

INSTITUTO FEDERAL DE SANTA CATARINA

VICTOR CESCONETTO DE PIERI

**Simulação de uma rede LoRa: subsídios a
operação com alta densidade de nodos**

São José - SC

Agosto/2022

Victor Cesconetto De Pieri

Simulação de uma rede LoRa: subsídios a operação com alta
densidade de nodos

São José - SC, 27 de agosto de 2022:

Eraldo Silveira e Silva
Eraldo Silveira e Silva
Instituto Federal de Santa Catarina

RESUMO

Os sistemas de IoT LoRa vem sendo amplamente utilizados para comunicação e transmissão de dados em uma variedade de aplicações, sendo a mais comum a de monitoramento com sensores. Com o uso dessa tecnologia é possível monitorar locais de difícil acesso com seu longo alcance sem a necessidade de trocar as baterias dos dispositivos frequentemente pelo seu baixo consumo de energia, além de seu baixo custo, o que torna maior a viabilidade de implantação. O objetivo deste trabalho é comparar um sistema real LoRa em funcionamento com um sistema simulado utilizando o framework FLoRa junto com a ferramenta Omnet++ e utilizar métricas como consumo de bateria, alcance, taxa de transferência de dados e tempo de transmissão. Utilizando essas métricas para a rede real e simulada, possibilita-se a análise da viabilidade de projeto, além disso, na simulação é possível aumentar o número de nós, distanciamento maior dos *gateways*, mudança de ambiente e uma possível adição de mobilidade aos nós.

Palavras-chave: FLoRa. Omnet. IoT. LoRaWAN. LPWAN.

LISTA DE ILUSTRAÇÕES

Figura 1 – Aplicações LPWAN	9
Figura 2 – Estrutura de quadros LoRa	10
Figura 3 – Rede LoRa	13
Figura 4 – Classe A - Nenhuma recepção de dados	14
Figura 5 – Classe A - Rx1 recebendo dados	14
Figura 6 – Classe A - Rx2 recebendo dados	14
Figura 7 – Módulos OMNet++	16
Figura 8 – Estrutura INET	18
Figura 9 – Estrutura de módulos FloRa	19
Figura 10 – Fatores de simulação no OMNet++	20

SUMÁRIO

1	INTRODUÇÃO	5
1.1	Objetivo Geral	6
1.2	Objetivos específicos	6
2	FUNDAMENTAÇÃO TEÓRICA	7
2.1	LPWAN e LoRa	7
2.1.1	LPWAN	7
2.1.2	LoRa	8
2.1.3	LoRaWAN	10
2.2	Simulação de redes LoRa e o simulador Omnet	15
2.2.1	Avaliação de desempenho por simulação	15
2.2.2	Omnet++ e INET	16
2.2.3	FloRa	17
3	PROPOSTA	21
3.1	Experimento de comparação com cenário real	21
3.1.1	Coleta de dados	22
3.1.2	Construção de um cenário de simulação	22
3.1.3	Análise de resultados do experimento	22
3.2	Experimento de simulação expandida	22
3.2.1	Aumento da densidade da rede	22
3.2.2	Expansão da mobilidade da rede e uso do algoritmo ADR	23
3.2.3	Análises de resultados do experimento	23
3.3	Cronograma	23
	REFERÊNCIAS	24

1 INTRODUÇÃO

A implementação de projetos utilizando tecnologias *Low Power Wide Area Network* (LP-WAN) é uma das alternativas da Internet das coisas(IoT). Como já existem inúmeras soluções de LP-WANs, fica fácil a escolha de uma tecnologia que melhor se adapte a um determinado cenário com soluções de baixo custo. As tecnologias são variadas, todas tem seus prós e contras, algumas demandam mais custo de bateria, outras menos, assim como taxa de transferência entre outras.

Entre as tecnologias de LP-WAN, existe o LoRa(Long Range), que é utilizado em muitos projetos que visam a baixa frequência de manutenção e a maior durabilidade de bateria, para conseguir manter uma comunicação em locais de difícil acesso. Existem alguns desafios para projetar uma rede LoRa atualmente, dentre eles: um número grande e altamente variável de nós; diversos cenários sem fio caracterizados por fatores ambientais exigentes (por exemplo, ambientes urbanos densos com muitos edifícios); interferência devido a outras redes co-localizadas operando nas mesmas bandas de frequência às vezes não-licenciadas(BOR et al., 2016).

Como existem variadas tecnologias de redes e uma diversidade de cenários de uso, é comum o uso de ferramentas para apoiar o projeto de rede. O uso de simuladores de redes de computadores é frequente e permite avaliar o desempenho de redes, apoiar a configuração e o posicionamento de dispositivos, ajustar a operação, tomar decisões de expansão da infraestrutura e testar novos algoritmos e protocolos. Neste sentido, os acadêmicos da universidade de Aalto na Finlândia criaram um *framework* de simulação chamado FloRa. O FLoRa implementa as camadas física e de controle de acesso médio do LoRa, suporta comunicações bidirecionais e permite simulações de ponta a ponta, incluindo a rede de backhaul segundo (SLABICKI GOPIKA PREMSANKAR, 2018), utiliza como base o Omnet++, que é uma biblioteca da linguagem C++ muito robusta para simulações de projetos de rede e possui muita precisão.

Neste trabalho pretende-se construir um comparativo entre uma rede real e uma rede simulada através de uma análise de métricas associadas ao consumo de energia, cobertura de rede, vazão, dentre outras, visando inicialmente verificar se a rede simulada se aproxima da real. A rede real escolhida foi uma que já está sendo implementada em outro trabalho acadêmico no IFSC-SJ, facilitando as análises e a escolha das métricas.

Além disso também será feito uma simulação de um cenário expandido, alterando as configurações iniciais da rede, aumentando o número de nodos, incluindo mobilidade de nodos e usando taxa de dados adaptativo, para apoiar a construção de uma rede maior e mais robusta.

1.1 Objetivo Geral

Simular um cenário real de uma rede LoRa implantado no IFSC São José, utilizando métricas como *taxa de transmissão*, *consumo de bateria*, *tempo de transmissão (time-on-air)* e *distância máxima*. Os resultados da simulação poderão ser usados para dar subsídios à operação da rede real e em possíveis cenários expandidos.

1.2 Objetivos específicos

- Simular um cenário real de rede Lora, comparando os resultados com as medições realizadas em nodos físicos.
- Incluir na simulação o aumento da densidade de nodos, e outros fatores tais como: carga adicional de dados, algoritmo de ADR e mobilidade, de forma a avaliar situações que normalmente não seriam possíveis no cenário real.
- Analisar os resultados da simulação e fornecer subsídios à operação da rede LoRa no cenário real investigado e em cenários expandidos

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão apresentados os conceitos básicos e as tecnologias citadas por este trabalho, como na simulação será utilizado conceitos de LPWAN, utilizará-se de uma aplicação LPWAN específica usando como camada física o LoRa. A simulação será desenvolvida com o framework Omnet++ junto com o INET que é o framework para simulação de redes, e por fim juntando-se ao FloRa que é um outro framework para simulações de aplicações LoRa especificamente.

