

Uma Arquitetura de Co-orquestração SDN e Aprendizado Federado para Seleção Adaptativa de Clientes

João Victor de Oliveira¹, Cândido de Queiroga¹, Fernando Matos¹

¹Universidade Federal da Paraíba (UFPB) – João Pessoa – PB – Brasil

{joao.monteiro, clqb}@academico.ufpb.br

fernando@ci.ufpb.br

Abstract. *Although promising, the performance of Federated Learning (FL) is directly impacted by the underlying network infrastructure. In congestion scenarios, the lack of proactive traffic management and objective client selection criteria can increase convergence time and compromise training quality. To address this, this work proposes and evaluates an architecture that integrates Software-Defined Networking (SDN) into the FL process, establishing a co-orchestration cycle in which an SDN orchestrator continuously monitors the network state, applies proactive flow rerouting, and exposes real-time metrics for adaptive client selection—based on local model quality and network conditions. The evaluation was conducted on an emulated hierarchical topology with 14 OpenFlow switches, training XGBoost models on the Epsilon, HIGGS, and MNIST datasets. The architecture reduced round execution time by approximately 45–58% (depending on the dataset) without statistically significant degradation in quality metrics—demonstrating that co-orchestration between network control and client selection creates a virtuous cycle capable of accelerating FL without compromising its predictive effectiveness.*

Resumo. *Embora promissor, o desempenho do Aprendizado Federado (FL) é diretamente impactado pela infraestrutura de rede subjacente. Em cenários de congestionamento, a ausência de gerenciamento proativo de tráfego e de critérios objetivos de seleção de clientes pode elevar o tempo de convergência e comprometer a qualidade do treinamento. Assim, este trabalho propõe e avalia uma arquitetura que integra Redes Definidas por Software (SDN) ao processo de FL, estabelecendo um ciclo de co-orquestração, onde um orquestrador SDN monitora continuamente o estado da rede, aplica reencaminhamento proativo de fluxos e expõe métricas em tempo real para a seleção adaptativa de clientes — baseada em qualidade do modelo local e condições de rede. A avaliação foi conduzida em topologia hierárquica emulada com 14 switches OpenFlow, treinando modelos XGBoost sobre os datasets Epsilon, HIGGS e MNIST. A arquitetura reduziu o tempo de execução das rodadas em aproximadamente 45–58% (variando conforme o dataset) sem degradação estatisticamente significativa nas métricas de qualidade — evidenciando que a co-orquestração entre controle de rede e seleção de clientes cria um ciclo virtuoso capaz de acelerar o FL sem comprometer sua eficácia preditiva.*

Palavras-chave: Redes Definidas por Software, Aprendizado Federado, Co-orquestração Inteligente

1. Introdução

A convergência entre infraestruturas de rede e paradigmas de aprendizado de máquina intensificou-se com a proliferação de dispositivos de borda e a crescente necessidade de processar dados de forma distribuída e privada [Aledhari et al. 2020]. O Aprendizado Federado (FL) emergiu como paradigma promissor ao permitir o treinamento colaborativo de modelos sem que os dados brutos trafeguem pela rede [McMahan et al. 2017]. Contudo, a heterogeneidade inerente às redes que conectam clientes ao servidor central representa um desafio técnico relevante: segmentos podem sofrer congestionamento por tráfego coexistente, resultando em atrasos assimétricos nas atualizações do modelo e comprometendo o tempo de convergência e a qualidade do treinamento [Mahmod et al. 2025, Konečný et al. 2016]. Dado que a comunicação em FL é significativamente mais custosa que a computação local, a otimização do canal de transporte torna-se determinante.

A literatura tem endereçado a eficiência do FL por vias predominantemente locais: otimização computacional nos clientes [Abdulrahman et al. 2021], seleção de participantes por critérios de hardware [Nishio and Yonetani 2019] e compressão de parâmetros transmitidos [Shahid et al. 2021, Libardi et al. 2025]. Essas abordagens melhoram o FL *apesar* da rede, não *através* dela. Conforme apontado por Wu et al. [Wu et al. 2024], a consciência da topologia de rede como alavanca de melhoria do FL permanece subexplorada, sobretudo em cenários que não assumem a topologia em estrela estática do FedAvg, onde a estrutura do caminho entre clientes e servidor influencia diretamente o tempo de cada rodada. As Redes Definidas por Software (SDN), ao proverem visibilidade global da topologia e reprogramação dinâmica dos planos de encaminhamento, oferecem instrumentos naturalmente adequados para preencher essa lacuna [Xie et al. 2018, Ma et al. 2022].

Este trabalho insere-se na corrente *network for AI* [Mahmod et al. 2025], que propõe adaptar ativamente a infraestrutura de rede às demandas das aplicações de aprendizado, e contribui com a proposta, implementação e avaliação experimental de uma arquitetura que organiza o sistema em três camadas funcionais: uma **camada de dados**, responsável pelo transporte do tráfego de modelos; uma **camada de controle**, programado por um controlador OpenDaylight; e uma **camada de inteligência**, onde um orquestrador SDN e o servidor FL estabelecem um ciclo de co-orquestração. O orquestrador monitora continuamente a camada de dados, expõe métricas de qualidade de caminho em tempo real e adapta proativamente o encaminhamento; o servidor FL consome essas métricas para selecionar adaptativamente os clientes mais aptos a cada rodada, reduzindo a pressão sobre a camada de dados e acelerando a convergência do modelo global.

Para avaliar a contribuição da solução, foram conduzidos três cenários experimentais sobre os conjuntos de dados Epsilon, HIGGS e MNIST utilizando o algoritmo XGBoost: a solução proposta, com roteamento sensível à carga combinado à seleção adaptativa de clientes; um baseline sem SDN com agregação convencional; e um cenário intermediário com roteamento SDN mas sem seleção adaptativa. Os resultados mostram que a solução proposta reduz o tempo de execução das rodadas em 45–58% em relação ao baseline, sem degradação estatisticamente significativa nas métricas de qualidade do modelo. Notavelmente, o cenário com roteamento isolado mostrou-se 9–13% *mais lento* que o baseline. Isto evidencia que o reencaminhamento proativo, quando desacompanhado de seleção adaptativa, perturba conexões TCP ativas e piora o desempenho, o que reforça

o papel central da seleção no ciclo de co-orquestração proposto. Além disso, a validação sobre dados tabulares estende os resultados a um domínio de grande importância prática e historicamente sub-representado na literatura de FL [Nilsson et al. 2018]: detecção de fraudes, diagnóstico clínico e monitoramento industrial são aplicações predominantemente tabulares e que operam sob restrições reais de interpretabilidade e recursos computacionais. O XGBoost é o algoritmo de referência nesses contextos [Chen and Guestrin 2016], sendo amplamente adotado na indústria precisamente por sua alta acurácia, custo computacional inferior ao de redes neurais profundas e capacidade de operar em dispositivos de borda com memória e processamento limitados.

