

Aprendizado Federado Hierárquico: Uma Perspectiva Analítica com Redes de Petri Estocásticas

Israel Araújo¹, Luís Guilherme Silva¹, Francinaldo Barbosa¹,
Iure Fé¹, Geraldo P. Rocha Filho² e Francisco Airton Silva¹,

¹Universidade Federal do Piauí (UFPI)

²Universidade Estadual do Sudoeste da Bahia – BA – Brasil

{israel.araujo, luis.e, francinaldo.barbosa, iure.fe, faps}@ufpi.edu.br

geraldro.rocha@uesb.edu.br

Resumo. Avaliar Aprendizado Federado Hierárquico (HFL) em ambientes densos é difícil devido a limitações de escala e à variabilidade de rede e recursos. Embora arquiteturas hierárquicas empreguem servidores intermediários para agregações parciais, analisar seu comportamento por simulação ou experimentação ainda é custoso para explorar múltiplas configurações. Este artigo propõe um modelo em Redes de Petri Estocásticas (SPN) para representar HFL com servidor central e servidores intermediários. O modelo estima métricas de taxa de conclusão de rodadas, tempo médio de rodada e probabilidade de descarte. Os resultados mostram que aumentar o número de servidores intermediários melhora a vazão, reduz o tempo médio de rodada e amplia a capacidade operacional sob cargas elevadas. Em um cenário avaliado, a expansão de dois para quatro servidores intermediários reduziu o tempo médio de rodada em mais de 60% e elevou a taxa de conclusão de rodadas em até 160%.

Abstract. Evaluating Hierarchical Federated Learning (HFL) in dense environments is difficult due to scalability limitations and network and resource variability. Although hierarchical architectures employ intermediate servers for partial aggregations, analyzing their behavior through simulation or experimentation is still costly to explore multiple configurations. This paper proposes a Stochastic Petri Net (SPN) model to represent HFL with a central server and intermediate servers. The model estimates metrics for round completion rate, average round time, and discard probability. The results show that increasing the number of intermediate servers improves throughput, reduces average round time, and expands operational capacity under high loads. In one evaluated scenario, expanding from two to four intermediate servers reduced the average round time by more than 60% and increased the round completion rate by up to 160%.

1. Introdução

O Aprendizado Federado (FL) tem sido adotado como alternativa prática para o treinamento de modelos de inteligência artificial em ambientes distribuídos, especialmente diante do crescimento acelerado de dados gerados na borda da rede. Projeções da Statista indicam que o número de dispositivos de Internet das Coisas (IoT) deverá crescer de 19,8 bilhões em 2025 para aproximadamente 40,6 bilhões em 2034 [Statista 2025]. Esse crescimento projetado torna a centralização em nuvens tradicionais cada vez menos

viável, tanto por limitações de largura de banda quanto por exigências de privacidade. Nesse cenário, o FL permite que esses dispositivos participem do treinamento do modelo sem transferir dados brutos, mantendo a informação local e explorando a capacidade computacional distribuída presente na periferia das redes atuais [Wen et al. 2023].

Embora o modelo tradicional de FL tenha permitido avanços no treinamento distribuído, sua aplicação em larga escala apresenta limitações de escalabilidade e latência quando milhares de dispositivos se comunicam simultaneamente com um único servidor em nuvem. Esse padrão de comunicação concentra tráfego, aumenta o tempo de resposta e impõe carga elevada ao ponto central de agregação. Para contornar esse cenário, o Aprendizado Federado Hierárquico (HFL) introduz camadas intermediárias de agregação, como servidores de borda e nós de névoa (*fog nodes*) [Sharma et al. 2025]. As atualizações locais passam a ser processadas e combinadas regionalmente antes do envio ao servidor global. Essa organização reduz o volume de dados transmitidos em enlaces de longa distância e ajusta o treinamento à estrutura física e lógica das infraestruturas de rede atuais, mantendo a coordenação do modelo sem sobrecarregar a comunicação central [Ooi et al. 2023].

Apesar dos ganhos estruturais do HFL, sua adoção adiciona complexidade à gestão de recursos e à previsão de comportamento do sistema. A presença de múltiplos níveis de agregação, taxas de transmissão variáveis e dispositivos com capacidades computacionais distintas dificulta a estimativa prévia de desempenho [Cho et al. 2024]. A literatura recente sobre HFL mostra a variedade de aplicações e os desafios técnicos associados a essa arquitetura. Trabalhos concentram seus esforços na organização da infraestrutura, empregando camadas de edge e fog para reduzir latência e volume de tráfego. Na ausência de métodos analíticos prévios, decisões de configuração passam a ser feitas por tentativa e erro, como a definição do número de servidores intermediários até que se observe saturação de comunicação ou degradação do tempo de rodada [Zhao et al. 2022].

Com isso, as Redes de Petri Estocásticas (SPNs) oferecem um mecanismo formal para representar arquiteturas de HFL a partir de estados, eventos e recursos compartilhados, preservando as dependências temporais entre as fases do ciclo federado. Essa representação permite antecipar o comportamento do sistema sob diferentes configurações, além de incorporar atrasos, sincronizações e falhas diretamente na dinâmica estocástica [Fé et al. 2024]. As SPNs capturam efeitos de contenção e saturação que não são evidentes em análises puramente empíricas e com isso, o dimensionamento de servidores intermediários e a parametrização do ciclo federado deixam de depender de tentativa e erro, tornando o planejamento previsível e eficiente [Silva et al. 2023].

Este artigo tem como objetivo analisar o comportamento temporal do HFL por meio de um modelo analítico baseado em SPNs, considerando a progressão das rodadas federadas em uma arquitetura com múltiplos níveis de agregação. Especificamente, o trabalho busca (i) derivar métricas temporais em regime estacionário, como o tempo médio de rodada e a taxa de conclusão de rodadas, a partir da estrutura do modelo estocástico; (ii) representar a dinâmica da agregação intermediária e da coordenação global, permitindo analisar como diferentes configurações de sincronização e frequência de agregação influenciam a evolução das rodadas; e (iii) avaliar a aderência do modelo proposto por meio de comparação quantitativa com medições obtidas em simulações na ferramenta Mercury [Silva et al. 2015], considerando médias, variabilidade e intervalos de confiança.

