

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Isadora Rezende Lopes

**Ataques de Negação de Serviço em
Dispositivos LoRa**

Uberlândia, Brasil

2021

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Isadora Rezende Lopes

Ataques de Negação de Serviço em Dispositivos LoRa

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, como parte dos requisitos exigidos para a obtenção título de Bacharel em Ciência da Computação.

Orientador: Rodrigo Sanches Miani

Universidade Federal de Uberlândia – UFU

Faculdade de Ciência da Computação

Bacharelado em Ciência da Computação

Uberlândia, Brasil

2021

Isadora Rezende Lopes

Ataques de Negação de Serviço em Dispositivos LoRa

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, como parte dos requisitos exigidos para a obtenção título de Bacharel em Ciência da Computação.

Trabalho aprovado. Uberlândia, Brasil, 17 de junho de 2021:

Rodrigo Sanches Miani
Orientador

Paulo Rodolfo da Silva Leite Coelho

Rafael Pasquini

Uberlândia, Brasil
2021

Agradecimentos

Agradeço aos meus pais, Edmo e Letícia, por me proporcionarem a melhor educação possível, pelo incentivo e apoio nos momentos difíceis.

Ao meu orientador, Professor Rodrigo Sanches Miani, pela oportunidade de executar este trabalho, pela orientação, confiança e incentivo.

Ao meu namorado Ian, pelo apoio, paciência e incentivo durante minha graduação, principalmente nos momentos difíceis, por sempre acreditar no meu potencial e compartilhar seus conhecimentos comigo.

Aos meus avós e irmã, Marcelino, Heloísa e Vitória, por sempre acreditarem no meu potencial.

Aos meus amigos e colegas, que me ajudaram em diversos momentos e estiveram comigo durante esta trajetória.

Aos professores da FACOM por todo o conhecimento compartilhado, paciência e atenção dedicada aos alunos.

Resumo

O desenvolvimento de tecnologias utilizando o conceito de Internet das Coisas (do inglês *Internet of Things* - IoT) evolui diariamente. Essas inúmeras tecnologias, como os transceptores LoRa (*Long Range*), são aplicadas em diversas áreas, inclusive no monitoramento de desastres naturais ou desastres provocados por falhas humanas. As vulnerabilidades de segurança surgem juntamente com o desenvolvimento de novas tecnologias ou atualização de dispositivos existentes. Ataques que buscam impactar a disponibilidade dos dispositivos, como os ataques de Negação de Serviço ou DoS *attacks* (do inglês *Denial of Service*) podem ser um meio efetivo de explorar vulnerabilidades em dispositivos LoRa e impedir que o monitoramento de desastres aconteça. Dessa forma, o presente trabalho avalia o impacto de ataques DoS em dispositivos LoRa utilizados no contexto de monitoramento de desastres. Para isto, foram realizados experimentos, utilizando o simulador LoRaSim, com o intuito de degradar o desempenho da rede LoRaWAN proposta para o monitoramento de enchentes. Após executar os experimentos, concluiu-se que existe um alto impacto no desempenho da rede quando elevam-se ou diminuem-se certos parâmetros da mesma.

Palavras-chave: IoT, LoRa, Segurança da Informação, Monitoramento de Desastres, Ataques de Negação de Serviço.

Lista de ilustrações

Figura 1 – Ecosistema IoT. Ecosistema IoT. Fonte: adaptada de (ROUSE, 2016).	19
Figura 2 – Ecosistema LoRa. Fonte: Adaptada de (LORAALLIANCE, 2019) . . .	20
Figura 3 – Comandos iniciais para executar o simulador.	30
Figura 4 – Exemplo de comando para utilizar o script loraDirMulBS.py.	31
Figura 5 – Processo para encontrar os parâmetros da rede.	32
Figura 6 – Bacia do córrego do Leitão. Fonte: Extraída de (ROSA, 2017).	33
Figura 7 – Variação do DER em função do aumento do intervalo médio de envio para todas as quantidades de estações base.	35
Figura 8 – Particionamento de nível de inundação. Fonte: Extraída de (RAGNOLI et al., 2020).	37
Figura 9 – Variação do NEC em função do aumento do intervalo médio de envio de pacotes para a configuração 3.3.	40
Figura 10 – Variação do DER em função do aumento do intervalo médio de envio de pacotes para a configuração 3.3.	41
Figura 11 – Variação do NEC em função do aumento do intervalo médio de envio de pacotes para a configuração 3.4.	41
Figura 12 – Variação do DER em função do aumento do intervalo médio de envio de pacotes para a configuração 3.4.	41
Figura 13 – Variação do DER médio das estações base em função do aumento da distância entre as mesmas para a configuração 3.3.1.	44
Figura 14 – Variação do número de colisões em função do aumento da distância entre as estações base para a configuração 3.3.1.	44
Figura 15 – Variação do DER médio das estações base em função do aumento da distância entre as mesmas para a configuração 3.4.1.	44
Figura 16 – Variação do número de colisões em função do aumento da distância entre as estações base para a configuração 3.4.1.	45
Figura 17 – Distância entre as estações base com o valor 10 para a configuração 3.3.1.	46
Figura 18 – Distância entre as estações base com o valor 100 para a configuração 3.3.1.	46
Figura 19 – Distância entre as estações base com o valor 500 para a configuração 3.3.1.	46
Figura 20 – Distância entre as estações base com o valor 10 para a configuração 3.4.1.	47
Figura 21 – Distância entre as estações base com o valor 100 para a configuração 3.4.1.	47
Figura 22 – Distância entre as estações base com o valor 500 para a configuração 3.4.1.	47

Figura 23 – Estrutura do pacote	54
Figura 24 – Cenário 1, experimento 1 - Comportamento do DER em função do aumento inicial do número de nós para a verificação padrão e verificação completa de colisão.	56
Figura 25 – Cenário 1, experimento 2 - Comportamento do DER em função do segundo aumento do número de nós para a verificação padrão e verificação completa de colisão.	57
Figura 26 – Cenário 1, experimento 3 - Tentativa de sobrecarregar a rede aumentando o número de nós para a verificação completa de colisão.	58
Figura 27 – Cenário 1, experimento 4 - Comportamento do NEC em função do aumento inicial do número de nós.	58
Figura 28 – Cenário 2, experimento 1 - Comportamento do DER em função da diminuição inicial do intervalo médio de envio de pacotes.	59
Figura 29 – Cenário 2, experimento 2 - Comportamento do DER em função da diminuição agressiva do intervalo médio de envio de pacotes.	60
Figura 30 – Cenário 2, experimento 3 - Tentativa de degradar a disponibilidade da rede para a verificação completa de colisão.	61
Figura 31 – Cenário 2, experimento 4 - Comportamento do NEC em função da redução do intervalo médio de envio de pacotes.	61
Figura 32 – Cenário 3, experimento 1 - Comportamento do DER em função da redução do número de estações base.	63
Figura 33 – Cenário 4, experimento 1 - Comportamento do DER em função do aumento do número de nós e redução do intervalo médio de envio de pacotes para a verificação completa de colisão.	64
Figura 34 – Cenário 4, experimento 1 - Comportamento do DER em função do aumento do número de nós e redução do intervalo médio de envio de pacotes para a verificação padrão de colisão.	64
Figura 35 – Cenário 5, experimento 1 - Comportamento do DER em função da redução do número de estações e do intervalo médio de envio de pacotes para a verificação completa de colisão.	65
Figura 36 – Cenário 5, experimento 1 - Comportamento do DER em função da redução do número de estações base e do intervalo médio de envio de pacotes para a verificação padrão de colisão.	65
Figura 37 – Experimento 6 - Comportamento do DER em função da redução do número de estações base e do aumento de nós para a verificação completa de colisão.	66
Figura 38 – Experimento 6 - Comportamento do DER em função da redução do número de estações base e do aumento de nós para a verificação padrão de colisão.	67

Figura 39 – Cenário 7, experimento 1 - Comportamento do DER em função do aumento da distância máxima entre os nós e as estações base.	68
Figura 40 – Cenário 7, experimento 1 - Distância máxima entre os nós e as estações base igual a 98,95.	68
Figura 41 – Cenário 7, experimento 1- Distância máxima entre os nós e as estações base igual a 598,95.	68
Figura 42 – Cenário 7, experimento 1 - Distância máxima entre os nós e as estações base igual a 1098,95.	69
Figura 43 – Cenário 9, experimento 1 - Comportamento do DER em função do aumento do tamanho do pacote e da redução do intervalo médio de envio de pacotes para a verificação completa de colisão.	70
Figura 44 – Experimento 9 - Comportamento do DER em função do aumento do tamanho do pacote e da redução do intervalo médio de envio de pacotes para a verificação padrão de colisão.	70
Figura 45 – Frequência de valores da métrica NEC para a verificação completa de colisão.	71
Figura 46 – Frequência de valores da métrica NEC para a verificação padrão de colisão.	72

Lista de tabelas

Tabela 1 – Configurações iniciais para monitoramento do ambiente de desastre. . .	36
Tabela 2 – DER e número de pacotes enviados obtidos para as configurações da Tabela 1.	36
Tabela 3 – Configurações para monitoramento do ambiente de desastre para 48 nós Parte-1.	38
Tabela 4 – Configurações para monitoramento do ambiente de desastre para 48 nós Parte-2.	38
Tabela 5 – Parâmetros de saída para as configurações utilizando 48 nós descritas na Tabela 3.	38
Tabela 6 – Parâmetros de saída para as configurações utilizando 48 nós descritas na Tabela 3. utilizando a verificação completa de colisão	38
Tabela 7 – Configurações para monitoramento do ambiente de desastre para o simulador loraDir.py	39
Tabela 8 – Parâmetros de saída para as configurações da Tabela 7 para a verificação padrão de colisão.	40
Tabela 9 – Parâmetros de saída para as configurações da Tabela 7 para a verificação completa de colisão.	40
Tabela 10 – Parâmetros de saída para a configuração 3.3 para monitoramento do ambiente de desastre (aumentando ainda mais o intervalo médio de envio de pacotes).	42
Tabela 11 – Configurações v4 para monitoramento do ambiente de desastre.	42
Tabela 12 – Variação das saídas em função do aumento da distância entre as estações base para a configuração 3.4.1.	45
Tabela 13 – Variação das saídas em função do aumento do número de redes LoRa para a configuração 3.3.1.	48
Tabela 14 – Variação das saídas em função do aumento do número de redes LoRa para a configuração 3.4.1.	48
Tabela 15 – Baseline para simular o ambiente de monitoração de desastres.	48
Tabela 16 – Descrição dos ataques de disponibilidade na rede LoRa.	50
Tabela 17 – Conjunto de experimentos do Cenário 1.	51
Tabela 18 – Conjunto de experimentos do Cenário 2.	52
Tabela 19 – Comportamento do NECx24 em função do aumento de nós da rede. . .	59
Tabela 20 – Comportamento do NEC * 24 em função da redução do intervalo médio de envio de pacotes.	62
Tabela 21 – Cenário 8, experimento 1 - Comportamento das métricas DER e NEC em função da modificação da estrutura do pacote.	69

Tabela 22 – Cenários, os quais o DER atingiu valores menores que 0,95 ou menores que 0,50.	73
--	----

Lista de abreviaturas e siglas

IoT	Internet of Things
CIA	Confidentiality, Integrity, Availability
DoS	Denial of Service
LPWAN	Low-Power, Wide-Area Network
LoRa	Long Range
TP	Transmission Power
CF	Carrier Frequency
SF	Spreading Factor
BW	Bandwidth
CR	Coding Rate
DER	Data Extraction Rate
NEC	Network Energy Consumption
BS	Base Station
RSSI	Received Signal Strength Indication
FEC	Forward Error Correction
CRC	Cyclic Redundancy Check

Sumário

1	INTRODUÇÃO	13
1.1	Objetivos	14
1.1.1	Objetivo geral	14
1.1.2	Objetivos específicos	14
1.2	Organização do Trabalho	15
2	REVISÃO BIBLIOGRÁFICA	16
2.1	Segurança da Informação	16
2.2	Ataques de Negação de Serviço	17
2.3	Internet das Coisas (IoT)	18
2.4	LoRa	20
2.5	Trabalhos Correlatos	22
3	DESENVOLVIMENTO	26
3.1	LoRasim	26
3.1.1	Entradas	27
3.1.2	Saídas	28
3.1.3	<i>Scripts</i>	29
3.1.4	Instalação e funcionamento	30
3.2	Ambiente de monitoramento	32
3.2.1	Definição do cenário de desastre	33
3.2.2	Validação experimental dos parâmetros - Número de nós, número de estações base e intervalo médio de envio de pacotes	34
3.2.3	Validação experimental dos parâmetros - número de estações base e intervalo médio de envio de pacotes para 48 nós	36
3.2.4	Validação experimental dos parâmetros - NEC para 48 nós	39
3.2.5	Validação experimental dos parâmetros - distância entre as estações base	43
3.2.6	Validação experimental dos parâmetros - número de redes LoRa	47
3.3	Ataques de disponibilidade em redes LoRa	49
3.3.1	Cenário 1 - Aumento do número de nós	50
3.3.2	Cenário 2 - Redução do intervalo médio de envio de pacotes	52
3.3.3	Cenário 3 - Diminuição do número de estações base	53
3.3.4	Cenário 4 - Aumento do número de nós e redução do intervalo médio de envio de pacotes	53
3.3.5	Cenário 5 - Diminuição do número de estações base e intervalo médio de envio de pacotes	53

3.3.6	Cenário 6 - Diminuição do número de estações base e aumento do número de nós	54
3.3.7	Cenário 7 - Aumento da distância máxima entre os nós e as estações base .	54
3.3.8	Cenário 8 - Aumento do tamanho do pacote	54
3.3.9	Cenário 9 - Aumento do tamanho do pacote e diminuição do intervalo médio de envio de pacotes	55
4	RESULTADOS	56
4.1	Cenário 1 - Aumento do número de nós	56
4.2	Cenário 2 - Redução do intervalo médio de envio de pacotes	59
4.3	Cenário 3 - Diminuição do número de estações base	62
4.4	Cenário 4 - Aumento do número de nós e redução do intervalo médio de envio de pacotes	63
4.5	Cenário 5 - Diminuição do número de estações base e intervalo médio de envio de pacotes	65
4.6	Cenário 6 - Diminuição do número de estações base e aumento do número de nós	66
4.7	Cenário 7 - Aumento da distância máxima entre os nós e as estações base	67
4.8	Cenário 8 - Aumento do tamanho do pacote	69
4.9	Cenário 9 - Aumento do tamanho do pacote e diminuição do intervalo médio de envio de pacotes	70
4.10	Discussão sobre os resultados	71
5	CONCLUSÃO	75
	REFERÊNCIAS	76

1 Introdução

O desenvolvimento de novos dispositivos utilizando o conceito de Internet das Coisas (IoT), possibilitou o uso dessa tecnologia para diversas áreas, como a de negócios, medicina, segurança, agricultura, transporte e entretenimento. A ideia básica desse conceito é utilizar uma variedade de objetos ou dispositivos, como identificadores por radiofrequência, sensores, atuadores, câmeras e *smartphones*, que por meio de técnicas de endereçamento, são capazes de interagir de forma a cooperar com o alcance de objetivos comuns (L.ATZORI, 2010).

Uma aplicação importante de IoT é o uso de sistemas de monitoramento, prevenção e alerta de desastres, tanto os naturais, quanto os provocados por falha humana. O monitoramento de atividades vulcânicas (AWADALLAH; MOURE; TORRES-GONZALEZ, 2019), o sistema de alerta antecipado de terremotos (RAVI, 2016) e o sistema de monitoramento online, que permite a detecção de incêndios em florestas (MIRIYALA et al., 2018), são exemplos desta aplicação. Os dispositivos de detecção e sistemas de alerta não podem impedir que um desastre natural aconteça, entretanto, são essenciais para a preparação da população e acionamento de medidas de contingência, com a finalidade de conter o nível das consequências do desastre (BUDHOLIYA, 2018).

LoRa (*Long Range*) é uma tecnologia de rádio frequência para transceptores, que utiliza técnicas de modulação de espalhamento espectral (BOR; VIDLER; ROEDIG, 2016), logo, utiliza mais largura de banda para a transmissão do que a mensagem original necessita e mantém a potência do sinal, tornando-o mais difícil de ser interceptado ou obstruído (INSTRUMENTS, 2020). A tecnologia LoRa foi desenvolvida pela Semtech para dispositivos IoT e aplicações, as quais necessitam coletar milhares de leituras de sensores espalhados em regiões diferentes, inclusive as de difícil acesso (BOR; VIDLER; ROEDIG, 2016), dessa forma, pode ser utilizada para detecção e prevenção de desastres.

O protocolo LoRaWAN define a arquitetura e as regras de comunicação sobre a tecnologia LoRa. LoRaWAN é uma rede de área ampla de baixa potência conhecida como LPWAN (*Low-Power, Wide-Area Network*) (LORAALLIANCE, 2019). Os transceptores empregados em redes LPWAN possuem grande alcance operacional, baixo consumo de energia, tecnologia barata e altamente escalável (MIKHAYLOV et al., 2019).

Devido à importância da detecção de desastres atualmente, faz-se necessário que esses dispositivos possuam mecanismos de segurança para garantir que seu serviço seja ininterruptível e a comunicação seja segura. Dessa forma, o principal problema envolvido nesse trabalho é avaliar o impacto causado por ataques de negação de serviço em dispositivos LoRa. Esses são conhecidos como ataques DoS (*Denial of Service*, em inglês)

e são caracterizados pela tentativa de um invasor de limitar ou impedir o acesso de recursos ou serviços por seus usuários legítimos (LAU et al., 2000). Esse ataque pode ser realizado por meio da inundação de um determinado *host* por um grande volume de tráfego, até que o mesmo não consiga responder requisições autênticas ou por *exploits* em vulnerabilidades do software ou sistema operacional do alvo (HUSSAIN; HEIDEMANN; PAPADOPOULOS, 2003).

Ataques DoS podem ser um meio efetivo de impedir a disponibilidade dos recursos fornecidos por sistemas. A suspensão do funcionamento de sistemas de alerta em determinado momento, pode impedir que os riscos do desastre sejam mitigados ou garantir que o desastre causado por falha humana aconteça. Espera-se que ao aplicar técnicas de inundação em um ambiente experimental de rede LoRaWAN que possui configurações para monitorar um alvo de potencial desastre natural, seja possível avaliar o impacto negativo de ataques de negação de serviço em dispositivos LoRa e provar a necessidade da melhoria da segurança dessa tecnologia para uso no contexto de detecção e alerta de desastres.

1.1 Objetivos

1.1.1 Objetivo geral

O objetivo geral deste trabalho é avaliar o impacto de ataques DoS em dispositivos LoRa no contexto de monitoramento e alerta de desastres, aplicando técnicas de inundação de rede em um ambiente de simulação de rede LoRaWAN.

1.1.2 Objetivos específicos

- Investigar a tecnologia LoRa no contexto de detecção e alerta de desastres.
- Estudar e desenvolver uma configuração de rede para monitorar o alvo de interesse.
- Investigar e aplicar técnicas de DoS em um ambiente de simulação de rede LoRa para degradar o desempenho da rede.
- Avaliar o impacto causado na rede após a aplicação de diferentes cenários para sobrecarregar a rede.
- Demonstrar o impacto causado em dispositivos LoRa por ataques DoS.

1.2 Organização do Trabalho

Este trabalho é organizado da seguinte forma. O Capítulo 2 aborda conceitos básicos como Segurança da Informação, Ataques DoS, IoT e LoRa, também descreve trabalhos correlatos ao trabalho desenvolvido. O Capítulo 3 descreve o simulador LoRaSim, que foi utilizado na execução de todos os experimentos deste trabalho, além disso, detalha informações sobre o ambiente de monitoramento escolhido e as simulações executadas para selecionar as configurações ideais para simular o monitoramento de determinado ambiente, por fim detalha todos os cenários propostos para degradar o desempenho a rede, ou seja, os experimentos executados para avaliar o impacto de ataques DoS na rede de dispositivos LoRa. O Capítulo 4 apresenta os resultados dos experimentos executados no final do Capítulo 3. O Capítulo 5 traz as conclusões deste trabalho e trabalhos futuros.

2 Revisão Bibliográfica

As primeiras quatro seções deste capítulo descrevem conceitos básicos utilizados na construção deste trabalho, necessários para a compreensão do mesmo. A seção 2.5 apresenta trabalhos correlatos ao trabalho desenvolvido.

2.1 Segurança da Informação

O conceito de segurança da informação refere-se aos processos, metodologias e estratégias, que são projetados e implementados para proteger informações impressas, eletrônicas, ou qualquer outro formato de informações ou dados confidenciais, privados e sensíveis contra acesso não autorizado, uso, uso indevido, divulgação, destruição, modificação ou interrupção da disponibilidade (INSTITUTE, 2020).

Os programas de segurança da informação são construídos a fim de alcançar os objetivos de preservar os três pilares da segurança da informação, confidencialidade, integridade e disponibilidade, conhecidos como tríade CIA (do acrônimo em inglês para *Confidentiality, Integrity and Availability*) (STALLINGS, 2010). Esses, podem ser descritos como:

- Confidencialidade – É o princípio que garante que apenas aqueles com direitos e privilégios para acessar as informações possam fazê-lo, ou seja, quando as informações são protegidas da divulgação ou exposição à indivíduos ou sistemas não autorizados. Exemplos de quebra da confidencialidade seriam um funcionário jogar um documento contendo informações críticas no lixo sem triturar o papel e outro funcionário, o qual não pode ter acesso à informação, recolher e ler o papel ou um *hacker* invadir um banco de dados interno de uma organização e roubar informações confidenciais sobre clientes (WHITMAN; MATTORD, 2011) (FRUHLINGER, 2020).
- Integridade - É o princípio que garante que as informações estejam completas e não sejam corrompidas, ou seja, deve assegurar que as informações não sejam modificadas ou deletadas de modo impróprio. A deleção ou modificação deve ocorrer apenas de maneira específica, quando necessário e por pessoas autorizadas. Um exemplo de quebra de integridade seria um aluno invadir o sistema de notas da universidade e alterar as informações referentes ao seu histórico (WHITMAN; MATTORD, 2011)(FRUHLINGER, 2020).