2.1 LPWAN e LoRa

2.1.1 LPWAN

Muito consolidada atualmente, o uso da tecnologia de redes de baixa potência LPWAN é muito comum em projetos para IoT. O poder de conectar muitos dispositivos simultaneamente, junto de seu baixo custo energético e seu alto alcance, LPWANs garantem um acesso a informações em locais de difícil acesso, apenas utilizando um sensor, já que os gateways podem ficar bem distantes dos nós. As tecnologias LPWAN podem ser utilizadas para uma ampla gama de aplicações, incluindo monitoramento ambiental, cidades inteligentes, serviços públicos inteligentes, agricultura, saúde, automação industrial, rastreamento de ativos, logística de transporte e muito mais(CHAUDHARI; ZENNARO; BORKAR, 2020).

As aplicações de LPWAN são muitas, cada solução LPWAN pode cobrir um determinado número de aplicações de acordo com o seu uso específico. Por exemplo, aplicações de cidades inteligentes requerem ampla cobertura de rádio para abranger todas as partes de uma cidade. Há muitos parâmetros que podem ser observados para uma solução LPWAN, o livro (CHAUDHARI; ZENNARO, 2020) cita 5 deles sendo os mais importantes: cobertura, capacidade, Custo, consumo e características adicionais de acordo com a Figura 1.

O longo alcance das redes LPWAN são possíveis por diversos fatores, um deles por exemplo é o uso de bandas abaixo dos GHz o que permite um comprimento de onda maior, possibilitando uma área maior de cobertura. O baixo custo de bateria é proporcionado por um *duty cycle* reduzido, os mecanismos *duty cycle* são adaptados baseados na aplicação. Se a aplicação precisa enviar ou receber mensagens os dispositivos finais acordam e enviam a mensagem, poupando bateria e ativando só quando necessário.(RAZA; KULKARNI; SOORIYABANDARA, 2017) Existem diversas tecnologias LPWAN que tem *duty cycle* diferentes, por exemplo a *LoRaWAN* tem classes de dispositivos que se diferenciam no

duty cycle.

Outro fator que também contribui para as redes LPWAN serem muito utilizadas em *IoT* é a simplicidade dos protocolos *MAC*. Muitas outras tecnologias wireless (redes celulares e wifi), utilizam uma complexidade maior nesses protocolos. O *CSMA/CA* um protocolo *MAC* muito comum nas tecnologias acaba não servindo para as LPWANs por elas terem uma quantidade excessiva de dispositivos finais, por isso, as LPWANs mais comuns como *LoRaWAN* e *SigFox* utilizam o protocolo *MAC* chamado *ALOHA*. Como existem muitos dispositivos finais, a maioria das tecnologias LPWAN optam por reduzir a complexidade deles. Eles apenas servem como receptores e transmissores de dados. A complexidade da rede é voltada para as estações radio base, onde a recepção e transmissão é feita por multicanais ou sinais ortogonais simultaneamente. (RAZA; KULKARNI; SOORIYABANDARA, 2017).

As redes LPWAN normalmente utilizam uma topologia de rede em estrela, onde os nós se comunicam com uma estação base (*gateway*) sem a necessidade da comunicação entre si, utilizando apenas um salto. Na maioria das soluções o formato *nó-gateway-servidor* de conexão é utilizado, com a comunicação do gateway até o servidor utilizando uma tecnologia diferente, como redes IP. No caso das LPWAN cada aplicação específica é utilizado uma tecnologia diferente, por exemplo para soluções de curto e medio alcance existem o ZigBee e o Sigfox que são tecnologias desenvolvidas para esse tipo de cenário. Quando se trata de longo alcance o LoRa é uma das LPWAN mais utilizadas, muito utilizada em locais de difícil acesso e permite uma longa distância entre os nós e os gateways, além de sua longa vida de bateria e seu custo ser muito baixo. Com esses pontos podemos observar que dentre as características mais importantes da LPWAN o LoRa consegue abranger todas com suficiência, e sendo assim, como nosso projeto também utiliza LoRa, a seguir discutiremos essa tecnologia.

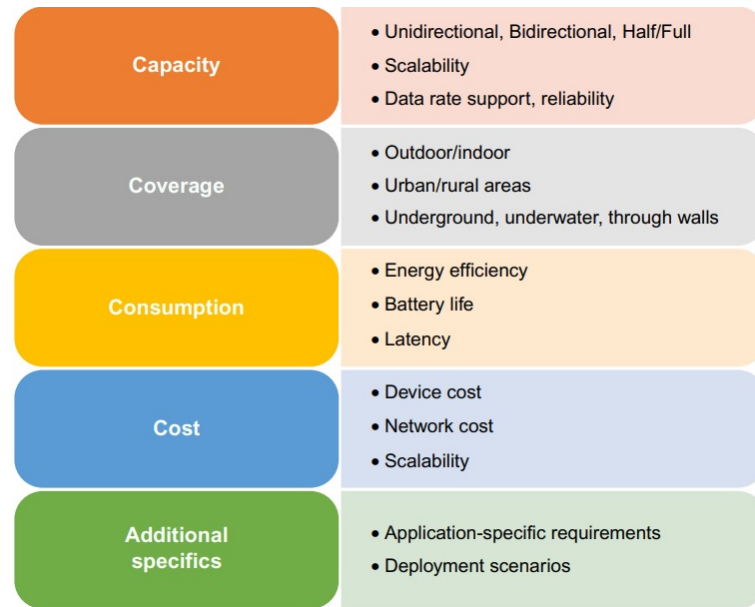
2.1.2 LoRa

LoRa é uma técnica de *Spread Spectrum Modulation*, derivada da *Chirp Spread Spectrum (CSS)* integrada com um código de correção de erros desenvolvida pela Semtech. O LoRa oferece um equilíbrio entre sensibilidade da recepção e taxa de dados, opera em um canal de largura de banda fixa de 125 KHz nos canais para uplink e 500 KHz nos canais de *downlink*. (SEMTECH, 2019)

A camada física permite longas distâncias, comunicação de baixa potência e opera em uma banda não-licenciada ISM sub-GHz. Como permite longas distâncias a taxa de bits tende a ser baixa. Segundo o artigo (BOR et al., 2016) a comunicação LoRa é baseada em cinco fatores principais :

- **Fator de Espalhamento** (*Spreading Factor*) : O fator de espalhamento é a razão

Figura 1 – Aplicações LPWAN



Fonte: (CHAUDHARI; ZENNARO, 2020)

entre a taxa de símbolos e a taxa de ruído. O fator de espalhamento está diretamente relacionado à taxa de dados e a distância alcançada das conexões. Taxas de dados menores estão relacionadas com fatores de espalhamento mais altos, o que resulta em uma distância maior de comunicação. Entretanto, quanto maior o fator de espalhamento, maior a relação sinal-ruído (SNR) isso implica em um tempo maior de *airtime*. Escolhendo diferentes fatores de espalhamento, pode-se obter sinais ortogonais permitindo que o receptor possa com sucesso receber diferentes sinais de um mesmo canal ao mesmo tempo.