O restante do artigo está organizado da seguinte forma: a Seção 2 apresenta os

trabalhos relacionados no contexto de HFL e avaliação analítica; a Seção 3 descreve a arquitetura hierárquica considerada no estudo; a Seção 4 detalha o modelo baseado em SPNs e as métricas extraídas; a Seção 5 apresenta os estudos de caso conduzidos a partir do modelo proposto; por fim, a Seção 6 reúne as conclusões do trabalho.

2. Trabalhos Relacionados

Nesta seção, são apresentados e discutidos os principais trabalhos que investigam o uso da arquitetura HFL, com o objetivo de identificar quais propostas são adotadas, quais métodos de avaliação são empregados, se há a utilização de modelos analíticos, quais ferramentas são usadas para análise e validação e quais métricas são consideradas para mensuração de desempenho. A Tabela 1 sintetiza essas informações, organizando os trabalhos de acordo com esses aspectos.

Proposta: Os estudos analisados exploram o HFL para otimizar o treinamento em redes distribuídas e heterogêneas. A eficiência de rede e redução de latência são fundamentais em [Liu et al. 2020] e [Manju et al. 2024], que utilizam camadas de *edge* e *fog* para agregações parciais. Em cenários críticos e de mobilidade, [Kumari et al. 2024] propõe frameworks para predição em veículos elétricos e gestão de desastres com UAVs. A robustez contra ataques e a privacidade são focos de [Siriwardhana et al. 2024] e [DP et al. 2023] no contexto de IoMT, enquanto [Li et al. 2022] foca na personalização via redes complexas e [Farajzadeh et al. 2025] expande o HFL para redes não-terrestres (NTN). Em contraste com essas abordagens, este trabalho introduz um modelo analítico em Redes de Petri Estocásticas (SPN), voltado à análise da dinâmica temporal de arquiteturas HFL e à representação de falhas no envio das atualizações dos clientes aos servidores intermediários.

Tipo de Avaliação: Os trabalhos da literatura apresentam uma forte inclinação para a validação por meio de simulações experimentais, abordagem adotada em quase a totalidade dos estudos, como em [Siriwardhana et al. 2024], [Kumari et al. 2024], [DP et al. 2023], [Manju et al. 2024] e [Li et al. 2022]. Uma vertente complementar utiliza simulações numéricas para corroborar provas de convergência ou modelos de canal específicos, técnica aplicada por [Liu et al. 2020] e [Farajzadeh et al. 2025]. Em vez de seguir a tendência predominantemente experimental ou numérica, este trabalho adota uma abordagem de modelagem formal baseada em Redes de Petri para prever o comportamento do sistema antes da execução empírica.

Modelo Analítico: No que tange à formalização teórica, a maioria dos trabalhos analisados não apresenta um modelo analítico estruturado, focando exclusivamente na experimentação empírica para validar suas hipóteses [Siriwardhana et al. 2024, Kumari et al. 2024, DP et al. 2023, Manju et al. 2024, Li et al. 2022, Farajzadeh et al. 2025]. Uma exceção é encontrada em [Liu et al. 2020], que utiliza formulações determinísticas voltadas à convergência e ao custo de comunicação. Este trabalho, por sua vez, emprega SPNs para representar explicitamente dependências temporais, contenção de recursos e falhas ao longo das rodadas federadas. Nesse contexto, a principal contribuição deste trabalho está na introdução de um modelo analítico baseado em SPNs, capaz de representar formalmente o comportamento dinâmico, contenção, falhas e temporal da arquitetura HFL.

Ferramentas: Para a implementação e validação das arquiteturas propostas, o ecossistema Python consolida-se como a base principal, com o uso extensivo de biblio-

Tabela 1. Trabalhos Relacionados

Trabalho	Proposta	Tipo de Avaliação	Modelo Analítico	Ferramenta de Análise/Validação	Métricas Analisadas
[Liu et al. 2020]	HierFAVG para redução de latência e sobrecarga em HFL.	Convergente e Numérica	✓	Análise e modelagem (CNN, MNIST e CIFAR-10)	Test Accuracy, Runtime, Local energy consumption, Training epochs, Training time
[Siriwardhana et al. 2024]	SHIELD: agregação robusta contra ataques de envenenamento em HFL.	Experimental	✗	Python 3.7 (TensorFlow e HDBSCAN)	Training Iterations, Training Loss, Class Level Accuracy, Average Euclidean Distance, Percentage of poisoning.
[Kumari et al. 2024]	S-EVPPF para predição segura e descentralizada em veículos elétricos.	Experimental	✗	Blockchain Ethereum e IPFS	Iteration, Accuracy, RMSE, MSE, Gas used, Performance, Mitigation Time Reduction, System Overhead
[DP et al. 2023]	Detecção de intrusões e privacidade em IoMT via HFL-HLSTM.	Experimental	✗	Python (Arquitura LS2DNN)	Accuracy, Precision, Recall, F1-Score, Sensitivity, Data Privacy
[Manju et al. 2024]	HFL com nós de névoa para redução de latência e sobrecarga de comunicação.	Experimental	✓	Python (TensorFlow e NumPy) e MATLAB	Convergence Time, Energy Consumption, Model Accuracy, Latency, Communication Overhead
[Li et al. 2022]	HPFL-CN: HFL personalizado com redução de heterogeneidade e sobrecarga.	Experimental	✗	Framework HPFL-CN (Algoritmo Walktrap e Redes Complexas)	Communication Rounds, Test Accuracy, Training Loss, Trade-off parameter, Aggregation round intervals
[Farajzadeh et al. 2025]	HFL com HAPS para redução de latência e melhoria de privacidade em NTN.	Numérica	✗	Simulador Numérico Customizado com Modelagem de Canal NTN (LEO/HAPS)	Communication Round, Model Accuracy, Total/Cumulative Latency, Average Training Loss
Este Trabalho	Modelo analítico em SPN para HFL com falhas cliente-servidor intermediário.	Modelagem	✓	Mercury Tools	Round rate, Round time, Probabilidade de descarte

