

Otimizando Energia no Aprendizado Federado em Redes de Baixa potência e com Alta Taxa de Perda de Pacotes

Johann J. Schmitz Bastos¹, João Pedro C. Batista¹, Ramon dos Reis Fontes²
Eduardo Cerqueira³, Rodolfo S. Villaca¹, Vinícius F. S. Mota¹

¹Departamento de Informática (DI) – Univ. Federal do Espírito Santo (Ufes)

{johann.bastos, joao.c.batista}@edu.ufes.br
{rodolfo.villaca, vinicius.mota}@inf.ufes.br

²Instituto Metr pole Digital (IMD) – Univ. Federal do Rio Grande do Norte (UFRN)

ramon.fontes@imd.ufrn.br

³Instituto de Tecnologia (Itec) – Univ. Federal do Par  (UFPA)

cerqueira@ufpa.br

Abstract. *Federated Learning (FL) enables the distributed training of machine learning models without the need to share raw data. In IoT networks characterized by resource-constrained devices and high packet loss, selecting devices for training presents significant challenges. This work proposes optimizing the selection process by focusing on the energy efficiency of devices utilizing the RPL protocol, a protocol that builds a tree like topology of the network. The proposed approach was evaluated through an extension of the MininetFed platform, incorporating support for RPL networks, energy consumption monitoring, and a selection algorithm designed to extend the lifetime of IoT networks during training without compromising effectiveness. The proposed approach achieved a reduction of circa 6,5% in the energy consumption of central nodes.*

Resumo. *O Aprendizado Federado (FL) permite o treinamento distribuído de modelos de aprendizado de máquina, sem necessidade de compartilhamento de dados brutos. Em redes IoT com dispositivos de recursos limitados e alta perda de pacotes, a seleção de dispositivos para treinamento é desafiadora. Este trabalho prop e otimizar essa sele  o focando na efici ncia energ tica de dispositivos que utilizam o protocolo RPL (Routing Protocol for Low-Power and Lossy Networks), um protocolo que constr i uma topologia de  rvore da rede. A proposta foi avaliada em uma extens o da plataforma MininetFed, que adicionou suporte a redes RPL, monitoramento de consumo de energia e um algoritmo de sele  o para prolongar a vida  til das redes IoT durante o treinamento sem comprometer a efic cia. A abordagem proposta atingiu uma redu  o de aproximadamente 6,5% no consumo de n s centrais.*

1. Introdu  o

A evolu  o das redes baseadas em dispositivos da Internet das Coisas (IoT) trouxe desafios significativos em termos de seguran a, privacidade, comunica  o e efici ncia energ tica, especialmente em aplica  es que envolvem dispositivos IoT com recursos limitados, como esperado em *smartgrids*, monitoramento ambiental e cidades inteligentes.

Neste contexto, o Aprendizado Federado (FL) [Kairouz et al. 2021, Li et al. 2020] surge como uma solução promissora de aprendizado de máquina distribuído, permitindo o treinamento colaborativo de modelos de aprendizado sem que os dados brutos sejam compartilhados entre dispositivos. Essa abordagem se alinha à crescente demanda por proteção de dados sensíveis e conformidade com regulamentações de privacidade, bem como com a forma distribuída apresentada em ambientes IoT. Em vez de enviar dados brutos para a nuvem, o FL permite que os dispositivos locais (como sensores IoT e smartphones) processem os dados de forma independente e compartilhem apenas os parâmetros ou gradientes do modelo treinado. Este processo descentralizado é coordenado por um servidor central, que agrega as contribuições locais para construir um modelo global mais robusto.

Apesar de mitigar problemas relacionados à privacidade, em cenários de alta perda de pacotes e cuja disponibilidade de energia é limitada, um mecanismo de seleção de clientes ineficiente pode consumir toda energia disponível em dispositivos IoT e comprometer o encaminhamento de dados. Em especial, estes cenários podem exigir uma topologia multi-saltos entre os dispositivos que formam a rede, e protocolos que devem coordenar o roteamento entre si. Entre os diversos protocolos propostos, o RPL (*Routing Protocol for Low-Power and Lossy Networks*) constrói uma topologia de árvore, chamada DoDAG (*Destination-Oriented Directed Acyclic Graph*) [Winter et al. 2012]. O RPL define um dispositivo como um nó raiz (*gateway*) e permite o roteamento de dados entre os dispositivos, dos dispositivos para a raiz e da raiz para um ou mais dispositivos.

Protocolos como o RPL são amplamente adotados para gerenciar o roteamento em redes IoT devido à sua eficiência em cenários de baixa potência e alta perda de pacotes. A integração de FL com o RPL apresenta um novo paradigma, unindo a inteligência distribuída com a capacidade de roteamento otimizado para aplicações críticas, como monitoramento ambiental, detecção de desastres e redes industriais. No entanto, o tempo de vida da rede depende de um consumo eficiente de energia nos nós que fazem parte do roteamento multi-salto. Por exemplo, em caso de desligamento do nó raiz da árvore RPL, a rede se tornaria desconexa. Neste sentido, há um *trade-off* entre eficiência energética e escolha dos dispositivos que participarão do processo de treinamento em cada rodada para garantir um maior tempo de vida da rede e sem comprometer o processo de federação e a qualidade dos modelos de aprendizado treinados nesse processo.

Este artigo propõe uma abordagem para a seleção de clientes em redes RPL que participarão do FL, considerando eficiência energética em redes IoT e desempenho do FL de forma integrada. Inicialmente, estendemos a plataforma Mininet-Fed [Bastos et al. 2024], um ambiente de experimentação e análise de algoritmos de aprendizado federado, para suportar topologias de rede com o protocolo RPL para roteamento entre os dispositivos. Nesta extensão, também foi adicionado o monitoramento de uso de processamento e rede para cálculo de consumo de energia. Visando aumentar o tempo de vida de redes RPL, propusemos um algoritmo de seleção de clientes que leva em consideração o consumo de energia dos dispositivos, sendo os dispositivos com menor consumo de energia mais aptos a participarem de uma etapa (*round*) do aprendizado. Por fim, demonstramos a eficiência e o desempenho do algoritmo proposto por meio da avaliação de diferentes técnicas de seleção de clientes em uma topologia típica em cenários de redes IoT. As contribuições deste trabalho podem ser sumarizadas como:

- Estender o MininetFed para permitir a criação e configuração de experimentos de

FL por meio da emulação de dispositivos IoT, considerando restrições energéticas e redes baseadas no protocolo RPL;

- Apresentar um modelo para emulação e monitoramento de consumo de energia em dispositivos IoT no MininetFed, baseados na utilização de CPU (processamento) e rede (em Bytes) em cada dispositivo;
- Propor um algoritmo de seleção de clientes no FL baseada no consumo de energia de cada participante, com foco no aumento do tempo de vida da rede IoT e mantendo a qualidade (acurácia) do modelo treinado.