- **Potência de Transmissão** (*Transmission Power*): A potência de transmissão pode ser definida de acordo com a largura de banda da região. O rádio LoRa pode ser ajustado de -4 dBm para 20 dBm , e passos de 1 dB . Isso é devido aos limites de hardware, além disso as potências mais altas que 17 dBm só podem ser usadas a 1 por cento de *duty cycle*.
- **Taxa de código** (*Coding Rate*): Taxa de código é a taxa de código corretor de erros usado pelo modulador LoRa que oferece proteção contra ondas de interferência, podem ser configurados entre $4/5$ e $4/8$. Quanto maior proteção de taxa de código maior o *airtime* da transmissão. Rádios com diferentes taxas de código (com a mesma frequência da portadora, fator de espalhamento e largura de banda), podem comunicar entre si usando um *header* explícito, onde a taxa de código dos dados do payload são armazenados no *header*.
- **Frequência da portadora** (*Carrier Frequency*): A frequência da portadora depende das bandas licenciadas da região onde será utilizado o rádio. Pode ser programada

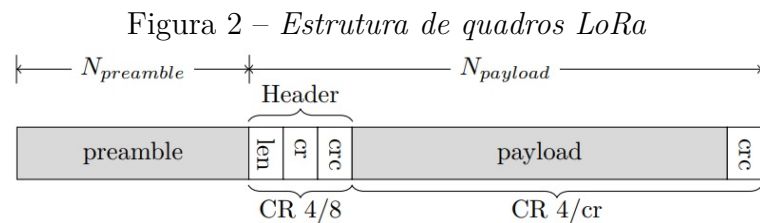
em passos de 61 Hz , entre 137 MHz até 1020 MHz e esse limite pode ser reduzido para 860 MHz dependendo do chip LoRa utilizado.

- **Largura de banda**(*Bandwidth*) : A largura de banda influencia diretamente na taxa de dados das transmissões. Uma grande largura de banda proporciona uma alta taxa de dados, porém reduz a sensibilidade do receptor por ter um ruído adicional. Já uma largura de banda menor permite uma sensibilidade mais alta e uma menor taxa de dados. A largura de banda pode ser configurável de 7.8 kHz até 500 kHz , uma rede LoRa comumente opera em 500 kHz , 250 kHz ou 125 kHz .

Alterando esses fatores, podemos criar vários cenários de rede onde podemos avaliar as taxas de dados, consumo de energia e resiliência aos ruídos.

A estrutura de pacotes do LoRa demonstrado na Figura 2, inicia com o *preâmbulo*, programável de 6 a 65535 *símbolos*, somados com o rádio, que adiciona mais 4.25 *símbolos*. Ainda possui um *header* opcional, que descreve o tamanho e a taxa de código de correção de erros do payload. O *payload* que contém de 1 até 255 *bytes* vem logo após o *header* opcional, o header que é sempre transmitido, utiliza uma taxa de correção de erros de 4/8 e tem seu próprio *CRC*(BOR et al., 2016).

O *airtime* da transmissão depende da combinação do tamanho do *payload*, fator de espalhamento, largura de banda e da taxa de correção de erros. Dependendo de cada configuração de transmissão o *airtime* pode variar, por exemplo um pacote com 20 bytes pode variar de 9 milissegundos a 2,2 segundos.



2.1.3 LoRaWAN

Para explicar melhor o funcionamento do LoRa, precisamos falar do LoRaWAN, que se constitui nas camadas de rede das redes LoRa, ou seja, é o protocolo da camada de rede(de acordo com as 7 camadas padrão da internet) que garante o funcionamento dos dispositivos que usam a camada física LoRa. Os *nós* LoRaWAN podem ser sensores ou atuadores que são conectados aos gateways por meio de *RF* utilizando a modulação LoRa.

Na maioria das aplicações LoRaWAN os nós são autônomos, operados por bateria, eles digitalizam as leituras do ambiente e as enviam ao gateway. Quando são produzidos, os

dispositivos LoRa possuem muitos identificadores únicos, esses identificadores são usados para administrar o dispositivo e ativá-lo com segurança. Esses identificadores únicos também são usados para o envio de dados criptografados a nuvem.(SEMTECH, 2019)

A especificação do LoRaWAN define um padrão aberto aos protocolos de rede da arquitetura de redes LoRa, que são baseados no protocolo ALOHA de enlace. A arquitetura de rede consiste em uma topologia hierárquica onde os *nós* LoRa comunicam com os *gateways* e os *gateways* se comunicam com os Servidores de rede LoRa(*Lora Network Server(LNS)*) conforme a [Figura 3](#).

Cada nó não está necessariamente associado a um *gateway*, qualquer *gateway* que é alcançado pelo nó pode receber dados. Os *gateways* trabalham como encaminhadores de mensagem, recebem as mensagens dos nós, e as enviam para o servidor LoRa. Os servidores LoRa por sua vez enviam as mensagens com os dados coletados pelos nós aos servidores de aplicação (SLABICKI GOPIKA PREMSANKAR, 2018).

Os pacotes de *uplink* enviados por um nó são recebidos por todos os *gateways* que ele alcançar. Esse esquema reduz muito o erro de pacotes e o consumo de bateria. Já os *gateways* enviam os dados ao *LNS* via tráfego IP, que pode ser por cabo *Ethernet*, *Wi-Fi* ou conexões celulares. Os *gateways* na rede LoRa funcionam apenas como encaminhadores de mensagens, eles somente checam a integridade dos dados recebidos via RF.

Se os dados não estão íntegros, ou seja, o *CRC* está incorreto, a mensagem é descartada. Se está correta, o *gateway* encaminha ao *LNS*, junto com os metadados que incluem o nível RSSI da mensagem, bem como um *timestamp* adicional. Para pacotes de *downlink*, o *gateway* executa as requisições de transmissão vindas do *LNS* sem nenhuma leitura do *payload*. Enquanto múltiplos *gateways* recebem a mesma mensagem LoRa de RF, o *LNS* faz o trabalho de remover as mensagens duplicadas. Baseado no nível RSSI das mensagens idênticas, o *LNS* seleciona o *gateway* que recebeu a mensagem com o melhor RSSI que implica no *gateway* mais próximo do nó.(SEMTECH, 2019)

O gerenciamento da rede LoRaWAN é feito pelo *LNS*, ele ajusta a rede para qualquer condição adversa possível, possui as configurações de funcionamento da rede e estabelece conexões seguras usando o algoritmo de 128-bits *AES* para o transporte dos dados fim-a-fim(dos nós as aplicações em nuvem). Também controla o tráfego que flui até o nó e vice-versa, garantindo que a recepção e a transmissão dos dados seja transportada pelo melhor sinal de transmissão, além de verificar a autenticidade de todos os sensores na rede, a integridade de todas as mensagens, e por fim, o *LNS* não tem acesso a nenhum dos dados da aplicação.

