

UNIVERSIDADE FEDERAL DO ABC
CENTRO DE ENGENHARIA, MODELAGEM E CIÊNCIAS SOCIAIS APLICADAS
ENGENHARIA DE INFORMAÇÃO

ANDRÉ LUIZ SILVA CAVALHEIRO

REDE LPWAN: PROPRIEDADES E TESTES EXPERIMENTAIS DA
TECNOLOGIA LORA

SANTO ANDRÉ

2018

UNIVERSIDADE FEDERAL DO ABC
CENTRO DE ENGENHARIA, MODELAGEM E CIÊNCIAS SOCIAIS APLICADAS
ENGENHARIA DE INFORMAÇÃO

ANDRÉ LUIZ SILVA CAVALHEIRO

**REDE LPWAN: PROPRIEDADES E TESTES EXPERIMENTAIS DA
TECNOLOGIA LORA**

Monografia apresentada como relatório
final da disciplina Trabalho de Graduação
3 (TG 3).

Orientador: Prof. Dr. João Henrique
Kleinschmidt

SANTO ANDRÉ

2018

AGRADECIMENTOS

Agradeço àqueles que contribuíram de uma maneira direta e indireta para que este trabalho fosse realizado. Aos colaboradores, técnicos, corpo docente e discentes, o meu sincero agradecimento.

Agradeço em especial ao professor Dr. João Henrique Kleinschmidt pela paciência e atenção durante todo o desenvolvimento desse trabalho.

Agradeço aos meu pai Wilson por todo apoio e compreensão durante a graduação, minha namorada Cíntia pelo incentivo em todos os momentos. Minha falecida mãe, Mara, por estar sempre comigo.

RESUMO

O aumento de dispositivos conectados à Internet torna a Internet das Coisas (IoT) um dos assuntos mais estudados sobre conectividade. Alguns paradigmas ainda são analisados, entre eles a melhor conectividade para esses equipamentos. Este trabalho de graduação estuda o padrão de rede LPWAN, a partir de uma comparação teórica de tecnologias feitas a partir desse padrão. A partir desse estudo, foi escolhido os dispositivos com tecnologia LoRa para obtenção de dados práticos, analisando seu comportamento em diferentes posições, alterando os parâmetros para que a comunicação possuísse diferentes características, as quais foram estudadas em seguida. Foi possível observar o funcionamento de uma rede própria para Internet das Coisas e desenvolver uma rede com funcionamento apropriado, analisando diferentes comportamentos que podem ser necessários para se montar uma rede de comunicação com o perfil necessário.

Palavras-chave: Internet das Coisas, IoT, LPWAN, LoRa, Arduino, comunicação.

SUMÁRIO

1 INTRODUÇÃO	3
1.1 OBJETIVOS.....	4
2 LPWAN.....	5
2.1 LONGO ALCANCE (COBERTURA)	5
2.1.1 Uso de banda não licenciada abaixo de 1 GHz	5
2.1.2 Técnicas de modulação	6
2.2 OPERAÇÃO DE BAIXA POTÊNCIA.....	6
2.3 BAIXO CUSTO	7
2.4 OUTROS FATORES DA REDE LPWAN	7
2.5 SISTEMA LORA	8
2.5.1 Camada Física LoRa	9
2.5.2 Rede LoRaWAN	10
2.5.3 Spreading Factor	11
2.5.4 <i>Coding Rate</i> e Largura de Banda	11
3 DESCRIÇÃO EXPERIMENTAL	13
3.1 EQUIPAMENTO CENTRAL.....	16
3.2 EQUIPAMENTO REMOTO.....	17
3.2.1 Arduíno Uno.....	18
3.2.2 Shield LoRa “Dragino”	19
3.2.3 DHT 11	20
3.3 PROGRAMAÇÃO DO ARDUÍNO	21
3.3.1 RSSI	21
3.4 RTT	22
3.5 TESTE ADICIONAL PARA MÁXIMA DISTÂNCIA DO EQUIPAMENTO.....	22
4 RESULTADOS E DISCUSSÕES	23
4.1 TEMPO DE ENVIO E RECEBIMENTO DAS MENSAGENS	25
4.2 POTÊNCIA.....	29
4.3 MENSAGENS PERDIDAS.....	31
4.4 TESTE ADICIONAL PARA DISTÂNCIA MÁXIMA	35
5 CONCLUSÃO.....	36
REFERÊNCIAS.....	38
ANEXOS	41

1 INTRODUÇÃO

A Internet das Coisas (IoT), é uma rede de dispositivos conectados, utilizando sensores e atuadores, equipados com uma interface de telecomunicações, além de unidades de processamento e armazenamento [1]. Essa forma de comunicação permite a integração com qualquer objeto que esteja conectado, permitindo novas formas de interações entre seres humanos e dispositivos, ou diretamente entre dispositivos, o que é comumente referido como comunicação *Machine to Machine* (M2M) [1].

Existem muitas opções de comunicação para a Internet das Coisas, as quais envolvem todas as camadas da pilha de protocolos, desde a transmissão física até a composição da aplicação [1]. De forma geral, todos esses dispositivos finais recebem informações por comunicação *wireless*.

Algumas tecnologias já existentes na época se apresentaram como primeiras opções para implementar serviços *IoT*, era o caso do *Bluetooth* e do *ZigBee*, por exemplo. Eles são caracterizados por consumo de energia muito baixo, o que é fundamental como requisito para dispositivos IoT [1]. Essas tecnologias, entretanto, não possuem todas as características necessárias para o uso de IoT, por possuírem um alcance abaixo do desejado para essas aplicações. Outras tecnologias, por exemplo, que possuam conexões de múltiplos saltos, apresentam maiores dificuldades de implementação para contemplar as necessidades de IoT. Uma rede *IoT* pode possuir cenários de diversos dispositivos espalhados pela cidade para monitoração de recursos, como rede elétrica, saneamento básico e sistema hidráulico, nesses casos, essa distribuição dificultaria a utilização de conexões de múltiplos saltos.

A partir disso, tecnologias de grande cobertura, como redes celulares, podem cobrir esse ponto. Entretanto, as novas tecnologias de redes celulares como LTE (*Long Term Evolution*) não foram elaboradas para suportar muitas máquinas se comunicando. Essa diferença aparece porque, diferentemente de serviços tradicionais de banda larga, a comunicação IoT prevê a transmissão de pacotes pequenos em curtos intervalos de tempo. Além disso, a centralização desses pacotes nas torres celulares pode criar problemas de sinalização e controle de tráfego [1].

Assim, a solução alternativa, entre as tecnologias já existentes que foram mencionados, foi o desenvolvimento de um novo padrão de rede. O padrão LPWAN, acrônimo para *Low Power Wide Area Network*, foi feito para suprir essa necessidade, utilizando baixa potência para a comunicação feita sem fio, cobrindo grandes distâncias. Essa combinação é possível utilizando larguras de banda estreitas, o que, de forma geral, não atrapalha a aplicação, já que usualmente, em Internet das Coisas, os sensores utilizados não necessitam de alta taxa de informação.

Uma das tecnologias desse padrão é o *LoRa*, uma tecnologia de camada física, atualmente mantida pela *Lora Alliance* [1], que permite conectividade sem fio para longas distâncias (aproximadamente até 10 km em áreas rurais, utilizando espectros de frequência não-licenciados (em torno de 900 MHz, no padrão americano, sendo o mesmo para o Brasil, e 868 MHz no padrão europeu [1]) permitindo bandas de até 250 KHz, dependendo da frequência utilizada.

Para a montagem de uma rede LoRa é necessário apenas a compra dos equipamentos, enquanto outras tecnologias LPWAN, como SigFox, funcionam num modelo de serviços, dificultando possíveis testes como os que serão abordados nesse trabalho. Pelo recente avanço da Internet das Coisas, ainda não é possível aferir os reais limites de distância e disponibilidade dessas tecnologias, o que será investigado a seguir, a tecnologia ainda permite a configuração dos parâmetros de configuração da comunicação, o que permitirá observar como se comporta uma comunicação montada a partir de equipamentos LoRa.

1.1 OBJETIVOS

Levando em consideração a ascensão das tecnologias citadas acima, nesse trabalho será apresentado as características que definem o padrão LPWAN, em seguida um comparativo entre diferentes tecnologias que utilizam esse padrão. A partir da escolha dos dispositivos LoRa, será feito um estudo de caso para a partir de seu uso com diferentes parâmetros seja possível verificar o comportamento da rede projetada. Para isso será necessário utilizar um par (*gateway / device*) de equipamentos LoRa em conjunto com uma placa Arduino.

2 LPWAN

As redes LPWAN oferecem conectividade com baixa potência para um grande número de dispositivos distribuídos em uma larga escala geográfica com baixo custo. Algumas tecnologias, como ZigBee, Bluetooth e LTE são utilizadas em equipamentos usados em redes de Internet das Coisas, e apesar de possuir algumas características melhores que dos dispositivos LPWAN, como alta taxa de transferência de dados e baixa latência [2], não possui uma proposta tão eficiente quanto o padrão LPWAN. Pode-se utilizar os seguintes tópicos para se estudar as características similares dos dispositivos LPWAN:

- Longo alcance (cobertura);
- Operação de baixa potência;
- Baixo custo;

2.1 LONGO ALCANCE (COBERTURA)

Para que a cobertura possua o maior alcance possível, utilizou-se frequências baixas. Isso permite que os dispositivos finais se conectem a estações centrais em distâncias abertas de algumas dezenas de quilômetros, dependendo do ambiente (por exemplo, em meios rurais a cobertura será maior). Para isso, usa-se uma banda de frequência não licenciada e técnicas de modulação que permitem a cobertura desejada, como será detalhado em seguida.

2.1.1 Uso de banda não licenciada abaixo de 1 GHz

A maior parte das tecnologias LPWAN utilizam bandas de frequência abaixo de 1 GHz. Estas oferecem robustez e disponibilidade com baixo custo. Ao comparar com a banda não licenciada de 2,4 GHz, a frequência mais baixa permite menor atenuação causado por obstáculos e superfícies densas como paredes de concreto [2]. A faixa de frequência de 900 MHz, é regida no Brasil pela ANATEL, para uso não licenciado como o segmento do espectro no intervalo de 902 a 928 MHz [3]. Essa faixa é menos congestionada que a faixa de 2,4 GHz, utilizada por exemplo

para Wi-Fi e Bluetooth. Isso também auxilia para maior disponibilidade para longas distâncias utilizando baixo consumo energético.

2.1.2 Técnicas de modulação

O cálculo de enlace final de potência (conhecido como *link budget*) para LPWAN é projetado para 150 ± 10 dB, permitindo uma cobertura de 1 à 5 quilômetros para áreas urbanas e dezenas de quilômetros para áreas rurais, até 20 quilômetros [2]. Para a camada física, é necessária alta potência em cada bit (ou símbolo) transmitido. Assim, os receptores (*receivers*) podem decodificar sinais muito atenuados corretamente. Para isso duas técnicas de modulação são adotadas por diferentes tecnologias LPWAN: modulação de faixa estreita (*narrowband modulation*) e espalhamento espectral (*spread spectrum*). O dispositivo LoRa utiliza o *spread spectrum*.

2.2 OPERAÇÃO DE BAIXA POTÊNCIA

Os equipamentos IoT e M2M são projetados para funcionar com alta autonomia, com duração de até 10 anos [2] e que possuam baixo custo de manutenção.

O formato da topologia da rede auxilia que o LPWAN atenda esse requisito. Para topologias de redes *mesh*, como as utilizadas em transmissões de curto alcance, existe um alto custo de implantação para dispositivos distribuídos geograficamente. Além disso, essa topologia tornaria alguns nós mais congestionados que outros, dependendo de sua localização, o que poderia consumir rapidamente a bateria [2].

Sendo assim, as tecnologias LPWAN de modo geral utilizam uma topologia estrela, permitindo considerável economia de energia. Dessa forma, os dispositivos da ponta não precisarão utilizar sua bateria gerenciando o tráfego que passa por ele e que iria para outros dispositivos.

Para a camada de enlace, várias tecnologias LPWAN, como *SigFox* e *LoRaWAN* optaram por utilizar “ALOHA”, um acesso aleatório como o protocolo MAC, no qual cada dispositivo transmite sem fazer qualquer detecção da portadora.

