolha automática de *master-seeds* e *future-seeds*.
- Implementar um auto-teste para que os dispositivos da rede possam selecionar de forma automática os melhores parâmetros de modulação, tais como:
 - Seleção do melhor canal e largura de banda para cada região.
 - Seleção automática de SF e CR para cada sub-rede de forma a minimizar efeitos de colisão de pacotes ou interferência com outras sub-redes LFR.

Referências

- ABDELFADEEL, K. **Fuotasim simulator**. 2020. Disponível em: <<https://github.com/kqorany/FUOTASim>>.
- ABDELFADEEL, K. et al. How to Make Firmware Updates over LoRaWAN Possible. **Proceedings - 21st IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM 2020**, p. 16–25, 2020.
- ALLIANCE, L. **LoRaWAN™ 1.1 Specification**. 2017.
- ALLIANCE, L. **RP002-1.0.2 LoRaWAN® Regional Parameters**. 2020.
- ANASTASIOU, A. et al. IoT Device Firmware Update over LoRa: The Blockchain Solution. **Proceedings - 16th Annual International Conference on Distributed Computing in Sensor Systems, DCOSS 2020**, n. October, p. 404–411, 2020.
- BEDI, G. et al. Review of internet of things (iot) in electric power and energy systems. **IEEE Internet of Things Journal**, v. 5, p. 847–870, 2018. ISSN 23274662.
- FABER, M. J. et al. A theoretical and experimental evaluation on the performance of lora technology. **IEEE Sensors Journal**, Institute of Electrical and Electronics Engineers Inc., v. 20, p. 9480–9489, 8 2020. ISSN 15581748.
- FERREIRA, A. E. et al. A study of the LoRa signal propagation in forest, urban, and suburban environments. **Annales des Telecommunications/Annals of Telecommunications**, v. 75, n. 7-8, p. 333–351, 2020. ISSN 19589395.
- MARAIS, J. M.; MALEKIAN, R.; ABU-MAHFOUZ, A. M. LoRa and LoRaWAN testbeds: A review. **2017 IEEE AFRICON: Science, Technology and Innovation for Africa, AFRICON 2017**, p. 1496–1501, 2017.
- MONTAGNY, S. **LoRa-LoRaWAN and Internet of Things for beginners**. 2021. Disponível em: <www.univ-smb.fr/lorawan>.
- MONTAGNY, S.; AUGAGNEUR, A. **LoRaWAN for advanced users**. 2022. Disponível em: <www.univ-smb.fr/lorawan>.
- MTETWA, N. S. et al. Ota firmware updates for lorawan using blockchain. **2020 2nd International Multidisciplinary Information Technology and Engineering Conference, IMITEC 2020**, 2020.
- NTSEANE, L.; ISONG, B. Analysis of lora/lorawan challenges: Review. **Proceedings - 2019 International Multidisciplinary Information Technology and Engineering Conference, IMITEC 2019**, IEEE, 2019.
- POURSAFAR, N.; ALAHI, M. E. E.; MUKHOPADHYAY, S. Long-range wireless technologies for IoT applications: A review. **Proceedings of the International Conference on Sensing Technology, ICST**, v. 2017-Decem, p. 1–6, 2018. ISSN 21568073.