- Disponibilidade – É o princípio que garante que usuários autorizados ou sistemas de computador acessem informações sem interferência ou obstrução quando necessário, ou seja, assegura que os sistemas operem prontamente e seus serviços não fiquem indisponíveis para usuários legítimos. Um exemplo clássico de causa de perda de disponibilidade é a ocorrência de um ataque de negação de serviço (WHITMAN; MATTORD, 2011)(WALKOWSKI, 2020).

Além dos princípios da tríade, a segurança da informação abrange outros conceitos, como a autenticidade, responsabilização e o princípio do não-repúdio. A autenticidade consiste em verificar se os usuários são quem dizem ser e se cada mensagem ou entrada recebida pelo destino é de uma fonte confiável. A responsabilização refere-se à possibilidade de rastrear ações de uma entidade, para que suas ações sejam atribuídas exclusivamente a ela (STALLINGS, 2010). O não-repúdio garante que o receptor da comunicação não possa negar o recebimento de uma mensagem ou transação e que o emissor não possa negar o envio de uma mensagem ou transação (GARG, 2020).

Para garantir a segurança da informação é necessário implementar controles administrativos, lógicos e físicos. Os controles administrativos consistem em políticas, procedimentos, padrões e diretrizes escritos e aprovados. Os controles lógicos utilizam softwares e dados para monitorar e controlar o acesso a informações e sistemas, são exemplos desses controles, firewalls, sistemas de detecção de intrusão de rede, criptografia, senhas, listas de controle de acesso, entre outros. Os controles físicos têm o objetivo de monitorar e controlar o ambiente e instalações de computação, como câmeras, trancas, alarmes de fumaça e incêndio, ar-condicionado, cercas etc (DESCONHECIDO, 2019).

2.2 Ataques de Negação de Serviço

Ataques de negação de serviço, ou ataques DoS (do acrônimo em inglês para *Denial of Service*) consistem na tentativa de impedir usuários legítimos de acessar serviços referentes a sistemas de informação, dispositivos ou outros recursos de rede, por meio de ações maliciosas que tornam esses recursos indisponíveis. Os serviços afetados podem incluir e-mails, sites, contas online (por exemplo, bancos) ou outros serviços que dependem do computador ou rede afetada (CISA, 2019).

Uma abordagem clássica de ataques DoS baseia-se em inundar o *host* ou rede de destino com tráfego até que o mesmo não consiga responder requisições autênticas ou trave impedindo o acesso ou uso por seus usuários. Exemplos dessa abordagem são *Buffer Overflow attacks*, *ICMP Flood* e *SYN Flood* (CISA, 2019)(PALOALTONETWORKS, 2020) e podem ser definidos como:

- Buffer Overflow Attacks - Esse ataque consiste em enviar mais tráfego para um de-

terminado endereço de rede, do que o sistema ou *buffer* foi construído para suportar (PALOALTONETWORKS, 2020).

- ICMP Flood - conhecido também como *Ping-of-Death* ou *Smurf attack*, é um ataque de negação de serviço comum, no qual um invasor tenta sobrecarregar um dispositivo alvo com solicitações de eco ICMP (pings). Ao inundar o alvo com pacotes de solicitação, a rede é forçada a responder com um número igual de pacotes de resposta. Isso faz com que o destino se torne inacessível ao tráfego normal (NETSCOUT, 2020).
- SYN Flood - Ocorre por meio do envio repetido de pacotes de solicitação de conexão (SYN), geralmente utilizando endereços IP falsificados. O servidor então responde a cada uma das solicitações de conexão e deixa uma porta aberta pronta para receber a resposta. Enquanto o servidor espera pelo pacote de confirmação final (ACK) da conexão, que nunca chega, o invasor continua a enviar mais pacotes SYN. A chegada de cada novo pacote SYN faz com que o servidor mantenha temporariamente uma nova conexão de porta aberta por um certo período de tempo, dessa forma, as portas ficam indisponíveis para usuários legítimos se conectarem (CLOUDFARE, 2020).

Outros ataques DoS simplesmente exploram vulnerabilidades no software, hardware ou sistema operacional, por meio de entradas, que fazem com que o sistema ou serviço de destino desestabilize ou falhe completamente, de modo que esse não possa ser acessado ou utilizado (PALOALTONETWORKS, 2020).

Embora não exista uma maneira de impedir completamente que ataques de negação de serviço ocorram, existem métodos para reduzir os efeitos do ataque na rede, como contratar serviços de monitoramento de tráfego da rede, os quais detectam anomalias e filtram tráfegos anormais. Além disso, criar planos de contingência e recuperação para garantir a comunicação, mitigação e recuperação eficientes e bem-sucedidas no caso de um ataque. Faz-se necessário também, investir em soluções e boas práticas de segurança para os dispositivos finais conectados à rede, como instalação e atualização de antivírus, instalação e configuração de firewall para filtrar o tráfego, controle de acesso, entre outros (CISA, 2019).

2.3 Internet das Coisas (IoT)

Internet das Coisas conhecida como IoT (do acrônimo em inglês para *Internet of Things*), consiste em um sistema composto por uma variedade de objetos ou dispositivos interconectados, como sensores, câmeras, atuadores, identificadores por radiofrequência, *smartphones*, que possuem a capacidade de transferir dados por uma rede, utilizando técnicas de endereçamento, para atingir um objetivo em comum, sem exigir muitas interações

de humano para humano ou humano para computador (L.ATZORI, 2010)(ROUSE, 2016). O conceito de tecnologia IoT é aplicado em diversas áreas como indústria, medicina, agricultura, agropecuária, *smart homes*, transporte, entretenimento, detecção de desastres, etc.

Um ecossistema IoT funciona da seguinte forma, primeiramente, os dispositivos (processadores, sensores e hardware de comunicação) coletam os dados do ambiente em que são utilizados. Em seguida, compartilham os dados coletados com um *gateway* IoT ou outro dispositivo de borda. A partir desse *gateway* os dados são enviados para a nuvem, para serem analisados e utilizados conforme a necessidade da aplicação. Esses dispositivos também podem se comunicar com outros dispositivos da rede e agir com base nas informações que recebem uns dos outros. Os dispositivos fazem a maior parte do trabalho sem intervenção humana, embora as pessoas possam interagir com os dispositivos para configurá-los, dar-lhes instruções ou acessar os dados (ROUSE, 2016). O ecossistema IoT está representado na Figura 1.

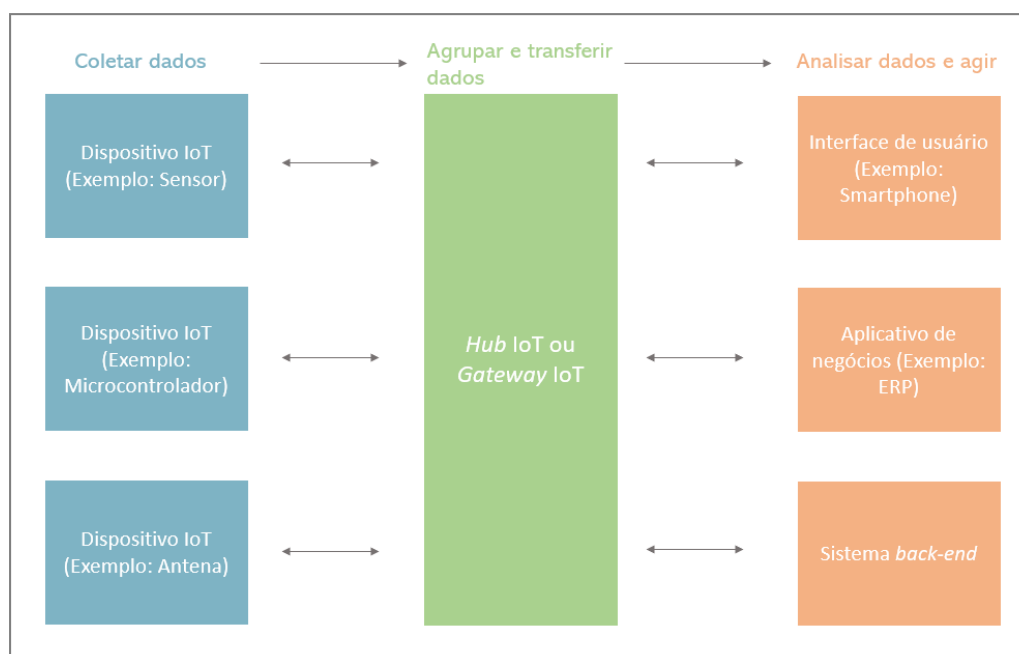


Figura 1 – Ecossistema IoT. Fonte: adaptada de (ROUSE, 2016).

A Internet das Coisas proporciona alta disponibilidade das informações, ou seja, é possível acessá-las de diversos lugares e dispositivos, a qualquer momento. Além disso, aprimora a comunicação entre dispositivos conectados, automatiza tarefas manuais, melhora a qualidade dos serviços e reduz a necessidade de intervenção humana. Entretanto, quanto maior o número de dispositivos conectados, mais informações são compartilhadas entre os dispositivos, dessa forma, torna-se mais difícil coletar e gerenciar um número alto de dispositivos e informações e a possibilidade de um invasor roubar informações confidenciais, tornar o serviço indisponível por meio de um ataque de negação de serviço ou

comprometer toda a rede por meio de uma vulnerabilidade também aumentam (ROUSE, 2016).

2.4 LoRa

LoRa (*Long Range*) é uma tecnologia de rádio frequência para transceptores desenvolvida pela Semtech, que utiliza técnicas de modulação de espalhamento espectral (BOR; VIDLER; ROEDIG, 2016), logo, utiliza mais largura de banda para a transmissão do que a mensagem original necessita e mantém a potência do sinal, tornando-o mais difícil de ser interceptado ou obstruído (INSTRUMENTS, 2020). A vantagem do LoRa está na capacidade de longo alcance da tecnologia. Um único *gateway* ou estação base pode cobrir cidades inteiras ou centenas de quilômetros quadrados. O alcance depende dos ambientes de implantação e suas obstruções (LORAALLIANCE, 2015).

LoRaWAN define o protocolo de comunicação e a arquitetura do sistema para a rede, enquanto a camada física LoRa permite o *link* de comunicação de longo alcance. LoRaWAN é uma rede de área ampla de baixa potência conhecida como LPWAN (*Low-Power, Wide Area Network*) (LORAALLIANCE, 2015)(LORAALLIANCE, 2019). As redes LoRaWAN utilizam a topologia de estrela e são compostas por dispositivos ou nós (*end-points, end-devices*), estações-base (*gateways*), servidores de rede e servidores de aplicação (JUNIOR, 2016). A arquitetura básica de uma rede LoRa está representada na Figura 2.

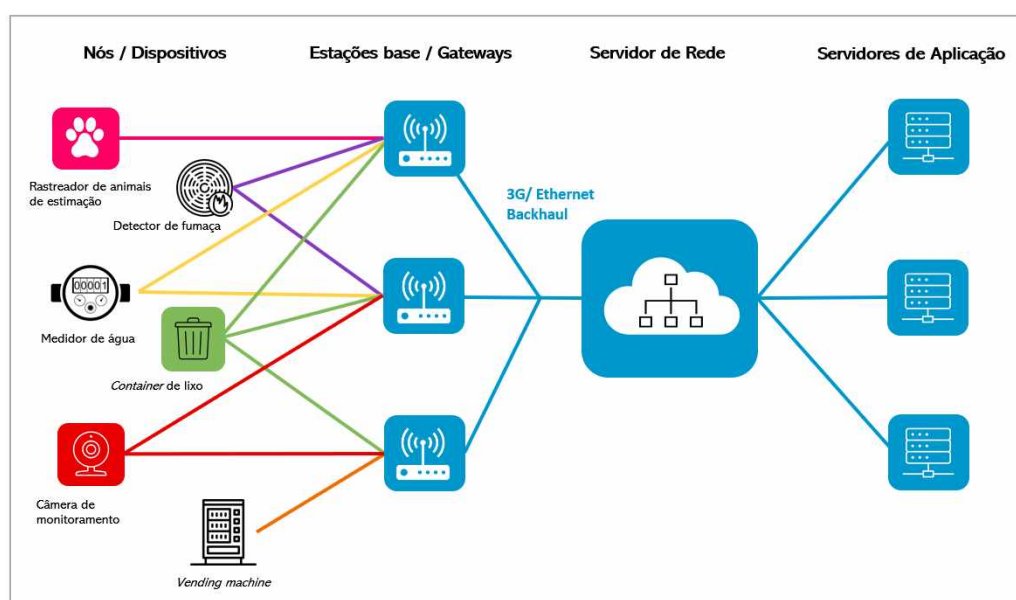


Figura 2 – Ecossistema LoRa. Fonte: adaptada de (LORAALLIANCE, 2019).

Os dados transmitidos por um nó (sensor, leitor) normalmente são recebidos por vários *gateways*, ou seja, os nós não estão associados a um *gateway* específico. Cada *ga-*

teway encaminha o pacote recebido do nó para o servidor de rede baseado em nuvem, por meio de um *backhaul* (celular, Ethernet, satélite ou Wi-Fi). O servidor de rede é responsável por gerenciar as informações enviadas pelos *gateways*, filtrar os pacotes redundantes, executar verificações de segurança e realizar a otimização da taxa de dados. O servidor de rede envia os pacotes ao servidor de aplicação, o qual executa ações específicas de acordo com as informações recebidas (LORAALLIANCE, 2015)(JUNIOR, 2016).

Outra característica importante de dispositivos que utilizam a tecnologia LoRa é o baixo consumo de energia. Isso ocorre, devido ao comportamento assíncrono dos nós, que se comunicam apenas quando possuem dados prontos para enviar, ou seja, são orientados a eventos ou programados. Esse método é conhecido como Aloha e é utilizado no protocolo LoRaWAN (LORAALLIANCE, 2015). Existem três classes de dispositivos LoRa, que podem ser descritas como:

- Classe A: Sensores alimentados por bateria – Permitem comunicação bidirecional, cada janela de transmissão *uplink* é seguida de duas janelas de recepção *downlink* de comprimento fixo, ou seja, os nós só podem receber dados da rede logo após a transmissão em janelas de tempo pré determinadas (JUNIOR, 2016)(ES; VRANKEN; HOMMERSOM, 2018).
- Classe B: Atuadores – Permitem comunicação bidirecional, a recepção (*downlink*) ocorre em janelas de recepção agendadas, que funcionam com base em uma referência de tempo nos avisos (*beacons*) transmitidos, logo essas janelas permitem que a rede entre em contato com o nó em horários específicos (JUNIOR, 2016)(ES; VRANKEN; HOMMERSOM, 2018).
- Classe C: Dispositivos finais bidirecionais com *slots* máximos de recepção – possuem janelas de recepção continuamente abertas, que são fechadas apenas durante as transmissões (LORAALLIANCE, 2015).

O *gateway* deve ter uma capacidade alta para receber mensagens de um grande volume de nós, essa capacidade é alcançada utilizando a técnica de taxa de dados adaptável e um transceptor multi-modem, multicanal no *gateway* para que mensagens simultâneas em vários canais possam ser recebidas. Os fatores críticos que afetam a capacidade do *gateway* são o número de canais utilizados simultaneamente, a taxa de dados (tempo no ar), o comprimento da carga útil do pacote e a frequência de transmissão dos nós (LORAALLIANCE, 2015).

LoraWAN utiliza duas camadas de segurança, uma de rede e uma de aplicação. A segurança de rede garante a autenticidade dos nós da rede, enquanto a camada de aplicação garante que a operadora da rede não tenha acesso aos dados do aplicativo do

usuário final. O algoritmo de criptografia AES é utilizado e a troca de chaves é realizada utilizando um identificador IEEE EUI64 (LORAALLIANCE, 2015).

2.5 Trabalhos Correlatos

Esta seção apresenta trabalhos correlatos ao projeto desenvolvido. Alguns trabalhos exploram a utilização de soluções e arquiteturas providas pela tecnologia LoRa para monitoramento de desastres, como em (ADNAN et al., 2018) e (RAGNOLI et al., 2020). Um dos trabalhos correlatos investiga a utilização de dispositivos IoT baseados na tecnologia LoRa para comunicações de emergência após desastres, como em (CENTELLES et al., 2019). Além disso, os dois trabalhos (ES; VRANKEN; HOMMERSOM, 2018) e (MIKHAYLOV et al., 2019) demonstram e analisam vulnerabilidades em dispositivos e redes LoRaWAN, as quais podem ser exploradas para causar ataques de negação de serviço.

Adnan et al. (2018) propõe um sistema de rede em malha LoRa para detectar a presença de fogo em florestas e alertar a localização do incêndio por meio do aplicativo Google Maps. Esse sistema, utiliza vários nós compostos por Arduino Uno, módulos LoRa, sensor de umidade e temperatura DHT 11 e sensor MQ2.

Os autores concluem que dentro da floresta, foi possível enviar dados usando um dispositivo LoRa na distância máxima de 500 metros e que o limite de nível RSSI (*Received signal strength indication*) aceitável para enviar dados dentro da floresta deve ser superior a -136 dBm. Se o nível do RSSI for menor que -136 dBm, os dados serão parcialmente ou totalmente perdidos. Além disso, também concluiu-se que o melhor modo de configuração LoRa dentro da floresta é BW (*Bandwidth*) 250 CR (*Coding Rate*) 4/5 SF (*Spreading Factor*) 10 Tx Power 14.

Com a configuração proposta pelos autores, é possível cobrir 10 hectares de floresta, utilizando apenas 4 dispositivos LoRa, e para uma área maior, apenas é necessário aumentar o número de dispositivos LoRa. Para evitar colisões de dados em uma rede de malha com 4 nós, recomendaram um intervalo de transmissão maior que 20200 milissegundos. Esse trabalho se assemelha ao desenvolvido pela busca de configurações ideais de uma rede LoRaWAN para monitorar um alvo de potencial desastre.

Em (RAGNOLI et al., 2020), o autor propõe o desenvolvimento e implementação de um sistema de monitoramento de enchentes baseado na tecnologia LoRa testado em um cenário no mundo real. As informações são armazenadas através de um dispositivo equipado com sensores e um microcontrolador, ligado a um módulo LoRa *wireless* para envio de dados, que são posteriormente processados e armazenados através de uma estrutura web, onde é implementada a função de alarme em caso de inundação. Esse trabalho também relaciona-se ao desenvolvido pelo monitoramento de enchentes.

Outros trabalhos reforçam a necessidade de um sistema de comunicação que não possa ser interrompido após o acontecimento de desastres, ou seja, quando ocorre mal funcionamento de provedores de energia e internet e é necessário optar por outros meios emergenciais de comunicação. Em (CENTELLES et al., 2019) a arquitetura LoRaWAN é utilizada para modelar as configurações necessárias para permitir que os cidadãos da cidade Coquimbo no Chile relatem sua situação às unidades de emergência e autoridades públicas após um terremoto. Nesse trabalho, diversas configurações do sistema foram exploradas a fim de determinar suas características e limitações, para melhor compreender sua escalabilidade e portabilidade para outros ambientes, e para delinear os desafios restantes para fazer o sistema atingir garantias de desempenho específicas.

Es, Vranken e Hommersom (2018) identificam três vulnerabilidades na especificação do protocolo LoRaWAN, as quais podem ser exploradas para causar ataques de negação de serviço. Os modelos foram simulados e analisados utilizando a ferramenta CPNTools que confirmou que as vulnerabilidades de fato podem ser exploradas em ataques DoS.

Na primeira vulnerabilidade, foi identificado que os *beacons* transmitidos por *gateways* para as janelas de recepção da classe de dispositivos B não são protegidos por criptografia ou assinatura digital. Dessa forma, os *beacons* são vulneráveis à manipulação por *eavesdropping* e ataques de repetição. Ao explorar essa vulnerabilidade, é possível dessincronizar as janelas de recepção dos dispositivos Classe B. Isso causará uma negação de serviço para o tráfego de recepção (*downlink*) do dispositivo de classe B. A análise do autor confirma que essa vulnerabilidade ainda está presente na especificação LoRaWAN v1.1.

Na segunda vulnerabilidade, foi identificado que o servidor de rede mantém registro do último *gateway* usado por cada dispositivo para enviar dados. Quando a rede precisa realizar uma transmissão para um nó (*downlink*), essa utiliza o *gateway* que foi usado pela última vez pelo nó para realizar uma transmissão para a rede (*uplink*). Dessa forma, espera-se que o nó ainda esteja ao alcance do *gateway* e seja capaz de receber a transmissão. Quando a rede usa o *gateway* incorreto para o tráfego *downlink*, a transmissão pode não ser recebida pelo nó. Como não há confirmação de recebimento de pacote no protocolo LoRaWAN, a transferência de dados *downlink* pode falhar sem ser conhecida pela rede. É possível explorar essa vulnerabilidade com *eavesdropping* e ataques de repetição. A análise do autor confirma que essa vulnerabilidade ainda está presente na especificação LoRaWAN v1.1.

Na terceira vulnerabilidade, foi identificado que ao utilizar-se o método *Over-the-Air Activation* (OTAA) para conectar um nó na rede pela primeira vez, os detalhes da conexão são trocados utilizando mensagens do tipo *Join-Request* e *Join-Accept*, nesse momento os contextos de segurança são gerados, mas a mensagem do tipo *Join-Accept* é

vulnerável a uma combinação de *eavesdropping* e ataques de repetição. A análise do autor confirma que essa vulnerabilidade foi atenuada na v1.1, mas essa versão inclui um modo de compatibilidade para dispositivos finais v1.0.2, que ainda possuem essa vulnerabilidade.

Em (MIKHAYLOV et al., 2019) o autor realiza um estudo sobre a robustez de LoRaWAN quanto a ataques de energia, ou seja, ataques que visam esgotar a energia dos dispositivos conectados à rede. O autor parte do princípio que o consumo máximo de energia do dispositivo ocorre durante as transmissões, um consumo um pouco menor ocorre durante as recepções e um consumo muito baixo ocorre quando o dispositivo está inativo. Dessa forma, concluiu-se que aumentar o tempo de consumo de um dispositivo LoRa durante as transmissões e recepções comprometerá o consumo de energia do dispositivo. Três meios de aumentar o consumo de energia foram analisados.