O restante deste artigo está organizado da seguinte forma: a Seção 2 revisa os trabalhos relacionados e posiciona a contribuição frente ao estado da arte; a Seção 3 detalha a arquitetura de co-orquestração SDN-FL e os mecanismos de seleção adaptativa; a Seção 4 descreve a metodologia experimental e o cenário de emulação de rede; a Seção 5 apresenta e discute os resultados obtidos; por fim, a Seção 6 encerra o trabalho com as conclusões e sugestões para pesquisas futuras.

2. Trabalhos Relacionados

O uso de SDN para mitigar gargalos de rede no FL é uma vertente consolidada. Mahmood, Pace e Iera [Mahmood et al. 2025] propuseram um framework baseado em SDN para reduzir atrasos decorrentes de congestionamento, utilizando o emulador GNS3 para demonstrar reduções de até 55% no tempo de treinamento, evoluindo posteriormente para a estratégia DCSS (Delay/Computational-resources client Selection Strategy), que combina índices de recursos computacionais e atraso de rede para a seleção de clientes. Contudo, a proposta não considera a contribuição histórica ao modelo global nem o impacto do reencaminhamento proativo sobre o transporte TCP.

Na vertente de otimização de tráfego, o framework Smart-FLow [Hamdan et al. 2025] utiliza a visão global do SDN para paralelizar o tráfego dinamicamente, superando o protocolo tradicional de caminho mínimo e reduzindo o tempo de sincronização em até 47%. Entretanto, a solução não incorpora critérios de qualidade do modelo ou contribuição do cliente no processo de decisão. Já Trindade et al. [Trindade et al. 2022] investigaram a gestão de recursos na borda da rede, evidenciando como políticas de alocação impactam a convergência. Apesar da relevância, o trabalho não explora a programabilidade do plano de dados via SDN para reencaminhamento dinâmico.

Abordagens focadas na aplicação, como o ACSP-FL [de Souza et al. 2024] e o PEGASUS [Alves et al. 2024], propõem seleção baseada, respectivamente, no desempenho do modelo global e no orçamento de privacidade diferencial dos clientes. Contudo, ambas ignoram a infraestrutura de rede subjacente, como por exemplo, considerar o congestionamento ou a perda de pacotes. Por fim, Seo, Niyato e Elmroth [Seo et al. 2021] sugerem um mecanismo de leilão suportado por SDN para criar redes de sobreposição dinâmicas. Embora utilize SDN para gerência de recursos, o trabalho não explora o roteamento dinâmico sensível à carga como critério de seleção, dimensão central do ciclo de co-orquestração proposto neste artigo.

Shen et al. [Shen et al. 2024] propõem uma abordagem de otimização multiobjetivo que equilibra desempenho global e justiça estatística em FL, mitigando conflitos de gradiente por ajuste fino orientado ao cliente mais divergente. Dakhia e Me-

renda [Dakhia and Merenda 2025], por sua vez, modelam a seleção de clientes via teoria dos jogos, ponderando simultaneamente acurácia, consumo de energia e privacidade em dispositivos heterogêneos. Contudo, ambas as propostas tratam a infraestrutura de rede como passiva, onde congestionamento e largura de banda não integram o critério de seleção.

A contribuição deste trabalho diferencia-se por implementar uma co-orquestração total, integrando o reencaminhamento proativo SDN com uma seleção dinâmica que pondera simultaneamente a qualidade da rede e a contribuição estatística ao modelo. Além disso, a validação com XGBoost sobre dados tabulares de larga escala endereça um cenário de alta relevância industrial e de gerência de redes, frequentemente negligenciado em favor de modelos de redes neurais profundas.

3. Arquitetura do Sistema

A arquitetura proposta integra o plano de controle de uma rede definida por software ao ciclo de treinamento federado, estabelecendo um ciclo de retroalimentação entre o estado da infraestrutura e as decisões do servidor FL. Ela se organiza em três camadas funcionais, ilustrados na Figura 1.

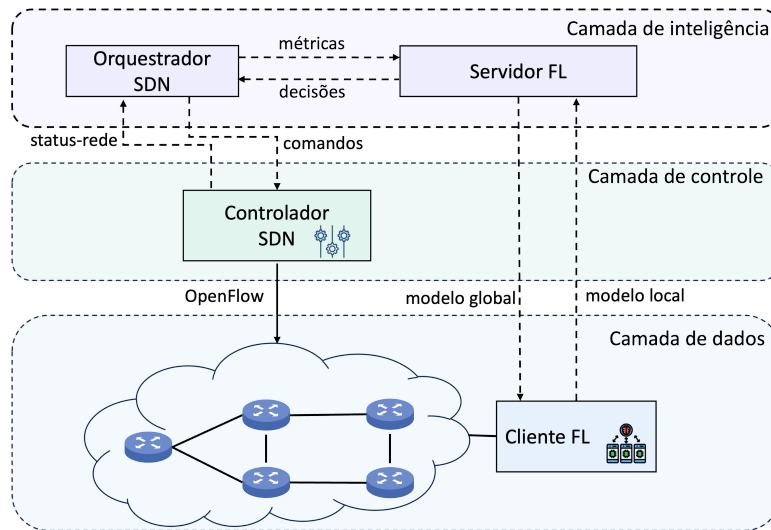


Figura 1: Arquitetura SDN-FL.