tecas de aprendizado de máquina como TensorFlow e NumPy [Siriwardhana et al. 2024, Manju et al. 2024]. Frameworks específicos para Aprendizado Federado, como o Flower, são empregados para gerenciar a orquestração distribuída, enquanto soluções customizadas são desenvolvidas para lidar com características de redes complexas [Li et al. 2022] ou modelagem de canais NTN [Farajzadeh et al. 2025]. Em contextos de segurança e descentralização, tecnologias de registro distribuído como Blockchain Ethereum e IPFS são integradas ao fluxo de validação [Kumari et al. 2024]. Para a etapa de validação, o trabalho apoia-se na ferramenta Mercury, explorando diretamente o modelo analítico em SPN, sem depender de ambientes de simulação experimental tradicionais.

Métricas: A avaliação do desempenho nos trabalhos correlatos divide-se, predominantemente, entre métricas de eficácia do modelo de aprendizado e métricas de efi-

ciência da infraestrutura de rede. No campo do aprendizado, destacam-se a acurácia do modelo (*Test/Model Accuracy*), perda (*Training Loss*), precisão, *recall* e convergência [Siriwardhana et al. 2024, DP et al. 2023, Li et al. 2022, Farajzadeh et al. 2025]. Já sob a perspectiva de infraestrutura, os estudos focam em latência, consumo de energia, tempo de treinamento e sobrecarga de comunicação (*overhead*) [Liu et al. 2020, Manju et al. 2024]. Métricas específicas de sistemas descentralizados, como o custo de gas e confiança, aparecem em propostas baseadas em blockchain [Kumari et al. 2024]. A análise concentra-se em métricas de desempenho do sistema associadas à vazão e ao fluxo. São avaliados o *round rate*, o *round time* e a probabilidade de descarte, permitindo caracterizar a capacidade de processamento da arquitetura HFL de forma contínua por meio do modelo analítico.

A síntese apresentada na Tabela 1 evidencia dois padrões recorrentes na literatura. A grande maioria dos trabalhos baseia sua avaliação em simulações experimentais, enquanto a presença de modelos analíticos formais é rara e restrita a casos pontuais. Esse cenário indica que, embora existam diversas propostas arquiteturais para HFL, ainda há pouca exploração de métodos capazes de prever o comportamento do sistema por meio de modelagem analítica antes da validação empírica.

3. Arquitetura do Aprendizado Federado Hierárquico

A arquitetura adotada organiza o Aprendizado Federado (FL) em uma estrutura hierárquica composta por três níveis operacionais, conforme ilustrado na Figura 1. O nível inferior é formado pelos dispositivos finais, responsáveis pelo treinamento local a partir de dados próprios. O nível intermediário é composto por servidores regionais, que coordenam subconjuntos de dispositivos e realizam agregações parciais. O nível superior corresponde ao servidor de agregação global, responsável pela consolidação final do modelo e pela coordenação entre rodadas.

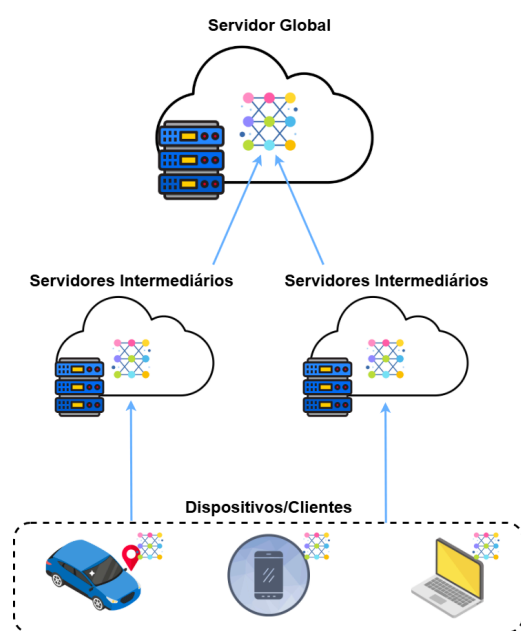


Figura 1. Arquitetura do Aprendizado Federado Hierárquico.

O ciclo federado tem início com a disponibilização do modelo global aos servidores regionais, que encaminham esse modelo aos dispositivos associados

[Zhang et al. 2020]. Após o treinamento local, as atualizações geradas são enviadas ao respectivo servidor intermediário, onde são coletadas e agregadas seguindo critérios de participação ou limites temporais definidos para a rodada. Essa agregação intermediária introduz um estágio adicional de sincronização, concentrando variações locais de conectividade e disponibilidade e evitando que atrasos individuais interfiram na progressão global.

Os servidores regionais enviam apenas os modelos consolidados ao servidor de agregação global, que combina as contribuições regionais e atualiza o estado do modelo compartilhado. Após essa etapa, o modelo atualizado é redistribuído aos servidores regionais, reiniciando o ciclo de treinamento de forma hierárquica. Essa separação funcional estabelece dependências temporais claras entre as fases do processo federado, fazendo com que o avanço das rodadas dependa tanto do treinamento local quanto da sincronização entre níveis. Essa organização melhora a escalabilidade, porém aumenta a complexidade de coordenação entre níveis, amplia o número de elementos monitorados e introduz novos pontos potenciais de falha.