O restante deste trabalho está organizado como se segue. A Seção 2 apresenta os fundamentos teóricos e os trabalhos relacionados. As extensões realizadas no MininetFed, bem como o algoritmo proposto, são apresentados na Seção 3. Os experimentos e resultados são discutidos na Seção 4. Por fim, a Seção 5 conclui este trabalho.

2. Trabalhos Relacionados

Para organizar melhor os trabalhos relacionados, destacamos os principais estudos que exploraram o uso do aprendizado federado e redes RPL em ambientes IoT. Em seguida, apresentamos trabalhos relevantes que abordam o fator energético como um elemento central na seleção de clientes, destacando suas contribuições e abordagens específicas.

2.1. Routing Protocol for Low-Power and Lossy Networks (RPL)

O Federated Learning (FL) aplicado redes baseadas no protocolo RPL abre possibilidades para aprimorar a eficiência energética e a resiliência em redes IoT com restrições severas. Sua aplicação pode, por exemplo, permitir roteamento dinâmico considerando padrões de tráfego e disponibilidade de energia, potencialmente tornando a rede mais autônoma e robusta frente a diferentes cenários.

O uso do Aprendizado Federado (FL) em redes RPL IoT não é algo novo, mas ainda se encontra em um estágio inicial de exploração. Por exemplo, [Sharma et al. 2023] propôs um modelo FL otimizado para proteger RPL, denominado OFL-SRPL, em dispositivos de infraestrutura de medição avançada (AMI). Esse modelo visa reduzir o tamanho dos dados transferidos dos modelos locais para o servidor global. Para isso, o OFL-SRPL adotou o algoritmo Particle Swarm Optimization (PSO) para selecionar o modelo local mais adequado no gateway Home Area Network (HAN) e, a partir disso, construir um modelo global com maior eficiência. [Tariq et al. 2024] propôs o Aprendizado Federado como mecanismo de detecção de ataques seletivos de encaminhamento em RPL. A abordagem utiliza um modelo leve, incorporando o IoT Routing Attack Dataset (IRAD), visando melhorar a capacidade de detecção de ataques de roteamento.

[Albishari et al. 2024] introduziu o Fed-RPL, um modelo baseado em redes recorrentes com portas (GRU) que utiliza aprendizado federado para protocolos de roteamento. Esse trabalho destacou o uso de rodadas de treinamento descentralizadas e a aplicação de um método de quantização (Q-8bit) para reduzir o número de atualizações de peso, mitigando a sobrecarga de comunicação e mantendo uma elevada precisão do modelo.

Esses estudos demonstram que, embora promissora, a exploração do FL em protocolos de roteamento para IoT ainda é relativamente tímida e concentra-se, na maioria, nos aspectos de segurança e privacidade. Esses fatores, de fato, representam os principais

motivadores para a adoção do FL em redes IoT. No entanto, existe uma lacuna evidente em relação a outros aspectos críticos, como a eficiência energética e a manutenção do funcionamento regular da rede RPL-IoT.

2.2. Restrição Energética

A seleção criteriosa de clientes para participação nas rodadas de treinamento em ambientes de IoT tem sido explorada para equilibrar eficiência energética e desempenho no aprendizado federado. O FL pode priorizar o consumo de energia em redes de sensores, onde cada nó processa dados localmente e transmite apenas modelos treinados.

Uma das propostas foi formulada por [Fu et al. 2023], que desenvolveram um problema de otimização estocástica para a seleção conjunta de clientes e alocação de largura de banda. Esse trabalho levou em consideração as restrições energéticas de longo prazo dos clientes e introduziu um algoritmo baseado em informações de canal sem fio disponíveis em tempo real, garantindo desempenho eficiente ao longo do tempo.

Em [AbdulRahman et al. 2020], foi apresentado o FedMCCS, uma abordagem multicritério que avalia CPU, memória, energia e tempo. Essa abordagem permite prever se os clientes possuem capacidade suficiente para executar tarefas de aprendizado federado, promovendo uma alocação de clientes mais eficiente. Para otimizar o *tradeoff* entre o número de clientes selecionados e o consumo total de energia, [Yu et al. 2021] propuseram um modelo que ajusta dinamicamente a seleção de clientes e a alocação de recursos como ajuste da frequência da CPU e controle da potência de transmissão.

Além disso, [Zhao et al. 2022] introduziram o FedNorm, uma estrutura que identifica clientes capazes de fornecer informações significativas em cada rodada de treinamento. Diferentemente de abordagens anteriores, o FedNorm inclui uma variante energeticamente eficiente, que realiza a seleção de clientes apenas em rodadas específicas, prometendo reduzir ainda mais o consumo energético da rede.

Por fim, até onde conhecemos, este artigo é pioneiro na aplicação de FL em redes IoT com o protocolo RPL, com foco na seleção de clientes baseada na redução do consumo de energia nos nós centrais da rede RPL formada durante o treinamento.

3. Uma arquitetura para aprendizado federado sobre redes RPL

Este trabalho estende o *MiniNetFed*¹ para suportar o uso de interfaces de redes IEEE 802.15.4 e, desta forma, prover uma arquitetura que utilize o protocolo de redes RPL. Devido à natureza do protocolo RPL, é proposto um algoritmo que visa balancear o consumo de energia e a participação dos nós da rede no processo de treinamento.

3.1. Visão Geral

O *MiniNetFed* [Sarmiento et al. 2024] é uma ferramenta que estende o Mininet-WiFi [Fontes et al. 2015] e Containernet [Peuster et al. 2016] e emula ambientes de aprendizado federado (FL) realistas, permitindo a criação de cenários com dispositivos heterogêneos e configuráveis em termos de CPU, memória e conectividade de rede. O emulador permite, dentre outras coisas, que os usuários definam: i) modo de divisão do conjunto de dados entre os dispositivos; ii) políticas de seleção de clientes; iii) função de agregação

¹<https://github.com/lprm-ufes/MininetFed>

de modelos; iv) utilizar modelos para os principais *datasets* disponíveis; e v) visualizações gráficas das principais métricas de desempenho. No *MiniNetFed*, os dispositivos se comunicam por meio da emulação de interfaces WiFi.