Dessa forma, é possível projetar o transceptor de forma simples e com baixo custo [2].

2.3 BAIXO CUSTO

O custo dos dispositivos finais LPWAN são na escala de unidade ou no máximo dezena de dólares tanto para o equipamento quanto para a conexão [2]. Esse é um dos principais fatores que permite uma grande variedade de aplicações IoT.

Comparado com as redes de curto alcance e celular, os dispositivos LPWAN possuem uma redução na complexidade, por causa dos fatores já abordados como a complexidade na camada de enlace. A infraestrutura necessária é mínima, sendo necessário apenas um ponto central cobrindo extensas áreas, na ordem de dezenas de quilômetros. Além disso, a utilização de bandas de frequência não licenciadas ajuda a manter o baixo custo de todo o sistema.

2.4 OUTROS FATORES DA REDE LPWAN

Alguns outros fatores também podem ser medidos ao se pensar num padrão de rede. A escalabilidade de sistemas LPWAN e a qualidade de serviço (QoS) são fatores que podem ser definidos, mas no caso desse padrão, os dois assuntos não possuem uma definição para o padrão. A escalabilidade possui uma relação inversa com a complexidade já exposta como uma das características que permite o baixo custo dos equipamentos e da comunicação. Assim, ainda não é definido de que forma uma rede LPWAN deve se comportar para alta escalabilidade.

Para o QoS, as aplicações de *IoT* não possuem uma necessidade única de disponibilidade, já que há aplicações, como alarmes, que precisam de velocidade na troca, enquanto outras, como monitoramento de recursos, prezam mais pela precisão. Sendo então este outro fator que fica aberto para o padrão LPWAN.

2.5 SISTEMA LORA

Algumas tecnologias proprietárias aparecem como emergentes na utilização de LPWAN para IoT. A tecnologia escolhida para o trabalho em questão foi a LoRa (acrônimo para *Long Range*). Outras tecnologias LPWAN como SigFox, Ingenu e Telensa, poderiam ser utilizadas para o mesmo objetivo. Na Tabela 1 abaixo é possível observar uma comparação entre as tecnologias citadas. Observe que as tecnologias são similares, funcionando em padrão não licenciado para todos os casos.

O SigFox usa uma tecnologia proprietária e não vende equipamentos separados a sua rede de conexão, o mesmo acontece para a rede Ingenu, que utiliza o padrão *RPMA (Random Phase Multiple Access)*, desenvolvido pela própria empresa. Já a Telensa oferece ao mercado as aplicações, isto é, o serviço fim-afim, como por exemplo soluções de estacionamento [4], também dificultando o estudo da tecnologia.

Existem outras tecnologias que também poderiam ser estudadas, como NB-IoT e Weighless. No caso do NB-IoT (*Narrow Band IoT*), que possui um padrão aberto de tecnologia desenvolvida pela 3GPP [5], o diferencial está na infraestrutura utilizada, no caso redes 4G/LTE. Já o Weightless é uma família de tecnologias abertas de LPWAN, sendo no momento o mais relevante o Weightless-P, que garante comunicação bidirecional de até 100kbps, com cobertura de até 2km em regiões urbanas. Entretanto, esta tecnologia ainda não possui muitos equipamentos disponíveis [5], o que dificultaria os testes que serão feitos neste trabalho.

A escolha leva em consideração a ascensão da tecnologia e do padrão LoRa, o acesso aos equipamentos que possibilitem a utilização da rede, como por exemplo a montagem em Arduíno, que será comentado em seguida. A escolha permitirá observar as características de utilização de um dos diversos padrões em voga para IoT.

LoRa é uma tecnologia de camada física para rede LPWAN, que utiliza a banda de frequência de 900 MHz, desenvolvido e fabricado pela Semtech Corporation [7]. A “LoRa Alliance”, um consórcio das empresas IBM, Actility, Semtech e Microchip, produzem o restante do protocolo, este por sua vez é um protocolo aberto e desenvolvido por esse consórcio.

Tabela 1 – Comparativo entre as tecnologias LPWAN (adaptado de [2])

	SigFox	LoRaWAN	Ingenu	Telensa
Frequência	Europa: 868 MHz EUA: 902 MHz	Europa: 868 MHz / 433 MHz EUA: 915 MHz Asia: 430 MHz	Padrão não licenciado de 2,4 GHz (ISM)	Europa: 868 MHz EUA: 915 MHz Asia: 430 MHz
Taxa de transferência de dados	Upload: 100 bps Download: 600 bps	0,3 – 37,5 kbps (simétrico)	Upload: 78 kbps Download: 19,5 kbps	Upload: 62,5 bps Download: 500 bps
Alcance	Urbano: 10 km Rural: 50 km	Urbano: 5km Rural: 15 km	Urbano: 15 km	Urbano: 1 km
Controle de acesso ao meio	ALOHA	ALOHA	Similar a CDMA	Sem informações
Tamanho de payload	Upload: 12B Download: 8B	Até 250 B	10 KB	Sem informações

2.5.1 Camada Física LoRa

A camada física, chamada de “LoRa PHY”, utiliza uma técnica proprietária de modulação de espalhamento espectral chirp (*chirp spread spectrum - CSS*). De forma resumida, a técnica CSS consiste num sistema no qual a portadora “varre” linearmente uma elevada gama de frequências a determinada taxa. Através da propriedade física na qual as frequências diferentes sofrem atrasos distinguíveis, o receptor comprime a energia do sinal num único impulso [8].

Essa técnica possibilita a sincronização de tempo e frequência, sem necessitar de componentes de alto custo [1]. Também possibilita a utilização de vários canais simultaneamente, a taxa de dados é limitada por regulações de ciclo de trabalho e potência de transmissão. Dependendo da distância, a taxa de transmissão varia de 300 bps à 37,5 kbps, também dependendo de fatores de espalhamento e largura de banda do canal. A distância pode chegar a 5 km em regiões urbanas e 15 km em regiões rurais. [2]

2.5.2 Rede LoRaWAN

A topologia utilizada para uma rede LoRaWAN é chamada de “estrela de estrelas”, no qual os dispositivos finais se conectam a um ou mais *gateways*, estes, por sua vez, se conectam a um servidor comum utilizando protocolo padrão IP [1], como é mostrado na Figura 1. Esses *gateways* são transparentes para os dispositivos finais e toda a complexidade de processamento ficam no servidor central. O padrão permite que os dispositivos finais troquem de *gateway* sem precisar enviar nenhuma sinalização.

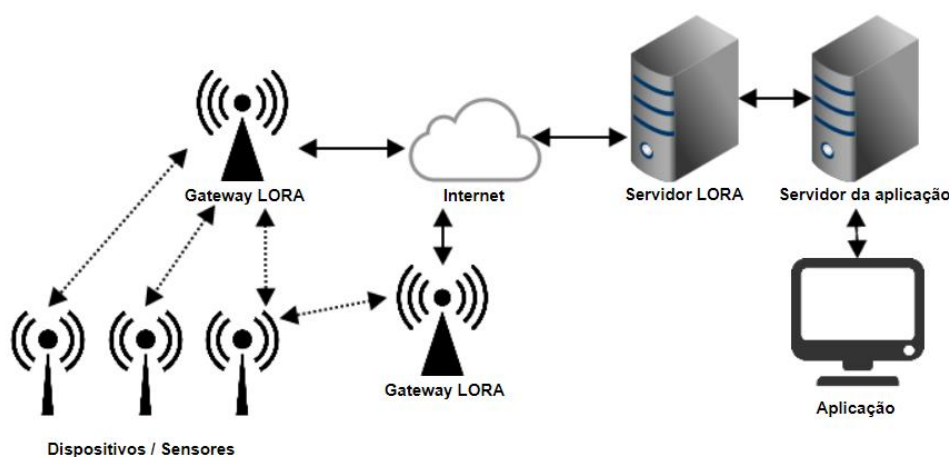


Figura 1 – Topologia Rede LoRa

Os dispositivos finais são divididos em três classes: Classe A (*A de All*), Classe B (*Beacon*) e Classe C (*Continuously listening*) [9]. A Classe A é o padrão para LoRaWAN. Nesta classe, os dispositivos finais sempre iniciam as transmissões de forma assíncrona, depois, o dispositivo final ficará aguardando qualquer comando ou pacote que poderá retornar do servidor. Os dispositivos Classe A são para aplicações de monitoramento, no qual as informações produzidas pelos dispositivos finais serão coletadas por uma base central [1].

A Classe B torna as transmissões de envio e recepção (*uplink* e *downlink*) independentes, com a sincronização ocorrendo no servidor central, atuando em conjunto com o gateway. Este envia pacotes de sincronismo (*beacon packets*) que permite essa troca de informações. Ideal para dispositivos que precisam receber comandos de um controlador remoto, como atuadores [1].

Já a Classe C é para os dispositivos que não possuem restrição de energia, os quais podem permanecer com a janela de recepção de dados sempre aberta, consequentemente com um consumo maior de eletricidade.

2.5.3 Spreading Factor

Algumas características dos equipamentos LoRa são determinantes na comunicação que será realizada entre os dispositivos. Alguns parâmetros que definem essas características serão objetos de estudo dessa monografia, como o *Spreading Factor*, a largura de banda e o *Coding Rate*.

O *Spreading Factor* (SF), fator de espalhamento, está relacionado com a técnica de modulação utilizada, conhecida como *chirp spread spectrum modulation* [16], que usa a mesma técnica utilizada em sistemas militares e de radar, possibilitando longo alcance e grande imunidade a ruídos. Esse protocolo utiliza uma modulação de frequência com amplitude fixa [17], o que permite utilizar todo o espectro alocado para transmitir sinais, produzindo um sinal que varre todo o canal. Assim, o *spreading factor* é a duração do sinal, que utiliza toda a banda alocada, a qual neste caso será a largura de banda do LoRa, entre 125 e 250 kHz.

O fator de espalhamento será, portanto, o “tempo no ar”, sendo 7 o menor tempo e 12 o maior. Em teoria, ao aumentar em 1 o SF o tempo no ar será dobrado para transmitir a mesma quantidade de informação. Assim, a mesma largura de banda resulta em menos informação transmitida por tempo.

2.5.4 Coding Rate e Largura de Banda

Os parâmetros de largura de banda e *coding rate*, ou taxa de codificação, utilizam conceitos mais usuais no estudo de redes de comunicação. A largura de banda possui exatamente o mesmo significado que pode ser observado em qualquer sistema de comunicação, e no caso, é tipicamente 125 ou 250 kHz, e serão os parâmetros utilizados para o estudo a seguir.

O *coding rate* é o parâmetro utilizado que possibilita uma forma de “FEC” (*Forward Error Correction*) utilizado na rede LoRa para correção dos bits transmitidos, permitindo a recuperação da informação que pode ocorrer por interferência [18]. O

fator de *Coding Rate* do LoRa possui os valores: 4/5, 4/6, 4/7, 4/8. O denominador representa quantos bits serão transmitidos no total para possibilitar a correção dos 4 bits transmitidos. A partir desses parâmetros, a taxa nominal de bits transmitidos pode ser definida pela equação a seguir [17], em bits por segundo:

$$Rb = SF * \frac{CR}{\frac{2^{SF}}{BW}} \quad (1)$$

Onde:

SF = spreading fator (7..12);

CR = code rate (4/5 .. 4/8);

BW = largura de banda (125.. 250 kHz)

A definição do tempo de duração do símbolo, T_{sym} , segue a fórmula (2) abaixo:

$$T_{sym} = \frac{2^{SF}}{BW} \quad (2)$$

A estrutura dos pacotes enviados numa rede LoRa pode ser observado na Figura 2 abaixo:

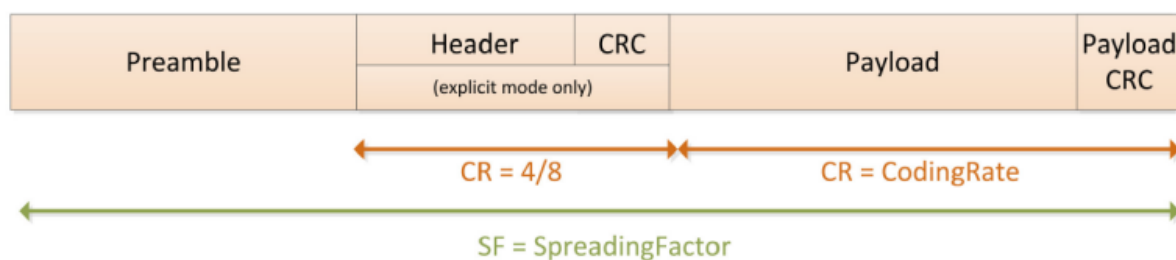


Figura 2 – Estrutura do pacote LoRa [18]

O pacote é formado por um preâmbulo, em seguida o cabeçalho com um CRC – Cyclic Redundancy Check – responsável pela detecção de erro. Em seguida, está o pacote em si, payload, e por fim a detecção de erro dos dados que estão sendo transmitidos.

