



INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

Área Departamental de Engenharia Eletrónica e Telecomunicações e de Computadores



Estudo do Desempenho Rádio LoRaWAN em Cenários de Mobilidade e Trânsito de Dispositivos

ANDRÉ ESTEVES SANTOS ALVES

(Licenciado em Engenharia Informática, Redes e Telecomunicações)

Dissertação de natureza científica para obtenção do grau de Mestre em Engenharia Eletrónica e Telecomunicações

Orientadores: Professor Doutor Luís Carlos Gonçalves
Professor Mestre Luís Miguel Pires

Júri:

Presidente: Professor Doutor Rui António Policarpo Duarte

Vogais: Professor Doutor Vítor Fialho

Professor Doutor Luís Carlos Barruncho dos Santos Gonçalves

Novembro de 2023

Agradecimentos

Com a realização deste trabalho concluo mais uma etapa na minha vida académica. Como tal, gostava de agradecer a todos os que me acompanharam e fizeram parte deste percurso e que de alguma forma contribuíram para a realização deste trabalho.

Gostaria de agradecer a minha família e amigos pelo apoio prestado, e ao meu colega Edgar Inácio por me ter acompanhado por todo o progresso académico.

Um agradecimento especial ao Professor Luís Miguel Pires e ao Professor Luís Carlos Gonçalves pelo apoio e dedicação fornecidos no decorrer desta tese. E um agradecimento a os restantes docentes com que me cruzei e tive o orgulho de chamar Sr. Engenheiro para tirar duvidas.

Por fim gostaria de agradecer ao ISEL por me acolher e dar um espaço onde pude desenvolver as minhas capacidades como pessoa, estudante e futuro engenheiro.

Resumo

A implementação de dispositivos IoT em massa, para as suas várias aplicações, causa os problemas da utilização de um meio partilhado. Nomeadamente a colisão de tramas, que é agravada pelo facto de todas as estações escutarem as transmissões, o que faz com que o aumento da carga de tráfego devido à mobilidade de dispositivos seja uma preocupação na implementação destas redes.

A probabilidade de colisões de tramas também é afetada por características de tráfego, particularmente na frequência com que transmite, o tamanho da transmissão e da carga útil. Não obstante, o fator mobilidade dos dispositivos Long Range (LoRa) introduz um nível adicional de complexidade, influenciando o desempenho das redes de forma muito relevante contribuindo para cenários interferentes e níveis de colisão que importam considerar e avaliar.

Pretende-se, assim, com esta tese investigar o desempenho desta rede em cenários de mobilidade e trânsito de dispositivos. Através do desenvolvimento e implementação de modelos de mobilidade e trânsito de dispositivos finais (nós), adaptativos às condições do canal, procura-se avaliar o desempenho global em cenários de mobilidade através de simulação (simulador LoRaSim), considerando como técnicas de acesso aleatório Carrier Sense Multiple Access (CSMA) Collision Avoidance and Detection (CAD) e CSMA Collision-Avoidance (CA).

Palavras-Chave

LoRa, LoRaWAN, WAN, LPWAN, Modelos de Mobilidade, CSMA

Abstract

The massive implementation of IoT devices and its several applications surfaces the problem of shared mediums. Namely frame collisions which are aggravated by the fact that every station is listening to transmissions which makes node mobility a concern on the implementation of these types of networks.

Frame collision probability is also affected by traffic characteristics, particularly the period between transmissions, the size of a transmission and the payload. The mobility of LoRa (Long Range) devices adds a degree of complexity, influencing the performance of a network and contributing to interference scenarios and high collision probabilities that are crucial to consider and study.

The goal of this thesis is to investigate the performance of LoRa networks with node mobility scenarios through the development and implementation of mobility models for the wireless nodes, which can adapt to channel conditions. Such Mobility scenarios will also be studied with the use of a python simulation environment named LoRaSim, in which several medium access control techniques like CSMA/CA and CSMA/CAD are implemented.

Palavras-Chave

LoRa, LoRaWAN, WAN, LPWAN, Mobility Models, CSMA

Índice

Agradecimentos	ii
Resumo	iii
Abstract	iv
Índice	v
Índice de Figuras	viii
Índice de Tabelas	ix
Lista de Acrónimos	x
1. Introdução	1
1.1. Enquadramento	1
1.2. Motivação e Objetivos	1
1.3. Estrutura da Dissertação	2
1.4. Publicações Realizadas	3
2. Rede LoRa	4
2.1. Redes IoT	4
2.2. LPWAN	7
2.3. LoRa	10
2.3.1. Spread Spectrum Communications	12
2.4. LoRaWAN	20
3. Técnicas de Acesso Aleatório	24
3.1. ALOHA	24
3.2. CSMA/CD	29
3.3. CSMA/CA	30
3.4. Novas Abordagens	33
4. Ambiente LoRaSim	37
4.1. Ambiente de Simulação	37
4.2. Detalhes do LoRaSim	39
4.3. Tratamento de Colisões no LoRaSim	42
4.4. Acesso Aleatório	44
4.4.1. Implementação no LoRaSim	46
5. Modelos de Mobilidade	50
5.1. Estado da Arte	50
5.2. Metodologias Analíticas	53
5.2.1. Random Walk	54
5.2.2. Gauss-Markov	56
5.3. Datasets	57

5.3.1. Integração com o LoRaSim	59
6. Modelo Desenvolvido	62
6.1. Implementação de Modelos	62
6.2. Parâmetros Utilizados	65
7. Resultados	68
8. Conclusões	76
9. Bibliografia	79
10. Anexos	85

Índice de Figuras

<i>Figura 2. 1 Tecnologias Long Range, retirado de [20]</i>	8
<i>Figura 2. 2 - Arquitetura LoRa, retirado de [30]</i>	11
<i>Figura 2. 3 - Processo de Modulação/Spreading, retirado de [1]</i>	13
<i>Figura 2. 5 Representação no espectro de um up-chirp e down-chirp, retirado de [32]</i>	14
<i>Figura 2. 4 - Arvore de códigos, adaptado de [31]</i>	14
<i>Figura 2. 6 - Símbolo LoRa, retirado de [32]</i>	15
<i>Figura 2. 7 - Largura de banda LoRa, retirado de [32]</i>	16
<i>Figura 2. 8 - Fator de Espalhamento LoRa, 7 a 12, retirado de [33]</i>	16
<i>Figura 2. 9 - Camadas da tecnologia LoRaWAN, adaptado de [37]</i>	20
<i>Figura 2. 10 - Diferentes tipos de Classes, retirado de [37]</i>	21
<i>Figura 2. 11 - Segurança em LoRaWAN, retirado de [37]</i>	23
<i>Figura 3. 1 Esquema do Protocolo ALOHA puro, retirado de [5]</i>	25
<i>Figura 3. 2 Tramas em ALOHA puro, adaptado de [30]</i>	25
<i>Figura 3. 3 - Desempenho do ALOHA puro</i>	26
<i>Figura 3. 4 - Slotted ALOHA</i>	27
<i>Figura 3. 5 - Slotted ALOHA vs ALOHA puro</i>	28
<i>Figura 3. 6 - Esquema do Protocolo CSMA/CD, adaptado de [40]</i>	29
<i>Figura 3. 7 - Esquema do Protocolo CSMA/CA, adaptado de [30]</i>	31
<i>Figura 3. 8 - IEEE 802.11 DCF CSMA/CA, adaptado de [40]</i>	32
<i>Figura 3. 9 - Funcionamento do standard IEEE 802.15.4, adaptado de [41]</i>	33
<i>Figura 3. 10 - Conceito WiseMAC, adaptado de [9]</i>	35
<i>Figura 4. 1 - Simulador LoRaSim</i>	39
<i>Figura 4. 2 - Distribuição de nós e Gateways</i>	46
<i>Figura 4. 3 - Distribuição Nós e 4 Gateways</i>	47
<i>Figura 4. 4 - Colisão com Tx: 2dBm e 1 GW</i>	47
<i>Figura 4. 5 - Consumo de energia por protocolo com Tx: 2 dBm</i>	48
<i>Figura 4. 6 - Probabilidade de Colisão com SF12</i>	48
<i>Figura 4. 7 - Consumo energético do protocolo ALOHA</i>	49
<i>Figura 5. 1 - Modelo Random Walk</i>	55
<i>Figura 5. 2 - Modelo Random Walk Bidirecional</i>	55
<i>Figura 5. 3 - Padrão de movimento de um nó utilizando o modelo de mobilidade Gauss-Markov</i>	57
<i>Figura 5. 4 - Área geográfica de um Dataset de Alces do Canada</i>	58
<i>Figura 5. 5 - Dados representados em python</i>	59
<i>Figura 5. 6 - Aglomeração de dados do dataset numa area</i>	61
<i>Figura 6. 1 - Distribuição Random Walk</i>	63
<i>Figura 6. 2 - Distribuição Gauss-Markov</i>	64

Índice de Tabelas

<i>Tabela 1 - Tecnologias LPWAN.....</i>	<i>6</i>
<i>Tabela 2 - Comparação entre tecnologias LPWAN, adaptada de [26].....</i>	<i>9</i>
<i>Tabela 3 - Detalhes da rede LoRa, adaptado de [30].....</i>	<i>10</i>
<i>Tabela 4 - Ritmo binário de uma transmissão LoRa, valores retirados de [34].</i>	<i>18</i>
<i>Tabela 5 - Tempo de transmissão de pacotes LoRa, valores retirados de [35].</i>	<i>18</i>
<i>Tabela 6 - Sensibilidade de recessão em dBm para diferentes BW e SF, adaptado de [11]</i>	<i>41</i>
<i>Tabela 7 - Parâmetros de Simulação.....</i>	<i>46</i>
<i>Tabela 8 - Parâmetros de uma transmissão LoRa.....</i>	<i>66</i>
<i>Tabela 9 - Parâmetros de uma transmissão com SF12</i>	<i>66</i>
<i>Tabela 10 - Parâmetros de Transmissão, alteração Tx para SF7.....</i>	<i>66</i>
<i>Tabela 11 - Parâmetros de Transmissão, alteração Tx para SF12</i>	<i>66</i>

Lista de Acrónimos

3GPP - 3rd Generation Partnership Project

5G – 5th Generation

ABP - Activation By Personalisation

ADR – Adaptive Data Rate

AES - Advanced Encryption Standard

AppSKey - Application Session Key

CA - Collision-Avoidance

CAD - Collision Avoidance and Detection

CD - Collision-Detection

CDMA - Code-Division Multiple Access

CF - Carrying Frequency

CR - Code Rate

CSMA - Carrier Sense Multiple Access

CSS - Chirp-Spread Spectrum

CTS - Clear To Send

DBPSK - Differential Binary Phase Shift keying

DSSS - Direct Sequence Spread Spectrum

FEC - Forward Error Correction

GSM - Global System for Mobile

IMS - Industrial, Scientific and Medical

IoT - Internet Of Things

IP - Internet Protocol

LoRa - Long Range

LoRaWAN –Long Range Wide Area Network

LP - Low Power

LPWAN - Low Power Wide Area Network

LTE - Long Term Evolution

LTE-M - Long Term Evolution Machine Type Communication

M2M - Machine-to-Machine

MAC - Medium Access Control

MCL - Maximum Coupling Loss

NwkSKey - Network Session Key

OTAA - Over The Air Activation

OVSF - Orthogonal Variable Spreading Factor

RFID - Radio Frequency Identification

RTS - Request to Send

PCS – Personal Communication Service

SF - Spreading Factor

SINR - Signal-to-Interference-plus-Noise-Ratio

SNO - SIGFOX Network Operator

UMTS - Universal Mobile Telecommunication System

UNB - Ultra Narrow Band

Introdução

Atualmente as redes de Internet das Coisas estão espalhadas por todas as áreas e setores, proporcionando um maior controlo sobre algumas atividades do dia a dia ou ajudando nas empresas que as utilizam.

As aplicações para este tipo de redes de longo alcance e baixo consumo são vastas, desde o controlo do ar de uma cidade, à medição da humidade do solo num campo, esta tecnologia auxilia estas aplicações, que se estivessem dependentes de uma rede de Internet tradicional, não eram exequíveis devido ao alcance e custo.

1.1. Enquadramento

Com esta tecnologia a mobilidade é um fator muito importante de considerar porque o facto de os dispositivos conseguirem ter uma autonomia de anos dependendo da aplicação e um tamanho pequeno, permite que sejam pensados de uma forma mais transportável, com variados usos nas áreas da pecuária, da criação de gado, na agricultura, no controlo da qualidade do ar, ou o consumo de água, medição do nível do lixo entre outras.

Conforme são implementadas novas aplicações desta tecnologia surge o problema da utilização de um meio partilhado, a colisão de tramas. Este problema cresce devido a todas as estações LoRa estarem a escuta de transmissões o que torna a mobilidade de dispositivos uma preocupação na implementação destas redes.

A probabilidade de colisões de tramas também é afetada por características de tráfego, particularmente na periodicidade, o tamanho da transmissão e da carga útil. Não obstante, o fator mobilidade dos dispositivos Long Range (LoRa) introduz um nível adicional de complexidade, influenciando o desempenho das redes de forma muito relevante contribuindo para cenários interferentes e níveis de colisão que importam considerar e avaliar.

1.2. Motivação e Objetivos

Pretende-se, assim, com esta tese investigar o desempenho desta rede em cenários de mobilidade e trânsito dos dispositivos. Através do desenvolvimento e implementação de

modelos de mobilidade aplicados aos dispositivos finais (nós), adaptativos às condições do canal, pretende-se avaliar o desempenho global em cenários de mobilidade através de simulação (simulador LoRaSim), considerando como técnicas de acesso aleatório Carrier Sense Multiple Access (CSMA) Collision Avoidance and Detection (CAD) e CSMA Collision-Avoidance (CA).

1.3. Estrutura da Dissertação

A estrutura da tese consiste no Capítulo 2, ser feita uma análise sobre a rede LoRa [1] e a especificação LoRa Wide-Area Network (LoRaWAN) [2], o Capítulo 3 é dedicado à apresentação dos protocolos de acesso ao meio. O Capítulo 4, menciona a arquitetura do simulador LoRaSim [3], já o Capítulo 5, apresenta os modelos de mobilidade geralmente utilizados no estudo das redes moveis. O Capítulo 6, aborda o modelo de mobilidade desenvolvido e a sua implementação no simulador LoRaSim. Por fim nos dois últimos capítulos, Capítulo 7 e Capítulo 8, são apresentados os resultados obtidos da simulação e as conclusões respetivamente.

Durante o estudo e pesquisa, foram escritos os Capítulos 2 e 3, que abordam respetivamente a tecnologia LoRa e os protocolos de acesso ao meio.

O Capítulo 2, aprofunda a tecnologia LoRa falando sobre a sua aplicação nas redes Internet of Things (IoT) no subcapítulo 2.1, passando então para o que caracteriza uma Low Power Wide Area Network (LPWAN) [4] no ponto 2.2, falando com mais detalhe sobre a tecnologia LoRa no subcapítulo 2.3 e por fim a especificação LoRaWAN no ponto 2.4.

Depois para se poder averiguar o melhor método de acesso ao meio, e que faça mais sentido no contexto que se pretende avaliar, estudou-se as técnicas atualmente utilizadas e algumas que estão atualmente a ser propostas para utilização.

O Capítulo 3 aprofunda as várias técnicas de acesso aleatório ao meio começando pelo ALOHA [5] e a sua variante Slotted ALOHA [6] [7] no subcapítulo 3.1, passando então para as variantes do protocolo CSMA [8] falando do CSMA Collision-Detection (CD) no subcapítulo 3.2 e o CSMA Collision-Avoidance (CA) no subcapítulo 3.3, por fim são faladas nas novas abordagens como a variante CSMA Collision Avoidance and Detection (CAD), WiseMAC [9] e SMAC [10] no subcapítulo 3.4.

O Capítulo 4 descreve o simulador LoRaSim [3] no subcapítulo 4.1 fala sobre o propósito da criação do simulador LoRaSim pelos autores de [11] e os requisitos para o utilizar. O subcapítulo 4.2 explora os detalhes do simulador, tal como, os parâmetros LoRa, os cálculos

utilizados, e as variáveis que são possíveis de alterar. No subcapítulo 4.3 é explicada a forma de tratamento das colisões no simulador e os critérios utilizados. Por fim no subcapítulo 4.4 são abordados os protocolos de acesso aleatório implementados, tal como o CSMA/CA e o CSMA/CAD.

No Capítulo 5 abordam-se os modelos de mobilidade, no subcapítulo 5.1 é feito um estado da Arte, no subcapítulo 5.2 explica-se os modelos determinísticos de mobilidade e no subcapítulo 5.3 fala-se sobre os *datasets*.

1.4. Publicações Realizadas

Durante o período de realização desta dissertação, com base na parte inicial do estudo sobre os protocolos de acesso ao meio em redes LoRa foi publicado um artigo [12] na conferência INForum 2022 [13]. O título do artigo é: “*Development and Analyses of a New Implementation of CSMA/CAD algorithm for LoRaSim Simulator Applied to LoRa Networks*” teve como base a adição que foi feita ao simulador LoRaSim de novos protocolos de acesso ao meio e seu desempenho. Isto levou a testar a sua eficácia comparando o desempenho do protocolo ALOHA, utilizado atualmente, com a implementação feita dos protocolos CSMA. Foram realizados testes com 1 gateway e com 4 gateways para ser observada a influência que a adição de gateways em conjunto com novos métodos de acesso ao meio, têm no desempenho de uma rede LoRa.

Foi ainda apresentado na conferência URSI da ANACOM [14] um artigo com o nome, “Estudo e Análise da Mobilidade de Dispositivos em Redes LoRa” que demonstra o efeito da mobilidade nos dispositivos moveis e o impacto que diferentes métodos de acesso ao meio têm sobre o seu desempenho e consumo energético. No artigo é descrita a implementação feita do modelo de mobilidade Gauss-Markov e analisado o efeito que tem sobre o consumo de um dispositivo e sobre a probabilidade de colisão.

Adicionalmente foi ainda publicado na revista Brazilian Journal of Development [15] o artigo “*Development and analysis of new implementations of MAC protocols and mobility models in LoRa networks*” [16], que aprofunda o tema abordado em [12] juntando uma comparação mais aprofundada dos impactos nas comunicações LoRa, tendo sido este publicado adicionalmente no livro “Ciências exatas e suas tecnologias” [17], e foi proposto e aceite para publicação o artigo “*Development and Analysis of Mobility Models in LoRa Networks*” [18] que engloba todo o estudo feito na tese bem como os modelos de mobilidade utilizados e desenvolvidos analisando a diferença que a mobilidade traz para o desempenho da rede LoRa.

Rede LoRa

A tecnologia LoRa, é uma tecnologia desenvolvida pela *Semtech* [1] e que utiliza a modulação de espalhamento no espectro, o que lhe confere um longo alcance, pouco consumo de energia, robustez de comunicação e uma grande capacidade para a rede.

Este capítulo aprofunda a tecnologia LoRa falando sobre a sua aplicação nas redes IoT no subcapítulo 2.1, passando então para o que caracteriza uma LPWAN no 2.2, falando com mais detalhe sobre a tecnologia LoRa no 2.3 e por fim a especificação LoRaWAN no subcapítulo 2.4.

2.1. Redes IoT

Foi um termo primeiro utilizado em 1999 para promover a tecnologia de identificação por radiofrequência, ou *Radio Frequency Identification* (RFID). O conceito de IoT é um conjunto de objetos com inteligência, meios de comunicação e capacidades de atuar, ligados por uma rede Internet Protocol (IP) ou por um dispositivo que o permita comunicar com a Internet. Normalmente, são utilizados em ambientes que criam algumas restrições tais como recursos limitados, computação, armazenamento de dados, memória, energia ou o custo dos equipamentos.

A rede LoRa foi criada para responder a necessidade de existir um protocolo que tenha um longo alcance, baixo consumo e comunicação bidirecional. Para estas aplicações se fosse utilizada uma rede de dados convencional o custo dos dispositivos e bateria bem como a escalabilidade da rede tornaria essa aplicação impossível de realizar. Este tráfego de baixa prioridade tornou-se assim uma rede paralela ao serviço destes dispositivos sem as exigências e capacidades de uma rede normal.

O interesse em desenvolver e adotar a tecnologia LoRa tem aumentado rapidamente. Um bom exemplo disto é o facto de uma das maiores operadoras de telecomunicações holandesas, KPN, ter anunciado uma cobertura total de LoRa pelo país [19]. A comunidade aberta The Things Network (TTN) que fornece código para a rede LoRaWAN, já possui uma presença pelo mundo inteiro. Para além disto o uso de frequências reservadas para fins

industriais, científicos e médicos (ISM) que não necessitam de licença abre a possibilidade para a criação de redes LoRa individuais e privadas.

Alguns dos desafios que são impostos nas redes IoT incluem a disponibilidade, mobilidade, o desempenho, a escalabilidade, segurança, gestão e confiança.

A disponibilidade vem da capacidade que o *hardware* de um dispositivo tem para integrar funcionalidades e protocolos para as redes IoT

A mobilidade é outro dos desafios de uma implementação IoT porque muitos dos serviços são fornecidos a utilizadores ou têm aplicações em movimento. Uma interrupção no serviço pode ocorrer quando os dispositivos são transferidos de uma *gateway* para outra.

O desempenho de um serviço IoT depende do desempenho dos componentes e das tecnologias por detrás deles. A escalabilidade de um sistema pode ser avaliada pela habilidade de adicionar novos dispositivos, serviços ou funções sem afetar negativamente a qualidade do serviço que já é prestado.

A segurança e privacidade apresentam um verdadeiro desafio para as implementações devido á falta de normas e arquiteturas para segurança em IoT. Em redes heterogéneas como é o caso do IoT, não é fácil garantir a segurança e privacidade dos utilizadores, visto que a funcionalidade mais básica é a troca de informação entre todos os objetos ligados à rede, é necessário assegurar as trocas de dados para evitar perder ou comprometer a privacidade dos dados sensíveis.

Para as aplicações de IoT que requeiram comunicações de longas distâncias, tais como em zonas rurais onde, ao contrário de cidades, não tem a possibilidade de ter uma abundância de infraestruturas, tecnologias como Bluetooth 4.0 e Wi-Fi [20] não são as indicadas devido ao seu curto alcance, isto levou ao desenvolvimento de várias tecnologias LPWAN como a LoRa. Esta modulação, que deriva do Chirp Spread Spectrum (CSS), faz com que ela seja robusta ao ruído no canal e adiciona alguma segurança, porque a transmissão é espalhada na frequência o que se torna parecido com ruído sendo possível assim a transmissão de informação através de longas distâncias com um baixo consumo de energia.

Devido aos seus aspetos únicos, como o longo alcance e o baixo consumo, as redes LoRa atraíram uma atenção significativa da indústria e dos setores académicos em anos recentes. LoRa utiliza uma banda de frequências de licença livre ISM e oferece a conectividade de longo alcance a dispositivos de pouco consumo. Por outro lado, o protocolo LoRaWAN proposto pela LoRa Alliance [2] define o protocolo de rede de controlo de acesso ao meio (MAC) e a camadas de redes.

Uma rede IoT implementada em LoRa consiste de vários componentes, tais como o módulo LoRa, uma *gateway* LoRa, um servidor de rede LoRa e um servidor de aplicação LoRa. Os módulos utilizados são geralmente alimentados por uma bateria, tem poder computacional limitado e é o servidor de rede LoRa que tem de ter a responsabilidade de processar os pacotes recebidos e suportar os módulos LoRa existentes na rede.

Outras tecnologias também prominentes em IoT são NB-IoT, ou Narrowband IoT (NB-IoT) e Sigfox [21]. Na Tabela 1 é possível ver uma breve comparação entre as tecnologias.

Tabela 1 - Tecnologias LPWAN.

	LoRa	SigFox	NB-IoT
Alcance	<15km	<50km	<15km
Taxa de Dados	50kbps	100kbps	250kbps
Custo por modulo	Baixo	Bastante Baixo	Baixo
Encriptação	Sim (AppSKey)	Não	Sim (EEAx)

A rede NB-IoT é um standard de LPWAN que utiliza a rede celular e permite que os dispositivos IoT enviem dados diretamente para a *cloud* sem a existência de uma *gateway*. Os standards de redes celulares tradicionais focam-se em aplicações de voz ou aplicações de dados de baixa latência. Para IoT estas aplicações não são escaláveis devido ao seu custo elevado, baixo alcance e o consumo do dispositivo. O NB-IoT reduz a complexidade, limita a largura de banda e os débitos binários dos dispositivos. [22]

Uma vez que o sistema é similar ao funcionamento da tecnologia de comunicação dos telemóveis, os dispositivos utilizam um cartão Subscriber Identity Module (SIM) com rede para operarem. Por outro lado, os equipamentos necessitam que a quantidade de dados enviados seja reduzida e a largura de banda pequena.

Como principais vantagens esta tecnologia, apresenta um reduzido custo de implementação, a possibilidade de interligação, em massa, de equipamentos, baixos consumos energéticos, o que permite uma maior duração das baterias, elevados alcances e uma maior penetração do serviço, sendo que pode operar em zonas em que outras tecnologias não chegam. [23]

Sigfox é uma empresa de tecnologia francesa fundada em 2009. A solução que oferecem, designada Sigfox, é uma rede baseada numa topologia de estrela de um salto e consiste de dispositivos de transmissão, estações base/*gateways* e o *back-end* Sigfox. A cobertura da rede

é proporcionada em cada país pelo parceiro escolhido designado de Sigfox Network Operator (SNO).

A banda utilizada pela tecnologia Sigfox é a banda radio ISM, nomeadamente frequências 868 MHz na Europa e 902 MHz nos Estados Unidos. Devido a natureza física da Ultra Narrow Band (UNB) com salto de canal aleatório, a contribuição do ruído é bastante baixa. [24]

A dezembro de 2021 a rede IoT Sigfox cobre 6 Milhões de metros quadrados e está presente em 75 países. [21]

2.2. LPWAN

A tecnologia LPWAN surgiu em 2013 como um termo para uma classe de tecnologias sem fios que se adequa as necessidades de comunicações *machine-to-machine* (M2M) e dispositivos de IoT [4].

A grande vantagem das LPWAN é a sua eficiência, permitindo que um dispositivo funcione até 10 anos numa só bateria. Esta durabilidade é ideal para soluções com mobilidade reduzida e falta de uma fonte de alimentação.

As redes de baixa potência e grande área são um tipo de comunicação de redes sem fios criada para permitir comunicações de longo alcance, mas de baixo débito entre vários dispositivos, tal como sensores. O baixo consumo, baixo débito e a sua utilização distinguem este tipo de rede de uma rede sem fios normal que tem o propósito de ligar utilizadores ou empresas, carregar mais dados e usando mais energia.

Os atributos destas redes são, o seu longo alcance em que o raio de operações de LPWAN varia de uns poucos quilómetros em zonas urbanas para mais de 10 km em zonas rurais. O seu baixo consumo visto que é otimizado para tal, os seus equipamentos podem usar baterias pequenas e pouco dispendiosas. O seu baixo custo devido aos protocolos usados serem relativamente simples, leva a uma redução em complexidade em hardware design e reduz os custos dos dispositivos.

O seu longo alcance combinado com a sua topologia de rede em estrela reduz custos em infraestruturas e o uso de uma banda de licença livre reduz os custos da rede, ambos estes fatores são o que propulsiona o grande crescimento deste tipo de redes.

O organismo 3rd Generation Partnership Project (3GPP) é um termo para as organizações que definem os protocolos e criam os *standards* para as comunicações moveis. Entre as quais são mais conhecidos pelo desenvolvimento e manutenção de tecnologias como o Global System

for Mobile (GSM), Universal Mobile Telecommunication System (UMTS), Long Term Evolution (LTE) e 5ª Geração (5G) [25]

O Long Term Evolution Machine Type Communication (LTE-M) é um standard publicado pelo 3GPP, também conhecido como enhanced Machine-Type Communication (eMTC). Esta tecnologia permite a utilização da infraestrutura LTE já existente o que permite uma alta eficiência e uma grande cobertura, suportando a mobilidade com *handover*. [26]

A tecnologia NB-IoT, pretende diminuir os custos dos terminais, sendo otimizado para módulos sem fios mais baratos o que diminui o consumo de energia fornecendo uma vida de bateria mais longa e estende a cobertura de radio para suportar longo alcance e comunicações dentro de edifícios. Estas características permitem que seja utilizado um número elevado de dispositivos conseguindo conectar até 50,000 dispositivos IoT por célula NB-IoT. [27] Como é possível ver na Figura 2.1 existem várias tecnologias às quais o 3GPP atribuiu standards e outras que não estão padronizadas.

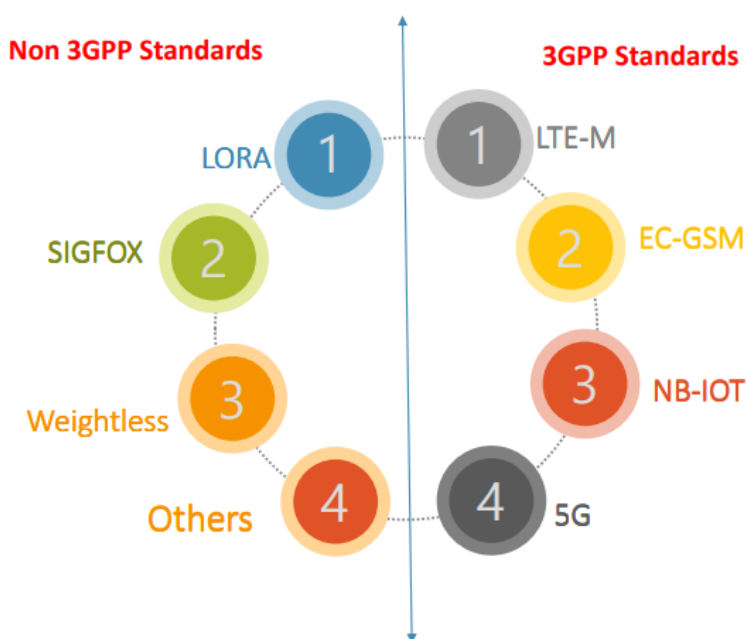


Figura 2. 1 Tecnologias Long Range, retirado de [20] .

Existem também várias normas e soluções de fabricantes na categoria de LPWAN, os mais predominantes são: LoRa, Sigfox, Zigbee, Weightless entre outros.

O comité 3GPP define a perda máxima por acoplamento (MCL) como a métrica pela qual avaliar a cobertura de uma tecnologia de acesso por rádio. Em teoria, pode ser definido como a perda máxima que um sistema pode ter e ainda ser operacional. MCL pode ser calculado como a diferença entre os níveis de potência medida no transmissor e o no recetor. [28]

$$MCL (dB) = P_{Tx} - (Noise Figure + SINR + Noise Floor) \quad (2.1)$$

Onde *Noise Figure* é a degradação de sinal devido ao ruído do próprio sistema, o *Signal-to-Interference-plus-Noise-Ratio* (SINR) mede a qualidade do sinal, comparando a potência do sinal com a potência da interferência e do ruído, e o *Noise Floor* é o nível menor de potência de sinal que consegue ser detetado pelo sistema na presença de ruído.

Tecnologias que utilizam o espectro não licenciado como LoRa e Sigfox são sujeitas a limitações como potência máxima de emissão e *duty cycle*. *Duty cycle* é definido como a percentagem máxima de tempo no ar por hora, geralmente recomendado ter um valor de 1%.

A tecnologia Sigfox [21] é a primeira tecnologia LPWAN e foi criada pela companhia francesa com o mesmo nome em 2009. Opera no espectro não licenciado dos 868 MHz na europa e tem um alcance de 30 a 50 km em áreas rurais e de 3 a 10 km em áreas urbanas. Utiliza a tecnologia patenteada de Ultra-Narrow Band (UNBac) que ocupa uma porção muito pequena do espectro e utiliza a modulação Differential Binary Phase Shift Keying (DBPSK).

A tecnologia LoRa é uma tecnologia de modulação de sinal desenvolvida pela empresa francesa Cycleo [29] e mais tarde comprada pela empresa Semtech, e os dispositivos e modulação tem vindo a ser reconhecidos como um padrão para IoT [20]. Podemos ver na Tabela 2 uma comparação entre várias tecnologias utilizadas em LPWANs.

Tabela 2 - Comparação entre tecnologias LPWAN, adaptada de [26].

Parâmetros	LoRa	SigFox	EC-GSM	LTE-M	NB-IoT
Alcance	<15km	<50km	<15km	<11km	<15km
Cobertura (MCL)	157dB	153dB	164dB	160dB	164dB
Taxa de Dados	50kbps	100kbps	74kbps	1Mbps	250kbps
Espectro	Não licenciado EU 868MHz, 433MHz US 915MHz	Não licenciado EU 868-869MHz US 902-928MHz	banda GSM	banda LTE	LTE em banda, banda guarda, stand-alone
Largura de Banda	<500kHz	100kHz	200KHz	1.08MHz	180kHz
Tecnologia de radio	Spread Spectrum	UNB	TDMA/FDMA	OFDM	OFDM
Potência de transmissão máximo	14dBm	20dBm	23, 33dBm	20, 23dBm	23dBm
Autonomia	>10 anos	>10 anos	>10 anos	>10 anos	>10 anos
Reutilização de redes celulares	Não	Não	Sim	Sim	Sim
Confidencialidade de Dados	Sim (AppSKey)	Não	Parcial (GEA4/5)	Sim (EEAx)	Sim (EEAx)
Autenticação de rede	Opcional	Não	UMTS AKA	LTE AKA	LTE AKA
Custo por modulo	Baixo	Bastante Baixo	Baixo	Médio	Baixo

A especificação LoRaWAN vem a definir a arquitetura a ser utilizada pela rede LoRa. A especificação define três modos de operação, Classe A, Classe B e Classe C. Grande parte das implementações utiliza os dispositivos em Classe A, que é o mais eficiente em energia e obrigatório em todos os dispositivos devido á especificação [2]. O modo de transmissão de dados utilizados na classe A é baseado no protocolo ALOHA Puro [5], onde se os dispositivos têm informação para enviar esta é enviada e se durante o processo é recebida uma mensagem de outro dispositivo significa que existiu uma colisão, contudo esta falta de sincronismo leva a um elevado número de colisões de pacotes especialmente se o número de dispositivos aumentar. No capítulo 2.4 é abordado o tópico de LoRaWAN mais aprofundadamente.