Inicialmente, foram adicionados módulos do *kernel Linux* para emular redes IEEE 802.15.4, usadas em dispositivos IoT de baixo custo energético. A Figura 1 mostra a arquitetura e os componentes do MininetFed, que interagem com o *kernel Linux* e as ferramentas relacionadas ao IEEE 802.15.4. O MininetFed emula uma WPAN com interfaces IEEE 802.15.4, criando interfaces virtuais (wpan0) para cada nó sensor emulado (sensor1-wpan0, sensor2-wpan0, etc.) em contêineres Docker gerenciados pelo Containernet. Cada nó sensor executa algoritmos de treinamento, gerando pesos compartilhados com um *broker* MQTT para coordenação eficiente dos clientes.

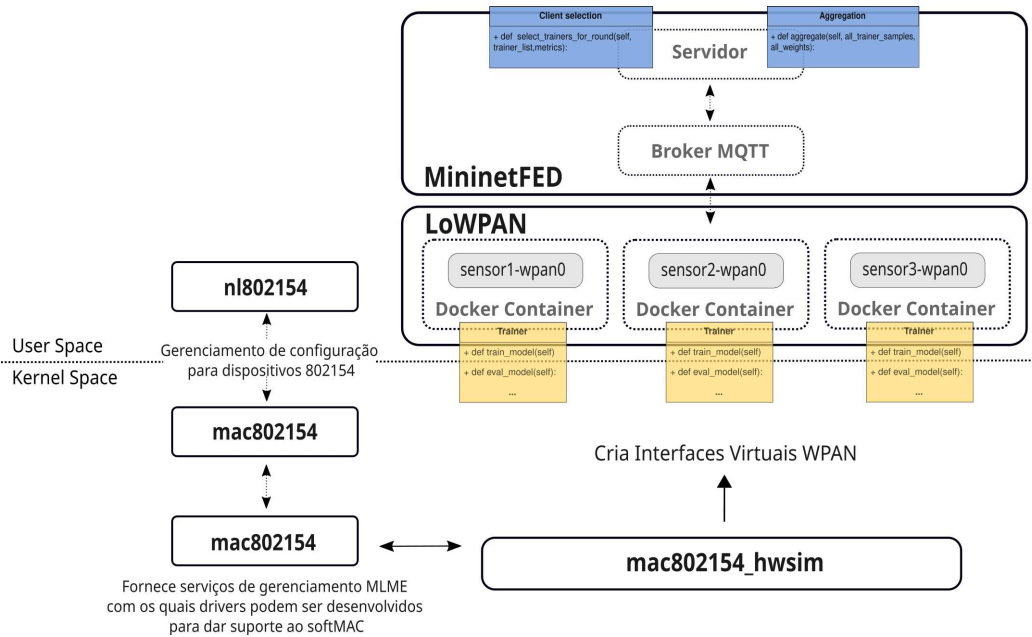


Figura 1. Componentes do MininetFed estendidos com nós de rede LowPan.

Como o padrão IEEE 802.15.4 permite comunicação um-salto ou multi-salto entre os nós que compõem a rede, faz-se necessário um protocolo de roteamento distribuído. Neste trabalho, o protocolo RPL será executado em cada nó sensor emulado. Foi desenvolvido também um módulo de monitoramento de consumo de energia para ser utilizado pelo algoritmo de seleção proposto.

3.2. Aprendizado Federado em redes RPL

O protocolo RPL é utilizado para roteamento em redes de baixo consumo de energia. Para isto, o RPL constrói um Grafo Acíclico Direcionado (*Directed Acyclic Graph* - DAG) para criar um ou mais destinos orientados (*Destination Oriented DAGs* - DODAGs). A Figura 2 ilustra uma topologia DAG típica de uma rede IoT baseada no RPL.

Neste trabalho, o nó raiz do DoDAG (*DoDAG root*) também desempenha o papel de servidor de agregação. Esse nó possui recursos de hardware adequados para tal função e não enfrenta limitações energéticas. Os outros nós da rede, organizados como nós pais e filhos, compõem a estrutura do DoDAG. No gerenciamento da seleção de clientes no

FL, os sensores pais (ou nós centrais), ilustrados na Figura 2, encaminham dados dos nós de níveis inferiores até a raiz, resultando em maior consumo de energia devido ao roteamento na rede. Por isso, eles devem ser selecionados com menor frequência para o processamento de modelos locais no aprendizado federado, a fim de preservar sua energia.

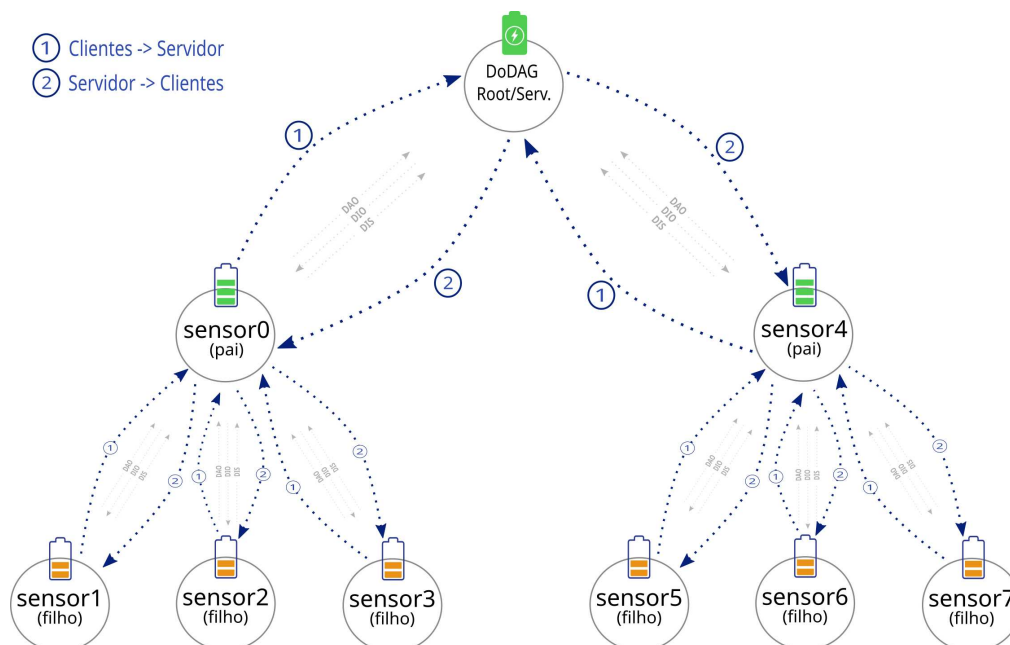


Figura 2. Topologia de Rede de um grafo acíclico direcionado DAG do RPL.