Além da coordenação hierárquica, a arquitetura considera eventos de falha nos servidores regionais, que podem impedir temporariamente a agregação e o encaminhamento das atualizações. Esses eventos afetam a disponibilidade do nível intermediário, o tempo de conclusão das rodadas e a taxa de progressão do processo federado. Essa dinâmica é representada no modelo analítico por meio de estados, transições e tempos associados, conforme descrito na Seção 4, fornecendo a base para a avaliação das métricas temporais analisadas. Como o foco deste trabalho está na análise temporal da arquitetura HFL, o modelo abstrai o conteúdo do treinamento e, por isso, não depende de um dataset específico, mas dos tempos associados às etapas do ciclo federado.

4. Modelo SPN Proposto

Esta seção apresenta o modelo SPN proposto para o sistema de HFL, organizado em partes funcionais que representam as principais fases do ciclo federado. Conforme ilustrado na Figura 2, o modelo é organizado em quatro partes funcionais, cada uma associada a uma etapa do ciclo federado. A parte I (Admissão e Seleção de Clientes) descreve a admissão e a preparação dos clientes, na qual se estabelece a disponibilidade inicial para participação na rodada. A parte II (conexão com os servidores), o modelo representa a fase de conexão com a infraestrutura de servidores, responsável por capturar atrasos e falhas associados ao estabelecimento da comunicação. A parte III (servidores intermediários) corresponde ao processamento nos servidores intermediários, onde ocorre a coordenação regional e a agregação local das atualizações. Por fim, a parte IV (servidor central) modela a agregação no servidor central e a finalização/avaliação da rodada. A separação dessas etapas reflete a progressão natural do treinamento federado ao longo da hierarquia e define um fluxo sequencial de eventos. Cada fase condiciona o avanço da rodada seguinte, permitindo analisar o impacto temporal das interações entre clientes, servidores intermediários e o nível central.

No modelo, os *places* representam estados lógicos do sistema ou a disponibilidade de recursos, enquanto as transições descrevem eventos que promovem a evolução do fluxo de *tokens*. Transições temporizadas modelam atividades sujeitas a atraso, associadas a distribuições exponenciais, em função da propriedade de memória nula, que simplifica a análise analítica e é amplamente adotada em modelos estocásticos de desempenho. A análise em regime estacionário permite capturar o comportamento médio de longo prazo

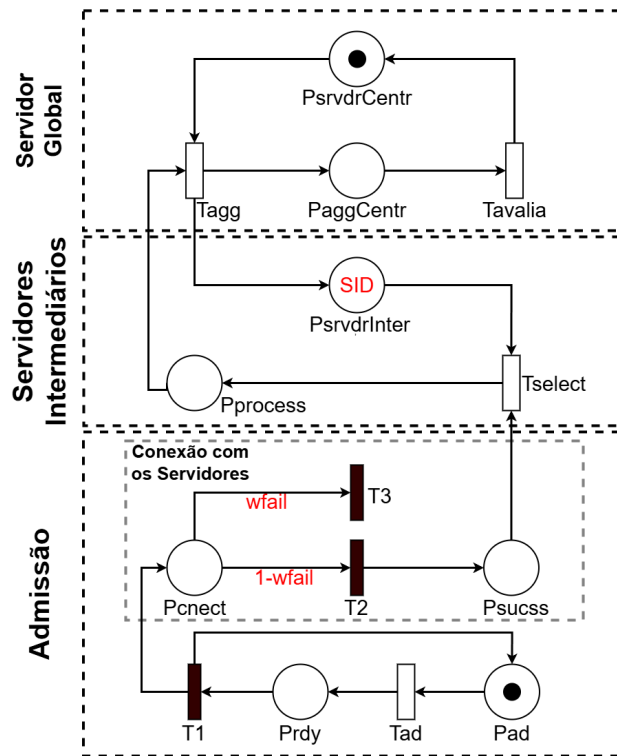


Figura 2. Modelo SPN do Aprendizado Federado Hierárquico.

do sistema, enquanto a análise transiente pode ser utilizada para observar a evolução temporal das rodadas em intervalos finitos, quando necessário.

A parte de admissão do modelo representa a entrada e a preparação dos clientes para participação em uma rodada federada. O *place Pad* armazena as *tokens* que representam clientes disponíveis no sistema, constituindo o ponto inicial do fluxo. A transição temporizada *Tad* modela o atraso associado ao processo de admissão, capturando o intervalo necessário para que um cliente se torne apto a prosseguir no ciclo federado. Após essa etapa, as *tokens* alcançam o *place Prdy*, que representa clientes prontos para tentar participar da rodada corrente. A progressão a partir desse estado é controlada pela transição *T1*, que atua como um ponto de sincronização entre a preparação dos clientes e a fase seguinte do modelo. Assim, a disponibilidade inicial da efetiva tentativa de conexão com a infraestrutura, permitindo que o impacto temporal da admissão seja tratado de forma independente das demais etapas do sistema.

A fase de conexão com a infraestrutura de servidores é representada pelo conjunto formado pelos lugares *Pconnect* e *Psucess* e pelas transições *T2* e *T3*. Após a liberação pela etapa de admissão, os *tokens* alcançam o *place Pconnect*, que representa clientes em processo de estabelecimento de conexão com os servidores intermediários. A transição temporizada *T2* modela o tempo necessário para a conclusão desse processo, incorporando atrasos associados à comunicação e à disponibilidade da infraestrutura. Em paralelo, a transição *T3* representa eventos de falha ou interrupção durante a tentativa de conexão, desviando o fluxo quando a comunicação não é concluída com sucesso. Os *tokens* que atravessam *T2* alcançam o *place Psucess*, indicando conexões bem-sucedidas e a aptidão para prosseguir no ciclo federado. Ao introduzir a variabilidade no número de clientes efetivamente conectados em cada rodada modelo, pode-se analisar o impacto da conecti-

vidade sobre a progressão temporal do sistema.

