

LoRaPlan: Ferramenta de Planejamento de Redes LoRaWAN

Rodrigo Torres¹, Edelberto Franco Silva¹

¹Laboratório NetLab
Programa de Pós-Graduação em Ciência da Computação (PPGCC)
Departamento de Ciência da Computação (DCC)
Universidade Federal de Juiz de Fora (UFJF)
Juiz de Fora/MG – Brasil

{rodrigotorres,edelberto}@ice.ufjf.br

Abstract. *The increased need for monitoring through sensors and decision-making based on large volumes of data has driven the growth of the Internet of Things (IoT) and the interconnection of devices. LoRaWAN emerges as a promising alternative due to its wide coverage and low energy consumption. However, the lack of adequate tools to plan and implement this network represents a challenge for network engineers. This work proposes a system to assist in the planning and implementation of LoRaWAN networks, addressing issues such as the appropriate quantity and location of gateways for a given scenario, aiming to optimize resources and guarantee a stable connection.*

Resumo. *O aumento da necessidade de monitoramento através de sensores e da tomada de decisões baseadas em grandes volumes de dados tem impulsionado o crescimento da Internet das Coisas (IoT) e a interconexão de dispositivos. A LoRaWAN surge como uma alternativa promissora devido à sua ampla cobertura e baixo consumo de energia. No entanto, a falta de ferramentas adequadas para planejar e implementar essa rede representa um desafio para os engenheiros de redes. Este trabalho propõe um sistema para auxiliar no planejamento e implementação de redes LoRaWAN, abordando questões como a quantidade e localização adequadas de gateways para um determinado cenário, visando otimizar recursos e garantir uma conexão estável.*

1. Introdução

Atualmente há uma crescente importância da Internet das Coisas (IoT) para monitoramento, orquestração de dispositivos e tomada de decisões, impulsionando o desenvolvimento de tecnologias como *Smart Health* [Solanas et al. 2014], *Smart Farms* [Yoon et al. 2020], *Smart Cities* [Talari et al. 2017] e Indústrias 4.0 [Pereira and Romero 2017]. Para garantir conectividade eficiente e economia de energia, destacam-se as redes sem fio de longa distância, como a LoRaWAN (*Low Power Wide Area Network*). A LoRaWAN implementa os protocolos necessários para a tecnologia LoRa, oferecendo ampla cobertura e baixo consumo de energia. O planejamento de redes LoRaWAN é desafiador devido a diversos fatores, como interferência e otimização da quantidade de *gateways*.

A falta de ferramentas adequadas para esse planejamento resulta em dificuldades para os administradores de rede, que podem adquirir mais *gateways* do que o necessário. Além disso, garantir a qualidade de comunicação exige a configuração precisa

de parâmetros da LoRaWAN. A escassez de conjuntos de dados disponíveis para estudo e otimização dessas redes também é um obstáculo. Portanto, este trabalho visa desenvolver um sistema de planejamento de redes LoRaWAN capaz de gerar *datasets* com soluções otimizadas, permitindo aos administradores ajustar os parâmetros conforme suas necessidades e facilitando novas pesquisas na área. O sistema receberá parâmetros de configuração da rede, regiões a serem atendidas e densidade de usuários, gerando como saída a localização dos nós clientes e dos *gateways* necessários para cobertura total da área.

O artigo está organizado da seguinte forma: a Seção 2 apresenta trabalhos relacionados aos desafios do posicionamento e escolha de *gateways* em uma rede LoRa. A Seção 3 introduz a proposta da ferramenta e a Seção 4 os detalhes da implementação. Por fim, as Seções 5 e 6 finalizam o trabalho e apresentam os *links* para mais informações.

2. Trabalhos Relacionados

Este capítulo revisa alguns estudos para determinar o método de otimização a ser implementado na ferramenta proposta por esse trabalho.