Segundo (SEMTECH, 2019) os *LNSs* em sua maioria contém:

- Checagem de endereço de dispositivo

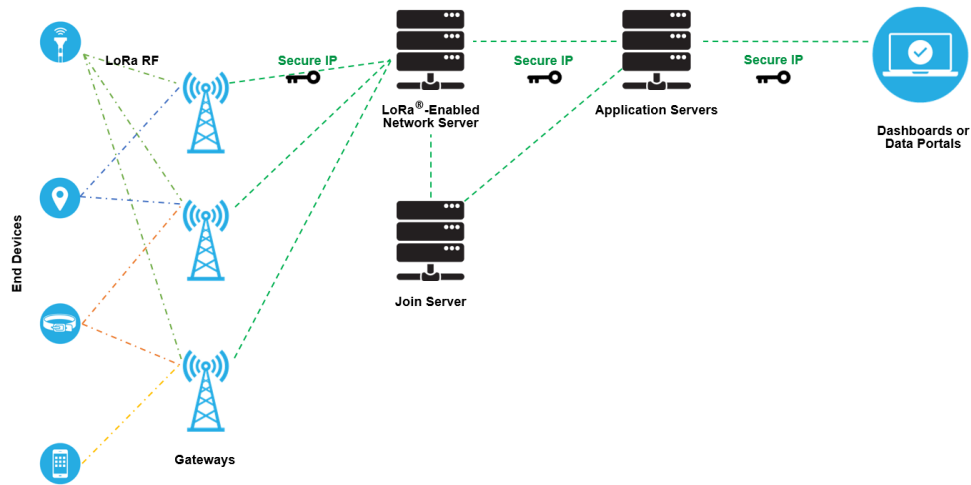
- Autenticação dos quadros e gerenciamento de contador de quadros
- Confirmações de mensagens recebidas
- Adaptação de taxa de dados usando o protocolo *ADR*
- Responde a todas as requisições dos dispositivos vindas da camada de enlace.
- Encaminha os *payloads* de uplink para os *servidores da aplicação* correspondente.
- Enfileira os *payloads* de donwlink vindos de qualquer *servidor de aplicação* para qualquer dispositivo conectado na rede.
- Encaminha mensagens de requisição e permissão entre os dispositivos e o *servidor de ingresso*(Para a versão 1.1 do LoRaWAN).

Os *servidores de aplicação(AS)* são responsáveis pelo gerenciamento do sistema. Eles possuem o papel de receber as informações dos dados dos sensores e para isso precisam de uma implementação livre de erros, utilizando ferramentas para o manuseio dos dados vindos dos sensores. Sendo assim, como os gateways garantem o encaminhamento dos dados ao servidor, sem modificá-los, o papel do *AS* é garantir que esses dados sejam consistentes, provendo além de uma atomicidade de dados, um nível de segurança adequado usando dados do dispositivo final como :

- *DevEUI* (Número serial de identificador único do *nó*)
- *AppKey* (Chave criptográfica de aplicação)
- *NETID* (Identificador de rede)
- *AppSKey* (Chave criptográfica de sessão)
- *NwkSkey* (Chave criptográfica do *LNS*)

Para adicionar um novo dispositivo final(*nó*) o *AS* checa se a entrada desse dispositivo específico está na lista de *dispositivos suportados*, verificando se o *DevEUI* do *nó* e sua *AppKey* combinam. Após uma verificação de sucesso, o *servidor de aplicação* responde para o *LNS* um *AppNonce*(valor randômico ou identificador único fornecido por um *LNS*). Assim o *LNS* concatena um *Network ID(NETID)* e também alguns parâmetros de configuração dentro do código de integridade de mensagem *MIC* para enviar de volta ao *nó*. O *nó* valida o *MIC* e descriptografa a mensagem para obter o *AppNonce* e o *NETID* que são usados para criar as chaves de sessão *AppSKey* (ELDEFRAWY et al., 2019). Os novos *nós* podem ser sincronizados com o sistema de duas maneiras, uma é *OTAA(Over-the-air Activation)*, esse modo de ativação corresponde a ativação remota

Figura 3 – Rede LoRa



Fonte: (SEMTECH, 2019)

do nó; a outra é (ABP) *Activation By Personalization*, esse modo corresponde a ativação manual onde as chaves são pré-configuradas no nó.

Os nós LoRaWAN operam em três modos diferentes : A,B e C. Todos os dispositivos devem permitir o suporte para a classe A e os classe B devem ter suporte para classes A e B, já os de classe C devem suportar todas as classes. Esses modos de operação definem como os nós irão se comunicar na rede.

- **Classe A** : Nos dispositivos classe A a maior parte do tempo eles se encontram no estado ocioso e permitem transmissões bidirecionais. Quanto observam uma mudança no ambiente seja relacionado a qualquer coisa que o dispositivo monitora, ele acorda e inicia o *uplink*, retornando os dados sobre as informações das mudanças pela rede.

Para a resposta ele possui duas janelas, Rx1 e Rx2, entre as janelas há um intervalo de *sleep* configurável geralmente em termos do delay da transmissão de *uplink* e cada janela opera com uma duração de um segundo(tipicamente configurado). Se o dispositivo não receber mais nada nessas janelas de Rx, ele entra em *sleep* novamente até ser acordado recebendo dados para enviar novamente(SEMTECH, 2019). Se o *downlink* da transmissão acontece durante a primeira janela(Rx1) o mesmo canal que está sendo usado por *uplink* é usado para *downlink* também, enquanto o *Fator de espalhamento* é determinado baseado no parâmetro *RX1DROffset*. No caso do segunda janela ser usada para *downlink*, um canal com um *Fator de espalhamento* fixo é usado.

É responsabilidade do LNS escalonar o tráfego do *downlink* no tempo exato para o

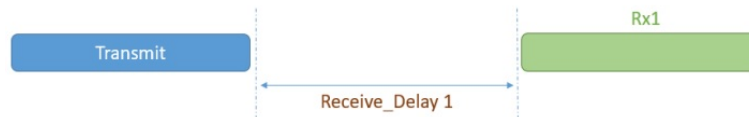
sucesso do controle de tempo. Os dispositivos Classe A por frequentemente estarem em *sleep*, são os que menos consomem bateria (HAXHIBEQIRI et al., 2018).