Os testes desse primeiro cenário mostraram que a região escolhida era adequada por já conter alguns pontos sem disponibilidade, isto é, seria possível analisar a rede alterando suas configurações.

Assim, o segundo cenário montado possui um modelo com apenas 4 pontos, como na Figura 4 abaixo, no qual será possível alterar os parâmetros e testar cada configuração de forma mais detalhada e assertiva.

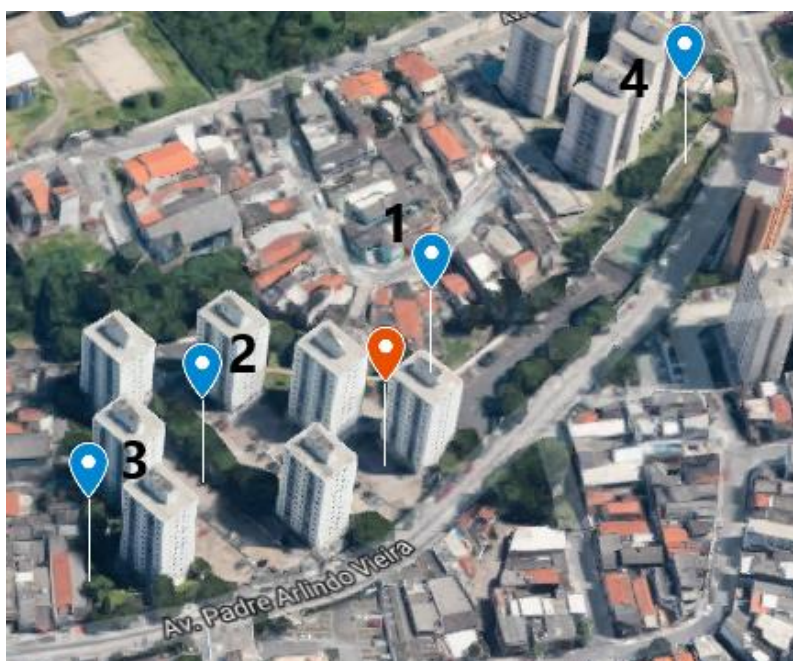


Figura 4 – Pontos escolhidos para análise

Os pontos foram enumerados conforme a distância do ponto laranja, no qual está o gateway. O ponto central (gateway) está em um apartamento, próximo a janela que permite visão dos pontos 2 e 3. O ponto 1 está em frente ao prédio, na face oposta ao ponto central.

As distâncias seguem conforme Tabela 2 abaixo.

Tabela 2 – Distâncias do *gateway* de cada ponto

Ponto	Distância do <i>gateway</i> [m]
1	28 \pm 2
2	54 \pm 2
3	105 \pm 2
4	164 \pm 2

Foram escolhidos a alteração de alguns parâmetros, conforme segue, para se estudar como cada alteração de configuração modifica a comunicação:

Spreading Factor: 7, 9 ou 10;

Largura de Banda: 125 kHz ou 250 kHz

Coding Rate: 4/5 ou 4/8

Alterando os parâmetros acima, para cada uma das 4 posições da Figura 4, foi possível realizar 12 testes, cada teste contendo pelo menos 200 mensagens enviadas para cada uma das posições.

Como pode ser observado no código em anexo, é transmitido os dados do sensor (DHT11) em cada mensagem. Para envio de pelo menos 200 mensagens, foram necessários para cada teste entre 12 e 20 minutos. A Tabela 3 a seguir mostra a configuração de cada teste.

Tabela 3: Configuração de cada teste

Teste	SF	BW	CR
1	7	125	4/5
2	7	125	4/8
3	7	250	4/5
4	7	250	4/8
5	9	125	4/5
6	9	125	4/8
7	9	250	4/5
8	9	250	4/8
9	10	125	4/5
10	10	125	4/8
11	10	250	4/5
12	10	250	4/8

3.1 EQUIPAMENTO CENTRAL

Para o equipamento central foi utilizado um *LG01-P LoRa Gateway* da fabricante “Dragino”. Trata-se de um equipamento *single channel* que utiliza código aberto [19]. Ele permite a conexão entre a rede LoRa e a rede IP, via *WiFi* ou *Ethernet*. Os testes realizados e o equipamento não usam LoRaWAN. Observe a Tabela 4 abaixo com mais algumas especificações.

Tabela 4 – Especificações do *LG01-P LoRa Gateway*

Processador Linux:	400 MHz
Memória Flash:	16 MB
Memória RAM:	64 MB
Microcontrolador:	ATMega328P
Memória Flash:	32 kB
Memória RAM:	2 kB
Chip LoRa:	SX2176/78

Interfaces:	2 portas RJ 45
	WiFi 802.11 b/g/n
	1 porta USB
	Wireless LoRa

Nota-se que temos um processador Linux, este interage diretamente com o micro controlador ATmega328P e com todas as interfaces, exceto o *Wireless LoRa*, que está conectado ao ATmega328P. O micro controlador é o mesmo utilizado no Arduino, sendo possível programá-lo, o que é necessário para a comunicação com o equipamento remoto e para a troca de mensagens da forma desejada. Na Figura 5 pode-se observar as interfaces físicas e a antena do equipamento, além dos LEDs de indicação acima, que marcam se está ligado e qual das conexões está ativa.



Figura 5 – Equipamento “LG01-P LoRa Gateway”

3.2 EQUIPAMENTO REMOTO

O equipamento remoto é responsável pela captação das informações desejadas, no caso temperatura e humidade, e por enviar essa informação para o

equipamento central. Foi utilizado um Arduíno Uno em conjunto com uma *shield LoRa*, e como sensor o DHT11.

3.2.1 Arduíno Uno

O Arduíno é uma plataforma eletrônica de código aberto projetada com um microcontrolador Atmel AVR. É construído em uma placa única, possui suporte de entrada e saída embutido e uma linguagem de programação padrão baseada em C/C++ [10].

Além do microcontrolador, o Arduíno se caracteriza por ter componentes complementares para facilitar a programação e incorporação de outros circuitos. Os conectores são expostos de maneira padronizada, permitindo a interligação do CPU a outros módulos de expansão. Esses módulos são conhecidos como *Shields*. É possível encontrar módulos de diferentes fabricantes de LORA para Arduíno [11][12].

A estrutura do microcontrolador é ideal para conectar e centralizar componentes utilizados em redes de Internet das Coisas, possibilitando a conexão de sensores e dispositivos que fazem a interação com o ambiente. Ele já é utilizado em aplicações em várias áreas como agricultura, impressão 3D e na engenharia de transportes [13][14][15].

O modelo utilizado foi o Uno R3 fabricado pela RobotLinking, baseado no microcontrolador MEGA328P da Atmel. Possui 14 pinos de entrada/ saída digital, possibilitando que 6 destes sejam usados como saída PWM. Possui 6 entradas analógicas, um cristal oscilador de 16 MHz, uma conexão USB e entrada de alimentação. As especificações técnicas põem ser observadas na Tabela 5 abaixo e na Figura 6 observa-se todas as portas disponíveis no Arduíno.

Tabela 5 – Especificações do “Arduíno Uno R3”

Tensão de operação	5V
Tensão de alimentação (recomendada)	7-12 V
Memória Flash	32 KB
Memória SRAM	2 KB
EEPROM	1 KB

Dimensões	68,6mm x 53,4 mm
Peso	Aprox. 150g



Figura 6 – Arduíno Uno R3

3.2.2 Shield LoRa “Dragino”

O módulo transceptor LoRa da Dragino foi montado em conjunto com o Arduíno, como é possível observar na Figura 7 abaixo. O dispositivo é baseado em um chip SX1276 da Semtech, seguindo as características explicadas na sessão 1.2 e com as propriedades observadas na Tabela 6.

Tabela 6 – Características do módulo LoRa com chip SX1276

Frequência utilizada	915 MHz
<i>Spreading Factor</i> (SF)	6 – 12
Largura de banda (BW)	125 – 250 kHz
Taxa de transferência efetiva (R _b)	0.18 – 37.5 kbps
Sensibilidade	-111 a -148 dBm

O fator de espalhamento (“Spreading Factor”) permite alterar a distância da comunicação entre os pontos. Aumentando esse valor, eleva-se o *airtime* do pacote transmitido, aumentando o seu tempo de transmissão. Essa propriedade altera a taxa de transferência como visto na Equação 1.

Os testes realizados permitiram observar as características citadas acima e observar o comportamento da comunicação ao utilizar diferentes variáveis para *Spreading Factor* e largura de banda.



Figura 7 – Módulo LoRa conectado ao Arduíno

3.2.3 DHT 11

O DHT 11 é um sensor de baixo custo que permite medir a umidade e a temperatura, sendo possível utilizar uma saída única para enviar essas informações por uma entrada analógica do Arduíno.

Segue na Tabela 7 algumas características do sensor. Também é possível observá-lo na Figura 8.

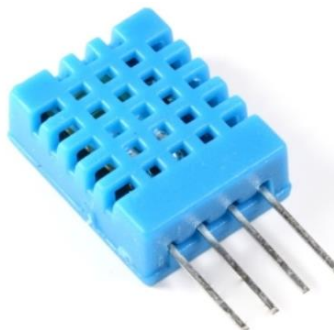


Figura 8 – Sensor DHT 11

Tabela 7 – Características do sensor DHT 11

Alimentação	3,0 a 5,0 VDC
Corrente	200uA a 500mA
Intervalo de umidade	20 a 90% UR +- 5%
Intervalo de temperatura	0 a 50°C +- 2°C
Tempo de resposta	< 5s
Dimensões	23mm x 12 mm x 5mm

3.3 PROGRAMAÇÃO DO ARDUÍNO

Em anexo estão os códigos utilizados tanto no *gateway* quanto no Arduino remoto, além do diagrama elétrico das ligações entre o módulo LoRa e o sensor. Utilizando uma biblioteca própria para comunicação LoRaWAN, chamada *RadioHead*, foi possível elaborar um algoritmo que, simplificada e segue os passos abaixo:

- 1º - Escolhe as variáveis desejadas da rede: Frequência, *Spreading Factor*, largura de banda, *Coding Rate*. *Gateway* e equipamento remoto;
- 2º - Equipamento remoto recebe os dados de temperatura e umidade;
- 3º - Envia esses dados para o *gateway*;
- 4º- O *gateway* recebe essas informações e envia uma resposta. Essa resposta contém uma mensagem de confirmação e o RSSI da comunicação.

Sendo assim, a cada bateria de teste, foi necessário alterar o algoritmo para seguir as variáveis desejadas.

3.3.1 RSSI

O acrônimo RSSI significa *Received Signal Strength Indication* e para os testes a seguir foi o principal parâmetro para a qualidade da rede, por ser a medida da potência do sinal recebido. A biblioteca *RadioHead* permite utilizar essa medida apenas chamando a função *lastRssi*. Quanto mais alto é este valor, maior a potência

do sinal recebido. Como os valores são negativos, significa que quanto mais próximo estiver de zero, é uma indicação de que o sinal recebido é mais potente.

3.4 RTT

Com os equipamentos configurados, seguindo os pontos mostrados na Figura 4, foi possível realizar uma bateria de testes que possibilitou medir o RSSI da comunicação, o número de pacotes enviados e o número de pacotes que o recebimento foi confirmado. A relação entre esses dois últimos dados será considerada como a diferença entre pacotes enviados e recebidos pelo gateway, isto é, assumiu-se que os pacotes não confirmados não tiveram a comunicação bem-sucedida.