O primeiro método consiste na implementação de um ataque DoS. Para garantir a conectividade e lidar com a mobilidade dos dispositivos, um dispositivo pode emitir periodicamente uma solicitação de verificação de *link* ou solicitar uma confirmação de seu pacote de dados. Nesse caso, se nenhuma resposta for recebida, o dispositivo pode presumir que as condições do canal se tornaram mais desafiadoras. Conseqüentemente, pode mudar para uma potência de transmissão ou SF (*Spreading Factor*) mais altos. Qualquer uma das opções aumentará o consumo de energia do dispositivo.

O segundo método é específico para casos em que um dispositivo envia seus dados em modo de confirmação e está configurado para retransmitir pacotes. Caso o dispositivo não receba a confirmação de seu pacote ele tentará retransmitir o pacote várias vezes, consumindo mais energia a cada envio.

No terceiro método, após transmitir dados, o dispositivo do tipo A inicia as janelas para recebimento. O dispositivo apenas verifica se pode detectar o preâmbulo válido do pacote recebido, se não, muda para o estado inativo. No caso, se um preâmbulo for detectado, o dispositivo prossegue com o recebimento do pacote. Como o cabeçalho da camada física (PHDR) de um pacote LoRaWAN não é criptografado e a verificação de integridade da mensagem (MIC) está localizada no final do pacote, mesmo o pacote sendo inválido, o dispositivo gastará energia para recebê-lo completamente e apenas depois de verificar a integridade da mensagem, irá descartá-lo.

Após a definição dos vetores de ataque, experimentos foram realizados para a validação do terceiro método. Os resultados obtidos confirmaram que os ataques de energia no LoRaWAN são possíveis e podem fazer com que o dispositivo afetado perca uma quantidade substancial de energia. Especificamente, dependendo do SF (fator de propagação) do dispositivo, o ataque demonstrado aumentou o consumo total de energia durante um único evento de comunicação de 36% a 576%. É importante ressaltar, que o ataque mostrado não exige que o invasor tenha chaves ou outros dados confidenciais e pode ser executado contra qualquer classe de dispositivos LoRa.

Os artigos (ES; VRANKEN; HOMMERSOM, 2018) e (MIKHAYLOV et al., 2019) apresentados nessa seção assemelham-se ao trabalho desenvolvido, pois validam vulnerabilidades que podem ser exploradas a fim de executar ataques de negação de serviço ou ataques de esgotamento de energia em dispositivos LoRa ou redes LoRaWAN e o trabalho desenvolvido tem como objetivo geral avaliar o impacto causado por ataques DoS em dispositivos LoRa no contexto de monitoramento e alerta de desastres.

A principal diferença do trabalho desenvolvido para os trabalhos correlatos está na avaliação do impacto de diversos cenários para degradar o desempenho da rede. Além disso, a escolha do ambiente de monitoramento de desastre, o *baseline* proposto para monitorar o mesmo e o simulador escolhido para executar os experimentos e validar os cenários são outros itens que não foram abordados nos trabalhos anteriores.

3 Desenvolvimento

Este capítulo aborda o desenvolvimento do trabalho. Primeiramente, a seção 3.1 descreve o simulador utilizado para realizar todos os experimentos deste trabalho. Na seção 3.2, apresenta-se o ambiente de desastre escolhido e a validação dos parâmetros de entrada da rede para simular o monitoramento de determinado ambiente. Por fim, os cenários para degradar a disponibilidade da rede são descritos na seção 3.3.

3.1 LoRasim

Configurar redes LoRaWAN com muitos nós LoRa físicos e estações base pode ser uma tarefa complicada e de alto custo, caso não seja planejada. Desta forma, o uso de simuladores pode ser útil para avaliar o impacto causado na rede após a aplicação de diferentes cenários para sobrecarregar a rede. Além disso, o simulador pode ser utilizado para testar configurações que melhor se adequam ao ambiente a ser monitorado por dispositivos LoRa, antes que uma implementação real dispendiosa ocorra.

O LoRaSim é um simulador de eventos discretos de construção personalizada. Este, foi desenvolvido por estudantes da universidade Lancaster no Reino Unido em linguagem Python, utilizando as bibliotecas SimPy, Matplotlib e Numpy. A ferramenta permite a simulação de uma rede de N dispositivos LoRa e M estações base em um espaço bidimensional, ou seja, em *layout* de grade ou distribuição aleatória (BOR et al., 2016).

De acordo com Bor et al. (2016), cada dispositivo LoRa possui uma característica de comunicação específica, definida pelos parâmetros de transmissão *Transmission Power* (TP), *Carrier Frequency* (CF), *Spreading Factor* (SF), *Bandwidth* (BW) e *Coding Rate* (CR). O comportamento de transmissão de cada nó é descrito pela taxa média de transmissão de pacotes λ e pela carga útil do pacote B . O comportamento do n -ésimo nó durante uma simulação é, portanto, descrito pelo conjunto $SN_n = \{TP; CF; SF; BW; CR; \lambda; B\}$. Cada estação base LoRa é capaz de receber para um determinado CF vários sinais com diferentes combinações de SF e BW.

As principais métricas utilizadas para medir a escalabilidade e o desempenho da rede são a taxa de extração dos dados, em inglês *Data Extraction Rate* (DER) e o consumo de energia da rede, *Network Energy Consumption* (NEC). A métrica DER representa a proporção de pacotes recebidos pelas estações base para pacotes transmitidos pelos nós da rede durante um período de tempo. A métrica NEC representa o consumo de energia em Joule da rede para extrair um pacote ou mensagem com sucesso

O valor de DER varia de 0 até 1 e depende do número, posição e comportamento

dos nós e estações base. Quanto mais próximo o valor de DER está de 1, mais eficaz é considerada a implementação da rede LoRa. A Equação 3.1 é utilizada para encontrar o valor do DER, sendo PR a quantidade de pacotes recebidos por pelo menos uma estação base durante a execução da simulação e PE a soma do número de pacotes enviados por todos os nós durante o intervalo de tempo da simulação.

$$DER = \frac{PR}{PE} \quad (3.1)$$

O valor do NEC é influenciado pelo número de nós da rede, frequência das transmissões e parâmetros de comunicação do transmissor. O consumo de energia de um nó LoRa depende principalmente do consumo de energia do transceptor. O consumo de energia de transmissão para cada mensagem depende de TP e da duração da transmissão, que é influenciada por SF, BW e CR. Quanto menor o valor de NEC, mais eficiente é considerada a implementação da rede LoRa.

A Fórmula 3.2 descreve o cálculo do NEC em Joules, na qual N é o número de dispositivos da rede, V é a voltagem, definida no *script* loraDir.py como 3.0, PE é o número de pacotes enviados, i é o índice do dispositivo ou nó, TX é o consumo de energia de transmissão de cada dispositivo final em mA. O parâmetro *airtime*, representa o tempo no qual uma transmissão está ocorrendo e depende dos parâmetros SF, CR, BW e do tamanho da carga útil do pacote para cada dispositivo final (SALLUM et al., 2020).

$$NEC = \sum_{i=0}^N (airtime_i * (TX_i)) * V * PE \quad (3.2)$$

3.1.1 Entradas

Segundo a página principal do projeto <<https://www.lancaster.ac.uk/scc/sites/lora/lorasim.html>>, o simulador recebe como entrada os seguintes parâmetros:

- *Nodes* (Nós) – número de nós ou dispositivos da rede LoRa;
- *Avgsend* (Intervalo médio de envio) – intervalo médio de tempo de envio de pacotes em milissegundos;
- *Experiment* (Experimento) – número inteiro que determina com quais configurações de rádio a simulação é executada:
 0. Utiliza as configurações com a taxa de dados mais lenta (SF12, BW125, CR4/8).
 1. Semelhante ao experimento 0, mas utiliza uma escolha aleatória de 3 frequências de transmissão.
 2. Utiliza as configurações com a taxa de dados mais rápida (SF6, BW500, CR4/5).

3. Otimiza a configuração por nó com base na distância do *gateway*.
4. Utiliza as configurações definidas em LoRaWAN (SF12, BW125, CR4/5).
5. Semelhante ao experimento 3, mas também otimiza a potência de transmissão (TP).

É importante ressaltar que o foco do trabalho desenvolvido são redes LoRaWAN, dessa forma, todos os experimentos foram executados utilizando o parâmetro *Experiment* com o valor 4, ou seja, as configurações definidas para LoRaWAN.

- *Simtime* (Tempo de simulação) – tempo total de execução da simulação em milissegundos.
- *Basestations* (Estações base) – número de estações base ou *gateways* utilizados para simular. Esse valor pode assumir os inteiros 1, 2, 3, 4, 6, 8 ou 24.
- *Collision* (Colisão) – pode assumir os valores 0 ou 1. O valor 1 é selecionado para ativar a verificação de colisão completa, 0 para usar uma verificação simplificada (padrão). Ao utilizar a verificação simplificada, duas mensagens colidem quando chegam ao mesmo tempo, na mesma frequência e fator de propagação. A verificação de colisão completa considera o efeito de captura, pelo qual uma das duas mensagens em colisão ainda pode passar dependendo do tempo relativo e da diferença no poder de recebimento.
- *Directionality* (Direcionalidade) - pode assumir os valores 0 ou 1. O valor 1 é selecionado para ativar a antena direcional para os nós.
- *Networks* (Redes LoRa) - número de redes LoRa.
- *Basedist* (Distância entre estações base) - Distância X entre duas estações base.

3.1.2 Saídas

Os principais parâmetros de saída encontrados durante a execução de experimentos utilizando o simulador são:

- *nrCollisions* – número de colisões de pacotes durante o experimento.
- *Energy (in J)* – NEC em Joule.
- *Sent packets* – número de pacotes enviados pelos nós da rede.
- *Received Packets* – número de pacotes recebidos pelas estações base.
- *Lost Packets* – número de pacotes perdidos, ou seja, pacotes que não foram recebidos por nenhuma estação base.

- DER – proporção de pacotes recebidos por pacotes enviados.
- *Packets at BS X* – número de pacotes recebidos pela estação base X (X varia de 1 até 24, de acordo com o número de estações base fornecido como entrada para o simulador).
- *Send to BS [X]* – número de pacotes enviados para a estação base X (X varia de 1 até 24, de acordo com o número de estações base fornecido como entrada para o simulador).
- *DER BS [X]* – DER específico de cada estação base X, ou seja, proporção entre pacotes enviados para a estação base X e recebidos pela estação base X (X varia de 1 até 24, de acordo com o número de estações base fornecido como entrada para o simulador).

Além das saídas descritas anteriormente, o simulador também fornece os valores para *minimum airtime*, Lpl (*path lost*), *symbol time*, *received signal strength indication* (RSSI), distância máxima possível entre os nós e as estações base, coordenadas xy das estações base, coordenadas xy dos nós e informações sobre as colisões.

3.1.3 Scripts

O simulador é composto por quatro *scripts* em Python, que recebem parâmetros de entrada diferentes e retornam parâmetros de saída diferentes. Os scripts são:

- **loraDir.py** : recebe como entrada os parâmetros *nodes*, *avgsend*, *experiment*, *simtime* e *collision* e retorna todos os valores de saída, exceto *Packets at BS X*, *Send to BS [X]* e *DER BS [X]*.
- **loraDirMulBS.py**: recebe como entrada os parâmetros *nodes*, *avgsend*, *experiment*, *simtime*, *basestations* e *collision* e retorna todos os valores de saída, exceto *Energy (in J)*, *Packets at BS X*, *Send to BS [X]* e *DER BS [X]*.
- **directionalLoraIntf.py**: Simula nós com antenas direcionais e recebe como entrada os parâmetros *nodes*, *avgsend*, *experiment*, *simtime*, *basestations*, *directionality*, *networks*, *basedist*, *collision* e retorna todos os valores de saída exceto *Energy (in J)*. Neste *script*, o número de nós recebidos, é o número de nós associados a cada estação base. Dessa forma, o número de nós total, é o número de nós inserido multiplicado pela quantidade de estações base.
- **oneDirectionalLoraIntf.py**: simula estações base com antenas direcionais e recebe como entrada os parâmetros *nodes*, *avgsend*, *experiment*, *simtime*, *basestations*, *directionality*, *networks*, *basedist*, *collision* e retorna todos os valores de saída exceto

Energy (in J). Neste *script*, o número de nós recebidos, é o número de nós associados a cada estação base. Dessa forma, o número de nós total, é o número de nós inserido multiplicado pela quantidade de estações base.

3.1.4 Instalação e funcionamento

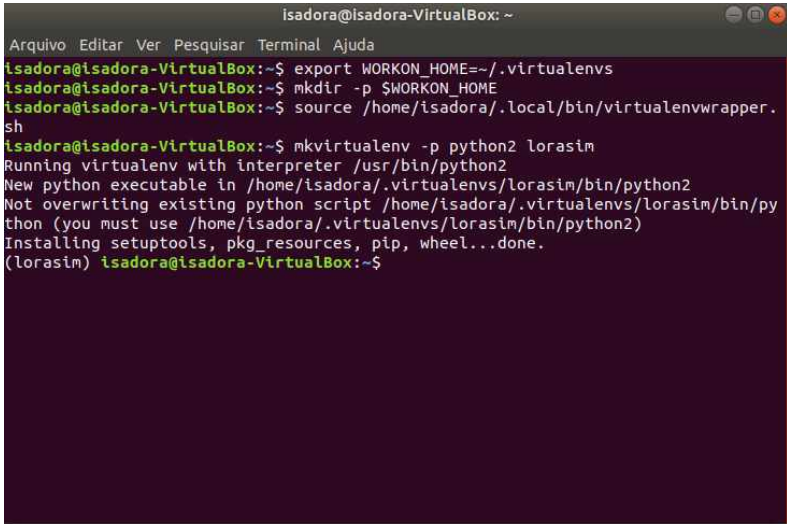
A página <<https://www.lancaster.ac.uk/scc/sites/lora/lorasim.html>> contém um tutorial sobre como utilizar o simulador. Deve-se instalar as aplicações Virtualenv e Virtualenvwrapper utilizando o comando pip, baixar o simulador e em seguida executar os seguintes comandos:

```
$ mkvirtualenv -p python2 lorasim $ export WORKON_HOME= /.virtualenvs.  
$ mkdir -p $WORKON_HOME.  
$ source /usr/local/bin/virtualenvwrapper.sh.  
$ mkvirtualenv -p python2 lorasim.
```

Posteriormente, deve-se acessar o diretório contendo o simulador e instalar os pacotes necessários utilizando o comando:

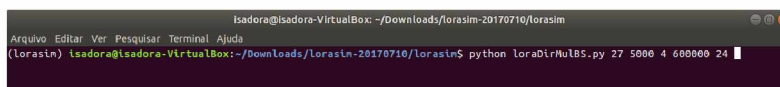
```
$ pip install -r requirements.txt.
```

Para utilizar o simulador é necessário executar os comandos mencionados acima, em seguida o comando python seguido do nome do *script* desejado e os parâmetros de entrada separados por espaços como ilustrado nas Figuras 3 e 4.



```
isadora@isadora-VirtualBox: ~  
Arquivo Editar Ver Pesquisar Terminal Ajuda  
isadora@isadora-VirtualBox:~$ export WORKON_HOME=~/virtualenvs  
isadora@isadora-VirtualBox:~$ mkdir -p $WORKON_HOME  
isadora@isadora-VirtualBox:~$ source /home/isadora/.local/bin/virtualenvwrapper.  
sh  
isadora@isadora-VirtualBox:~$ mkvirtualenv -p python2 lorasim  
Running virtualenv with interpreter /usr/bin/python2  
New python executable in /home/isadora/.virtualenvs/lorasim/bin/python2  
Not overwriting existing python script /home/isadora/.virtualenvs/lorasim/bin/py  
thon (you must use /home/isadora/.virtualenvs/lorasim/bin/python2)  
Installing setuptools, pkg_resources, pip, wheel...done.  
(lorasim) isadora@isadora-VirtualBox:~$
```

Figura 3 – Comandos para executar o simulador.

A terminal window screenshot showing a command being executed. The window title is "Isadora@Isadora-VirtualBox: ~/Downloads/lorasim-20170710/lorasim". The terminal content shows the prompt "(lorasim) Isadora@Isadora-VirtualBox: ~/Downloads/lorasim-20170710/lorasim\$" followed by the command "python loraDirMulBS.py 27 5000 4 600000 24".

```
Isadora@Isadora-VirtualBox: ~/Downloads/lorasim-20170710/lorasim
Arquivo Editar Ver Pesquisar Terminal Ajuda
(lorasim) Isadora@Isadora-VirtualBox: ~/Downloads/lorasim-20170710/lorasim$ python loraDirMulBS.py 27 5000 4 600000 24
```

Figura 4 – Exemplo de comando executado para utilizar o script loraDirMulBS.py.

3.2 Ambiente de monitoramento

Esta seção, descreve o ambiente de monitoramento escolhido para este trabalho, além disso, apresenta os experimentos executados para encontrar as configurações ideais para simular o monitoramento deste cenário de desastres no LoRaSim. Esse processo será chamado de validação experimental dos parâmetros. O diagrama de fluxo da Figura 5 representa os passos realizados para estabelecer os parâmetros da simulação.

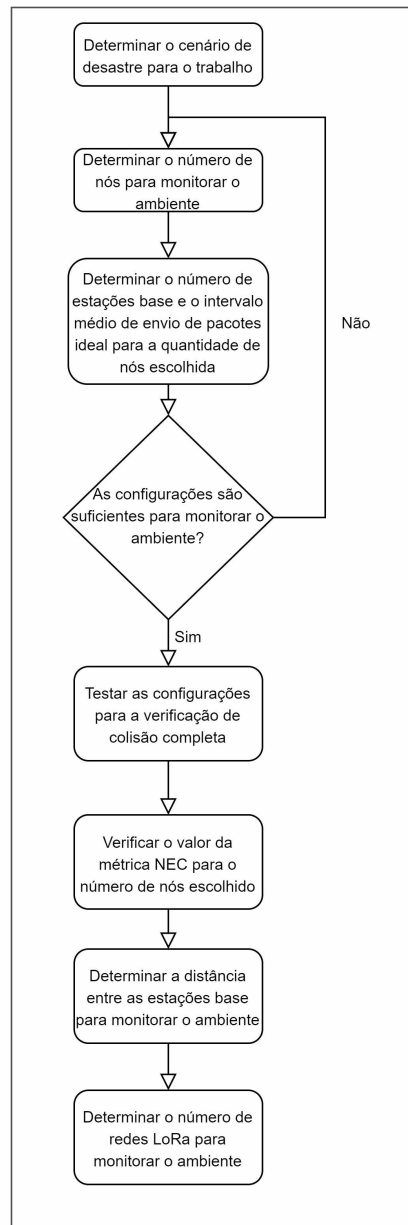


Figura 5 – Processo para encontrar os parâmetros da rede.

O primeiro passo consiste em determinar o cenário de desastre para monitoramento, este passo está descrito na subseção 3.2.1. A Subseção 3.2.2 descreve a escolha inicial do número de nós e os experimentos realizados para determinar o número de esta-

ções base e o intervalo médio de envio de pacotes necessários para suportar a quantidade de nós determinada. A Subseção 3.2.3 também define o número de estações base e o intervalo médio de envio de pacotes, entretanto, em relação à outra quantidade de nós, para um monitoramento mais preciso. Na Subseção 3.2.4, a mesma configuração definida na subseção anterior é utilizada para verificar o desempenho da rede quanto a métrica NEC. As subseções 3.2.5 e 3.2.6 definem, respectivamente, a distância entre as estações base e o número de redes LoRa necessárias para monitorar o ambiente.

3.2.1 Definição do cenário de desastre

Anualmente, a cidade de Belo Horizonte sofre com enxurradas provocadas por chuvas intensas com grandes volumes de precipitação. As chuvas torrenciais causam desabamentos, inundações, deslizamentos de terra, destruição, perda de patrimônio e até mortes. A cidade possui nove pontos críticos de alagamento, espalhados em sete regiões diferentes (ROSA, 2017)(MINAS, 2019).

A bacia hidrográfica do Córrego do Leitão localizada na região Centro-Sul de Belo Horizonte, possui vinte e sete sub-bacias e é propícia para o tipo de desastre descrito acima. Isso ocorre devido às elevadas declividades e diversos trechos de estreitamento do córrego. A região da bacia está representada na Figura 6.

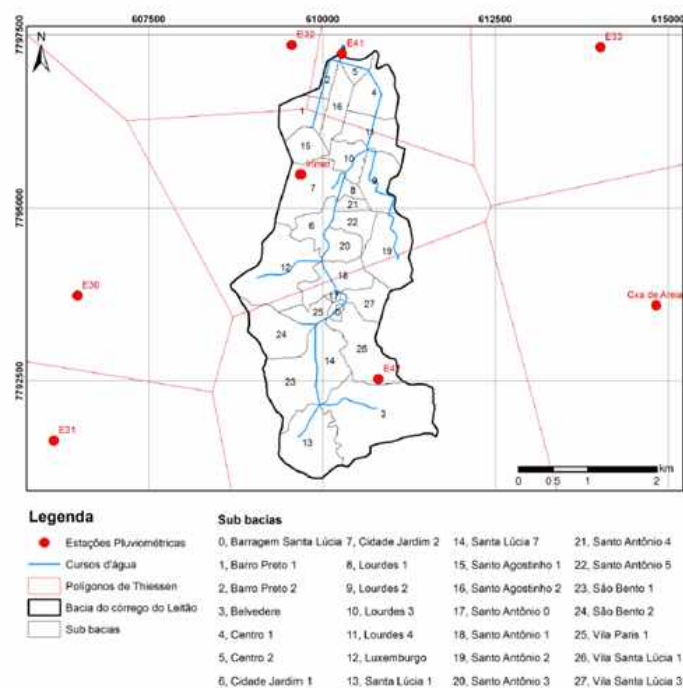


Figura 6 – Bacia do córrego do Leitão. Fonte: extraída de (ROSA, 2017).

A região da bacia hidrográfica do Córrego do Leitão foi selecionada dentre as outras regiões da cidade como ambiente de monitoramento simulado para a execução deste

trabalho. As configurações do ambiente de monitoramento determinadas nessa etapa, foram utilizadas posteriormente como *baseline* para a execução dos cenários de ataque de disponibilidade da rede LoRa.