2.3. LoRa

A tecnologia LoRa, é uma técnica de modulação de espectro proprietária desenvolvida pela Semtech [1] de baixo consumo e longo alcance, que se baseia na modulação CSS o que lhe confere robustez contra ruído e um maior alcance mantendo um baixo consumo energético, fazendo parte das tecnologias de LPWAN.

Esta modulação troca ritmo binário por sensibilidade dentro de uma largura de banda fixa num canal. Ao aumentar a duração de símbolo com o aumento do SF o ritmo binário diminui, mas possibilita a operação a menores potências e a longas distâncias. A tecnologia LoRa é uma implementação da camada física e é agnóstica das implementações de camadas superiores o que permite a coexistência com arquiteturas de redes existentes.

Como a tecnologia utiliza uma modulação inferior a banda GHz é possível obter um longo alcance, um pouco consumo, uma grande capacidade de rede e robustez de comunicações.

Tabela 3 - Detalhes da rede LoRa, adaptado de [30].

Parametro	Europa	E.U.A	China	Coreia	Japão	India
Banda de frequência (MHz)	867-869	902-928	470-925	920-925	920-925	867-869
Canais	10	64+8+8	Definidos pelo Comité técnico			
Largura de Banda - UpLink (KHz)	125/250	125/500				
Largura de banda - DownLink	125	500				
Potência de Transmissão - UpLink	+16 dBm (40 mW)	+20 dBm (100 mW)				
Potência de Transmissão - DownLink	+16dB	+27dB				
Spreading Factor	7-12	7-10				
Velocidade de Transmissão	250 bps-50 kbps	980 bps - 21,9 kbps				

Esta rede usa uma banda de frequências de licença livre tal como os 433 MHz, 868 MHz (Europa), 915 MHz (Austrália e América do Norte) e 923 MHz (Ásia) (ver Tabela 3).

Na região europeia existem duas bandas de frequência, 433 MHz e 868 MHz. A banda nos 433 MHz que é classificada como uma banda livre teoricamente teria um alcance maior que um dispositivo na banda dos 868 MHz, contudo para estar nos 433 MHz a antena teria de ser maior e teria uma restrição de potencia que contrariava a possibilidade de ter mais alcance.

Como essa banda é a utilizada para rádio amador, entre outras tecnologias ela é mais congestionada, logo não obtemos as vantagens que seriam de esperar por utilizar uma frequência inferior. Por estas razões as várias aplicações desenvolvidas para LoRa são feitas na banda dos 868 MHz. Estas aplicações geralmente estão divididas em vários níveis, como podemos ver na Figura 2.2, o físico, que é representado pelos dispositivos terminais e as *gateways* onde ocorre a comunicação rádio, o nível da rede, onde ao chegarem à *gateway* os pacotes são reencaminhados para a rede a que o dispositivo se encontra ligado, e a camada mais alta que é a da aplicação, onde os dados são processados e apresentados.

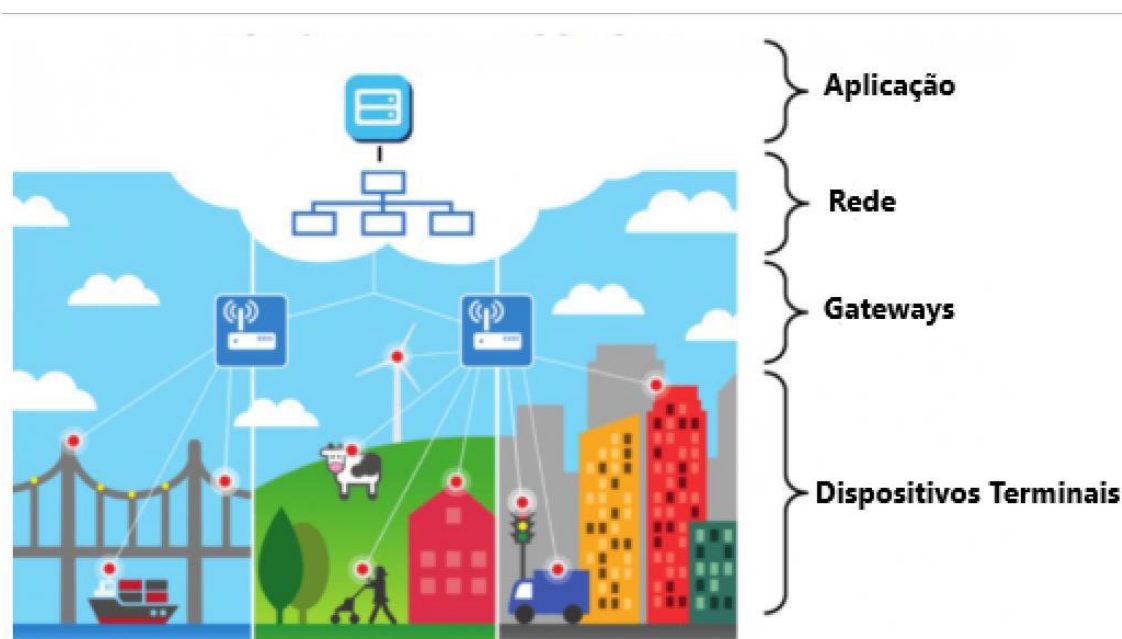


Figura 2. 2 - Arquitetura LoRa, retirado de [30]

As transmissões LoRa podem ser caracterizadas por alguns parâmetros chave como a Largura de Banda, a Frequência portadora, o *code rate*, o fator de espalhamento e a potência de transmissão.

A largura de banda que define um leque de frequências rádio usadas na transmissão. Um aumento da largura de banda de um sinal permite um uso de uma taxa de dados superior, pelo

que reduz o tempo que ocupação do meio sacrificando sensibilidade no recetor uma vez que se torna mais sensível ao ruído.

A Frequência da Portadora (CF) define a frequência central utilizada para a transmissão na rede LoRa.

O Code Rate (CR) define o Forward Error Correction (FEC) a ser usado para proteção contra erros. Pode ser configurado para 4/5, 4/6, 4/7 ou 4/8 onde o seu valor mostra a relação entre bits de dados e bits de FEC, logo se se utilizar o valor mais baixo de CR, 4/5 ter-se-á 4 bits de informação e 1 de paridade e como oposto quanto maior o CR maior a proteção contra erros, porque com um CR de 4/8 só cada 4 em 8 bits é que contém a informação que é enviada, aumentando, em simultâneo, o tempo de ocupação do meio que em torno aumenta a energia gasta.

O Spreading Factor (SF) influencia a taxa de dados, o tempo no ar, o tempo de bateria e a sensibilidade do recetor. O fator de espalhamento pode ir de SF7 a SF12, um valor baixo de SF significa uma taxa de transmissão mais alta, mas o recetor fica mais sensível ao ruído, um SF mais alto, como SF12, consome mais energia porque fica mais tempo a transmitir, tem uma taxa de transmissão mais baixa, mas é mais robusto contra o ruído.

A potência de transmissão influencia o alcance da transmissão e o consumo energético, pode ser ajustada de 2 dBm a 20 dBm mas potências superiores a 17dBm têm de utilizar um duty-cycle de 1%.

2.3.1. Spread Spectrum Communications

O teorema Shannon-Hartley [1] dita o ritmo máximo de transmissão a que informação pode ser transmitida sobre um meio de comunicação numa largura de banda na presença de ruído.

$$C = B * \log_2\left(1 + \frac{S}{N}\right) \quad (2.2)$$

Onde:

- C , Capacidade do canal (bit/s)
- B , Largura de banda (Hz)
- S , Potência média do sinal recebido [W]
- N , Potência média do nível de ruído [W]

Segundo o teorema ao aumentar a largura de banda do sinal é possível compensar a degradação da relação sinal ruído (SNR) do canal.

Num sistema Direct Sequence Spread Spectrum (DSSS), a fase da portadora muda de acordo com uma sequência de código. Este processo é geralmente feito multiplicando os dados que queremos e enviando com um código de espalhamento, também conhecido como *chip sequence*. Esta sequência ocorre a uma velocidade muito maior que o sinal de dados e, como tal, aumenta a largura de banda do sinal para além da ocupada pelo sinal original. Na Figura 2.3 pode-se observar o processo de modulação onde os dados são modulados com a sequência e origina o sinal que irá ser transmitido.

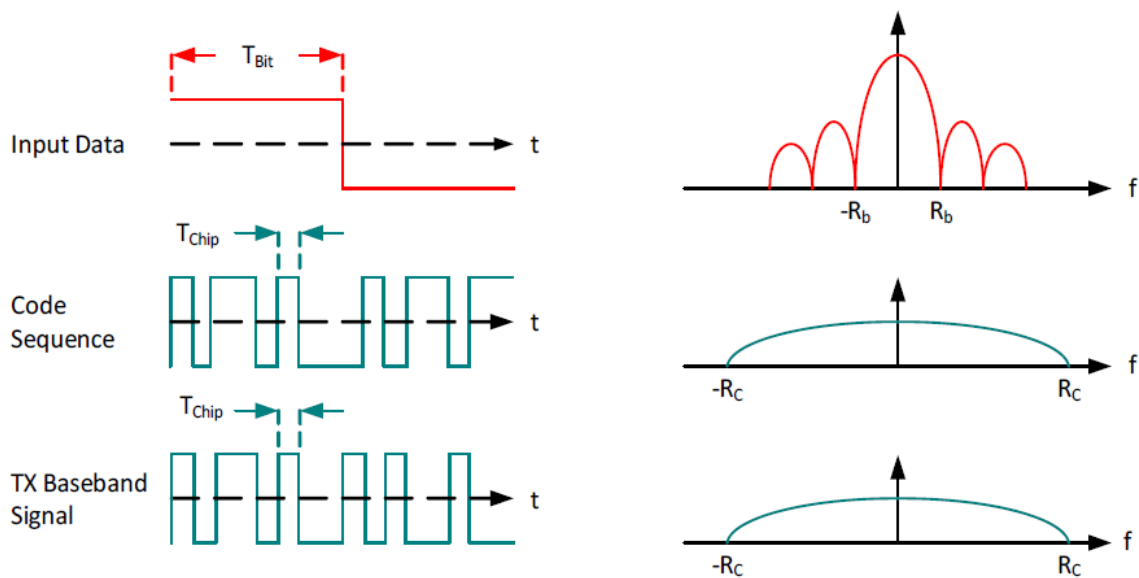


Figura 2. 3 - Processo de Modulação/Spreading, retirado de [1].

$$R_c = 2^{SF} R_b \quad (2.3)$$

No recetor, o sinal é multiplicado por uma replica local da sequência de espalhamento. Esta multiplicação faz com que o sinal volte a comprimir a largura de banda para o tamanho original sendo este processo o inverso da Figura 2.3.

Contudo, o DSSS requer que o sistema tenha um relógio bastante preciso em adição ao facto de quanto maior o fator de espalhamento ou a sequência, maior o tempo que é requerido pelo recetor para fazer a correlação com a sequência ou código de espalhamento.

Como os dispositivos utilizados com LoRa tem restrições de energia não lhes é possível estar ligados constantemente, e como tal, é necessário que eles se sincronizem rapidamente.

O CSS foi desenhado para radares e é fortemente utilizado no ramo militar devido a baixa energia de transmissão, robustez ao nível da degradação de canal e robustez ao efeito de Doppler, que ocorre quando um emissor está em movimento o que causa uma variação do sinal transmitido na frequência. O desvio na frequência causado pelo efeito de Doppler na comunicação LoRa traduz-se num negligível desvio no tempo que não afeta a comunicação.

Uma vez que utiliza toda a largura de banda que lhe é atribuída para enviar um sinal, e como o modo de funcionamento de descodificação do sinal não necessita de precisão na frequência devido ao utilizar toda a largura de banda, torna a transmissão forte contra o ruído de canal.

A modulação LoRa vem resolver os problemas associados com o sistema DSSS para fornecer uma solução de baixo consumo e baixo custo, mas acima de tudo uma alternativa robusta as comunicações de espalhamento de espectro.

Um sistema implementado de DSSS é o método Code Division Multiple Access (CDMA) que para aumentar a capacidade de um sistema utiliza códigos de canais que mantem a ortogonalidade entre eles chamado de Orthogonal Variable Spreading Factor (OVSF)[31]. O OVSF é geralmente representado na forma de uma arvore.

A Figura 2.4 mostra uma arvore de códigos com K camadas, onde a camada inferior, chamada de camada folha, representa a capacidade da camada.

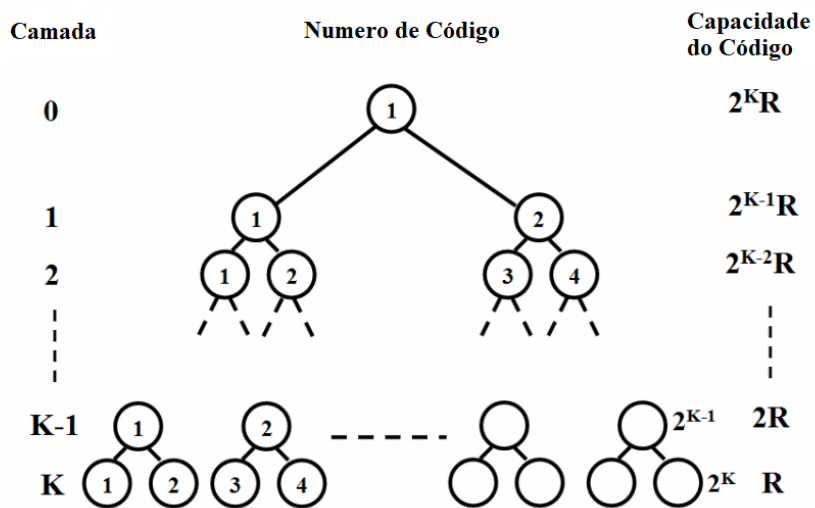


Figura 2. 4 - Arvore de códigos, adaptado de [31].

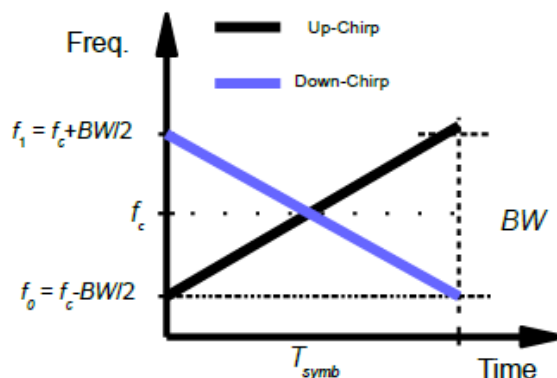


Figura 2. 5 Representação no espectro de um up-chirp e down-chirp, retirado de [32].

Em Lora o espalhamento do espectro é obtido pela geração de um sinal Compressed High Intensity Radar Pulse (Chirp) onde a frequência aumenta ou diminui com o tempo, o aumento da frequência é designado de *upchirp* e uma diminuição é designada de *downchirp*.

Um Chirp começa a passar na frequência linearmente, começando num valor aleatório, e quando a frequência instantânea chega ao seu valor mais alto, por exemplo $f_c + BW/2$, a frequência volta para o seu mínimo, por exemplo $f_c - BW/2$ até o fim do Chirp. Como se pode ver na Figura 2.5. Uma vantagem deste método é que a desfasagem no tempo e na frequência é igual entre emissor e recetor, o que baixa significativamente a complexidade do recetor.

Na Figura 2.6 é possível ver o preambulo da rede LoRa com SF de 7, composto por seis *upchirps* e dois *downchirps* seguidos pelos símbolos 0, 16 e 112.

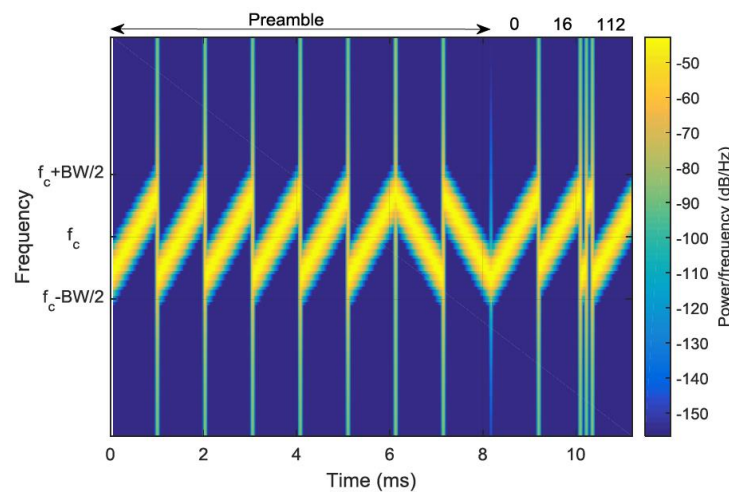


Figura 2. 6 - Símbolo LoRa, retirado de [32].

Um sinal LoRa pode ter três larguras de bandas diferentes, 125 kHz, 250 kHz e 500 kHz. Uma largura de banda maior permite um ritmo binário maior, o que reduz o tempo no ar, mas sacrifica sensibilidade no recetor devido a introdução de mais ruído. Na Figura 2.7 é possível ver o espaço que a LB vai ocupar na frequência.

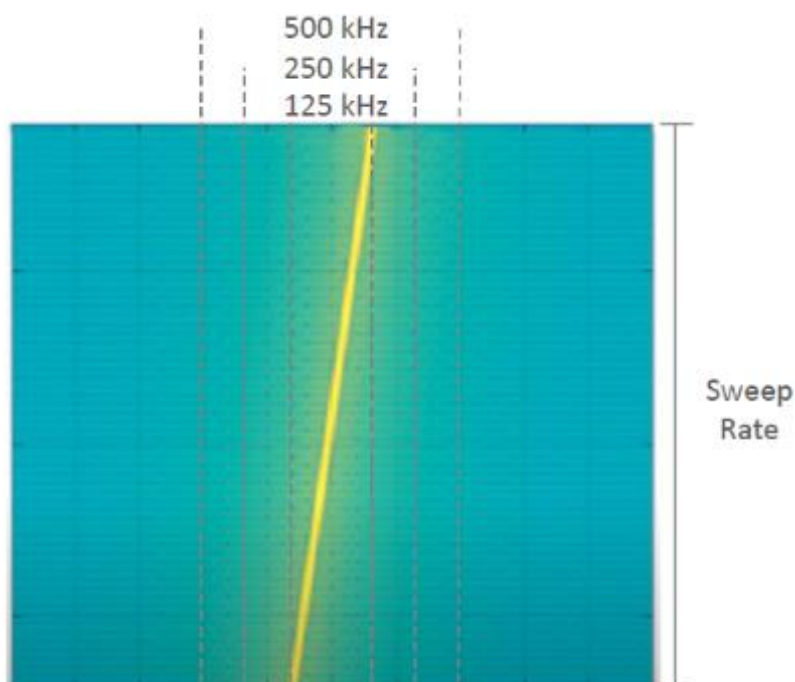


Figura 2. 7 - Largura de banda LoRa, retirado de [32].

Outro grande fator na emissão nesta modulação é o fator de espalhamento, SF, que é classificado de 7 a 12. Dependendo das condições em que a emissão está a ser efetuada é

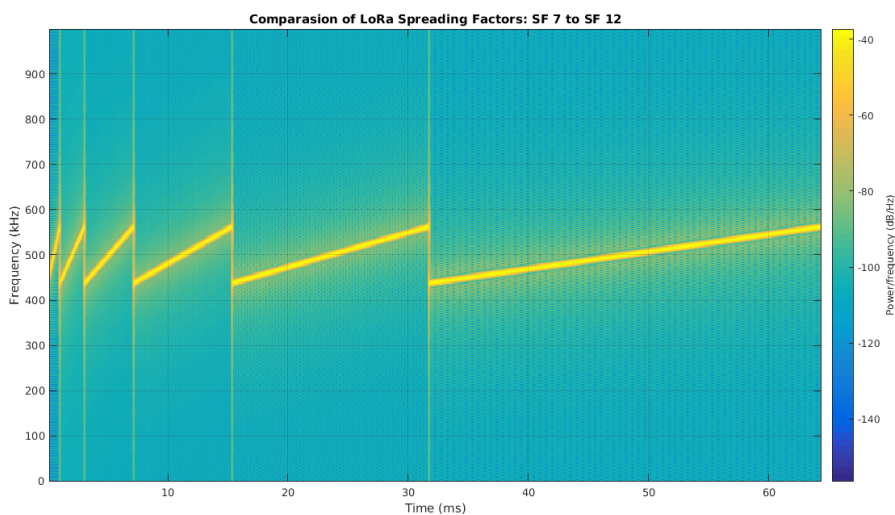


Figura 2. 8 - Fator de Espalhamento LoRa, 7 a 12, retirado de [33].

importante utilizar um SF adequado, se for um valor baixo, como 7, significa que mais *chirps* são enviados por segundo, logo, pode ser codificada mais informação por segundo, um fator de espalhamento mais alto, como 12, implica que menos *chirps* são efetuados por segundo, o que se traduz numa velocidade de transmissão mais lenta. Na Figura 2.7 está representado o espalhamento do sinal no tempo, com uma largura de banda constante de 125 kHz

Transmitindo a mesma quantidade de dados esta alteração de SF tem benefícios e fraquezas. Por um lado, um fator de espalhamento menor consome menos energia e envia informação mais rápido o que significa uma menor ocupação do meio, mas, requer que o recetor tenha mais sensibilidade ou a distância de transmissão é diminuída. Por outro lado, um SF maior significa que o recetor consegue uma melhor sensibilidade, o que permite que o sinal pode ser recebido de mais longe, e torna o sinal mais resistente a ruído, contudo, a utilização do canal por mais tempo significa que transmissor está ligado durante mais tempo o que gasta mais energia.

O tempo de símbolo de um sinal, T_S pode ser obtido através da equação:

$$T_S = \frac{2^{SF}}{BW} [s] \quad (2.4)$$

Onde:

- SF , Fator de espalhamento (7..12)
- BW , Largura de banda (Hz)

O ritmo de símbolo, R_S , que é o inverso do tempo de símbolo:

$$R_S = \frac{1}{T_S} = \frac{BW}{2^{SF}} [símbolos/s] \quad (2.5)$$

O *bit rate* da modulação, R_{bm} , pode ser calculado utilizando a seguinte formula:

$$R_{bm} = SF \times \frac{BW}{2^{SF}} [bits/s] \quad (2.6)$$

A modulação LoRa inclui também um esquema de correção de erros, *code rate* (CR), que aumenta a robustez do sinal transmitido. O CR vai de 1 a 4 e quanto maior o valor, maior a proteção contra erros, mas ao mesmo tempo maior a informação que tem de ser transmitida porque um CR de 4 significa que está a ser enviado o dobro da informação. Podemos definir o bit rate nominal, R_b como:

$$R_b = SF \times \frac{BW}{2^{SF}} \times \frac{4}{4+CR} [bits/s] \quad (2.7)$$

Onde:

- CR , Code Rate (1..4)

Tabela 4 - Ritmo binário de uma transmissão LoRa, valores retirados de [34].

BW[kHz]	SF	CR	Rb [kbit/s]
125	7	1	5.47
125	8	1	3.13
125	10	1	0.98
125	12	1	0.29
250	12	1	0.59
500	12	1	1.17
500	12	2	0.98
500	12	3	0.84
500	12	4	0.73

Na Tabela 4 estão representados alguns valores de ritmo binário que variam com o valor utilizado de tanto fator de espalhamento, largura de banda e *code rate*.

Podem ser feitas algumas simulações de dados a ser transmitidos definindo os parâmetros relevantes para a comunicação.

Tabela 5 - Tempo de transmissão de pacotes LoRa, valores retirados de [35].

SF	Ttx [ms]			
	1 byte	8 bytes	30 bytes	51 bytes
7	26.88	39.17	82.18	119.04
8	53.76	78.34	139.78	213.5
9	107.52	132.1	254.98	377.86
10	215.04	264.19	509.95	706.56
11	430.08	528.38	1,019.90	1,511.42
12	860.16	1,056.77	1,843.20	2,826.24

Na Tabela 5 é utilizada uma largura de banda de 125 kHz, um CR de 2 (4/6) e um preambulo de 8 bytes obtemos os seguintes resultados para o tempo eu um emissor demoraria a transmitir o pacote [1].

Um dispositivo em a transmitir á potência máxima de 14 dBm consome aproximadamente 25 mW. Se quisermos calcular a energia gasta em cada transmissão é necessário descobrir quanto tempo o dispositivo demora a enviar um pacote. Tendo em atenção as tabelas anteriores podemos assumir um SF de 7 e uma largura de banda de 125 kHz. Como o consumo em transmissão por hora é de 25 mW é necessário ver o tempo que um pacote demora a ser enviado para podermos calcular efetivamente o gasto por transmissão.

Um dispositivo que transmita 8 byte vai um tempo no ar de 56.6 ms, o que significa que para respeitar as regras de *duty-cycle* o dispositivo necessita de esperar 5.6 segundos. O consumo de energia na transmissão é possível calcular seguindo a fórmula

$$P_{tx} = T * \frac{P_d}{3600} \quad (2.8)$$

Onde:

- T , Tempo no ar (s)
- P_d , Potência transmissão dispositivo, [W]

Com a função é possível verificar que um dispositivo a enviar 8 byte gasta sensivelmente 0.4 μ W e pode voltar a transmitir ao fim de 5.6 segundos o que o leva a poder transmitir 642 vezes por hora.

Com SF de 12 e 250 kHz de largura de banda já obtemos um tempo de transmissão de 659.5ms o que leva ao consumo do dispositivo a ser 5.48 μ W. O efeito do SF sobre o consumo de energia é bastante significativo comparativamente com a potência de emissão ou a quantidade de dados enviados.

2.4. LoRaWAN

A especificação LoRaWAN é um protocolo de redes baseado em LoRa que define a comunicação e a arquitetura do sistema para a mesma rede. As especificações de LoRaWAN criadas e mantidas pela LoRa Alliance™ [2] regulam o uso da tecnologia LoRa nas bandas de frequências sem necessidade de licença, ISM [36] e atua como um protocolo de controlo de acesso ao meio, mas desempenha principalmente o papel de protocolo de camada de rede. Na Figura 2.9 pode-se observar as camadas da especificação LoRaWAN, a camada física é a modulação LoRa, e a camada MAC é feita pela especificação LoRaWAN.

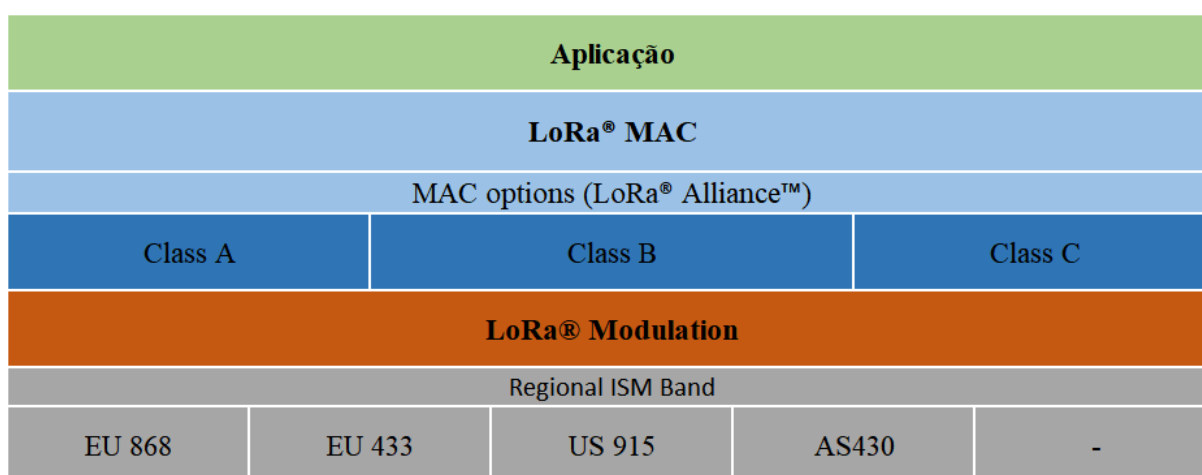


Figura 2. 9 - Camadas da tecnologia LoRaWAN, adaptado de [37].

Utilizando a modulação LoRa como camada física, a especificação LoRaWAN implementa as camadas *link*, *network* e *session* do modelo OSI especificando também o formato e a forma como os pacotes são processados na rede.

A arquitetura de uma rede LoRaWAN é composta pelos dispositivos finais, uma *gateway*, um servidor de rede e um servidor de aplicação. Os dispositivos finais são os sensores que utilizam a modulação LoRa para comunicar com as *gateways*, sendo que não é possível associar um dispositivo a uma *gateway*, todas as *gateways* á escuta no canal conseguem receber a mensagem. As *gateways* servem para desmodular os sinais recebidos e através da sua ligação a uma rede de Internet, celular ou satélite envia os dados para um servidor de rede.

O servidor de rede é responsável pela ligação com as *gateways*, a de-duplicação de pacotes recebidos por diferentes *gateways* ou recebidos de outras redes (*roaming*). É também responsável pelo registo dos dispositivos e gera os identificadores necessários, o *device address*, *Session Keys* e *frame Counters* para o dispositivo se ligar a rede. Com esta

implementação as *gateways* tornam-se transparentes para os dispositivos finais porque passam a estar associados com um servidor de rede.

Os dispositivos têm um identificador global único, DevEUI, e um identificador único na rede, DevAddr, devido a estes identificadores únicos, os dispositivos ao enviarem mensagens para a rede elas podem ser capturadas por mais que uma *gateway*, contudo no servidor de rede as mensagens duplicadas são retiradas e é o servidor de rede que decide a *gateway* mais indicada para fazer as mensagens de *downlink*. Com a adição de uma aplicação também é necessário lhe fornecer a identificação da aplicação que lhe está associada, AppEUI.

Numa rede LoRaWAN os dispositivos estão divididos entre classes diferentes, classe A, classe B e classe C. Cada classe representa um modo diferente de funcionamento, cada um com as suas vantagens e desvantagens. O funcionamento das diferentes classes encontra-se representado na Figura 2.10 que se segue.

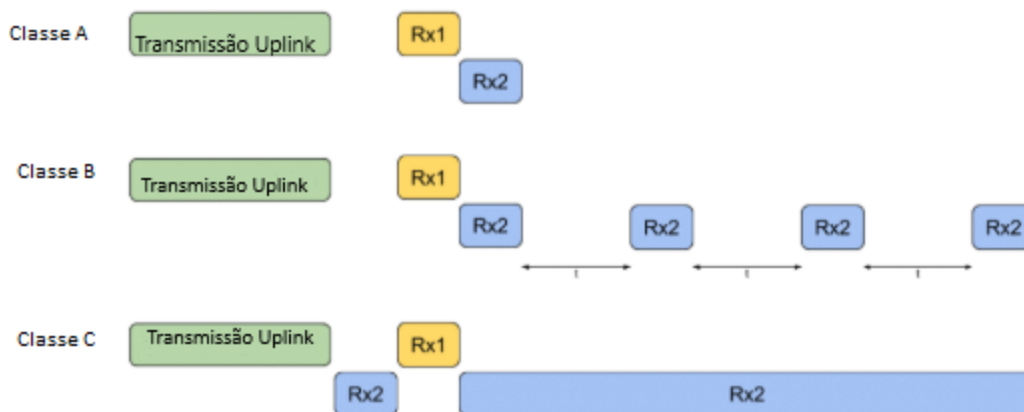


Figura 2. 10 - Diferentes tipos de Classes, retirado de [37].

A classe A suporta o funcionamento básico e obrigatório de todos os dispositivos para LoRaWAN, bem como comunicação bidirecional. Na transmissão *uplink* o dispositivo utiliza um tempo de acesso aleatório, na transmissão *downlink* existem longos atrasos e apenas podem ser utilizados os intervalos de tempo dedicados, chamados de janelas de recepção, que são precedidas pela transmissão *uplink* bem-sucedida. Desta forma a classe A consegue, para redes com tráfego reduzido, atingir consumos de energia inferiores às outras duas classes.

A classe B implementa comunicação bidirecional com tempos de ligação *downlink* programados. Essa informação é disseminada pela *gateway* via *beacons* com o propósito de manter os dispositivos em sintonia.

Os dispositivos de classe C tem os mesmos tempos de comunicação que os dispositivos de classe A mas mantém a janela de escuta sempre ativa, provendo assim a menor latência

downlink, contudo requer um consumo de energia bastante elevado. Só o servidor de aplicação é que consegue saber se é um dispositivo de classe A ou classe C.

As *gateways* em adição a desmodularem o sinal e enviarem para o servidor de rede, envia também meta-dados com informação sobre o estado da ligação, o canal utilizado e outras métricas do sinal recebido. Como a *gateway* é transparente para os dispositivos não guarda nenhuma informação sobre o dispositivo e confere a funcionalidade de *roaming*.

A rede LoRaWAN é maioritariamente constituída pelo servidor de rede e o servidor de aplicação. O servidor de rede é responsável por quase todos os mecanismos LoRaWAN, tem a responsabilidade de ligação com as *gateways*, a seleção dos pacotes e o registo dos dispositivos.

Um dispositivo para estar registado tem de ter três componentes, uma sessão segura, através dos identificadores e das chaves de sessão que o dispositivo tem, as definições dos parâmetros rádio, o modo de operação, a classe do dispositivo, o agendamento da janela de recessão e a frequência a que esta a transmitir, e por fim tem de ter o *data rate* para *downlink* definido.

O controlo de um dispositivo é feito através da identificação do dispositivo que é baseado no endereço, que pode ser partilhado entre vários dispositivos, e requer a chave da sessão na rede para identificar um dispositivo específico. Controla a data rate através da funcionalidade de data rate adaptável (ADR)

O servidor de aplicação recebe e envia mensagens para o servidor de rede, cifra e decifra os conteúdos recebidos dos dispositivos, tornando a informação que recebe legível para as aplicações. Devido a este fator criptográfico da comunicação é importante o servidor ser local para a proteção dos dados recebidos. A grande vantagem do servidor de aplicação é que oferece uma interface web para gerir os dispositivos e consegue ter uma integração com APIs para gerir os dispositivos, receber e enviar dados.