- QUERALTA, J. P. et al. Comparative study of LPWAN technologies on unlicensed bands for M2M communication in the IoT: Beyond Lora and Lorawan. **Procedia Computer Science**, Elsevier B.V., v. 155, n. 2018, p. 343–350, 2019. ISSN 18770509. Disponível em: <<https://doi.org/10.1016/j.procs.2019.08.049>>.
- SAĞIR, S. et al. Evaluation of low-power long distance radio communication in urban areas: Lora and impact of spreading factor. In: **2019 Seventh International Conference on Digital Information Processing and Communications (ICDIPC)**. [S.l.: s.n.], 2019. p. 68–71.
- SCHILDT, H. C, **completo e total**. Makron / McGrawil, 1990. Disponível em: <<https://books.google.com.br/books?id=AAD4pwAACAAJ>>.
- SEGGER. **J-Link**. 2023. Disponível em: <<https://www.segger.com/products/debug-probes/j-link/>>.
- SEMTECH. **an1200.22 LoRa Modulation Basics**. 2015.
- SEMTECH. **SX1261/2 Long Range, Low Power, sub-GHz RF Transceiver**. 2021.
- SORNIN, N. **LoRaWAN® Firmware Updates Over-the-Air**. 2020.
- SPRINGER, A. et al. Spread spectrum communications using chirp signals. **IEEE/AFCEA - EUROCOMM 2000: Information Systems for Enhanced Public Safety and Security**, v. 00, p. 166–170, 2000.
- STAIKOPOULOS, A.; KANAKARIS, V.; PAPAKOSTAS, G. A. Image Transmission via LoRa Networks - A Survey. **2020 IEEE 5th International Conference on Image, Vision and Computing, ICIVC 2020**, p. 150–154, 2020.
- STMICROELECTRONICS. **Multiprotocol wireless 32-bit MCU Arm® -based Cortex®-M4 with FPU, Bluetooth® 5.2 and 802.15.4 radio solution**. 2022.
- STMICROELECTRONICS. **Atollic TrueSTUDIO**. 2023. Disponível em: <<https://www.st.com/en/development-tools/truestudio.html>>.
- TANENBAUM, A. S. **Redes de Computadores**. 2002.
- TURCINOVIC, F. et al. Analysis of LoRa Parameters in Real-World Communication. **Proceedings Elmar - International Symposium Electronics in Marine**, v. 2020-Septe, n. September, p. 87–90, 2020. ISSN 13342630.
- ZOURMAND, A. et al. Internet of Things (IoT) using LoRa technology. **2019 IEEE International Conference on Automatic Control and Intelligent Systems, I2CACIS 2019 - Proceedings**, IEEE, n. June, p. 324–330, 2019.