Também foi medido o tempo para essa mensagem ser transmitida e foi somado ao tempo da mensagem de confirmação retornar, sendo então o RTT (tempo de ida e volta) da mensagem, conforme equação 3 abaixo.

$$RTT = T_{ida} + T_{processamento} + T_{volta} \quad (3)$$

Para a mensagem de volta é necessário um tempo de processamento do equipamento, por isso o " $T_{processamento}$ " precisa ser considerado.

3.5 TESTE ADICIONAL PARA MÁXIMA DISTÂNCIA DO EQUIPAMENTO

Com o intuito de observar até que distância é possível comunicação em uma rede LoRa, foi feito um teste adicional qualitativo.

Mantendo o *gateway* na mesma posição, observou-se a troca de 10 mensagens. Os locais considerados como último ponto de comunicação manteve entre 2 a 5 mensagens com confirmação de recebimento.

A configuração utilizada para esse teste, detalhada abaixo, foi a que obteve menor percentual de mensagens perdidas ao se efetuar os testes dos cenários anteriores:

- *Spreading Factor* = 10;
- Largura de banda = 125 kHz;
- *Coding Rate* = 4/5.

4 RESULTADOS E DISCUSSÕES

Os resultados, de maneira geral, podem ser observados nas tabelas a seguir: As mensagens perdidas é a quantidade de mensagens que não receberam confirmação em cada teste. Para o RSSI, o valor mostrado é o valor médio sendo a incerteza o desvio padrão encontrado. Todos esses dados foram possíveis pois o algoritmo enviava as informações para a porta serial, sendo que esta estava conectado ao notebook, que capturava os dados.

Tabela 8: Resultados das medidas para a Posição 1

Posição	Parâmetros			Resultados			
	SF	BW	CR	Mens. Perdidas	RSSI [dBm]	Tempo [ms]	
1	7	125	4/5	2%	-85 ± 3	259 ± 29	
			4/8	4%	-91 ± 3	330 ± 29	
		250	4/5	2%	-83 ± 4	173 ± 30	
			4/8	0%	-79 ± 1	211 ± 33	
		9	125	4/5	2%	-80 ± 2	653 ± 33
				4/8	5%	-74 ± 2	884 ± 28
	10	250	4/5	1%	-75 ± 1	370 ± 25	
			4/8	5%	-88 ± 3	489 ± 28	
		125	4/5	0%	-79 ± 3	1125 ± 19	
			4/8	0%	-85 ± 2	1547 ± 30	
		250	4/5	1%	-77 ± 1	603 ± 17	
			4/8	4%	-91 ± 3	330 ± 29	

Tabela 9: Resultados das medidas para a Posição 2

Posição	Parâmetros			Resultados			
	SF	BW	CR	Mens. Perdidas	RSSI [dbm]	Tempo [ms]	
2	7	125	4/5	1%	-80 ± 2	256 ± 24	
			4/8	0%	-85 ± 3	327 ± 22	
		250	4/5	1%	-84 ± 2	168 ± 17	
			4/8	0%	-77 ± 1	207 ± 25	
		9	125	4/5	2%	-74 ± 1	649 ± 18
				4/8	8%	-72 ± 1	882 ± 22
	10	250	4/5	1%	-76 ± 3	369 ± 28	
			4/8	1%	-85 ± 4	488 ± 33	
		125	4/5	1%	-73 ± 1	1123 ± 15	
			4/8	0%	-81 ± 2	1544 ± 24	
		250	4/5	2%	-78 ± 1	609 ± 24	
			4/8	0%	-85 ± 1	327 ± 22	

Tabela 10: Resultados das medidas para a Posição 3

Posição	Parâmetros			Resultados			
	SF	BW	CR	Mens. Perdidas	RSSI [dbm]	Tempo [ms]	
3	7	125	4/5	71%	-95 ± 1	253 ± 14	
			4/8	23%	-98 ± 1	328 ± 23	
		250	4/5	96%	-94 ± 1	165 ± 2	
			4/8	11%	-93 ± 1	209 ± 23	
		9	125	4/5	4%	-96 ± 1	651 ± 27
				4/8	9%	-95 ± 1	882 ± 24
	250		4/5	9%	-92 ± 1	367 ± 19	
			4/8	0%	-89 ± 1	489 ± 29	
	10		125	4/5	0%	-95 ± 2	1125 ± 21
				4/8	0%	-95 ± 1	1544 ± 20
	250	4/5	41%	-91 ± 2	608 ± 21		
		4/8	23%	-98 ± 1	328 ± 23		

Tabela 11: Resultados das medidas para a Posição 4

Posição	Parâmetros			Resultados			
	SF	BW	CR	Mens. Perdidas	RSSI [dbm]	Tempo [ms]	
4	7	125	4/5	97%	-95 ± 1	275 ± 53	
			4/8	97%	-99 ± 1	362 ± 57	
		250	4/5	100%	-	-	
			4/8	100%	-	-	
		9	125	4/5	95%	-100 ± 2	676 ± 48
				4/8	6%	-99 ± 1	883 ± 29
	250		4/5	22%	-94 ± 1	369 ± 26	
			4/8	7%	-93 ± 1	490 ± 33	
	10		125	4/5	1%	-96 ± 3	1125 ± 21
				4/8	4%	-97 ± 1	1545 ± 24
	250	4/5	39%	-94 ± 1	603 ± 17		
		4/8	97%	-99 ± X	362 ± 57		

- Posição: Segue a referência indicada na Tabela 2 e na Figura 4;
- SF: Trata-se do *Spreading Factor*. Para a comunicação ocorrer normalmente todos os parâmetros devem ser configurados da mesma forma tanto no terminal remoto quanto no *gateway*;
- BW: *Bandwidth*, i.e., largura de banda, em KHz;

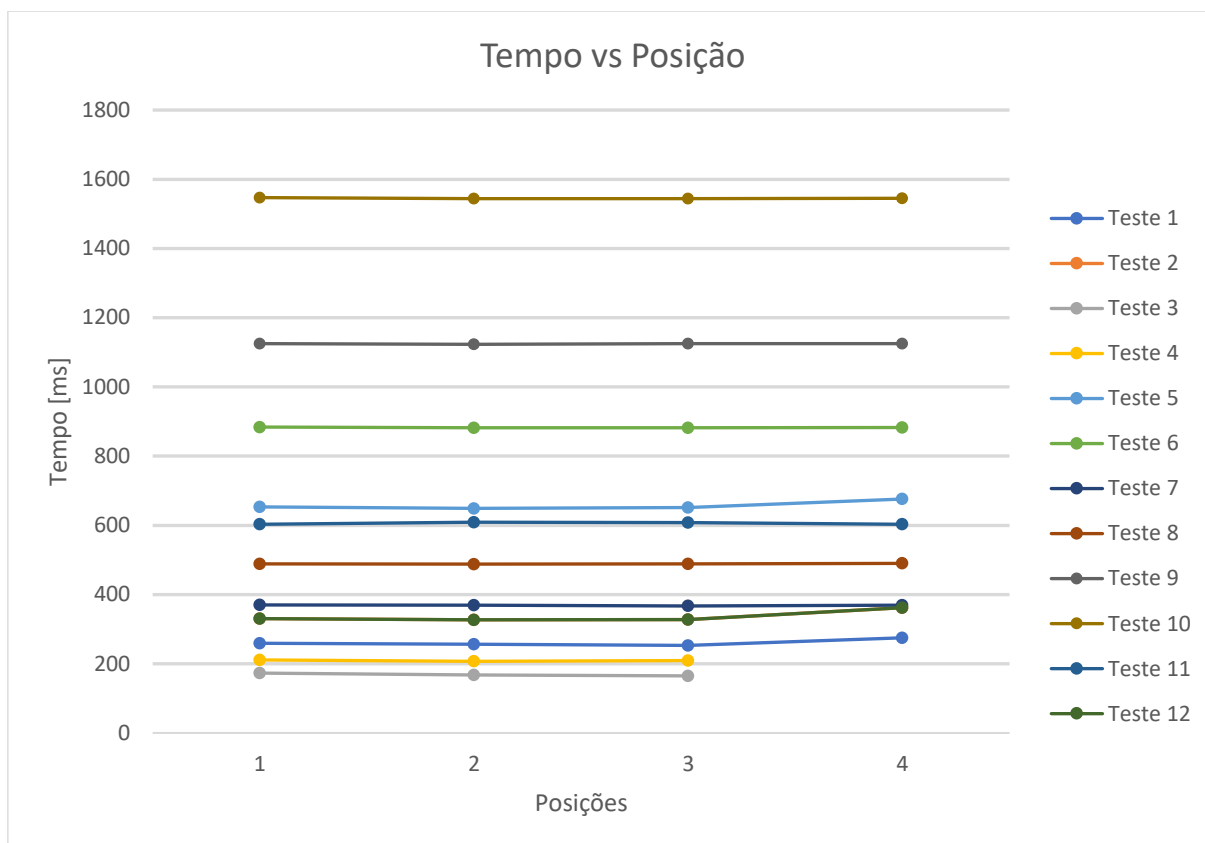
- Mens. Perdidas: Percentual de mensagens perdidas durante a comunicação. Cada teste em cada posição tinha como premissa a troca de pelo menos 200 mensagens. Dessa forma, o tempo de cada teste foi entre 12 e 20 minutos.
- Potência: Trata-se do RSSI (*Received Signal Strength Indication*), medindo a potência do sinal recebido pelo ponto remoto;
- Tempo: Mede o intervalo de tempo entre o envio da mensagem, o processamento, e o recebimento da mensagem de retorno indicando que a mensagem foi recebida pelo *gateway*.

Observa-se que, para a Posição 4, nenhuma mensagem foi recebida para as configurações de SF = 7, BW = 250 kHz para as duas opções de *Coding Rate*. Dessa forma, não foi possível aferir nenhuma medida de potência e tempo para esses dois casos.

4.1 TEMPO DE ENVIO E RECEBIMENTO DAS MENSAGENS

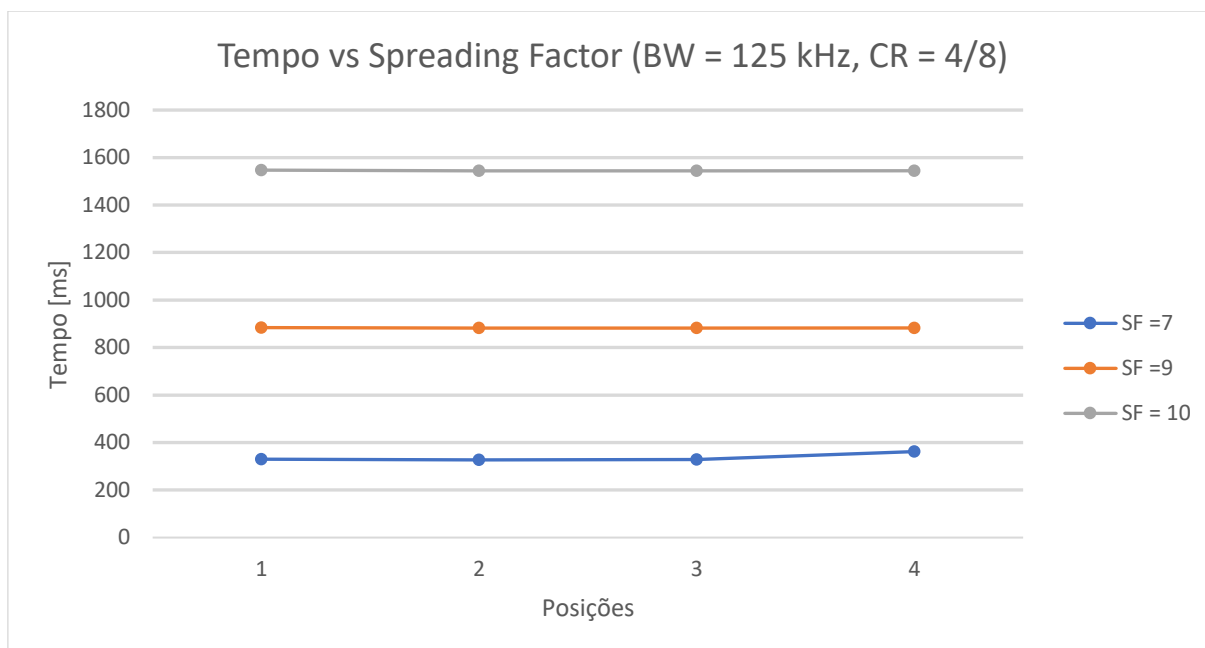
É possível observar no Gráfico 1 a seguir que, ao variar a distância sem variar nenhuma configuração, o tempo encontrado para a comunicação é constante. As linhas foram traçadas para ser possível observar várias configurações no mesmo gráfico, e evidenciar a tendência de permanecerem constantes.