Em termos de segurança existem duas camadas de segurança. Na camada da rede e na camada da aplicação. Todos os *payloads* são encriptados, inclusive nas verificações de integridade. Na camada de rede é feito a verificação do identificador do dispositivo e é feita uma verificação á integridade. Para encriptar os dados ao nível da rede é utilizada a Network Session Key (NwkSKey) que é uma chave de 128 bit Advanced Encryption Standard (AES).

Na camada da aplicação são encriptados e desencriptados os *payloads* das aplicações com a Application Session Key (AppSKey) esta também uma chave de 128 bit AES.

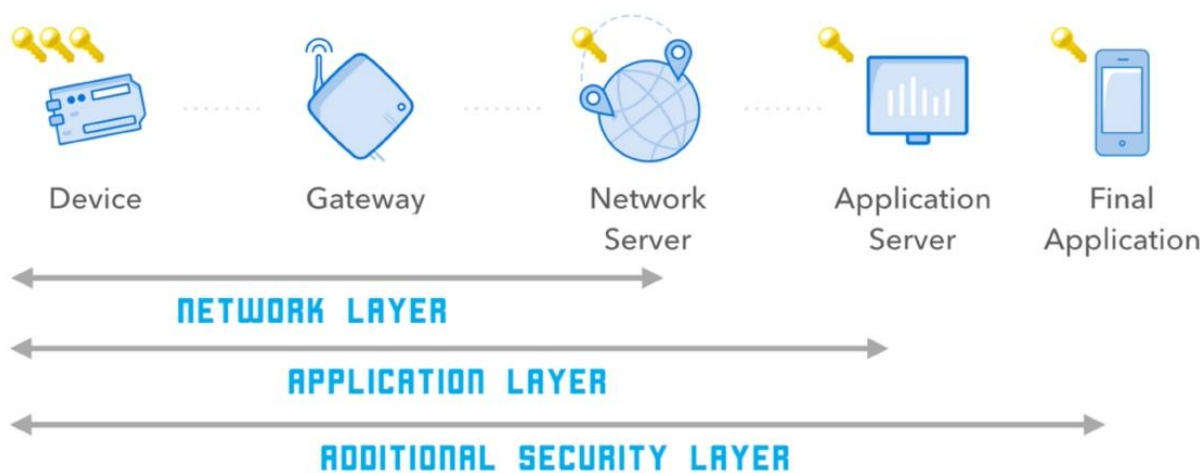


Figura 2. 11 - Segurança em LoRaWAN, retirado de [37].

Os dispositivos em LoRaWAN tem duas formas de serem ativados, Activation By Personalisation (ABP) ou Over The Air Activation (OTAA). No ABP as chaves de sessão e o endereço é programado nos dispositivos. A NwksKey só é conhecida pelo servidor de rede e a chave da aplicação, AppSKey, é comum entre todos os dispositivos que utilizem a mesma aplicação e só é conhecido pelo servidor de aplicação.

O OTAA requer que o dispositivo se junte à rede sendo que a rede é que fornece as chaves para as sessões de segurança e para o endereço do dispositivo. Em adição à NwksKey a AppKey, que é diferente da AppSKEY, é uma chave única em cada dispositivo e é utilizado para derivar as outras chaves. Para se ligar a uma rede o dispositivo envia a identificação do dispositivo, o DevEUI, a identificação da aplicação, o AppEUI e a Appkey e o servidor de rede retorna com os dados necessários para derivar a AppSKEY e a NwksKey.

Uma vantagem da tecnologia LoRaWAN é o facto de a ligação com a nuvem não ser proprietária e fornecer IP ao dispositivo, por outro lado tem a desvantagem de como utiliza IP tem um *overhead* introduzido (20 bytes IP, 20 bytes TCP e 8 bytes UDP) e adiciona a camada de *link layer* e *application*.

Técnicas de Acesso Aleatório

A colisão de tramas é um problema constante em qualquer tipo de rede, seja ela com ou sem fios. Numa comunicação ponto a ponto, tal como as por cabo é mais fácil gerir as colisões, mas nas redes sem fios o acesso ao meio é livre o que leva à dificuldade de controlar e coordenar a troca de informação. Para ambas as situações foram desenvolvidos diversos protocolos que pretendem ajudar, contribuindo cada um deles para melhorar um aspeto da comunicação.

Este capítulo aprofunda as várias técnicas de acesso aleatório ao meio começando pelo ALOHA no subcapítulo 3.1, passando então para as variantes do protocolo CSMA falando do CSMA Collision-Detection (CD) no subcapítulo 3.2 e o CSMA Collision-Avoidance (CA) no subcapítulo 3.3, por fim são faladas nas novas abordagens como a variante Collision Avoidance and Detection (CAD) no subcapítulo 3.4.

3.1. ALOHA

A ALOHAnet foi uma rede de distribuição de pacotes através de rádio e a técnica de multiplexagem utilizada por esta rede foi uma de acesso aleatório. Esta técnica era baseada puramente no acesso aleatório ao meio e ficou conhecida como a técnica de acesso ALOHA puro.[6]

O acesso aleatório ao canal pode ser considerado um recurso que é partilhado entre todos os utilizadores. Cada dispositivo está a competir entre si pela utilização do recetor. Se dois módulos transmitirem ao mesmo tempo ocorre uma colisão e ambos os pacotes são rejeitados. Na ALOHAnet, sempre que cada dispositivo enviava um pacote ele tinha de receber uma mensagem de *acknowledgement*, proveniente da *gateway*, dentro de um certo período de tempo. Se esta confirmação não fosse recebida dentro do intervalo o dispositivo automaticamente iria transmitir o pacote dentro de um intervalo de tempo aleatório de forma a diminuir futuras colisões. Contudo, este número de colisões limita o número de utilizadores e a quantidade de dados que podem ser transferidos na rede. Na Figura 3.1 esta representado o funcionamento do protocolo ALOHA e a razão pela qual não podem chegar dois pacotes simultaneamente a um recetor.

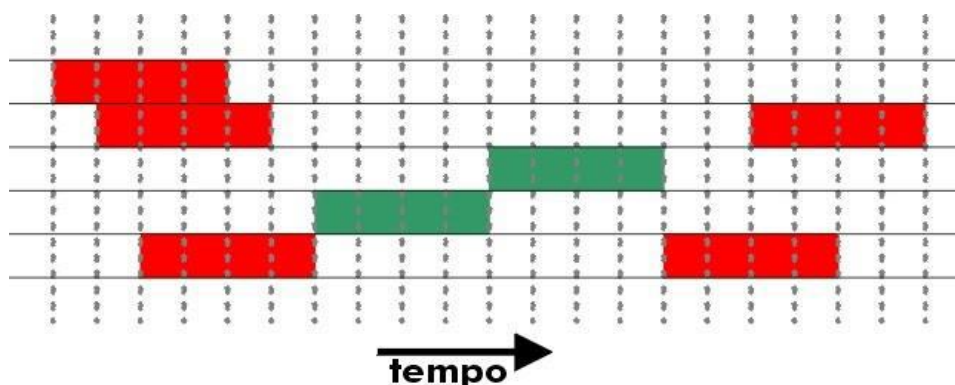


Figura 3. 1 Esquema do Protocolo ALOHA puro, retirado de [5].

Em redes LoRaWAN o esquema utilizado pelos dispositivos de classe A, que demonstra a melhor eficiência energética e é obrigatória a sua implementação é o esquema ALOHA Puro, este é um protocolo simples de acesso aleatório que não tem nenhuma mecânica para evitar colisões. Em particular, o acesso ao canal é controlado pelos dispositivos e depende da sua necessidade de enviar dados. Quando um dispositivo tem dados para enviar eles são enviados, depois da transmissão estar completa o dispositivo espera uma quantidade predeterminada de tempo definida na especificação LoRaWAN antes de abrir a janela de recepção e aceitar os dados transmitidos pela *gateway*, geralmente um *acknowledgement*. Se o pacote *downlink* não for recebido uma segunda janela de recepção é aberta com uma frequência e ritmo binários diferentes definidos ambos na *gateway* e no dispositivo.

Para se poder estudar o ALOHA puro, é necessário prever o seu desempenho.[31] Se todas as tramas tiverem o mesmo tamanho, não for possível gerar novas tramas enquanto se transmite e os dispositivos tentarem transmitir tramas novas e antigas, que tenham colidido, de acordo com uma distribuição de *Poisson*, pode-se prever a capacidade da rede. Sendo que “*T*” refere-se ao tempo de transmissão de uma trama no canal, “*G*” refere-se a media utilizada na distribuição de *Poisson* sobre a quantidade de tentativas de transmissão, ou seja, existem *G* tentativas de transmissão por tempo de trama.

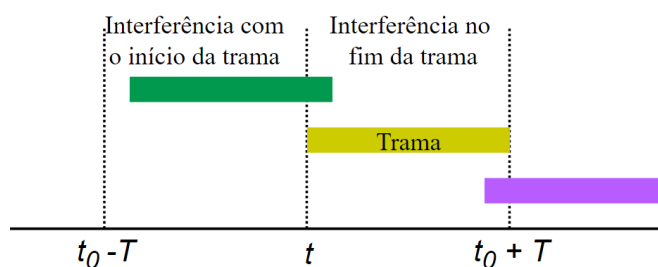


Figura 3. 2 Tramas em ALOHA puro, adaptado de [30].

Considerando o que necessita de acontecer para uma trama conseguir ser transmitida, consideramos “t” como a altura em que se pretende enviar uma trama. Para o envio ter sucesso não se pode ter uma trama a ser enviada no canal entre $t-T$ e t porque uma trama enviada entre esse intervalo iria-se sobrepor a que se vai enviar. Como é possível ver na Figura 3.2.

Para cada tempo de trama, a probabilidade de existirem k tentativas de transmissão é:

$$\frac{G^k e^{-G}}{k!} \quad (3.1)$$

A média de tentativas de transmissão para duas transmissões consecutivas é $2G$. Logo, para cada par de transmissões consecutivas, a probabilidade de k tentativas de transmissão existir é:

$$\frac{(2G)^k e^{-2G}}{k!} \quad (3.2)$$

Logo a probabilidade ($Prob_{pure}$) de não existirem tentativas de emissão entre $t-T$ e $t+T$, ou seja, uma transmissão com sucesso, é:

$$Prob_{pure} = e^{-2G} \quad (3.3)$$

A taxa de transferência da rede (S_{pure}) pode ser calculada como o rácio das tentativas de transmissão multiplicado pela probabilidade de sucesso.

$$S_{pure} = Ge^{-2G} \quad (3.4)$$

A máxima taxa de transferência de uma rede é $0.5/e$ tramas por tempo de trama, que é atingido quando G é igual a 0.5 , que dá aproximadamente 0.184 tramas por tempo de trama. Isto significa que no ALOHA puro só 18.4% do tempo é utilizado para transmissões com sucesso.

Na Figura 3.3 podemos ver o desempenho do ALOHA puro em função da taxa de tentativas de transmissão, realizado no MATLAB [38] através da aplicação da expressão 3.4.

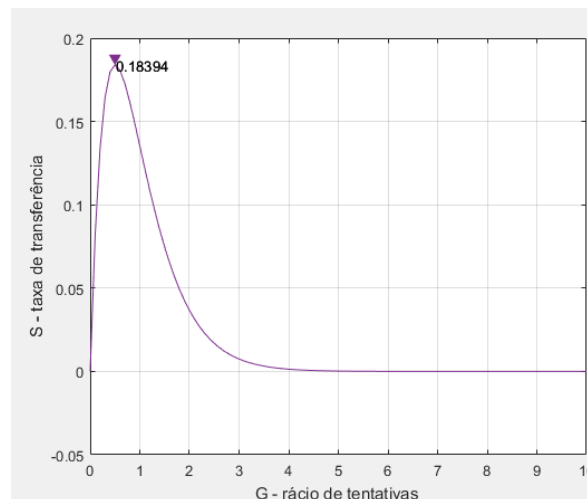
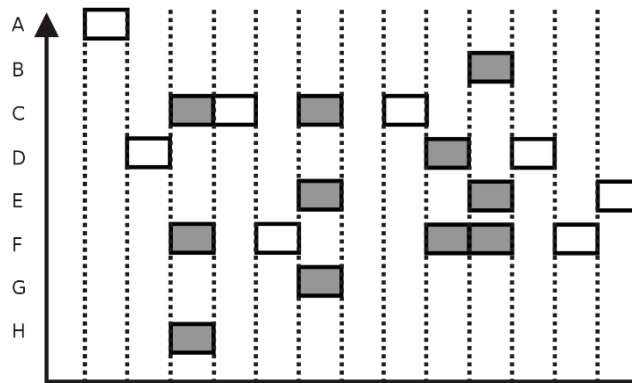


Figura 3. 3 - Desempenho do ALOHA puro.

Uma melhoria do protocolo ALOHA é o *Slotted ALOHA* [39], representado na Figura 3.4, que introduz a divisão de tempo em espaços e cada intervalo corresponde a uma trama, o que aumenta a taxa de transferência. Uma estação só pode começar a transmissão de dados no início de um espaço de tempo, e assim as colisões são diminuídas.

O *Slotted ALOHA* também é um esquema simples, contudo inclui uma sincronização adicional. Ele divide o tempo em *timeslots*, espaços com o mesmo tamanho, e insere-se nas aplicações de baixo tráfego das redes LoRaWAN. Com sincronização apropriada dos relógios entre os dispositivos e as *gateways*, a dimensão do espaço de tempo para cumprir com as regulações de tempo de transmissão, *delay*, e recepção é possível obter uma capacidade de rede



maior ao instruir os dispositivos para começarem a transmitir só no início de cada espaço temporal. [39]

Figura 3. 4 - Slotted ALOHA.

Neste caso, só é necessário preocupar com as tentativas de transmissão dentro do tempo de transmissão de uma trama e não duas consecutivas, sendo que colisões só podem ocorrer durante cada *timeslot*. Sendo G a média de pacotes por tempo de transmissão. Ou seja se um pacote demorar 5 ms a ser transmitido, $G=1$ significa que é transmitido 1 pacote a cada 5 ms. Logo a probabilidade de existirem zero transmissões num *timeslot* é:

$$Prob_{slotted} = e^{-G} \tag{3.5}$$

E a probabilidade de k pacotes é:

$$Prob_{slotted}k = e^{-G}(1 - e^{-G})^{k-1} \tag{3.6}$$

A taxa de transferência da rede ($S_{slotted}$) pode ser:

$$S_{slotted} = Ge^{-G} \quad (3.7)$$

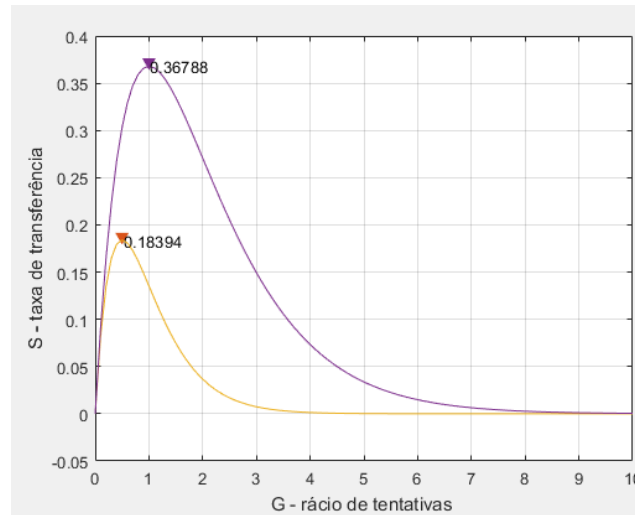


Figura 3. 5 - Slotted ALOHA vs ALOHA puro.

Utilizando o MATLAB conseguimos obter uma representação gráfica da equação 3.4 que representa o desempenho do ALOHA puro e a equação 3.7 que representa o Slotted ALOHA.

Pode-se observar na Figura 3.5, que a taxa de transferência do *slotted* ALOHA é bastante superior à do ALOHA puro, tendo um máximo de $1/e$ tramas por *timeslot*, que é aproximadamente 0.368 tramas por *timeslot*, ou seja 36,8% do tempo é utilizado para transmissões com sucesso.

Na rede LoRaWAN o protocolo utilizado é semelhante ao ALOHA, onde o acesso ao canal é controlado pelos dispositivos e depende da sua necessidade de enviar dados. Caso o dispositivo pretenda enviar dados, estes são enviados e é esperada uma resposta da *gateway*. Se o pacote não for recebido é dado como perdido. No Capítulo 4 é falado mais a fundo da implementação deste comportamento no simulador LoRaSim

3.2. CSMA/CD

O CSMA é um protocolo de acesso aleatório ao meio onde o modulo que quer transmitir informação verifica o meio para a ausência de tráfego antes de transmitir num meio partilhado, tanto por cabo como por uma banda do espectro eletromagnético.

Quando este protocolo se encontra a ser utilizado, o dispositivo que pretende enviar dados, inicialmente irá escutar o meio em que se encontra para tentar determinar se já existe alguma transmissão a ser decorrida. Isto é, tenta detetar a presença de uma portadora de outro dispositivo. Se for sentida alguma portadora no meio, o dispositivo que deseja transmitir aguarda que a transmissão em progresso acabe antes de iniciar a sua. [40]

O protocolo CSMA/CD se tiver a trama já preparada está pronto para transmitir onde verifica se o meio está ocupado e caso esteja espera uma quantidade aleatória de tempo. Quando determinar que o meio está livre começa a transmitir e enquanto transmite escuta para ver se ocorre alguma colisão, se uma colisão for detetada ele procede para o indicado procedimento de colisão. Quando uma colisão é detetada a transmissão continua, mas com um sinal de interferência até todos os recetores detetarem a colisão, incrementa o contador de retransmissões e verifica se já atingiu o número máximo de retransmissões. Se isso ocorrer, aborta a transmissão, se não estiver, calcula o *backoff factor* baseado no número de colisões, quando estiver pronto para transmitir novamente repete o processo de transmissão de novo. Na Figura 3.6 é possível ver o esquema que este protocolo segue.

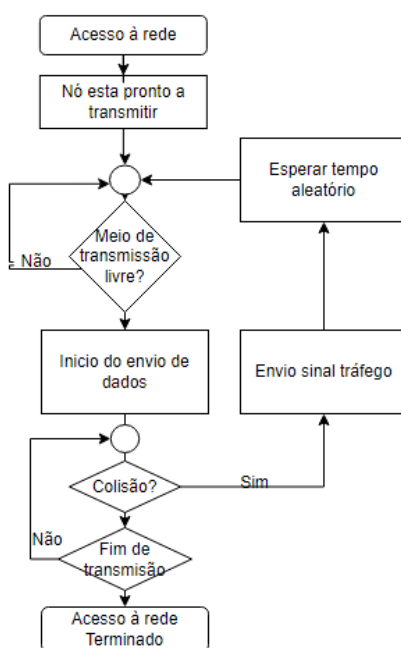


Figura 3. 6 - Esquema do Protocolo CSMA/CD, adaptado de [40].

Devido às características do meio sem fios é difícil a implementação de CSMA/CD que requer ao dispositivo a transmissão e a receção simultâneas. Dispositivos sem fios comunicam unidireccionalmente e durante a transmissão dessensibilizam o recetor, esta característica torna a sua aplicação impraticável em ambientes sem fios.

3.3. CSMA/CA

O CSMA/CA é um protocolo de acesso múltiplo ao meio, que é uma melhoria do tradicional CSMA, tentando dividir o canal igualmente entre todos os transmissores dentro do domínio de colisão. Primeiramente antes de ser realizada qualquer transmissão faz uma deteção de portadoras no meio, de forma a garantir que este se encontra livre ou outro dispositivo encontra-se a meio de uma transmissão. Em seguida, caso seja detetada uma transmissão, aguarda-se um período aleatório, referido como *backoff counter*, para que a mesma seja concluída antes de verificarmos novamente a acessibilidade ao meio, este processo será repetido até o meio estar livre de maneira que o terminal possa começar a sua transmissão.

Desta maneira conseguimos evitar um grande número de colisões, podemos ainda usar o método Request to Send /Clear to Send (RTS/CTS), para aumentar esta segurança. O RTS/CTS é usado para mediar o acesso ao meio e ajuda a aliviar o problema do terminal escondido, uma vez que o Access Point (AP) numa rede wireless só envia um CTS para um terminal de cada vez. No entanto em implementações de rede wireless 802.11 não usam tipicamente RTS/CTS para todas as transmissões, este método pode ser desligado ou apenas utilizado em transmissões grandes, uma vez que o cabeçalho do RTS/CTS é grande demais para ser usado em transmissões de dados pequenas. Se o meio for detetado como livre ou o terminal receber um CTS a indicar explicitamente que pode transmitir então será dado início à transmissão da trama na sua totalidade. Na Figura 3.7 é possível ver o esquema do CSMA/CA

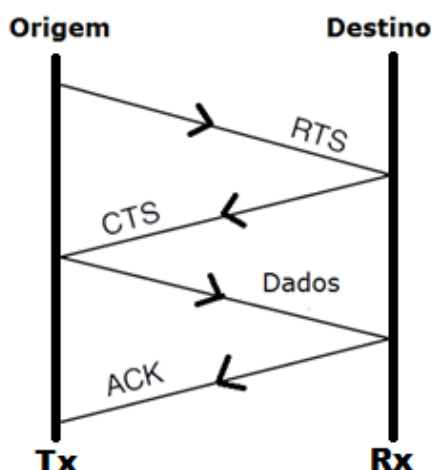


Figura 3. 7 - Esquema do Protocolo CSMA/CA, adaptado de [30].

Ao contrário do CSMA/CD, é bastante difícil para um terminal “ouvir” o meio ao mesmo tempo que transmite (enquanto a transmissão ocorre a sua capacidade de escutar é altamente degradada). Continuando o exemplo das redes wireless, o terminal espera pela recepção de um pacote de *Acknowledge* (ACK) pela *gateway* para indicar que o envio foi recebido com sucesso e o *checksum* estava com o valor correto, caso o mesmo nunca chegue no intervalo de tempo definido pelo ACK Timer então o terminal assume que ocorreu uma colisão de tramas com uma outra transmissão qualquer, e tentará realizar uma retransmissão dos dados.

Apesar do CSMA/CA ser usado numa variedade de situações tem usos particularmente benéficos em redes wireless, devido aos problemas comuns de múltiplas estações serem capazes de verem o dispositivo, mas não de se verem umas às outras. Isto deve-se a diferenças de potências de transmissão e de sensibilidade de recepção, bem como distância e localização ao dispositivo.[40]

Entre as várias variantes do CSMA, a implementada em IEEE 802.11 (WiFi) é a mais próxima dos esquemas de acesso aleatório ao meio utilizados, com o uso de um processo de *backoff*.

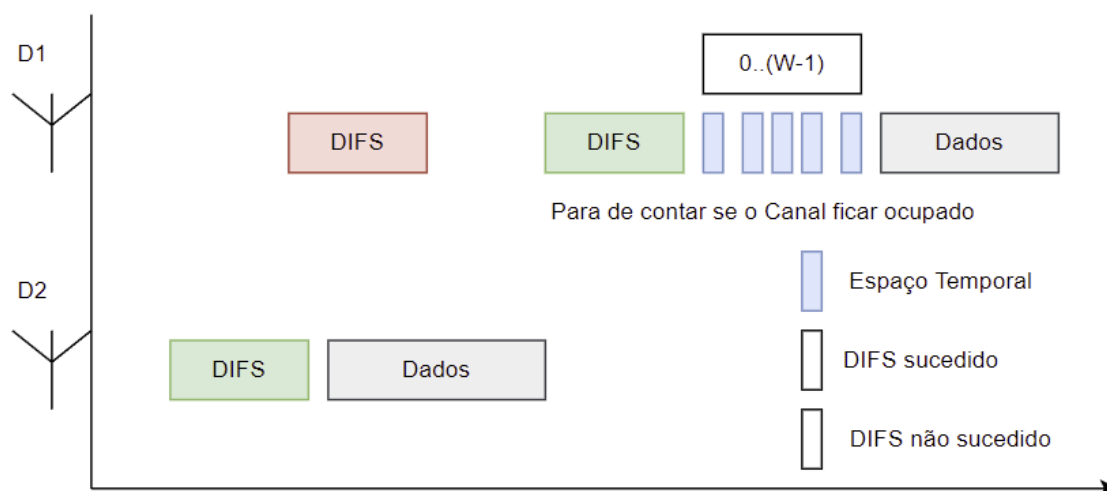


Figura 3. 8 - IEEE 802.11 DCF CSMA/CA, adaptado de [40],

A Figura 3.8 demonstra o mecanismo IEEE 802.11 CSMA utilizado no modo de distribuição coordenada de funções (DCF) que é o modo de operação comum de redes WiFi. Neste modo de operação, o modo opcional RTS/CTS não é utilizado.

O DCF IEEE 802.11CSMA/CA começa por ter o dispositivo a escutar o canal para determinar se existe uma transmissão em curso. Se o meio se encontrar livre para um DCF *inter-frame space* (DIFS) a transmissão é efetuada, se o meio estiver ocupado, o dispositivo guarda a sua transmissão até ao fim da transmissão atual e espera um DIFS adicional antes de gerar um número aleatório de tempo de *backoff* $[0, W-1]$.

O tempo de *backoff* é decrementado enquanto o meio não estiver ocupado, parado quando é detetada uma transmissão e retomado quando o canal é detetado como em espera após outro DIFS. Quando o *backoff* retorna a 0, o pacote é transmitido. No *backoff* o W inicial é definido como 1 e aumentado exponencialmente até chegar a um valor máximo.

O tempo de *backoff* é aplicado depois de uma transmissão terminar porque essa é a altura onde a probabilidade de colisão é maior, devido ao facto de vários utilizadores podem estar a espera que o meio esteja liberto para transmitir.

A norma IEEE 802.15.4 [41] define a operação de *low-rate wireless personal area network* (LR-WPAN) e especifica a camada física e a camada de acesso para estas redes. Esta norma propõe ambos um modo sem *beacon* com *Unslotted CSMA/CA* e com *beacon* e *slotted CSMA/CA*. O último modo com *beacon* necessita de um coordenador e de um alto nível de sincronismo que não é ideal para redes LoRa.

No modo sem *beacon*, o dispositivo, antes de transmitir espera um número aleatório de períodos de *backoff* entre $[0, 2^{BE}-1]$. BE é inicialmente definido como 3. Se ao fim do período

de espera o meio estiver livre a transmissão começa. Se o meio estiver ocupado, a transmissão é parada, o BE é aumentado até um valor máximo definido e espera por um período adicional de $[0, 2^{BE}-1]$. Na Figura 3.9 pode-se ver o modo de funcionamento do protocolo.

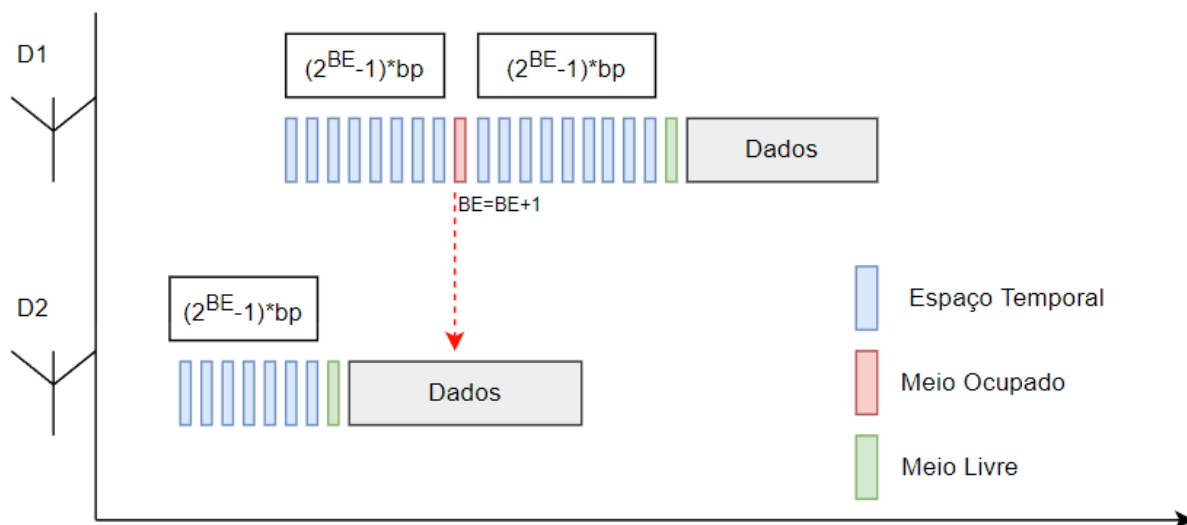


Figura 3. 9 - Funcionamento do standard IEEE 802.15.4, adaptado de [41].

A vantagem que o IEEE 802.15.4 tem sobre o IEEE 802.11 é o facto de em vez de estar continuamente a verificar o estado do canal, espera o tempo de *backoff* o que acaba por consumir menos energia que determinar quando uma transmissão termina.

3.4. Novas Abordagens

Embora o ALOHA e o CSMA sejam técnicas bastante utilizadas no que toca a transmissão são sempre avaliadas novas formas de tornar a comunicação mais fiável.

As *wireless sensor networks* são bastante procuradas por investigadores devido ao seu potencial em áreas como a deteção e seguimento do dispositivo, monitorização ambiental e industrial.

Contudo áreas mais densas em dispositivos afetam a taxa de ocupação da *gateway*, o que torna necessário alcançar métodos de acesso ao meio mais eficientes de forma a otimizar estas redes.

Vários protocolos para a camada MAC são propostos de forma a tentar encontrar o mais adequado. Mas para desenhar um bom protocolo MAC é necessário ter em conta a sua eficiência energética, de forma a perlongar a vida da rede, outro atributo importante é a escalabilidade e a adaptabilidade a mudanças.[10]

Um dos protocolos sugeridos é o Sensor-MAC (S-MAC), que tem por base a sincronização, que é gerida localmente, e períodos de escuta e fala. Os dispositivos adjacentes formam conjuntos que criam um horário de dormida comum, e se dois dispositivos residirem em dois conjuntos virtuais diferentes, eles acordam nos períodos de escuta de ambos os conjuntos. Uma das desvantagens do algoritmo S-MAC é a possibilidade de seguir dois horários diferentes, o que resulta num consumo de energia mais elevado devido ao estar a escuta em ambos os períodos.

As trocas de horários são feitas por pacotes SYNC periódicos trocados entre os vizinhos. O período para cada nó enviar o pacote SYNC é chamado de período de sincronização.

O S-MAC inclui também o conceito de passagem de mensagem, onde mensagens mais longas são divididas em tramas e enviadas em conjunto. Com esta técnica, é possível minimizar a utilização de energia devido à diminuição do cabeçalho à custa de aceder ao meio por um tempo excessivo.

Este protocolo tem como vantagens o facto de a energia gasta por o dispositivo estar a escuta ser reduzida devido ao horário de *sleep*. Em adição à sua simples implementação a interação de sincronização pode ser feita através dos anúncios de *sleep*.

As desvantagens são que os *broadcasts* não utilizam o RTS/CTS, que aumenta a probabilidade de colisão. A escuta adaptativa significa que existe a possibilidade ouvir mais que o necessário ou pacotes que não lhe são destinados. O *sleep* e o ouvir são períodos pré-definidos e constantes o que diminui a eficiência do algoritmo sobre cargas de tráfego variáveis.

Outro protocolo que tem obtido algum estudo é o WiseMAC [9]. O WiseMAC é baseado numa técnica que consiste em regularmente ver o meio de acesso para atividade, escutando o canal rádio por uma duração muito curta. Todos os módulos na rede de sensores escutam o meio com o mesmo período T_w , mas a altura a que o fazem é independente. Se um módulo encontrar o meio a ser utilizado depois de acordar, ele continua a escuta até receber um pacote de dados ou o meio deixar de ser ocupado. O tamanho do preambulo é igual ao período de amostra do módulo. Contudo o recetor pode não estar preparado para o fim do preambulo devido a interferências.

Para reduzir o consumo de potência que é gerado pelo preambulo ter um tamanho fixo o protocolo WiseMAC oferece um método para dinamicamente determinar o tamanho de um preambulo, que utiliza o conhecimento dos tempos de *sleep* dos módulos vizinhos, estes são aprendidos e atualizados pelos módulos durante cada troca de dados como parte da mensagem

de ACK. Baseando-se em tabelas destes tempos de *sleep*, o WiseMAC coordena as transmissões para que o período de escuta corresponda ao meio do preamble do emissor.

Outro parâmetro que afeta a escolha da dimensão do preamble é o fenómeno de desvio do relógio entre a fonte e o destino. Um mínimo período de preamble é calculado tendo em conta o período de escuta mínimo do destino, T_w , e o desvio do relógio que é um múltiplo do tempo de chegada do último pacote ACK. Considerando estas restrições o tamanho mínimo de preamble (T_p) é escolhido aleatoriamente. Podemos ver um exemplo do seu funcionamento na Figura 3.10.

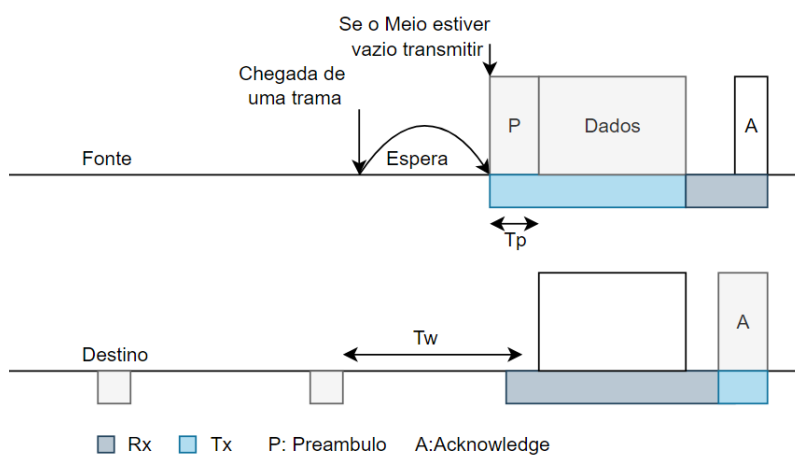


Figura 3. 10 - Conceito WiseMAC, adaptado de [9].