Com o propósito de aproximar o ambiente de simulação do ambiente real, diversos experimentos foram executados para determinar o número de nós (dispositivos LoRa), o número de estações base necessárias para suportar os nós do ambiente, a distância entre elas, o intervalo médio de envio de pacotes e o número de redes LoRa necessárias para monitorar o ambiente.

3.2.2 Validação experimental dos parâmetros - Número de nós, número de estações base e intervalo médio de envio de pacotes

Nesta subseção, está descrita a validação inicial do número de nós, estações base e o intervalo médio de envio de pacotes. Inicialmente, devido à bacia do Córrego do Leitão possuir vinte e sete sub-bacias espalhadas pela região Centro-Sul de Belo Horizonte, o número de nós utilizados como padrão para determinar os demais parâmetros da simulação foi 27, ou seja, um sensor para medir o nível de água de cada sub-bacia da bacia hidrográfica.

A fim de determinar o número de estações base e o intervalo médio de envio de pacotes para monitorar o ambiente composto por 27 nós, simulações foram realizadas utilizando o LoRaSim, variando o tempo médio de envio de pacotes, começando em 1000ms, em seguida 5000ms e acrescentando os outros valores em 5000ms, até 60000ms. O mesmo experimento foi realizado para cada quantidade possível de estações base proposta pelo simulador (1, 2, 3, 4, 6, 8 e 24) utilizando o *script* loraDirMulBS.py. O tipo de experimento foi fixado com o valor 4 para representar as configurações de rádio de uma rede LoRaWAN, o tempo de simulação como 600000ms, a verificação de colisão utilizada foi a simples (padrão).

Primeiramente, a métrica DER foi utilizada para medir o desempenho da rede após a execução dos experimentos. O simulador posiciona os nós de forma aleatória e as colisões também ocorrem de forma aleatória, por isso, foram realizadas 100 simulações com cada configuração de tempo médio de envio e número de estações base e a média do DER resultante das simulações foi calculada para obter um resultado mais preciso. Os resultados obtidos no experimento descrito nesta subseção estão ilustrados na Figura 7.

De acordo com o gráfico da Figura 7, quanto maior o número de estações base, mais próximo o DER está de 1, ou seja, melhor o desempenho. Por exemplo, nos pontos em que o valor do intervalo médio de envio de pacotes é 1000ms, o valor da métrica DER para 1 estação base é abaixo de 0,50, enquanto o DER para 24 estações base é superior à 0,95. O mesmo efeito ocorre para o intervalo médio de envio de pacotes, ou seja, quanto

maior o intervalo, maior é o desempenho, por exemplo, para 1 estação base o valor da métrica DER quando o intervalo vale 1000ms é de 0,49481, já quando o intervalo é de 60000ms, a métrica obtém o valor 0,97958.

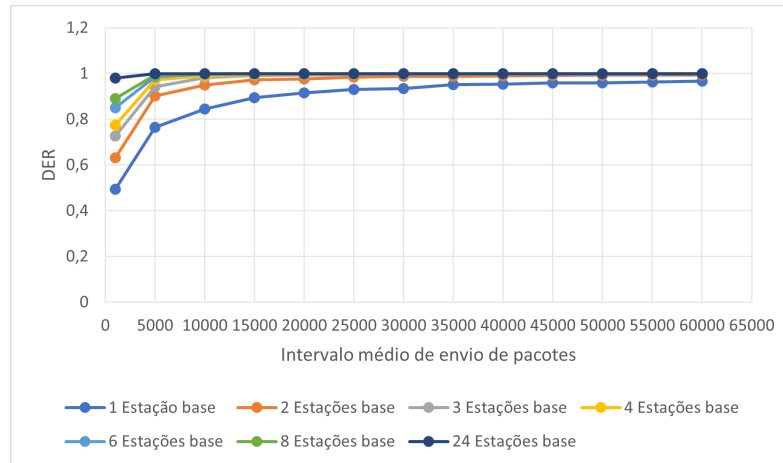


Figura 7 – Variação do DER em função do aumento do intervalo médio de envio para todas as quantidades de estações base.

Aplicações de monitoramento e alerta de desastres podem demandar um DER alto, principalmente nos minutos anteriores ao desastre para que seja possível iniciar as medidas de contingência necessárias. Dessa forma, assume-se um ótimo desempenho da rede quando o DER é superior ao valor 0,95.

Apenas as configurações de parâmetros de entrada (número de estações base e intervalo médio de envio de pacotes) em que a taxa de extração de dados resultante é maior ou igual a 95% e o intervalo médio de envio de pacotes é o menor possível foram selecionadas dentre o conjunto de resultados inicial de experimentos. A Tabela 1 sumariza os cenários onde o critério definido anteriormente foi atingido.

O termo configurações, na primeira coluna da Tabela 1 serve como um identificador para facilitar a descrição do conjunto de parâmetros de entrada da rede e para identificar a saída gerada para determinada entrada, nesse caso apenas o número de estações base e o intervalo médio de envio de pacotes são diferentes para cada configuração. Os demais parâmetros de entrada da rede são iguais para todas as configurações, como o número de nós (27), o tempo de simulação (600000ms) e Colisão (0). A Tabela 2 apresenta o valor dos parâmetros de saída DER e o número de pacotes enviados obtidos no experimento desta seção para as configurações da Tabela 1.

Configurações	Nós	Intervalo médio de envio de pacotes	Experimento	Tempo de simulação	Estações base	Colisão
1.1	27	35000ms	4	600000ms	1	0
1.2	27	15000ms	4	600000ms	2	0
1.3	27	10000ms	4	600000ms	3	0
1.4	27	5000ms	4	600000ms	4	0
1.5	27	5000ms	4	600000ms	6	0
1.6	27	5000ms	4	600000ms	8	0
1.7	27	5000ms	4	600000ms	24	0
1.8	27	1000ms	4	600000ms	24	0

Tabela 1 – Configurações iniciais para monitoramento do ambiente de desastre.

Configurações	DER	Pacotes enviados
1.1	0,951841	458,9
1.2	0,972402	1058,79
1.3	0,982278	1584,08
1.4	0,972054	3090,17
1.5	0,986424	3085,67
1.6	0,993876	3083,81
1.7	0,999428	3082,32
1.8	0,979585	13511,78

Tabela 2 – DER e número de pacotes enviados obtidos para as configurações da Tabela 1.

De acordo com os dados obtidos, quanto maior o número de estações base utilizadas, um intervalo médio de envio de pacotes menor é necessário para garantir um DER maior que 0,95. Além disso, quanto maior o intervalo médio de envio, menor é o número de pacotes enviados, o que pode impactar ou impedir no aviso de algum problema durante o monitoramento. As melhores configurações, ou seja, as que possuem o DER resultante alto, o intervalo médio de envio de pacotes abaixo de 10000ms e o número de pacotes enviados superior à 3000 são as configurações 1.4, 1.5, 1.6, 1.7 e 1.8, que estão destacadas em negrito na Tabela 1.

3.2.3 Validação experimental dos parâmetros - número de estações base e intervalo médio de envio de pacotes para 48 nós

Na subseção 3.2.2 ocorreu uma validação experimental dos parâmetros de entrada número de estações base e intervalo médio de envio de pacotes em relação a uma rede contendo 27 dispositivos ou sensores de monitoramento. As melhores configurações de parâmetros de entrada foram obtidas visando obter um valor alto para a métrica DER e o envio elevado de pacotes. Nessa subseção, o objetivo principal é executar experimentos

semelhantes utilizando as melhores configurações obtidas na subseção anterior, entretanto utilizando 48 nós ou dispositivos na rede.

Em [Ragnoli et al. \(2020\)](#) um sistema de monitoramento de enchentes é implementado utilizando a tecnologia LoRa. O autor divide a presença de água em um determinado ambiente em três zonas diferentes. A primeira, é considerada uma zona de não-alerta, a segunda, uma zona intermediária não crítica e a terceira zona é considerada como crítica de inundação. Essas áreas são delimitadas pelos níveis de limiar onde os sensores estão posicionados, de modo a obter uma correspondência direta entre o nível da água e o alarme acionado. A divisão está representada na Figura 8. Nesse caso, dois sensores ou nós são utilizados para monitorar o ambiente.

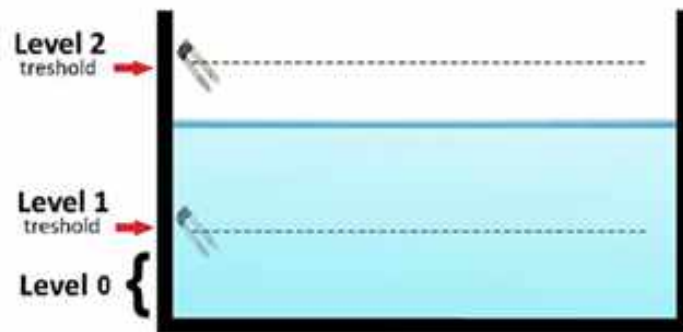


Figura 8 – Particionamento de nível de inundação. Fonte: extraída de ([RAGNOLI et al., 2020](#)).

A fim de obter o número de redes LoRa e a distância máxima entre as estações base necessárias para monitorar o ambiente nos próximos experimentos, o *script* `oneDirectionalLoraIntf.py` foi utilizado nos demais conjuntos de experimentos. Nesse *script*, é necessário fornecer quantos nós estarão associados a cada estação base, apesar das estações base receberem pacotes de todos os nós que estão dentro da região de alcance do sinal. Por isso, o número de nós total foi alterado de 27 para 24, pois o número de nós resultante deve ser múltiplo do número de estações base.

Tendo em vista a arquitetura proposta para monitoramento de enchentes descrita acima, retirada do artigo ([RAGNOLI et al., 2020](#)), o comportamento do *script* `oneDirectionalLoraIntf.py` do LoRaSim e o ambiente de monitoramento, o número de nós utilizados no segundo conjunto de experimentos foi 48, ou seja, dois nós ou sensores para monitorar aproximadamente cada sub-bacia do córrego do Leitão.

A Tabela 3 e a Tabela 4 descrevem o conjunto de parâmetros de entrada das simulações, identificados pela coluna Configurações. Com o intuito de garantir a precisão dos dados foram realizados 100 experimentos para cada uma das configurações. Inicialmente, os parâmetros Experimento, Colisão, Direcionalidade, Redes e Distância entre estações

base não foram modificados. A média do DER e do número de pacotes enviados para cada conjunto de 100 experimentos estão representadas na Tabela 5.

Configurações	Nós totais	Nós por EB	Estações base	Intervalo médio de envio
2.1	48	12	4	5000ms
2.2	48	8	6	5000ms
2.3	48	6	8	5000ms
2.4	48	2	24	5000ms
2.5	48	2	24	1000ms

Tabela 3 – Configurações para monitoramento do ambiente de desastre para 48 nós Parte 1.

Experimento	Tempo de simulação	Colisão	Direcionalidade	Redes LoRa	Distância EB
4	600000ms	0	0	1	100

Tabela 4 – Configurações para monitoramento do ambiente de desastre para 48 nós Parte 2.

Configurações	DER	Pacotes enviados
2.1	0,905350	5493,92
2.2	0,942671	5485,3
2.3	0,958119	5482,96
2.4	0,990526	5482,96
2.5	0,807949	24023,85

Tabela 5 – Parâmetros de saída para as configurações utilizando 48 nós descritas na Tabela 3.

O mesmo conjunto de experimentos foi realizado, entretanto utilizando a verificação completa de colisão para todas as configurações, ou seja, utilizando o parâmetro Colisão com o valor 1. Os resultados obtidos estão representados na Tabela 6.

Configurações	DER	Pacotes enviados
2.1	0,99563	5492,32
2.2	0,998719	5480,37
2.3	0,999193	5503,23
2.4	0,999574	5500,71
2.5	0,999434	24062,36

Tabela 6 – Parâmetros de saída para as configurações utilizando 48 nós descritas na Tabela 3. utilizando a verificação completa de colisão

De acordo com os dados obtidos na Tabela 5 e na Tabela 6, as melhores configurações para monitoramento são a 2.3 e a 2.4. Essas configurações mantêm um DER alto tanto com o parâmetro Colisão padrão, quanto utilizando a verificação de colisão completa. Além disso, utilizam um tempo baixo de intervalo médio de envio de pacotes, os nós enviam muitos pacotes e a rede possui um número alto de estações base para garantir maior robustez.

3.2.4 Validação experimental dos parâmetros - NEC para 48 nós

Os experimentos executados nas subseções 3.2.2 e 3.2.3 deste trabalho têm como métrica de análise o valor do DER. Com o intuito de avaliar melhor os parâmetros de monitoramento do ambiente, o terceiro conjunto de experimentos foi analisado com base no valor da métrica NEC, ou seja, o consumo de energia da rede LoRa em Joule.

Apenas o *script* loraDir.py fornece a métrica NEC como resultado para as simulações. Nesse *script*, não é possível fornecer o número de estações base da rede, pois apenas uma estação base é utilizada. Dessa forma, para estimar o valor do NEC para cada uma das configurações da Tabela 3, o valor de nós total de cada uma das configurações foi dividido pelo número de estações base. O resultado dessa operação foi usado como quantidade de nós de entrada para o simulador em cada configuração, como representado na Tabela 7.

Os resultados para a métrica DER e NEC utilizando as configurações propostas na Tabela 7 para a verificação padrão de colisão estão na Tabela 8 e para a verificação completa de colisão na Tabela 9. Em ambas as tabelas, a terceira coluna representa o valor do NEC multiplicado pela quantidade de estações base da configuração, para estimar o valor da métrica para 48 nós.

Configurações	Nós	Intervalo médio de envio	Experimento	Tempo de simulação	Colisão
3.1	12	5000	4	600000	0 e 1
3.2	8	5000	4	600000	0 e 1
3.3	6	5000	4	600000	0 e 1
3.4	2	5000	4	600000	0 e 1
3.5	2	1000	4	600000	0 e 1

Tabela 7 – Configurações para monitoramento do ambiente de desastre para o simulador loraDir.py

Configurações	NEC	NEC * Estações base	DER
3.1	197,42	789,68	0,00404
3.2	132,36	794,16	0,03098
3.3	98,69	789,52	0,08238
3.4	33,60	806,4	0,59851
3.5	90,49	2.171,76	0,11393

Tabela 8 – Parâmetros de saída para as configurações da Tabela 7 para a verificação padrão de colisão.

Configurações	NEC	NEC * Estações base	DER
3.1	198,45	793,8	0,06724
3.2	132,23	793,38	0,11215
3.3	99,90	799,2	0,17852
3.4	33,12	794,88	0,68347
3.5	90,34	2.168,16	0,23736

Tabela 9 – Parâmetros de saída para as configurações da Tabela 7 para a verificação completa de colisão.

Devido à redução do número de estações base para um, o valor do DER caiu consideravelmente para as configurações estabelecidas nos experimentos anteriores. Quanto maior o número de nós, maior é o consumo de energia da rede. Para selecionar um DER alto e um NEC estimado baixo foi necessário aumentar o tempo de intervalo médio de envio de pacotes para as configurações 3.3 e 3.4. Os resultados obtidos estão representados nas Figuras 9, 10, 11, 12.

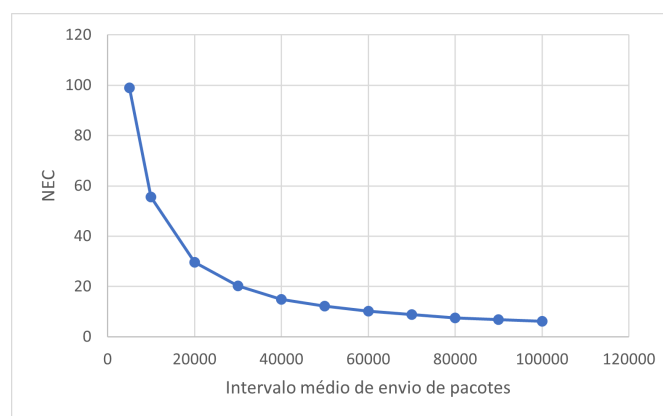


Figura 9 – Variação do NEC em função do aumento do intervalo médio de envio de pacotes para a configuração 3.3.

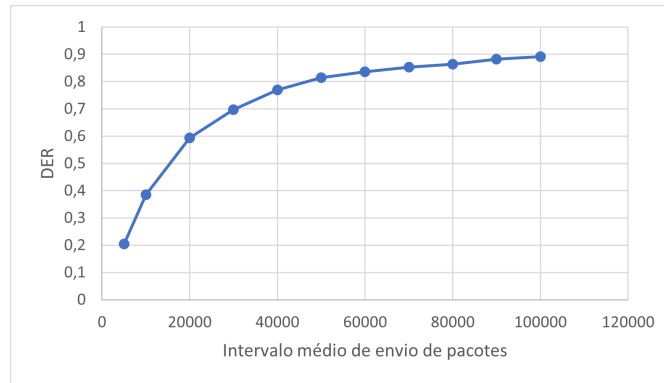


Figura 10 – Variação do DER em função do aumento do intervalo médio de envio de pacotes para a configuração 3.3.

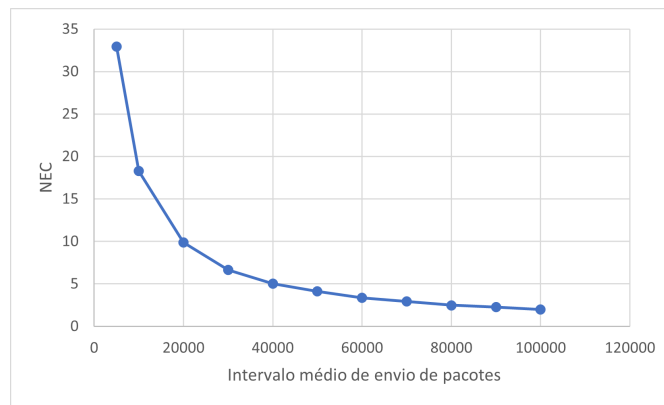


Figura 11 – Variação do NEC em função do aumento do intervalo médio de envio de pacotes para a configuração 3.4.

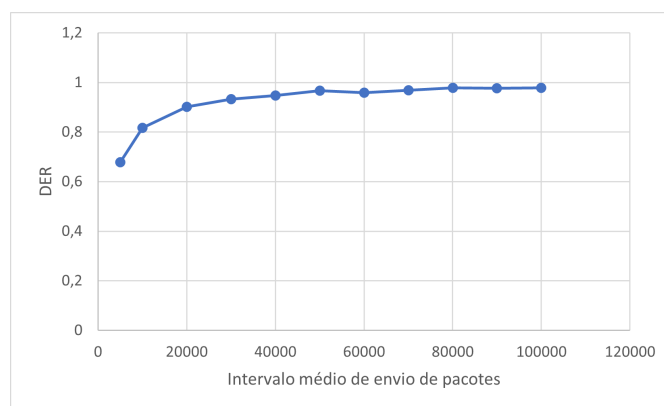


Figura 12 – Variação do DER em função do aumento do intervalo médio de envio de pacotes para a configuração 3.4.

Apesar do aumento do intervalo médio de envio de pacotes para 100000ms a configuração 3.3 ainda obteve um DER inferior a 0,95. Para obter um DER superior a 0,95

no caso desta configuração, foi necessário aumentar o intervalo médio de envio de pacotes até 210000ms, como representado na Tabela 10.

Intervalo médio de envio	NEC	DER
110000ms	5,72603	0,91388
120000ms	5,019199	0,916712
130000ms	4,836398	0,917266
140000ms	4,315849	0,937875
150000ms	4,105193	0,940278
160000ms	3,854494	0,933396
170000ms	3,565494	0,943201
180000ms	3,502819	0,933335
190000ms	3,36006	0,933385
200000ms	3,020572	0,947222
210000ms	2,792506	0,950924
220000ms	2,83603	0,957108
230000ms	2,766392	0,956368
240000ms	2,682825	0,961214
250000ms	2,489578	0,950384

Tabela 10 – Parâmetros de saída para a configuração 3.3 para monitoramento do ambiente de desastre (aumentando ainda mais o intervalo médio de envio de pacotes).

As melhores configurações obtidas no experimento anterior são a 3.4, utilizando o tempo médio de envio de pacotes 50000ms e a configuração 3.3, utilizando o tempo médio de envio de pacotes 210000ms. No caso da configuração 3.4, ao multiplicar o DER por 24 para obter uma média da energia gasta obtém-se 99,36 J para 48 nós. No caso da configuração 3.3 obtém-se um NEC total de 22,34 J. As novas configurações estão representadas na Tabela 11.

Configurações	Nós	Estações base	Intervalo médio de envio de pacotes
3.3.1	48	8	50000
3.4.1	48	24	210000

Tabela 11 – Configurações v4 para monitoramento do ambiente de desastre.

Experimentos foram realizados utilizando 48 nós e o intervalo médio de envio para as configurações 3.3.1 e 3.4.1 no simulador loraDir.py para confirmar os valores da métrica NEC. A configuração 3.3.1 obteve um NEC de 23,81 J enquanto a 3.4.1 obteve um NEC de 97,05. Os valores são semelhantes aos obtidos, ao multiplicar o DER obtido anteriormente pelo número de estações base da configuração, logo, prova-se a linearidade da métrica.

A configuração ideal depende dos requisitos necessários para a aplicação, no caso do monitoramento de desastres, um DER alto e um NEC relativamente baixo indicam um

bom desempenho da rede. Além disso, a aplicação deve ter um tempo médio de envio de pacotes baixo ou médio para que seja possível enviar um número médio de pacotes, logo, ambas as configurações 3.4.1 e 3.3.1 podem ser utilizadas para o ambiente determinado neste trabalho.

Caso os requisitos necessários da aplicação envolvam enviar mais pacotes a configuração 3.4.1 é melhor, porém o custo de implantação devido ao uso do triplo de estações base é maior e o consumo de energia também devido ao intervalo médio de envio de pacotes menor. Caso seja necessário enviar menos pacotes e gastar menos energia a configuração 3.3.1 seria mais adequada.

3.2.5 Validação experimental dos parâmetros - distância entre as estações base

Os experimentos desta seção têm como objetivo encontrar a distância ideal entre as estações base. De acordo com o código dos *scripts* `directionalLoraIntf.py` e `oneDirectionalLoraIntf.py` do LoRaSim, quando o parâmetro Direcionalidade está ativo, o RSSI varia para cada nó, de acordo com a proximidade que este tem com cada uma das estações base. Quanto mais próximo da estação base, maior é a força do sinal e quanto menos próximo ou mais distante, menor é a força do sinal.