As setas menores e mais claras, na Figura 2, representam as mensagens de controle do protocolo de roteamento RPL, trocadas entre os nós em intervalos regulares de 5 segundos (ajustáveis). Essas mensagens têm a função de manter a topologia da rede, similarmente ao que ocorre em outros protocolos de roteamento dinâmico. No entanto, não serão exploradas em detalhes neste trabalho, pois não constituem o foco do estudo. Já as setas mais escuras indicam as transmissões dos modelos do processo de aprendizado federado, detalhadas a seguir:

- **Clientes -> Servidor.** No início do processo de aprendizado federado, os dispositivos sensores (clientes) treinam seus modelos utilizando seus dados locais. Durante este treinamento, os modelos locais ajustam os pesos com base nas atualizações derivadas do processo de otimização. Após o cálculo dos pesos, eles são enviados ao servidor;
- **Servidor -> Clientes.** Após receber os pesos dos clientes, o servidor realiza a agregação dos dados para atualizar o modelo global. Com o modelo global atualizado, ele envia as novas informações aos dispositivos clientes, permitindo que os modelos locais sejam ajustados com base nas melhorias globais, garantindo a sincronização do aprendizado entre todos os participantes. As atualizações recebidas são incorporadas aos modelos locais dos clientes, iniciando o próximo ciclo de aprendizado federado.

3.3. Modelos de Consumo de Energia em Dispositivos IoT

Foram desenvolvidos modelos de consumo de energia de uso de CPU e tráfego de rede para monitorar e calcular o consumo energético dos dispositivos no *MiniNetFed*.

3.3.1. Consumo Baseado em CPU

Este modelo monitora o uso da CPU em dispositivos IoT para calcular o consumo energético, considerando parâmetros como voltagem, corrente, e percentual de utilização da CPU, conforme pode ser visto na Equação 1, alinhando-se a abordagens descritas na literatura, como em [Flinn and Satyanarayanan 1999], que destacam a relação entre a carga de trabalho da CPU e o consumo de energia. Este modelo funciona monitorando o uso da CPU por meio de leituras do sistema de arquivos do dispositivo, especificamente do `/sys/fs/cgroup/cpu.stat`, para determinar o tempo de uso acumulado da CPU. A cada intervalo de 0,1 s (ajustável), calcula-se a diferença entre as leituras consecutivas, obtendo o Δ de uso da CPU. Com isso, é possível calcular o percentual de utilização da CPU no intervalo, usado para determinar o consumo energético (C_{CPU}).

$$C_{CPU} = \frac{\text{Voltagem (V)} \times \text{Corrente (A)} \times \text{Uso da CPU (\%)}}{3600} \quad (1)$$

A energia consumida é então calculada multiplicando a voltagem do processador, a corrente consumida, e o percentual de utilização da CPU. Este valor é convertido de Watts para Watt-hora (Wh) considerando o intervalo de tempo monitorado. Além disso, o modelo registra informações detalhadas, como *timestamp*, percentual de uso da CPU e energia consumida, permitindo uma análise histórica do comportamento energético do dispositivo. Este modelo é particularmente útil para aplicações onde o consumo da CPU é predominante, como em cenários de processamento local de dados.

3.3.2. Consumo Baseado em Tráfego de Rede

Este modelo calcula o consumo energético (C_{NET}) com base na quantidade de dados transmitidos e recebidos pelas interfaces de comunicação do dispositivo, conforme a Equação 2, adaptada de [Polastre et al. 2005, Ye et al. 2002]. A implementação do modelo monitora os Bytes transmitidos e recebidos utilizando informações do arquivo `/proc/net/dev`. Diferenças entre leituras consecutivas dos contadores de Bytes são usadas para calcular a quantidade de dados transferidos no intervalo de tempo. A partir disso, o modelo utiliza custos fixos de energia por Byte transmitido e recebido para determinar o consumo energético em Joules. Esse valor é então convertido para Watt-hora (Wh), facilitando comparações com outros dispositivos ou padrões de consumo.

$$C_{NET} = (\text{Bytes RX} \times \text{Custo RX} + \text{Bytes TX} \times \text{Custo TX}) \times \frac{1}{3600} \quad (2)$$

A integração desses dois modelos proporciona uma visão abrangente do consumo energético em dispositivos IoT. O modelo baseado no uso da CPU permite otimizar o processamento local, enquanto o modelo baseado na transmissão e recepção de Bytes ajuda a gerenciar o consumo associado à comunicação. Essa abordagem é especialmente valiosa em redes IoT que utilizam 6LoWPAN, permitindo identificar oportunidades para otimização energética, como a redução de transmissões desnecessárias ou o gerenciamento

inteligente da carga de trabalho nos dispositivos. Tais otimizações são críticas em cenários onde o consumo eficiente de energia é essencial para prolongar a vida útil dos dispositivos e garantir a continuidade das operações.

3.4. Seleção de nós do processo de treinamento

Durante o processo de treinamento, deve-se reduzir o consumo energético dos nós pais, mantendo a rede conexa. Isso é obtido minimizando o número de vezes em que esses nós são utilizados para treinamento dos modelos ao longo das rodadas de treinamento. Caso sejam selecionados indiscriminadamente para participar das rodadas de treinamento, esse consumo pode comprometer a sustentabilidade da rede por aumentar ainda mais o consumo baseado no processamento dos modelos. Por isso, torna-se essencial a aplicação de uma estratégia criteriosa para a seleção dos clientes envolvidos no treinamento federado, visando preservar a eficiência energética e garantir a longevidade da infraestrutura.

Surge, portanto, um *tradeoff*: enquanto a seleção de nós centrais consome mais energia, excluí-los totalmente pode prejudicar o desempenho do modelo global, pois a diversidade dos dados locais nesses nós é essencial para alcançar melhor generalização e acurácia. É crucial equilibrar eficiência energética e desempenho do modelo; selecionar os nós centrais com muita frequência pode esgotar sua energia, comprometendo o funcionamento da rede, enquanto nunca selecioná-los pode comprometer a qualidade do modelo de aprendizado federado devido à insuficiência de diversidade nos dados.