As vantagens deste protocolo são o seu melhor desempenho face o S-MAC, e a capacidade de suportar condições de tráfego variáveis. Devido aos desvios de tempo serem controlados pelos módulos não é necessária intervenção externa para os ajustar.

Antes de aprofundar em que técnicas de CSMA mais se aplicam para ser adaptadas para LoRa é necessário saber como é que um canal LoRa pode ser determinado como ocupado ou disponível para implementar um mecanismo de escuta de canal.

Para a avaliação do canal existe um procedimento de deteção de atividade de canal [42] (CAD ou Channel Activity Detection). Durante um CAD o recetor captura o equivalente a um preamble de dados do canal, com um consumo de 10 mA durante este procedimento. Através de processamento de sinal é procurada uma correlação entre a amostra capturada e a onda do preamble. O tempo para uma deteção de canal está dependente dos parâmetros LoRa utilizados, como por exemplo, o fator de espalhamento e a largura de banda.

Segundo os valores previstos na tabela em [42] e os confirmados pelos colegas em [40] e expressados em múltiplos do período de símbolo. Deste período o radio esta em modo de recção por um *SymbolTime*.

$$SymbolTime = \frac{2^{SF+32}}{BW} [s] \quad (3.8)$$

No artigo [42] o autor propõe a utilização de CADs sucessivos para substituir a função dos DIFS no standard IEEE 802.11 diminuindo assim o consumo energético do dispositivo.

Podendo assim ser aplicado este princípio a outros protocolos de acesso ao meio, permitindo a sua implementação.

| 4.

Trabalho de Simulação

O ambiente de simulação utilizado foi criado para emular as condições de operação da comunicação LoRa, com o nome de LoRaSim, desenvolvido na Universidade de Lancaster [3], por Martin Bor. Embora este simulador tenha uma variedade de resultados com mérito de serem analisados a sua grande vantagem é a facilidade com que pode ser modificado para simular outros cenários

Este capítulo aprofunda sobre o simulador LoRaSim começando por brevemente falar sobre o simulador no subcapítulo 4.1 e aprofundando sobre as expressões utilizadas no cálculo do valor de energia utilizada bem como a perda por caminho no subcapítulo 4.2. No subcapítulo 4.3 é descrito o processo de colisão de tramas e o seu tratamento e por fim no subcapítulo 4.4 são apresentadas as modificações realizadas ao simulador para incluir outros protocolos de acesso

4.1. Ambiente de Simulação

Para conseguirmos simular uma rede LoRa foi necessário encontrar um simulador que conseguisse ir de acordo aos critérios necessários para o estudo que queremos executar.

No artigo [43] é feita uma análise entre os vários simuladores disponíveis com a capacidade de simular uma rede LoRa. Sendo estes o PhySimulator, FloRa, NS3 e o LoRaSim.

O PhySimulator é utilizado para a avaliação ao nível do canal, mostrando que enquanto teoricamente os fatores de espalhamento são ortogonais, a realidade é que a colisão entre fatores de espalhamento é um problema em LoRa.

O simulador FloRa é utilizado para avaliar o desempenho de LoRa utilizando o mecanismo Adaptive Data Rate (ADR), mostrando que o ADR é uma forma de aumentar a probabilidade de receção de forma energeticamente efetiva.

O LoRaSim é utilizado para testar a escalabilidade de redes LoRa LPWAN, tendo a vantagem de ter um site dedicado ao simulador bem como documentação em como utilizar.

Por último, o modulo LoRaWAN para o NS3, foi criado para dar a comunidade uma ferramenta para simular redes reais, em vez de utilizar protocolos de MAC simplificados. Ele

implementa a possibilidade de existirem Ack por parte da gateway, para dar aos utilizadores a habilidade de testar uma rede onde o fator de espalhamento pode ser alterado dependendo da resposta da gateway.

Quanto à linguagem dos simuladores o PhySimulator e o FloRa são escritos em C++ enquanto o NS3 é escrito em C++ e em Python. Já o LoRaSim é escrito inteiramente em Python. No que toca a documentação, de todos os mencionados só o NS3 é que não possui website nem documentação do seu funcionamento.

Tendo em conta esta análise optou-se pela utilização do LoRaSim devido a ser um simulador com base em Python com um site próprio e documentação sobre a sua utilização.

O simulador foi desenvolvido durante a realização do artigo sobre a escalabilidade da rede LoRa [11]. O LoRaSim é um simulador de eventos discretos, implementado com o modulo SimPy [44] e construído na linguagem de programação Python [45].

O simulador pode ser executado tanto em Windows como em Linux desde que este tenha o a versão de Python 2.7 ou 3.8, a versão aconselhada para o uso do programa é a 3.8. Contudo é recomendado que se isole o ambiente em que vamos trabalhar, e para tal criou-se uma máquina virtual com o sistema operativo Ubuntu [46].

No pacote LoRaSim descarregado de [3], estão presentes 5 ficheiros, o primeiro a ser observado é o requirements.txt que contem a lista de dependências para o funcionamento dos programas (estas são matplotlib, simpy e numpy). Os restantes ficheiros, `directionalLoraIntf.py`, que contem a simulação de nós com antenas direcionais e várias redes, `loraDir.py`, que faz uma simulação só com uma estação base (GW), `loraDirMulBS.py`, cria uma simulação com vários nós e até 24 GW, e `oneDirectionalLoraIntf.py` que simula uma GW com antenas direcionais e múltiplas redes. O ficheiro utilizado para realizar os testes e as modificações ao programa foi o `loraDirMulBS.py`

Para executar o programa é necessário passar os parâmetros utilizados, sendo estes, o tempo médio entre o envio de tramas (*AVGSEND*), a experiência a ser realizada (*EXPERIMENT*), que representa os parâmetros da comunicação LoRa, o tempo de simulação (*SIMTIME*), o modo de verificação de colisões(*COLLISION*), utilização de antenas direcionais (*DIR*), numero de redes a simular(*NrNETWORKS*), o número de estações base(*BASESTATIONS*) e o número de dispositivos que vão ser simulados(*NODES*).

```
./loraDirMulBS.py <AVGSEND> <EXPERIMENT> <SIMTIME> <COLLISION> <DIR> <NrNETWORKS>  
<BASESTATIONS> <NODES>
```

Um exemplo do comando utilizado:

python3 loraDirMulBS.py 5700 0 300000 1 1 1 1 10

A simulação de eventos discretos modela a operação de um sistema com uma sequência de eventos discretos no tempo. Cada evento ocorre num determinado instante de tempo e marca uma mudança de estado no sistema. Entre eventos consecutivos, é considerado que o sistema não sofre mudanças, assim, a simulação pode saltar diretamente do instante de ocorrência de um evento para o próximo.

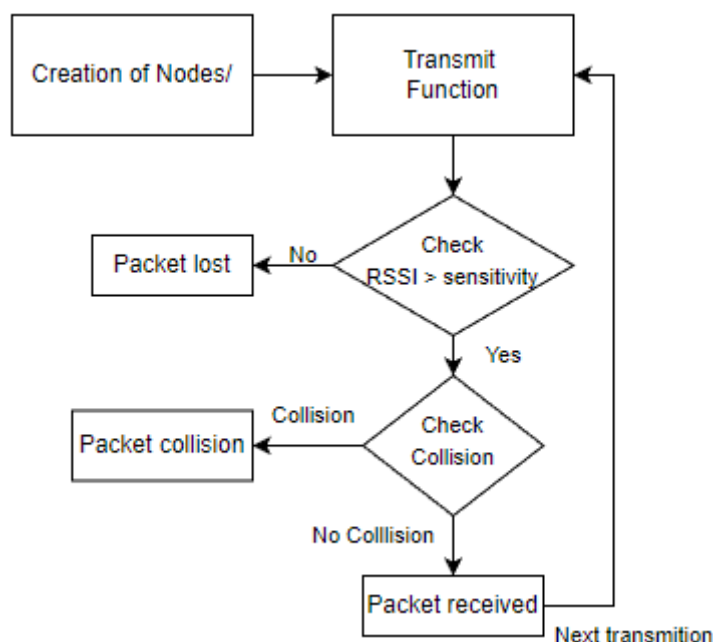


Figura 4. 1 - Simulador LoRaSim

Na Figura 4.1 esta presente a representação feita do funcionamento do LoRaSim.

O simulador começa por recolher os argumentos utilizados na chamada do programa e são introduzidos como variáveis globais, cria cada estação base e a lista dos pacotes recebidos por cada. Começa então a criação dos dispositivos onde primeiro define a posição, calcula a distância a que está de cada estação base e cria um pacote para cada estação utilizando os parâmetros LoRa definidos na experiência. De seguida são adicionados todos os eventos de transmissão através da função *transmit* que cria uma lista global de pacotes a serem transmitidos onde é vista a potência enviada a colisão de tramas e ainda a alteração de posição. Uma vez adicionados todos os eventos ao programa é possível corrido o ambiente de simulação.

4.2. Detalhes do LoRaSim

A tecnologia de rádio LoRa tem cinco parâmetros para configuráveis sendo estes a potência de transmissão (TP), frequência portadora (CF), SF, BW e CR.

No capítulo 2 foram explicados os parâmetros utilizados pela rede LoRa, contudo, a potência de transmissão utilizada na tecnologia radio LoRa pode ser ajustada de -4 dBm a 20 dBm, em incrementos de 1 dB, mas devido a limitações de hardware, o alcance utilizado é limitado de 2 dBm a 20 dBm. Devido a mais limitações de hardware os níveis de potência maiores que 17 dBm têm de ser utilizados em conjunto com um duty-cycle de 1% [11].

Na Europa existem também limitações na distribuição de frequências e a sua utilização para Short Range Devices (SRD) definidos em CEPT/ERC/REC 70-03 [47]. A banda de utilização livre ISM, utilizada por LoRa (863 MHz a 870 MHz) é dividida em sete sub bandas. Cada sub-banda tem um específico requerimento referente a potência máxima irradiada (Effective Radiated Power ou ERP), espectro e espaçamento de canal. Para a maioria das sub bandas, o ERP é de 25 mW (14 dBm).

Para o acesso ao espectro existe a opção de utilizar um duty-cycle (1% ou 0.1%) ou um esquema de ouvir antes de transmitir (Listen Before Talk ou LBT), em conjunto com Adaptive Frequency Agility (AFA) dependendo da sub banda e do ERP exigido [47].

A potência do sinal recebido P_{rx} depende da potência do sinal transmitido P_{tx} e todos os ganhos e perdas da comunicação.

$$P_{rx} = P_{tx} + G_{tx} - L_{tx} - L_{pl} - L_m + G_{rx} - L_{rx} \quad (4.1)$$

P_{rx} é o valor da potência recebida em dB, P_{tx} é o valor da potência transmitida em dB, G_{tx} é o ganho da antena em dB_i, L_{tx} é a perda do transmissor, L_{pl} é a perda de caminho, L_m são outras perdas do sistema em dB, G_{rx} é o ganho da antena de recepção em dB_i e o L_{rx} são as perdas do recetor. A implementação feita no simulador é retirada da equação 4.1.

$$P_{rx} = P_{tx} + GL - L_{pl} \quad (4.2)$$

A equação 4.2 é o cálculo da potência recebida. O GL combina todos os ganhos e perdas enquanto o L_{pl} representa a perda de caminho, determinada pela natureza do ambiente da comunicação.

No lado do transmissor, o alcance só pode ser alterado mudando a potência de transmissão. Outros parâmetros como o SF, BW ou CR não influenciam a potência da transmissão, nem os ganhos ou perdas. No lado do recetor o alcance está limitado pelo limiar de sensibilidade, S_{rx} , que é influenciado pelos parâmetros LoRa, SF e BW.

Existem vários modelos que descrevem as perdas por caminho em dependência dos diferentes ambientes (campo ou urbano). No simulador foi utilizado um modelo de perda *log-distance* [48], implementado pela expressão 4.3, que é comumente utilizado em cenários de zonas com população densa.

$$L_{pl}(d) = \overline{L_{pl}}(d_0) + 10\gamma \log\left(\frac{d}{d_0}\right) + X_\sigma \quad (4.3)$$

Onde $L_{pl}(d)$ é a perda de caminho em dB, $\overline{L_{pl}}(d_0)$ é a média da perda por caminho á distancia de referência d_0 , γ é o expoente da perda por caminho e $X_\sigma \sim N(0, \sigma^2)$, a distribuição normal gaussiana com media de 0, e σ^2 representa o desvio padrão em decibéis.

Nos testes empíricos feitos em [11] com d_0 a 40m, conseguiram determinar que no ambiente que criaram, $\overline{L_{pl}}(d_0)$ tem o valor de 127,41dB, γ é 2.08 e σ tem o valor de 3.35. Como tal estes são os parâmetros que utilizaram no simulador.

A sensibilidade de um recetor rádio á temperatura ambiente pode ser determinada com a equação presente em [37] e é dado pela equação 4.4.

$$S = -174 + 10 \log_{10}(BW) + NF + SNR \quad (4.4)$$

O primeiro valor é devido ao ruído térmico com uma largura de banda de 1 Hz e só pode ser influenciado alterando a temperatura do recetor. No segundo valor, o BW , é a largura de banda do recetor. NF é a figura do ruído e é fixa para cada implementação física. Finalmente, SNR representa a relação sinal ruído requerida pelo esquema de modulação e é determinado pelo SF. Quanto maior o SF maior o SNR.

Os valores obtidos na Tabela 6 provêm de testes feitos para várias combinações de parâmetros LoRa.

Tabela 6 - Sensibilidade de recessão em dBm para diferentes BW e SF, adaptado de [11]

SF	Largura de banda (kHz)		
	125	250	500
7	-126.50	-124.25	-120.75
8	-127.25	-126.75	-124.00
9	-131.75	-128.25	-127.50
10	-132.75	-130.25	-128.75
11	-134.50	-132.75	-128.75
12	-133.25	-132.25	-132.25

Utilizando as equações 4.2, 4.3 e 4.4 é possível estimar se a transmissão LoRa vai ser recebida ou não.

4.3. Tratamento de Colisões no LoRaSim

Quando duas transmissões LoRa se sobrepõem no recetor, existem condições que determinam se o recetor consegue decodificar um, dois ou nenhum pacote. Estas condições são a frequência portadora, o fator de espalhamento, a potência e a altura da receção.

A receção de um pacote LoRa começa a um tempo a e termina a um tempo b . Definimos que a receção de um pacote i começa em a_i e termina em b_i . É possível definir o ponto intermedio de uma transmissão $m_i = \frac{a_i+b_i}{2}$ e o tamanho de metade da transmissão $d_i = \frac{b_i-a_i}{2}$.

Dois pacotes, x e y , sobrepõem-se quando os seus intervalos de receção se sobrepõem.

$$O(x, y) = |m_x - m_y| < d_x + d_y \quad (4.5)$$

Quando duas transmissões se sobrepõem no tempo, mas não utilizam a mesma frequência portadora, elas não interferem entre si e podem ambas ser decodificadas, caso o recetor esteja a escutar em ambas as frequências. A sobreposição na frequência é definida pela diferença absoluta entre a frequência portadora de cada comunicação, e o limiar definido que depende da largura de banda da comunicação. Logo podemos definir a condição de colisão entre duas transmissões, C_{freq} .

$$C_{freq}(x, y) = \begin{cases} 1 & \text{se } |f_x - f_y| < f_{limiar} \\ 0 & \end{cases} \quad (4.6)$$

Onde f_x e f_y são as frequências centrais da transmissão x e y , e o f_{limiar} é o limiar da frequência. A limiar mínimo tolerado para o Semtech SX1272 é de 60 kHz para uma largura de banda de 125 kHz, 120 kHz para uma largura de banda de 250 kHz e 240 kHz para a largura de banda de 500 kHz.

Devido ao fator de espalhamento utilizado em LoRa ser ortogonal, transmissões com diferentes SF podem ser recebidas. Como tal uma transmissão só colide no SF se o utilizado for igual em ambas as transmissões.

$$C_{sf} = \begin{cases} 1 & \text{se } SF_x = SF_y \\ 0 & \end{cases} \quad (4.7)$$

Como LoRa é uma forma de modulação na frequência, sofre do efeito de captura. O efeito de captura ocorre quando dois sinais estão presentes num recetor e o sinal mais fraco é suprimido pelo sinal mais forte. Caso a diferença seja pequena o recetor alterna entre os dois sinais e não consegue decodificar ambas as transmissões.

$$C_{pwr}(x, y) = \begin{cases} 1 & \text{se } (P_x - P_y) < P_{limiar} \\ 0 & \end{cases} \quad (4.8)$$

Onde P_x é a potência do sinal recebido da transmissão x , P_y a potência do sinal recebido y e P_{limiar} é o limiar da potência.

Da experiência realizada em [11], concluíram que existindo 5 símbolos do preâmbulo é possível recuperar o pacote mesmo que se sobreponham temporalmente. A altura crítica na recepção de um pacote começa nos últimos 5 símbolos do preâmbulo, logo o intervalo de transmissão x pode ser definido como $x_{cs} = (a_x + T_{sym} * (N_{pp} - 5), b_x)$, onde T_{sym} é o tempo de símbolo e N_{pp} é o número de símbolos no preâmbulo. Logo um pacote x colide com o pacote y quando se sobrepõem na secção crítica x_{cs} .

$$C_{cs}(x, y) = \begin{cases} 1 & \text{se } O(x_{cs}, y) \\ 0 & \end{cases} \quad (4.9)$$

Quando as condições definidas na equação 4.5, equação 4.6, equação 4.7, equação 4.8 e equação 4.9 forem verdade então o pacote x e y colidem.

$$C(x, y) = O(x, y) \wedge C_{freq}(x, y) \wedge C_{sf}(x, y) \wedge C_{pwr}(x, y) \wedge C_{cs}(x, y) \quad (4.10)$$

A equação 4.10 é o modelo de colisões utilizado no simulador.

O seguinte excerto de código é a implementação presente no simulador da colisão de tramas. Nele existem dois modos de verificação, um simples onde é apenas verificada a colisão na frequência e no fator de espalhamento, e o outro modo verifica ainda se a trama colidiu temporalmente ou na potência. Este último modo simplesmente adiciona a verificação de se são ambas as tramas perdidas ou é possível receber uma.

```
def checkcollision(packet):
    if packet.lost:
        return 0
    ...
    # simple collision
    if frequencyCollision(packet, other.packet[packet.bs]) \
        and sfCollision(packet, other.packet[packet.bs]):
        if full_collision:
            if timingCollision(packet, other.packet[packet.bs]):
                c = powerCollision(packet, other.packet[packet.bs])
            else:
                pass
        else:
            ...
            col = 1
    return col
return 0
```

4.4. Acesso Aleatório

Para poderem ser implementados no simulador os novos modelos de acesso ao meio é necessário primeiro ver o comportamento do LoRaSim.

Começando por implementar o CSMA/CA foi criada uma variável `backoff timer`, `botcounter`, ao processo de transmissão. Quando um dispositivo vai transmitir primeiro verifica se vai ocorrer uma colisão se a trama for enviada no momento previsto. Se for ocorrer uma colisão isto significa que o meio está ocupado, o que leva a um tempo de espera aleatório antes de enviar. Para adicionar tempo ao simulador é utilizada a função `yield` que ao chegar ao fim do tempo `backoff` continua a execução daquele pacote.

```

botcounter=0
for botcounter in range(5):
    node.packet[bs].addTime = env.now
    if (checkcollision(node.packet[bs])==1):
        node.packet[bs].rxtime = node.packet[bs].rxtime+Rx
        rndm_TCol=random.randrange(5, 400, 1)
        node.packet[bs].slp = node.packet[bs].slp + rndm_TCol
        yield env.timeout(rndm_TCol)
    else:
        break

```

Ainda nesta fase foram adicionadas funções para o controlo da energia gasta em cada estado de um dispositivo LoRa, transmissão, recessão e *sleep*.

```

AvgenergyRx = (sum(node.packet[0].rxtime*10*node.sent* V for node in
nodes) /1e6) /len(nodes)
AvgenergyTx = (sum(node.packet[0].rectime *
TX[int(node.packet[0].txpow)+2] * V * node.sent for node in nodes) /1e6
) /len(nodes)
AvgenergySlp = (sum(node.packet[0].slp*0.04 * node.sent *V for node in
nodes) /1e6) /len(nodes)

```

Como as operações executadas pelos dispositivos podem ser classificadas por um estado, o programa guarda todas as interações com a *gateway* na variável `node`, em conjunto com outras informações pertinentes a serem analisadas, tal como o tempo no ar dos pacotes em `rectime`, o tempo que o dispositivo fica a escuta em `rxtime` e a quantidade de pacotes enviados em `sent`.

Tendo em conta as informações do fabricante sobre o consumo dos dispositivos e outros estudos na área em conta, é possível estimar o consumo que um dispositivo tem em cada estado.

Na implementação do CSMA/CAD é feito o envio de um RTS/CTS em que se o canal continuar vazio ao fim das duas operações o pacote é enviado.

```
botcounter=0
for botcounter in range(5):
    if (checkcollision(node.packet[bs])==1):
        ...
    else:
        #espera o tempo de um RTS/CTS
        yield env.timeout(Rx*2)
        node.packet[bs].slp =node.packet[bs].slp + Rx*2
        if (checkcollision(node.packet[bs])==0):
            # se nao colidir tenta enviar o pacote
            node.packet[bs].collided = 0
            break
```

4.4.1. Implementação no LoRaSim

Para testar as adições feitas ao simulador utilizamos um cenário de teste simples com uma *gateway* e dispositivos a variar de 1 a 30. A distância a que os dispositivos estavam da *gateway* depende dos valores utilizados nos parâmetros da comunicação. Nomeadamente a potência de transmissão e o SF. Na Tabela 7 estão descritos os parâmetros utilizados durante a simulação

Tabela 7 - Parâmetros de Simulação

Tensão	3.3 V
Frequência	868 MHz
Code Rate	4/5
Largura de Banda	125 kHz
Potência de Emissão	14 dBm
Payload Length	20 Bytes
Preamble Length	12.5 Symbols
Spreading Factor	SF7/SF9/SF12

Para certificar que os dispositivos se mantinham na mesma posição e os resultados não eram afetados pela sua distribuição aleatória optou-se por implementar no código um padrão em forma de girassol. Embora um cenário pouco realista, o objetivo era a avaliação dos diferentes protocolos MAC, e esta distribuição permite aos dispositivos estarem dispostos mais uniformemente entre *gateways* mantendo-se inalterados entre simulações.

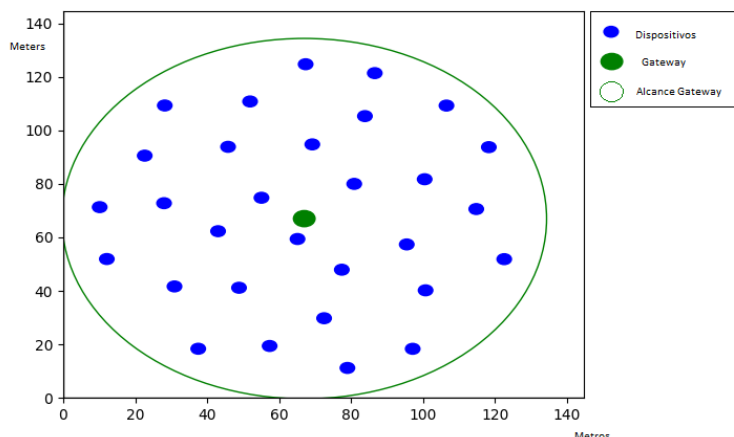


Figura 4. 2 - Distribuição de nós e Gateways

O LoRaSim vem com o protocolo ALOHA implementado e a possibilidade de adicionar outros protocolos de acesso através de programação. Na Tabela 7 estão presentes os parâmetros de simulação corridos. São considerados vários cenários, começando com 1 *gateway*, Figura 4.2, até 4 *gateways*, Figura 4.3, e entre 1 e 500 dispositivos.

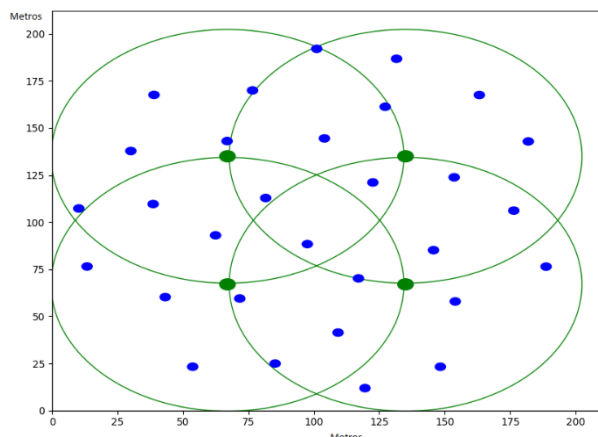


Figura 4. 3 - Distribuição Nós e 4 Gateways

Para avaliar o desempenho de uma rede LoRa, um dos aspetos críticos é a presença da colisão de tramas que ocorre quando dois ou mais dispositivos tentam transmitir dados ao mesmo tempo. Numa primeira fase foram adicionadas variantes do protocolo CSMA.

Começando pela implementação do CSMA/CA, que antes de transmitir verifica o meio para ver se está livre. Se detetar que está a ser utilizado faz o período de *backoff* e volta a verificar o meio no fim desse período onde se estiver livre inicia a transmissão.

Outra variação do protocolo CSMA é o CSMA/CAD e a grande diferença é a adição de tempos de escuta entre a transmissão.

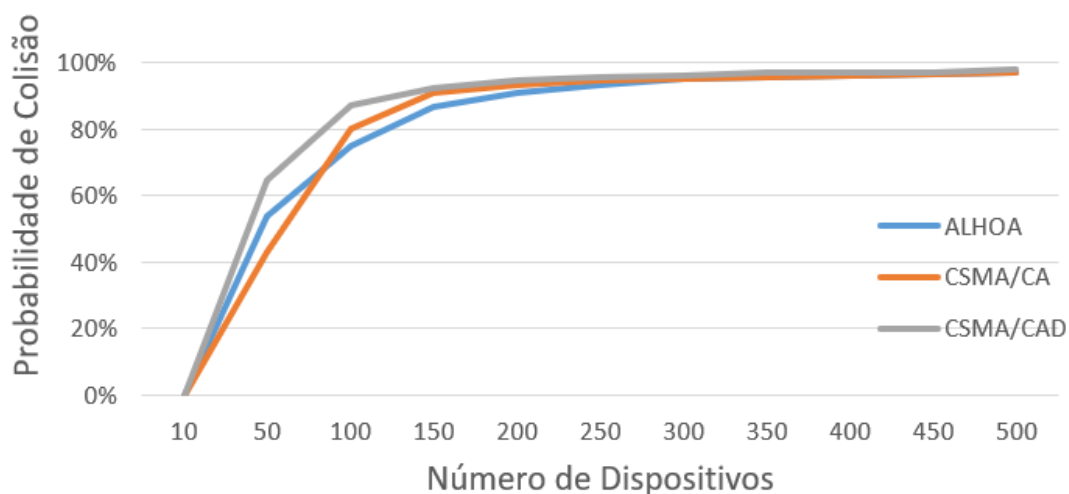


Figura 4. 4 - Colisão com Tx: 2dBm e 1 GW

Foi realizado um teste com uma *gateway* para ver o desempenho dos protocolos quando os dispositivos estão a transmitir com a mínima potência de transmissão de forma a observar o consumo. Na Figura 4.4 podemos ver a probabilidade de colisão de uma trama em função do número de dispositivos na rede.

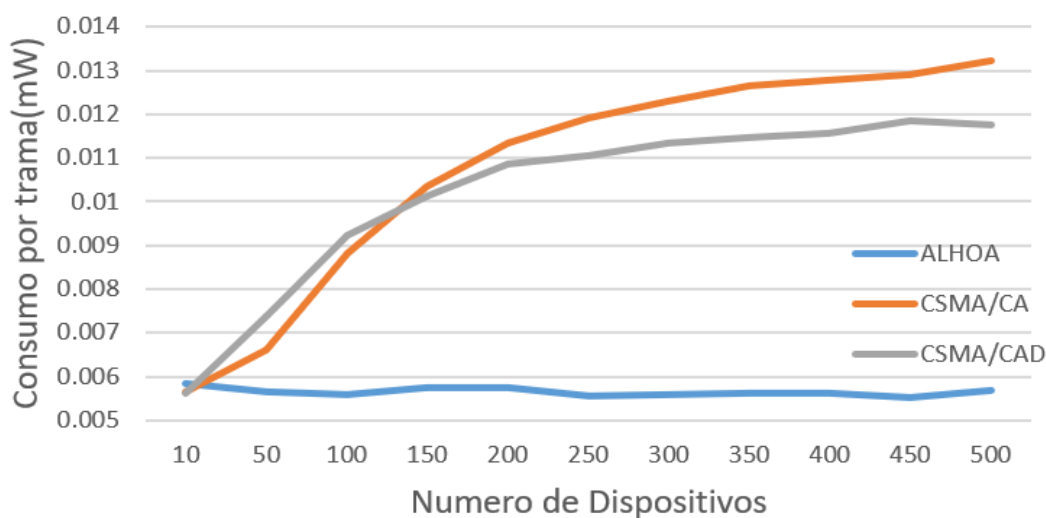


Figura 4. 5 - Consumo de energia por protocolo com Tx: 2 dBm

Na Figura 4.5 podemos ver que o ALOHA mantém o consumo de energia constante independentemente da ocupação do canal. Enquanto protocolos como o CSMA/CA e o CSMA/CAD aumentam o consumo energético por trama para valores de ocupação de canal mais elevados, devido aos dispositivos necessitarem de estar ativos durante mais tempo para verificar o estado do canal e à perda de pacotes.

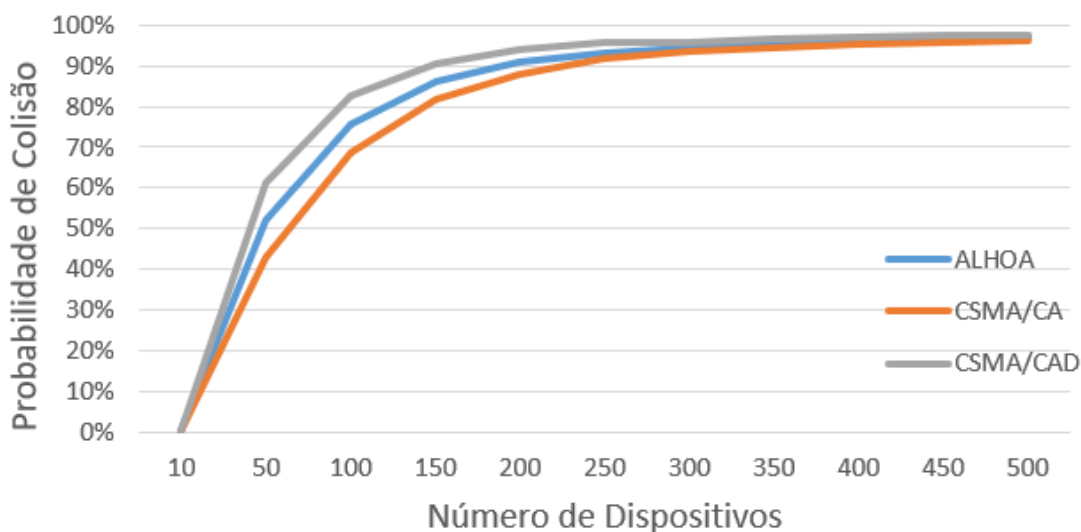


Figura 4. 6 - Probabilidade de Colisão com SF12

Na Figura 4.6 podemos ver o comportamento das implementações de acesso ao meio num ambiente com SF12. Que devido ao tempo de ocupação de um canal por transmissão ser maior a probabilidade de colisão aumenta. Os resultados são semelhantes a utilização do SF7, o que é expectável devido a ocupação de canal ser a mesma percentagem em ambos os casos.

Na Figura 4.7 é possível ver o consumo que o protocolo ALOHA tem dependendo dos parâmetros que utiliza. É possível observar ainda que o aumento do consumo de energia depende de que parâmetro é alterado numa transmissão. O aumento de potência de transmissão não aumenta o consumo tanto como um aumento no SF, sendo SF12 o que consome mais devido ao baixo ritmo binário, o que o faz despendar mais energia para a transmissão.

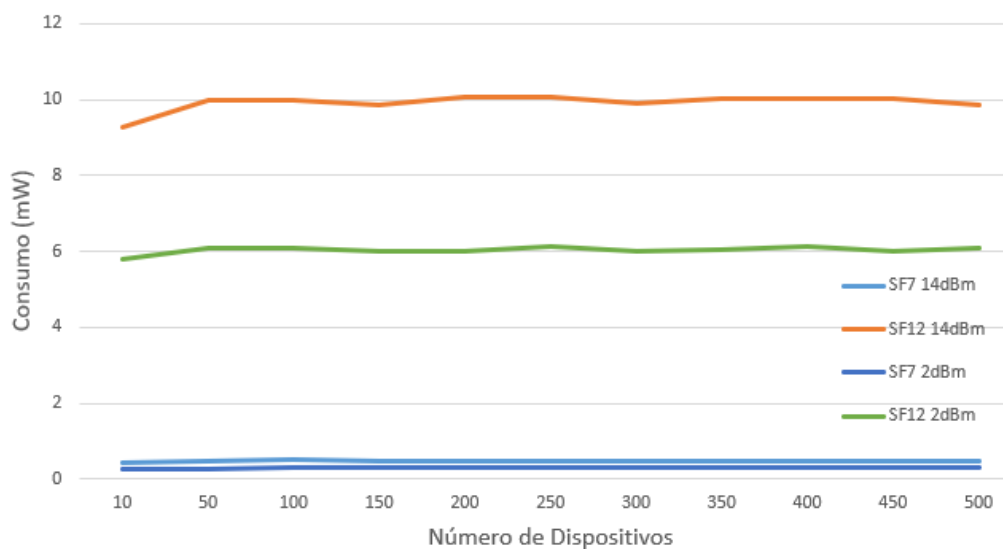


Figura 4. 7 - Consumo energético do protocolo ALOHA

Com este estudo foi possível publicar na conferência INForum 2022 [13]. O título do artigo é: “Development and Analyses of a New Implementation of CSMA/CAD algorithm for LoRaSim Simulator Applied to LoRa Networks” teve como base a adição que foi feita ao simulador LoRaSim de novos protocolos de acesso ao meio. Isto levou a testar a sua eficácia comparando com o desempenho do protocolo ALOHA utilizado atualmente.

|5.