Gráfico 1: Tempo versus posição



Assim a diferença entre os tempos demonstrados em cada teste deve-se aos outros parâmetros. No gráfico a seguir, vemos a alteração do tempo ao se levar em conta somente o *Spreading Factor* de cada teste, no qual é possível observar uma influência muito maior. Observe que os testes 3 e 4 não obtiveram resultados, pois não foi possível comunicação. São os testes com Spreading Factor 7 e largura de banda de 250 KHz, alterando apenas o Coding Rate.

Gráfico 2: Tempo versus Spreading Factor



Para esse gráfico utilizou-se uma configuração fixa para os outros parâmetros e alterou-se apenas o Spreading Factor. No caso, para todos os dados:

- BW = 125kHz
- CR = 4/8

Para outros gráficos em seguida será utilizado essa mesma forma de comparação, com o intuito de facilitar a sua análise.

Na relação visto acima, nota-se um aumento de tempo conforme esperado, já que o *Spreading Factor* é o fator mais determinante para o tempo no ar na comunicação.

Outro fator parece ser importante para esta análise, que é a taxa de informação transmitida por segundo, a qual segue a fórmula já vista anteriormente (Equação 1).

Produzindo para SF = 10 e alterando os outros valores *Coding Rate* e largura de banda, encontra-se valores entre 610 bps e 1,9 kbps. Apesar de já parecer suficiente para a comunicação que está sendo feita, é possível que essa taxa já esteja influenciando na quantidade de pacotes enviados por mensagem e assim aumentando ainda mais o tempo de envio por mensagem.

Nos três gráficos a seguir, relacionando a largura de banda e a taxa de correção de erros com o tempo, percebe-se a relação já confirmada em teoria: uma largura de banda menor permite menos dados sendo transmitidos, portanto é maior

o tempo para transmissão da mesma quantidade de informação. Assim como para um *coding rate* menor, também se nota que as mensagens foram enviadas em menos tempo, já que a redundância necessária para a correção de erro ocupa parte dessa banda.

Gráfico 3: Tempo versus Coding Rate(I)

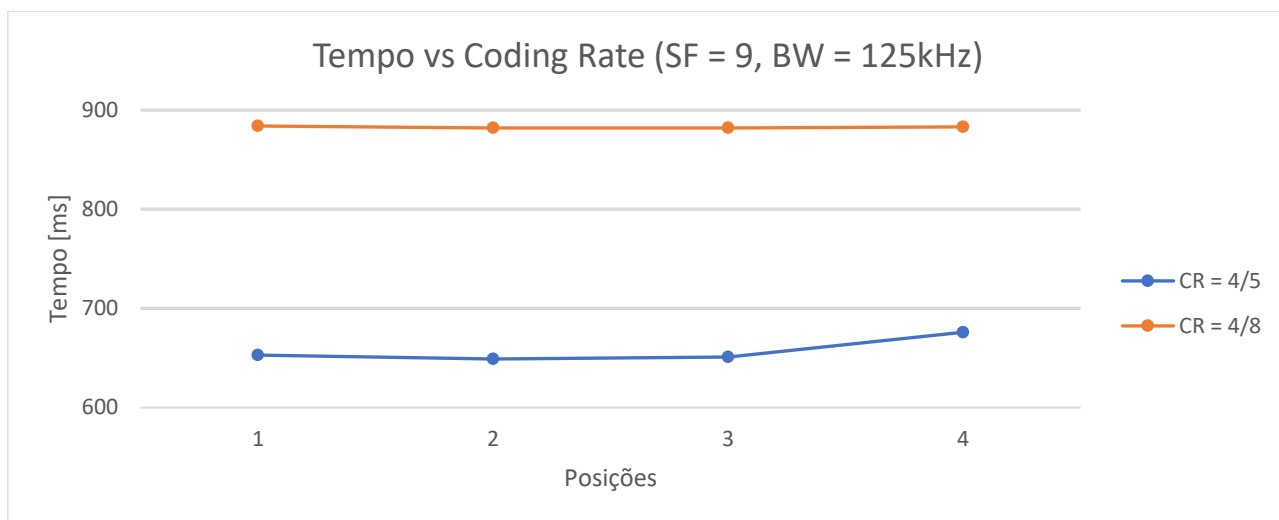


Gráfico 4: Tempo versus Coding Rate (II)

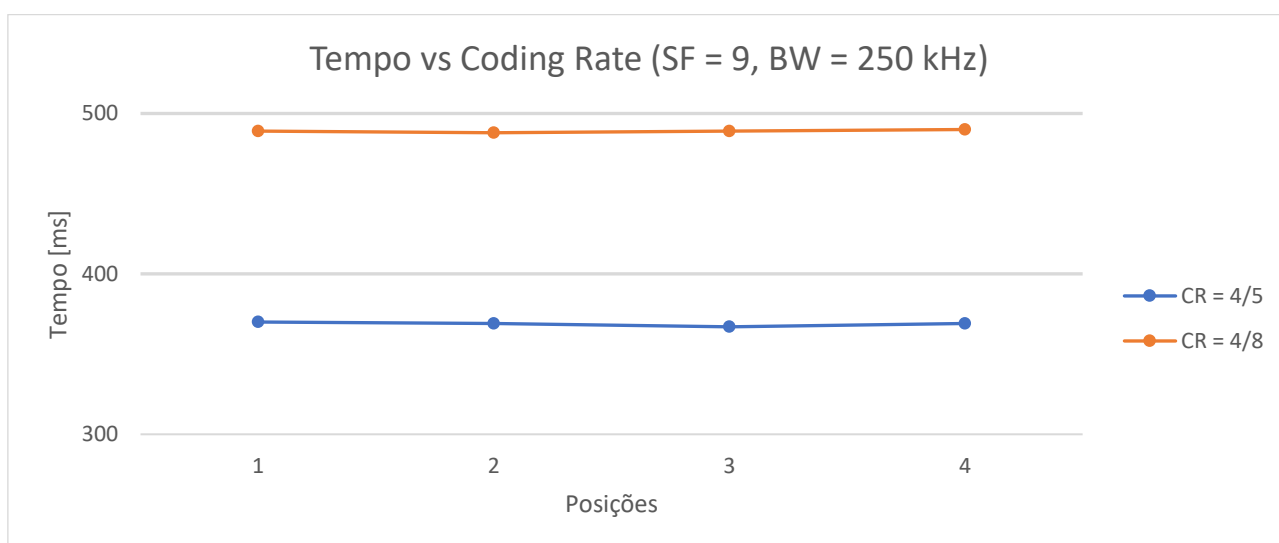
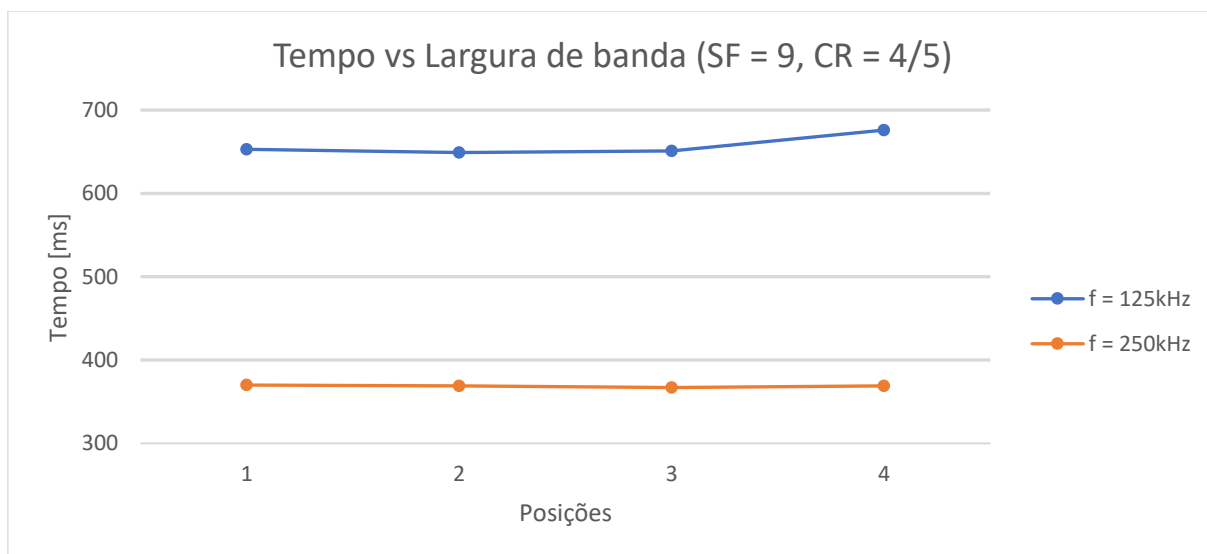


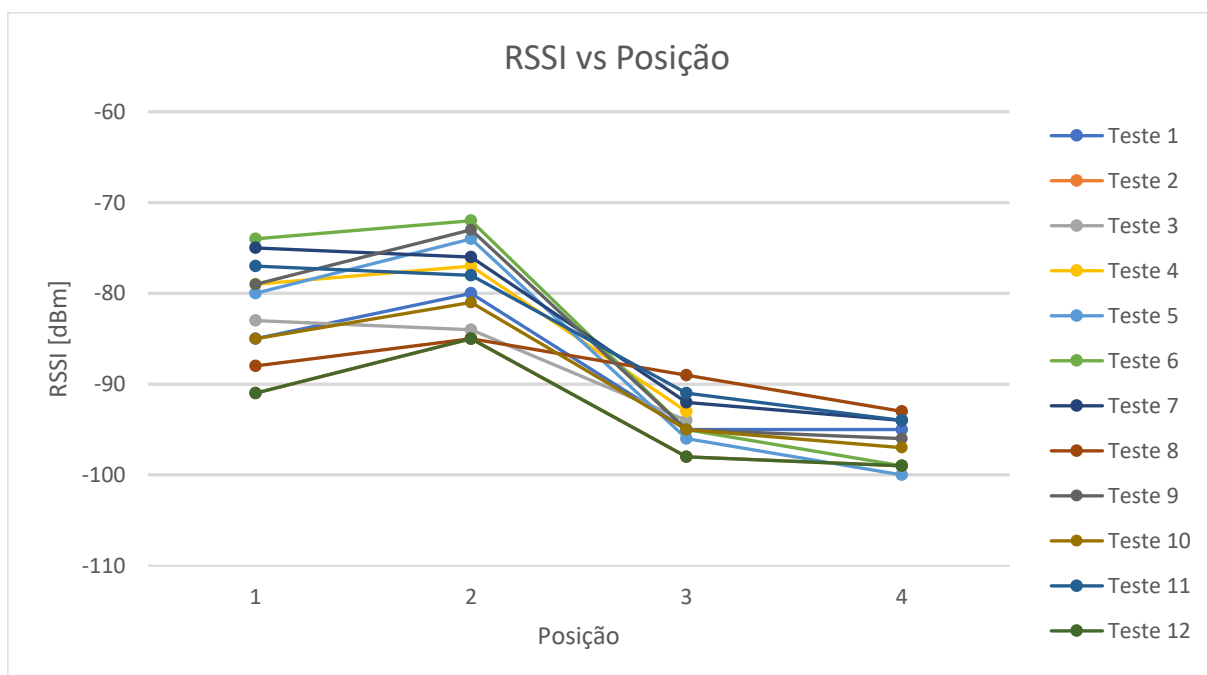
Gráfico 5: Tempo versus largura de banda