Apêndices

APÊNDICE A – Diagrama de Sequências

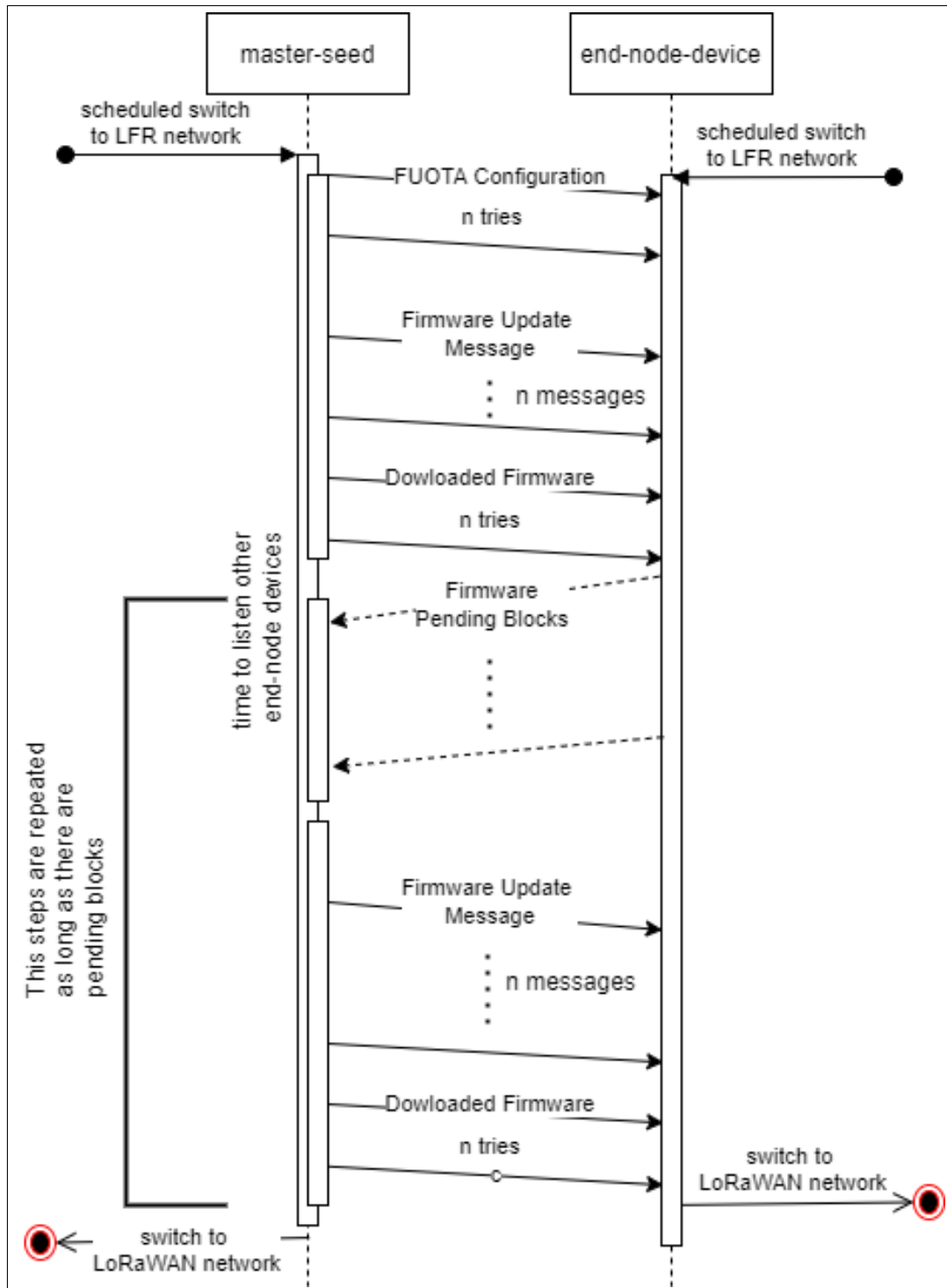
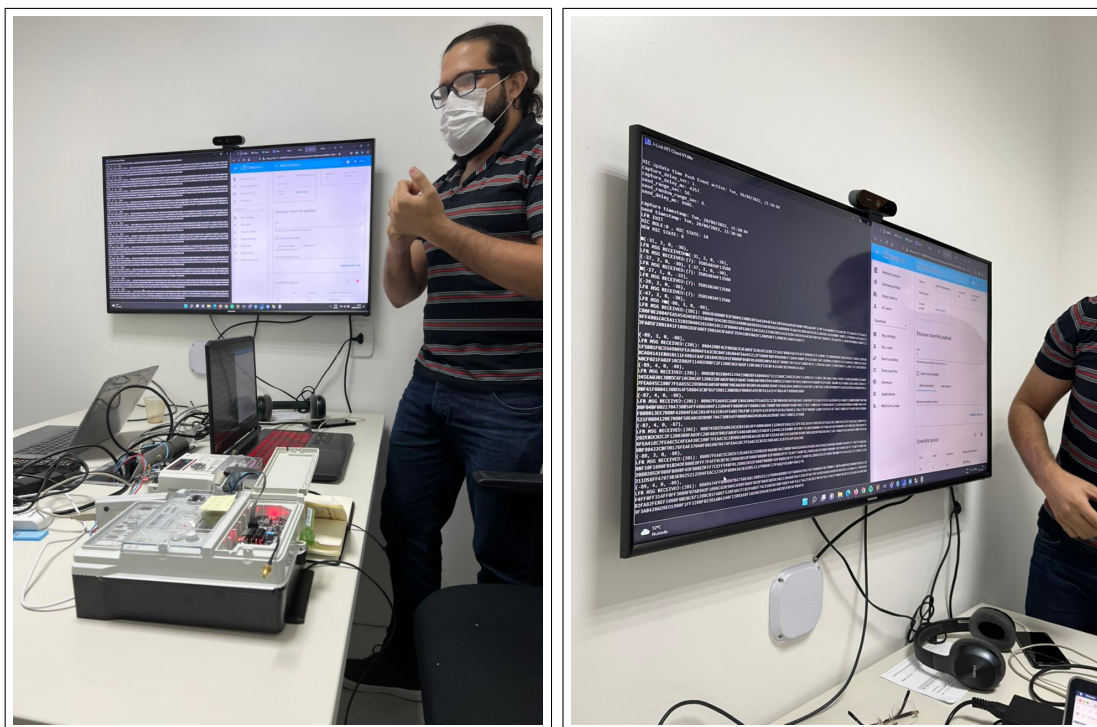