A **camada de dados** é composta pelos switches OpenFlow e pelos enlaces que interligam clientes ao servidor FL; é nesta camada que o tráfego de modelos efetivamente transita e onde o congestionamento se manifesta. A **camada de controle** é implementada pelo Controlador OpenDaylight (ODL). Ele mantém uma visão centralizada e atualizada da topologia via LLDP, traduz decisões de encaminhamento em regras OpenFlow e as instala nos switches. A **camada de inteligência** fecha o ciclo de co-orquestração. O Orquestrador SDN monitora a camada de dados, através do Controlador SDN, computa métricas de qualidade de caminho e as disponibiliza ao Servidor FL, que por sua vez as incorpora nos critérios de seleção de clientes para cada rodada federada.

A interdependência entre essas três camadas é precisamente o que distingue a proposta das abordagens que tratam a rede como meio passivo: a infraestrutura observa o tráfego, reage ao congestionamento e informa ao processo de aprendizado. O processo

de aprendizado, por sua vez, seleciona participantes de modo a reduzir a pressão sobre a infraestrutura. Esse ciclo virtuoso é o cerne do conceito de co-orquestração que permeia todo o sistema. A Figura 2 ilustra o fluxo de co-orquestração entre Servidor FL, Orquestrador SDN, Controlador OpenFlow e Clientes FL, que se desdobra em dois ciclos operacionais distintos, porém interconectados: um ciclo de gerenciamento proativo da rede e um ciclo de execução do round federado.

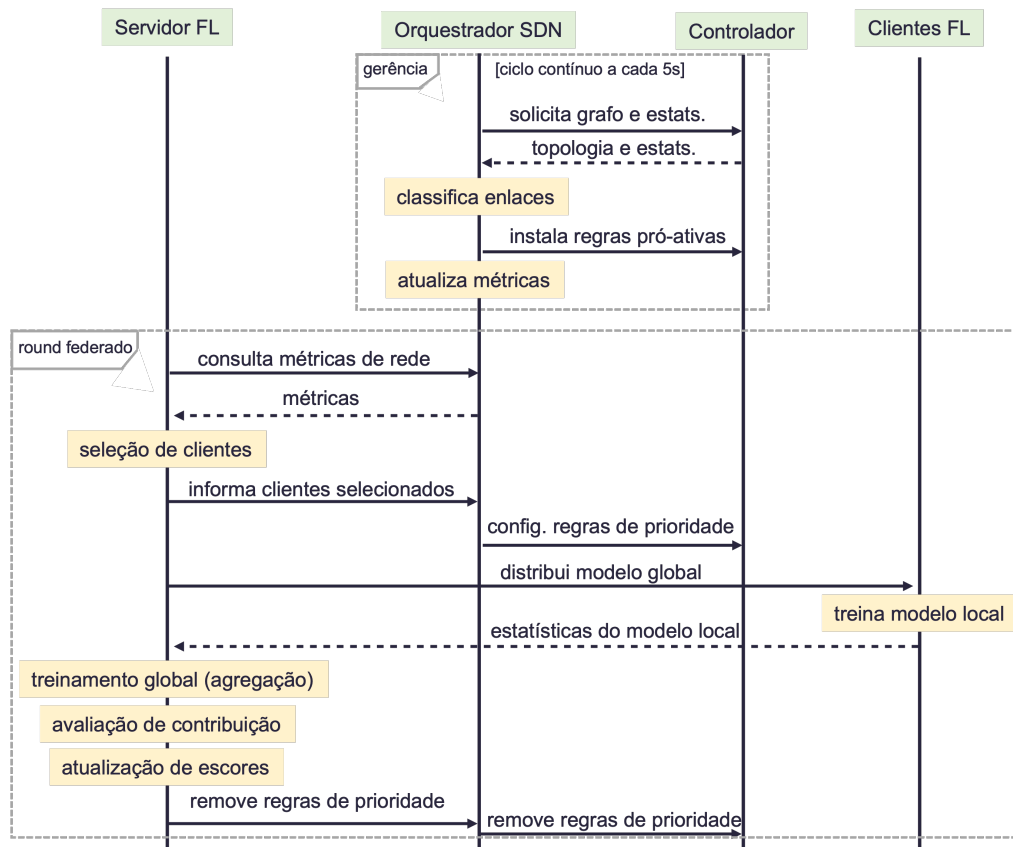


Figura 2: Ciclo de co-orquestração.

Ciclo de Gerenciamento Proativo da Rede

De forma assíncrona e contínua, o Orquestrador SDN executa um ciclo de gerenciamento para manter uma visão atualizada e otimizada da rede, independentemente da execução dos rounds de FL. Este ciclo consiste em três etapas:

1. Coleta de Topologia e Estatísticas: O Orquestrador solicita ao Controlador SDN (via sua API Northbound) o grafo da topologia de rede e as estatísticas de tráfego de todos os enlaces e portas dos switches OpenFlow. O uso de um canal de gerência *Out-of-Band* (OOB) garante que este tráfego de controle não seja afetado pelo congestionamento na camada de dados, assegurando a resiliência do sistema [Caria and Jukan 2018].
2. Classificação e Ponderação de Enlaces: Com os dados em mãos, o Orquestrador classifica o estado de cada enlace (e.g., ocioso, moderado, congestionado) com base em métricas como utilização de banda, latência, perda de pacotes e jitter. Cada

enlace do grafo recebe um peso inversamente proporcional à sua qualidade, então enlaces congestionados recebem um peso maior.

3. Instalação de Regras Pró-ativas: Utilizando o grafo ponderado, o Orquestrador calcula as rotas de menor custo entre todos os pares de nós da rede. Em seguida, ele instala proativamente regras de encaminhamento nos switches OpenFlow, garantindo que, por padrão, o tráfego flua pelos caminhos mais eficientes disponíveis naquele momento. As métricas consolidadas são então armazenadas para consulta pelo Servidor FL.