O estudo de [Karthikeya et al. 2016] concentra-se na redução do número total de *gateways* em uma *Smart City*, visando diminuir os custos de instalação através do algoritmo “*Network Intersection based Candidate Gateway Location Selection*”. Tem como principais objetivos o cálculo dos locais candidatos para instalação de *gateways* incorporando algoritmos de k-cobertura e k-conectividade.

[Matni et al. 2019] propõe o algoritmo Fuzzy C-Means para otimizar o posicionamento dos *gateways* LoRa, visando melhorar a Qualidade de Serviço (QoS) e reduzir os custos de capital (CAPEX) e operacionais (OPEX).

[Ali et al. 2022] introduz uma infraestrutura IoT para Assistência Médica 4.0, propondo três algoritmos evolutivos baseados em *swarm intelligence*: *Discrete Fireworks Algorithm (DFWA)*, *Discrete Artificial Bee Colony (DABC)* e *Low-complexity bio-geography-based Optimization (LC-BBO)* onde o objetivo é o planejamento da capacidade e cobertura da rede, minimização do custo de *hardware* e infra-estrutura e também a diminuição computacional e operacional.

[Ferreira Jr et al. 2022] propõe um método para planejamento e otimização da implantação de *gateways* IoT em *Smart Campus*, utilizando quatro tipos de algoritmos: otimização de programação linear, K-means, PSO-Simples e PSO-Híbrido. O objetivo é propor um método para planejamento e otimização da implantação de *gateways* IoT tendo em vista o uso do *gateway* SOFTWAY4IoT que implementa um software de rede virtualizado capaz de integrar a computação em nuvem com a computação em névoa.

Ao final, o trabalho conclui que todos os modelos propostos podem ser usados para o planejamento de uma rede sem fio, desde que o cenário de implementação seja apropriado ao algoritmo. Portanto, para o cenário proposto, o algoritmo K-means foi escolhido para ser implementado.

3. LoRaPlan

Primeiramente, para a criação da aplicação proposta, foi necessário definir o comportamento esperado em termos de entrada, processamento e saída, além de como seria sua

usabilidade e confiabilidade. Portanto os seguintes requisitos funcionais da aplicação foram definidos:

- RF01: O sistema deve exibir um mapa interativo que permita o usuário localizar qualquer região do mundo;
- RF02: O usuário deve conseguir desenhar no mapa todas as regiões que serão atendidas pela solução;
- RF03: O usuário deve conseguir definir os pesos que cada área terá na geração de clientes;
- RF04: O sistema deve permitir que o usuário insira os seguintes parâmetros: seed, quantidade de clientes, potência de transmissão, ganho da antena, frequência e SF (fator de espalhamento de sinal - *Spreading Factor*);
- RF05: Ao gerar a instância o sistema deve exibir a solução encontrada no mapa;
- RF06: Um arquivo .JSON deve ser disponibilizado para download com todas as informações da instância gerada.

Na Figura 1 é possível observar, de forma simplificada, a arquitetura geral do sistema que implementa as RFs apresentadas. LoRaPlan foi focado em uma arquitetura REST (*Representational State Transfer*) onde o Cliente (*frontend*) consome a API (*backend*) por meio da comunicação de dados em JSON (*JavaScript Object Notation*).

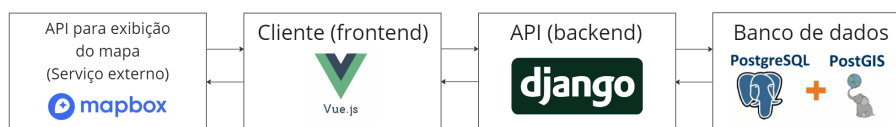


Figura 1. Arquitetura da Solução em Alto Nível

Já na Figura 2, apresenta as principais telas do sistema. Sendo a primeira um exemplo de entrada das configurações da instância e a segunda a saída do sistema.