O experimento desta seção consistiu em variar a distância das estações base de 100 em 100 para as configurações 3.4.1 e 3.3.1. Os resultados do DER médio das estações base e do número de colisões em função do aumento da distância entre as estações base estão representados graficamente nas Figuras 13, 14, 15 e 16 e na Tabela 12. Cada experimento foi realizado 100 vezes com cada configuração, dessa vez com o parâmetro Direcionalidade igual a 1.

Como visto anteriormente nas especificações do simulador, o DER médio de cada estação base consiste no número de pacotes recebidos pela estação base dividido pelo número de pacotes enviados para determinada estação base. O DER médio de cada estação base pode assumir valores superiores a 1, pois cada estação base pode receber outros pacotes, além dos pacotes destinados para a mesma.

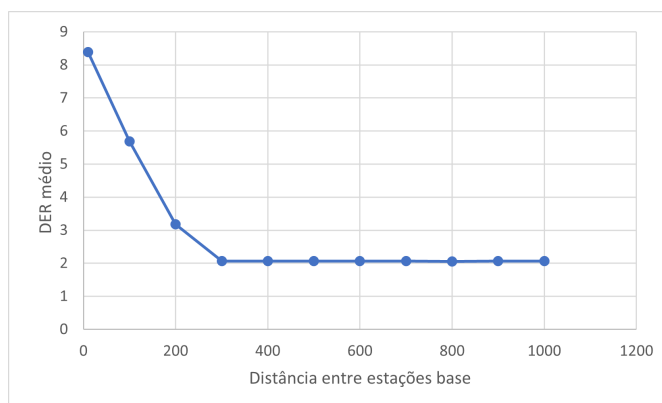


Figura 13 – Variação do DER médio das estações base em função do aumento da distância entre as mesmas para a configuração 3.3.1.

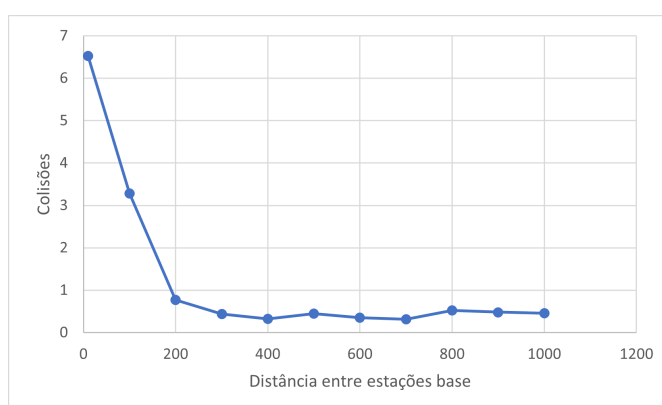


Figura 14 – Variação do número de colisões em função do aumento da distância entre as estações base para a configuração 3.3.1.

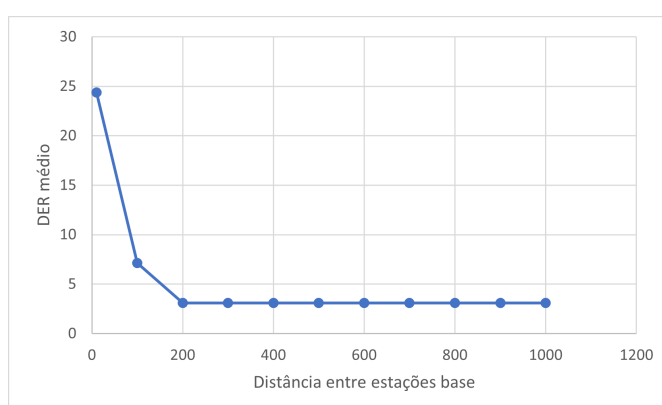


Figura 15 – Variação do DER médio das estações base em função do aumento da distância entre as mesmas para a configuração 3.4.1.

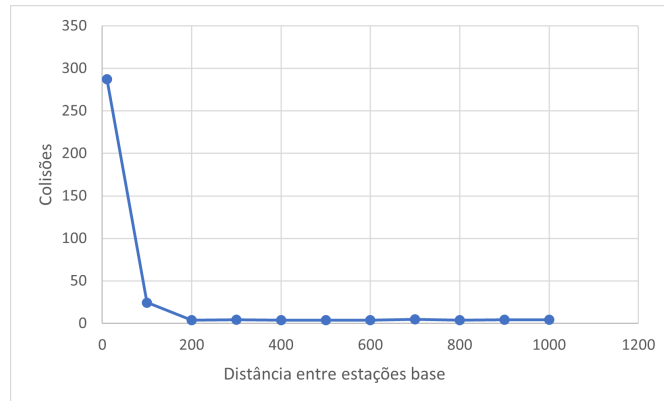


Figura 16 – Variação do número de colisões em função do aumento da distância entre as estações base para a configuração 3.4.1.

Distância entre EBs	DER médio EBs	DER	Colisões
10	24,37857	0,99963	287,38
100	7,140321	0,999596	24,47
200	3,080168	0,999552	3,75
300	3,081537	0,999683	4,48
400	3,082475	0,999653	3,97
500	3,084353	0,999613	3,72
600	3,078306	0,999495	3,88
700	3,085058	0,999495	4,56
800	3,082449	0,999374	3,78
900	3,08216	0,999403	4,34
1000	3,081766	0,999468	4,35

Tabela 12 – Variação das saídas em função do aumento da distância entre as estações base para a configuração 3.4.1.

Estações base muito próximas recebem pacotes de nós que não estão associados às mesmas, conseqüentemente o DER médio das estações base ultrapassa o valor 1 e mais colisões ocorrem, entretanto o parâmetro Direcionalidade não influencia diretamente no DER geral, a não ser que a verificação de colisão utilizada seja a simples (padrão).

O número de colisões diminui significavelmente por causa do aumento da distância das estações base e a média do DER das estações base também, pois cada estação base passa a receber menos pacotes que não são diretamente direcionados para ela. O número não diminui até um pois apenas é possível mudar a distância das estações base na horizontal, enquanto na vertical pares de estações estão sempre perto uma da outra. Quanto ao DER geral (número total de pacotes recebidos independente da estação base dividido pelo número total de pacotes enviados por todos os nós) permanece o mesmo independente da distância das estações base.

As Figuras 17, 18, 19, 20, 21 e 22 geradas pelo próprio LoRaSim, representam visualmente a distribuição dos nós, que estão representados pelos círculos preenchidos menores, as estações base, que estão representadas pelos círculos preenchidos maiores e o alcance do sinal de cada estação base está representado pelas circunferências não preenchidas grandes. Cada cor representa uma estação base, os nós associados a mesma e seu alcance de sinal. A distância entre as estações base varia de uma figura para a outra para as configurações 3.3.1 e 3.4.1.

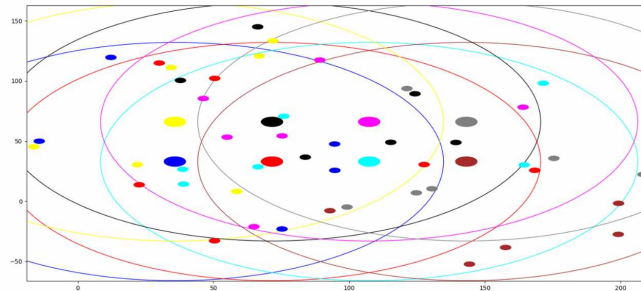


Figura 17 – Distância entre as estações base com o valor 10 para a configuração 3.3.1.

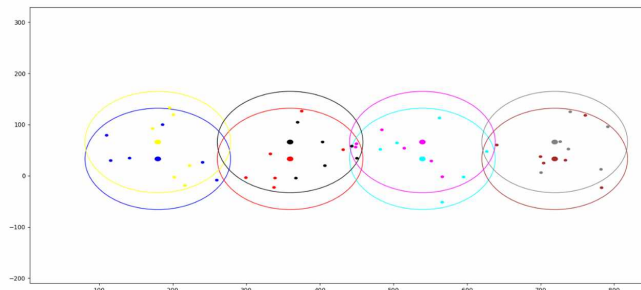


Figura 18 – Distância entre as estações base com o valor 100 para a configuração 3.3.1.

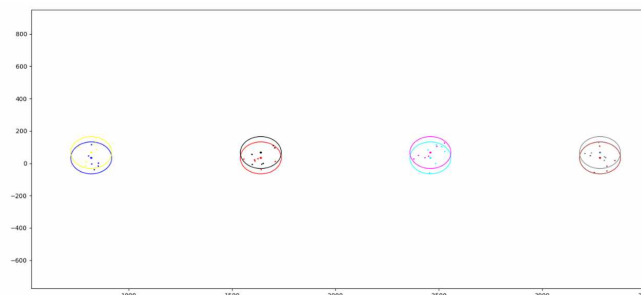


Figura 19 – Distância entre as estações base com o valor 500 para a configuração 3.3.1.

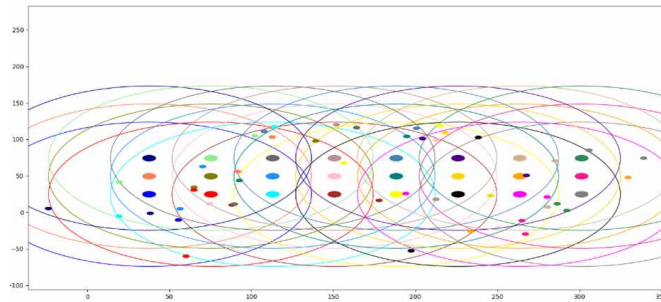


Figura 20 – Distância entre as estações base com o valor 10 para a configuração 3.4.1.

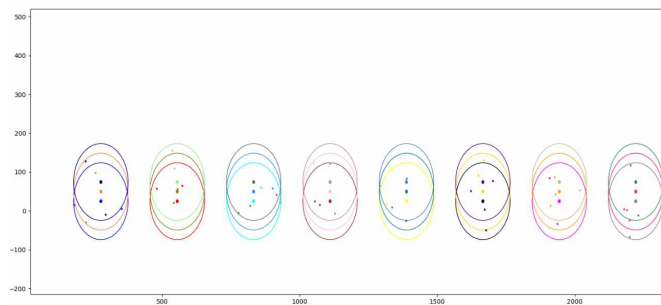


Figura 21 – Distância entre as estações base com o valor 100 para a configuração 3.4.1.

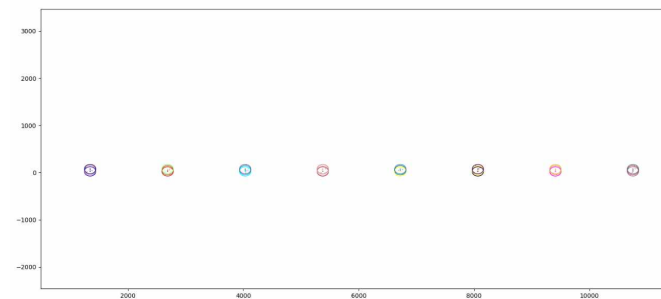


Figura 22 – Distância entre as estações base com o valor 500 para a configuração 3.4.1.

A partir dos gráficos gerados e da Tabela 12 definiu-se o valor 200 como a distância entre as estações base, pois a partir deste valor o número de colisões torna-se pequeno e estável, com pequenas variações entre 3 e 5. Da mesma forma, o DER médio das estações base torna-se estável.

3.2.6 Validação experimental dos parâmetros - número de redes LoRa

Os experimentos finais para determinar as configurações ideais para simular um ambiente de monitoração de desastres envolvem determinar a quantidade de redes LoRa necessárias para o monitoramento. Para isso, foram executadas 100 simulações para cada uma das configurações 3.3.1 e 3.4.1, desta vez com o parâmetro Direcionalidade 1 e com

a distância entre as estações base igual a 200, variando apenas o número de redes LoRa de 1 até 5. Os resultados gerados estão representados nas tabelas 13 e 14.

Nº de redes	DER médio	DER	Colisões
1	3,17051	0,999841	0,9
2	0,998106	0,999636	0,84
3	0,99843	0,999641	0,83
4	0,998121	0,999255	0,76
5	0,999001	0,999696	0,88

Tabela 13 – Variação das saídas em função do aumento do número de redes LoRa para a configuração 3.3.1.

Nº de redes	DER médio	DER	Colisões
1	3,088672	0,999471	4,534653
2	0,997014	0,999706	4,66
3	0,997097	0,99944	4,4
4	0,996628	0,999363	4,58
5	0,997279	0,999565	4,03

Tabela 14 – Variação das saídas em função do aumento do número de redes LoRa para a configuração 3.4.1.

A mudança do número de redes não modificou a métrica DER para nenhuma das configurações, portanto, presume-se que o aumento do número de redes LoRa seria uma medida de contingência caso a primeira rede viesse a falhar em um determinado momento, a outra entraria no lugar, logo o sistema seria mais robusto. A partir da sequência de experimentos executados na seção 3.2 deste trabalho, obteve-se as configurações representadas na Tabela 15.

Nós	Intervalo médio de envio de pacotes	Exp	Tempo de Simulação	Estações base	Colisão	Direcionalidade	Redes LoRa	Distância EBs
48	210000ms	4	600000ms	8	1	1	1	200
48	50000ms	4	600000ms	24	1	1	1	200

Tabela 15 – Baseline para simular o ambiente de monitoração de desastres.

A configuração escolhida como *baseline* para a próxima etapa deste trabalho foi a segunda configuração da Tabela 15. É uma configuração robusta, pois utiliza o número máximo de estações base possíveis e tem apenas 2 nós relacionados a cada estação base e um tempo médio de envio relativamente alto, logo a métrica DER possui um valor ótimo, tanto para o parâmetro colisão padrão, quanto para a verificação completa de colisão.

3.3 Ataques de disponibilidade em redes LoRa

Esta seção do desenvolvimento descreve as estratégias utilizadas para degradar a rede, executados sobre o *baseline* definido na seção 3.2 para simular o monitoramento do ambiente de desastre. Os resultados destes experimentos serão apresentados no Capítulo 4 deste trabalho.

Para sobrecarregar a rede e avaliar o impacto de ataques DoS em dispositivos LoRa foram propostos cenários modificando parâmetros de entrada da rede, ou seja, aumentando o número de nós, reduzindo o tempo médio de envio e reduzindo o número de estações base. Além disso, alguns experimentos consistem em diminuir ou aumentar mais de um parâmetro por vez e outros em modificar parâmetros no código do simulador.

A maioria dos experimentos foram realizados para ambos os tipos de verificação de colisão, ou seja, a verificação de colisão padrão e a verificação completa de colisão. Como visto anteriormente, a verificação de colisão completa considera o efeito de captura, pelo qual uma das duas mensagens em colisão ainda pode passar dependendo do tempo relativo e da diferença no poder de recebimento, enquanto a verificação padrão não considera esse efeito.

A Tabela 16 descreve o objetivo dos cenários propostos para afetar a disponibilidade da rede. Além disso, descreve as variáveis associadas, ou seja, que serão modificadas em cada cenário e a quantidade de experimentos realizados para cada cenário. Cada um dos cenários e seus experimentos são descritos de forma detalhada no restante desta seção.

Cenário	Objetivo	Variáveis associadas	Qtd de experimentos
1	Verificar o impacto do aumento do número de nós na disponibilidade da rede	n° de nós	4
2	Verificar o impacto da redução do intervalo médio de envio de pacotes na disponibilidade da rede	Intervalo médio de envio de pacotes	4
3	Verificar o impacto da redução do número de estações base na disponibilidade da rede	n° de estações base	1
4	Verificar o impacto do aumento do número de nós e na redução do intervalo médio de envio de pacotes na disponibilidade da rede	n° de nós e intervalo médio de envio de pacotes	1
5	Verificar o impacto da redução do número de estações base e do intervalo médio de envio de pacotes na disponibilidade da rede	n° de estações base e intervalo médio de envio de pacotes	1
6	Verificar o impacto da redução do número de estações base e aumento do número de nós na disponibilidade da rede	n° de estações base e n° de nós	1
7	Verificar o impacto do aumento da distância máxima entre os nós e as estações base na disponibilidade da rede	Distância máxima entre os nós e as estações base	1
8	Verificar o impacto do aumento do pacote na disponibilidade da rede	Tamanho do pacote	1
9	Verificar o impacto do aumento do pacote e redução do intervalo médio de envio de pacotes na disponibilidade da rede	Tamanho do pacote e intervalo médio de envio de pacotes	1

Tabela 16 – Descrição dos ataques de disponibilidade na rede LoRa.

3.3.1 Cenário 1 - Aumento do número de nós

O primeiro cenário consiste no aumento do número de nós da rede LoRa, mantendo os demais parâmetros intactos. O intuito da execução destas simulações é verificar se o aumento de dispositivos conectados à rede simulada de monitoramento da bacia hidrográfica do Córrego do Leitão causa uma sobrecarga sobre a mesma, pois quanto maior o número de nós mais pacotes são enviados em um certo intervalo de tempo.

O *script* `oneDirectionalLoraIntf.py` foi utilizado para as simulações que envolviam a métrica DER para mensurar os resultados, pois é o *script* que considera mais parâmetros de entrada e engloba todo o *baseline* estabelecido na seção anterior. O *script* `loraDir.py` foi utilizado em simulações que envolviam a métrica NEC, pois é o único que retorna o NEC como saída, entretanto possui as limitações de utilizar apenas uma estação base e não é possível configurar a direcionalidade, o número de redes LoRa e a distância entre estações base.

No primeiro cenário quatro experimentos foram executados. O primeiro, aumen-

tando o número de nós minimamente, o segundo aumentando ainda mais a quantidade de nós e o terceiro aumentando os nós com o objetivo de diminuir o desempenho da rede para a verificação completa de colisão. Os três primeiros experimentos envolvem a análise da métrica DER, enquanto o quarto experimento consiste em aumentar minimamente o número de nós e verificar o desempenho da métrica NEC. Os experimentos do cenário 1 estão descritos na Tabela 17.

Nºexperimento	Nº mínimo de nós	Nº máximo de nós	Variação	Métrica analisada
1	24	240	24	DER
2	24	12000	1200	DER
3	12000	48000	2400	DER
4	1	10	1	NEC

Tabela 17 – Conjunto de experimentos do Cenário 1.

No primeiro experimento do Cenário 1, expandiu-se minimamente o número de nós, ou seja, de 1 em 1 nó associado a cada estação base, como o *baseline* utiliza 24 estações base, o número de nós aumentou de 24 em 24 até atingir 10 nós por estação base (240 nós totais). O mesmo experimento foi executado utilizando a verificação de colisão padrão e a verificação completa de colisão. Devido à aleatoriedade das colisões e do posicionamento dos nós no LoRaSim, foram realizadas 100 simulações com cada configuração de número de nós e a média do DER foi calculada.

No segundo experimento do Cenário 1, foram realizadas simulações aumentando de 50 em 50 a quantidade de nós por estação base, logo a quantidade de nós aumentou de 1200 em 1200, a partir de 24 nós, até atingir 12000 nós. O intuito deste experimento foi comparar o desempenho da rede para a verificação de colisão padrão e para a verificação completa de colisão, quando ocorre o aumento do número de dispositivos conectados à rede. Nesse caso, por causa do aumento do tempo de execução provocado pelo grande aumento dos nós, apenas 10 simulações com cada uma das configurações foi executada. A média do DER em função do aumento dos nós foi calculada.

A fim de degradar o desempenho da rede, mesmo com a verificação de colisão completa ativada, o terceiro experimento aumentando o número de nós foi executado. Desta vez, incrementando a quantidade de nós por estação base em 100, ou seja, de 2400 em 2400 nós, a partir de 12000 nós, até atingir 48000 nós.

Também foram realizadas simulações para verificar o comportamento do NEC com o aumento de nós. O único *script* que calcula o NEC é o *loradir.py*, neste, só é possível utilizar uma estação base. O quarto experimento do cenário 1, consiste em aumentar de 1 em 1 o número de nós até 10. Assumindo um crescimento linear, multiplicou-se o NEC médio resultante das 100 simulações executadas para cada quantidade de nós, pelo número de estações base (24).

3.3.2 Cenário 2 - Redução do intervalo médio de envio de pacotes

O objetivo do segundo cenário de experimentos é decrementar o valor do intervalo médio de envio de pacotes e manter os demais parâmetros de entrada, para avaliar o impacto que a redução deste parâmetro tem na rede LoRa. Espera-se uma sobrecarga na rede de monitoramento, pois quanto menor o intervalo médio de envio de pacotes, mais pacotes são enviados por nó. Novamente, o *script* `oneDirectionalLoraIntf.py` foi utilizado para as simulações que envolviam a métrica DER para mensurar os resultados, já o *script* `loraDir.py` foi utilizado em simulações que envolviam a métrica NEC. A Tabela 18 descreve os experimentos do cenário 2.

Nº experimento	Intervalo médio máximo de envio de pacotes	Intervalo médio mínimo de envio de pacotes	Variação	Métrica analisada
1	50000ms	1000ms	5000ms	DER
2	1000ms	100ms	100	DER
3	100ms	10ms	10ms	DER
4	50000ms	1000ms	5000ms	NEC

Tabela 18 – Conjunto de experimentos do Cenário 2.

No Experimento 1 do Cenário 2, decrementou-se o valor do intervalo médio de envio de pacotes em 5000ms, iniciando o decremento pelo valor 50000ms proposto pelo *baseline* determinado anteriormente, até atingir o valor 5000ms. O experimento também foi executado para o valor 1000ms. Cada simulação foi executada 100 vezes devido à aleatoriedade do posicionamento dos nós e das colisões. A média do DER em função do decremento do parâmetro foi calculada para cada configuração de intervalo médio de envio de pacotes.

O segundo experimento desta subseção consiste em reduzir ainda mais o valor do intervalo médio de envio de pacotes, desta vez para comparar os resultados gerados para a verificação de colisão padrão e para a verificação de colisão completa. O valor do parâmetro inicia-se em 1000ms e é reduzido de 100ms em 100ms, até atingir o valor 100ms. A mesma simulação foi executada 10 vezes para cada configuração de intervalo médio de envio. A média do DER em função do decremento do parâmetro foi calculada.