Ciclo de Execução do Round Federado

O ciclo do round federado é iniciado pelo Servidor FL e orquestrado em conjunto com a camada SDN, podendo ser dividido em 4 etapas:

1. Consulta de Métricas e Seleção de Clientes: No início de um round, o Servidor FL consulta o Orquestrador SDN para obter as métricas de rede mais recentes de todos os clientes elegíveis. Com base nessas informações e no histórico de desempenho dos clientes (detalhado na Seção 3.1), o servidor executa a seleção adaptativa, escolhendo o subconjunto de clientes mais aptos a participar.
2. Configuração de Prioridade e Distribuição do Modelo: O Servidor FL informa ao Orquestrador SDN quais clientes foram selecionados. O Orquestrador, por sua vez, instrui o Controlador a instalar regras de Qualidade de Serviço (QoS) nos switches, marcando o tráfego de e para os clientes selecionados com alta prioridade. Imediatamente após, o Servidor FL distribui o modelo global para os clientes escolhidos. O roteamento proativo previamente configurado garante que essa distribuição ocorra pelos caminhos mais eficientes.
3. Treinamento Local e Retorno das Estatísticas: Os clientes executam o treinamento local sobre seus dados. Ao final, eles enviam suas estatísticas de treinamento (e.g., acurácia e AUC-ROC do modelo local) ao Servidor FL.
4. Agregação, Avaliação e Atualização de Escores: O Servidor FL realiza a agregação dos modelos, avalia a contribuição de cada cliente para o modelo global e atualiza seus índices de qualidade. Ao final do round, ele informa ao Orquestrador para que as regras de prioridade de QoS sejam removidas, liberando os recursos de rede.

3.1. Seleção Adaptativa de Clientes

A seleção adaptativa de clientes é o mecanismo que integra operacionalmente as três camadas a cada rodada federada. Esta seleção é composta de duas etapas. É realizado um primeiro filtro de clientes baseado nos parâmetros de rede coletados pelo Orquestrador SDN. Neste primeiro filtro, os clientes que não atendem a requisitos mínimos de conectividade são excluídos da rodada. Um cliente é excluído nesta etapa se apresentar ao menos uma das seguintes condições: (i) largura de banda disponível inferior a 15 Mbps (medida no gargalo do caminho fim-a-fim); (ii) latência superior a 50 ms; ou (iii) taxa de perda de pacotes superior a 10%. Esses limiares podem ser calibrados de modo a excluir um cliente que poderia causar um custo de comunicação superior ao benefício esperado de sua contribuição ao modelo global, independentemente de sua qualidade histórica. A exclusão nessa etapa é rígida e não sujeita ao limite de exclusões da etapa seguinte.

Os clientes aprovados pelo filtro de rede são classificados por um índice de qualidade $Q_i \in [0, 1]$ que combina três componentes com seus respectivos pesos, conforme a Equação 1:

$$Q_i = 0,4 \cdot S_i^{\text{contrib}} + 0,3 \cdot S_i^{\text{treino}} + 0,3 \cdot S_i^{\text{rede}} \quad (1)$$

O componente S_i^{contrib} quantifica o quanto cada cliente melhora o modelo global. O sistema implementa uma técnica de avaliação por remoção individual: ao fim de cada rodada, o servidor mede a acurácia do modelo federado e, em seguida, repete a medição excluindo cada cliente um por vez. A diferença entre as duas acurácias revela o impacto marginal daquele cliente (Equação 2). Os valores de δ_i são normalizados em relação ao maior impacto absoluto observado na rodada, de modo que $S_i^{\text{contrib}} = 1$ corresponde ao cliente de maior contribuição positiva, 0,5 ao cliente neutro e 0 ao de maior impacto negativo (Equação 3). O peso majoritário dessa dimensão (40%) reflete a prioridade central do aprendizado federado, que é preservar a qualidade do modelo global.

$$\delta_i = \text{Acc}_{\text{todos}} - \text{Acc}_i \quad (2) \quad S_i^{\text{contrib}} = 0,5 + \frac{\delta_i}{2 \cdot \max_j |\delta_j|} \quad (3)$$

O componente S_i^{treino} define o custo do treinamento local e é usado para penalizar clientes que atrasam a rodada global por limitações computacionais ou de volume de dados. Em aprendizado federado síncrono, o servidor aguarda o cliente mais lento para iniciar a agregação. Então excluir o participante mais lento reduz diretamente o tempo total da rodada. A pontuação é calculada em relação ao tempo máximo de treinamento observado (Equação 4), onde t_i é o tempo de treinamento local do cliente i .

$$S_i^{\text{treino}} = 1 - \frac{t_i}{\max_j t_j} \quad (4)$$

O componente S_i^{rede} define a qualidade da rede e é calculado pelo orquestrador conforme a Equação (5).

$$S_i^{\text{rede}} = 0,5 \cdot \frac{b_i}{b_{\text{max}}} + 0,3 \cdot \left(1 - \frac{l_i}{l_{\text{max}}}\right) + 0,2 \cdot \left(1 - \frac{p_i}{p_{\text{max}}}\right) \quad (5)$$

onde b_i (Mbps), l_i (ms) e $p_i \in [0, 1]$ são, respectivamente, a largura de banda do enlace gargalo, a latência e a taxa de perda de pacotes do cliente i ; $b_{\text{max}} = 20$ Mbps é a capacidade física dos enlaces; e $l_{\text{max}}, p_{\text{max}}$ são os limiares de exclusão definidos na primeira etapa desta seção. Os pesos atribuídos a cada termo refletem sua influência relativa no desempenho da comunicação: a largura de banda recebe maior peso (0,5), por ser o principal determinante do tempo de transferência de modelos, especialmente em cenários de congestionamento, enquanto latência (0,3) e perda de pacotes (0,2) atuam como fatores complementares. Essa componente estabelece o elo entre o estado da infraestrutura de rede e o processo de seleção de clientes. Clientes com caminhos congestionados tendem a apresentar menores valores de S_i^{rede} , aumentando sua probabilidade de exclusão em rodadas nas quais sua participação implicaria maior custo de comunicação para o sistema.

Ao final, clientes com $Q_i < 0,50$ são candidatos à exclusão, com limite de no máximo dois clientes excluídos por rodada para preservar a diversidade do modelo federado. O mecanismo de exclusão é ativado somente a partir da terceira rodada, permitindo acumular histórico de contribuição antes de qualquer decisão de exclusão.