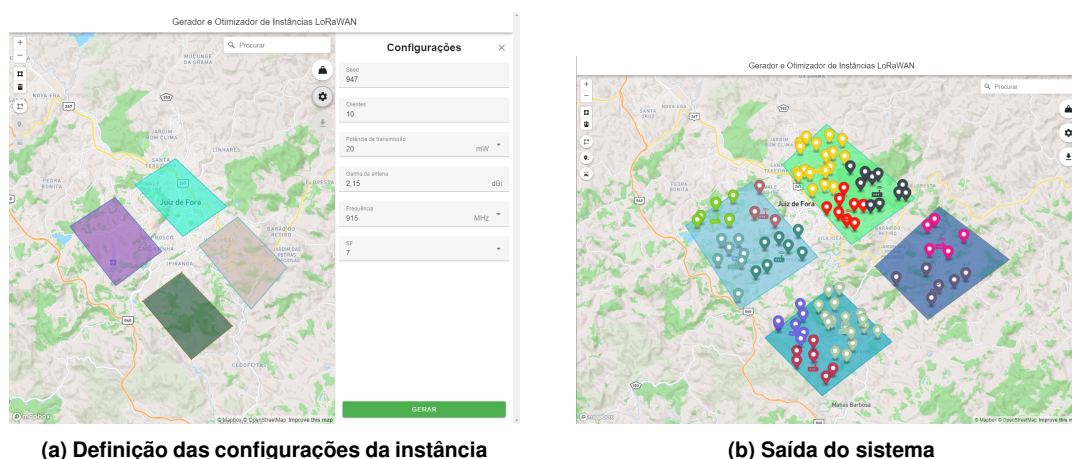


Figura 2. Telas

4. Implementação

Para atender aos requisitos funcionais descritos, a implementação do sistema pode ser dividida em três etapas que serão detalhadas ao longo deste documento:

1. Validação dos *gateways*
2. Geração dos clientes
3. Geração dos *gateways*

4.1. Validação dos *gateways*

Antes de gerar um *gateway* é preciso definir que tipo de validação será feita para dizer se os clientes serão atendidos. No sistema duas verificações foram feitas: se a distância física entre o *gateway* e o cliente é válida e se o *gateway* irá sofrer uma interferência (SNR) que irá inibir a comunicação com o *gateway*.

Esse trabalho utilizou os valores aproximados do trabalho [Islam et al. 2020] e retirou a Tabela 1 para a validação da distância.

Tabela 1. Limites de distância de acordo com o SF e a Potência de Transmissão (mW)

SF/ Potência de Transmissão (mW)	5	20	50	100
7	2000 metros	2500 metros	3200 metros	3800 metros
8	2400 metros	3000 metros	3600 metros	4400 metros
9	2600 metros	3500 metros	4200 metros	5000 metros
10	3000 metros	4000 metros	5000 metros	6000 metros
11	3500 metros	4800 metros	6000 metros	7400 metros
12	4000 metros	5800 metros	7300 metros	9000 metros

Já para validar a interferência foi considerado o cenário onde todos os clientes que estão ligados ao *gateway* estão tentando comunicar ao mesmo tempo. Nesse cenário, para calcular o SNR do *gateway* é necessário primeiro calcular o somatório de todas as potências de envio de dados que os clientes exercem para se comunicar. A potência de cada cliente pode ser calculada pela Fórmula 1, baseada na equação de Friis em espaço livre [Friis 1946, Aref and Sikora 2014], onde PC é a potência recebida do cliente, P a potência (mW), GA o ganho da antena (dBi), D (metros) a distância física entre cliente e *gateway* e F a frequência (MHz). Cada um desses parâmetros são informados pelo usuário no *frontend*.

$$PC = P \times GA^2 \times \left(\frac{300}{4 \times \pi \times D \times F} \right)^2 \quad (1)$$

Por fim, o SNR do *gateway* para um cliente é calculado utilizando a Fórmula 2, baseada no teorema de Shannon-Hartley [Shannon 1948], onde P é a potência informada pelo usuário e TPC é o somatório de todas as potências recebidas dos outros clientes que estão comunicando com o *gateway*.

$$SNR = 20 \times \log_{10} \left(\frac{P}{TPC} \right) \quad (2)$$

Com o SNR calculado a Tabela 2 é consultada para identificar se o valor do SNR irá impossibilitar a comunicação. A Tabela foi retirada do trabalho [Jouhari et al. 2023] e leva em consideração o SF para indicar o limite.