Com o intuito de degradar o desempenho da rede mesmo com a verificação de colisão completa ativada, o terceiro experimento reduzindo o intervalo médio de envio de pacotes foi executado. Desta vez, decrementando o tempo em 10ms, a partir de 100ms, até atingir 10ms.

Para avaliar o impacto na rede utilizando a métrica NEC, simulações foram realizadas diminuindo o intervalo médio de envio de pacotes considerando dois nós para uma estação base, devido a limitação do *script* de utilizar apenas uma estação base. Após a

execução do experimento, os valores do NEC foram multiplicados por 24, ou seja, pelo número de estações base do *baseline*.

3.3.3 Cenário 3 - Diminuição do número de estações base

A finalidade do terceiro cenário é diminuir a quantidade de estações base sobre o *baseline* estabelecido. Nesse caso, apenas o DER é utilizado como métrica de análise, devido às limitações destacadas anteriormente sobre o simulador *loradir.py*. O experimento foi executado para a verificação padrão e para a verificação completa de colisão.

O número de estações base foi reduzido de acordo com os valores possíveis de estações base propostos pelo simulador, ou seja, 24, 8, 6, 4, 3, 2, 1. Foram realizadas 100 simulações com cada quantidade de estações base e a média do DER em função da redução do número de estações base foi calculada.

3.3.4 Cenário 4 - Aumento do número de nós e redução do intervalo médio de envio de pacotes

No Cenário 4, apenas um experimento foi executado dobrando o número de nós a partir de 48 até 1536 e reduzindo o intervalo médio de envio de pacotes a partir de 50000ms, até 1000ms. O intuito deste experimento é verificar o desempenho da rede para a métrica DER, ao receber um número muito alto de pacotes, pois ambas as alterações nos parâmetros provocam sobrecarga na quantidade de pacotes.

Este experimento foi executado para a verificação padrão de colisão e para a verificação completa de colisão. Cada configuração de intervalo médio de envio e quantidade de nós foi simulada 10 vezes e a média do DER em função do aumento do número de dispositivos na rede e da redução do intervalo médio de envio de pacotes foi calculada.

3.3.5 Cenário 5 - Diminuição do número de estações base e intervalo médio de envio de pacotes

O Cenário 5 tem como objetivo avaliar o comportamento da rede quando poucas estações base estão recebendo muitos pacotes em pouco tempo. O experimento para este cenário foi realizado diminuindo o intervalo médio de envio de pacotes de 5000ms em 5000ms para cada quantidade possível de estações base (24, 8, 6, 4, 3, 2, 1), tanto para a verificação padrão de colisão, quanto para a verificação completa de colisão. O mesmo experimento foi executado 100 vezes para cada configuração de intervalo médio de envio de pacotes e quantidade de estações base e a média do DER foi calculada.

3.3.6 Cenário 6 - Diminuição do número de estações base e aumento do número de nós

O intuito do Cenário 6 é simular um ataque no qual muitos dispositivos enviam pacotes para poucas estações base realizarem o processamento e envio para os servidores. Dessa forma, 100 simulações para cada configuração foram executadas, dobrando o número de nós a partir de 48 até 3072 utilizando cada um dos números possíveis de estações base (24, 8, 6, 4, 3, 2, 1) tanto para a verificação padrão de colisão, quanto para a verificação completa de colisão.

3.3.7 Cenário 7 - Aumento da distância máxima entre os nós e as estações base

No experimento deste cenário, foi necessário modificar a distância máxima possível entre os nós e as estações base no código do *script* `oneDirectionalLoraIntf.py` para verificar o efeito provocado na métrica DER, a partir da perda da intensidade do sinal entre os nós e as estações base. Foram realizados 10 experimentos com cada configuração de distância máxima, aumentando de 100 em 100 unidades a distância e a média do DER foi calculada para cada configuração. O valor inicial para este parâmetro era de 98,95 unidades.

3.3.8 Cenário 8 - Aumento do tamanho do pacote

O pacote LoRa é dividido em quatro partes, um preâmbulo (*preamble*), programável de 6 a 65535 símbolos, aos quais o rádio adiciona mais 4,25 símbolos, um cabeçalho opcional (*Header*) que descreve o comprimento do pacote, a taxa FEC da carga útil e indica a existência de um CRC. Em seguida, está localizada a própria carga útil do pacote que pode variar de 1 até 255 bytes (*payload*) e por fim o CRC opcional (BOR et al., 2016). A estrutura do pacote está representada na Figura 23, extraída de (BOR et al., 2016).

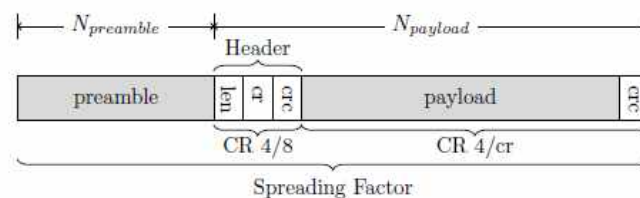


Figura 23 – Estrutura do pacote. Fonte: Extraída de (BOR et al., 2016)

Para simular um ataque enviando pacotes de tamanho alterado foi necessário modificar os *scripts* `oneDirectionalLoraIntf.py` e `loraDir.py`. Inicialmente, o valor do preâmbulo era de 8 símbolos e o cabeçalho adicional estava desabilitado. Após a modificação o valor

do preâmbulo passou a ser o número máximo de símbolos (65535) e o cabeçalho adicional foi habilitado. As médias resultantes do DER e do NEC do experimento foram obtidas para a execução de 100 simulações com as configurações. O próximo cenário trabalha com o aumento da carga útil do pacote.

3.3.9 Cenário 9 - Aumento do tamanho do pacote e diminuição do intervalo médio de envio de pacotes

A ideia por trás do cenário 9 é simular um ataque enviando muitos pacotes grandes em pouco tempo. Dessa forma, para cada configuração de tempo médio de envio, de 50000ms até 1000ms (diminuindo de 10000ms em 10000ms), a carga útil do pacote foi aumentada de 51 bytes em 51 bytes até 255 bytes. Foram realizados 100 experimentos para cada configuração e a média dos resultados obtidos para a métrica DER foi calculada.

4 Resultados

Este capítulo aborda os resultados dos experimentos para degradar o desempenho da rede LoRa, que foram descritos na seção 3.3 e executados sobre o *baseline* para monitoramento do ambiente de desastre definido na seção 3.2. A numeração utilizada para separar e descrever os cenários na seção anterior será a mesma utilizada para descrever os resultados obtidos neste capítulo.

4.1 Cenário 1 - Aumento do número de nós

Os resultados para o primeiro experimento do Cenário 1 estão representados na Figura 24. A curva azul representa a variação do DER em função do aumento do número de nós para a verificação completa de colisão e a curva laranja representa a variação do DER em função do aumento do número de nós para a verificação padrão de colisão.

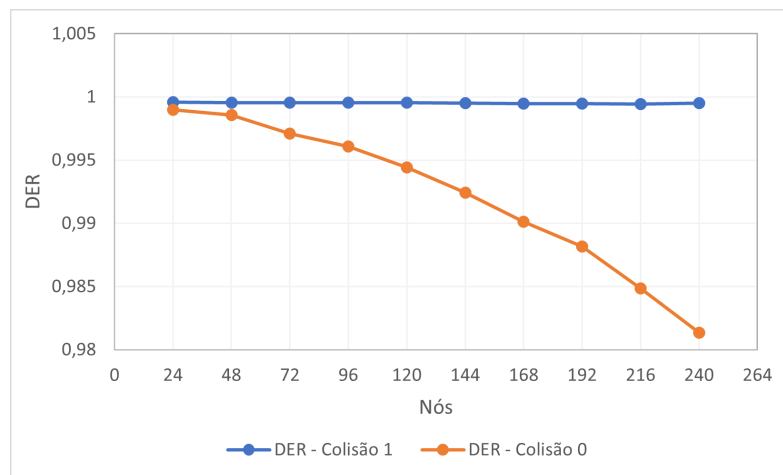


Figura 24 – Cenário 1, experimento 1 - Comportamento do DER em função do aumento inicial do número de nós para a verificação padrão e verificação completa de colisão.

A partir do gráfico gerado em função do aumento inicial do número de nós, mantendo os demais parâmetros da rede intactos, observa-se que este aumento inicial não impactou o desempenho da rede para a métrica DER. A taxa de pacotes recebidos por pacotes enviados (DER) resultante continua superior a 0,95 para os dois tipos de verificação de colisão, mesmo operando com 10 nós por estação base, ou seja, 240 nós na rede LoRa no total. Dessa forma, ainda atende aos requisitos propostos para o monitoramento.

Embora o aumento do número de nós provoque o aumento do número de pacotes enviados e conseqüentemente o crescimento do número de colisões, a rede configurada

possui um intervalo médio de envio de pacotes de aproximadamente 1 minuto e 24 estações base para receber os pacotes e transmiti-los para os servidores da rede, ou seja, uma configuração robusta para sustentar essa quantidade de nós.

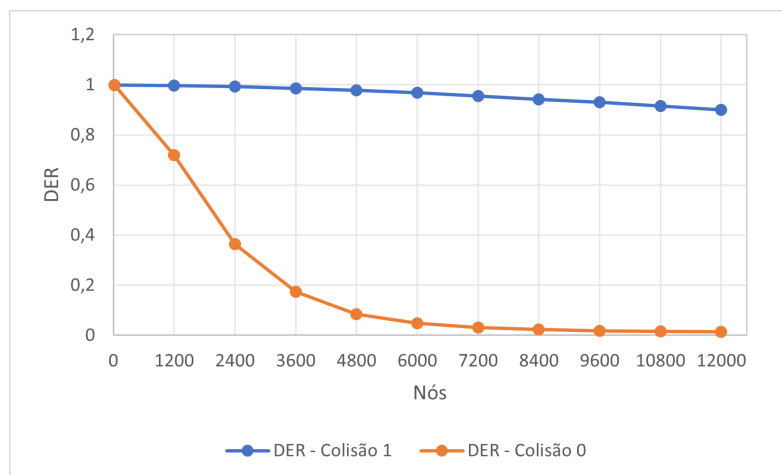


Figura 25 – Cenário 1, experimento 2 - Comportamento do DER em função do segundo aumento do número de nós para a verificação padrão e verificação completa de colisão.

O gráfico da Figura 25 representa os resultados do segundo experimento do cenário 1, ou seja, a diferença do efeito causado na métrica DER da rede LoRa para a verificação padrão de colisão e para a verificação completa de colisão em função do grande aumento do número de nós. Verifica-se que a partir de 1200 nós o DER para a verificação padrão de colisão (0,719737) é inferior a 0,95 e decresce até 0,01412 quando a rede possui 12000 nós. Já para a verificação completa de colisão, o DER continua superior à 0,95 e passa a ser inferior a esse valor apenas quando a rede possui 8400 nós, ou seja, uma quantidade bem alta de dispositivos conectados.

Como descrito anteriormente na subseção 3.3.1, o objetivo principal do terceiro experimento do primeiro cenário é tentar sobrecarregar a rede aumentando o número de nós excessivamente para a verificação completa de colisão. Nesse caso, não foi necessário executar o experimento para a verificação de colisão padrão, pois no experimento anterior o DER atingiu um valor próximo de zero utilizando este parâmetro. A Figura 26 ilustra os resultados do Experimento 3 do primeiro cenário.

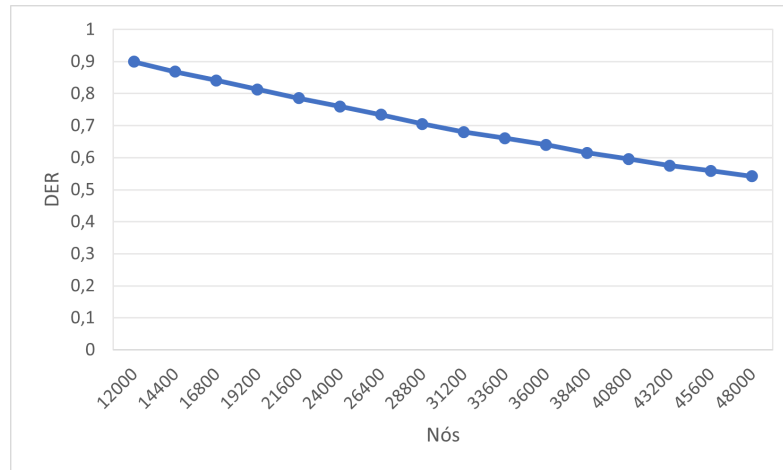


Figura 26 – Cenário 1, experimento 3 - Tentativa de sobrecarregar a rede aumentando o número de nós para a verificação completa de colisão.

A partir dos resultados obtidos, é possível observar a robustez da implementação da rede LoRa quanto ao aumento agressivo do número de nós, quando a verificação de colisão completa é utilizada. Mesmo com uma grande quantidade de nós (48000), a rede LoRa ainda opera com uma capacidade maior que 50%.

Os primeiros experimentos do Cenário 1 tinham como métrica de análise a taxa de extração de dados (DER). A Figura 27 e a Tabela 19 ilustram a variação do consumo de energia (NEC) em Joule da rede, em função do aumento do número de nós.

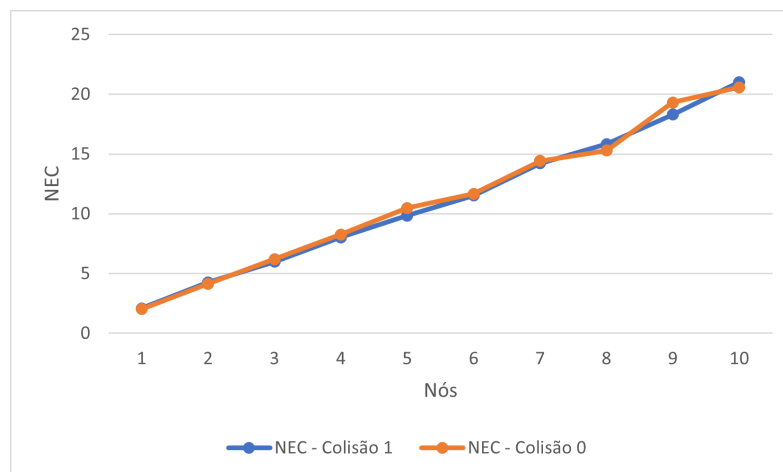


Figura 27 – Cenário 1, experimento 4 - Comportamento do NEC em função do aumento inicial do número de nós.

Nós	NEC
24	50,13975859
48	102,3686738
72	143,7339746
96	192,6202392
120	236,492528
144	276,6043349
168	341,3681897
192	379,8086714
216	439,5585504
240	504,3224052

Tabela 19 – Comportamento do NECx24 em função do aumento de nós da rede.

Como visto anteriormente o consumo de energia em Joule é influenciado pelo número de nós da rede, frequência das transmissões e parâmetros de comunicação do transmissor. A partir da Figura 27 e da Tabela 19 é possível confirmar que o número de nós conectados à rede e o NEC são grandezas diretamente proporcionais. Além disso, a métrica não depende do parâmetro Colisão, pois os valores para a verificação padrão e para a verificação completa de colisão são muito parecidos.

4.2 Cenário 2 - Redução do intervalo médio de envio de pacotes

A Figura 28 ilustra os resultados do primeiro experimento do Cenário 2, que foi descrito na subseção 3.3.2. A curva azul representa a variação do DER em função da redução do intervalo médio de envio de pacotes para a verificação completa de colisão e a curva laranja para a verificação padrão de colisão.

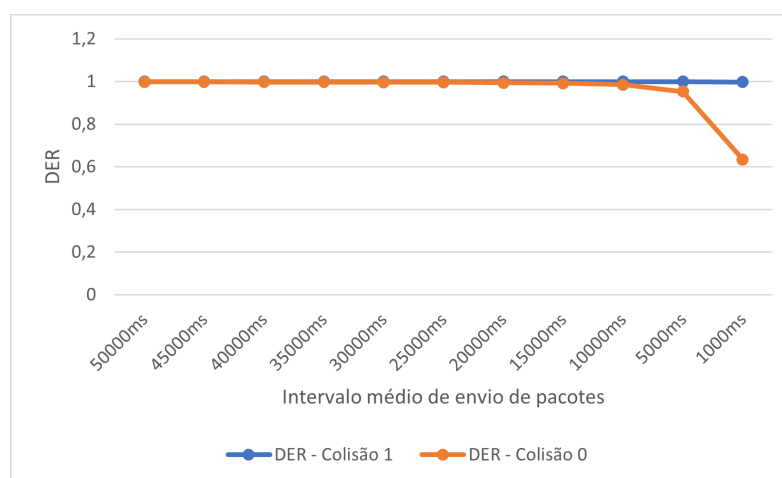


Figura 28 – Cenário 2, experimento 1 - Comportamento do DER em função da diminuição inicial do intervalo médio de envio de pacotes.

Os resultados do gráfico da Figura 28 mostram que mesmo abaixando o tempo médio de envio até 5000ms, a rede LoRa ainda fornece um DER maior que 0,95 tanto para a verificação padrão de colisão, quanto para a verificação completa de colisão. Isso pode ser explicado pelo uso de apenas 2 nós pela alta quantidade de estações base (24). Dessa forma, mesmo aumentando a quantidade de pacotes enviados durante um período de tempo, a quantidade de nós enviando pacotes é pequena e as estações base ainda são capazes de processá-los. Observa-se uma queda do valor do DER para a verificação padrão de colisão quando o tempo médio de envio de pacotes é de 1 segundo.

A Figura 29 ilustra os resultados do segundo experimento do Cenário 2. Este, tem o intuito de comparar o desempenho da rede LoRa sobre o efeito da alta redução do tempo médio de envio de pacotes para a verificação completa de colisão e para a verificação padrão de colisão.

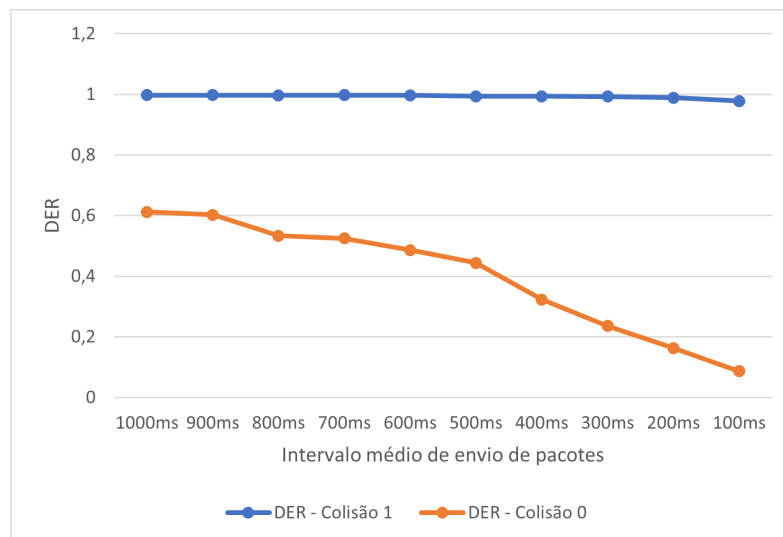


Figura 29 – Cenário 2, experimento 2 - Comportamento do DER em função da diminuição agressiva do intervalo médio de envio de pacotes.

No caso deste experimento o intervalo médio de envio chegou a valores menores que 1 segundo. Percebe-se que o DER para a verificação completa de colisão continuou maior que 0,95 mesmo para 1/10 de segundo. Entretanto, o DER para a verificação padrão de colisão atingiu valores próximos de zero para 1/10 de segundo.

Essa diferença entre o DER para as verificações de colisão pode ser explicada pelo seguinte fator, ao diminuir muito o intervalo médio de envio de pacotes, mais pacotes podem ser enviados e chegar ao mesmo tempo, com a mesma frequência e fator de propagação. No caso da verificação de colisão completa alguns desses pacotes ainda passarão e serão entregues à alguma estação base, porém no caso da verificação padrão esse pacote não será entregue, diminuindo a taxa de pacotes entregues por pacotes enviados.

Os resultados do terceiro experimento do Cenário 2, para tentar diminuir o desempenho da rede LoRa, mesmo com a verificação completa de colisão ativada estão representados graficamente na Figura 30. Apesar de reduzir o valor do intervalo médio de envio de pacotes para 1/100 de segundo, o DER para a verificação completa de colisão não sofre nenhuma redução. Isso pode ser explicado pelo comportamento das colisões descrito no experimento anterior.

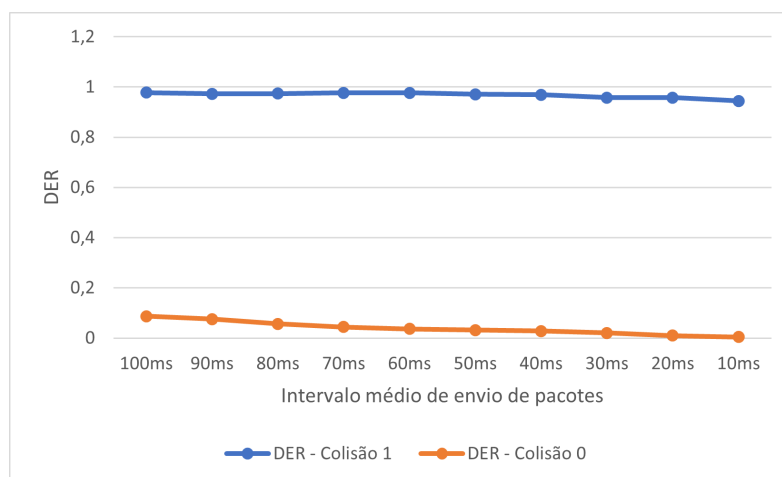


Figura 30 – Cenário 2, experimento 3 - Tentativa de degradar a disponibilidade da rede para a verificação completa de colisão.

Os primeiros experimentos do Cenário 2 tinham como métrica de análise a taxa de extração de dados (DER). A Figura 31 e a Tabela 20 ilustram a variação do consumo de energia (NEC) em Joule da rede, em função do decremento inicial do intervalo médio de envio de pacotes.