Neste sentido, é proposto o Algoritmo 1 para fazer seleção criteriosa de clientes com base no seu consumo de energia. Esse algoritmo é executado no nó servidor. Como entrada ele recebe: uma lista de todos os treinadores/sensores conectados no momento da seleção; e uma tabela associativa que identifica as métricas individuais de cada sensor. Cada sensor é responsável por enviar suas próprias métricas antes da execução do algoritmo, as quais são dispostas nessa tabela.

O algoritmo seleciona os clientes que consumiram menos energia do que a média. Entre as linhas 2 a 5 é calculada uma média simples da energia consumida por cada sensor até o momento atual do treinamento. Então, nas linhas de 6 a 10, são selecionados aqueles sensores com o consumo abaixo dessa média, colocando esses em uma lista. Por fim, na

Algoritmo 1 Seleção de treinadores com menor consumo de energia

Require: *trainer_list*: Lista de treinadores, *metrics*: Métricas associadas aos treinadores

Ensure: Lista de treinadores com consumo de energia menor ou igual à média

```

1: mean_energy_consumption  $\leftarrow$  0
2: for all trainer  $\in$  trainer_list do
3:   mean_energy_consumption  $\leftarrow$  mean_energy_consumption +
     metrics[trainer]["energy_consumption"]
4: end for
5: mean_energy_consumption  $\leftarrow$  mean_energy_consumption / |trainer_list|
6: selected_trainers  $\leftarrow$  []
7: for all trainer  $\in$  trainer_list do
8:   if metrics[trainer]["energy_consumption"]  $\leq$  mean_energy_consumption then
9:     selected_trainers.append(trainer)
10:  end if
11: end for
12: return selected_trainers

```

linha 12, essa lista é retornada.

4. Estudos de Casos

A seleção de clientes para a participação no treinamento federado deve considerar não apenas a disponibilidade de dados e poder computacional, mas também o consumo de energia dos dispositivos para garantir a longevidade da rede. Serão apresentados abaixo três estudos de casos para diferentes critérios de seleção de clientes: (i) o primeiro, aqui nominado como “*all*” diz respeito à escolha indiscriminada de todos os sensores; (ii)np segundo é realizada uma escolha aleatória, nominado como “*random*” e, no último (iii) utilizamos um algoritmo que prioriza sensores com menor consumo de energia, nominado como “*energy*”. Em todos os casos é utilizada a topologia ilustrada na Figura 2. Os artefatos necessários para reprodutibilidade dos estudos de casos estão disponíveis no repositório MininetFed-LoWPAN no Github².

A seguir, os estudos de caso são apresentados e detalhados, com os resultados de suas avaliações. O treinamento foi realizado utilizando a rede LeNet5 [Lecun et al. 1998] com o *dataset* MINST³ e 10 épocas. As rodadas foram limitadas em 20, que se mostraram suficientes para que o experimento atingisse um platô no aumento da acurácia dos modelos. O objetivo dos estudos de caso é mostrar os efeitos no consumo energético causado por diferentes etapas do treinamento em sensores selecionados e não selecionados, e o resultado dos modelos treinados, no qual não se observa redução de desempenho (acurácia).

O que difere cada um dos casos é o método de seleção de cliente. No primeiro caso, todos os sensores são selecionados, no segundo 5 sensores são selecionados aleatoriamente, e no último é proposto um algoritmo de seleção que considera o consumo de energia na seleção a fim de atingir um consumo mais homogêneo de energia e reduzir o impacto que o consumo elevado dos nós pais pode causar no consumo de bateria, sem prejudicar o desempenho do modelo global.

4.1. Seleção de Todos os Clientes (*all*)

A Figura 3 apresenta a variação do consumo de energia (Wh) ao longo do tempo no primeiro estudo de caso, quando todos os sensores foram selecionados em todas as rodadas. Todos os gráficos estão na mesma escala. Existem dois padrões distintos que podem ser observados: uma elevação com alguns espaços não ocupados, similar a uma onda quadrada nesse experimento; e picos que ocorrem ao fim de cada uma dessas elevações, os quais são bem evidentes no `sensor0` e no `sensor4`. O primeiro padrão observado é referente ao consumo de energia da CPU de um cliente selecionado para o treino, e o segundo padrão é referente ao consumo de energia da transmissão após cada rodada de treinamento.

Ambos os padrões podem ser observados de maneira repetitiva no gráfico de cada sensor. Essa seleção implica que todos os sensores tiveram uma carga de trabalho similar em termos de CPU, mas pode-se ver que o consumo energético dos sensores 0 e 4 tiveram mais picos devido ao fato que são sensores centrais que tem que lidar com a transmissão dos pesos de todos: deles próprios e dos sensores filhos.

²<https://github.com/lprm-ufes/MininetFed-LoWPAN>

³<https://yann.lecun.com/exdb/mnist/>

4.2. Seleção Aleatória (*random*)

A Figura 4 apresenta gráficos de consumo de energia (Wh) ao longo do tempo, quando os sensores foram selecionados de forma aleatória nas rodadas de treinamento. Embora o consumo de energia mostre padrões semelhantes aos observados no experimento anterior, há momentos em que alguns sensores apresentam consumo de energia muito baixo, quebrando esse padrão. Isso sugere que, em certas rodadas, nem todos os sensores estão ativos ou envolvidos no processamento de tarefas, o que é consistente com a distribuição desigual de carga entre os dispositivos proposta pelo algoritmo de seleção, causando assim períodos de inatividade seletiva.

4.3. Seleção Considerando o Consumo de Energia (*energy*)

A Figura 5 apresenta os gráficos de consumo de energia (Wh) ao longo do tempo, quando os sensores foram selecionados utilizando o algoritmo apresentado na Seção 3.4. O objetivo é reduzir o consumo de energia dos nós durante as rodadas de treinamento, especialmente os nós pais da topologia. Observa-se que os sensores apresentam comportamentos distintos de consumo energético. Os sensores *sensor0* e *sensor4* (pais) não apresentam os picos de consumo característicos da seleção para o treinamento em uma rodada, apenas os picos característicos da transmissão. Os demais sensores exibem padrões consistentes com a seleção para o treinamento.