Tabela 2. Limites de SNR por SF. Fonte: [Jouhari et al. 2023]

SF	7	8	9	10	11	12
Limite de SNR	-7,5	-10	-12,5	-15	-17,5	-20

4.2. Geração dos clientes

Para a geração dos clientes os seguintes *inputs* devem ser informados pelo usuário, seguindo as regras listadas na Tabela 3.

Tabela 3. Inputs e suas validações

Parâmetro	Tipo	Validação
Coordenadas das regiões	Conjunto de pontos que formam uma região	Os pontos devem formar um polígono válido
Peso das regiões	Porcentagem	A soma deve ser igual a 100%
Seed	Número inteiro	Valor ≥ 1
Quantidade de clientes	Número inteiro	Valor ≥ 1
Potência de transmissão	Número inteiro em mW	Apenas os valores 5, 20, 50 e 100 são permitidos ¹
Ganho da antena	Número decimal em dBi	Valor $\geq 0,01$
Frequência	Número inteiro em MHz	Apenas os valores 433 e 915 são permitidos ²
SF	Número inteiro	Apenas os valores 7, 8, 9, 10, 11 e 12 são permitidos ³

Com todos os dados corretos o sistema armazena as coordenadas e os pesos das áreas e segue para a geração de clientes da instância. Primeiramente é criado um dicionário para auxiliar na distribuição dos clientes. O preenchimento do dicionário segue a lógica do seguinte exemplo: supomos que o usuário adicione três áreas, a primeira com 50%, a segunda com 30% e terceira com 20%, totalizando os 100%. O Algoritmo 1 irá criar um dicionário onde as chaves de 0 a 49 terão como valor o identificador da primeira área, as chaves 50 a 79 serão vinculadas ao identificador da segunda área e o identificador da terceira área será preenchido nas chaves 80 a 99.

Algoritmo 1: Algoritmo de criação do dicionário de pesos

Output: *Dicionário de pesos*

```
1 dicionarioDePesos  $\leftarrow$  {}
2 inicioDaFaixa  $\leftarrow$  0
3 for area  $\leftarrow$  primeiraArea to ultimaArea do
4   | idDaArea  $\leftarrow$  area.getId()
5   | pesoDaArea  $\leftarrow$  area.getPeso()
6   | for n  $\leftarrow$  inicioDaFaixa to inicioDaFaixa + pesoDaArea do
7   |   | dicionario[n]  $\leftarrow$  idDaArea
8   |   | n  $\leftarrow$  n + 1
9   | end for
10  | inicioDaFaixa  $\leftarrow$  inicioDaFaixa + pesoDaArea
11 end for
12 return dicionarioDePesos;
```

¹As potências foram restringidas a apenas quatro valores, pois esse trabalho leva em consideração o estudo dos limites distância em [Islam et al. 2020]

²Conforme [Committee 2020], apenas os dois valores de frequência citados estão disponíveis no Brasil.

³De acordo com [Figueiredo and Silva 2020], o SF pode variar dentre os seis disponíveis.

A partir do dicionário a quantidade de clientes é distribuída utilizando uma função de geração de números pseudo-aleatórios que recebe a *seed* informada pelo usuário no *frontend*. Assim um novo dicionário é gerado onde cada chave representa o identificador da área e o valor representa a quantidade de clientes.