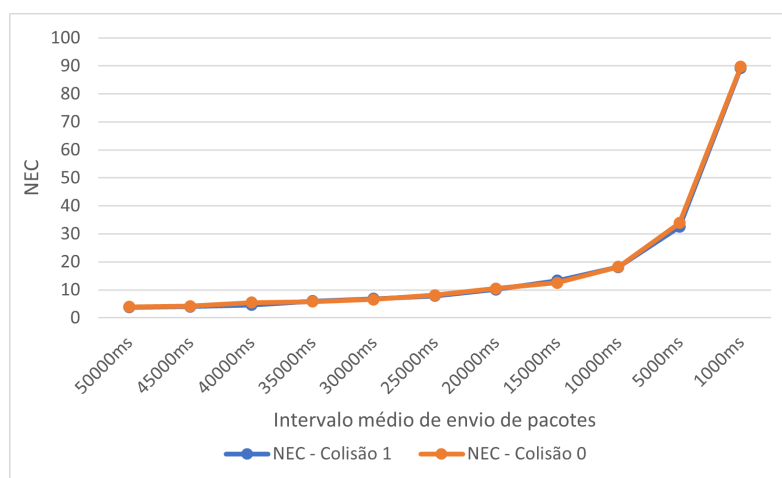


Figura 31 – Cenário 2, experimento 4 - Comportamento do NEC em função da redução do intervalo médio de envio de pacotes.

Intervalo médio de envio de pacotes	NEC
50000ms	91,08723
45000ms	96,93687
40000ms	111,1431
35000ms	143,3161
30000ms	163,372
25000ms	188,0241
20000ms	243,5957
15000ms	318,3875
10000ms	435,7981
5000ms	780,0911
1000ms	2143,06

Tabela 20 – Comportamento do NEC * 24 em função da redução do intervalo médio de envio de pacotes.

O consumo de energia em Joule da rede LoRa é influenciado pela frequência das transmissões. A frequência das transmissões aumenta à medida que o intervalo médio de envio de pacotes diminui. Dessa forma, quando o intervalo médio de envio de pacotes diminui, o consumo de energia aumenta, logo são grandezas inversamente proporcionais. É possível confirmar esses fatores por meio da Figura 31 e pela Tabela 20. Além disso, a métrica não depende do parâmetro Colisão, pois os valores para a verificação padrão e para a verificação completa de colisão são visivelmente parecidos.

4.3 Cenário 3 - Diminuição do número de estações base

O comportamento da rede em função da diminuição do número de estações base está representado na Figura 32. A curva azul representa a variação do DER em função da redução do número de estações base utilizando a verificação completa de colisão e a curva laranja quando utiliza-se a verificação padrão de colisão

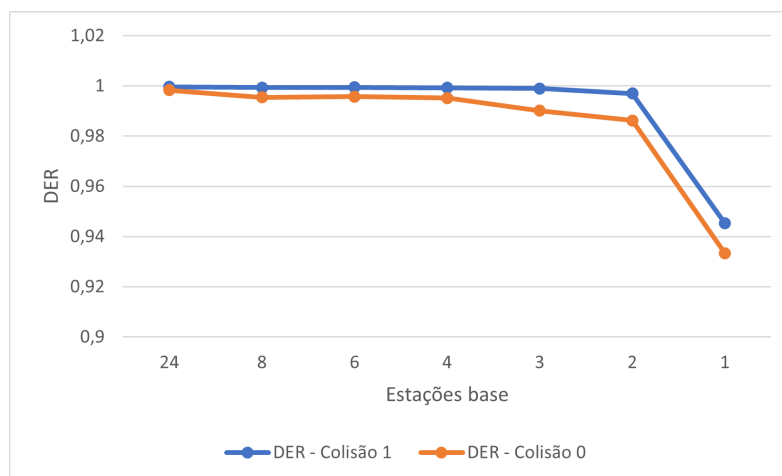


Figura 32 – Cenário 3, experimento 1 - Comportamento do DER em função da redução do número de estações base.

Apesar da taxa de extração dos dados (DER) depender do número de estações base, o valor da métrica continua alto para os dois tipos de verificação de colisão. Este, fica abaixo de 0,95 apenas para a configuração com uma estação base (0,9452795 para a verificação completa de colisão e 0,933301 para a verificação padrão de colisão). A redução pequena do valor do DER pode ser explicada pela robustez da implementação da rede. Apesar da redução do número de estações base, duas estações base ainda são suficientes para manter o desempenho ótimo da rede para o número de nós proposto para o monitoramento, desde que o intervalo médio de envio de dados seja de 50000ms.

4.4 Cenário 4 - Aumento do número de nós e redução do intervalo médio de envio de pacotes

Os resultados do experimento executado para o Cenário 4, utilizando a verificação completa de colisão estão ilustrados na Figura 33. Os resultados para a verificação padrão de colisão estão ilustrados na Figura 34. Cada curva colorida representa a variação do DER em função da redução do intervalo médio de envio de pacotes para determinada quantidade de nós.

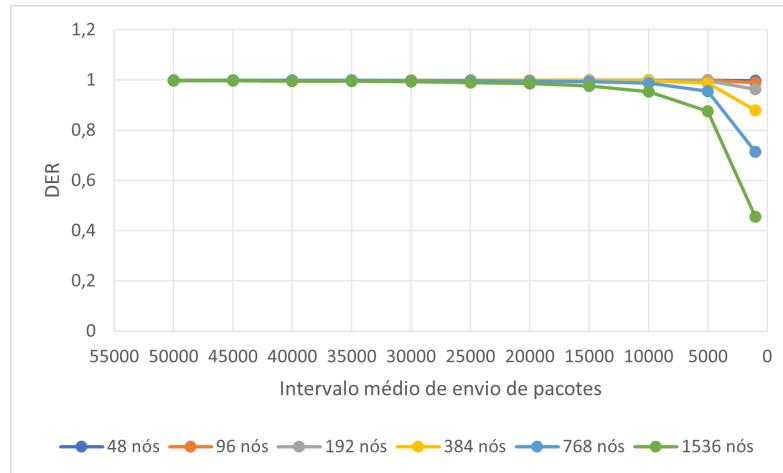


Figura 33 – Cenário 4, experimento 1 - Comportamento do DER em função do aumento do número de nós e redução do intervalo médio de envio de pacotes para a verificação completa de colisão.

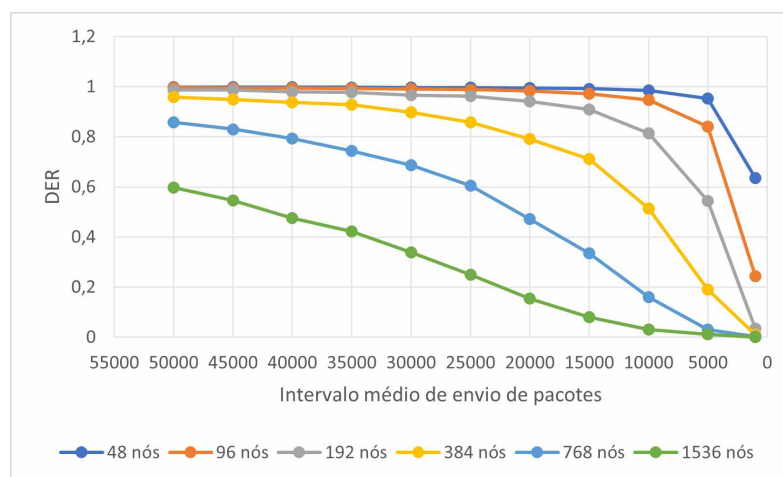


Figura 34 – Cenário 4, experimento 1 - Comportamento do DER em função do aumento do número de nós e redução do intervalo médio de envio de pacotes para a verificação padrão de colisão.

A partir das Figuras 33 e 34 é possível perceber que pela primeira vez, durante a execução dos experimentos o DER reduziu para ambos os tipos de verificação de colisão. Isso ocorre, pois com a modificação dos dois parâmetros propostos pelo experimento, o número de pacotes aumenta ao ponto das 24 estações base propostas pelo *baseline* não suportarem o recebimento.

4.5 Cenário 5 - Diminuição do número de estações base e intervalo médio de envio de pacotes

Os resultados do experimento executado para o Cenário 5, utilizando a verificação completa de colisão estão ilustrados na Figura 35. Os resultados para a verificação padrão de colisão estão ilustrados na Figura 36. Cada curva colorida representa a variação do DER em função da redução do intervalo médio de envio de pacotes para determinada quantidade de estações base.

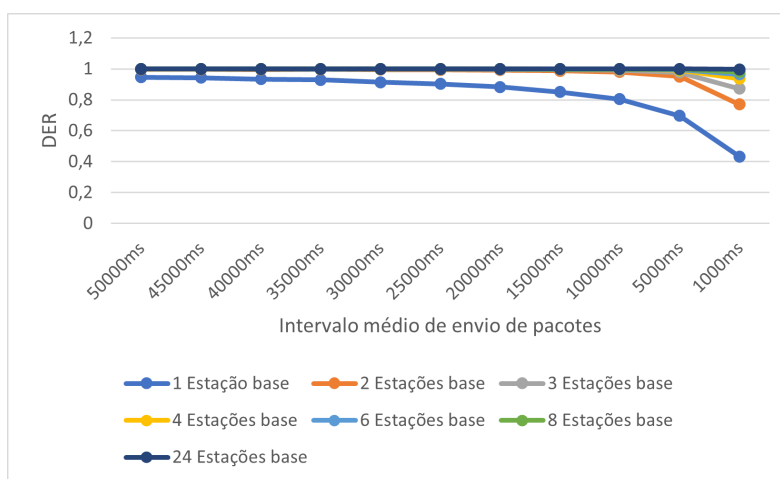


Figura 35 – Cenário 5, experimento 1 - Comportamento do DER em função da redução do número de estações base e do intervalo médio de envio de pacotes para a verificação completa de colisão.

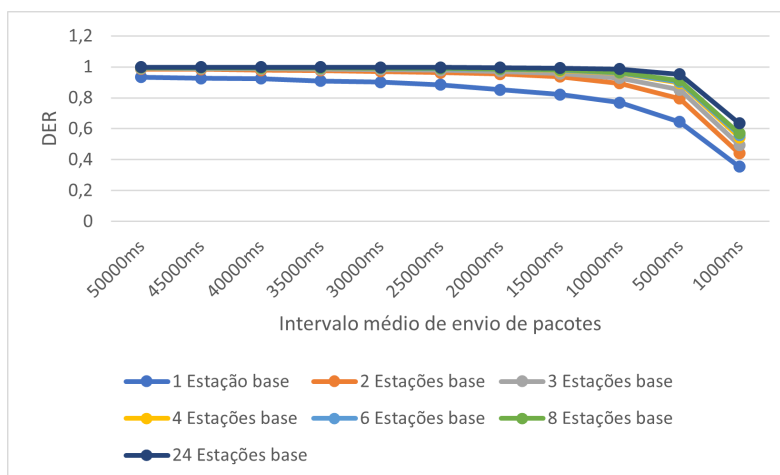


Figura 36 – Cenário 5, experimento 1 - Comportamento do DER em função da redução do número de estações base e do intervalo médio de envio de pacotes para a verificação padrão de colisão.

A partir das Figuras 35 e 36 é possível perceber que o DER reduziu para ambos os tipos de verificação de colisão. Isso ocorre, porque quando o intervalo médio de envio

de pacotes é pequeno, mais pacotes são enviados pelos dispositivos durante uma fração de tempo e existem menos estações base para receber e transmitir a alta quantidade de pacotes para os servidores. Observa-se que a para uma estação base e 1 segundo (1000ms) de tempo médio de envio de pacotes o DER para ambos os valores do parâmetro Colisão é muito inferior a 0,95.

4.6 Cenário 6 - Diminuição do número de estações base e aumento do número de nós

O DER em função do aumento do número de nós e decremento do número de estações base para a verificação completa de colisão está representado na Figura 37 e para a verificação padrão de colisão na Figura 38. As curvas coloridas representam o valor do DER em relação ao aumento do número de nós para cada quantidade de estações base.

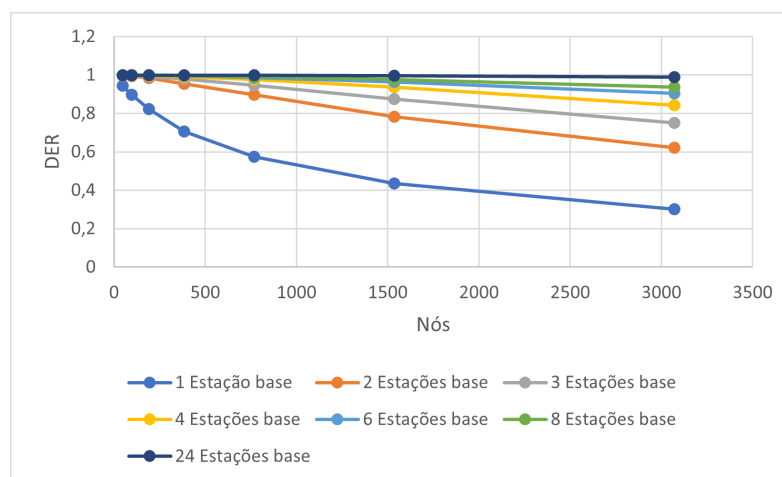


Figura 37 – Experimento 6 - Comportamento do DER em função da redução do número de estações base e do aumento de nós para a verificação completa de colisão.

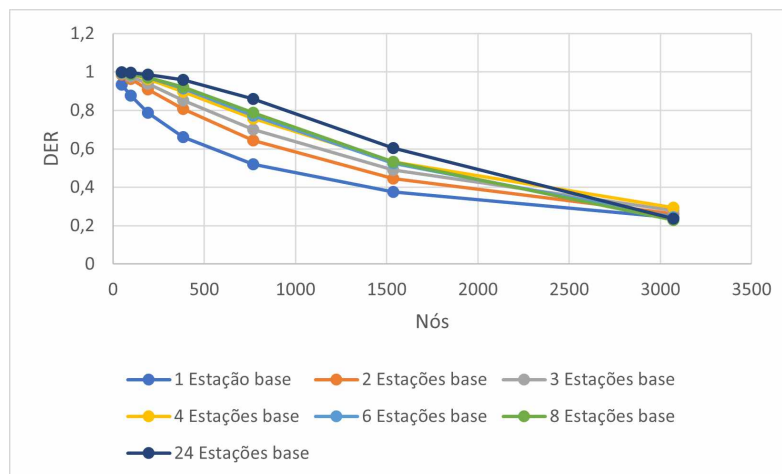


Figura 38 – Experimento 6 - Comportamento do DER em função da redução do número de estações base e do aumento de nós para a verificação padrão de colisão.

A partir dos resultados ilustrados nas Figuras 37 e 38, é possível perceber como o aumento de nós causa um impacto cada vez maior no DER quando menos estações base estão conectadas à rede, tanto para a verificação completa, quanto para a verificação padrão de colisões. Quando a rede possui 768 nós e utiliza-se a verificação padrão de colisão, o DER passa a ser menor que 0,95 para qualquer quantidade de estações base. Já para a verificação completa de colisão, a partir de 3072 nós, é possível perceber que para todas as quantidades de estações base, exceto 24 o DER é inferior à 0,95.

Dessa forma, é possível concluir que ao aumentar a quantidade de nós e diminuir o número de estações base o desempenho da rede cai consideravelmente. Isso pode ser explicado pelo fato de que mais nós enviam mais pacotes em um mesmo intervalo de tempo e poucas estações base não são suficientes para suportar o recebimento de pacotes de muitos nós. O número de colisões também torna-se alto, isso explica o fato da alta redução do desempenho da rede quando ocorre apenas a verificação padrão de colisão.

4.7 Cenário 7 - Aumento da distância máxima entre os nós e as estações base

A Figura 39 representa graficamente o desempenho do DER ao aumentar a distância máxima entre os nós e as estações base. As Figuras 40, 41, 42 representam visualmente a distribuição dos nós, estações base e alcance do sinal das estações base da rede LoRa para diferentes distâncias máximas entre nós e estações base. É importante ressaltar, que o experimento para este cenário foi executado para a métrica NEC, entretanto, não houveram alterações no consumo de energia da rede, ao variar a distância máxima entre os nós e as estações base.

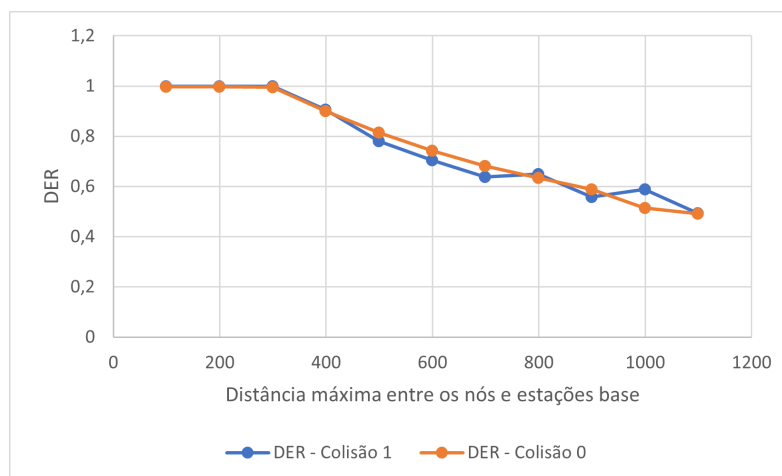


Figura 39 – Cenário 7, experimento 1 - Comportamento do DER em função do aumento da distância máxima entre os nós e as estações base.

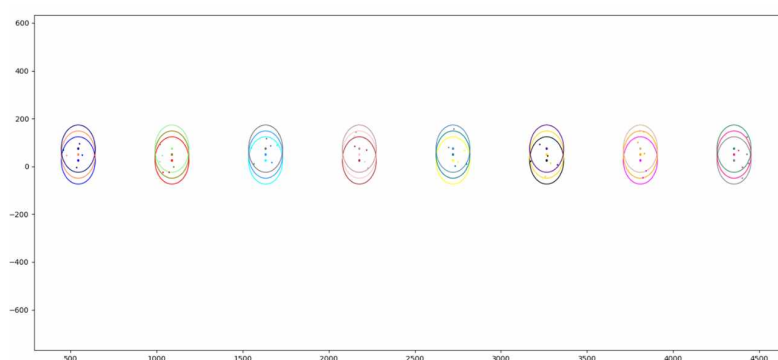


Figura 40 – Cenário 7, experimento 1 - Distância máxima entre os nós e as estações base igual a 98,95

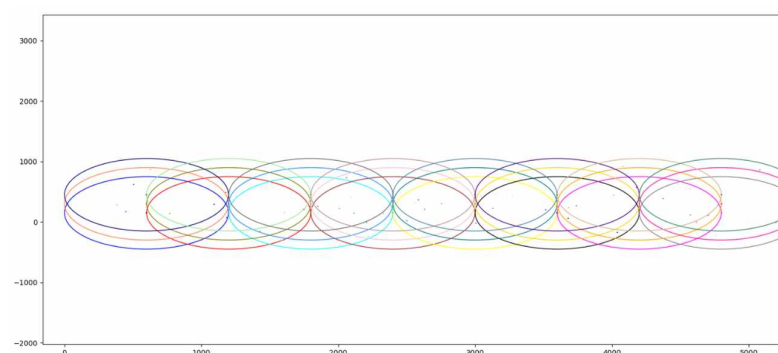


Figura 41 – Cenário 7, experimento 1 - Distância máxima entre os nós e as estações base igual a 598,95

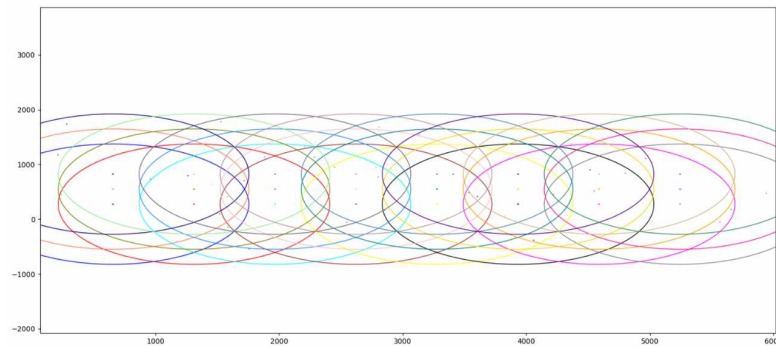


Figura 42 – Cenário 7, experimento 1 - Distância máxima entre os nós e as estações base igual a 1098,95

De acordo com o gráfico da Figura 39, é possível identificar uma queda alta da métrica DER, a partir da distância máxima entre os nós e as estações base de 400 unidades. Esta redução na proporção de pacotes recebidos por pacotes enviados pode ser causada pelo fato de alguns nós ficarem distantes das estações base, ou seja, em locais onde o alcance da intensidade do sinal das estações base são mais baixos.

4.8 Cenário 8 - Aumento do tamanho do pacote

Os resultados referentes às modificações executadas sobre os pacotes da rede LoRa estão representados na Tabela 21. A configuração dos pacotes proposta pelo *script* `one-DirectionalLoraIntf.py.`, consiste em 8 símbolos no preâmbulo, o cabeçalho opcional desabilitado e uma carga útil de 20 bytes. Essa configuração está representada na coluna *baseline* da tabela abaixo. A variação da carga útil do pacote (*payload*) será apresentada no Cenário 9.

	<i>Baseline</i>	65535 símbolos	Cabeçalho habilitado
DER - Colisão 1	0,999835796	0,8392563	0,998963832
DER - Colisão 0	0,998072591	0,05798706	0,999654266
NEC - Colisão 1	4,039036109	567,2060191	3,864939725
NEC - Colisão 0	4,039036109	567,2060191	4,039036109
NEC(x24) - Colisão 1	96,936866616	13.612,944457	92,7585534
NEC(x24) - Colisão 0	96,936866616	13.612,944457	96,936866

Tabela 21 – Cenário 8, experimento 1 - Comportamento das métricas DER e NEC em função da modificação da estrutura do pacote.

Observa-se pela Tabela 21 que o valor das métricas DER e NEC sofrem um alto impacto quando o número máximo de símbolos do preâmbulo é utilizado. Já a habilitação do cabeçalho opcional não provoca mudanças no desempenho da rede.