3.2. Estratégia de Agregação e Refinamento de Modelos Federados

Diferente da maioria das abordagens na literatura de Aprendizado Federado (FL), que se concentram quase exclusivamente em Redes Neurais Profundas (DNNs), esta arquitetura adota o XGBoost [Chen and Guestrin 2016] como modelo de referência. Esta escolha é fundamentada na busca por um equilíbrio entre eficiência computacional e desempenho preditivo, especialmente em cenários de computação de borda (Edge Computing) onde os dispositivos participantes podem possuir restrições de hardware. Algoritmos de *Gradient Boosting Decision Trees* (GBDT) são reconhecidos por sua capacidade de atingir alta acurácia com um custo computacional e de memória significativamente inferior ao de DNNs equivalentes [Nilsson et al. 2018], tornando o processo de treinamento local mais ágil e menos oneroso para os clientes.

Para validar a robustez da co-orquestração SDN-FL em problemas de larga escala e alta dimensionalidade, o sistema utiliza conjuntos de dados de benchmark consolidados (e.g., Epsilon e HIGGS). A federação desses modelos, entretanto, impõe desafios não triviais, uma vez que não existe uma operação de média direta entre parâmetros de árvores. Para viabilizar o treinamento distribuído, utilizamos a estratégia de agregação de modelos com *warm start*. A cada rodada r , o servidor distribui o modelo global federado acumulado e cada cliente o refina adicionando um conjunto de novas árvores, treinadas sobre seus dados locais, sem reconstruir o modelo do zero. O modelo final resultante é, portanto, um conjunto de modelos federados provenientes das contribuições de cada participante.

Para mitigar o risco de sobreajuste (*overfitting*) e garantir a estabilidade do aprendizado à medida que o número de árvores cresce, aplicamos um mecanismo de decaimento exponencial adaptativo tanto no número de novas árvores (n_{new}) quanto na taxa de aprendizado (η), conforme as Equações (6) e (7), respectivamente:

$$n_{\text{new}}(r) = \max(E \cdot \delta_{\min}, E \cdot \alpha^{r-1}) \quad (6) \quad \eta(r) = \max(\eta_0 \cdot \gamma_{\min}, \eta_0 \cdot \beta^{r-1}) \quad (7)$$

onde E é o número base de árvores e η_0 a taxa inicial. Os fatores de decaimento α e β garantem que, nas rodadas iniciais ($r = 1$), o modelo explore rapidamente o espaço de busca com o máximo de árvores (E) e taxa de aprendizado (η_0). Nas rodadas subsequentes, os incrementos tornam-se progressivamente menores e mais conservadores, respeitando os pisos mínimos ($\delta_{\min}, \gamma_{\min}$) definidos na Tabela 1 para evitar a estagnação prematura do aprendizado.

Tabela 1: Parâmetros de decaimento do treinamento federado.

Parâmetro	Valor	Significado
E	50	Número base de árvores novas por rodada
η_0	0,1	Taxa de aprendizado inicial
α	0,85	Fator de decaimento do número de árvores
β	0,93	Fator de decaimento da taxa de aprendizado
δ_{\min}	0,30	Piso do decaimento das árvores (30% de E)
γ_{\min}	0,40	Piso do decaimento da taxa de aprendizado (40% de η_0)

O controle de complexidade é reforçado por um teto de 500 árvores acumuladas e pela técnica de *early stopping*, que interrompe o treinamento local se não houver

melhora na perda logarítmica (*log loss*) de validação por 10 iterações consecutivas. Na fase de agregação, o servidor combina as previsões dos modelos locais em um modelo global final, utilizando pesos proporcionais à acurácia individual de cada cliente, garantindo que a co-orquestração SDN-FL resulte em um modelo federado robusto e eficiente, independentemente da complexidade do domínio de dados.

4. Metodologia e Cenário Experimental

Para avaliar a eficácia da arquitetura de co-orquestração SDN-FL, conduziu-se uma série de experimentos em um ambiente de simulação, com o objetivo de quantificar o impacto da gerência proativa de rede e da seleção adaptativa de clientes no tempo de convergência e na qualidade do modelo global em cenários de congestionamento. Os experimentos foram realizados no simulador GNS3 [GNS3 Technologies Inc. 2024] sobre uma topologia hierárquica, conforme Figura 3, composta por 14 switches Open vSwitch (OVS) operando com o protocolo OpenFlow 1.3 e controlador OpenDaylight (ODL). Os switches foram organizados em três camadas (núcleo, agregação e borda), garantindo a existência de caminhos alternativos para o reencaminhamento dinâmico de fluxos.

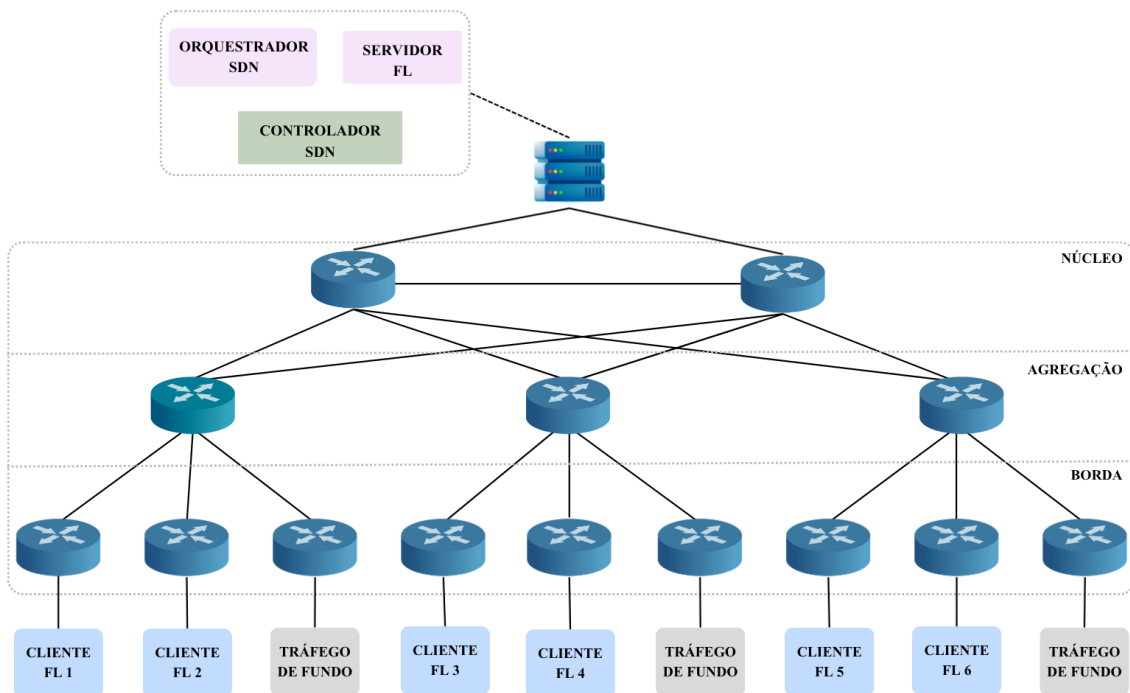


Figura 3: Topologia hierárquica implementada no GNS3.