O processamento nos servidores intermediários é modelado a partir do conjunto de lugares *Tselect*, *Pprocess* e *PsvdrInter*, conectados pelas transições que controlam a seleção e a agregação intermediária. Os *tokens* provenientes de *Psucess* são encaminhados à transição *Tselect*, que representa o mecanismo de seleção dos clientes para participar da agregação intermediária definidos para a rodada. Uma vez selecionados, os *tokens* alcançam o *place Pprocess*, no qual ocorre o processamento associado à combinação das atualizações no nível intermediário. A disponibilidade do servidor intermediário é representada pelo *place PsvdrInter*, que atua como um recurso compartilhado e condiciona a execução dessa etapa. O parâmetro *SID* corresponde ao número máximo de servidores intermediários no nível regional. No modelo SPN, esse parâmetro define a capacidade do nível intermediário ao limitar o número de agregações regionais que podem ocorrer simultaneamente. Ao término desse estágio, as contribuições são encaminhadas para a fase seguinte do modelo, encerrando a coordenação intermediária da rodada federada.

A etapa correspondente ao servidor central representa o final das atualizações ao longo do HFL. Os *tokens* resultantes da agregação intermediária são encaminhados ao *place PaggCentr*, que armazena as contribuições da combinação global. A execução dessa etapa é condicionada pela disponibilidade do recurso central, modelada pelo *place PsvdrCentr*, garantindo que a agregação global ocorra apenas quando o servidor está acessível. A transição associada a esse estágio representa o tempo necessário para combinar os modelos regionais e atualizar o estado global do sistema. Após a conclusão da agregação, os *tokens* avançam para a transição *Tavalia*, que marca a finalização da rodada federada e o encerramento do ciclo representado no modelo, mantendo a separação entre coordenação central e processamento distribuído da arquitetura hierárquica. De forma geral, o modelo representa a progressão de uma rodada federada desde a admissão dos clientes até a agregação central, separando o fluxo em etapas que refletem conexão, processamento intermediário e consolidação global.

4.1. Métricas de Avaliação do Modelo

As métricas consideradas neste trabalho são extraídas diretamente da dinâmica estocástica do modelo e refletem propriedades temporais associadas à progressão das rodadas federadas ao longo da hierarquia. Em SPNs, métricas correspondem a grandezas derivadas da distribuição estacionária do sistema ou de taxas de disparo de transições, permitindo quantificar o comportamento médio do modelo sem a necessidade de execuções experimentais repetidas. Esse método possibilita avaliar o ritmo de operação do sistema e os efeitos das dependências entre estados, recursos e sincronizações impostas pela arquitetura hierárquica.

A Equação 1 *round_rate* indica quantas rodadas completas o sistema consegue concluir por unidade de tempo, ou seja, expressa a taxa de conclusão de rodadas como função do tempo médio de serviço da agregação central (*service_time*), que representa a duração da etapa de consolidação final no servidor global. No modelo proposto, essa taxa está associada ao *place PaggCentr*, que representa o acúmulo de agregações globais finalizadas ao longo do tempo. A razão entre o número médio de *tokens* neste *place* e o tempo médio de serviço da agregação central fornece uma medida direta da vazão do processo federado. Com isso, podem indicar maior frequência de conclusão das rodadas, refletindo menor impacto de atrasos introduzidos pelas fases de admissão, conexão e agregação intermediária.

$$\mathbf{round_rate} = \frac{E\{\#PaggCentr\}}{service_time} \quad (1)$$

O tempo médio de rodada denotado pela Equação 2, *round_time*, representa a duração média necessária para que uma rodada federada percorra todas as etapas do modelo, desde a progressão inicial até a agregação no servidor central. Essa métrica é definida como o inverso da taxa de rodadas e traduz o intervalo médio entre conclusões sucessivas de rodadas completas. No contexto do modelo, o *round_time* sintetiza o efeito combinado das taxas de admissão, dos atrasos de conexão, do processamento intermediário e da agregação central, além de permitir comparar diferentes configurações operacionais em termos de duração média das rodadas. Assim, quanto maior a taxa de rodadas, menor o tempo médio necessário para concluir uma rodada completa. Essa definição decorre do fato de que o tempo médio entre rodadas completas pode ser expresso como o inverso da taxa de conclusão de rodadas.

$$\mathbf{round_time} = \frac{1}{\left(\frac{E\{\#PaggCentr\}}{service_time}\right)} \quad (2)$$

A Equação 3 representa a probabilidade de descarte, denotada por *Drop Probability (DP)*, quantifica a ocorrência de situações em que não há contribuições disponíveis para a agregação intermediária durante uma rodada. Essa métrica está associada à probabilidade estacionária de o *place PaggInter* não conter *tokens*, indicando ausência de atualizações válidas provenientes do nível intermediário. A DP reflete diretamente o impacto de falhas de conexão, restrições de sincronização ou políticas de seleção excessivamente restritivas sobre a continuidade do processo federado onde valores elevados podem indicar maior frequência de rodadas improdutivas. Essa métrica indica a ocorrência de rodadas sem contribuições válidas no nível intermediário.

$$\mathbf{DP} = P\{\#PaggInter = 0\} \times 100 \quad (3)$$

5. Estudos de Caso

Esta Seção apresenta os estudos de caso conduzidos a partir do modelo analítico baseado em SPN. Os resultados foram obtidos com a ferramenta Mercury, versão 5.0.1, por meio de análise em regime estacionário para caracterizar o comportamento médio do sistema, adotando uma margem de erro aproximada de 2%. Os parâmetros utilizados, com valores empíricos parametrizados, assim como os componentes do modelo SPN associados às fases de admissão e agregação do ciclo de FL, são apresentados na Tabela 2. Os valores adotados foram definidos para explorar o comportamento do modelo sob diferentes condições operacionais, sem reproduzir diretamente uma implementação específica. A partir dessa configuração base, os estudos de caso avaliam métricas de desempenho, estabilidade e comportamento probabilístico do sistema.