Diagrama de Sequências de uma rede simples do LFR contendo Master-Seeds, imagem em alta definição.

APÊNDICE C – Testes de Campo CEASA São Luís/MA





Imagens das instalações e treinamento realizado com a equipe da Equatorial Energia em junho de 2022

APÊNDICE D – Trecho do Código LFR

```

136@ /**
137  * @typedef lfr_lora_parameters_t
138  * @brief Typedef to represent the LFR parameters struct that can configured in the radio LoRa
139  */
140@ typedef struct{
141     RadioModems_t radio_modem;
142     lfr_bandwidth_t bandwidth;
143     lfr_datarate_t dr;
144     lfr_coderate_t cr;
145     uint8_t preamble;
146     lfr_crc_t crc_status;
147     lfr_payload_len_t payload_len_conf;
148     int8_t power_tx;           /**< In dBm */
149     uint32_t freq_channel_lora; /**< In Hertz */
150     uint8_t channel_selected;
151 }lfr_lora_parameters_t;
152
153@ /**
154  * @typedef lfr_status_t
155  * @brief Possible status relative to LFR protocol communication
156  */
157@ typedef enum
158 {
159     LFR_SUCCESS                = 0,
160     LFR_INVALID_PARAMETER     = 1,
161     LFR_ERROR                  = 2,
162     LFR_NOT_READY              = 3,
163     LFR_CRC_FAILURE            = 4,
164     LFR_COMMAND_NOT_IMPLEMENTED = 5,
165     LFR_BB_OVERFLOW            = 6,
166     LFR_INVALID_RESPONSE_LEN   = 7,
167     LFR_TIMEOUT                = 8,
168     LFR_PENDING_BLOCKS         = 9
169 }lfr_status_t;
170
171@ /**
172  * @typedef lfr_mhdr_types_t
173  * @brief MAC header of an LFR message
174  */
175@ typedef enum
176 {
177     RANGE_SELFTEST_DONE        = 0x20, /**< Self test finished */
178     FUOTA_CONFIGURATION        = 0x40, /**< FUOTA initial setup message */
179     NIC_FIRMWARE_UPDATE_MESSAGE = 0x50, /**< NIC type firmware update block */
180     SMARTMETER_FIRMWARE_UPDATE_MESSAGE = 0x60, /**< Smart meter type firmware update block */
181     DOWNLOADED_FIRMWARE        = 0x70, /**< Finished sending the blocks containing the firmware image */
182     FIRMWARE_PENDING_BLOCKS     = 0x80, /**< Message containing all pending blocks for a FUOTA process */
183 }lfr_mhdr_types_t;