Para evitar que o processamento dos switches se tornasse um gargalo sistêmico, a capacidade de cada enlace foi limitada a 20 Mbps. Ressalta-se que o foco da avaliação não reside na magnitude absoluta da largura de banda, mas sim na dinâmica de contenção de recursos. Ao induzir tráfego de fundo superior à capacidade nominal, estabelecemos um cenário de saturação que permite validar a eficácia dos mecanismos de priorização e reencaminhamento em condições de estresse. Além disso, 20 Mbps é uma largura de banda realista para cenários de *Edge Computing* conectados por tecnologias como 4G ou Wi-Fi compartilhado, conforme evidenciado por medições em redes móveis reais [Cristino and Caminha 2024]. A camada de controle é gerenciada através de uma rede

dedicada (*Out-of-Band*), isolando o tráfego de controle das flutuações de carga da camada de dados. O congestionamento foi induzido pelos três nós geradores que injetam fluxos UDP paralelos nos enlaces de agregação e núcleo.

O processo de treinamento foi executado no framework Flower, utilizando o algoritmo XGBoost com a estratégia de *warm start* e decaimento exponencial detalhada na Seção 3.2. Foram utilizados três conjuntos de dados de benchmark: (i) Epsilon ($\approx 400k$ amostras, 2.000 atributos, binário); (ii) HIGGS ($\approx 500k$ amostras, subconjunto, 28 atributos físicos, binário); e (iii) MNIST (70k amostras, 784 atributos, 10 classes). A partição dos dados foi realizada de forma IID (*Independent and Identically Distributed*), com cada cliente recebendo igual proporção das amostras (65% treino, 20% teste, 15% validação). Nessa condição, as contribuições S_i^{contrib} tendem a ser homogêneas entre nós, de modo que Q_i é determinado principalmente pelos componentes de rede e tempo de treinamento.

Para isolar os benefícios de cada componente da arquitetura, definimos três cenários comparativos detalhados na Tabela 2: o Baseline I, sem SDN e com agregação convencional; o Baseline II, com roteamento sensível à carga utilizando SDN, mas seleção de clientes convencional; e o SDN-FL, que integra o ciclo completo de co-orquestração. Os parâmetros de treinamento, idênticos em todos os cenários, estão consolidados na Tabela 3. O desempenho foi avaliado através do tempo de convergência e métricas de qualidade do modelo final (Acurácia, AUC-ROC e F1-Score). Os valores da Tabela 3 foram definidos empiricamente, buscando estabilidade de convergência dentro de 20 rodadas com viabilidade computacional no ambiente emulado.

Tabela 2: Cenários experimentais.

Cenário	SDN	Seleção
Baseline I	Inativo	Fixa
Baseline II	Ativo	Fixa
SDN-FL	Ativo	Adaptativa

Tabela 3: Parâmetros de treinamento.

Parâmetro	Valor
Rodadas / Clientes	20 / 6
Épocas locais (base)	50
Teto de árvores	500
Validação / Teste	15% / 20%

5. Resultados e Discussão

A Tabela 4 consolida o desempenho final da arquitetura SDN-FL em termos de tempo de convergência e métricas de qualidade preditiva. Os resultados mostram que a solução supera o Baseline I em todos os dados avaliados, alcançando uma aceleração média de $2,07\times$ (speedup). No dataset MNIST, a redução no tempo de treinamento atingiu 57,6%, o que evidencia a escalabilidade da solução para problemas multiclasse. É fundamental observar que este ganho de eficiência operacional não acarreta degradação estatisticamente relevante na qualidade dos modelos: as variações em Acurácia, AUC e F1-Score permaneceram dentro de uma margem de $\pm 0,002$, o que valida a eficácia do índice de qualidade de clientes em equilibrar a velocidade de rede com a relevância dos dados locais.

A Figura 4 detalha a dinâmica temporal do aprendizado, correlacionando a evolução da AUC com o tempo de execução. Os resultados mostram que a vantagem da co-orquestração SDN-FL acontece já nas rodadas iniciais e amplia-se progressivamente, atingindo patamares de qualidade equivalentes à convergência final do Baseline I em

aproximadamente metade do tempo total de execução. As curvas do Baseline I e do Baseline II apresentam uma inclinação mais suave devido à latência imposta por clientes lentos (*stragglers*). Já a curva do SDN-FL exibe uma trajetória mais íngreme e estável. Este comportamento decorre da estabilidade de round proporcionada pela seleção adaptativa. Ao filtrar participantes, o sistema não apenas acelera cada rodada individualmente, mas reduz a variabilidade temporal entre elas.

Tabela 4: Tempo de convergência e métricas de qualidade.

Dataset	Cenário	Tempo (s)	vs Baseline	AUC	F1-Score
Epsilon	SDN-FL	421	-51,4%	0,5893	0,5625
	Baseline I	866	—	0,5878	0,5612
	Baseline II	978	+13,0%	0,5878	0,5612
HIGGS	SDN-FL	764	-45,0%	0,8233	0,7587
	Baseline I	1.389	—	0,8241	0,7592
	Baseline II	1.516	+9,1%	0,8241	0,7592
MNIST	SDN-FL	481	-57,6%	0,9963	0,9755
	Baseline I	1.134	—	0,9966	0,9776
	Baseline II	1.243	+9,6%	0,9964	0,9768

Na Figura 4, um achado interessante é o desempenho do cenário Baseline II. Embora utilize o reencaminhamento proativo sensível à carga, ele apresentou uma piora de desempenho entre 9% e 13% em relação ao Baseline I (sem SDN). Esta degradação é explicada pelo comportamento do TCP quando a instalação dinâmica de novas rotas durante transferências ativas de modelos causa o reordenamento de pacotes (*out-of-order delivery*). O TCP interpreta esse reordenamento como perda, disparando retransmissões desnecessárias e reduzindo a janela de congestionamento, fenômeno análogo ao impacto de atrasos em sistemas distribuídos reportado por [Oliveira et al. 2025]. Sem a seleção adaptativa de clientes para mitigar esse efeito, o benefício do caminho alternativo é anulado pela instabilidade do transporte, reforçando a tese de que a gerência de rede isolada pode ser insuficiente para otimizar aplicações de IA distribuída.