5.1. Estudo de Caso 1 - Impacto da Taxa de Chegada e do Número de Servidores Intermediários no Desempenho do HFL

Esta subseção apresenta os resultados de desempenho do modelo analítico com base nas métricas ROUND_RATE, ROUND_TIME e DP. A Figura 3 analisa o efeito da taxa de chegada considerando configurações com 2, 3 e 4 servidores intermediários, sendo a taxa de

Tabela 2. Componentes e parâmetros utilizados no modelo.

Tipo	Componente	Valor	Definição
Places	Par	1	Entrada de requisições e modelos no sistema.
	PsrvdInter	SID	Capacidade dos servidores intermediários.
	PsrvdCenter	1	Capacidade do servidor central.
Transições Temporizadas	Tad	1.00 s	Admissão e encaminhamento de requisições.
	Tselect	0.005 s	Seleção e alocação dos servidores intermediário.
	Tagg	0.065 s	Agregação das atualizações intermediárias.
	Teval	2.452 s	Avaliação do modelo após agregação global.
Parâmetros	wfail	0.1	Acionado quando uma requisição falha.
	1-wfail	0.9	Indica a conclusão bem-sucedida da requisição.
	AD	1.0	Intervalo médio entre admissões.
	SID	4	Nº máximo de servidores intermediários.

conclusão de rodadas estimada pela métrica ROUND_RATE da Equação 1. Para baixas taxas de chegada, observa-se crescimento aproximadamente linear da taxa de rodadas em todas as configurações. Com 2 servidores intermediários, o ROUND_RATE aumenta de cerca de 1,1 para aproximadamente 2,5 rodadas por segundo, estabilizando-se a partir de taxas próximas de 2,5 msg/s, enquanto, com 3 e 4 servidores, a taxa alcança valores próximos de 4,3 e 6,5 rodadas por segundo, respectivamente, mantendo-se superior ao longo do intervalo analisado. Esse comportamento reflete o aumento da capacidade de processamento no nível intermediário e sua relação direta com a redução do ROUND_TIME, conforme as Equações 1 e 2.

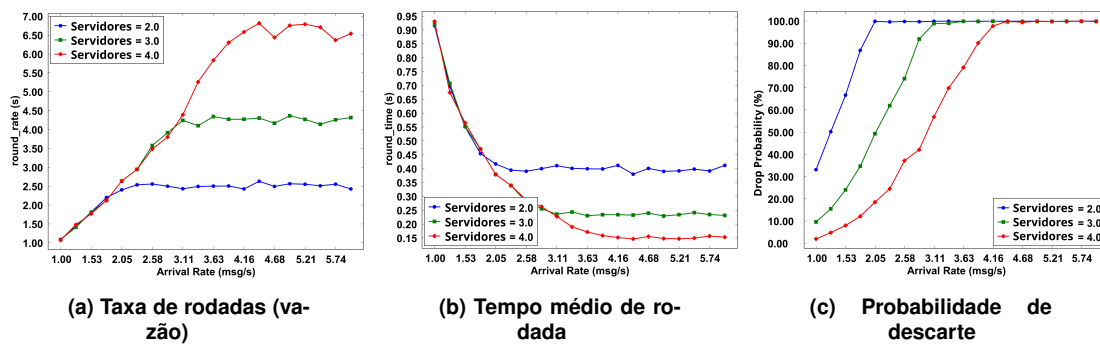


Figura 3. Análise Estacionária do HFL Variando a Taxa de Chegada.

A Figura 3(b) apresenta o tempo médio de rodada em função da taxa de chegada para diferentes quantidades de servidores intermediários, estimado pela métrica ROUND_TIME da Equação 2. Para taxas próximas de 1,0 msg/s, o tempo de rodada assume valores elevados e semelhantes entre as configurações, variando entre aproximadamente 0,92 s e 0,94 s, reduzindo-se com o aumento da taxa de chegada até atingir regime estacionário. Nesse regime, o ROUND_TIME estabiliza-se em torno de 0,40 s com 2 servidores intermediários, enquanto converge para aproximadamente 0,23 s e 0,15 s nos cenários com 3 e 4 servidores, respectivamente. Esses resultados indicam que o aumento do número de servidores intermediários reduz a duração média das rodadas ao mitigar a contenção no nível intermediário.

A Figura 3(c) apresenta a probabilidade de descarte (DP), estimada pela Equação 3. Com 2 servidores intermediários, a DP cresce de aproximadamente 33% para valores

próximos de 100% quando a taxa de chegada varia de 1,0 até cerca de 2,1 msg/s, permanecendo saturada a partir desse ponto. Para 3 servidores, a probabilidade de descarte evolui de cerca de 10% para aproximadamente 100% à medida que a taxa de chegada se aproxima de 3,1 msg/s, enquanto, com 4 servidores intermediários, o crescimento é mais gradual, atingindo valores próximos de 100% apenas para taxas superiores a 4,2 msg/s. O deslocamento das curvas ao longo do eixo horizontal indica que o aumento do número de servidores intermediários amplia a região operacional antes da saturação por descarte, caracterizando um regime no qual a agregação intermediária passa a operar sem contribuições suficientes para a progressão das rodadas. De forma geral, os resultados indicam que a ampliação do nível intermediário aumenta a capacidade de processamento do sistema e adia a saturação da arquitetura sob cargas mais elevadas.