Figura 4 – Classe A - Nenhuma recepção de dados



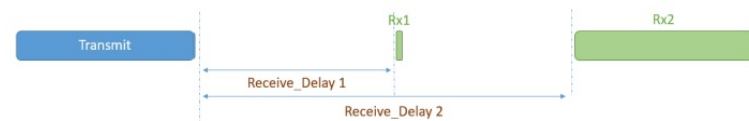
Fonte: (SEMTECH, 2019)

Figura 5 – Classe A - Rx1 recebendo dados



Fonte: (SEMTECH, 2019)

Figura 6 – Classe A - Rx2 recebendo dados



Fonte: (SEMTECH, 2019)

- **Classe B** : Também operando em classe A os dispositivos classe B, também chamados de *beacons* classe B, tem em seu funcionamento um processo chamado *beaconing*. Esse processo o *beacon* sincronizado com o tempo precisa fazer um *broadcast* pela rede pelos gateways, onde os *nós* alinham seu tempo de acordo com a referência imposta pela rede (SEMTECH, 2019). Em relação a bateria os dispositivos de classe B, têm um consumo grande em relação aos dispositivos de classe A, porque precisam abrir mais janelas de recepção que podem ser apenas de *beacons*, não necessariamente de um *downlink*.
- **Classe C** : Os dispositivos classe C também implementam a classe A, além de ter sua comunicação sempre ativa, sem *sleep*. O único momento que os dispositivos classe C estão impossibilitados de receber dados, são quando estão em modo de transmissão. O funcionamento da recepção é semelhante a classe A, porém na segunda janela (Rx2) a recepção nunca é desligada até alguma informação ser transmitida. (HAXHIBEQIRI et al., 2018) O consumo de bateria dos dispositivos classe C é o maior dentre os tres modos de operação, justamente por seu funcionamento nunca ter um *sleep* e sempre estar pronto para receber novos *downlinks*.

O protocolo LoRaWAN facilita muitas configurações de *nós*. Além de ter uma certa facilidade com a adição de um novo dispositivo na rede, o LoRaWAN possui um instrumento chamado *Adaptive Data Rate(ADR)*, que gerencia dinamicamente os parâmetros do *nó* no objetivo de aumentar a taxa de entrega de pacotes. O *ADR* gerencia a taxa de dados e a potência de transmissão dos *nós*, para caso um dispositivo entrar na rede e o *uplink* for configurado pelo *LNS* o *ADR* adicionará um *uplink ADR bit*. O *ADR* se encarrega de configurar automaticamente as taxas de *uplink* do *nó* caso o dispositivo não for configurado pelo *LNS*. O *ADR* do *nó* e do *LNS* rodam assincronamente(HAXHIBEQIRI et al., 2018).

2.2 Simulação de redes LoRa e o simulador Omnet

Para o propósito deste estudo foi escolhido o simulador *OMNet++*, que mais a frente será explicado, permitindo com seu framework *INET* desenvolver uma simulação de rede robusta, onde pode-se selecionar protocolos de rede entre outros parâmetros. Para as simulações de redes LPWAN e mais especificamente redes LoRa, existe uma ferramenta dentro do *Omnet++*, o *FloRa*, que será explicado adiante, é um outro framework que trabalha junto com o *INET* possibilitando uma implementação fiel ao sistema LoRa que será utilizado no projeto.

2.2.1 Avaliação de desempenho por simulação

Para avaliar o desempenho de um sistema pode-se utilizar três técnicas: modelo analítico, análises por simulação e um sistema real ou uma combinação dessas três. Em particular, a simulação é interessante pois permite estudar cenários que normalmente não poderiam ser avaliados em experimentos reais de forma simples. Com métricas pré-estabelecidas pode-se construir uma simulação com muitas variáveis em um ambiente desejado e assim fazer testes muito próximos do mundo real, permitindo ajustar pontos de funcionamento, prevendo comportamento de expansões da rede e otimizando o funcionamento da mesma.

Nesta seção será exposto quais os conceitos para construir uma simulação de um sistema de eventos discretos, para a construção de uma simulação com o framework *OMNet++* junto com o *INET* e o *FloRa*.

A partir do *OMNet++* precisamos de outro framework que implementa as simulações de rede, o *INET*, que permite construção de simulações de redes utilizando protocolos já conhecidos, como por exemplo : TCP, UDP, IP e IEEE 802.11.(INET-DOCUMENTATION, 2022) E além disso temos o *FloRa* que é um framework para simulações de redes LoRa, criado pela universidade Aalto na Finlândia. Esse framework permite criar simulações LoRa fiéis, que permite a análise de desempenho de quaisquer redes LoRa em ambientes variados.

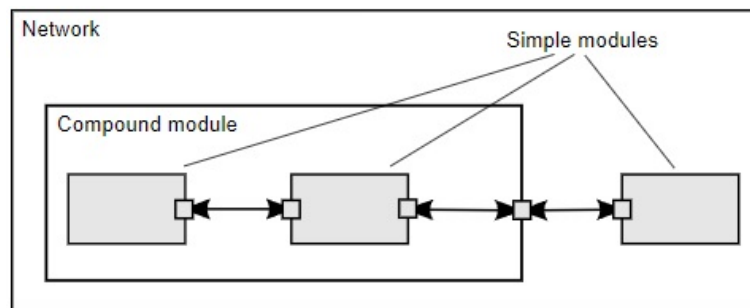
2.2.2 Omnet++ e INET

Uma das ferramentas a ser utilizadas é o *OMNet++* desenvolvido pela OpenSim, uma robusta biblioteca *opensource* para a linguagem C++ que permite construir simulações de todo o tipo e é especializado em simulações de tempo discreto. Um sistema de eventos discretos é um sistema onde mudanças de estado (eventos) acontecem em instâncias discretas no tempo, e os eventos levam tempo zero para acontecer. Não é um simulador de uma aplicação ou sistema em particular, mas ele proporciona uma arquitetura flexível e genérica para escrever ferramentas de simulação(KIM et al., 2010).

O modelo de sistema do *OMNet++* possui módulos onde eles se comunicam com a passagem de mensagens. As mensagens podem ser enviadas por meio de conexões que abrangem módulos(módulo composto) ou diretamente para outros módulos. O modelo em si é chamado de rede no *OMNet++*, e os módulos ficam agrupados em algo chamado *módulo composto* de acordo com a Figura 7.(VARGA; LTD, 2021).

Os modelos *OMNet++* são organizados hierarquicamente e dentro desses modelos existem módulos conectados que se comunicam entre si. O módulo *top-level* é o *módulo do sistema*, é o módulo principal, onde contém os sub módulos que podem conter outros submódulos, esses por sua vez são chamados de *módulos compostos*.

Figura 7 – Módulos OMNet++



Fonte: (VARGA; LTD, 2021)

Cada módulo contém uma implementação em C++ usando a biblioteca *OMNet++*(VARGA; LTD, 2021). quando montados em maiores componentes usando uma linguagem de alto nível(*NED - Network Description*). Os módulos não podem chamar funções de outros módulos, apesar de serem conectados. Entretanto, eles interagem passando *mensagens* pelas *portas* de conexão. Essas *mensagens* podem carregar estrutura de dados, incluindo pacotes de comunicação de rede. A arquitetura do *OMNet++* que desenvolvedores encapsulam um leque de funções em um componente que se assemelha a interação entre entidades em sistemas distribuídos(KIM et al., 2010).