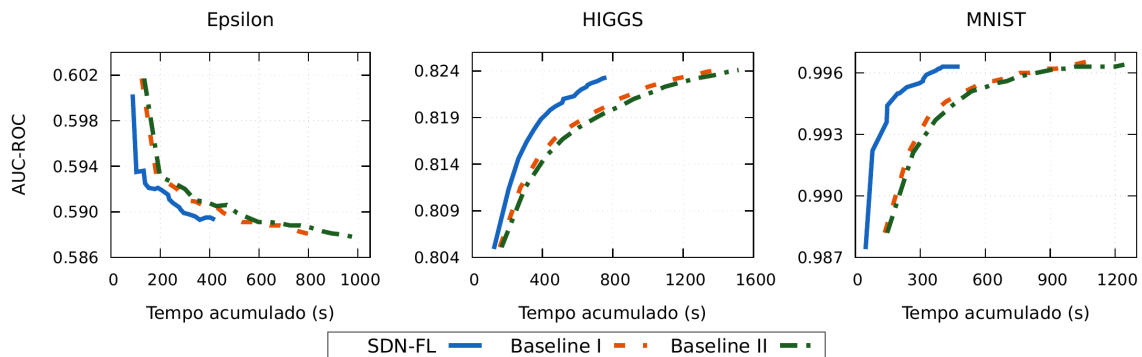


Figura 4: AUC x Tempo acumulado.

A curva do Epsilon exibe oscilações de AUC ausentes nos demais datasets, comportamento esperado em *gradient boosting* de alta dimensionalidade, onde a adição sequencial de árvores ao longo das rodadas eleva o risco de sobreajuste [Chen and Guestrin 2016].

A convergência acelerada do SDN-FL reduz o número de rodadas necessárias, limitando a exposição a esse fenômeno e as métricas finais equivalentes entre cenários (Tabela 4) confirmam que a aceleração não compromete a qualidade do modelo.

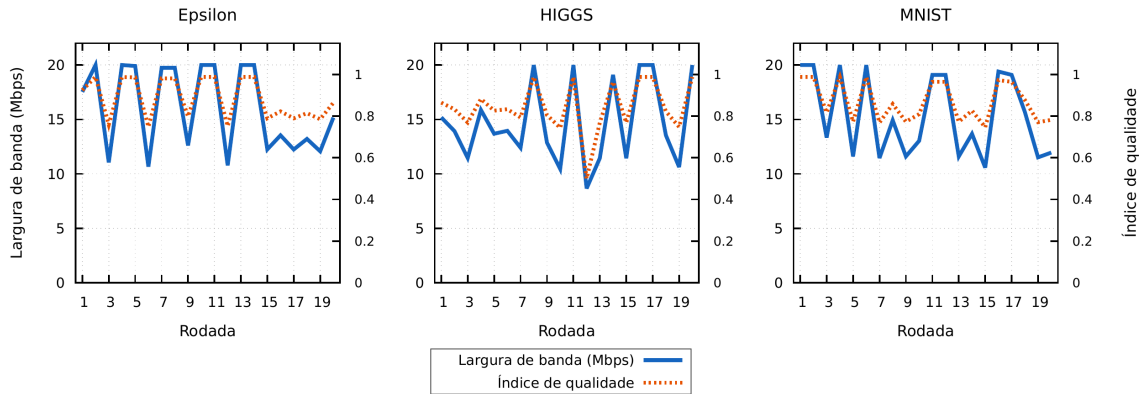


Figura 5: Largura de banda estimada e índice de qualidade médio Q_i por rodada (cenário SDN-FL).

A Figura 5 ilustra, para o cenário SDN-FL, a correlação entre a largura de banda estimada (variando entre 10 e 20 Mbps) e o índice de qualidade médio Q_i dos clientes selecionados em cada rodada, computado pela Equação (1). Esta sensibilidade permite que o sistema reaja a flutuações que medições simplistas no enlace de borda não conseguiriam capturar. Em suma, a arquitetura SDN-FL transforma a rede de um canal passivo em um componente ativo e inteligente, essencial para a viabilidade de sistemas de Aprendizado Federado em infraestruturas de rede compartilhadas e dinâmicas.

Os resultados demonstram que a arquitetura SDN-FL é eficaz em acelerar o aprendizado federado sem comprometer a qualidade dos modelos em cenários com particionamento IID dos dados. Reconhece-se, contudo, que essa condição representa uma limitação do presente estudo: em cenários *non-IID*, nos quais os clientes possuem distribuição de dados heterogêneas, o limiar $Q_i < 0,50$ (Equação 1) pode penalizar clientes com conhecimento local único. Por isso, a calibração cuidadosa dos parâmetros de exclusão e a investigação de limiares adaptativos em função da heterogeneidade dos dados constituem uma extensão natural deste trabalho, atualmente em desenvolvimento.

6. Conclusão

Este trabalho propôs e avaliou uma arquitetura de co-orquestração SDN-FL organizada em três camadas funcionais: uma camada de dados que transporta o tráfego de modelos, uma camada de controle programado por um controlador e uma camada de inteligência onde um Orquestrador SDN e um Servidor FL estabelecem um ciclo de retroalimentação. O orquestrador monitora continuamente a rede, realiza roteamento proativo sensível à carga e reencaminhamento reativo ao detectar congestionamento, estima a largura de banda disponível pelo gargalo do caminho completo e aplica diferenciação de tráfego por marcação DSCP. Essas informações alimentam o Servidor FL, que as incorpora em um índice de qualidade tridimensional combinando contribuição histórica ao modelo global (40%), eficiência de tempo de treinamento local (30%) e qualidade de rede (30%) para selecionar adaptativamente os clientes a cada rodada.