5.2. Estudo de Caso 2 - Impacto da Probabilidade de Falha no Desempenho do HFL

Esta subseção apresenta os resultados de desempenho do modelo analítico a partir das métricas `ROUND_RATE` e `ROUND_TIME`, por refletirem de forma mais direta o impacto da probabilidade de falha sobre a progressão temporal das rodadas. A Figura 4 analisa o impacto da probabilidade de falha (`Falha`) sobre essas métricas em função da taxa de chegada. Na Figura 4(a), a taxa de conclusão de rodadas, estimada pela métrica `ROUND_RATE` da Equação 1, cresce de aproximadamente 1,0 para valores próximos de 10,0 rodadas por segundo quando `Falha = 0.1` e a taxa de chegada varia de 1,0 a cerca de 5,8 msg/s. Com `Falha = 0.5`, o `ROUND_RATE` evolui de cerca de 0,75 para aproximadamente 6,7 rodadas por segundo, enquanto, no cenário com `Falha = 0.9`, a taxa aumenta de aproximadamente 0,55 para valores próximos de 4,6 rodadas por segundo. O deslocamento consistente das curvas ao longo do eixo indica que o aumento da probabilidade de falha reduz a vazão do sistema em todo o intervalo analisado, limitando a taxa de conclusão de rodadas.

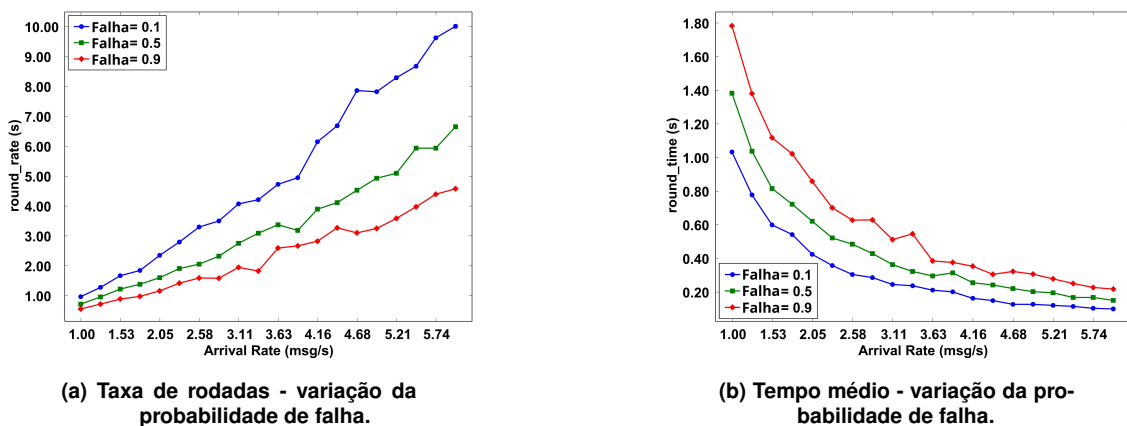


Figura 4. Desempenho do HFL sob Diferentes Prob. de Falha.

A Figura 4(b) apresenta o tempo médio de rodada em função da taxa de chegada para diferentes valores de `wfail`, obtido pela métrica `ROUND_TIME` da Equação 2. Para `wfail = 0.1`, o tempo de rodada decresce de aproximadamente 1.03 s para valores próximos de 0.10 s quando a taxa de chegada varia de 1.0 a cerca de 5.8 msg/s. Com `wfail = 0.5`, observa-se redução de aproximadamente 1.38 s para valores em torno de 0.15 s ao longo do mesmo intervalo. No cenário com `wfail = 0.9`, o `ROUND_TIME` assume valores mais elevados em toda a faixa analisada, decrescendo de cerca de 1.78

s para aproximadamente 0.22 s. Em todas as configurações, o tempo médio de rodada apresenta comportamento decrescente com o aumento da taxa de chegada, mantendo-se sistematicamente maior para valores mais elevados de w_{fail} , indicando que o aumento da probabilidade de falha amplia a duração das rodadas federadas. Portanto, o aumento da probabilidade de falha compromete diretamente a progressão das rodadas, reduzindo a vazão do sistema e ampliando o tempo necessário para conclusão do ciclo federado.

5.3. Estudo de Caso 3 - Análise Transiente do Desempenho do HFL.

Esta subseção apresenta os resultados da análise transiente do modelo analítico, descrevendo a evolução do sistema ao longo do tempo desde o estado inicial até a estabilização do desempenho [Araújo et al. 2025]. A Figura 5 analisa a evolução temporal da métrica ROUND_RATE sob diferentes configurações do sistema. Na Figura 5(a), com 2 servidores intermediários, a taxa de rodadas cresce nos instantes iniciais, atingindo aproximadamente 2,5 rodadas por segundo em menos de 1 s, mantendo-se posteriormente em regime quase estacionário, com pequenas oscilações. Com 3 servidores, a taxa converge para valores próximos de 4,3 rodadas por segundo após cerca de 1 s, enquanto, no cenário com 4 servidores intermediários, o crescimento é mais acentuado, alcançando aproximadamente 6,8 rodadas por segundo e permanecendo superior ao longo do intervalo analisado. Os resultados mostram que o número de servidores intermediários define o nível estacionário da taxa de rodadas, sem impacto significativo sobre o tempo de estabilização do sistema.

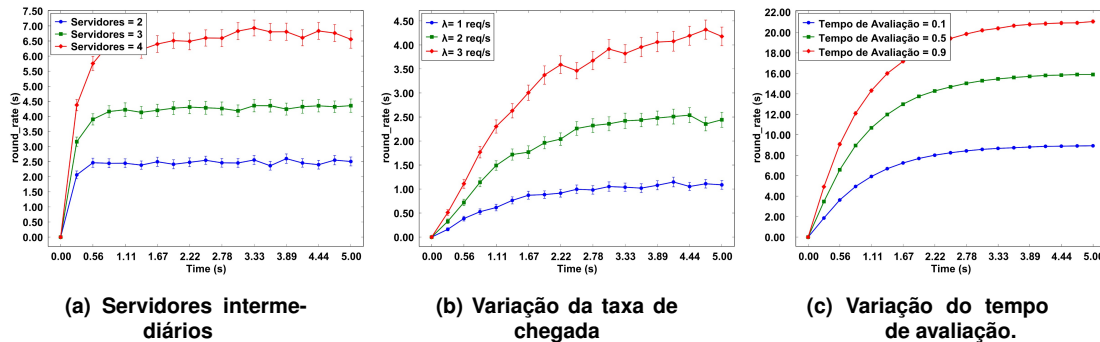


Figura 5. Análise transiente focando na métrica de taxa de rodadas.