Ao final, o PostGIS é utilizado novamente para gerar as coordenadas dos clientes. Utilizando a função **ST_GeneratePoints**¹, ao informar as coordenadas da área, a quantidade de clientes e a *seed* nos parâmetros da função, a extensão gera pontos pseudo-aleatórios que estão contidos na área informada. Os dados dos clientes são armazenados bem como sua relação com a área a qual pertence. O Algoritmo 2 mostra a lógica utilizada nessa etapa.

Algoritmo 2: Algoritmo de criação dos clientes

Output: *Clientes e suas coordenadas*

```
1 dictClientesPorArea ← {}
2 n ← 0
3 while n < quantidadeClientes do
4   idArea ← dicionarioDePesos[gerarNumeroRandomico0a99()]
5   dictClientesPorArea[idDaArea] ←
     dictClientesPorArea[idArea] + 1
6   n ← n + 1
7 end while
8 area ← dictClientesPorArea.proximaArea()
9 while area ≠ null do
10  areaCoords ← area.getCoordenadas()
11  quantClientes ← area.getQuantidadeClientes()
12  clientes ← ST_GeneratePoints(areaCoords, quantClientes, seed)
13  salvaClientes(clientes)
14  area ← dictClientesPorArea.proximaArea()
15 end while
```

4.3. Geração dos *gateways*

Para a geração da localização dos *gateways* da instância o Algoritmo 3 foi criado e tem como principal recurso o uso do K-means para a otimização da quantidade de *gateways*.

Na primeira execução apenas um *cluster* é gerado, ou seja, todos os clientes são alocados em apenas um *gateway* que está sempre localizado no centro do *cluster*. Caso esse *gateway* seja inválido uma nova chamada ao algoritmo K-means é feita, mas agora gerando dois *clusters*. Nessa segunda iteração cada *gateway* terá seu próprio conjunto de clientes e, caso algum deles não seja válido, uma nova chamada do K-means é feita com apenas os clientes daquele determinado *gateway* e uma nova divisão de dois *clusters* é feita e assim sucessivamente até que todos os *gateways* gerados sejam válidos.

¹https://postgis.net/docs/ST_GeneratePoints.html

Algoritmo 3: Algoritmo de criação dos gateways

Output: Gateways e suas coordenadas

```
1 gatewaysValidos ← []
2 criarGateways(clientes, 1)
3 Function criarGateways (clientes, quantidadeClusters) :
4   clusters ← kmeans(clientes, quantidadeClusters)
5   for cluster ← clusters[n] to n = clusters.size() do
6     if gatewayValido(cluster) then
7       | gatewaysValidos ← cluster
8     else
9       | criarGateways(cluster.getClientes(), 2)
10    end if
11  end for
```

5. Conclusões e trabalhos futuros

Este trabalho apresentou um sistema para auxiliar os administradores de rede LoRaWAN na concepção e redução de custos, além de fornecer conjuntos de dados para futuras pesquisas em algoritmos de otimização de redes LPWANs. A dificuldade na definição de áreas válidas no mapa foi superada com a API do Mapbox e a extensão PostGIS. A pesquisa intensiva sobre a validação de gateways resultou na estratégia de colocá-los no centro de um cluster, otimizando sua quantidade. Como trabalhos futuros podem ser considerados novos parâmetros para a economia de energia e a interferência urbana no cálculo do SNR, uma vez que o modelo de propagação atualmente utilizado é o de Friss em espaço livre. Assim, espera-se avaliar outros modelos de propagação, como Okumura-Hata, além de outros algoritmos de otimização para posicionamento e escolha de *gateways*.