Experimentos em topologia hierárquica emulada com 14 switches OpenFlow sob congestionamento controlado demonstraram que a arquitetura SDN-FL reduz o tempo de convergência em 45–58% em relação ao baseline sem SDN, com speedup médio de $2,07\times$ e sem degradação relevante nas métricas de qualidade dos modelos. O cenário com roteamento SDN isolado (sem seleção adaptativa) revelou-se 9–13% mais lento que o próprio baseline, demonstrando que o reencaminhamento dinâmico durante transmissões TCP ativas impõe overhead que supera o benefício dos caminhos alternativos — e que a seleção adaptativa de clientes é o mecanismo dominante para o ganho de desempenho observado, não o roteamento isolado. A arquitetura mostra-se igualmente aplicável em cenários industriais de gerência de redes, com ganhos significativos sobre dados tabulares.

Como trabalhos futuros pretende-se: avaliar outros modelos de *gradient boosting*, redes neurais profundas e conjuntos de dados com heterogeneidade estatística (Non-IID); desenvolver mecanismos de agregação paramétrica para heterogeneidade de épocas locais; e utilizar políticas de reencaminhamento que minimizem a disrupção de conexões TCP ativas (e.g., janelas de migração de fluxo ou *hysteresis*).

Referências

- Abdulrahman, S., Tout, H., Ould-Slimane, H., Mourad, A., Talhi, C., and Guizani, M. (2021). A survey on federated learning: The journey from centralized to distributed on-site learning and beyond. *IEEE Internet of Things Journal*, 8(7):5476–5497.
- Aledhari, M., Razzak, R., Parizi, R. M., and Saeed, F. (2020). Federated learning: A survey on enabling technologies, protocols, and machine learning models. *IEEE Access*, 8:140699–140725.
- Alves, V. R. M., da Costa, J. B. D., Gonzalez, L. F. G., de Souza, A. M., and Villas, L. A. (2024). Seleção de clientes adaptativa baseada em privacidade diferencial para aprendizado federado. In *Anais Estendidos do XLII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 225–232. SBC.
- Caria, M. and Jukan, A. (2018). A comprehensive analysis on control plane deployment in SDN: In-band versus out-of-band solutions. *IEEE Communications Surveys & Tutorials*, 20(4):3473–3488.
- Chen, T. and Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794.
- Cristino, L. F. C. and Caminha, P. H. C. (2024). Caracterização da latência de borda sob efeito de mobilidade a partir de dados reais. In *Anais do Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*. SBC.
- Dakhia, Z. and Merenda, M. (2025). Client selection in federated learning on resource-constrained devices: A game theory approach. *Applied Sciences*, 15:7556.
- de Souza, A. M., Maciel, F., da Costa, J. B. D., Bittencourt, L. F., Cerqueira, E., Loureiro, A. A. F., and Villas, L. A. (2024). Adaptive client selection with personalization for communication efficient federated learning. *Ad Hoc Networks*, 157:103462.
- GNS3 Technologies Inc. (2024). GNS3: The software that empowers network professionals. <https://www.gns3.com>.
- Hamdan, O. A., Che, H., Arslan, E., and Arifuzzaman, M. (2025). SmartFlow: A communication-efficient SDN framework for cross-silo federated learning. Preprint.

- Konečný, J., McMahan, H. B., Yu, F. X., Richtárik, P., Suresh, A. T., and Bacon, D. (2016). Federated learning: Strategies for improving communication efficiency. In *NIPS Workshop on Private Multi-Party Machine Learning*.
- Libardi, G. M. A., Kimura, B. Y. L., and da Costa, J. B. D. (2025). Menos é mais: Avaliação do impacto da compressão de modelos na eficiência do aprendizado federado. In *Anais do Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*. SBC. UNIFESP – São José dos Campos.
- Ma, Z., Ma, J., Miao, Y., Li, X., Liu, Z., and Li, H. (2022). Applying federated learning in software-defined networks: A survey. *Symmetry*, 14(2):195.
- Mahmod, A., Pace, P., and Iera, A. (2025). The role of SDN to improve client selection in federated learning. *IEEE Communications Magazine*, 63(3):212–218.
- McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1273–1282.
- Nilsson, A., Smith, S., Ulm, G., Gustavsson, E., and Jirstrand, M. (2018). A performance evaluation of federated learning algorithms. arXiv preprint arXiv:1811.09655.
- Nishio, T. and Yonetani, R. (2019). FedCS: Communication-efficient federated learning with client selection. In *IEEE International Conference on Communications (ICC)*, pages 1–7.
- Oliveira, V. R., Oliveira, J. M., Macedo, D., and Nogueira, J. M. (2025). Tolerância a falhas com aprendizado por reforço para tomada de decisão em cenários distribuídos. In *Anais do Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*. SBC.
- Seo, E., Niyato, D., and Elmroth, E. (2021). Auction-based federated learning using software-defined networking for resource efficiency. In *2021 17th International Conference on Network and Service Management (CNSM)*, pages 42–48.
- Shahid, A. et al. (2021). Communication efficiency in federated learning: Achievements and challenges. arXiv preprint arXiv:2107.10996.
- Shen, Y., Xi, W., Cai, Y., Fan, Y., Yang, H., and Zhao, J. (2024). Multi-objective federated learning: Balancing global performance and individual fairness. *Future Generation Computer Systems*.
- Trindade, S., Bittencourt, L. F., and da Fonseca, N. L. S. (2022). Resource management at the network edge for federated learning. *Data and Communication Networks*, 8(4):555–571.
- Wu, J., Dong, F., Leung, H., Zhu, Z., Zhou, J., and Drew, S. (2024). Topology-aware federated learning in edge computing: A comprehensive survey. *ACM Computing Surveys*, 56(10):1–40.
- Xie, J., Yu, F. R., Huang, T., Xie, R., Liu, J., Wang, C., and Liu, Y. (2018). A survey of machine learning techniques applied to software defined networking (SDN): Research issues and challenges. *IEEE Communications Surveys & Tutorials*, 21(1):393–430.