A diferença de comportamento entre os sensores sugere que o mecanismo proposto neste artigo está evitando a seleção de alguns sensores como parte de uma abordagem de balanceamento de carga ou economia de energia, onde apenas um subconjunto

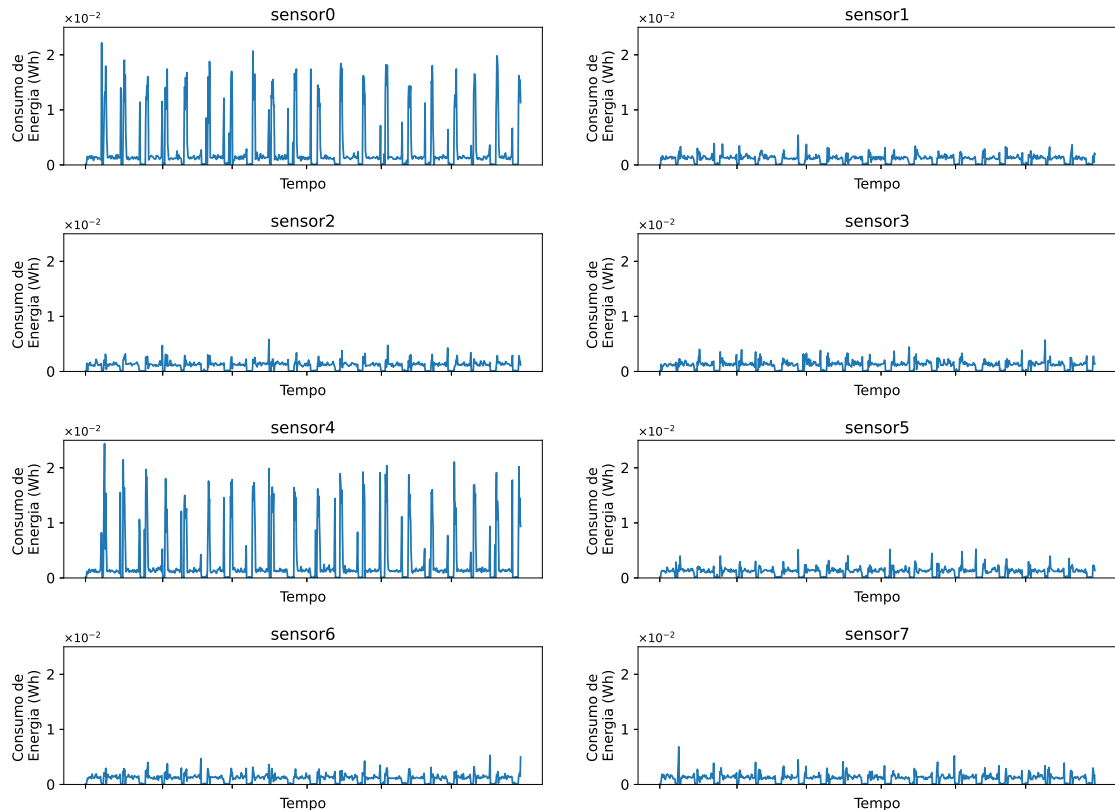


Figura 3. Seleção de Todos os Clientes - Estratégia (*all*).

de sensores é utilizado em cada rodada. Isso confirma a eficácia da estratégia de seleção para reduzir o consumo energético em sensores centrais que naturalmente consomem mais energia devido ao tráfego de dados. É importante lembrar aqui que os sensores 0 e 4 atuam como nós centrais na arquitetura, o que justifica o seu maior consumo. O algoritmo de seleção baseado em consumo de energia foi capaz de identificar esse comportamento e, como estratégia de preservação energética, reduziu a frequência de participação desses nós nas rodadas de treinamento, contribuindo para um consumo um pouco mais equilibrado e sustentável ao longo do experimento.

A redução da frequência de participação dos nós centrais resultou em um menor consumo de energia, conforme ilustrado na Figura 6. Observa-se que os sensores 0 e 4 apresentaram um consumo energético inferior em comparação aos demais nós da rede, com reduções de aproximadamente 6,7% e 6,3%, respectivamente. Esse comportamento pode indicar uma maior preservação da conectividade entre os nós da rede, especialmente em cenários de operação prolongada, onde a eficiência energética é essencial para garantir a continuidade do funcionamento da infraestrutura. Embora este trabalho não tenha definido um nível específico de bateria para os sensores, caso essa informação estivesse disponível, poderia ser utilizada para que sensores pais priorizassem sua permanência ativa na rede, garantindo a conectividade contínua de todos os sensores filhos a eles vinculados.

4.4. Impacto no Desempenho do Treinamento

A Figura 7 apresenta uma comparação detalhada da evolução da acurácia ao longo de múltiplas rodadas de treinamento, considerando três abordagens distintas de seleção de clien-