Modelos de Mobilidade

Os modelos de mobilidade caracterizam o movimento de dispositivos móveis no que respeita a sua localização, velocidade e direção ao longo de um período. Muitas das vezes estes simuladores têm um papel vital no teste de redes móveis. Permitem que o utilizador escolha o modelo de mobilidade utilizado visto que este modelo vai representar os movimentos dos dispositivos.

Os modelos de mobilidade apresentados em literatura têm vários níveis de realismo, desde padrões aleatórios, padrões realistas e *datasets* com informações recolhidas de diversos cenários reais. A utilização destes modelos contribui para o teste de protocolos utilizados em redes *ad hoc*.

5.1. Estado da Arte

No artigo [49] os autores procuraram estudar e compreender os efeitos que as alterações da configuração de dispositivos móveis impactam a fiabilidade e o consumo de energia dos dispositivos. E com base nesses fatores desenvolver um algoritmo que permita que o dispositivo adapte as suas definições automaticamente para assegurar um nível de serviço mínimo num cenário com múltiplas *gateways*. Numa parte inicial começaram por definir os parâmetros que mais influenciavam a área de estudo, o fator de espalhamento (SF), que influencia mais diretamente o tempo no ar de uma comunicação, e a potência de transmissão (P_{Tx}), que determina a energia utilizada para enviar um pacote. Para analisar os resultados das medições observam a relação sinal-ruído (SNR), o identificador de receção de sinal (RSSI) e a probabilidade de erro de um pacote (PER). O cenário que utilizaram para testes é um de duas *gateways* separadas por aproximadamente 3 km e colocadas no topo de edifícios. Fazendo uma análise empírica concluíram que um aumento no fator de espalhamento influencia mais o consumo de energia que o aumento da potencia de transmissão, contudo o aumento da potência de transmissão influencia a fiabilidade da mesma em comparação com um aumento do fator de espalhamento. Com estes resultados desenvolveram um algoritmo que otimiza os parâmetros

da emissão, P_{Tx} e SF de forma que o PER seja inferior a 10%. Estes ensaios foram feitos numa zona industrial com espaços abertos, um outro teste que se propõe seria a realização destes ensaios numa zona urbana, fora de linha de vista de forma a estudar a alteração do SF e da potência de transmissão consoante a presença de edifícios ou espaços abertos e fechados. Outro teste pertinente seria ver o efeito que a presença de vários dispositivos numa só *gateway* tem sobre a escolha dos parâmetros.

No artigo [50] é feito um estudo sobre a mobilidade em redes celulares 5G considerando o posicionamento aleatório de estações base e um posicionamento em forma de padrão. Para fazer a análise primeiro começam por identificar os fatores que são influenciados pela mobilidade como o handoff, Sojourn time (*dwell time*), Direction switch time, que representa o tempo a ir de uma localização para outra e o tempo em que fica parado e ainda é visto o parâmetro handover probability. Tendo em conta os parâmetros a monitorizar são vistos diversos modelos de mobilidade que podem potencialmente imitar o padrão de movimentos de um utilizador ou dispositivo. São estudados modelos puramente aleatórios (random walk, random waypoint), relacionados espacialmente (pursue mobility, column mobility) e relacionados temporalmente (Gauss-Markov, Levy flight). O primeiro teste que fizeram foi utilizando um algoritmo que imite o movimento humano e que o handoff ocorre quando este sai do alcance da célula ao longo do seu trajeto, que pode ser definido de acordo com o padrão de mobilidade escolhido. Esta abordagem fornece o Sojourn time e o handoff rate. A outra abordagem feita assume que o evento de handoff ocorre quando existe uma estação base vizinha que lhe fornece um sinal mais forte. Na segunda abordagem é possível extrair a probabilidade de handoff durante os períodos de movimento. Os autores chegaram a conclusão de que ambas as abordagens eram válidas para a modelação da mobilidade nas redes celulares, na primeira abordagem poderiam ser utilizados outros modelos para a mobilidade do utilizador, contudo ficaria mais difícil a avaliação dos dados, nomeadamente as bordas e o trajeto do utilizador, se a localização das estações base na rede for mais densa.

Esta abordagem a mobilidade é interessante no aspeto que aborda vários modelos de mobilidade, contudo no estudo da rede LoRa os parâmetros analisados em [50] não são adequando uma vez não existir o conceito de handover nas redes LoRa. Contudo, na nossa análise da mobilidade parâmetros importantes da tecnologia LoRa: o SF, o consumo energético do dispositivo (considerando as técnicas de acesso aleatório implementadas) e transmissões bem-sucedidas.

Na abordagem tomada pelos colegas em [51], é possível ver outro estudo do desempenho energético baseado em simulações. Neste artigo são abordados dois problemas, o primeiro é a criação de um módulo que consiga representar o comportamento de uma rede LoRa, e para validar os resultados do módulo, são utilizados dados retirados de testes práticos e de outros trabalhos para suportar o módulo criado. O segundo problema era a melhoria dos dispositivos LoRa sem impactar o consumo de energia. Eles utilizam o simulador para avaliar a implementação do CSMA em LoRaWAN de forma a reduzir a probabilidade de colisão.

Para poderem basear os resultados que obtêm no simulador, eles utilizam a implementação do consumo de energia feito em [52] para poderem estimar o consumo de um dispositivo ou da rede inteira. Todas as operações dos dispositivos são representadas como estados com o seu respectivo consumo de energia. No caso utilizado, como era radio, foram assumidos três estados definidos como transmissão, recessão e inativo. A energia consumida total é a soma de todos estes estados.

Nos testes feitos são testados dois tipos de utilização de CSMA, a primeira é a mais simples, o dispositivo fica a escuta no canal e se estiver livre envia a trama, se não estiver espera um tempo aleatório de segundos e volta a tentar. A segunda implementação envolve o dispositivo ficar a escuta no canal durante um intervalo de 10 milissegundos antes da transmissão. Nestes testes o consumo dos dispositivos é retirado do datasheet da Semtech [53] e o consumo de transmissão, recessão e inatividade é correspondente a 419.6 mW, 44.06 mW e 4.32 μ W respectivamente.

O estudo acabou por comprovar que a utilização de um protocolo como o CSMA reduz consideravelmente a probabilidade de colisão embora só aumente o consumo ligeiramente para um número grande de dispositivos.

O estudo realizado neste artigo é de alguma semelhança ao que se pretende embora não tenha em atenção a relação entre os parâmetros da transmissão LoRa com o consumo de energia pois como foi utilizado um SF variável e um code rate fixo não foi possível ver a sua influência nem na colisão nem no consumo de energia. Na nossa análise pretendemos abordar essas questões e ver a influência que isso tem na mobilidade dos dispositivos.

5.2. Metodologias Analíticas

Para conduzir uma análise sobre o impacto que a mobilidade de dispositivos LoRa tem na rede, é necessário avaliar os valores de desempenho medíveis tal como a ocupação da rede e o alcance. Neste contexto o número máximo de dispositivos numa rede, a relação sinal ruído e a taxa de colisão são valores de interesse para avaliar o desempenho das redes.

Para podermos avaliar a mobilidade observamos primeiro, modelos que imitem os padrões de movimento de uma pessoa ou dispositivo. Estes modelos incluem modelos puramente aleatórios (random walk, e random way point, random direction), relacionados espacialmente (pursue mobility, column mobility) e correlacionados temporalmente (Gauss-Markov, Levy flight).

A precisão destes modelos de mobilidade pode ser medida na proximidade que eles apresentam a padrões realistas de movimento. Contudo obter esta precisão pode ter um grande custo computacional ou complexidade matemática.

Estes modelos são modelos aleatórios sintéticos, ou seja, modelos matemáticos que caracterizam o movimento dos dispositivos. Os modelos são geralmente simples e fáceis de seguir analiticamente, mas podem não refletir padrões de mobilidade realísticos [54]. Exemplos incluem os modelos Bayesianos [55] capazes de imitar o comportamento aleatório dos dispositivos, movimentos Brownianos [56] que caracteriza o movimento de partículas num intervalo de tempo e padrões de Levy [57] que é mais difusivo que os movimentos Brownianos e é uma boa aproximação do andar humano em ambientes ao ar livre.

Outros modelos de mobilidade chamados de “trace-based” são obtidos através de medições de sistemas (*logs* de ligações ou informação da localização de utilizadores móveis). Estes modelos são realísticos no movimento e na topologia da área, contudo existe um limite no número de casos devido ao tamanho da amostra. Um exemplo de dados deste tipo são *datasets*, ou o projeto CRAWDAD [58].

Os modelos sintéticos podem ainda ser classificados como mobilidade individual sem memória (sem correlação temporal ou espacial) e com memória (correlação temporal) ou mobilidade de grupo (correlação espacial).

Em modelos individuais sem memória o movimento de um utilizador não é independente de outros utilizadores, ou seja, a localização, velocidade e a direção do movimento não são influenciadas por outros utilizadores nas proximidades nem em função da posição anterior nem velocidade. Este modelo embora fácil de seguir matematicamente tem o problema de não considerar situações como a posição do dispositivo à sua frente. Este modelo também é

vulnerável a paragens bruscas, aceleração instantânea e curvas acentuadas. Nos movimentos de veículos e pedestres a aceleração tende a ser incremental em vez de aleatória e a mudança de direção também suave levando á existência de uma correlação temporal entre os parâmetros de mobilidade.

Nos modelos individuais com memória ou dependência temporal, o utilizador continua a mover independente de outros dispositivos, mas o seu movimento é em função da sua posição anterior e velocidade.

Os modelos de mobilidade com dependência espacial ou de grupo, podem utilizar funções matemáticas para descrever a mobilidade de um grupo como o modelo de mobilidade de coluna [59] ou o modelo de comunidade. Outro tipo de modelos de mobilidade de grupo descreve o comportamento de um dispositivo individual e atribui-lhe a função de líder fazendo os outros dispositivos imitar o seu caminho, um exemplo deste conceito é modelo de mobilidade de grupo com ponto de referência.

5.2.1. Random Walk

Uns dos modelos de mobilidade mais utilizados por investigadores são o Random Walk e o Random Waypoint. Sendo o Random Walk um dos modelos de mobilidade escolhido.

Este modelo foi primeiro descrito matematicamente por Einstein em 1926 [60]. Como várias entidades na natureza movimentam-se de forma errática, o modelo Random Walk foi desenvolvido para imitar este movimento imprevisível. Especialmente neste modelo, um utilizador move-se da sua localização atual ao escolher aleatoriamente uma direção e uma velocidade para viajar e cada movimento neste modelo é executado num constante intervalo de tempo no fim do qual volta a calcular a direção e a velocidade.

Existem vários modelos derivados do Random Walk, incluindo o de uma dimensão, duas dimensões, três dimensões e d dimensões.

O modelo de mobilidade de uma dimensão é o equivalente a andar numa linha onde só é possível andar para um dos lados e no fim desse movimento é novamente escolhido aleatoriamente o lado para que se anda.

O modelo 2-D Random Walk é de especial interesse, uma vez que a superfície terrestre é modelada utilizando uma representação de duas dimensões. O dispositivo móvel começa o movimento no centro de uma área. A cada ponto, é escolhido uma direção entre 0 e 2π e uma velocidade que determina a distância a percorrer. No modelo de mobilidade Random Walk um nó pode alterar a sua direção depois de viajar uma certa distância ou após uma certa quantidade de tempo. Na Figura 5.1 esta presente um exemplo deste modelo onde o dispositivo anda numa direção com uma certa velocidade.



Figura 5. 1 - Modelo Random Walk

Na Figura 5.2 está presente outra forma de implementação do modelo Random Walk onde o nó anda nas quatro direções cardinais uma distância fixa.

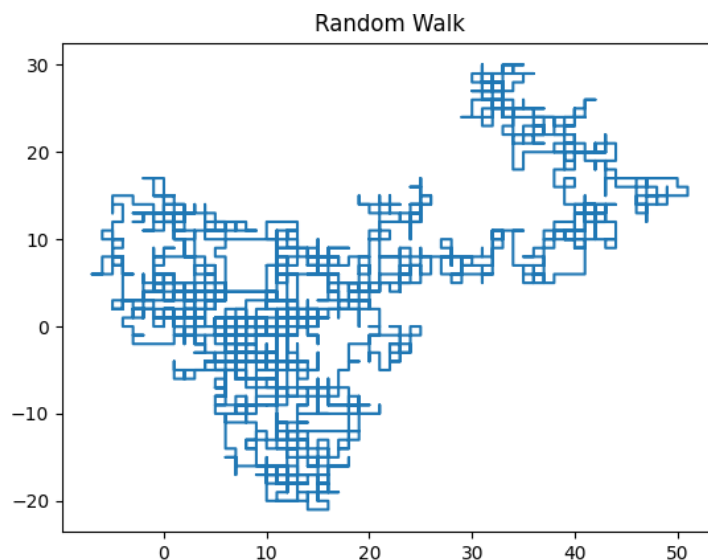


Figura 5. 2 – Modelo Random Walk Bidirecional

O modelo de mobilidade Random Walk é considerado um modelo sem memória porque não retém conhecimento sobre as suas localizações passadas e valores de velocidade. A velocidade e direção atual de um nó é independente da sua velocidade e direção passada. Esta característica pode gerar movimentos irrealistas com travagens súbitas e mudanças de direção abruptas. Contudo a sua relevância no estudo da mobilidade ainda se mantém devido a sua baixa complexidade e fácil implementação.

5.2.2. Gauss-Markov

O modelo de mobilidade Gauss-Markov foi originalmente proposto para a simulação de uma rede de serviços de comunicação pessoal (PCS) [61], contudo este modelo já foi utilizado para simulações de um protocolo de rede ad-hoc [62].

O modelo de mobilidade Gauss-Markov foi desenhado para se adaptar a diferentes níveis de aleatoriedade, via um parâmetro de viragem [59]. Inicialmente é atribuído uma velocidade e direção a cada nó. A intervalo de tempo fixo, n , o movimento ocorre ao atualizar a velocidade e direção de cada nó. Especialmente, o valor da velocidade e a direção na instância n_x é calculado com base nos valores de velocidade, s , e direção, d , na instância $(n-1)$ e uma variável aleatória seguindo a equação 5.1 e 5.2.

$$s_n = \alpha s_{n-1} + (1 - \alpha)\bar{s} + \sqrt{(1 - \alpha^2)}s_{x_{n-1}} \quad (5.1)$$

$$d_n = \alpha d_{n-1} + (1 - \alpha)\bar{d} + \sqrt{(1 - \alpha^2)}d_{x_{n-1}} \quad (5.2)$$

Onde s_n e d_n são os novos valores de velocidade e direção do nó no intervalo de tempo n , e α (α), onde $0 \leq \alpha \leq 1$, é o parâmetro de viragem utilizado para variar a aleatoriedade, \bar{s} e \bar{d} são constantes representando a media da velocidade e direção, $s_{x_{n-1}}$ e $d_{x_{n-1}}$ são variáveis aleatórias com uma distribuição Gaussiana. Valores aleatórios, semelhantes a um movimento Browniano como o Random Walk, são obtidos com $\alpha = 0$ e movimentos lineares são obtidos com $\alpha = 1$. Valores intermédios de aleatoriedade são obtidos ao variar α entre 0 e 1.

A cada intervalo de tempo a próxima localização é calculada com base na corrente posição, velocidade, e direção de movimento. Especificamente, a um determinado intervalo n , a posição de um nó pode ser dada pelas equações 5.3 e 5.4.

$$x_n = x_{n-1} + s_{n-1} \cos d_{n-1} \quad (5.3)$$

$$y_n = y_{n-1} + s_{n-1} \sin d_{n-1} \quad (5.4)$$

Onde (x_n, y_n) e (x_{n-1}, y_{n-1}) são as coordenadas da posição do nó no intervalo n e $n-1$, respetivamente, s_{n-1} e d_{n-1} são respetivamente a velocidade e a direção do nó no intervalo de tempo $n-1$.

Para certificar que o nó não fica perto da borda da janela de operação durante muito tempo eles são forçados a sair quando chegam a uma certa distancia da mesma.

Isto é feito modificando a variável \bar{d} na equação 5.2.

A Figura 5.3 ilustra um exemplo do caminho tomado por um nó utilizando o modelo de mobilidade Gauss-Markov.

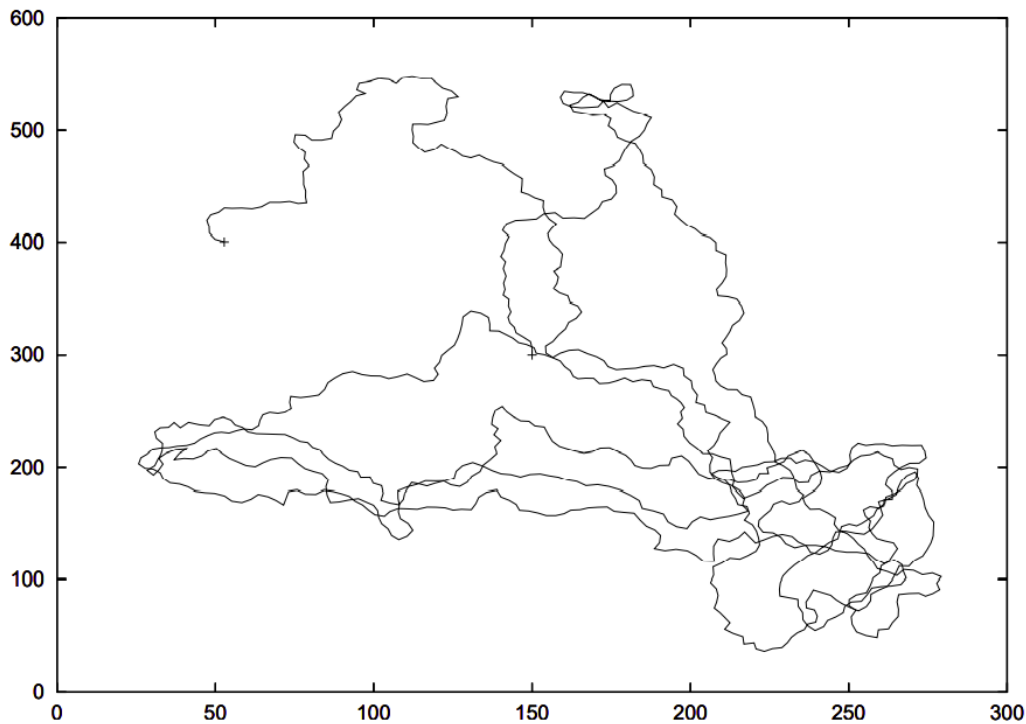


Figura 5. 3 - Padrão de movimento de um nó utilizando o modelo de mobilidade Gauss-Markov

5.3. Datasets

Nos últimos tempos tem havido um grande aumento na quantidade de *datasets* existentes sobre o movimento de mobilidade humana com grande detalhe e número de sujeitos. Exemplos destes tipos de dados são posições geográficas retiradas do GPS do dispositivo ou veículos. Grande parte destes *datasets* é proprietária e não disponível ao público. Contudo existem projetos como o CRAWDAD [58] que fornece informações retiradas de redes sem fios, o MoveBank [63] é um repositório de dados apoiado pela Max Planck Society e contém uma

variedade de dados sobre a posição de animais, foi possível arranjar ainda um *dataset* com o percurso de vários taxie [64][65].

Um *dataset* é uma coleção de dados, relacionados, que podem ser acedidos individualmente ou como um inteiro. A vantagem da utilização de data sets em oposição a modelos matemáticos é o facto de data sets serem uma coleção de dados reais o que credibiliza a sua utilização. Podem ainda ser organizados em estruturas de dados, no caso da mobilidade os fatores mais importantes a ter nos dados é a posição, coordenadas x e y, e um identificador único do fornecedor dos dados.

Através da plataforma MoveBank foi possível obter os dados de mobilidade de 25 alces [66] [67] durante 2 anos sendo retirada a posição geográfica a cada hora tendo sido obtidos 133 519 resultados. Na Figura 5.4 é possível ver as posições obtidas através do estudo.

Utilizando um dataset com representações reais de dados é possível observar o comportamento do simulador em função do padrão de movimento destes animais.

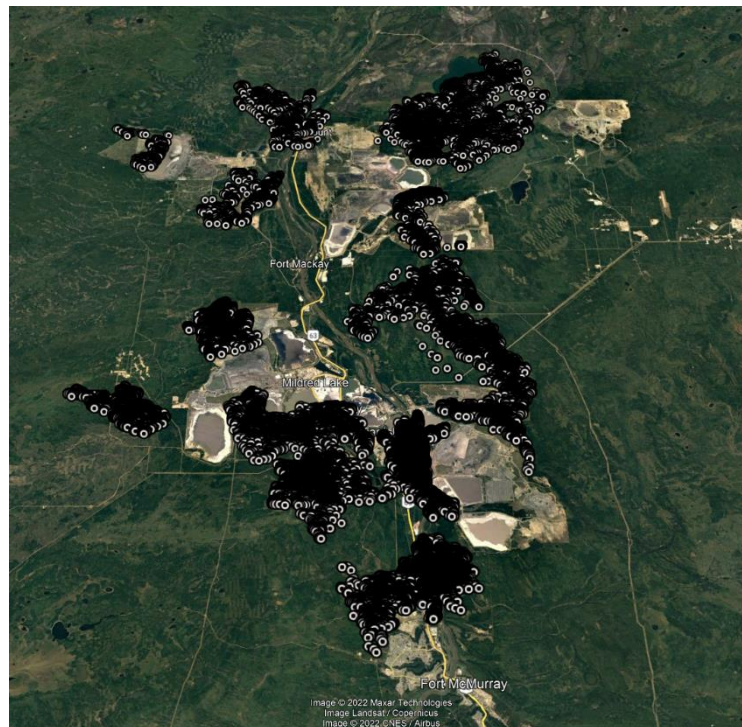


Figura 5. 4 - Área geográfica de um Dataset de Alces do Canada

5.3.1. Integração com o LoRaSim

Para integrar a aquisição destes dados com o simulador LoRaSim é necessário processar os dados de forma a obter a posição, o identificador e o tempo de captura.

Para tratar dos dados no formato CSV recorreu-se a biblioteca pandas [68] que disponibiliza um conjunto de funções que auxilia o tratamento destas grandes quantidades de dados, transformando os em *dataframes* que possuem várias funções que podem ser aplicadas sobre os dados.

Depois de extrair a informação do ficheiro CSV é necessário a tratar. Para tal foram alteradas algumas variáveis e retirar colunas desnecessárias para o nosso estudo como a temperatura, a altura do mar entre outros.

Existe a necessidade de alterar 3 valores para estes dados poderem ser utilizados. O registo no tempo, de forma que com os dados presentes possamos ter uma simulação onde a transmissão dos dispositivos seja de 3 em 3 minutos. A longitude e a latitude precisam de ser alteradas de forma a ser possível utilizar os dados de mobilidade e aplicar na zona que estamos a observar.

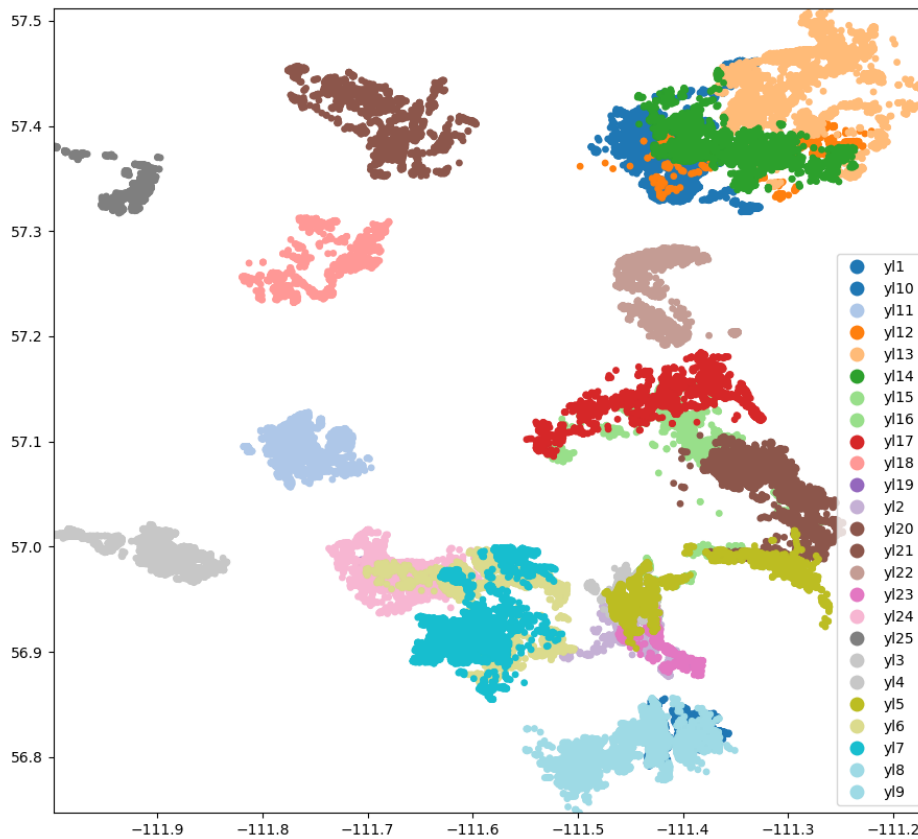


Figura 5. 5 - Dados representados em python

Na Figura 5.5 é possível ver as várias zonas de ação de cada animal já processados pela biblioteca Pandas.

Para os dados poderem ser concentrados na área que queremos observar, é necessário definir a zona em que cada identificador atua, de forma a ser determinada a alteração necessária a cada ponto para mover o grupo.

```
i = True
for id in uniqueDF:
    #retrieve the smaller long and lat
    if i:
        i=False
        minlong, minlat = df.loc[df['lid']==id, ['long', 'lat']].min()
    else:
        minLongId, minLatId = df.loc[df['lid'] == id, ['long', 'lat']].min()
        #vector by which the groups are centered
        vlong = minlong - minLongId
        vlat = minlat - minLatId
        for index, row in df.loc[df['lid']==id].iterrows():
            df.loc[index, ['long']] = df.loc[index, ['long']] + vlong
            df.loc[index, ['lat']] = df.loc[index, ['lat']] + vlat
    t=0
    for index, row in df.loc[df['lid']==id].iterrows():

        df.loc[index, ['timestamp']] = t
        #value in minutes
        t=t+3
```

Numa parte inicial obtém-se qual o valor mínimo que o primeiro identificador tem, de forma a ser obtido o ponto de referência sobre o qual todos os outros pontos se vão mexer. Para esta operação é utilizada a função `.loc` do objeto `dataframe` como é possível ver no excerto de código acima. Nas seguintes iterações é retirado o valor mínimo de cada identificador e com base no mínimo dos primeiros dados é criado um vetor que é aplicado ao grupo de pontos com o mesmo identificador desta forma é possível agrupar os movimentos individuais numa área que queiramos estudar, como é possível ver na Figura 5.6.

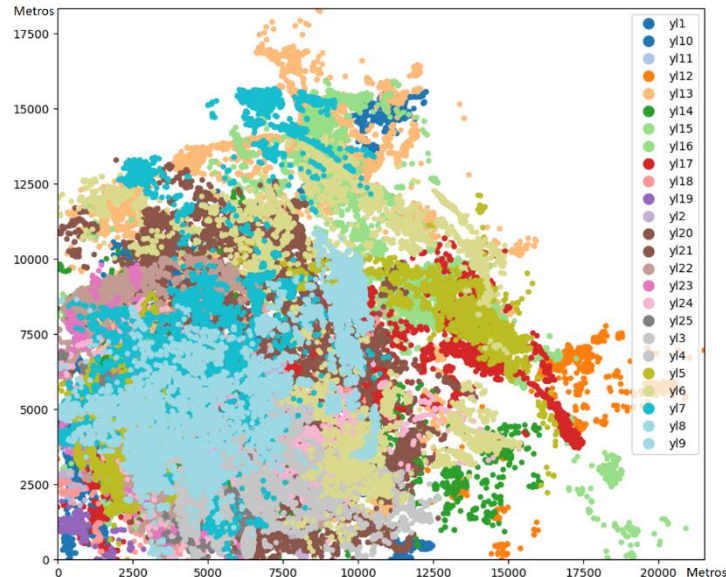


Figura 5. 6 - Aglomeração de dados do dataset numa area

Para se poder integrar com o simulador é necessário modificar as coordenadas GPS para lineares. Para isto poder ser feito a fórmula de *Haversine* [69] que permite saber a distância entre dois pontos de uma esfera a partir da sua longitude, λ , e latitude, φ .

$$d = 2R \arcsin \left(\sqrt{\sin^2 \left(\frac{\varphi_2 - \varphi_1}{2} \right) + \cos(\varphi_1) \cos(\varphi_2) \sin^2 \left(\frac{\lambda_2 - \lambda_1}{2} \right)} \right) \quad (5.1)$$

Usando esta função para cada coordenada é possível obter o ponto num plano xy e com isso obter a velocidade entre cada posição. De forma a não ser necessário fazer o processamento dos dados novamente são exportados os resultados para um ficheiro para ser carregado caso não se altere a informação.

A utilização de dados de *datasets* tornou-se mais difícil que esperado. Uma vez que a recolha de informação é de hora a hora e a informação abrange uma área de dimensão elevada. Os dados acabaram por não se enquadrar na utilização do simulador LoRaSim que foi configurado utilizando os valores obtidos através de experiências em meio urbano.

| 6. Modelo Desenvolvido

Neste capítulo são abordados os modelos de mobilidade desenvolvidos no simulador e a sua implementação em código bem como a sua integração com as diferentes técnicas de acesso ao meio.

De forma a conseguir estudar a influência que o protocolo de acesso ao meio tem sobre a mobilidade de um dispositivo é necessário simular uma rede LoRA. Para este efeito utilizamos o simulador anteriormente referido, LoRaSim fazendo alterações ao seu código para acomodar os protocolos e modelos desenvolvidos.

Antes de criar os módulos foi necessário descobrir em que ponto é possível alterar a posição do dispositivo, para tal foi criada uma função que chama o modelo de mobilidade, atualiza a posição e adiciona a posição anterior a uma lista, que mais tarde serve para ver o percurso que tomou.

6.1. Implementação de Modelos

Devido à popularidade do modelo de mobilidade Random Walk e à sua baixa complexidade este modelo foi o primeiro a ser implementado no simulador.

A nova posição dos dispositivos só é calculada quando o período de tempo entre transmissões respeita o *duty-cycle*. Para calcular a distância percorrida em ambos os modelos é utilizada a velocidade média, tendo em conta que um ser humano anda à velocidade média de 1.4 metros por segundo, no modelo de mobilidade Gauss-Markov essa é a velocidade inicial e no modelo Random Walk é calculada a distância percorrida entre transmissões utilizando a média de velocidade.

Foram implementadas as duas versões do modelo de mobilidade anteriormente descritas. Primeiro foi implementada uma versão onde o dispositivo só se desloca em quatro direções uma distância fixa. Depois foi implementada a variante onde um dispositivo anda numa direção aleatória com uma certa velocidade. Na Figura 6.1 temos a segunda implementação do modelo de mobilidade Random Walk.

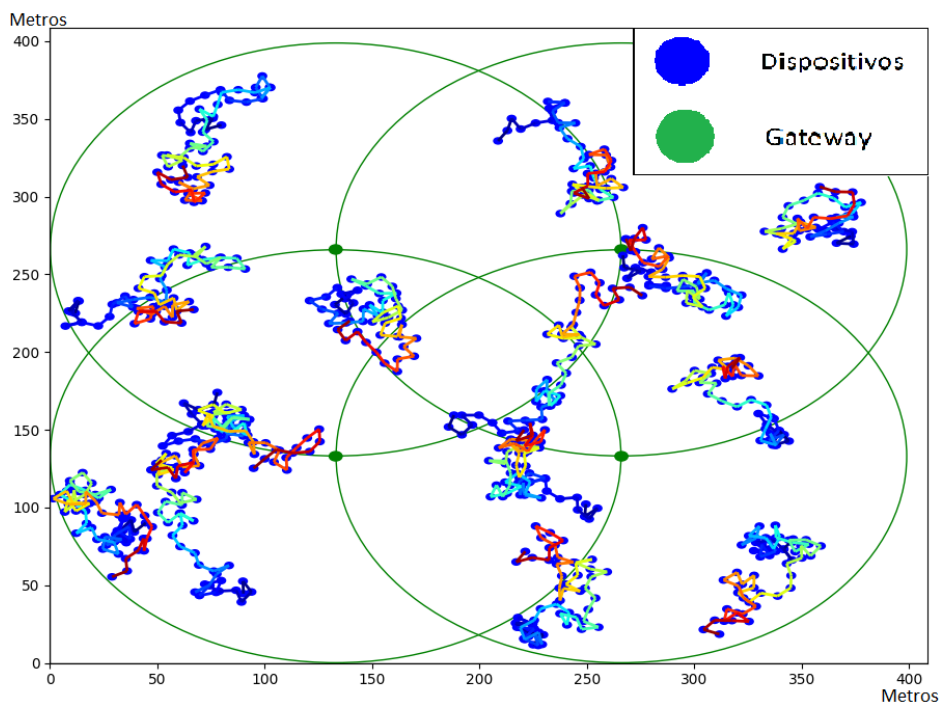


Figura 6. 1 - Distribuição Random Walk

O modelo Random Walk é um modelo sem memória e a cada ponto é calculada uma nova direção. Cada ponto na imagem representa uma transmissão de trama e a cor altera de azul para vermelho para mostrar a deslocação que o nó teve durante a simulação.

```
def random_walk(node):
    wlkTime = node.period/1000
    wlkDist = wlkTime*1.4
    if rnd_mode == 1:
        val = random.randint(1,4)
        if val == 1:
            node.x +=wlkDist
        elif val == 2:
            node.x -=wlkDist
        elif val == 3:
            node.y += wlkDist
        else:
            node.y -= wlkDist
    else :
        # Choose a random direction
        direction = random.uniform(0, 2*3.14)
        # Update the position based on the step size and direction
        node.x=node.x + wlkDist * math.cos(direction)
        node.y=node.y + wlkDist * math.sin(direction)
    return node
```

Podemos ver no excerto de código acima que a primeira metade é uma implementação do Random Walk que só anda nas direções cardinais enquanto a segunda metade implementa a segunda versão onde é escolhida uma direção aleatória.