4.2 POTÊNCIA

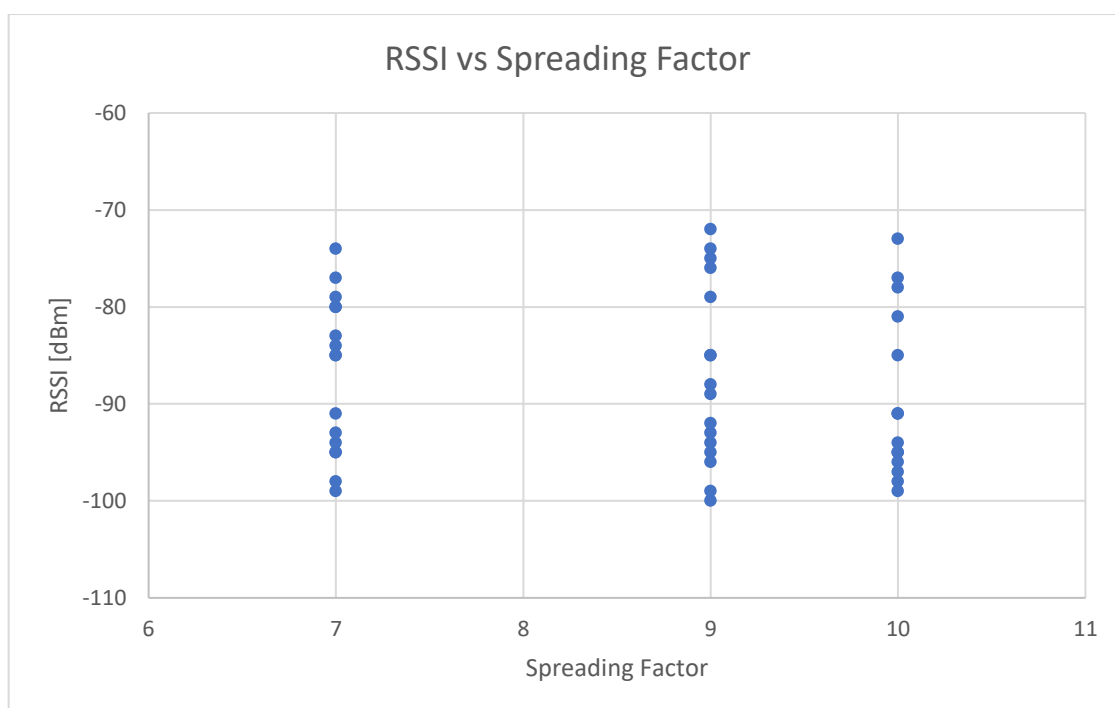
No Gráfico 5 é possível observar que o fator preponderante é a relação entre a potência do sinal recebido e a distância. De forma geral, observa-se a atenuação que ocorre conforme a distância aumenta entre o *gateway* e o ponto central. Observa-se também que o aparelho, para as configurações mais robustas, recebe sinais de RSSI até 100 dBm.

Gráfico 6: RSSI versus Posição



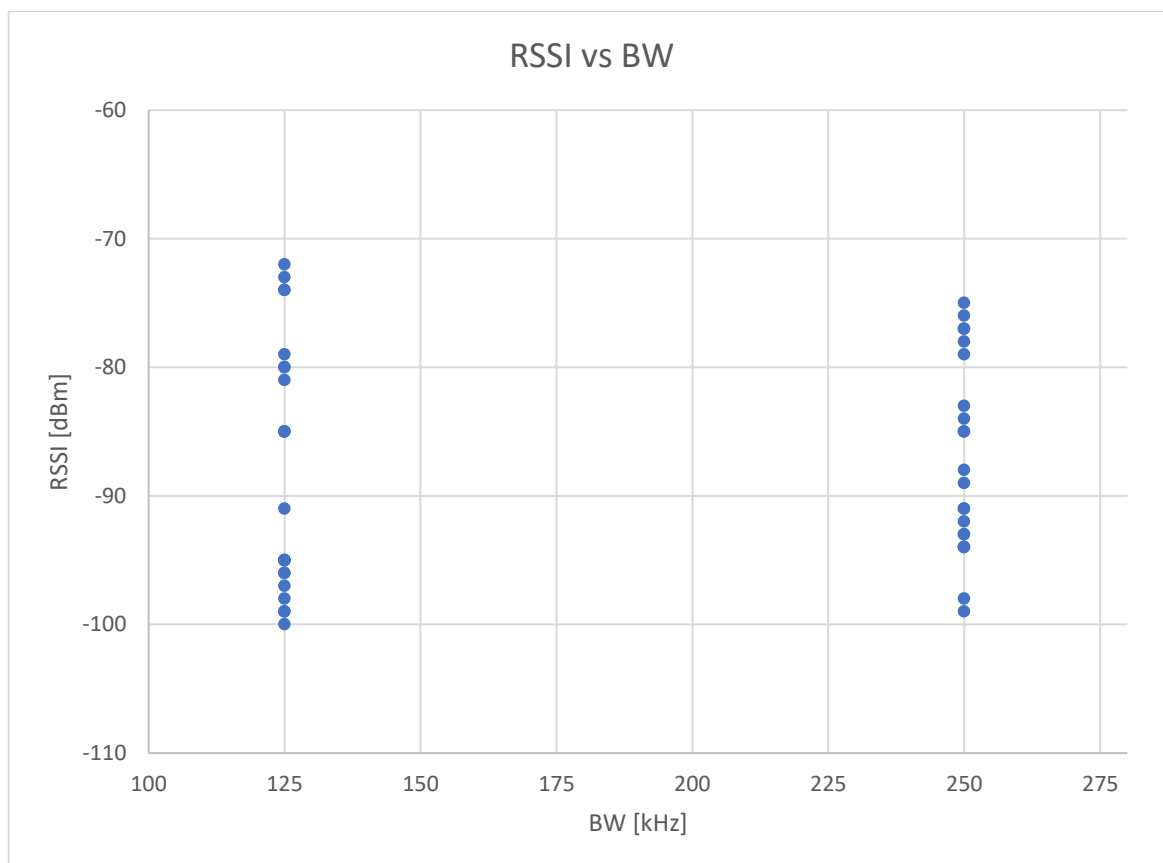
Como foi descrito anteriormente, o local do *gateway* possuía uma visibilidade maior dos pontos 2 e 3, no gráfico nota-se que para o ponto 2 o sinal foi menos atenuado, o que pode ser explicado por essa questão geográfica. Há uma diferença entre os testes de forma geral, na primeira posição os pontos variam em pouco menos de 20 dBm e na última posição 10 dBm. Essa diferença de atenuação deve-se aos parâmetros, que ao alterar a robustez da comunicação, observa-se uma diferença na potência recebida.

Gráfico 7: RSSI *versus* Spreading Factor



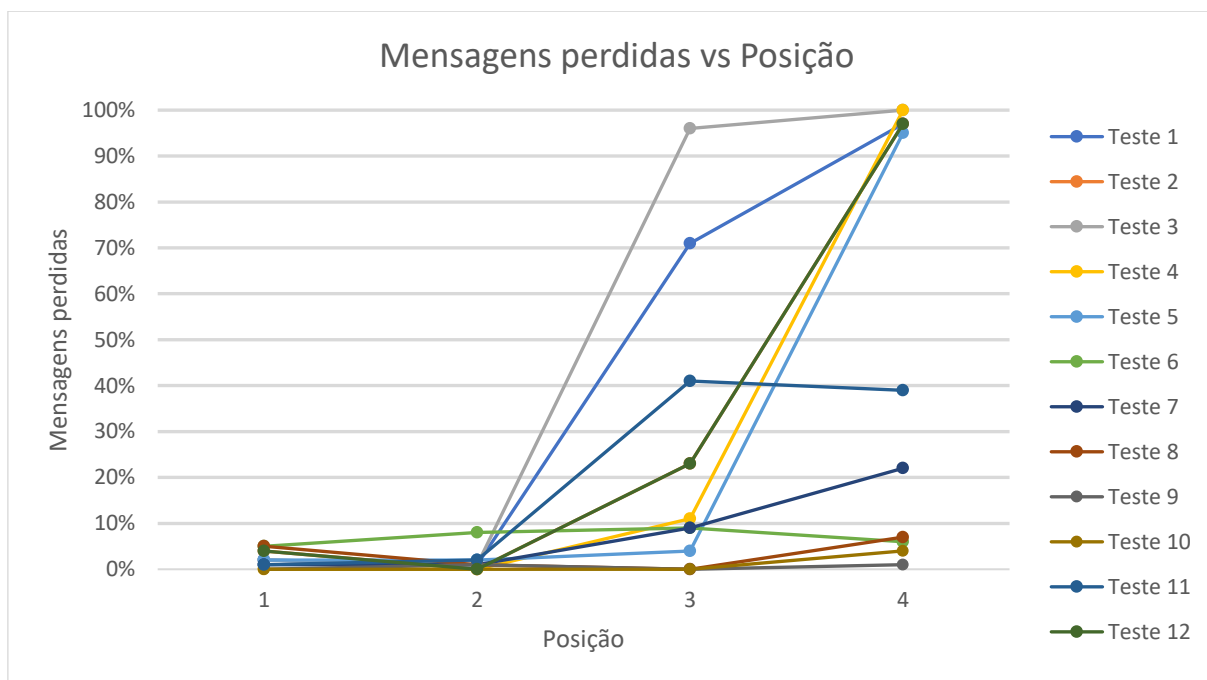
Ao observar o Gráfico 6 nota-se um RSSI praticamente constante ao alterar-se o *Spreading Factor*. Assim, é possível dizer que não há uma mudança significativa no RSSI com a mudança de *Spreading Factor*.

Para o Gráfico 8 abaixo, há uma ligeira diminuição na variação de RSSI ao aumentar-se a banda para 250 kHz, entretanto nenhum dos outros parâmetros mostrou exercer importância significativa para a potência recebida, diferentemente da distância e a posição do equipamento, conforme era esperado.

Gráfico 8: RSSI *versus* largura de banda

4.3 MENSAGENS PERDIDAS

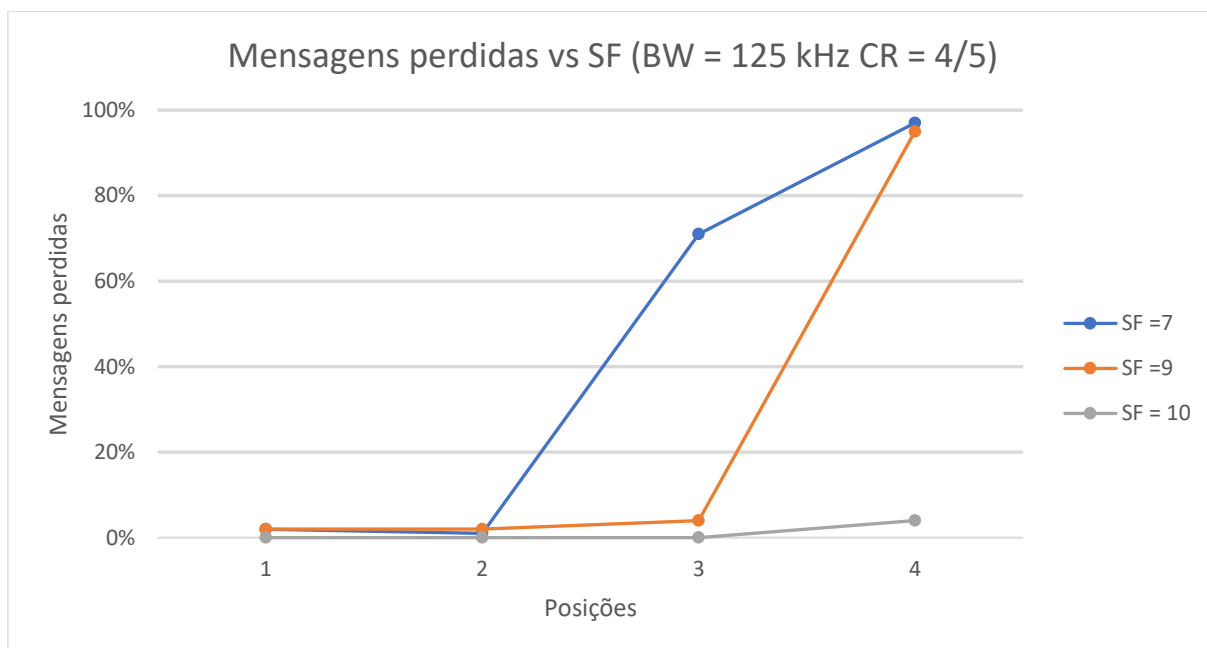
No Gráfico 9 observa-se a seguinte relação entre a distância e o percentual de mensagens perdidas: ao estar mais distante, maior será o percentual de mensagens perdidas. Salienta-se que, assim como na análise de RSSI, o ponto 1 possui uma média maior de mensagens perdidas em relação ao ponto 2, já que a visibilidade deste torna a atenuação menor e a comunicação mais eficiente.