As *mensagens* representam quadros ou pacotes de uma rede, trabalhos ou clientes numa rede de filas ou em outros tipos de entidades móveis. As *mensagens* podem conter uma complexa estrutura de dados arbitrários(VARGA; LTD, 2021). O *gates* ou *portas*

são as interfaces de entrada e saída dos módulos, as *mensagens* passam por eles para irem de um módulo para o outro. Cada conexão ou *link* é criada com um único nível de hierarquia de módulos, dentro de um módulo composto por exemplo, pode-se conectar as *portas* correspondentes de dois submódulos, ou a *porta* de um submódulo e a *porta* de um módulo composto, de acordo com a figura 7(VARGA; LTD, 2021).

Os módulos possuem parâmetros, que podem ser configurados por *NED* ou por arquivos de configuração. Os parâmetros podem ser usados para customizar um simples módulo ou para parametrizar a topologia do modelo. Usam de variáveis ou métodos desenvolvidos em *C++* para iniciar junto com um módulo, além disso, pode-se definir o número de submódulos de um módulo composto e as rotinas que cada um irá executar em cada um deles.

Uma das ferramentas do *OMNet++* mais utilizadas para simulação de redes de computadores é o *INET*, como já dito anteriormente, ele implementa os protocolos da camada de rede, enlace e roteamento, o que facilita a construção de módulos de simulação para aplicações de rede. Os protocolos são representados como módulos, e podem ser combinados para implementar *hosts* e dispositivos de rede como *roteadores*, *switchs*, *access points* entre outros.

Na Figura 8, temos um exemplo de uma aplicação sem fio construída com o *OMNet++* usando *INET*. Dentro do *host* os módulos para os protocolos são conectados de acordo com suas camadas no modelo de rede. Os módulos das camadas superiores representam as aplicações, os módulos nas camadas do meio implementam os protocolos de transporte e por fim os módulos na camada inferior é um módulo composto que implementa os protocolos de enlace e rede.(KIM et al., 2010)

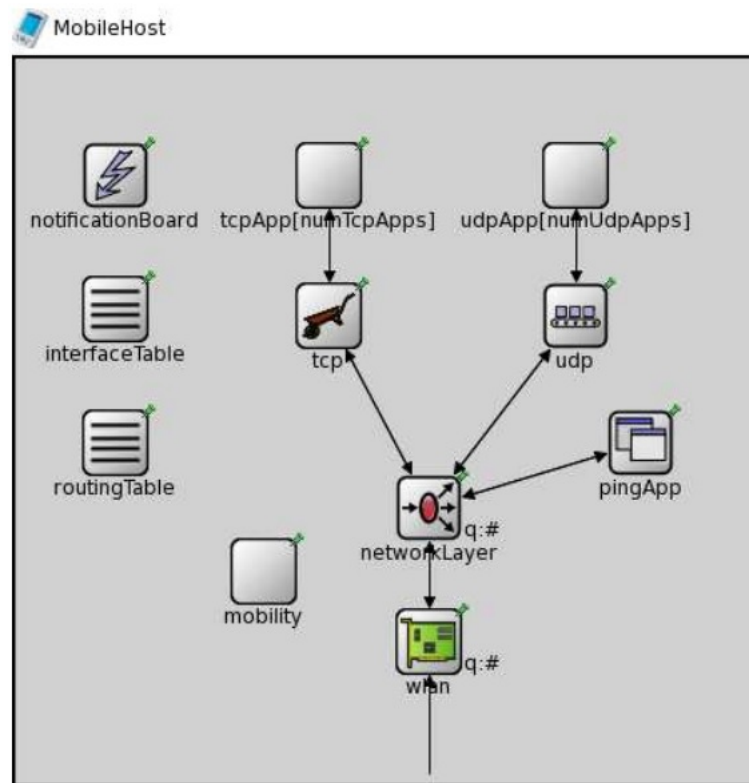
Além do *INET*, as simulações para redes específicas utilizam outros *framework*, no caso deste projeto será utilizado o *framework* para redes LoRa que será apresentado logo adiante.

2.2.3 FloRa

Existe um *framework* específico para simulações *LoRa* dentro do *OMNet*. Esse *framework* inclui uma biblioteca com código aberto chamada *FloRa*, publicada em 2017, desenvolvida por pesquisadores na Universidade de Aalto na Finlândia onde existem vários projetos de pesquisa no tema *LoRa* e *LoRaWAN*. Este *framework* permite a implantação de vários *nós* e vários *gateways* na rede, além de permitir configurar diversos fatores como a posição dos dispositivos em coordenadas x e y, coeficientes de perda de propagação, ativar ou desativar a tecnologia *ADR* nos dois nós do *LNS*.(YASCARIBAY et al., 2022)

Como dito anteriormente, a simulação feita com o *FloRa* permite configurar todos os parâmetros de transmissão da camada física *LoRa*: Fator de espalhamento,

Figura 8 – Estrutura INET



Fonte: (KIM et al., 2010)

frequência central, largura de banda, taxa de código e potência de transmissão. (SLABICKI GOPIKA PREMSANKAR, 2018) O *framework* também permite coletar métricas para analisar os problemas da rede sendo simulada, com isso é possível resgatar os resultados mais facilmente.

O foco deste projeto é verificar a escalabilidade da rede com maior número de *nós* além dos existentes na rede física, mobilidade dos *nós* para análise da cobertura e erros de rede, além de testar o funcionamento do *ADR* que é um mecanismo automático de adaptação de dados dos dispositivos, que também é possível de ser implementado na simulação. Todos esses dados o *framework FloRa* permite configurar de acordo com o desejo do usuário. Erros de propagação, colisão de pacotes, consumo de energia e taxa de entrega de pacotes também são outras métricas possíveis de análise da rede LoRa no simulador. (YASCARIBAY et al., 2022) No *FloRa* há disponíveis métricas para a simulação:

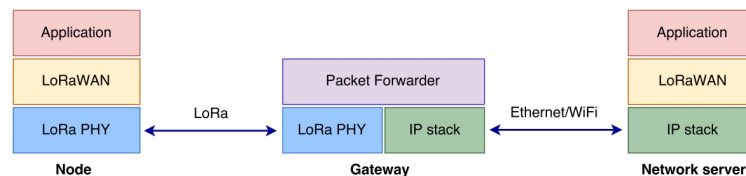
- **Consumo de energia** : Consumo total de energia dos *nós*, normalmente informada em *Joules(J)*.
- **Número de nós**: Numero de dispositivos finais *nós* é determinado pelo projeto, há possibilidade de aumentar o numero de *nós* para mais de 3000, porém cada simulação pode demorar mais de 2h (YASCARIBAY et al., 2022)

- **Numero de gateways:** A configuração de varios *gateways* é possível, pois o *LNS* é configurado para filtrar os pacotes duplicados e tratá-los para que seja enviado somente um ao *nó* com melhor qualidade de sinal.(SLABICKI GOPIKA PREMSANKAR, 2018)
- **ADR(Adaptative Data Rate):** É possível ativar ou desativar o ADR tanto nos dispositivos quanto nos *LNSs* para testes.
- **Perdas de propagação:** Nesse quesito o objetivo é ter um sistema sem perdas, apesar das diversas configurações de simulação.
- **Mobilidade dos nós:** Simular um sistema onde os *nós* não são fixos e podem se movimentar livremente no ambiente.