Outro modelo muito utilizado para o teste de redes sem fios é o modelo de mobilidade Gauss-Markov. Este modelo, ao contrário do Random Walk, incorpora um grau de previsão e correlação temporal no movimento de um nó, tornando-o mais representativo de cenários reais onde o movimento não é completamente imprevisível nem certo.

O modelo de mobilidade Gauss-Markov é um modelo com memória, onde a nova posição é determinada pelo cálculo de uma direção e velocidade. Estes novos valores de direção e velocidade são influenciados utilizando uma variável, α , e os valores calculados na posição

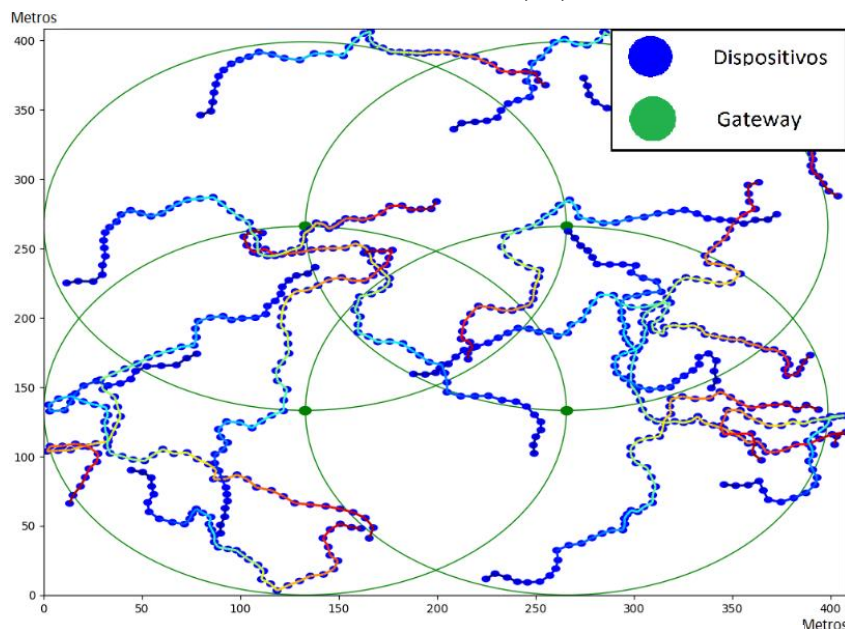


Figura 6.2 - Distribuição Gauss-Markov

anterior. Na Figura 6.2 é possível observar o comportamento tomado por um dispositivo onde a direção se mantém constante e melhor representa o movimento humano em termos de comportamento, sendo a mudança de direção menos acentuada.

Este modelo oferece um balanço entre a complexidade do modelo e a precisão da simulação o que o torna bastante favorável para o estudo do impacto que mobilidade tem nas redes.

```
def gauss_markov(node, velocity_mean=6., alpha=1., variance=1.):
    alpha2 = 1.0 - alpha
    alpha3 = np.sqrt(1.0 - alpha * alpha) * variance
    if node.vel == 0:
        node.vel += velocity_mean
    if node.dir == 0:
        node.dir = np.random.rand() * 2*np.pi
        node.mean_dir = node.dir
    Xn = node.x + node.vel * np.cos(node.dir)
    Yn = node.y + node.vel * np.sin(node.dir)
    if Xn<0:
        Xn = - Xn; node.dir = np.pi-node.dir; node.mean_dir = np.pi-node.mean_dir
    if Xn>xmax:
        Xn = 2*xmax - Xn; node.dir = np.pi-node.dir; node.mean_dir = np.pi-node.mean_dir
    if Yn<0:
        Yn = - Yn; node.dir = -node.dir; node.mean_dir = -node.mean_dir
    if Yn>ymax:
        Yn = 2*ymax - Yn; node.dir = -node.dir; node.mean_dir = -node.mean_dir
    # calculate new speed and direction based on the model
    node.vel = (alpha * node.vel + alpha2 * velocity_mean + alpha3 * np.random.normal(0.0, 1.0))
    node.dir = (alpha * node.dir + alpha2 * node.mean_dir + alpha3 * np.random.normal(0.0, 1.0))
    node.x=Xn
    node.y=Yn
    return node
```

No excerto de código acima está representada a implementação deste modelo de mobilidade utilizando a equação 5.1 e 5.2. Começamos por inicializar os valores de alpha2 e alpha3 que são variantes de alpha, que por sua vez representa o grau de influência na velocidade e direção. Durante os testes foi utilizado um alpha igual a 0.8, o que significa que 80% do novo estado é influenciado pela antiga posição, enquanto o resto é influenciado por uma distribuição gaussiana. No meio são criadas regras para conter os dispositivos a área que estamos a observar. Por fim é calculada a nova velocidade e a nova direção.

6.2. Parâmetros Utilizados

A posição de um dispositivo pode ser calculada após uma certa distância percorrida ou a passagem de um determinado período de tempo. Como foi implementado o modelo de mobilidade a posição de um nó é alterada após um período específico de tempo, influenciado primariamente pelo SF. Para todos os testes foram feitas variações de parâmetros de transmissão, de mobilidade e a quantidade de estações existentes.

Tabela 8 - Parâmetros de uma transmissão LoRa

SF	7
BW	125 kHz
CR	4/5
Tx	14 dBm

Como sabemos que para os parâmetros na Tabela 8 um nó tem um tempo no ar de 56.6 ms antes de poder voltar a transmitir é necessário que o nó espere 5.7s para cumprir com o *Duty-Cycle* de 1%. Sendo que, em média, uma pessoa anda á velocidade de 1.4 m/s, no tempo entre transmissões o nó anda 8 m antes de voltar a transmitir.

Tabela 9 - Parâmetros de uma transmissão com SF12

SF	12
BW	125 kHz
CR	4/5
Tx	14 dBm

Para os parâmetros na Tabela 9 devido a mudança de SF para 12 o tempo no ar por transmissão vai aumentar para 1318.9 milissegundos o que significa que para termos um *duty-cycle* igual ou menor a 1% os equipamentos tem de esperar 132 segundos entre transmissões.

Na Tabela 10 e Tabela 11 temos a alteração da potência de transmissão na comunicação LoRa, feito para analisar a diferença do consumo energético entre os vários protocolos em transmissões de baixa energia.

Tabela 10 - Parâmetros de Transmissão, alteração Tx para SF7

SF	7
BW	125 kHz
CR	4/5
Tx	2 dBm

Tabela 11 - Parâmetros de Transmissão, alteração Tx para SF12

SF	12
BW	125 kHz
CR	4/5
Tx	2 dBm

Para além destas alterações de parâmetros de comunicação LoRa foi também feita uma variação da quantidade de estações base, e a variação do tempo de simulação em função do

SF. Para testes com SF de 7 foi utilizado um tempo de execução do ambiente de simulação de 5 minutos. Em testes com SF12 foi realizado um teste com 100 minutos. Para cada mudança de parâmetros de simulação como por exemplo a alteração do número de dispositivos é necessário correr o simulador.

7. Resultados

De forma a avaliarmos o desempenho de uma rede de dispositivos IoT utilizamos o simulador anteriormente mencionado, LoRaSim.

Os resultados obtidos na simulação são passados para um ficheiro CSV, de onde são extraídos os parâmetros relevantes à análise dos resultados, estando entre eles, a probabilidade de colisão, o número de pacotes recebidos, o número de colisões e a energia gasta no envio do pacote. A energia gasta tem interesse em ser avaliada neste contexto devido aos dispositivos utilizarem maioritariamente baterias como fonte de alimentação.

O primeiro teste realizado foi um teste de controlo para ver a resposta padrão do sistema à simulação de uma comunicação da rede. Com um número a variar de 1 a 500 dispositivos e quatro *gateways* com uma disposição igual à da Figura 7.1.

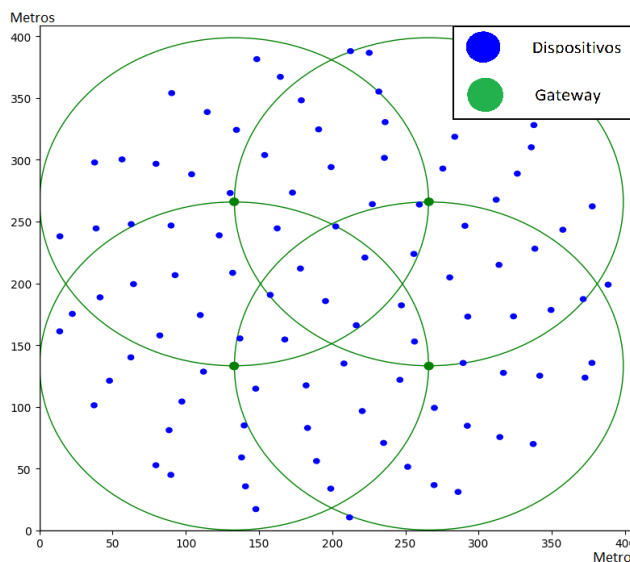


Figura 7. 1 - Posição de dispositivos LoRa numa implementação fixa

Esta disposição tem o intuito de simular uma aplicação estática da tecnologia. Um teste padrão que podemos realizar é utilizando a configuração de dispositivos e *gateways* estática e com os parâmetros utilizados na Tabela 8 são obtidos vários valores, entre eles a probabilidade de colisão em função do número de dispositivos numa rede.

Este cenário é utilizado como referência porque a única alteração entre os vários testes é o protocolo utilizado, mantendo-se constante o SF a potência e a largura de banda. No total entre os vários protocolos e variações foram realizadas mais de 150 simulações de rede com o programa demorando cada uma por media 3 minutos, uma da vantagem de um simulador de eventos discreto pois cada simulação de rede envolveu a subsequente utilização do simulador para cada mudança do número de dispositivos na rede.

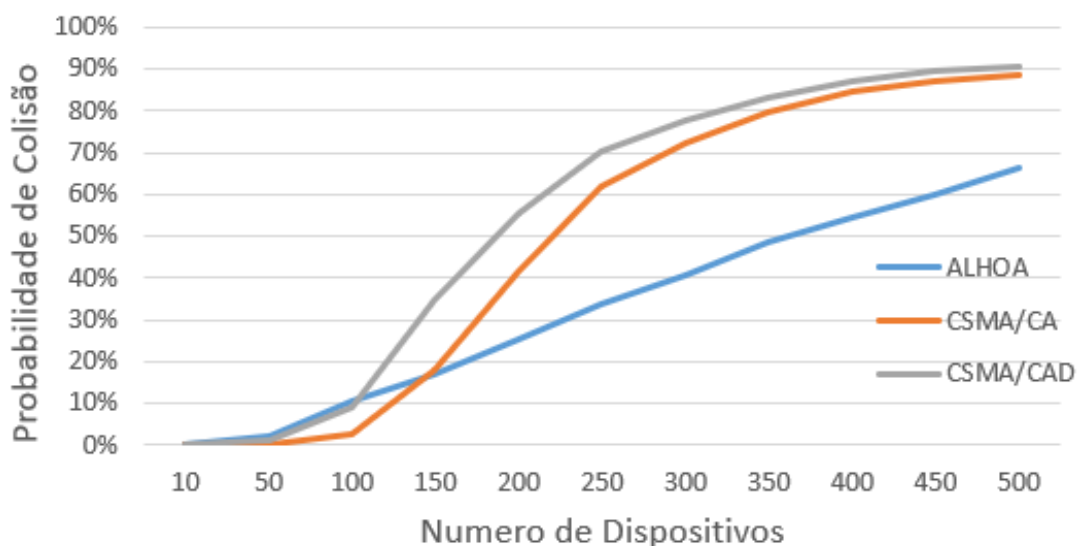


Figura 7.2 - Probabilidade de Colisão SF7 com posição estática

É possível ver na Figura 7.2 que a probabilidade de colisão aumenta com a ocupação de rede. Para uma comunicação normal LoRa o aumento é linear o que significa que a um certo ponto a probabilidade de colisão vai ser de 100%. Para o CSMA/CA e o CSMA/CAD a probabilidade de colisão cresce mais rápido em redes de médio tamanho, mas tem um comportamento logarítmico onde ele se aproxima do limite da probabilidade de colisão.

Foi executado um teste com SF12, para ser observada a diferença que a variação de SF tem no desempenho dos protocolos. Devido ao tempo de transmissão ser superior com SF maiores o dispositivo, por consequência, tem um tempo de ocupação de canal superior, o que leva a ser necessário aumentar o tempo entre transmissões nos parâmetros do LoRaSim para cumprir com o *duty-cycle*.

Utilizando os valores presentes na Tabela 9 é possível ver na Figura 7.3 que a probabilidade de colisão em redes com um pequeno número de dispositivos aumenta devido a ocupação do meio. Tal como com SF7 é possível ver um aumento linear do protocolo ALOHA enquanto o CSMA/CA obteve um melhor desempenho com valores mais baixos de equipamentos.

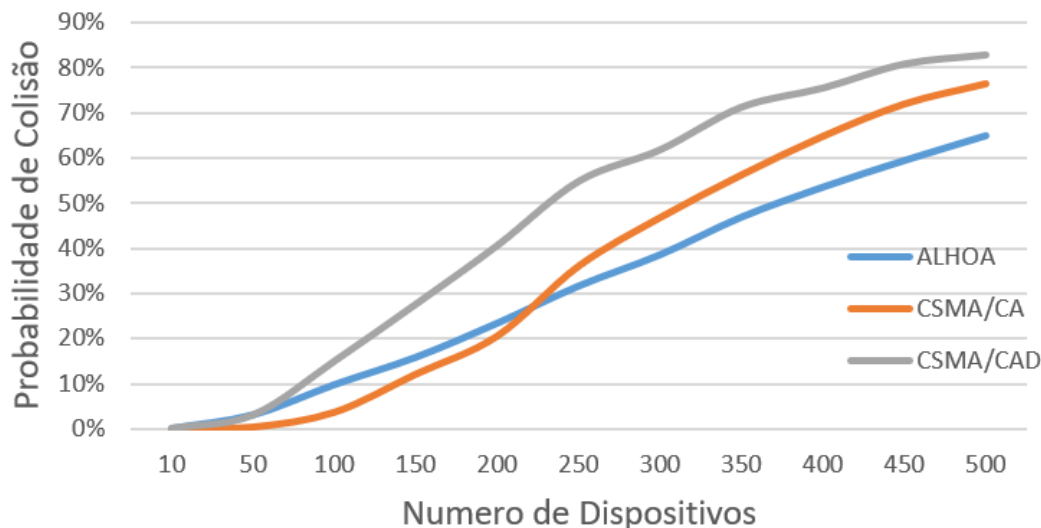


Figura 7. 4 - Probabilidade de colisão para SF12 com posição estática

Outra métrica importante de avaliar é o gasto energético dos dispositivos para cada cenário avaliado. Neste sentido é possível ver na Figura 7.4 o consumo de energia para cada simulação. Conseguimos ainda observar que a energia gasta pelo protocolo ALOHA mantém-se constante independentemente do número de dispositivos existentes na rede. O CSMA/CA e o CSMA/CAD têm um consumo superior devido aos mecanismos de *backoff*.

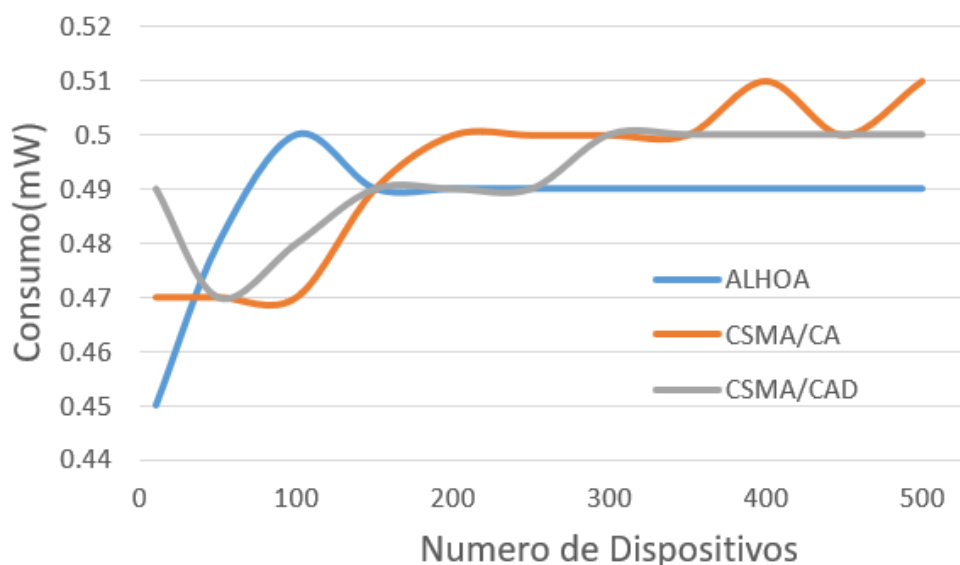


Figura 7. 3 - Consumo de energia por protocolo com SF7

Na Figura 7.5 temos o consumo de energia para uma comunicação com SF12 e comparando os valores com os obtidos em SF7 é possível ver um aumento significativo na energia gasta durante a simulação. O ALOHA manteve o valor de consumo constante entre todos os testes, enquanto os protocolos CSMA/CA e CSMA/CAD demonstram um aumento devido a adição do mecanismo de *backoff*.

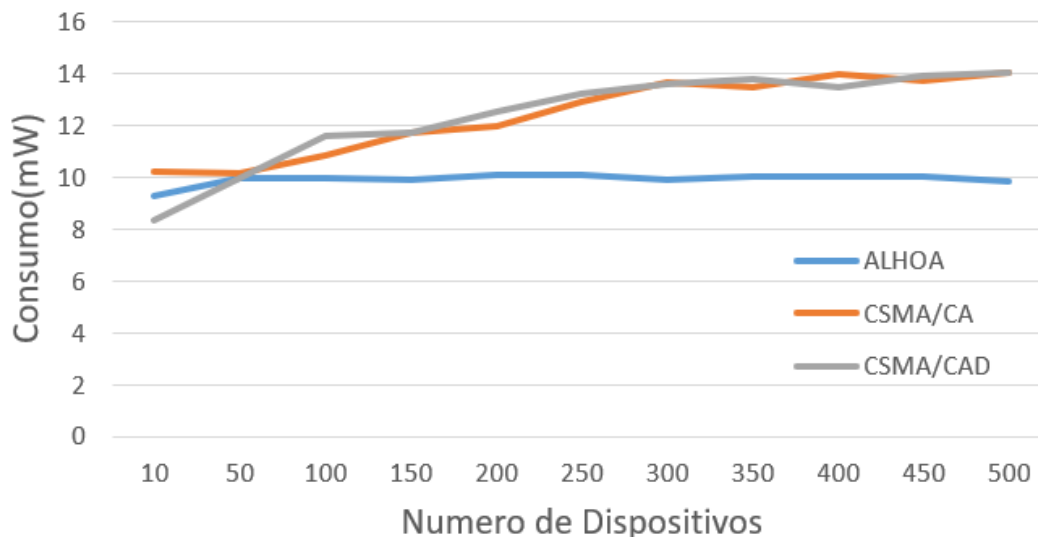


Figura 7. 6 - Consumo de energia por protocolo com SF12

Outro teste realizado foi a alteração da potência de transmissão de 14 dBm para 2 dBm. De forma a ver que influência a mínima e a máxima potência de transmissão esta alteração tem nas comunicações LoRa nomeadamente no consumo energético dos protocolos, o alcance de um

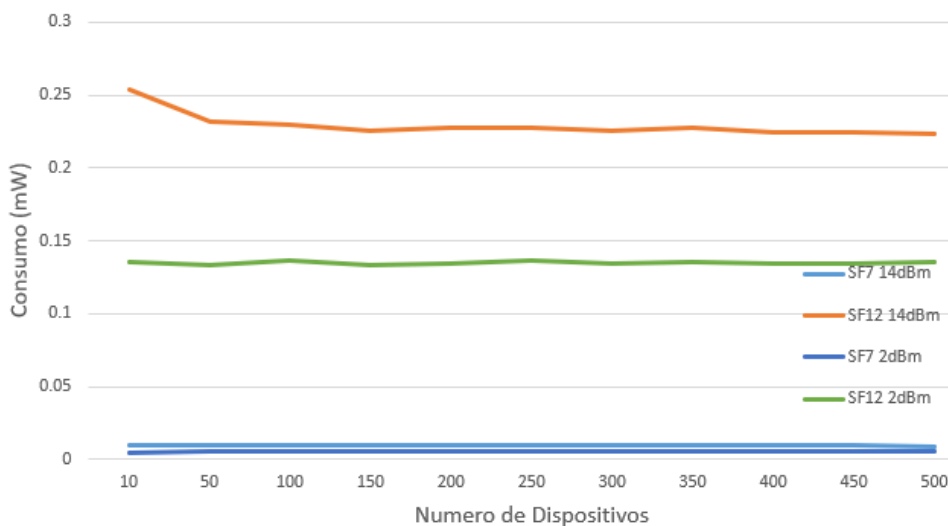


Figura 7. 5 - Consumo medio de energia por trama durante simulação.

dispositivo, e a probabilidade de colisão.

Na Figura 7.6 conseguimos observar o consumo de energia médio por trama enviada. Podendo ser observado que para SF7 a diferença de consumo é mínima para o acréscimo em alcance dos dispositivos. Em SF12 o consumo médio por trama aumenta e a diferença de potência de transmissão já causa um aumento significativo.

Com a adição dos modelos de mobilidades, foram repetidos os testes anteriormente realizados.

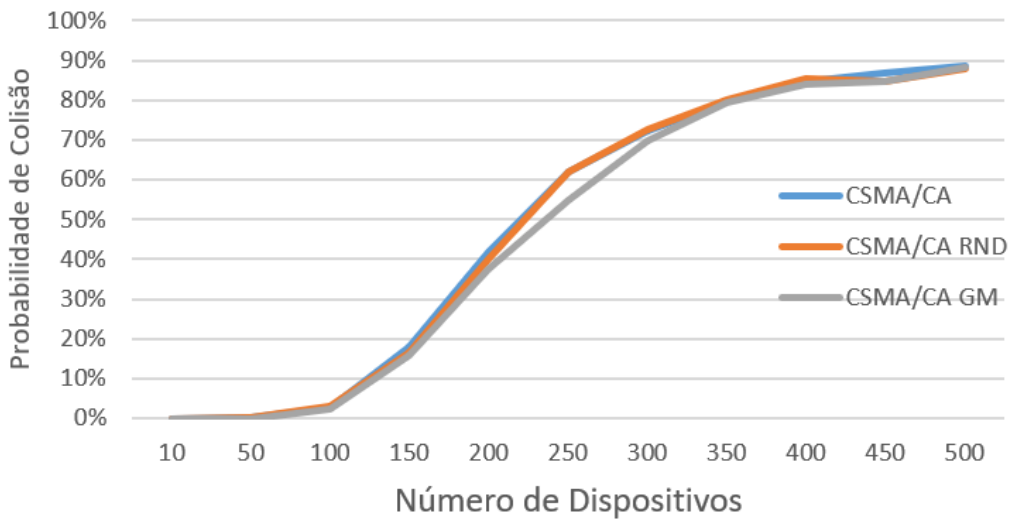


Figura 7. 7 - Colisão em redes LoRa com Modelos de Mobilidade

Na Figura 7.7 é possível ver o comportamento de uma rede LoRa com a posição estática de dispositivos, demonstrado pela linha azul, e a posição determinada pelos modelos de mobilidade representado pelas restantes cores.

É possível ver um ligeiro decréscimo na probabilidade de colisão com a adição da mobilidade de nó. Esta diminuição é resultante do movimento dos dispositivos, o que introduz flutuação no número de dispositivos que uma célula tem a um dado momento. O modelo de mobilidade Random Walk não mantém memória da posição anterior nem da direção do movimento de um dispositivo. Como é possível ver na Figura 6.1 existe uma tendência a se manter na mesma área, logo o seu resultado é mais idêntico ao teste com a posição de cada nó estática.

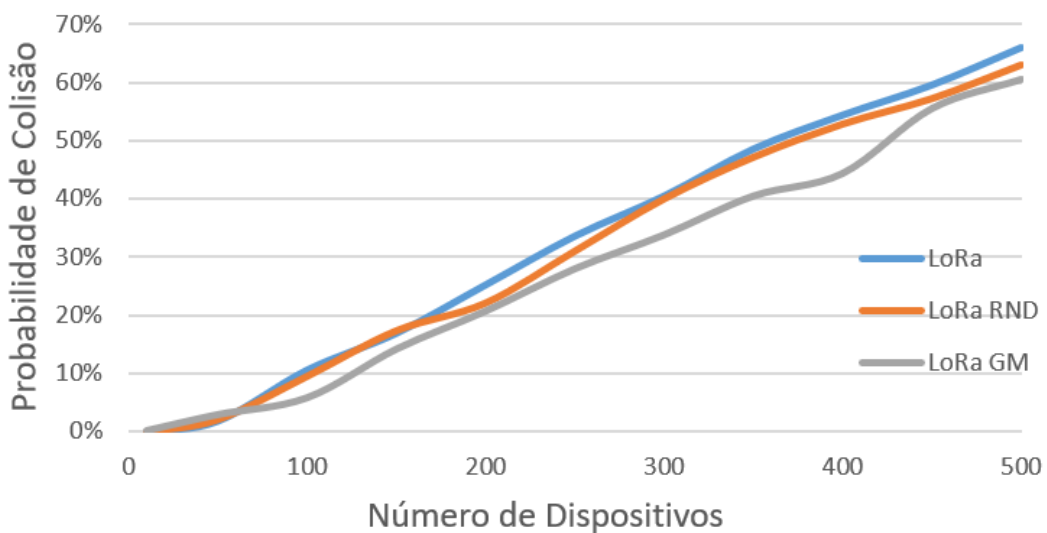


Figura 7. 8 - Colisão em redes LoRa com protocolo CSMA/CA

Por outro lado, o modelo de mobilidade Gauss-Markov retém a memória da posição previa do dispositivo e a direção em que ele se move, desta forma a probabilidade de trocar de *gateways* para uma menos ocupada aumenta.

A Figura 7.8 demonstra a utilização de um modelo de mobilidade com o protocolo de acesso ao meio CSMA/CA e demonstra que a utilização de um modelo de mobilidade diminui a probabilidade de colisão

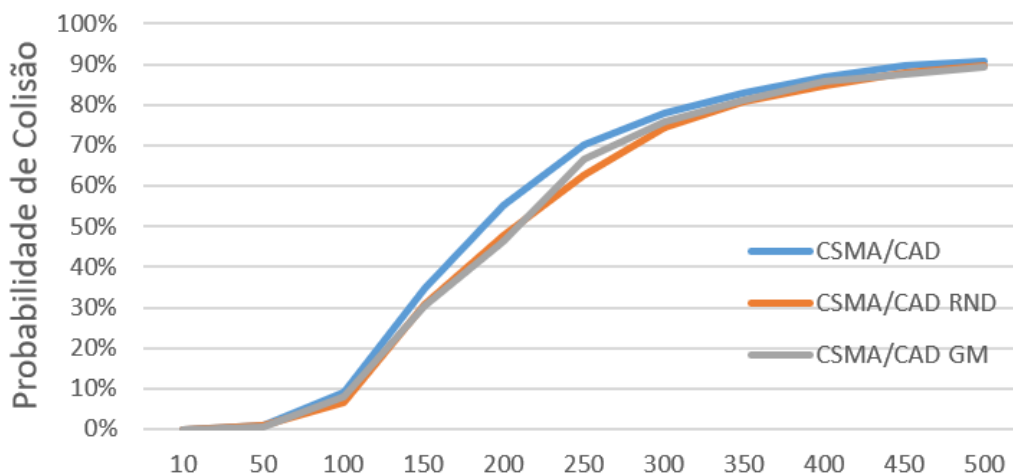


Figura 7. 9 - Colisão em redes LoRa com protocolo CSMA/CAD

Numa rede com um número pequeno de dispositivos a probabilidade de colisão é pequena independentemente da mobilidade dos nós, em redes com ocupação média a utilização de modelos de mobilidade demonstrou uma menor probabilidade de colisão devido ao movimento dos dispositivos entre as diferentes *gateways*. Em redes mais ocupadas a probabilidade da colisão de tramas cresce novamente e o efeito da mobilidade dos dispositivos é diminuída.

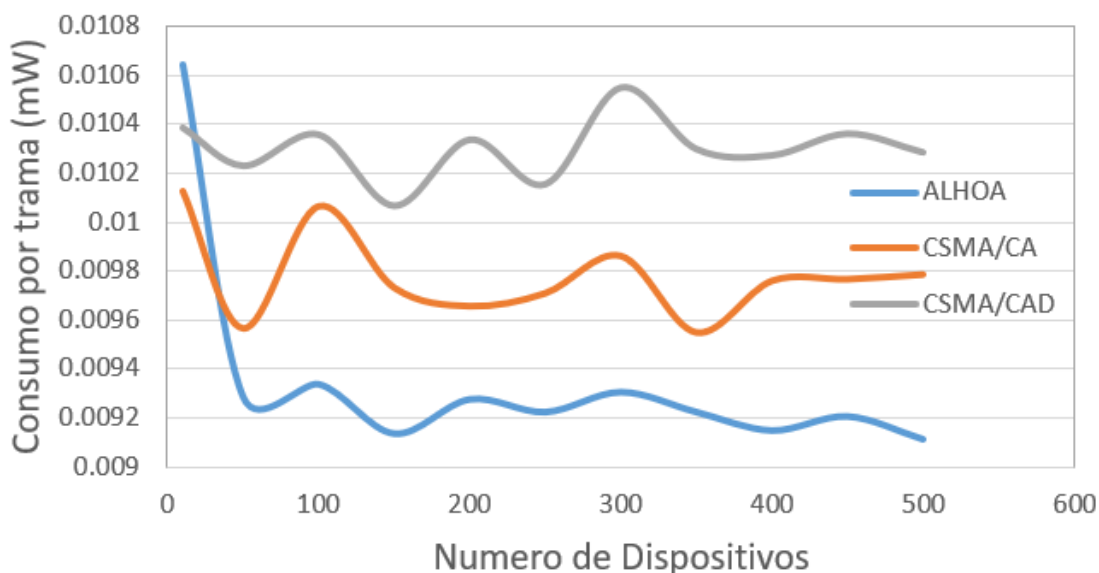


Figura 7. 10 - Consumo por trama com Movimento Random Walk

A introdução dos modelos de mobilidade também levou a alteração da energia gasta pelos dispositivos nas redes.

Na Figura 7.10 esta representado o consumo energético médio por trama. É possível ver que por trama os protocolos CSMA/CA e CSMA/CAD gastaram mais que a utilização normal da rede LoRa, isto é esperado porque a adição de tempos de espera e escuta aumentam o gasto de energia.

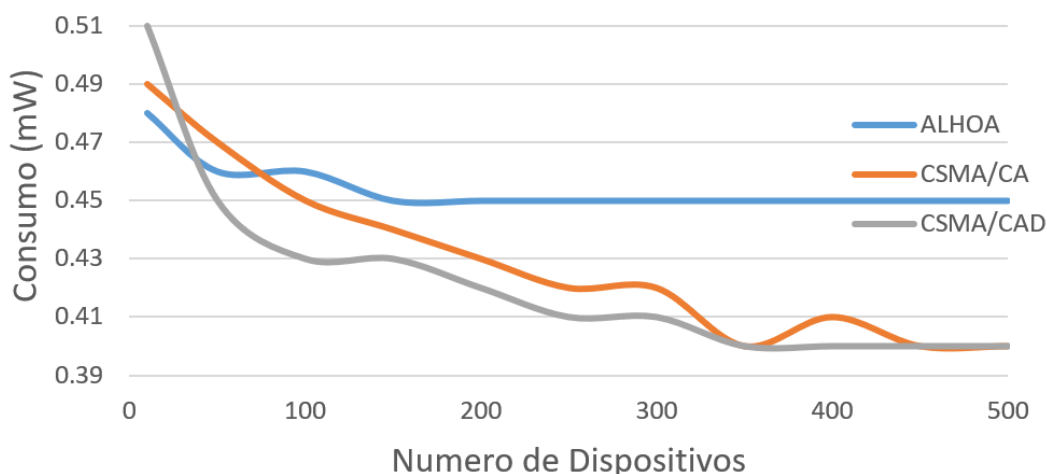


Figura 7. 11 - Consumo total por protocolo com Random Walk

A Figura 7.11 apresenta o consumo de uma simulação completa do modelo de mobilidade Random Walk, podemos ver que o valor de energia gasta durante a simulação é menor com o aumento do número de dispositivos. Comparando com a Figura 7.10 podemos ver que embora o gasto energético por trama seja maior o consumo total durante o tempo de simulação é menor devido aos tempos de espera e retransmissões que atrasam o envio de uma trama.