4.9 Cenário 9 - Aumento do tamanho do pacote e diminuição do intervalo médio de envio de pacotes

O resultado do experimento para o Cenário 9, utilizando a verificação completa de colisão está representado graficamente na Figura 43 e para a verificação padrão de colisão na Figura 44. As curvas coloridas representam o valor do DER em relação a redução do intervalo médio de envio de pacotes para cada valor de tamanho de pacote.

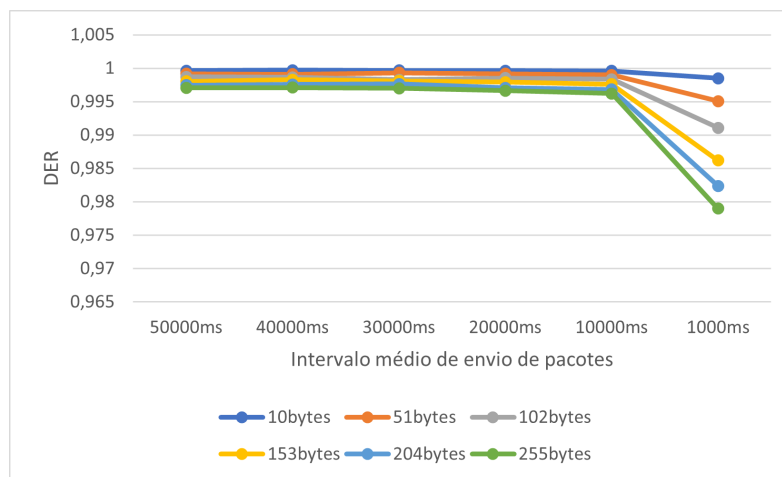


Figura 43 – Cenário 9, experimento 1 - Comportamento do DER em função do aumento do tamanho do pacote e da redução do intervalo médio de envio de pacotes para a verificação completa de colisão.

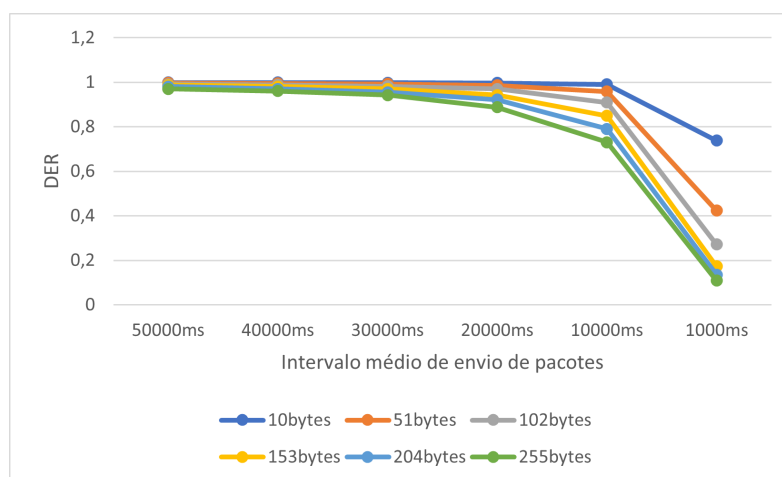


Figura 44 – Experimento 9 - Comportamento do DER em função do aumento do tamanho do pacote e da redução do intervalo médio de envio de pacotes para a verificação padrão de colisão.

A partir desse experimento, é possível concluir que ao aumentar o tamanho do pacote enviado e diminuir o intervalo médio de envio de pacotes, a rede sofre uma perda

de desempenho quando a verificação de colisão é a padrão, por exemplo, quando o tempo médio de envio é 1000ms e o tamanho do pacote é 255 bytes, o DER é resultante é aproximadamente 0,11, ou seja, muito baixo para aplicações sensíveis como monitoramento de desastres. Entretanto, nos casos em que a verificação completa de colisão está ativada mesmo com a carga útil máxima do pacote sendo utilizada e o tempo médio de envio baixo (1000ms) obtém-se um DER superior à 0,95.

4.10 Discussão sobre os resultados

Esta subseção apresenta uma discussão geral sobre os resultados encontrados. O histograma da Figura 45 representa a frequência dos valores da métrica DER, obtidos para todos os cenários e seus respectivos experimentos, quando utiliza-se a verificação completa de colisão. O histograma da Figura 46 representa a frequência do DER, entretanto para os experimentos que a verificação padrão foi utilizada.

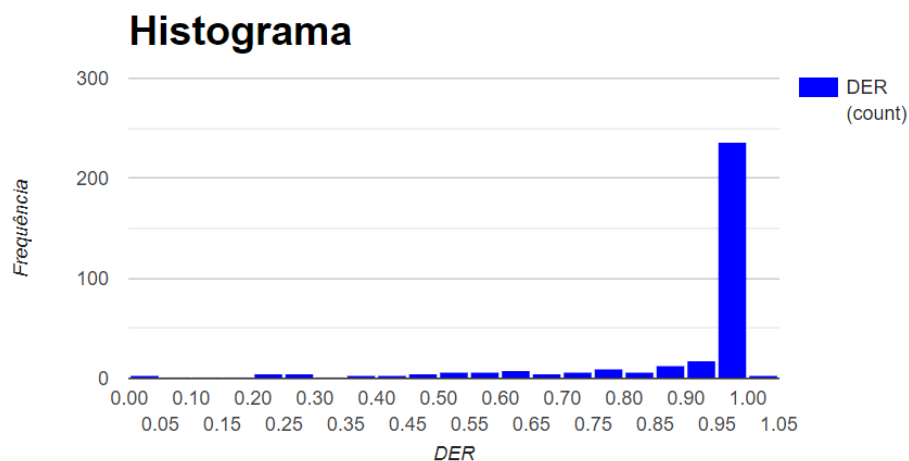


Figura 45 – Frequência de valores da métrica NEC para a verificação completa de colisão.

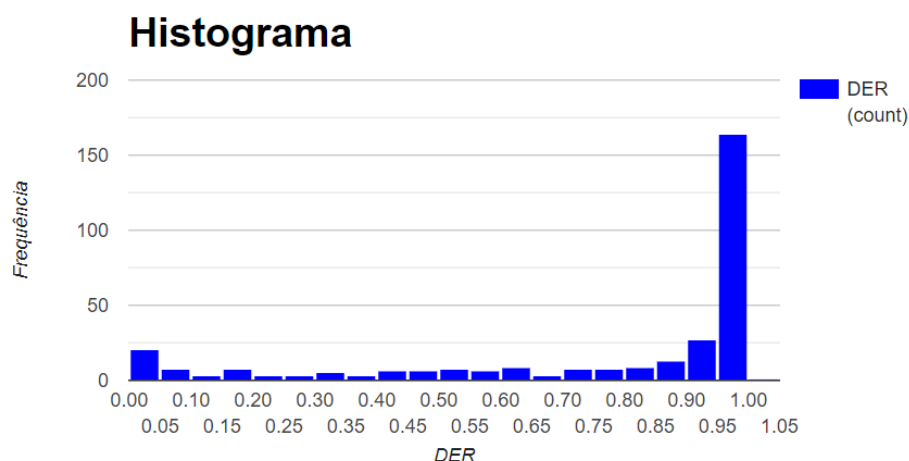


Figura 46 – Frequência de valores da métrica NEC para a verificação padrão de colisão.

O número total de métricas DER obtidas para os cenários que utilizam a verificação completa de colisão é 316, destes, 235 têm valor superior ou igual a 0,95, ou seja, em 74% das configurações de entrada possíveis dos cenários, o desempenho da rede atingiu o valor ótimo, proposto neste trabalho. Entretanto, 26% das configurações de entrada causaram um impacto negativo no desempenho da rede, além disso o DER chegou a valores inferiores a 0,50 em 13 configurações diferentes de parâmetros de entrada.

O número total de valores para a métrica DER obtidos para os cenários que utilizam a verificação padrão de colisão é 300, destes, 137 têm valor inferior a 0,95, ou seja, aproximadamente 46% das configurações de entrada propostas pelos cenários de ataque de disponibilidade causaram impacto negativo no desempenho da rede LoRa no contexto de monitoramento de desastres. Além disso, o DER atingiu valores menores que 0,50 em aproximadamente 19% das vezes. No contexto de monitoramento de desastres, a rede funcionando com apenas 50% de sua capacidade pode impedir que pacotes contendo informações importantes sejam recebidos.

A Tabela 22 indica com um X, os experimentos dos cenários, nos quais o DER atingiu valores menores que 0,95 e menores que 0,50 pelo menos uma vez nos resultados. O valor n/a (Não se aplica), é utilizado quando determinado experimento não foi executado para determinado parâmetro de entrada. Esse comportamento foi analisado tanto para a verificação completa de colisão, quanto para a verificação padrão.

Cenário	Experimento	Verificação completa de colisão		Verificação padrão de colisão	
		DER <0.95	DER <0.50	DER <0.95	DER <0.50
1	1				
1	2			X	X
1	3	X		n/a	
2	1				
2	2			X	X
2	3			X	X
3	1	X		X	
4	1	X	X	X	X
5	1	X	X	X	X
6	1	X	X	X	X
7	1	X	X	X	X
8	1	X		X	X
9	1			X	X

Tabela 22 – Cenários, os quais o DER atingiu valores menores que 0,95 ou menores que 0,50.

A partir dos dados obtidos nas Figuras 45 e 46, na Tabela 22 e dos experimentos executados, é possível concluir que o impacto dos cenários de ataques de disponibilidade na rede, para os casos em que utiliza-se a verificação padrão de colisão é mais alto, do que quando utiliza-se a verificação completa de colisão.

Um exemplo do comportamento descrito anteriormente é o Experimento 2 do Cenário 2, no qual, diminui-se o intervalo médio de envio de pacotes até 100ms. O desempenho da rede para a verificação completa de colisão continua alto mesmo para o pior caso dos resultados, ou seja maior ou igual a 0,95, enquanto para a verificação padrão de colisão, o valor do DER se aproxima de zero no pior caso. Este comportamento ocorre para outros cenários, como o Cenário 9, em que aumenta-se o tamanho da carga útil do pacote e diminui-se o intervalo médio de envio de pacotes.

Observa-se pela descrição dos cenários, resultados e pela Tabela 22, que ao combinar duas variáveis para degradar a rede, torna-se mais fácil deixar a rede indisponível. Por exemplo, os resultados obtidos para os cenários 4, 5, 6 e 7, tanto para a verificação completa de colisão, quanto para a verificação padrão de colisão atingiram um valor inferior a 0,50 do DER nos piores casos.

A partir dos resultados descritos neste capítulo, percebe-se que a redução ou aumento de alguns parâmetros de forma separada pode afetar o desempenho da rede. Por exemplo, o número de estações base no Cenário 3. No pior caso, o parâmetro afetou a rede sozinho para ambas as verificações de colisão, tornando o DER abaixo de 0,95, ou seja, o desempenho deixou de ser ótimo no contexto de monitoramento de desastres. Nos resultados descritos para o Cenário 7, quando aumenta-se a distância máxima entre os

nós e as estações base, obteve-se a métrica DER inferior à 0,50 nos piores casos para os resultados dos experimentos, ou seja, isso ocorreu modificando apenas um parâmetro da rede, tanto para a verificação padrão de colisão, quanto para a verificação completa.

No caso da métrica NEC, percebe-se pelos cenários propostos e experimentos executados que ao aumentar o número de nós, ocorre um crescimento proporcional do consumo de energia em Joule, ou seja, quanto mais dispositivos conectados à rede, maior é o consumo de energia da rede como um todo. Entretanto, isso não impacta na disponibilidade do próprio dispositivo da rede, pois não aumenta o consumo de energia para determinado nó. Já quando trata-se da redução do intervalo médio de envio de pacotes, mais pacotes são enviados pelos nós, logo, estes passam mais tempo transmitindo pacotes do que inativos, o que aumenta o consumo de energia do dispositivo, podendo esgotá-la e tornar o dispositivo indisponível.

Conclui-se que a configuração de uma rede LoRaWAN no contexto de monitoramento de desastres deve ser robusta para garantir que, embora um dos parâmetros da rede seja afetado, os demais ainda conseguirão manter um ótimo desempenho e o monitoramento continue, garantindo a mitigação das consequências do desastre. Recomenda-se a utilização de serviços de monitoramento de tráfego da rede, para a detecção de anomalias e também, o desenvolvimento de planos de contingência e recuperação caso a rede sofra um ataque DoS.

Também conclui-se que o impacto de ataques DoS em redes LoRa pode ser alto, principalmente em casos que utiliza-se apenas a verificação padrão de colisão. Além disso, as variáveis que mais impactam na indisponibilidade da rede são a quantidade de estações base e a distância entre os nós e estações base. Portanto, os gestores de redes LoRa devem se atentar à tais parâmetros.

5 Conclusão

O principal objetivo deste trabalho foi avaliar o impacto de ataques DoS em dispositivos LoRa. Para isso, primeiramente foi necessário obter as configurações para simular o monitoramento do ambiente de desastre proposto, ou seja, a bacia hidrográfica do Córrego do Leitão.

Obtivemos as configurações ideais de simulação de monitoramento para o ambiente determinado, por meio da execução de experimentos no LoRaSim, utilizando diversos parâmetros de entrada na rede LoRa e avaliando as saídas da rede para as métricas DER, que consiste na proporção de pacotes recebidos por pacotes enviados e a métrica NEC, que fornece o consumo de energia da rede LoRa em Joule.

Após definir o *baseline* para monitoramento da bacia hidrográfica do Córrego do Leitão, experimentos foram realizados para analisar o desempenho da rede LoRa, quando os parâmetros de entrada são modificados de modo à degradar a disponibilidade da mesma, ou seja, simulando o efeito de um ataque de negação de serviço.

Conclui-se que os experimentos provocaram alto impacto no desempenho da rede, principalmente quando utiliza-se a verificação padrão de colisão. Entretanto, para a verificação completa de colisão, o desempenho da rede continua ótimo em diversos cenários. Além disso, a partir dos experimentos realizados conclui-se que ao utilizar mais de um parâmetro para atacar a rede, o desempenho da mesma cai consideravelmente.

Para trabalhos futuros, sugere-se utilizar o *baseline* estabelecido nesse trabalho em outro simulador LoRa, no qual a distribuição dos nós não seja aleatória e em seguida testar os cenários propostos para sobrecarregar a rede e observar os resultados. Também recomenda-se focar os experimentos no consumo de energia (NEC), simultaneamente ao (DER), o que não foi possível de ser realizado devido às limitações do *script* loraDir.py. E por fim, para outro trabalho futuro, fica o objetivo de implementar a rede fisicamente e testar alguns parâmetros plausíveis de degradação da rede, como redução do intervalo médio de envio de pacotes, redução da quantidade de estações base e aumento do tamanho do pacote.

Referências

- ADNAN et al. Forest fire detection using lora wireless mesh topology. In: *2018 2nd East Indonesia Conference on Computer and Information Technology (EIConCIT)*. [S.l.: s.n.], 2018. p. 184–187. Citado na página 22.
- AWADALLAH, S.; MOURE, D.; TORRES-GONZALEZ, P. An internet of things (iot) application on volcano monitoring. *Sensors*, v. 19, p. 4651, 10 2019. Citado na página 13.
- BOR, M. et al. Do lora low-power wide-area networks scale? In: . [S.l.: s.n.], 2016. Citado 2 vezes nas páginas 26 e 54.
- BOR, M.; VIDLER, J. E.; ROEDIG, U. Lora for the internet of things. *ACM Digital Library*, Junction Publishing, 2016. Citado 2 vezes nas páginas 13 e 20.
- BUDHOLIYA, A. Efforts in disaster prediction take a step further with iot sensors. *ReadWrite — IoT*, 2018. Citado na página 13.
- CENTELLES, R. P. et al. A lora-based communication system for coordinated response in an earthquake aftermath. *Proceedings*, v. 31, p. 73, 11 2019. Citado 2 vezes nas páginas 22 e 23.
- CISA. *"Understanding Denial-of-Service Attacks"*. 2019. Site CISA. Disponível em: <https://us-cert.cisa.gov/ncas/tips/ST04-015>. Acesso em: 22 out. 2020. Citado 2 vezes nas páginas 17 e 18.
- CLOUDFLARE. *"SYN Flood Attack"*. 2020. Site Cloudflare. Disponível em: <https://www.cloudflare.com/learning/ddos/syn-flood-ddos-attack/>. Acesso em: 22 out. 2020. Citado na página 18.
- DESCONHECIDO. *Information Security*. 2019. Site Wikipedia. Disponível em: https://en.wikipedia.org/wiki/Information_security. Acesso em: 21 out. 2020. Citado na página 17.
- ES, E. van; VRANKEN, H.; HOMMERSOM, A. Denial-of-service attacks on lorawan. In: . New York, NY, USA: Association for Computing Machinery, 2018. (ARES 2018). ISBN 9781450364485. Disponível em: <https://doi.org/10.1145/3230833.3232804>. Citado 4 vezes nas páginas 21, 22, 23 e 25.
- FRUHLINGER, J. *The CIA triad: Definition, components and examples*. 2020. Site CSO Online. Disponível em: <https://www.csoonline.com/article/3519908/the-cia-triad-definition-components-and-examples>. Acesso em: 20 out. 2020. Citado na página 16.
- GARG, R. *What is Information Security?* 2020. Site geeksforgeeks. Disponível em: <https://www.geeksforgeeks.org/what-is-information-security/>. Acesso em: 21 out. 2020. Citado na página 17.

- HUSSAIN, A.; HEIDEMANN, J.; PAPADOPOULOS, C. A framework for classifying denial of service attacks. In: *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*. [S.l.: s.n.], 2003. p. 99–110. Citado na página 14.
- INSTITUTE, S. *Information Security Resources*. 2020. Site SANS Institute. Disponível em: <<https://www.sans.org/information-security/>>. Acesso em: 20 out. 2020. Citado na página 16.
- INSTRUMENTS, N. *Understanding Spread Spectrum for Communications*. 2020. National Instruments. Disponível em: <<https://www.ni.com/pt-br/innovations/white-papers/06/understanding-spread-spectrum-for-communications.html>>. Acesso em: 05 maio. 2020. Citado 2 vezes nas páginas 13 e 20.
- JUNIOR, V. P. da S. "Conheça a tecnologia LoRa® e o protocolo LoRaWAN™". 2016. Site embarcados. Disponível em: <<https://www.embarcados.com.br/conheca-tecnologia-lora-e-o-protocolo-lorawan/>>. Acesso em: 22 out. 2020. Citado 2 vezes nas páginas 20 e 21.
- L.ATZORI, D. A. G. *The internet of things*. [S.l.]: Springer, 2010. Citado 2 vezes nas páginas 13 e 19.
- LAU, F. et al. Distributed denial of service attacks. In: IEEE. *Smc 2000 conference proceedings. 2000 ieee international conference on systems, man and cybernetics.'cybernetics evolving to systems, humans, organizations, and their complex interactions'(cat. no. 0*. [S.l.], 2000. v. 3, p. 2275–2280. Citado na página 14.
- LORAALLIANCE. Lorawan, what is it? a technical overview of lora and lorawan. 2015. Citado 3 vezes nas páginas 20, 21 e 22.
- LORAALLIANCE. What is the lorawan® specification? *Lora Alliance*, 2019. Citado 3 vezes nas páginas 5, 13 e 20.
- MIKHAYLOV, K. et al. Energy attack in lorawan: Experimental validation. In: *Proceedings of the 14th International Conference on Availability, Reliability and Security*. [S.l.: s.n.], 2019. p. 1–6. Citado 4 vezes nas páginas 13, 22, 24 e 25.
- MINAS, G. "BH tem nove pontos críticos de alagamento para o período chuvoso". 2019. G1.Globo. Disponível em: <<https://g1.globo.com/mg/minas-gerais/noticia/2019/09/27/bh-tem-nove-pontos-criticos-de-alagamento-para-o-periodo-chuvoso.ghtml>>. Acesso em: 10 Apr. 2020. Citado na página 33.
- MIRIYALA, T. et al. Iot based forest fire detection system. *International Journal of Engineering and Technology*, v. 7, p. 124, 03 2018. Citado na página 13.
- NETSCOUT. "What is an ICMP Flood Attack?". 2020. Site Netscout. Disponível em: <<https://www.netscout.com/what-is-ddos/icmp-flood>>. Acesso em: 22 out. 2020. Citado na página 18.
- PALOALTONETWORKS. "What is a denial of service attack (DoS) ?". 2020. Site Paloalto Networks. Disponível em: <<https://www.paloaltonetworks.com/cyberpedia/what-is-a-denial-of-service-attack-dos>>. Acesso em: 22 out. 2020. Citado 2 vezes nas páginas 17 e 18.

- RAGNOLI, M. et al. An autonomous low-power lora-based flood-monitoring system. *Journal of Low Power Electronics and Applications*, v. 10, 05 2020. Citado 3 vezes nas páginas 5, 22 e 37.
- RAVI, G. Earthquake early warning system by iot using wireless sensor networks. In: . [S.l.: s.n.], 2016. Citado na página 13.
- ROSA, D. W. B. Resposta hidrológica de uma bacia hidrográfica urbana à implantação de técnicas compensatórias de drenagem urbana - bacia do córrego do leitão, belo horizonte, minas gerais. p. 220, 4 2017. Citado 2 vezes nas páginas 5 e 33.
- ROUSE, M. "DEFINITION internet of things (IoT)". 2016. Site techtarget. Disponível em: <<https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT>>. Acesso em: 22 out. 2020. Citado 3 vezes nas páginas 5, 19 e 20.
- SALLUM, E. et al. Improving quality-of-service in lora low-power wide-area networks through optimized radio resource management. *Journal of Sensor and Actuator Networks*, v. 9, p. 10, 02 2020. Citado na página 27.
- STALLINGS, W. *Cryptography and Network Security: Principles and Practice*. 5th. ed. USA: Prentice Hall Press, 2010. ISBN 0136097049. Citado 2 vezes nas páginas 16 e 17.
- WALKOWSKI, D. *What is the CIA Triad?* 2020. Site f5. Disponível em: <<https://www.f5.com/labs/articles/education/what-is-the-cia-triad>>. Acesso em: 21 out. 2020. Citado na página 17.
- WHITMAN, M. E.; MATTORD, H. J. *Principles of Information Security*. 4th. ed. Boston, MA, USA: Course Technology Press, 2011. ISBN 9781111138219. Citado 2 vezes nas páginas 16 e 17.