O *FloRa* suporta ambientes urbanos e suburbanos, assim como os dados de perda de caminho da transmissão de cada ambiente simulado. Seguindo experimentos feitos em (BOR et al., 2016) os casos de simulações feitas com ambientes urbanos têm maior perda de caminho do que em suburbanos, portanto tem menor área de cobertura. A transmissão eficiente do *LoRa* depende se as transmissões se interferem entre si. Assume-se que duas transmissões com fator de espalhamento diferentes não se colidem devido à seus canais ortogonais.

De acordo com a Figura 9 , o *FloRa* implementa módulos *OMNet* que replicam a camada física *LoRa* e a camada de enlace(*LoraWAN*). As transmissões vindas dos *nós* de múltiplos canais podem ser recebidas pelos *gateways* simultaneamente de acordo com as especificações do *LoRaWAN* na simulação.(SEMTECH, 2019). A camada física pode ser implementada utilizando módulos *INET* como os enlaces de *Ethernet* e *Wifi*.

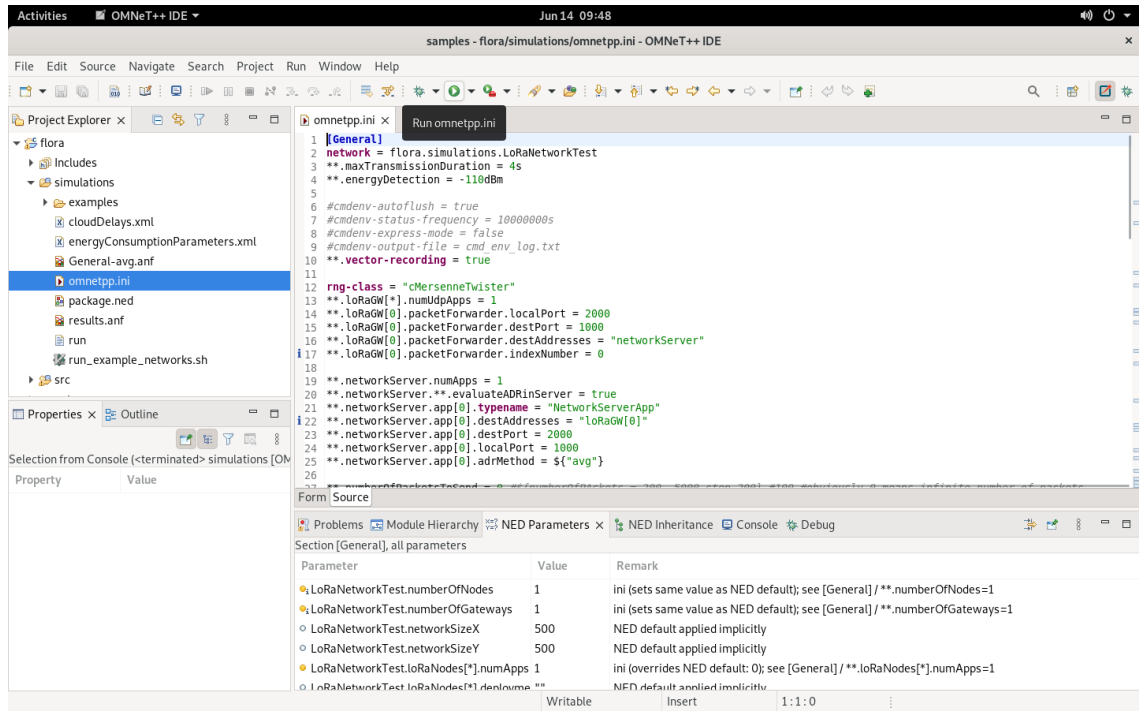
Figura 9 – Estrutura de módulos *FloRa*



Fonte: (SLABICKI GOPIKA PREMSANKAR, 2018)

A implementação do *FloRa* nos proporciona modelos de *perda de caminho*, atualmente existem dois já implementados, os modelos *Hata Okumura* e o *Log-normal shadowing*. Tudo isso está no github do *FloRa*, além disso em sua página de tutoriais possui algumas informações da implementação. Conforme Figura 10 há várias configurações semelhantes às do *OMNet* que permitem definir os fatores de simulação *LoRa* e *LoRaWAN*(FLORA-DOCUMENTATION, 2022).

Figura 10 – Fatores de simulação no OMNet++



Fonte: (FLORA-DOCUMENTATION, 2022)

Por fim o *FloRa* nos permite construir uma simulação fiel à uma rede *LoRa*, configurando os principais dados do sistema, para uma melhor análise de dados de desempenho de rede sobre o projeto definido. Isso implica que podemos construir uma simulação com muitos fatores aleatórios como mudanças de ambiente e contemplar uma solução viável para algum projeto de aplicação na área.

3 PROPOSTA

Este capítulo apresenta a proposta de uma simulação a partir de uma rede LoRa implantada no campus IFSC SJ, tendo como meta levantar subsídios para a operação da rede quando submetida a condições que normalmente não poderiam ser investigadas dado a quantidade de nodos envolvidos e condições de mobilidade.

Será feito inicialmente um conjunto de experimentos para comparar os resultados do cenário real com o cenário simulado. Na sequência, serão realizadas modificações na topologia da rede, incluindo um aumento de nós na rede, mobilidade de nodos, aumento de tráfego, distanciamento do *gateway* para testes de área de cobertura e mudança de ambiente com pouca área de visada. Mudanças nas métricas estabelecidas anteriormente serão realizadas. Assim possibilitando uma melhor visualização de uma possível solução para um grande projeto envolvendo as tecnologias apresentadas.

Para um melhor desenvolvimento de projeto, serão elaborados **dois** experimentos:

- Experimento de comparação com o cenário real envolvendo:
 - Coleta de dados;
 - Construção de um cenário de simulação;
 - Análise de resultados do experimento e possíveis recomendações de operação.
- Experimento de simulação de um cenário expandido, envolvendo.
 - Aumento da densidade da rede;
 - Inclusão da mobilidade de nodos da rede;
 - Inclusão de algoritmo de ADR;
 - Análises de resultados do experimento e possíveis recomendações de experimentos.

3.1 Experimento de comparação com cenário real

Neste primeiro experimento será realizada a comparação de uma simulação de rede LoRa com o cenário da rede LoRa real do IFSC. Será implementado a simulação referente ao cenário físico, utilizando as mesmas configurações e estabelecendo as mesmas métricas a serem avaliadas. Um modelo de perda no caminho será selecionado. No pacote do FloRa, existem duas implementações: *Hata Okumura* e *Log-normal shadowing*. Por fim, prover uma análise de desempenho, expondo os resultados com tabelas e determinando se são fieis ao cenário real.