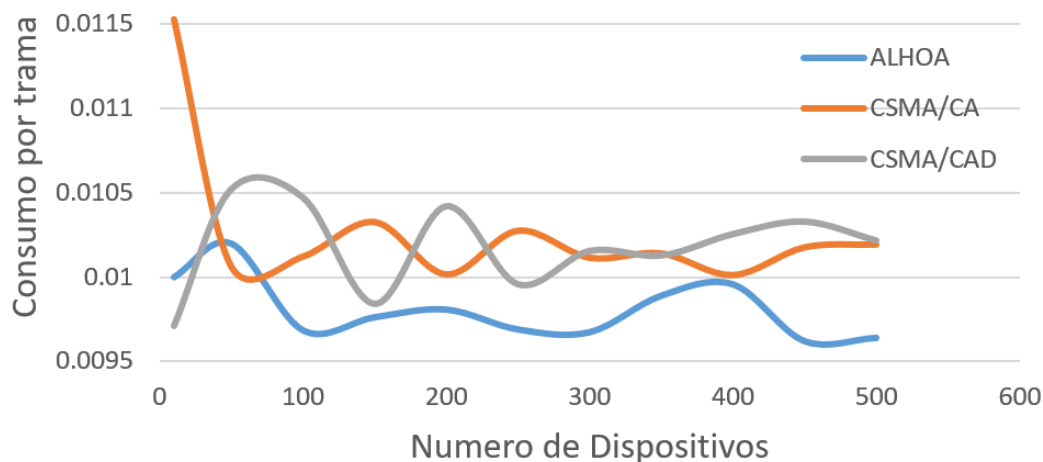


Figura 7. 12 - Consumo por trama com Movimento Gauss-Markov

Como podemos ver na Figura 7.12, tal como no modelo Random Walk, o modelo Gauss-Markov apresenta um maior consumo de energia por trama que o funcionamento normal da rede LoRa. E na Figura 7.13 podemos ver a diminuição do consumo total com a adição de dispositivos na rede.

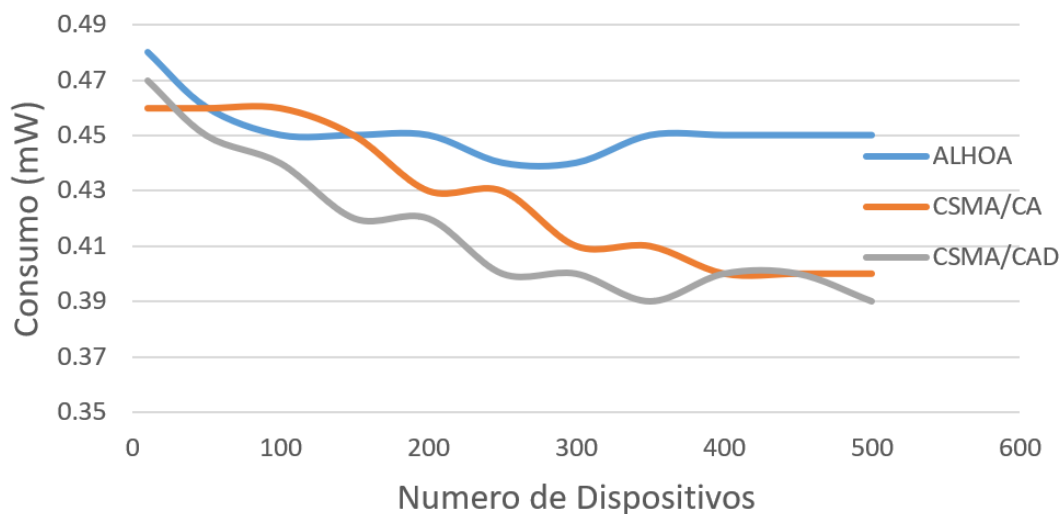


Figura 7. 13 – Consumo total por protocolo com Gauss-Markov

É possível ver que a introdução da mobilidade de dispositivos influencia o comportamento de uma rede LoRa e é importante considerar estes cenários no planeamento e construção de redes.

| 8. Conclusões

Esta dissertação teve como objetivo estudar a influência da mobilidade em redes IoT mais concretamente a rede LoRa. Ao longo deste trabalho foram desenvolvidos conhecimentos ao nível das comunicações sem fios *wireless*, em particular a tecnologia LoRa, e outros métodos de acesso ao meio utilizados. Foi estudado e avaliado o simulador LoRaSim e com base nele foram desenvolvidos e adicionados vários módulos.

No que diz respeito ao objetivo deste trabalho podemos concluir que este foi atingido, uma vez que utilizando o simulador como base foram desenvolvidos e estudados protocolos de acesso alternativos ao atualmente utilizado na rede LoRa, como o CSMA/CA e o CSMA/CAD e o seu desempenho com um número elevado de dispositivos. Para avaliar o tema da dissertação sobre o desempenho da rede LoRaWAN em cenários de mobilidade foram implementados modelos de mobilidade teóricos como o *Random Walk* e *Gauss-Markov* e foi ainda equacionada a utilização de *datasets* como forma de extrair posições reais para utilização.

Com o estudo feito dos protocolos de acesso ao meio foi possível ver que o uso atual da tecnologia LoRa obtém um melhor resultado em redes com uma maior ocupação, sendo que em redes com menor ocupação não causa grande diferença devido ao facto da colisão de tramas ser esporádica e o acrescento de complexidade e gasto energético no desenho do dispositivo pode não compensar o acréscimo na fiabilidade da recessão de dados.

Quanto aos testes realizados, podemos concluir que os dados seguem o padrão esperado, sendo que a introdução de mobilidade diminui ligeiramente a carga na rede, devido a ocupação e desocupação das *gateways*. A adição de mobilidade diminui a probabilidade de colisão dentro de um cenário com o mesmo número de *gateways*, exceto com uma *gateway* onde o conceito de mobilidade não entra em efeito porque o dispositivo só comunica com uma. No que toca a probabilidade de colisão, esta diminui com as adições de *gateways*.

No cenário padrão é possível ver que até aos 150 dispositivos os protocolos adicionados obtém um melhor desempenho, sendo o CSMA/CA ficado perto dos 0% de colisão até aos 100 dispositivos e só regista valores parecidos aos do CSMA/CAD e do ALOHA com 150 dispositivos. A adição dos modelos de mobilidade é essencial para a simulação e otimização de redes LoRa, especialmente em ambientes onde os dispositivos podem ser móveis. Neste intuito

os modelos de mobilidade Random Walk e o Gauss-Markov foram adicionados para ser possível avaliar uma aproximação ao movimento humano.

O modelo Random Walk é um bom teste de mobilidade porque se consegue aproximar de movimentos imprevisíveis e aleatórios tal como o movimento de pedestres ou animais. Devido à sua simplicidade e requerimentos computacionais mínimos pode ser facilmente integrado em vários ambientes de simulação nomeadamente o LoRaSim. Este modelo é particularmente bom para testes de stress a protocolos de rede para ver como se comportam em cenários menos previsíveis. Embora não consiga capturar todos os tipos de mobilidade, tal como veículos, a sua facilidade de utilização, implementação e baixa complexidade computacional conferem a este modelo um renovado interesse na sua utilização para o cenário que queremos avaliar.

Durante os testes foi possível ver que a adição do modelo de mobilidade reduziu ligeiramente a probabilidade de colisão em todos os protocolos mais notavelmente no protocolo adicionado, CSMA/CAD. Quanto ao consumo durante a utilização do modelo Random Walk, comparando com o cenário padrão onde os dispositivos estavam fixos podemos ver que os valores de consumo por trama do CSMA/CA e do CSMA/CAD são superiores ao funcionamento normal LoRa em ambos os casos, contudo o consumo total da simulação diminui com a introdução do Random Walk.

O modelo de mobilidade Gauss-Markov é considerado uma boa escolha para simular movimentos dimensionais ou correlacionais devido a ser um modelo com memória, comparando com a natureza aleatória do modelo Random Walk. Em cenários onde é necessário uma correlação temporal ou um grau de previsibilidade no movimento o modelo Gauss-Markov consegue gerar uma representação mais fidedigna.

Este modelo é particularmente utilizado em situações onde é necessário simular a mobilidade com graus variados de aleatoriedade e de direção, uma vez que permite ajustar o parâmetro α para controlar a variação dos valores. Um valor baixo torna o movimento mais aleatório, enquanto um α mais alto torna o movimento mais corelacionado com o estado anterior.

A desvantagem do modelo Gauss-Markov é o facto de ser mais intensivo computacionalmente que o Random Walk, devido aos cálculos para a correlação direcional e velocidade. Em contrapartida oferece um nível de realismo mais alto para certos tipos de movimentos, tal como a deslocação a pé.

Após a implementação do modelo de mobilidade Gauss-Markov foram feitos os mesmos testes que no cenário padrão. A introdução deste modelo diminui a probabilidade de colisão significativamente em comparação com os testes do modelo Random Walk e a implementação estática da tecnologia no cenário padrão. A desocupação de uma rede devido à mobilidade dos dispositivos torna este cenário uma realidade que tem interesse em ser estudada.

Comparando o consumo por trama na utilização do Gauss-Markov os protocolos CSMA/CA e CSMA/CAD tiveram um consumo superior ao protocolo atual da rede LoRa. Já o consumo durante o tempo de simulação foi menor com o aumento do número de dispositivos.

Para trabalho futuro poderão ser estudados e aplicados outros modelos de mobilidade e outros métodos de abordagem uma vez que estes modelos não conseguem representar todos os cenários. A utilização de *datasets* fornece dados de mobilidade baseados em cenários reais, contudo necessita de alinhar com o contexto do simulador e com os testes realizados. Pois não poderia ser aplicado dados de uma aplicação em ambiente livre, com dados a cada hora, com uma simulação de um ambiente urbano com a frequência que necessitamos para o teste que queremos fazer.

9. Bibliografia

- [1] SEMTECH, “LoRa™ Modulation Basics,” Semtech Corporation, 2015.
- [2] LoRa Alliance, "LoRa Alliance," [Online]. Available: <https://lora-alliance.org/>.
- [3] Voigt, M. Bor <https://mcbor.github.io/lorasim/#installation>
- [4] “Low Power Wide Area Technology,” [Online]. Available: <https://www.thalesgroup.com/en/markets/digital-identity-and-security/iot/resources/innovation-technology/low-power-wide-area-technology>.
- [5] “ALOHAnet,” Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/ALOHAnet#Pure_ALOHA. [Accessed 1 2022].
- [6] F F. Kuo, "Computer Networks–The ALOHA System" Journal of research of the National Bureau of Standards (1977) vol. 86,6 (1981): 591-595. doi:10.6028/jres.086.027
- [7] A. Xanthopoulos, A. Valkanis, G. Beletsioti, G. I. Papadimitriou and P. Nicopolitidis, "On the Use of Backoff Algorithms in Slotted ALOHA LoRaWAN Networks," 2020 International Conference on Computer, Information and Telecommunication Systems (CITS), Hangzhou, China, 2020, pp. 1-4, doi: 10.1109/CITS49457.2020.9232577.
- [8] L. Kleinrock and F. Tobagi, "Packet Switching in Radio Channels: Part I - Carrier Sense Multiple-Access Modes and Their Throughput-Delay Characteristics," in IEEE Transactions on Communications, vol. 23, no. 12, pp. 1400-1416, December 1975, doi: 10.1109/TCOM.1975.1092768.
- [9] A. El-Hoiydi and J. . -D. Decotignie, "WiseMAC: an ultra low power MAC protocol for the downlink of infrastructure wireless sensor networks," Proceedings. ISCC 2004. Ninth International Symposium on Computers And Communications (IEEE Cat. No.04TH8769), Alexandria, Egypt, 2004, pp. 244-251 Vol.1, doi: 10.1109/ISCC.2004.1358412.
- [10] I. Demirkol, C. Ersoy and F. Alagoz, "MAC protocols for wireless sensor networks: a survey," in IEEE Communications Magazine, vol. 44, no. 4, pp. 115-121, April 2006, doi: 10.1109/MCOM.2006.1632658.
- [11] M. Bor, U. Roedig, T. Voigt, & J. Alonso, “Do LoRa low-power wide-area networks scale?” in MSWiM '16 Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems. ACM Press, New York, pp. 59-67. <https://doi.org/10.1145/2988287.2989163>

- [12] A.Alves L.Pires L.Gonçalves “Development and Analyses of a New implementation of CSMA/CAD algorithm for LoRaSim Simulator applied to LoRa networks” INForum, 2022, ISBN. 978-989-654-836-0
- [13] INForum [online] <https://inforum.org.pt>
- [14] A.Alves L.Pires L.Gonçalves “Estudo e Análise da Mobilidade de Dispositivos em Redes LoRa” 16.º Congresso do Comité Português da URSI
- [15] razilian Journals Publicações. “Brazilian Journal of Development.” Brazilian Journal of Development, edited by Brazilian Journals Publicações de Periódicos e Editora Ltda., CNPJ 32.432.868/0001-57, 1. Available online: Brazilian Journal of Development, ISSN: 2525-8761
- [16] A. E. S. Alves, L. M. R. Pires, and L. C. Gonçalves, “Development and analysis of new implementations of MAC protocols and mobility models in LoRa networks”, Braz. J. Develop., vol. 9, no. 3, pp. 9188–9205, Mar. 2023.
- [17] Lygia Aparecida das Graças Gonçalves Corrêa, “Ciências exatas e suas tecnologias”, Brazilian Journals, 2023, ISBN: 978-65-6016-006-4
- [18] A. E. S. Alves, L. M. R. Pires, and L. C. Gonçalves, “Development and Analysis of Mobility Models in LoRa Networks
- [19] I-SCOOP, “The Netherlands is the first country to have nationwide LoRa IoT network coverage,” I-SCOOP, [Online]. Available: <https://www.i-scoop.eu/the-netherlands-is-the-first-country-to-have-nationwide-lora-iot-network-coverage/>. [Accessed 14 12 2021].
- [20] N. Cota, “3. Low Power Wireless Communications Infrastructure and Protocols,” ISEL, 2020.
- [21] Devopedia 2021, “NB-IoT,” [Online]. Available: <https://devopedia.org/nb-iot>. [Accessed 16 1 2022].
- [22] A. G. TEIXEIRA, “Rede de sensores wireless baseada na tecnologia LoRa para monitorização de processo na indústria,” 2019.
- [23] G. Dregvaite and R. Damasevicius, “Information and Software Technologies: 22nd International Conference, ICIST 2016, Druskininkai, Lithuania, October 13-15, 2016, Proceedings,” in Communications in Computer and Information Science, vol. 639, Springer, 2016.
- [24] SIGFOX, “OUR STORY,” [Online]. Available: <https://www.sigfox.com/en/sigfox-story>. [Accessed 20 1 2022].

- [25] 3GPP, “GSM Spec history,” [Online]. Available: <https://www.3gpp.org/specifications/gsm-history>. [Accessed 10 12 2021].
- [26] 3GPP, “Standards for the IoT,” [Online]. Available: https://www.3gpp.org/news-events/1805-iot_r14. [Accessed 14 12 2022].
- [27] Group Thales, “Narrowband IoT,” [Online]. Available: <https://www.thalesgroup.com/en/markets/digital-identity-and-security/iot/resources/innovation-technology/nb-iot>. [Accessed 15 12 2021].
- [28] Techplayon, “Maximum Coupling Loss (MCL) and Maximum Path Loss (MPL),” [Online]. Available: <https://www.techplayon.com/maximum-coupling-loss-mcl-and-maximum-path-loss-mpl/>. [Accessed 22 December 2021].
- [29] CAMARILLO, “Semtech Acquires Wireless Long Range IP Provider Cycleo,” 7 3 2012. [Online]. Available: <https://www.design-reuse.com/news/28706/semtech-cycleo-acquisition.html>. [Accessed 17 1 2022].
- [30] E. Inacio. A. Alves, “Estudo e Análise da Colisão de Tramas na LoRaWAN 868 MHz”.
- [31] R. N. M. Askari M., “Assignment of OVFS Codes in Wideband CDMA,” in *Advances in Computer Science and Engineering*, 2009, pp. 723-727.
- [32] V. Fialho, F. Azevedo, "Wireless Communication Based on Chirp Signals for LoRa IoT Devices" i-ETC: ISEL Academic Journal of Electronics, Telecommunications and Computers IoT-2018 Issue, Vol. 4, n. 1 (2018) ID-6
- [33] K. L. J. K. Dong-Hoon Kim, “Experiencing LoRa Network Establishment on a Smart Energy Campus Testbed,” [Online]. Available: https://www.researchgate.net/figure/LoRa-uses-a-chirp-spread-spectrum-modulation-technique-The-figure-also-shows-that-LoRa_fig3_332151302.
- [34] LoraTools, “Calculate the air time of your LoRa frame.,” [Online]. Available: <https://loratools.nl/#/airtime>.
- [35] A. Mishra, "Enhanced slotted aloha," 2011 3rd International Conference on Electronics Computer Technology, 2011, pp. 278-282, doi: 10.1109/ICECTECH.2011.5942002.
- [36] Federal Communications Commission, “Authorization of Spread Spectrum Systems Under Parts 15 and 90 of the FCC Rules and Regulations,” 1985.
- [37] N. Cruz, “Connecting to the Cloud”.
- [38] The MathWorks, Inc. (2022). *Optimization Toolbox version: 9.4 (R2022b)*. Accessed: January 01, 2023. Available: <https://www.mathworks.com>

- [39] A. Xanthopoulos, A. Valkanis, G. Beletsioti, G. I. Papadimitriou and P. Nicopolitidis, "On the Use of Backoff Algorithms in Slotted ALOHA LoRaWAN Networks," 2020 International Conference on Computer, Information and Telecommunication Systems (CITS), Hangzhou, China, 2020, pp. 1-4, doi: 10.1109/CITS49457.2020.9232577.
- [40] C. Pham, "Investigating and experimenting CSMA channel access mechanisms for LoRa IoT networks," 2018 IEEE Wireless Communications and Networking Conference (WCNC), Barcelona, Spain, 2018, pp. 1-6, doi: 10.1109/WCNC.2018.8376997.
- [41] "IEEE Standard for Low-Rate Wireless Networks Corrigendum 1: Correction of Errors Preventing Backward Compatibility," in IEEE Std 802.15.4-2020/Cor 1-2022 (Corrigendum to IEEE Std 802.15.4-2020 as amended by IEEE Std 802.15.4z-2020, IEEE Std 802.15.4w-2020, IEEE Std 802.15.4y-2021, and IEEE Std 802.15.4aa-2022) , vol., no., pp.1-22, 10 Jan. 2023, doi: 10.1109/IEEESTD.2022.10014667.
- [42] Semtech "Reading Channel RSSI During a CAD" Available: <https://www.semtech.com>
- [43] SimPy – Event discrete simulation for Python. <https://simpy.readthedocs.io>.
- [44] C. Bouras, A. Gkamas, S. A. K. Salgado, and V. Kokkinos, "Comparison of lora simulation environments," SpringerLink, https://link.springer.com/chapter/10.1007/978-3-030-33506-9_33 (accessed Dec. 23, 2023).
- [45] Van Rossum G, Drake FL. Python 3 Reference Manual. Scotts Valley, CA: CreateSpace; 2009.
- [46] Ubuntu, "Enterprise open source and linux," Ubuntu, <https://ubuntu.com/> (accessed Dec. 23, 2023).
- [47] ERC Recommendation 70-03: Relating to the use of Short Range Devices (SRD), Sept. 2015. CEPT/ERC/REC 70-03.
- [48] B. Robbe and W. Danny, "A QoS-Aware Adaptive Mobility Handling Approach for LoRa-Based IoT Systems," 2018 IEEE 12th International Conference on Self-Adaptive and Self-Organizing Systems (SASO), Trento, Italy, 2018, pp. 130-139, doi: 10.1109/SASO.2018.00024.
- [49] S. Rappaport et al. Wireless communications: principles and practice, volume 2. Prentice Hall PTR New Jersey, 1996.
- [50] H. Tabassum, M. Salehi and E. Hossain, "Fundamentals of Mobility-Aware Performance Characterization of Cellular Networks: A Tutorial," in IEEE Communications Surveys & Tutorials, vol. 21, no. 3, pp. 2288-2308, thirdquarter 2019, doi: 10.1109/COMST.2019.2907195.

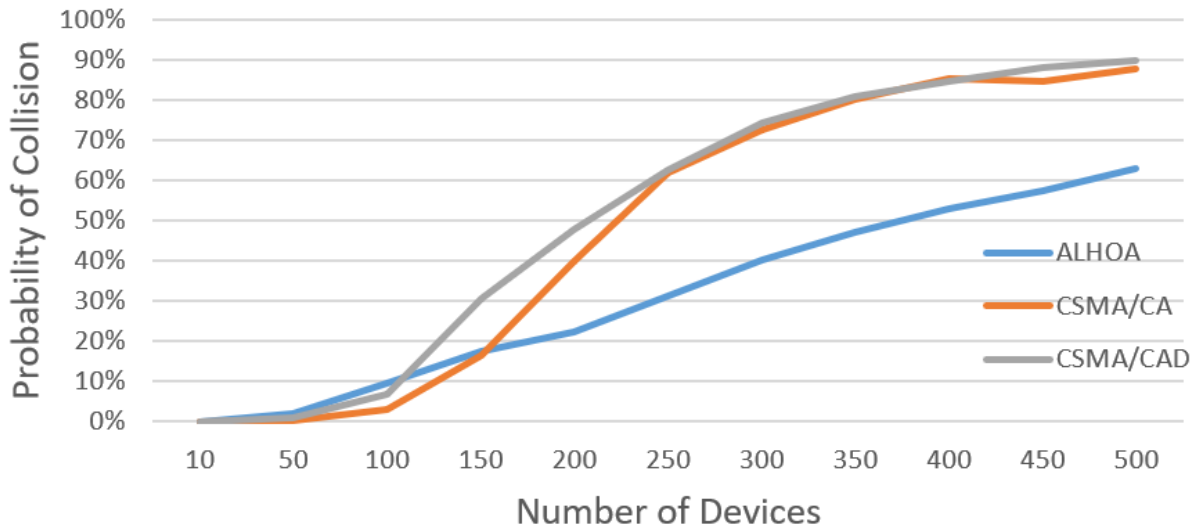
- [51] T. -H. To and A. Duda, "Simulation of LoRa in NS-3: Improving LoRa Performance with CSMA," 2018 IEEE International Conference on Communications (ICC), Kansas City, MO, USA, 2018, pp. 1-7, doi: 10.1109/ICC.2018.8422800.
- [52] K. Mikhaylov, -. Juha Petaejaevaervi and T. Haenninen, "Analysis of Capacity and Scalability of the LoRa Low Power Wide Area Network Technology," European Wireless 2016; 22th European Wireless Conference, Oulu, Finland, 2016, pp. 1-6.
- [53] Semtech, "Low Energy Consumption Design for SX1272/3/6/7/8 LoRa Modem," 2013. [Online]. Available: https://www.semtech.com/uploads/documents/LoraLowEnergyDesign_STD.pdf
- [54] F. Soma, C. Adjih, I. E. Korbi and L. A. Saidane, "A Bayesian model for mobility prediction in wireless sensor networks," *2016 International Conference on Performance Evaluation and Modeling in Wired and Wireless Networks (PEMWN)* , 2016, pp. 1-7, doi: 10.1109/PEMWN.2016.7842898.
- [55] F. Bai and A. Helmy, "A survey of mobility models," in *Wireless Adhoc Networks*, vol. 206. Los Angeles, CA, USA: Univ. Southern California, 2004, p. 147
- [56] A. Einstein, "Investigations on the Theory of the Brownian Movement", Courier Corporat., 1956.
- [57] I. Rhee, M. Shin, S. Hong, K. Lee and S. Chong, "On the Levy-Walk Nature of Human Mobility," *IEEE INFOCOM 2008 - The 27th Conference on Computer Communications* , 2008, pp. 924-932, doi: 10.1109/INFOCOM.2008.145.
- [58] (2005). *Crowdad Trace Dartmouth*. [Online]. Available: <http://crowdad.cs.dartmouth.edu>
- [59] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," in *Wireless Communications and Mobile Computing*, vol. 2, pp. 483-502, 2002.
- [60] E. W. Weisstein. *The CRC Concise Encyclopedia of Mathematics*. CRC Press, 1998
- [61] B. Liang and Z. Haas. Predictive distance-based mobility management for PCS networks. In *Proceedings of the Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, March 1999
- [62] V. Tolety. Load reduction in ad hoc networks using mobile servers. Master's thesis, Colorado School of Mines, 1999.
- [63] MoveBank [online] <https://www.movebank.org/cms/movebank-main>
- [64] J. Yuan, Y. Zheng, C. Zhang, W. Xie, X. Xie, "Guangzhong Sun, and Yan Huang. T-drive: driving directions based on taxi trajectories". In *Proceedings of the 18th*

SIGSPATIAL International Conference on Advances in Geographic Information Systems, GIS '10, pages 99-108, New York, NY, USA, 2010. ACM.

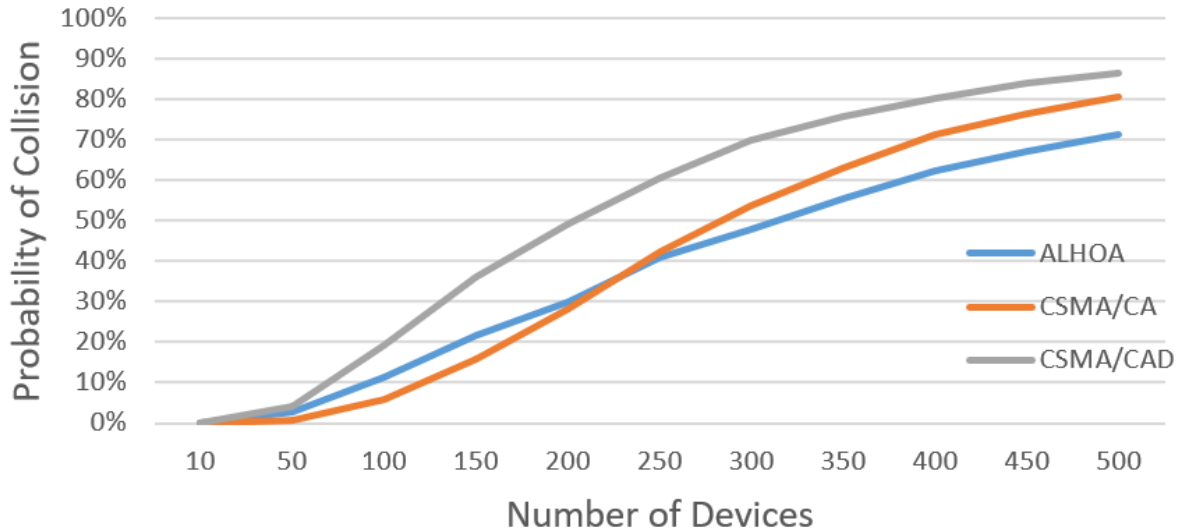
- [65] J Yuan, Y Zheng, X Xie, and G Sun. “Driving with knowledge from the physical world”. In The 17th ACM SIGKDD international conference on Knowledge Discovery and Data mining, KDD'11, New York, NY, USA, 2011. ACM.
- [66] MoveBank: ABoVE: Boutin Alberta Moose [online] https://www.movebank.org/cms/webapp?gwt_fragment=page=studies,path=study302664172
- [67] H. Bohm, E. Neilson, C. de la Mare, and S. Boutin, “Wildlife habitat effectiveness and connectivity: moose ecology project summary report 2010–2012: Final report,” 41 p., 2014.
- [68] Pandas [online] <https://pandas.pydata.org/docs/index.html>
- [69] R. W. Sinnott, "Virtues of the Haversine", *Sky and Telescope* **68** (2), 159 (1984).

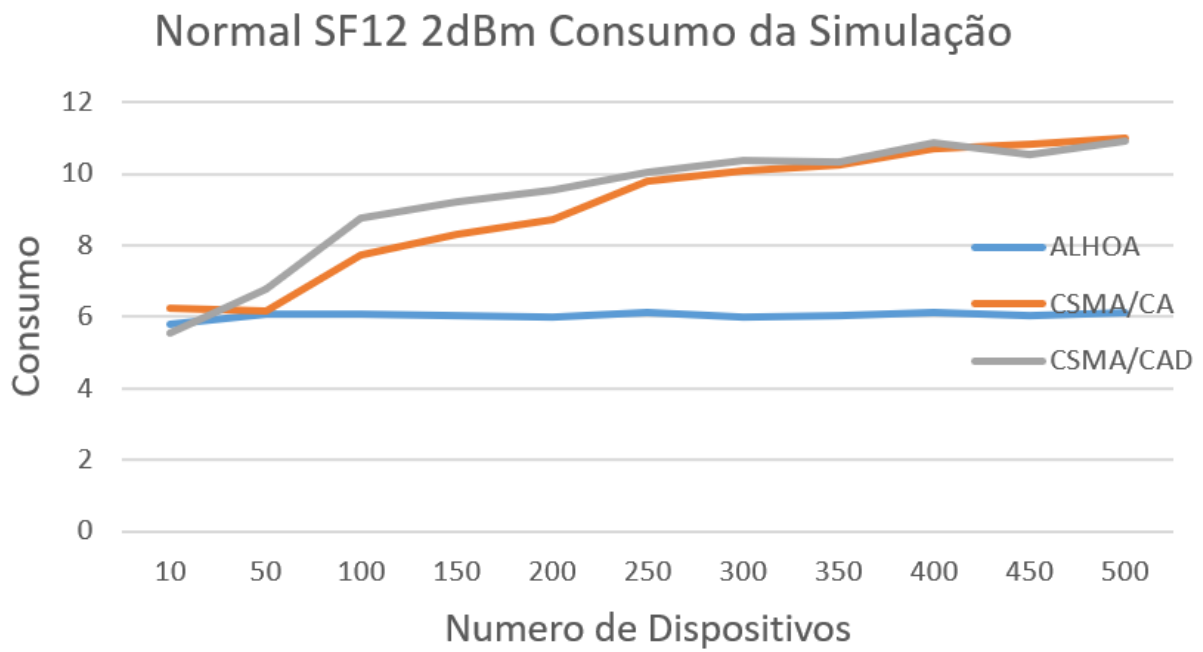
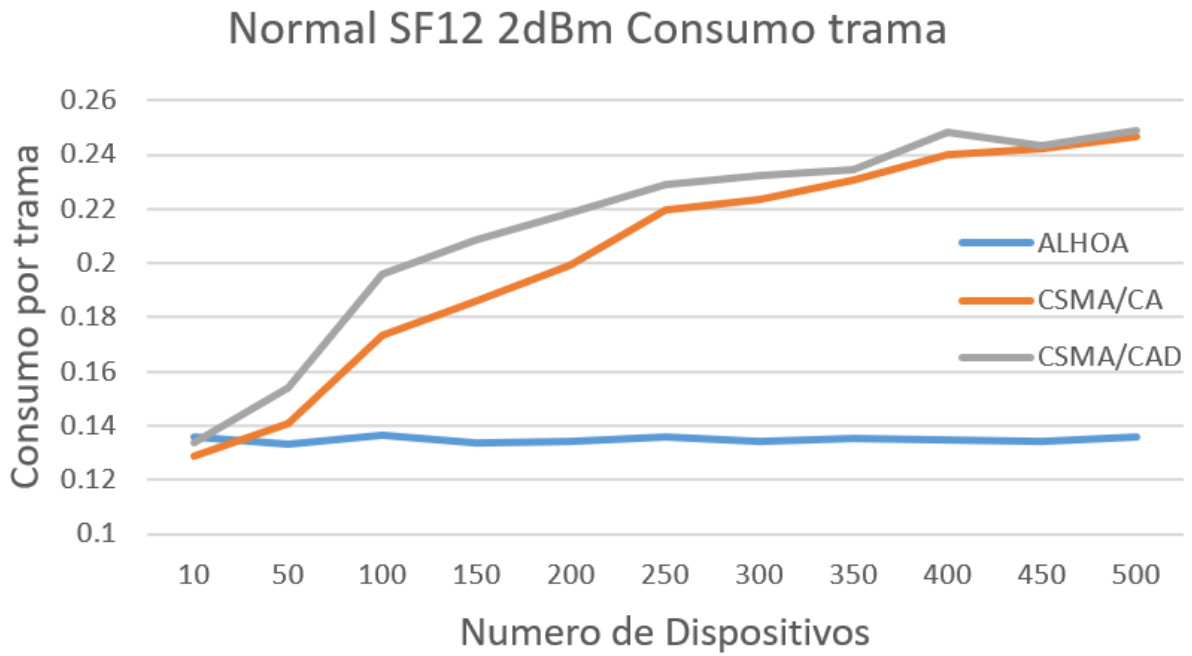
10. Anexos