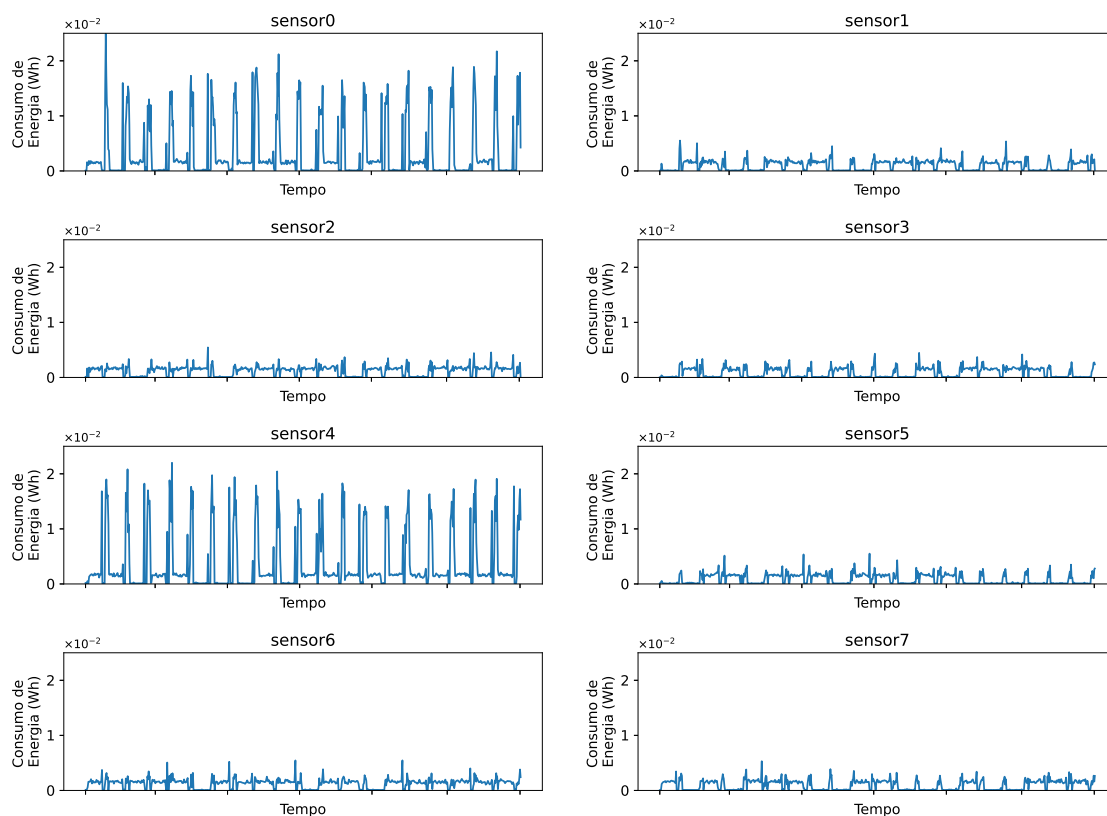


Figura 4. Seleção Aleatória de Clientes - Estratégia (*random*).

tes: (i) todos os clientes selecionados, (ii) clientes selecionados aleatoriamente e (iii) clientes selecionados com base no critério de otimização energética. Observa-se que, nos experimentos realizados, a acurácia converge de forma muito semelhante nos cenários onde os dados estão balanceados (a) ou levemente desbalanceados (b), indicando que a estratégia de seleção de clientes focada no consumo de energia não comprometeu o desempenho do treinamento. Essa estabilidade sugere que é possível alcançar uma economia de energia significativa sem prejuízos relevantes à qualidade do modelo treinado, o que é especialmente vantajoso em redes IoT com restrições energéticas.

Entretanto, quando os dados estão completamente desbalanceados entre os nós da rede RPL, há um impacto significativo no desempenho do modelo global, conforme observado na Figura 7(c). Este resultado é um indicador para um trabalho futuro de uma estratégia híbrida que considere a seleção com base na minimização do consumo energético dos nós centrais e na maximização da acurácia do modelo global.

5. Conclusão

Neste trabalho, propusemos uma abordagem para a seleção de clientes em Aprendizado Federado em redes RPL, visando eficiência energética em cenários IoT de baixa potência e alta perda de pacotes. A extensão da plataforma *MiniNetFed* permitiu a integração do protocolo RPL, o monitoramento do consumo energético e a implementação de um algoritmo de seleção de clientes baseado nesse consumo. Os experimentos mostraram que a estratégia proposta não apenas prolonga a vida útil da rede, mas também mantém a qualidade do modelo treinado, mesmo em cenários de alta perda de pacotes. Essa abordagem

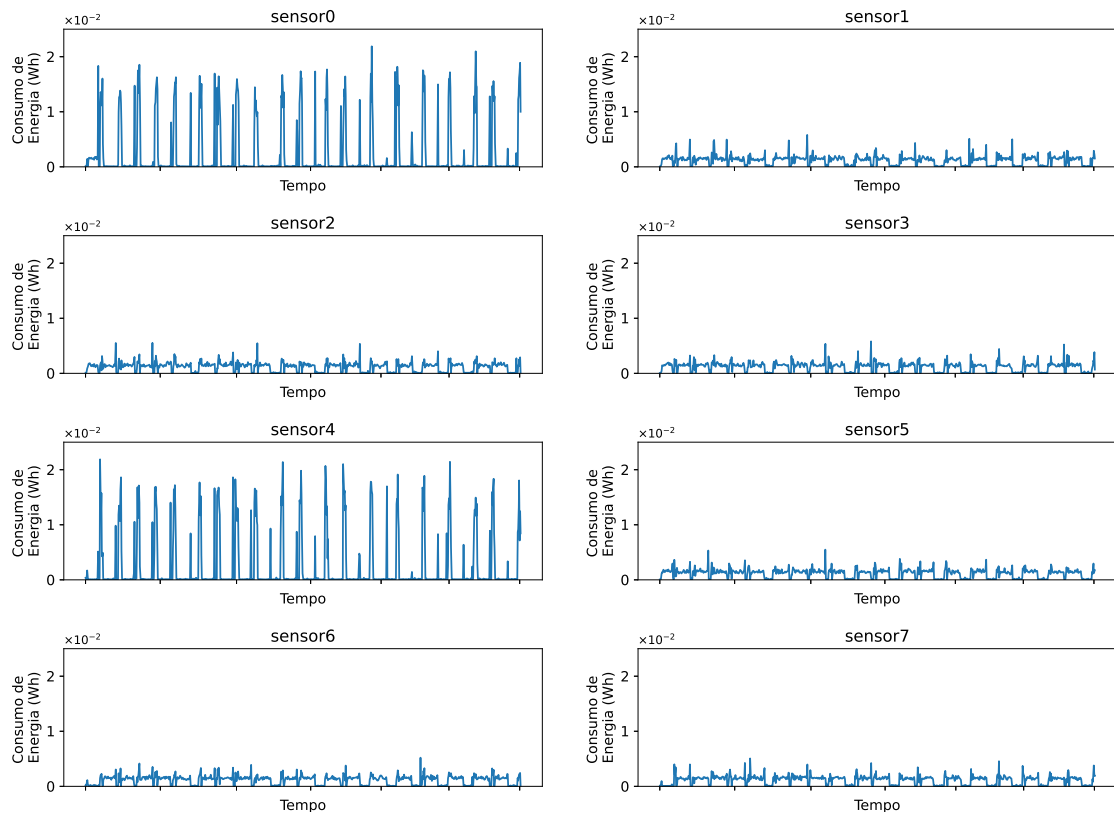


Figura 5. Seleção com otimização energética.

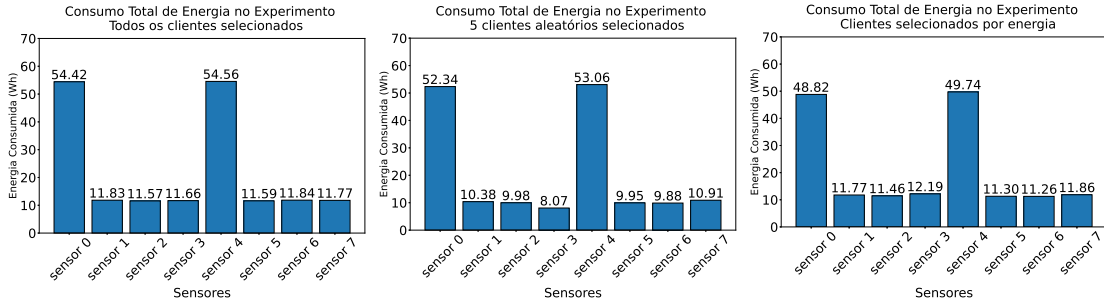


Figura 6. Consumo de energia acumulado.