3.1.1 Coleta de dados

Para a coleta de dados do experimento físico, será utilizado uma rede física implantada pelo Prof. Saul, do IFSC Campus São José, com objetivo de análise de cobertura de um *gateway*. Este dispositivo foi posicionado inicialmente na caixa d'água do campus, sendo que um nodo será posicionado em vários locais diferentes para os testes. Além disso, há a possibilidade de um complemento adicionando mais um dispositivo final, para fazer testes de mobilidade.

3.1.2 Construção de um cenário de simulação

O cenário de simulação será construído logo após a coleta de dados da rede física, utilizando as bibliotecas INET e os módulos do FloRa para seu desenvolvimento. As métricas e configurações da simulação serão iguais as do cenário físico, tendo em vista que no próximo item serão analisados os resultados das duas simulações e será feito um comparativo. Serão utilizadas as facilidades de construção de objetos que simulam edifícios e prédios. Um modelo de propagação será escolhido dentre os disponíveis para o FLoRa.

3.1.3 Análise de resultados do experimento

Após a execução da simulação será feita uma análise dos resultados comparando as métricas fornecidas dos dois cenários (real e simulado). O comparativo tem como objetivo verificar se o cenário simulado se aproxima do real, dando possibilidades de extensão do experimento, como indicado para os próximos tópicos. Possíveis recomendações de operação podem ser geradas nesta fase.

3.2 Experimento de simulação expandida

Além da construção da simulação do cenário físico da rede do IFSC, a ideia do projeto é adicionar variáveis a rede para que possamos produzir uma simulação robusta de forma que possa ser utilizada para a criação de novos projetos e soluções para a grande indústria. O grande desafio das aplicações de redes IoT é os problemas de mobilidade dos nós, o LoRa possui o ADR (Adaptative Data Rate) que permite os dispositivos finais uma autonomia de modificar e adaptar as configurações de transmissão e recepção de acordo com o ambiente. Com isso podemos construir uma simulação no FloRa levando isso em conta e apresentar resultados próximos a uma rede física.

3.2.1 Aumento da densidade da rede

A construção da simulação copiando o cenário real é só o primeiro passo. Após fazer o primeiro experimento, a ideia é fazer upgrades no cenário simulado, verificando

a escalabilidade da rede, adicionando diversos dispositivos finais e fazendo análises de desempenho desse cenário. A ideia é adicionar dezenas ou até centenas de novos dispositivos para testes.

3.2.2 Expansão da mobilidade da rede e uso do algoritmo ADR

A mobilidade de dispositivos é algo inerente as redes sem fio. Os simuladores de rede normalmente disponibilizam vários modelos de mobilidade. No Omnet, a biblioteca INET disponibiliza estes modelos. Nos *showcases* do OMNet(VARGA; LTD, 2021), são discutidos muitos modelos interessantes de mobilidade que podem ser utilizados no projeto. Os modelos de mobilidade serão adaptados ao cenário do projeto. Dado as variações das condições produzidas pode ser interessante avaliar o algoritmo ADR para melhoria do comportamento da rede.

3.2.3 Análises de resultados do experimento

As análises finais do resultado de experimento tem como objetivo demonstrar quais aplicações possíveis dos cenários simulados, bem como tabelas comparativas com múltiplos fatores de simulação e as métricas resultantes. Construir uma conclusão sobre os resultados informando quais são os melhores cenários, quais os piores e quais deles podem são viáveis para implementação real.

3.3 Cronograma

Na Tabela 1 apresenta-se o cronograma de atividades que foi planejado para o desenvolvimento do projeto.

Tabela 1 – Cronograma das atividades previstas

Etapa	Fevereiro	Março	Abril	Maiο	Junho	Julho
3.1.1	x					
3.1.2		x				
3.1.3			x			
3.2.1			x			
3.2.2				x		
3.2.3					x	
Escrita					x	x

REFERÊNCIAS

- BOR, M. C. et al. Do lora low-power wide-area networks scale? In: *Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*. New York, NY, USA: Association for Computing Machinery, 2016. (MSWiM '16), p. 59–67. ISBN 9781450345026. Disponível em: <<https://doi.org/10.1145/2988287.2989163>>. 5, 8, 10, 19
- CHAUDHARI, B. S.; ZENNARO, M. *Lpwan Technologies for IOT and M2M applications*. [S.l.]: Academic Press, 2020. 7, 9
- CHAUDHARI, B. S.; ZENNARO, M.; BORKAR, S. Lpwan technologies: Emerging application characteristics, requirements, and design considerations. *Future Internet*, v. 12, n. 3, 2020. ISSN 1999-5903. Disponível em: <<https://www.mdpi.com/1999-5903/12/3/46>>. 7
- ELDEFRAWY, M. et al. Formal security analysis of lorawan. *Computer Networks*, v. 148, p. 328–339, 2019. ISSN 1389-1286. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1389128618306145>>. 12
- FLORA-DOCUMENTATION. 2022. <<https://flora.aalto.fi/>>. Acessado em : 13-12-2022. 19, 20
- HAXHIBEQIRI, J. et al. A survey of lorawan for iot: From technology to application. *Sensors*, v. 18, n. 11, 2018. ISSN 1424-8220. Disponível em: <<https://www.mdpi.com/1424-8220/18/11/3995>>. 14, 15
- INET-DOCUMENTATION. 2022. <<https://inet.omnetpp.org/docs/>>. Acessado em : 03-12-2022. 15
- KIM, H. et al. A simulation framework for performance analysis of multi-interface and multi-channel wireless networks in inet/omnet++. In: *Proceedings of the 2010 Spring Simulation Multiconference*. San Diego, CA, USA: Society for Computer Simulation International, 2010. (SpringSim '10). ISBN 9781450300698. Disponível em: <<https://doi-org.ez130.periodicos.capes.gov.br/10.1145/1878537.1878643>>. 16, 17, 18
- RAZA, U.; KULKARNI, P.; SOORIYABANDARA, M. Low power wide area networks: An overview. *IEEE Communications Surveys Tutorials*, v. 19, n. 2, p. 855–873, 2017. 7, 8
- SEMTECH. *LoRa® and LoRaWAN®: A Technical Overview*. 1. ed. [S.l.], 2019. Disponível em: <https://lora-developers.semtech.com/uploads/documents/files/LoRa_and_LoRaWAN-A_Tech_Overview-Downloadable.pdf>. 8, 11, 13, 14, 19
- SLABICKI GOPIKA PREMSANKAR, M. D. F. M. Adaptive configuration of lora networks for dense iot deployments. 2018. Disponível em: <<https://flora.aalto.fi/resources/slabicki-2018-noms.pdf>>. 5, 11, 18, 19
- VARGA, A.; LTD, O. *Omnet++ Simulation Manual*. 1. ed. [S.l.], 2021. Disponível em: <<https://doc.omnetpp.org/omnetpp/SimulationManual.pdf>>. 16, 17, 23

YASCARIBAY, G. et al. Performance evaluation of communication systems used for internet of things in agriculture. *Agriculture*, v. 12, n. 6, 2022. ISSN 2077-0472. Disponível em: <<https://www.mdpi.com/2077-0472/12/6/786>>. 17, 18