Gráfico 9: Mensagens perdidas *versus* posição

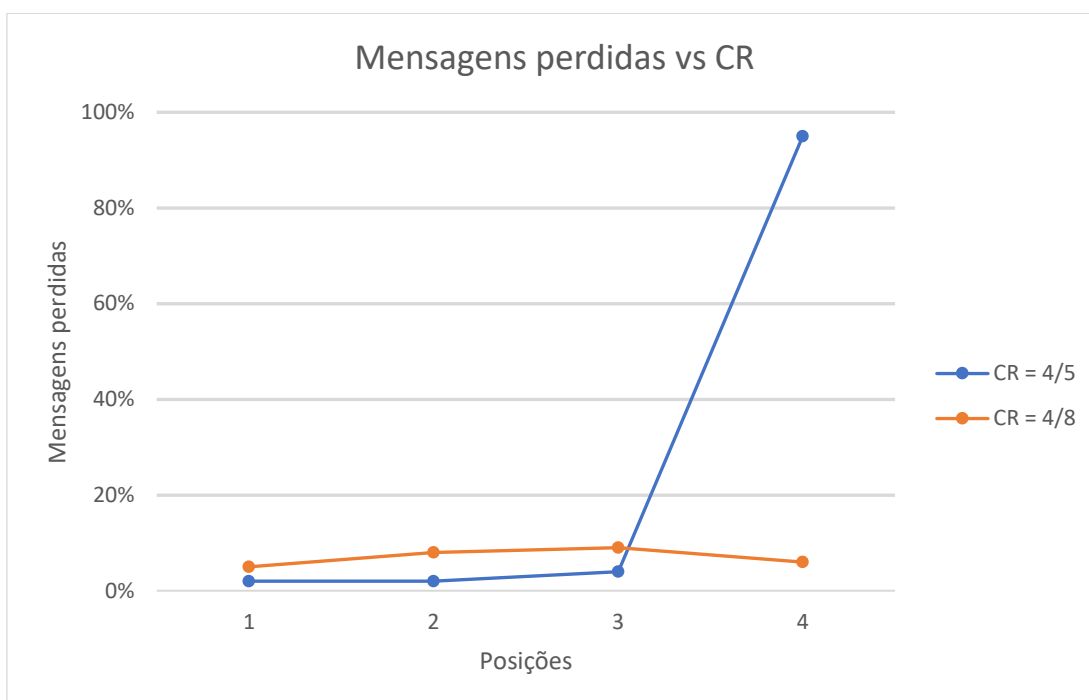
É possível observar que para os últimos testes, com SF mais alto, o percentual de mensagem perdida permaneceu baixo e praticamente constante para os últimos pontos. Essa característica segue conforme esperado, já que para SF maiores espera-se alcançar distâncias maiores, ultrapassando de forma significativa os pontos utilizados, conforme tabela 2.

Para o Gráfico 10, observa-se que para os dois primeiros pontos, mais próximos, a escolha do Spreading Factor foi irrelevante. Para a posição 3, já se nota que um SF maior garantiria uma taxa de mensagens perdidas aceitável, o que não acontece para SF = 7. Em conjunto com a posição 4, encontra-se o que é esperado pela teoria novamente, que ao aumenta o Spreading Factor, menos mensagens sejam perdidas.

Gráfico 10: Mensagens perdidas versus Spreading Factor

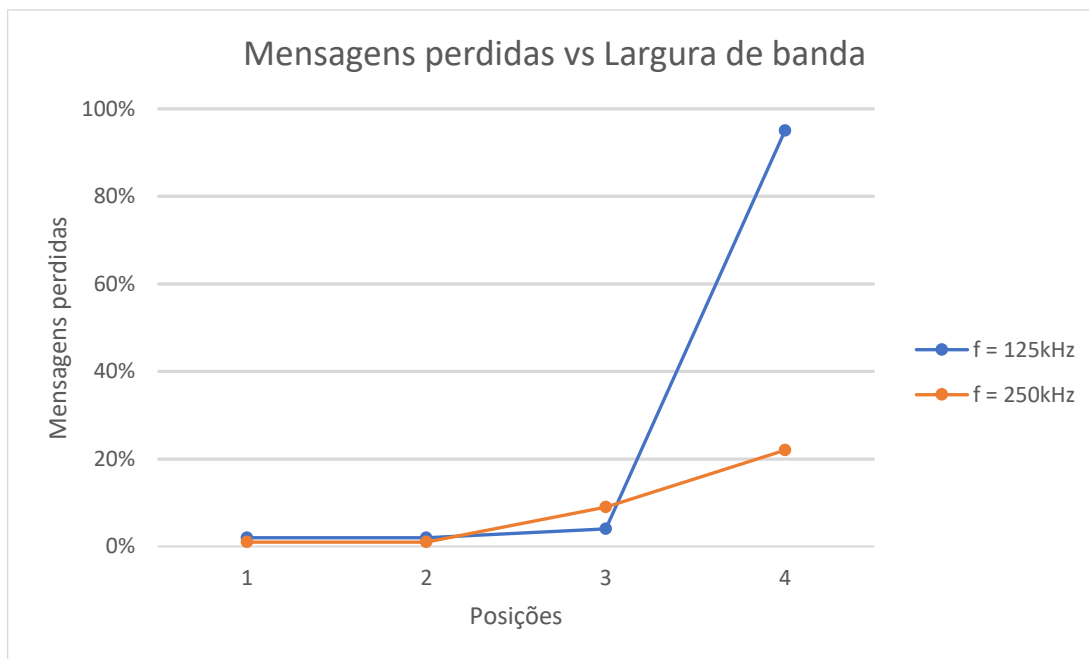


No Gráfico 11, é possível observar uma melhora quando se utiliza um *Coding Rate* mais robusto, como era esperado. Para as primeiras posições há um menor percentual de mensagens perdidas para CR = 4/5 comparado ao CR = 4/8. Além dessa diferença ser pequena, outros fatores, por exemplo alguma interferência, podem estar influenciando nessa comparação.

Gráfico 11: Mensagens perdidas *versus* Coding Rate

Para o Gráfico 12, a largura de banda não aparenta influenciar de forma significativa as mensagens perdidas, um ligeiro aumento de mensagens perdidas observadas pode se dar por conta de que, o aumento da largura de banda obriga uma maior capacidade de detecção do sinal, ainda que aumente a taxa de informação transmitida.

Gráfico 12: Mensagens perdidas *versus* largura de banda



4.4 TESTE ADICIONAL PARA DISTÂNCIA MÁXIMA

A Figura 9 mostra os últimos locais em que a comunicação ocorreu normalmente.



Figura 9 – Pontos de distância máxima de comunicação para SF = 10

Sendo assim, as distâncias encontradas, foram:

Ponto 1: 192 ± 2 m

Ponto 2: 242 ± 2 m

Considerando que o *gateway* está disposto de forma a favorecer o ponto 1, esperava-se comunicação com distâncias maiores para essa direção. Os prédios e outros bloqueios, como árvores e casas, interferindo na comunicação podem explicar esse resultado. Dado que o ponto 1 está em local mais íngreme que o ponto 2, tem-se que o ponto 2 está mais alinhado com o *gateway*, o que pode ter favorecido a comunicação.

5 CONCLUSÃO

O trabalho permitiu observar as diferentes características de uma rede LPWAN, mais especificamente de uma rede LoRa, com o estudo teórico e posteriormente uma análise prática de uma rede LoRa em funcionamento. Com a ascensão da “Internet das Coisas”, esse tipo de tecnologia será relevante para que seja possível utilizar todo o potencial de redes com dispositivos conectados para diferentes razões, com aplicações que se encaixam nesse tipo de comunicação, com baixa energia sendo utilizada, e grande alcance. Essas características aliadas ao baixo custo, possuem como desvantagem a taxa de informação reduzida, mas que para esse tipo de comunicação pode ser suficiente ou adaptada para isso.

Na parte prática, o estudo permitiu observar algumas características do funcionamento, demonstrando que diferentes perfis de uso, levando em conta a distância dos equipamentos e suas aplicações, podem ser utilizados, sendo necessária a configuração para que a rede e os equipamentos possuam a característica adequada para as aplicações que serão utilizadas.

O algoritmo montado para esse trabalho possui por padrão uma comunicação que funciona baseado em confirmação, isto é, para o envio da próxima mensagem, o algoritmo aguarda uma resposta. Comunicações que não precisem desse tipo de confirmação podem ser as mais adequadas para uma rede LoRa, entretanto para

ser possível o estudo e a análise percentual de mensagens perdidas, esse tipo de comunicação era a mais adequada.

Os testes realizados, no entanto, não são suficientes para se observar todas as características de uma rede LoRa. Apesar de ser possível observar algumas limitações de distância, priorizou-se um estudo com mais dados em certos locais, para se obter uma maior precisão nas informações, o que limitou a quantidade de pontos observados e assim não foi possível a aferição de pontos mais distantes. Apesar disso, a quantidade de testes feitos não foi suficiente para se observar como alguns fatores externos podem influenciar no resultado, como temperatura, pressão e interferências. O posicionamento do *gateway* idealmente deveria ser numa localização que possuísse nenhum obstáculo e que fosse aberto, em cima de um prédio, por exemplo.

Os parâmetros aqui estudados mostraram possuir uma importância muito relevante para a montagem e configuração de um sistema utilizando equipamentos LoRa. O Spreading Factor altera de forma significativa o atraso e a perda de mensagens, já que ele é o principal fator a permitir mensagens enviadas para distâncias maiores. O Coding Rate e a largura de banda devem levar outros fatores em consideração, como a taxa de informação desejada para a largura de banda e a robustez para o Coding Rate.

Ao permitir essa configuração, o projeto é adequado para aplicações de IoT, permitindo a transmissão de informações para longas distância e necessitando de pouca potência. De forma geral, aplicações IoT precisam de baixa taxa de transmissão, o que permite o uso de parâmetros mais robustos. Ao simular uma aplicação, em alguns cenários foi possível observar um percentual de mensagens perdidas adequado para todas as posições, ficando abaixo de 10% para algumas configurações com SF igual à 9 ou 10.

Levando em consideração os pontos levantados acima, o desenvolvimento desse trabalho permitiu uma análise satisfatória das características de uma rede LoRa. Outros estudos adicionais, como a utilização das informações em serviços pela internet que permitem o monitoramento online dos dispositivos remotos, são vislumbrados como um próximo passo, indo ao encontro do seu principal uso, que é a Internet das Coisas. Ainda assim, o uso local já permitiu a análise das características desses equipamentos.