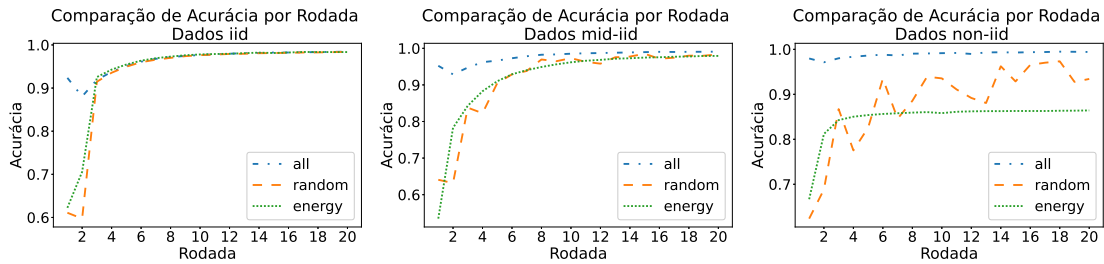


Figura 7. Acurácia do FL com dados balanceados (*iid*), levemente desbalanceados (*mid-iid*) e completamente desbalanceados (*non-iid*).

resultou em uma economia de energia superior a 6% para os nós centrais, contribuindo para a preservação da conectividade entre os nós da rede.

A partir da abordagem apresentada neste trabalho, vários caminhos podem ser explorados para avançar na área. Em primeiro lugar, pretendemos desenvolver e avaliar novos algoritmos de seleção de clientes, que integrem métricas adicionais, como qualidade do link entre os nós e recursos de CPU e memória disponíveis. Isso permitirá que a seleção de clientes leve em consideração não só a eficiência energética, mas também o potencial computacional de cada nó na rede, resultando em uma utilização mais equilibrada e eficiente dos recursos da rede. Além disso, pretende-se explorar uma abordagem de seleção de clientes que busque otimizar o desempenho do modelo global em cenários desafiadores, quando os dados estão desbalanceados dentre os clientes do FL.

Agradecimentos

Este trabalho possui financiamento parcial de: CNPq, CAPES (Finance Code #001 e 88887.005666/2024-00), Fapes (#2023/RWXSZ, #2022/ZQX6, #2022/NGKM5, #2021/GL60J) e Fapesp/ MCTI/ CGI.br (#2020/ 05182-3 e #2023/00148-0).

Referências

- AbdulRahman, S., Tout, H., Mourad, A., and Talhi, C. (2020). Fedmccs: Multicriteria client selection model for optimal iot federated learning. *IEEE Internet of Things Journal*, 8(6):4723–4735.
- Albishari et al. (2024). Federated deep learning models for detecting rpl attacks on large-scale hybrid iot networks. *Computer Networks*, 254:110837.
- Bastos, J., Sarmiento, E., Villaça, R., and Mota, V. (2024). Mininetfed: Uma ferramenta para emulação e análise de aprendizado federado com dispositivos heterogêneos. In *Anais Estendidos do XLII SBRC*, pages 57–64, Porto Alegre, RS, Brasil. SBC.

- Flinn, J. and Satyanarayanan, M. (1999). Energy-aware adaptation for mobile applications. *ACM SIGOPS Operating Systems Review*, 33(5):48–63.
- Fontes, R. R., Afzal, S., Brito, S. H., Santos, M. A., and Rothenberg, C. E. (2015). Mininet-wifi: Emulating software-defined wireless networks. In *11th CNSM*. IEEE.
- Fu, L., Zhang, H., Gao, G., Zhang, M., and Liu, X. (2023). Client selection in federated learning: Principles, challenges, and opportunities. *IEEE Internet of Things Journal*.
- Kairouz et al. (2021). Advances and open problems in federated learning. *Foundations and trends® in machine learning*, 14(1–2):1–210.
- Lecun, Y. et al. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.
- Li, T., Sahu, A. K., Talwalkar, A., and Smith, V. (2020). Federated learning: Challenges, methods, and future directions. *IEEE signal processing magazine*, 37(3):50–60.
- Peuster, M., Karl, H., and van Rossem, S. (2016). Medicine: Rapid prototyping of production-ready network services in multi-pop environments. In *IEEE Conference on Network Function Virtualization and Software Defined Networks*, pages 148–153.
- Polastre, J., Szewczyk, R., and Culler, D. (2005). Telos: Enabling ultra-low power wireless research. In *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005.*, pages 364–369. IEEE.
- Sarmiento et al. (2024). Mininetfed: A tool for assessing client selection, aggregation, and security in federated learning. In *IEEE 10th World Forum on Internet of Things (WF-IoT)*, pages 1–6. IEEE.
- Sharma, P., Sakthivel, T., and Alnajjar, I. A. (2023). Optimized federated learning with ensemble of sequential models for detecting rpl routing attacks for ami networks.
- Tariq, N., Khan, R., Almufareh, M. F., Humayun, M., and Shaheen, M. (2024). A federated learning mechanism for mitigating selective forwarding attacks in rpl-based internet of things. In *IEEE ICC Workshops*, pages 871–877. IEEE.
- Winter, T., Thubert, P., Brandt, A., Hui, J., and Group, R. W. (2012). RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. *RFC 6550, IETF*.
- Ye, W., Heidemann, J., and Estrin, D. (2002). An energy-efficient mac protocol for wireless sensor networks. In *Proceedings. Twenty-first annual joint conference of the IEEE computer and communications societies*, volume 3, pages 1567–1576. IEEE.
- Yu, L., Albelaihi, R., Sun, X., Ansari, N., and Devetsikiotis, M. (2021). Jointly optimizing client selection and resource management in wireless federated learning for internet of things. *IEEE Internet of Things Journal*, 9(6):4385–4395.
- Zhao, J., Feng, Y., Chang, X., and Liu, C. H. (2022). Energy-efficient client selection in federated learning with heterogeneous data on edge. *Peer-to-Peer Networking and Applications*, 15(2):1139–1151.