6. Materiais complementares

- Ferramenta: O sistema está disponível online e pode ser testado em <https://loraplan.vercel.app>
- Código: O código fonte pode ser encontrado em <https://github.com/Rodrigo947/gerador-otimizador-instancias-lorawan>
- Documentação: A documentação pode ser encontrada tanto no README quanto na seção *Wiki* do repositório <https://github.com/Rodrigo947/gerador-otimizador-instancias-lorawan/wiki>
- Vídeo de instalação: <https://youtu.be/gbmJZJV7sD0>
- Vídeo da explicação das funcionalidades: <https://youtu.be/ka7KGQt5l0Y>

Para a demonstração das funcionalidades abordadas no vídeo, é necessário apenas uma máquina com internet para acessar o site <https://loraplan.vercel.app>.

Referências

- Ali, H. M., Liu, J., Bukhari, S. A. C., and Rauf, H. T. (2022). Planning a secure and reliable iot-enabled fog-assisted computing infrastructure for healthcare. *Cluster Computing*, 25(3):2143–2161.

- Aref, M. and Sikora, A. (2014). Free space range measurements with semtech lora™ technology. In *2014 2nd international symposium on wireless systems within the conferences on intelligent data acquisition and advanced computing systems*, pages 19–23. IEEE.
- Committee, L. A. T. (2020). Rp2-1.0.1 lorawan® regional parameters. <https://resources.lora-alliance.org/technical-specifications/rp2-1-0-1-lorawan-regional-parameters>. Acessado: 20/06/2023.
- Ferreira Jr, D., Oliveira, J. L., Santos, C., Filho, T., Ribeiro, M., Freitas, L. A., Moreira, W., and Oliveira-Jr, A. (2022). Planning and optimization of software-defined and virtualized iot gateway deployment for smart campuses. *Sensors*, 22(13):4710.
- Figueiredo, L. M. and Silva, E. F. (2020). Cognitive-lora: adaptation-aware of the physical layer in lora-based networks. In *2020 IEEE Symposium on Computers and Communications (ISCC)*, pages 1–6. IEEE.
- Friis, H. T. (1946). A note on a simple transmission formula. *Proceedings of the IRE*, 34(5):254–256.
- Islam, N., Ray, B., and Pasandideh, F. (2020). Iot based smart farming: Are the lpwan technologies suitable for remote communication? In *2020 IEEE International Conference on Smart Internet of Things (SmartIoT)*, pages 270–276. IEEE.
- Jouhari, M., Saeed, N., Alouini, M.-S., and Amhoud, E. M. (2023). A survey on scalable lorawan for massive iot: Recent advances, potentials, and challenges. *IEEE Communications Surveys & Tutorials*.
- Karthikeya, S. A., Vijeth, J., and Murthy, C. S. R. (2016). Leveraging solution-specific gateways for cost-effective and fault-tolerant iot networking. In *2016 IEEE Wireless Communications and Networking Conference*, pages 1–6. IEEE.
- Matni, N., Moraes, J., Rosário, D., Cerqueira, E., and Neto, A. (2019). Optimal gateway placement based on fuzzy c-means for low power wide area networks. In *2019 IEEE Latin-American Conference on Communications (LATINCOM)*, pages 1–6. IEEE.
- Pereira, A. C. and Romero, F. (2017). A review of the meanings and the implications of the industry 4.0 concept. *Procedia Manufacturing*, 13:1206–1214.
- Shannon, C. E. (1948). A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423.
- Solanas, A., Patsakis, C., Conti, M., Vlachos, I. S., Ramos, V., Falcone, F., Postolache, O., Pérez-Martínez, P. A., Di Pietro, R., Perrea, D. N., et al. (2014). Smart health: A context-aware health paradigm within smart cities. *IEEE Communications Magazine*, 52(8):74–81.
- Talari, S., Shafie-Khah, M., Siano, P., Loia, V., Tommasetti, A., and Catalão, J. P. (2017). A review of smart cities based on the internet of things concept. *Energies*, 10(4):421.
- Yoon, C., Lim, D., and Park, C. (2020). Factors affecting adoption of smart farms: The case of korea. *Computers in Human Behavior*, 108:106309.