Random Walk SF7 14dBm Probabilidade de Colisão

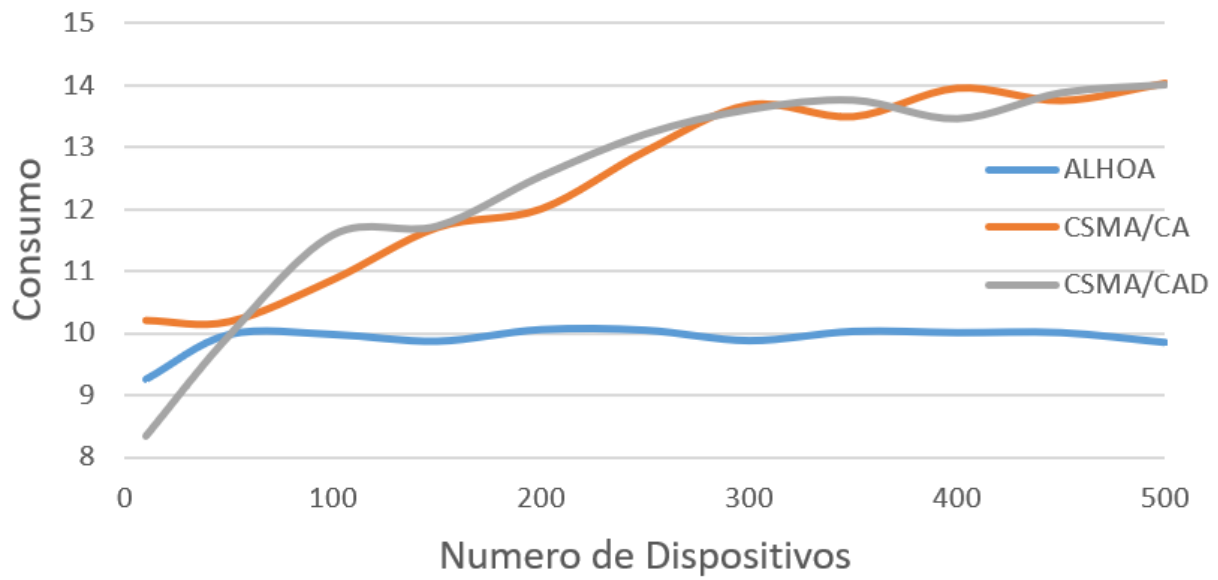


Normal SF12 2dBm Probabilidade de Colisão

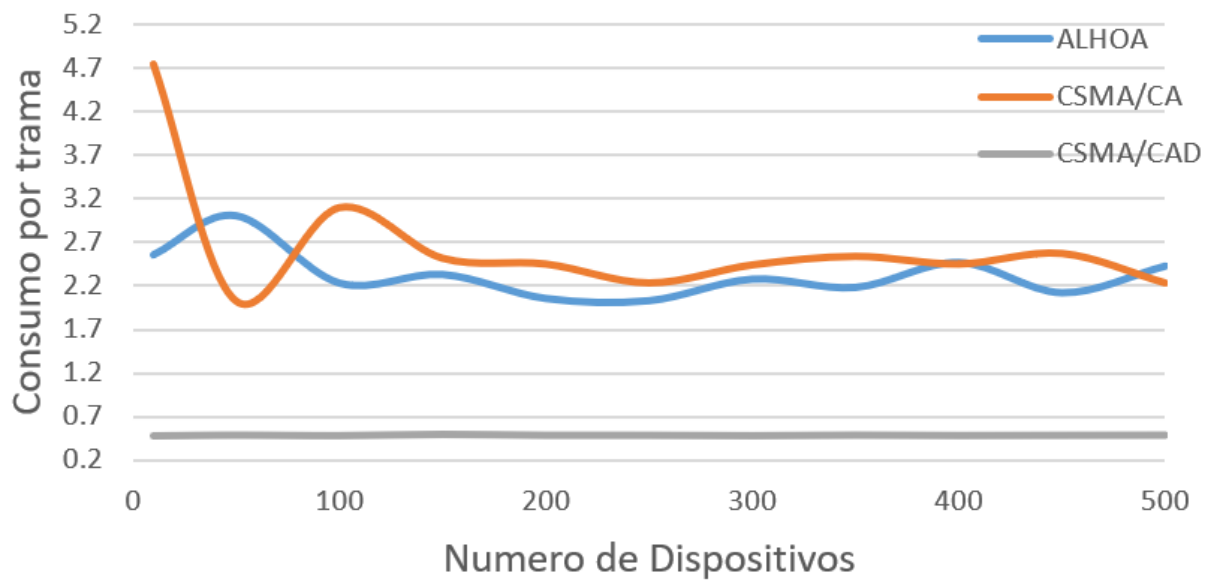




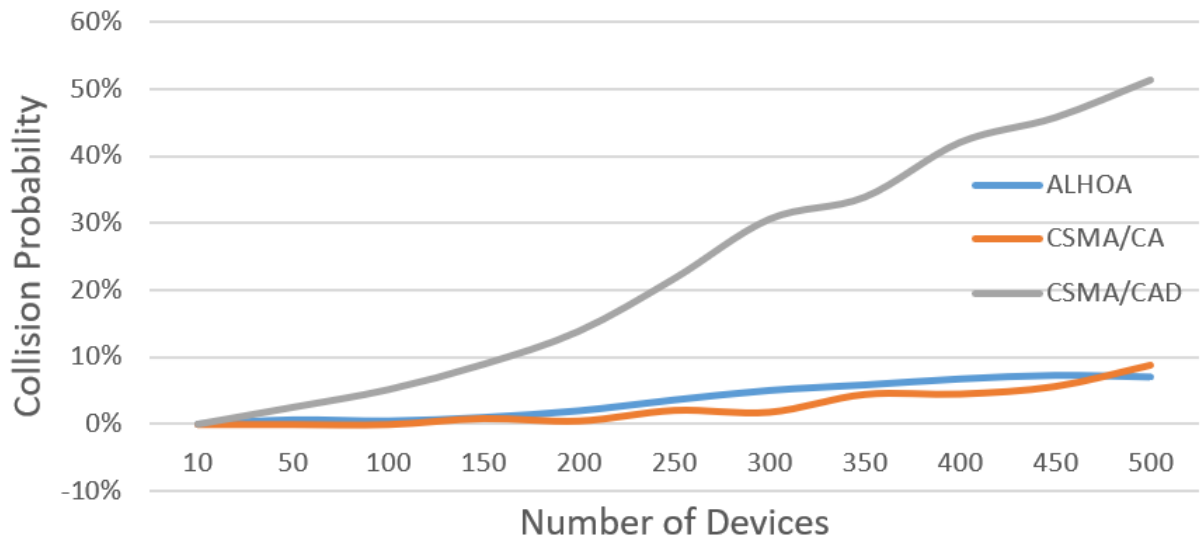
Gauss-Markov SF12 Consumo da Simulação



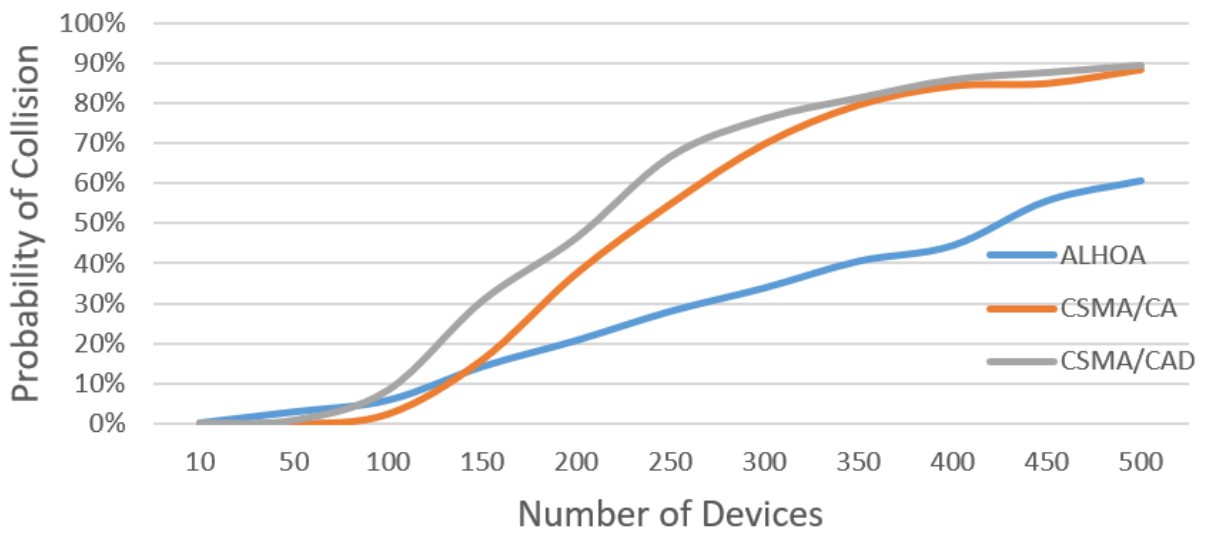
Gauss-Markov SF12 Consumo por Trama



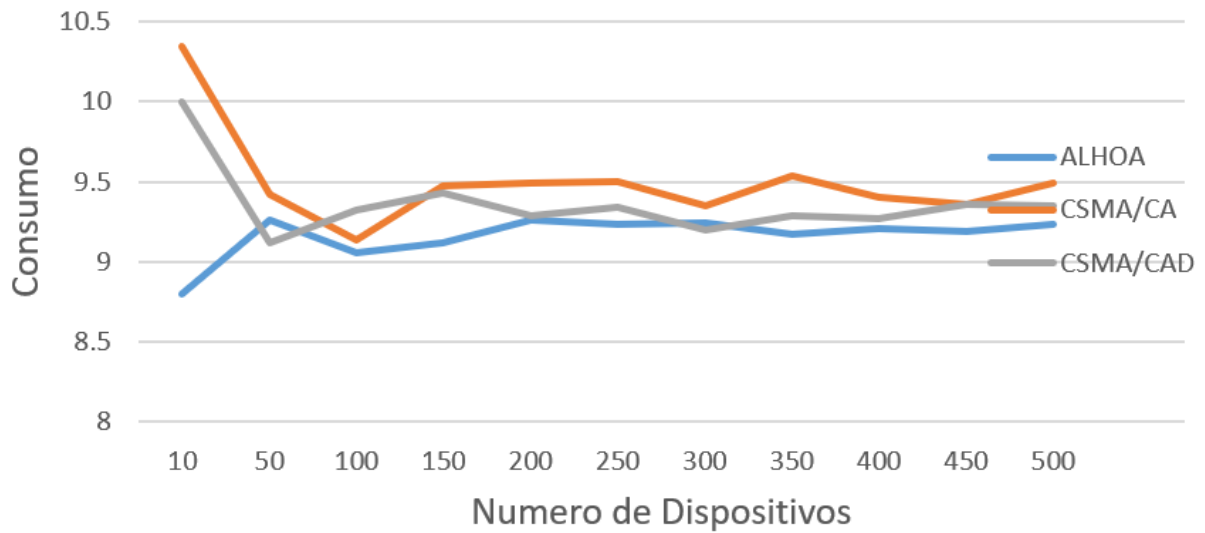
Random Walk SF12 14dBm Probabilidade de Colisão



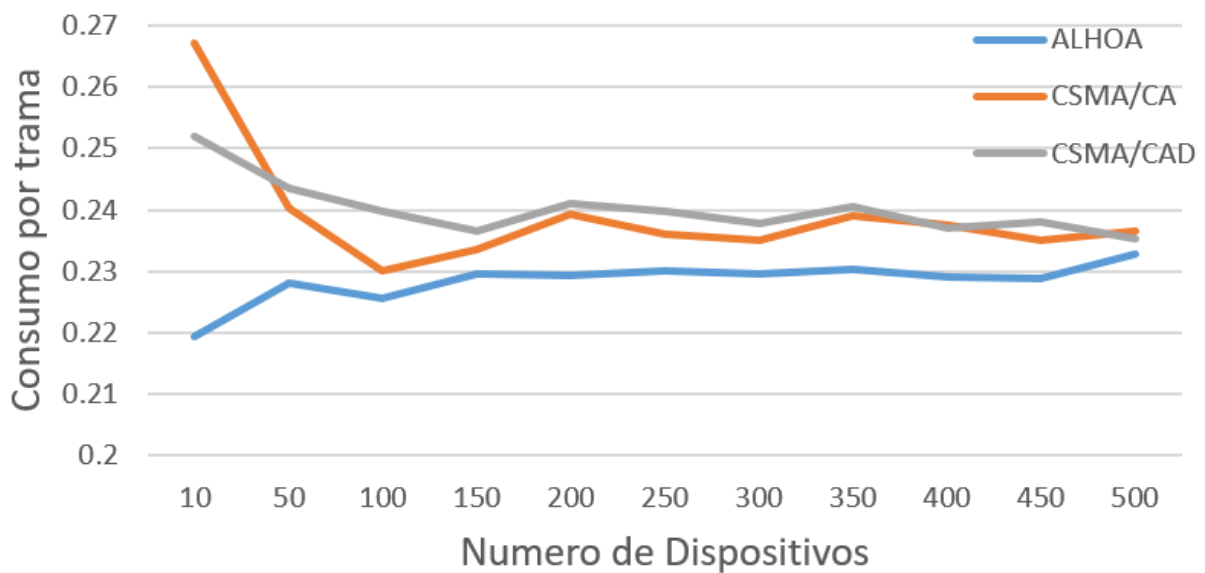
Gauss-Markov SF7 14dBm Probabilidade de Colisão

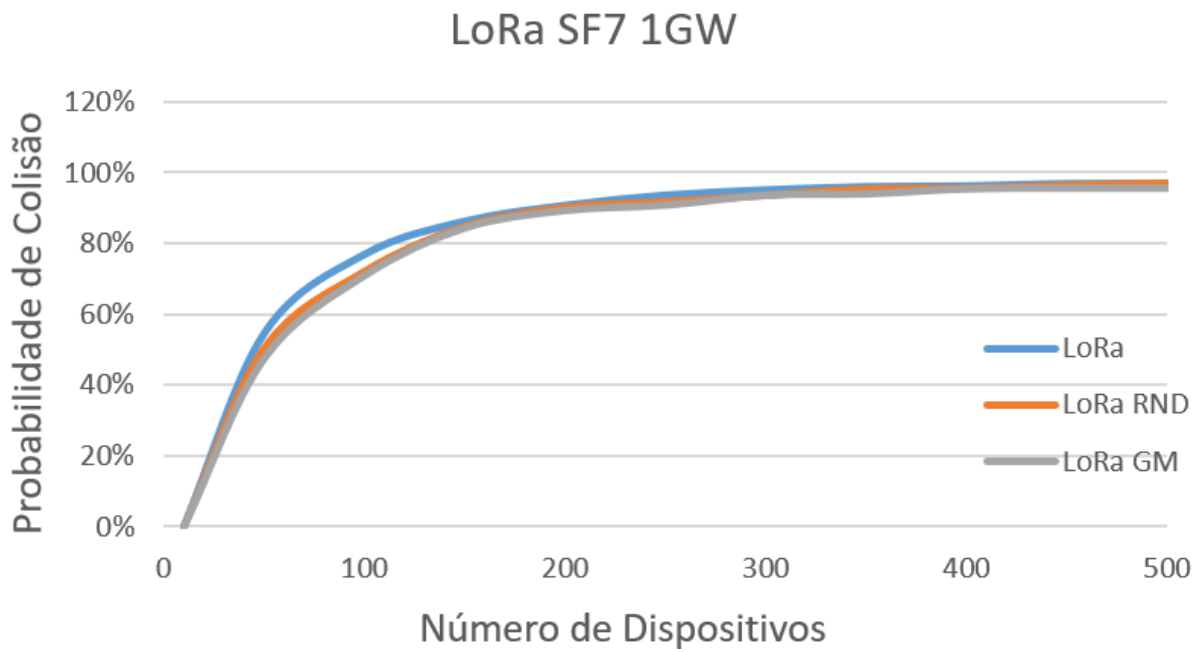
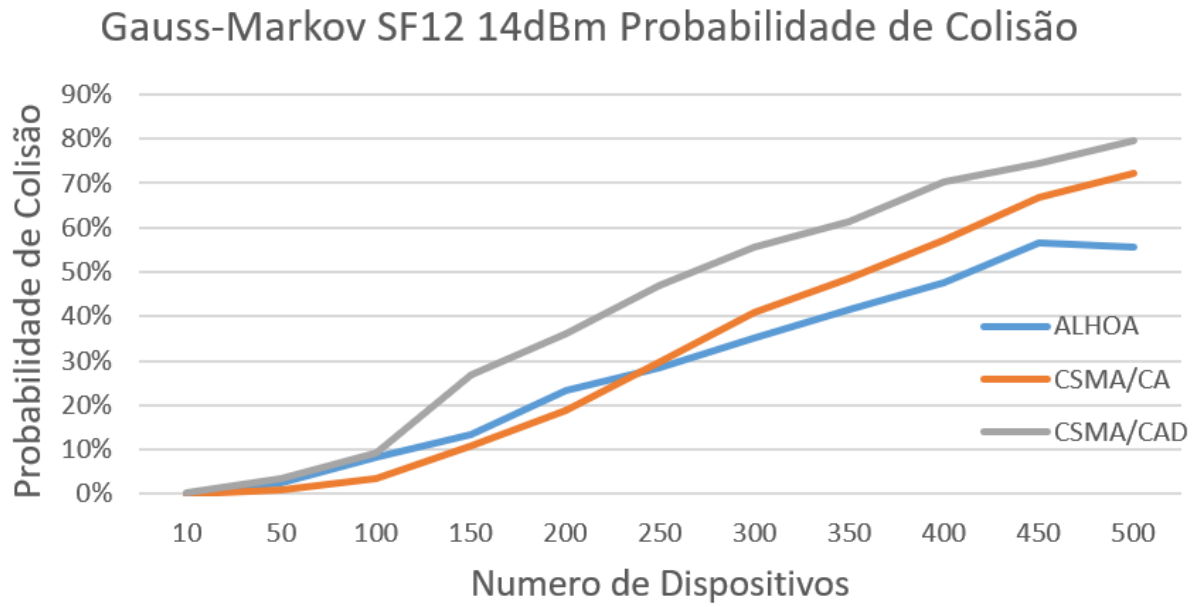


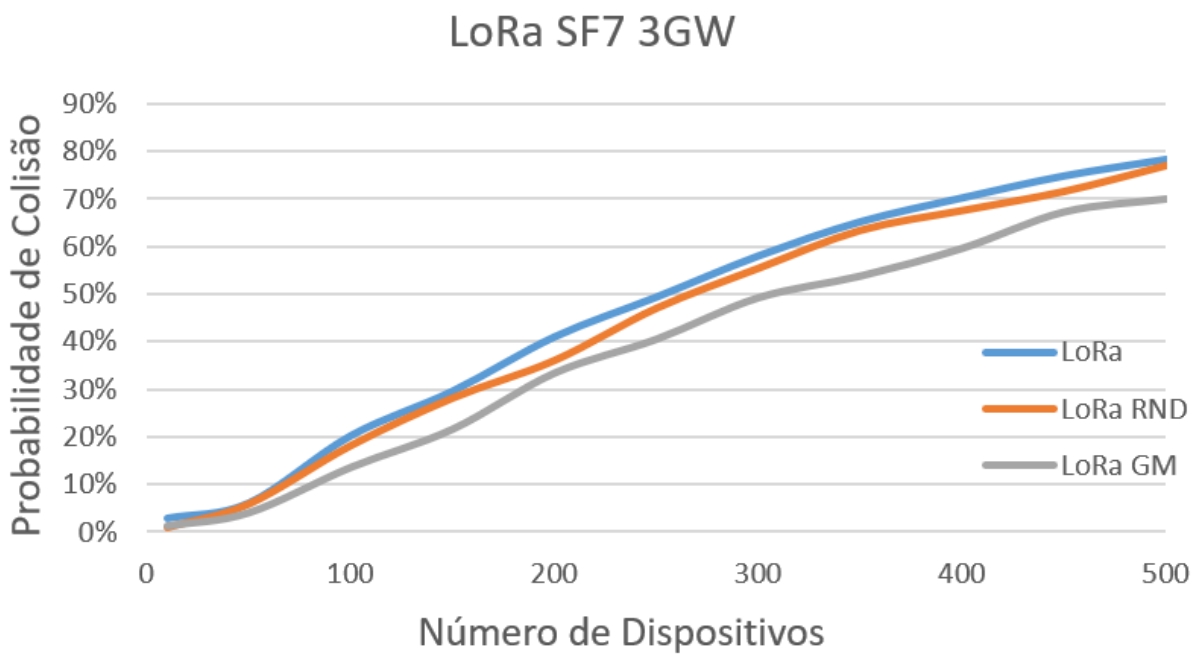
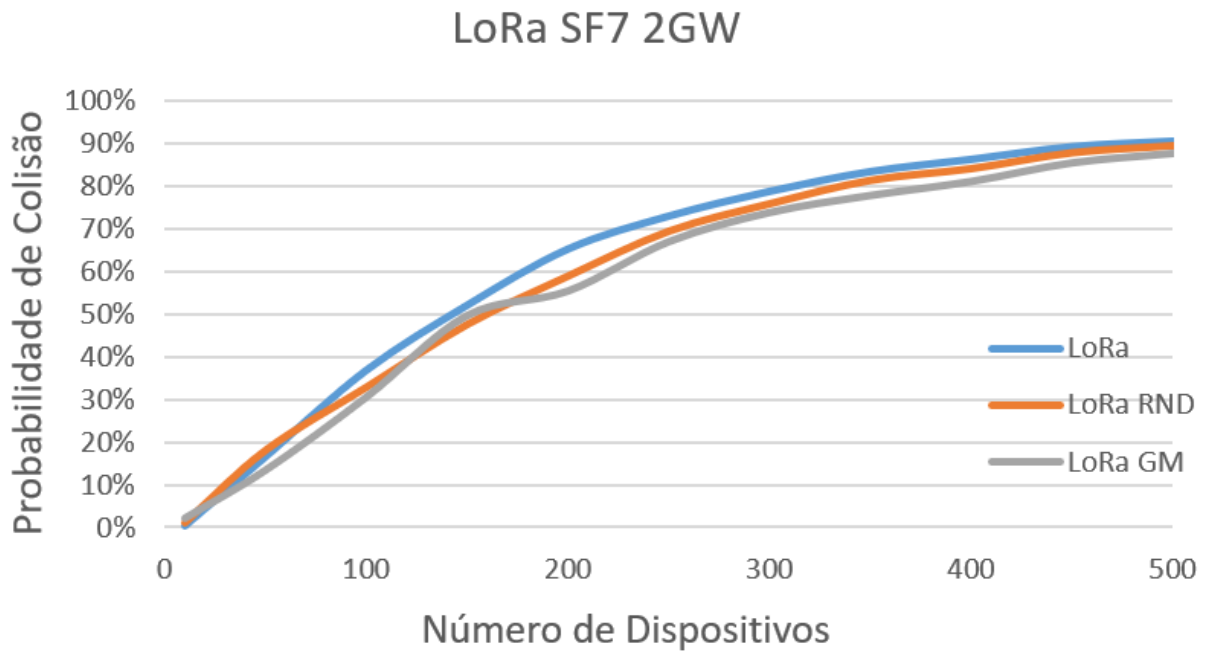
Gauss-Markov SF7 14dBm Consumo da Simulação

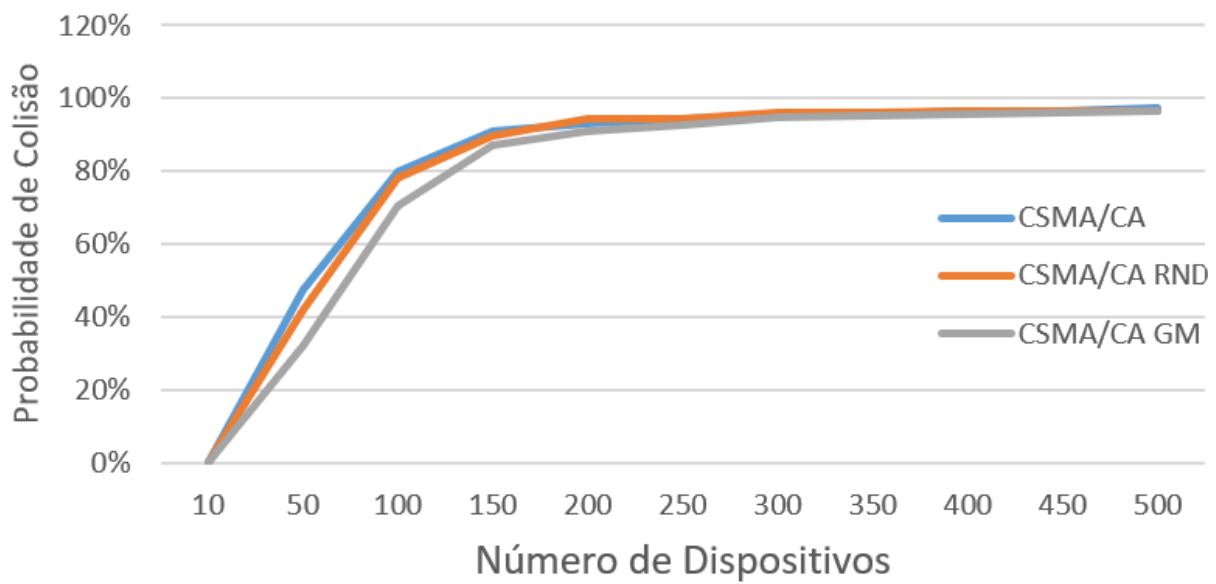
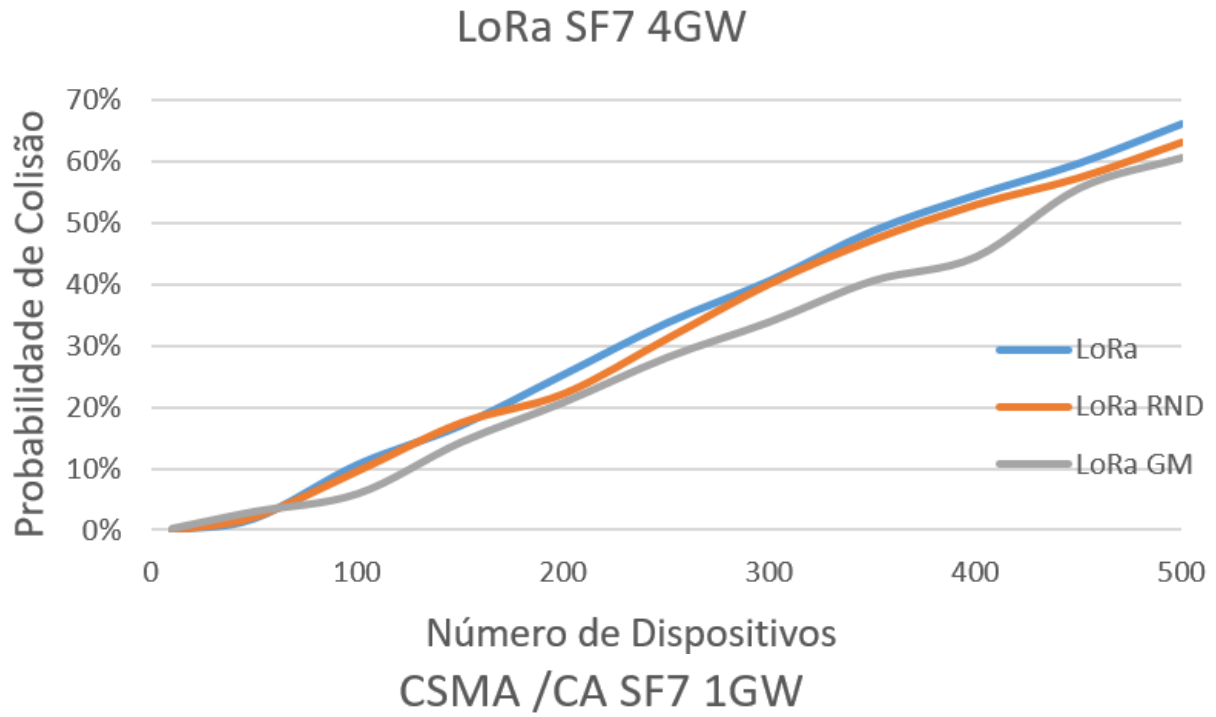


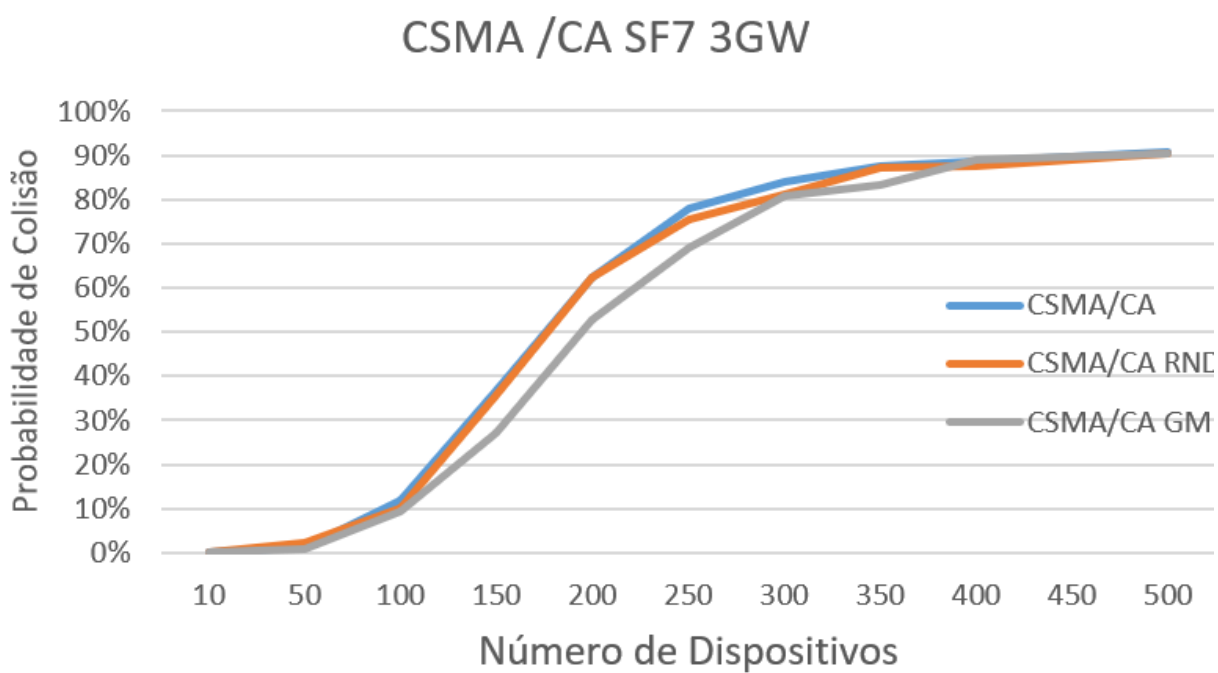
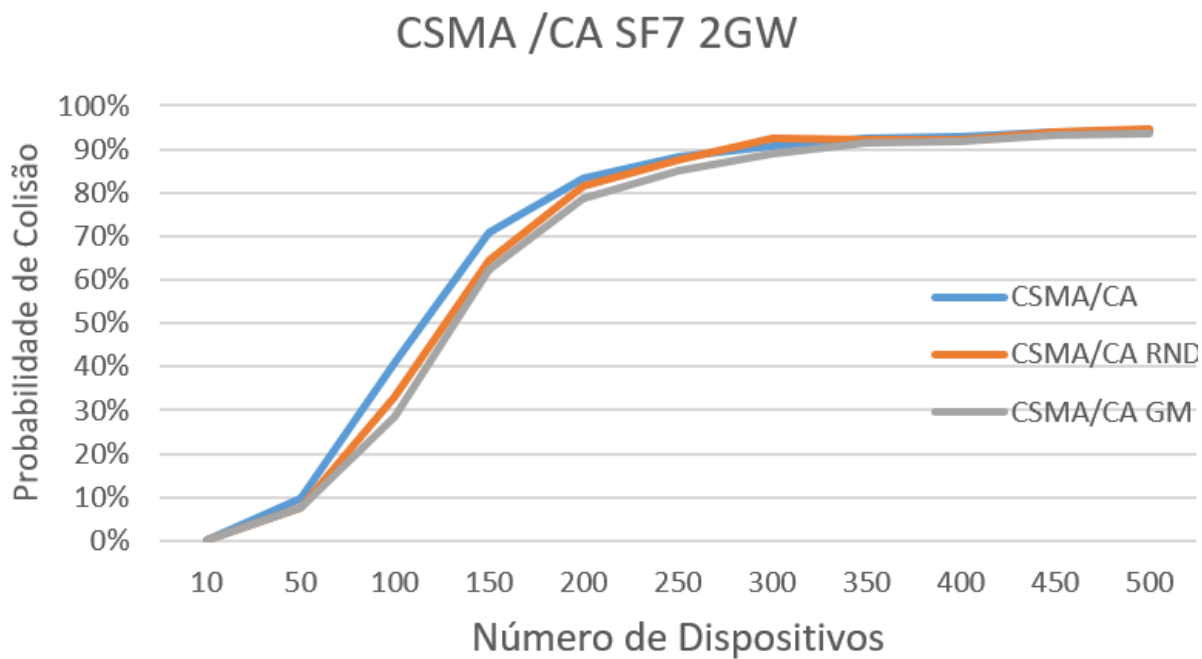
Gauss-Markov SF12 14dBm Consumo trama



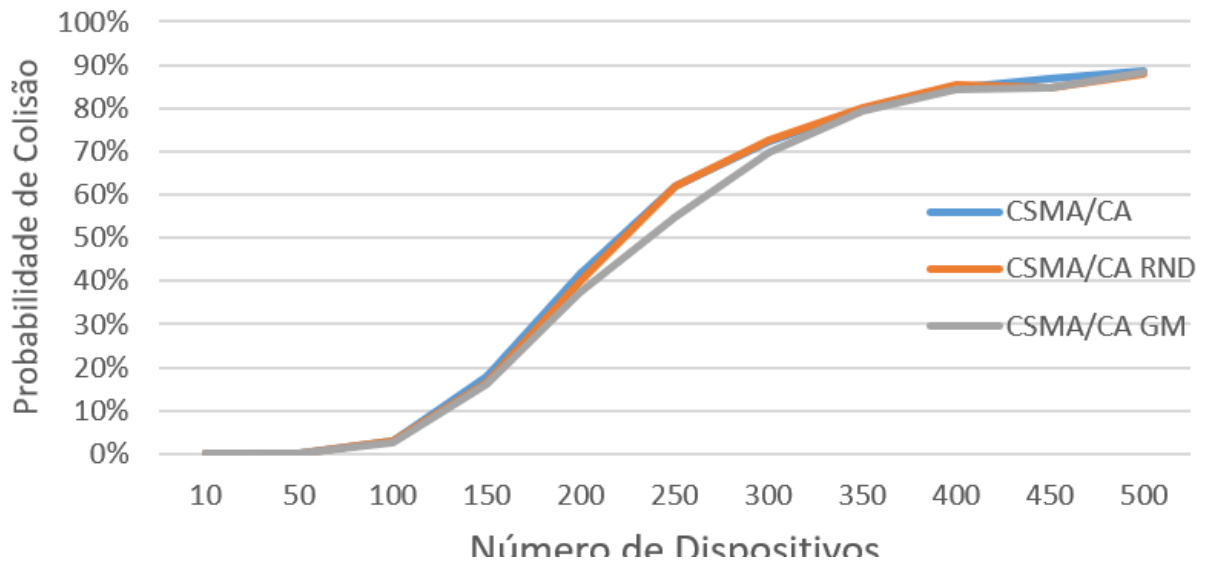








CSMA /CA SF7 4GW



CSMA/CAD SF7 1GW