```

```

233- /**
234-  * @struct st_ui_t
235-  * @brief Contain the unique_id buffer
236-  */
237- typedef struct{
238-     uint8_t lfr_deveui_source[LFR_DEV_EUI_SIZE];    /**< LFR DevEui Source*/
239-     uint8_t lfr_deveui_dest[LFR_DEV_EUI_SIZE];      /**< LFR DevEui Destiny*/
240- }lfr_fhdr_t;
241-
242- /**
243-  * @struct lfr_payload_t
244-  * @brief Struct containing the Payload of a LFR Message
245-  */
246- typedef struct{
247-     uint16_t lfr_block_id;                          /**< LFR Block ID */
248-     uint8_t lfr_payload_size;                       /**< LFR Payload Size */
249-     uint8_t lfr_payload[201];//[MAX_LFR_PAYLOAD_DATA_SIZE]; /**< LFR Payload Data */
250-     byte_buffer_t lfr_payload_bb;                   /**< LFR ByteBuffer */
251- }lfr_payload_t;
252-
253- typedef struct{
254-     uint16_t lfr_block_id;                          /**< LFR Block ID */
255-     uint8_t lfr_payload_size;                       /**< LFR Payload Size */
256-     uint8_t lfr_payload[MAX_LFR_PAYLOAD_DATA_SIZE]; /**< LFR Payload Data */
257-     byte_buffer_t lfr_payload_bb;                   /**< LFR ByteBuffer */
258- }lfr_cmd_payload_t;
259-
260- /**
261-  * @struct lfr_message_t
262-  * @brief Struct containing the LFR Message <MHDR>|<FHDR>|<FPORT>|<LFR_PAYLOAD>|<MIC>
263-  */
264- typedef struct{
265-     lfr_mhdr_types_t lfr_mhdr; /**< MAC Header from LFR Message */
266-     lfr_fhdr_t lfr_fhdr;      /**< Frame Header from LFR Message */
267-     uint8_t lfr_fport;        /**< LFR Frame Port */
268-     lfr_payload_t lfr_payload; /**< Payload from LFR Message */
269-     mic_type_t lfr_MIC;       /**< LFR Message Integrity Code */
270- }lfr_message_t;
271-
272-
273- typedef struct{
274-     lfr_mhdr_types_t lfr_mhdr; /**< MAC Header from LFR Message */
275-     lfr_fhdr_t lfr_fhdr;      /**< Frame Header from LFR Message */
276-     uint8_t lfr_fport;        /**< LFR Frame Port */
277-     fuota_configuration_t fuota_conf; /**< LFR Fuota Conf parameters */
278-     lfr_cmd_payload_t lfr_datablock_2_Send; /**< LFR struct containing the firmware image data block to send */
279-     mic_type_t lfr_MIC;       /**< LFR Message Integrity Code */
280- }lfr_cmd_t;
281-
282- typedef enum{
283-     FUOTA_CONF_SIZE = 7, /**< Fuota Conf Size equal 2B(Block Total Number) + 4B(Freq Operation) + 1B(Image Type) */
284- }size_cmds_t;

```

```

185 /**
186  * @union mic_type_t
187  * @brief Union containing the data from Message Integrity Code
188  */
189 typedef union{
190     struct{
191         uint8_t first_byte;    /**< First byte from MIC */
192         uint8_t second_byte;   /**< Second byte from MIC */
193         uint8_t third_byte;    /**< Third byte from MIC */
194         uint8_t fourth_byte;   /**< Fourth byte from MIC */
195     };
196     uint32_t MIC;              /**< Complete data with all bytes from MIC */
197 }mic_type_t;
198
199 /**
200  * @struct st_ui_t
201  * @brief Contain the unique_id buffer
202  */
203 typedef struct{
204     uint8_t lfr_deveui_source[LFR_DEV_EUI_SIZE];    /**< LFR DevEui Source*/
205     uint8_t lfr_deveui_dest[LFR_DEV_EUI_SIZE];      /**< LFR DevEui Destiny*/
206 }lfr_fhdr_t;
207
208 /**
209  * @struct lfr_payload_t
210  * @brief Struct containing the Payload of a LFR Message
211  */
212 typedef struct{
213     uint16_t lfr_block_id;    /**< LFR Block ID */
214     uint8_t lfr_payload_size; /**< LFR Payload Size */
215     uint8_t lfr_payload[201]; /**< LFR Payload Data */
216     byte_buffer_t lfr_payload_bb; /**< LFR ByteBuffer */
217 }lfr_payload_t;
218
219 typedef struct{
220     uint16_t lfr_block_id;    /**< LFR Block ID */
221     uint8_t lfr_payload_size; /**< LFR Payload Size */
222     uint8_t lfr_payload[MAX_LFR_PAYLOAD_DATA_SIZE]; /**< LFR Payload Data */
223     byte_buffer_t lfr_payload_bb; /**< LFR ByteBuffer */
224 }lfr_cmd_payload_t;
225
226 /**
227  * @struct lfr_message_t
228  * @brief Struct containing the LFR Message <MHDR>|<FHDR>|<FPORT>|<LFR_PAYLOAD>|<MIC>
229  */
230 typedef struct{
231     lfr_mhdr_types_t lfr_mhdr; /**< MAC Header from LFR Message */
232     lfr_fhdr_t lfr_fhdr; /**< Frame Header from LFR Message */
233     uint8_t lfr_fport; /**< LFR Frame Port */
234     lfr_payload_t lfr_payload; /**< Payload from LFR Message */
235     mic_type_t lfr_MIC; /**< LFR Message Integrity Code */
236 }lfr_message_t;