REFERÊNCIAS

- [1] CENTENARO, M. et al. Long-Range Communications In Unlicensed Bands: The Rising Stars In The IoT And Smart City Scenarios. **IEEE Wireless Communications**, p. 60-67, out. 2016.
- [2] RAZA, U.; KULKARNI, P; SOORIYABANDARA. Low Power Wide Area Networks: An Overview. **IEEE Communications Surveys & Tutorials**, p. 855-873, 2º quadrimestre 2017.
- [3] ANATEL. Resolução nº 506, de 1º de julho de 2008. Republica o Regulamento sobre Equipamentos de Radiocomunicação de Radiação Restrita. 2008. Disponível em: <<http://www.anatel.gov.br/legislacao/resolucoes/2008/104-resolucao-506>>. Acesso em: 29 jul. 2017.
- [4] ELECTRONIC ENGINEERING JOURNAL. Low Power, Wide Area Disponível em: <<http://www.eejournal.com/article/20150907-lpwa/f>>. Acesso em: 06 abr. 2017.
- [5] FLESPI. Top 7 technologies for IoT connectivity 2017. Disponível em: <<https://flespi.com/blog/top-7-technologies-for-iot-connectivity-2017>>. Acesso em: 30 jul. 2017.
- [6] UNIVERSIDADE FEDERAL DO RIO DE JANEIRO. Grupo de Teleinformática e Automação (GTA). **Redes de Sensores Sem Fio: Protocolos**. Rio de Janeiro: UFRJ, 2011. Disponível em: <https://www.gta.ufrj.br/grad/11_1/rssf/>. Acesso em: 30 jul. 2017.
- [7] LORA ALLIANCE. LoRaWAN™ 101 – A Technical Introduction. Disponível em: <https://docs.wixstatic.com/ugd/eccc1a_20fe760334f84a9788c5b11820281bd0.pdf>. Acesso em: 30 jul. 2017.
- [8] MIRANDA, H. do C. **Realização de um Sistema de Comunicação de Espalhamento Espectral Usando Técnicas de Rádio Digital**. Porto: UNIVERSIDADE DO PORTO, Jul. 1998: Tese (Mestrado em Engenharia Eletrotécnica e de Computadores) - Departamento de Engenharia Eletrotécnica e de Computadores, Universidade do Porto, 1998. . Disponível em: <http://paginas.fe.up.pt/~hmiranda/papers/msc_thesis.pdf>. Acesso em: 30 jul. 2017.

[9] SORNIN, N. et al. **LoRaWAN™ Specification**. LORA ALLIANCE. Jul. 2016. Disponível em: < <https://www.lora-alliance.org/what-is-lora> >. Acesso em: 30 jul. 2017.

[10] ARDUINO. **So you want to make an Arduino**. Disponível em: < <https://www.arduino.cc/en/Main/Policy> >. Acesso em: 8 ago. 2017.

[11] UDOIT. **LORA SHIELD FOR ARDUINO**. Disponível em: < <http://udoit.eu/lora-shield-arduino/lora> >. Acesso em: 8 ago. 2017.

[12] DRAGINO. **LoRa Shield**. Disponível em: <<http://www.dragino.com/products/module/item/102-lora-shield.html>>. Acesso em: 8 ago. 2017.

[13] CRUZ, R. USP cria centro de tecnologias interativas. **O Estado de S. Paulo**. 30 out. 2011. Disponível em: <<http://economia.estadao.com.br/noticias/geral,usp-cria-centro-de-tecnologias-interativas-imp-,7924571>>. Acesso em: 8 ago. 2017.

[14] MORO, B.; SUZIN, R. Automação de sistema de irrigação de pastagem. **Fundação Araucária**. Paraná, 2012. Disponível em: <<https://www.fag.edu.br/novo/arquivos/graduacao/engenharias/resumos/resumo03.pdf>>. Acesso em: 9 ago. 2017.

[15] SATO, T. Sistema Automotivo de Notificação de Acidente. **Universidade Federal do Paraná**. Curitiba, 2010. Disponível em: <<http://www.eletrica.ufpr.br/ufpr2/tccs/147.pdf>>. Acesso em: 9 ago. 2017.

[16] INSTITUTO NCB. **Conheça a tecnologia LoRa® e o protocolo LoRaWAN™ (LOR001)** Disponível em: <<http://newtoncbraga.com.br/index.php/electronica/52-artigos-diversos/11992-conheca-a-tecnologia-lora-e-o-protocolo-lorawan-lor001>>. Acesso em: 25 mar. 2018.

[17] EXPLORATORY ENGINEERING. **Data Rate and Spreading Factor**. Disponível em: <https://docs.exploratory.engineering/lora/dr_sf/>. Acesso em: 25 mar. 2018.

[18] SEMTECH. **LoRa Modem Design Guide**. Jul. 2013. Disponível em: <https://www.semtech.com/uploads/documents/LoraDesignGuide_STD.pdf>. Acesso em: 26 mar. 2018.

[19] SEMTECH. **LoRa Modulation Basics**. Mai. 2015. Disponível em: <<https://www.semtech.com/uploads/documents/an1200.22.pdf>>. Acesso em: 26 mar. 2018.

ANEXOS

I. Código para o arduíno do equipamento central (*gateway*)

```

/*
11/11/2017 - TG UFABC
André Luiz Silva Cavalheiro
Server_LoRa_v3
Última versão: 24/03/2018
*/

#define BAUDRATE 115200

#include <Console.h>
#include <SPI.h>
#include <RH_RF95.h>

// Singleton instance of the radio driver
RH_RF95 rf95;

int led = A2;
float frequency = 915.0;

void setup()
{
  pinMode(led, OUTPUT);
  Bridge.begin(BAUDRATE);
  Console.begin();
  while (!Console) ; // Wait for console port to be available
  Console.println("Start Sketch");
  if (!rf95.init())
    Console.println("init failed");
  // Setup ISM frequency
  rf95.setFrequency(frequency);
  // Setup Power, dBm
  rf95.setTxPower(13);

  // Setup Spreading Factor (6 ~ 12)
  rf95.setSpreadingFactor(8);

  // Setup BandWidth, option:
  125000,250000
  rf95.setSignalBandwidth(125000);

  // Setup Coding Rate:5(4/5),6(4/6),7(4/7),8(4/8)
  rf95.setCodingRate4(5);

  Console.print("Listening on frequency: ");
  Console.println(frequency);
}

void loop()
{
  if (rf95.available())
  {
    // Should be a message for us now
    uint8_t buf[RH_RF95_MAX_MESSAGE_LEN];
    uint8_t len = sizeof(buf);
  }
}

```

```

    if (rf95.recv(buf, &len))
    {
        if(buf[0] == 1 & buf[1] == 1 & buf[2] ==1) //Get sensor data from node
id 111
        {
            int newData[4] = {0, 0, 0, 0};
            for (int i = 0; i < 2; i++)
            {
                newData[i] = buf[i + 3];
            }
            int a = newData[0];
            int b = newData[1];
            Console.print("ID = 111 :");
            Console.print("Current humdity = ");
            Console.print(a);
            Console.print("% ");
            Console.print("temperature = ");
            Console.print(b);
            Console.println("C ");
            Console.print("RSSI: ");
            Console.println(rf95.lastRssi(), DEC);
            uint8_t data[] = "And hello back to you";
            rf95.send(data, sizeof(data));
            rf95.waitPacketSent();
            Console.println("Sent a reply");
            digitalWrite(led, LOW);
        }
        if(buf[0] == 1 & buf[1] == 1 & buf[2] == 2) //Get sensor data from
node id 112
        {
            int newData[4] = {0, 0, 0, 0};
            for (int i = 0; i < 2; i++)
            {
                newData[i] = buf[i + 3];
            }
            int a = newData[0];
            int b = newData[1];
            Console.print("ID = 112 :");
            Console.print("Current humdity = ");
            Console.print(a);
            Console.print("% ");
            Console.print("temperature = ");
            Console.print(b);
            Console.println("C ");
        }
    }
    else
    {
        Console.println("recv failed");
    }
}
}

```

II. Código para o arduíno do equipamento remoto

```

/*
11/11/2017 - TG UFABC

```

```

André Luiz Silva Cavaleiro
Server_LoRa_v1
Última versão: 24/03/2018
*/

#include <SPI.h>
#include <RH_RF95.h>
#include <String.h>

RH_RF95 rf95;
float frequency = 915.0;

#define dht_dpin A0 // Use A0 pin to connect the data line of DHT11
byte bGlobalErr;
char dht_dat[5];

void InitDHT() // Initiate DHT11
{
    pinMode(dht_dpin,OUTPUT);//
    digitalWrite(dht_dpin,HIGH);//Set A0 to output high
}

void ReadDHT() // Read Temperature and Humidity value
{
    bGlobalErr=0;
    byte dht_in;
    byte i;

    //pinMode(dht_dpin,OUTPUT);
    digitalWrite(dht_dpin,LOW);//pull low data line to send signal.
    delay(30);//Add a delay higher than 18ms so DHT11 can detect the start
    signal

    digitalWrite(dht_dpin,HIGH);
    delayMicroseconds(40);
    pinMode(dht_dpin,INPUT);
    // delayMicroseconds(40);
    dht_in=digitalRead(dht_dpin);//Get A0 state
    // Serial.println(dht_in,DEC);
    if(dht_in){
        bGlobalErr=1;
        return;
    }
    delayMicroseconds(80);//Get DHT11 response , pull low data lineDHT11
    dht_in=digitalRead(dht_dpin);

    if(!dht_in){
        bGlobalErr=2;
        return;
    }
    delayMicroseconds(80);//
    for (i=0; i<5; i++)//Ger Temperature and Humidity value
        dht_dat[i] = read_dht_dat();

    pinMode(dht_dpin,OUTPUT);
    digitalWrite(dht_dpin,HIGH);
    byte dht_check_sum =
        dht_dat[0]+dht_dat[1]+dht_dat[2]+dht_dat[3];//Calculate check
    sum
    if(dht_dat[4]!= dht_check_sum)//Error when check sum mismatch
        {bGlobalErr=3;}
}

```

```

};

byte read_dht_dat() {
  byte i = 0;
  byte result=0;
  for(i=0; i< 8; i++){
    while(digitalRead(dht_dpिन)==LOW); //
    delayMicroseconds(30); //
    if (digitalRead(dht_dpिन)==HIGH)
      result |= (1<<(7-i)); //
    while (digitalRead(dht_dpिन)==HIGH); //
  }
  return result;
}

// Singleton instance of the radio driver

void setup()
{
  InitDHT();
  Serial.begin(9600);
  //while (!Serial) ; // Wait for serial port to be available
  Serial.println("Start LoRa Client");
  if (!rf95.init())
    Serial.println("init failed");
  // Setup ISM frequency
  rf95.setFrequency(frequency);
  // Setup Power, dBm
  rf95.setTxPower(13);

  // Setup Spreading Factor (6 ~ 12)
  rf95.setSpreadingFactor(8);

  // Setup BandWidth, option:
  125000,250000
  //Lower BandWidth for longer distance.
  rf95.setSignalBandwidth(125000);

  // Setup Coding Rate:5(4/5),6(4/6),7(4/7),8(4/8)
  rf95.setCodingRate4(5);
}

void loop()
{
  ReadDHT();
  uint8_t data[50] = {0} ;
  data[0] = 1 ;
  data[1] = 1 ;
  data[2] = 1 ; // Use Data [0].Data[1], Data[2] to combine a Device ID.
  data[3] = dht_dat[0]; // store humidity data
  data[4] = dht_dat[2]; // store temperature data
  rf95.send(data, sizeof(data)); // Send out ID + Sensor data to LoRa
gateway
  //Serial.println("Sending to LoRa Server");
  // Send a message to LoRa Server
  switch (bGlobalErr)
  {
  case 0:
    Serial.print(dht_dat[0], DEC);

```

```

        Serial.print(".");
        Serial.print(dht_dat[1], DEC);
        Serial.print("%");
        Serial.print(dht_dat[2], DEC);
        Serial.print(".");
        Serial.print(dht_dat[3], DEC);
        Serial.print("C");
        Inicio = millis();
        Serial.print(Inicio);
        Serial.print(";");
        break;
    case 1:
        Serial.println("Error 1: DHT start condition 1 not met.");
        break;
    case 2:
        Serial.println("Error 2: DHT start condition 2 not met.");
        break;
    case 3:
        Serial.println("Error 3: DHT checksum error.");
        break;
    default:
        Serial.println("Error: Unrecognized code encountered.");
        break;
}

rf95.waitPacketSent();
// Now wait for a reply
uint8_t buf[RH_RF95_MAX_MESSAGE_LEN];
uint8_t len = sizeof(buf);

if (rf95.waitForAvailable(5000))
{
    // Should be a reply message for us now
    if (rf95.recv(buf, &len))
    {
        Retorno = millis();
        Serial.print(";");
        Serial.print(Retorno);
        Tempo_Passado = Retorno - Inicio;
        Serial.print(";");
        Serial.print(Tempo_Passado);
        Serial.print(";");
        Serial.println(rf95.lastRssi(), DEC);
    }
    else
    {
        Serial.println("recv failed");
    }
}
else
{
    Serial.println("No?");
}
delay(5000);
}

```


III. Diagrama eléctrico shield Lora / Arduino