```

```

30 /**
31  * @typedef lfr_sm_t
32  * @brief States of LFR Device
33  */
34 typedef enum
35 {
36     LFR_MASTER_SEED_READY           = 0,
37     LFR_MASTER_SEED_SENDER         = 1,
38     LFR_MASTER_SEED_LISTENER       = 2,
39     LFR_FUTURE_SEED_READY          = 3,
40     LFR_FUTURE_SEED_SENDER         = 4,
41     LFR_FUTURE_SEED_LISTENER       = 5,
42     LFR_COMMON_DEVICE_LISTENER     = 6,
43     LFR_COMMON_DEVICE_SENDER       = 7,
44     LFR_FUOTA_COMPLETED             = 8,
45     LFR_MASTER_SEED_SENDER_DONE    = 9,
46     LFR_CHECKING_HASH              = 10,
47     LFR_COMMON_DEVICE_WAITING_TO_SENDER = 11,
48     LORAWAN_DEVICE                 = 12,
49 }lfr_sm_t;
50
51 typedef enum
52 {
53     COMMON_DEVICE      = 0,
54     FUTURE_SEED        = 1,
55     MASTER_SEED        = 2,
56     MAX_SIZE_LFR_ROLE
57 }lfr_role_t;
58

```

```

292 /**
293  * PUBLIC PROTOTYPES
294  * *****
295  */
296 /**
297  * @brief Function that send a message by LFR network protocol.
298  * @param[in] p_message Pointer to a structure of type lfr_message_t, which contains the data to the assemble LFR packet.
299  * @param[in] p_parameters Pointer to a structure of type lfr_lora_parameters_t, which contains the radio parameters to LFR transmission.
300  * @return lfr_status_t
301  */
302 extern lfr_status_t lfr_send(lfr_cmd_t *p_cmd_message, lfr_lora_parameters_t *p_parameters);
303
304 /**
305  * @brief Function that send a message by LFR network protocol.
306  * @param[in] p_parameters Pointer to a structure of type lfr_lora_parameters_t, which contains the radio parameters to LFR reception.
307  * @param[in] lfr_packet_size T000 lfr_packet_size
308  * @return lfr_status_t
309  */
310 extern lfr_status_t lfr_rx_config(lfr_lora_parameters_t *p_parameters, uint8_t lfr_packet_size);
311
312 /**
313  * @brief Function that parses the LFR message.
314  * @param[in] p_bb_message Pointer to a bytearray, which contains the payload LoRa Message.
315  * @return lfr_status_t
316  */
317 extern lfr_status_t lfr_parser_message(byte_buffer_t *p_bb_message);
318
319 /**
320  * @brief Function that verify firmware image transferred and select the next step in LFR state machine.
321  * @param[in] void
322  * @return lfr_status_t
323  */
324 extern lfr_status_t lfr_checking_hash_firmware(void);
325
326 /**
327  * @brief Function that set first block pending flag.
328  * @param[in] void
329  * @return void
330  */
331 extern void set_first_block_pending(void);
332

```

Trechos de código do LFR, com ênfase nas estruturas que servem de base para a criação do protocolo, nos estados da máquina-de-estado LFR e na assinatura das funções principais conforme descrito na secção 3.2.2.