A Figura 5(b) apresenta a evolução transiente da taxa de rodadas para diferentes taxas de chegada (l). Para $l=1$ req/s, a taxa de rodadas cresce gradualmente até valores próximos de 1.1 rodadas por segundo. Com $l=2$ req/s, o sistema converge para valores em torno de 2.5 rodadas por segundo, enquanto para $l=3$ req/s a taxa de rodadas atinge valores próximos de 4.2 rodadas por segundo. Em todas as configurações, observa-se um crescimento monotônico seguido por estabilização, indicando que o regime estacionário é alcançado após um curto intervalo transiente, sendo o nível final determinado pela intensidade da carga de chegada.

A Figura 5(c) apresenta a evolução transiente da taxa de rodadas para diferentes tempos de avaliação, entendidos no modelo como o intervalo associado à fase de validação e coordenação após a agregação. Embora tempos de avaliação maiores representem maior permanência nessa etapa, observa-se um aumento da taxa transiente devido à maior estabilização do fluxo de rodadas, reduzindo variações durante a fase inicial de operação. Com tempo de avaliação igual a 0.1 s, a taxa de rodadas converge para valores próximos de 9 rodadas por segundo. Para tempo de avaliação igual a 0.5 s, observa-se estabilização

em torno de 16 rodadas por segundo, enquanto no cenário com tempo de avaliação igual a 0.9 s a taxa de rodadas atinge valores próximos de 21 rodadas por segundo. As curvas evidenciam que o tempo de avaliação influencia diretamente o nível estacionário da taxa de rodadas, mantendo comportamento transiente semelhante entre as configurações e diferindo principalmente no valor de regime permanente. Esses resultados mostram que os parâmetros operacionais alteram principalmente o nível de desempenho em regime estacionário, mais do que o tempo necessário para a estabilização inicial do sistema.

6. Conclusão

Este trabalho apresentou um modelo analítico em Redes de Petri Estocásticas para descrever o comportamento temporal do HFL com múltiplos níveis de agregação. O modelo representa a progressão das rodadas, incluindo admissão de clientes, conectividade, agregação intermediária e coordenação central, conectando a estrutura do sistema ao seu comportamento no tempo. A avaliação considerou análises estacionárias e transientes, usando tempo médio de rodada, taxa de conclusão e probabilidade de descarte. Nos estudos de caso, aumentar os servidores intermediários de dois para quatro reduziu o tempo médio de rodada em cerca de 62% (0,40 s para 0,15 s) e elevou a taxa máxima em torno de 160% (2,5 para 6,5 rodadas/s). Em cargas altas, taxas de chegada elevadas e falhas anteciparam a saturação, levando o descarte próximo de 100%. A análise transiente indicou convergência para o regime estacionário em menos de 1 s, mas com níveis finais de desempenho variando conforme número de servidores, taxa de chegada e tempos do ciclo federado. No geral, o modelo permite estimar previamente vazão, tempo de rodada e descarte sem novas execuções experimentais, apoiando dimensionamento e evitando saturação. Como continuidade, pretende-se incluir seleção adaptativa de clientes, controle dinâmico de cortes para reduzir os descartes e o escalonamento automático dos servidores intermediários.

Referências

- Araújo, I., Silva, L. G., Brito, C., Min, D., Lee, J.-W., Nguyen, T. A., Leão, E., and Silva, F. A. (2025). Dds-p: Stochastic models based performance of iot disaster detection systems across multiple geographic areas. *ICT Express*, 11(1):34–40.
- Cho, S., Lim, S., and Lee, J. (2024). Drl-enabled hierarchical federated learning optimization for data heterogeneity management in multi-access edge computing. *IEEE Access*.
- DP, M. K., NagaSireesha, J., Venkatesh, B., Siddiqua, A., et al. (2023). Hierarchical federated learning-based method for privacy-preserving in the healthcare environment. In *2023 International Conference on Evolutionary Algorithms and Soft Computing Techniques (EASCT)*, pages 1–5. IEEE.
- Farajzadeh, A., Yadav, A., and Yanikomeroglu, H. (2025). Federated learning in ntn: Design, architecture, and challenges. *IEEE Communications Magazine*, 63(6):26–33.
- Fé, I., Silva, L. G., Soares, A., Silva, F. A., Mei, A., Rego, P. A., Choi, E., Nguyen, T. A., Lee, J. W., and Min, D. (2024). Resilient and efficient microservices: Stochastic modeling and quantification of energy consumption and recovery times. In *GLOBECOM 2024-2024 IEEE Global Communications Conference*, pages 5307–5312. IEEE.
- Kumari, A., Shukla, D., Datt, R., Patel, A., Kumar, P., Kumar, S., Katarmal, U., et al. (2024). S-evpf: A secure and decentralized ev prediction framework using federated

- learning. In *2024 Eighth International Conference on Parallel, Distributed and Grid Computing (PDGC)*, pages 658–663. IEEE.
- Li, Z., Chen, Z., Wei, X., Gao, S., Ren, C., and Quek, T. Q. (2022). Hpfl-cn: Communication-efficient hierarchical personalized federated edge learning via complex network feature clustering. In *2022 19th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pages 325–333. IEEE.
- Liu, L., Zhang, J., Song, S., and Letaief, K. B. (2020). Client-edge-cloud hierarchical federated learning. In *ICC 2020-2020 IEEE international conference on communications (ICC)*, pages 1–6. IEEE.
- Manju, A., Kumar, C. P., Jegan, J., Jagadeeshan, D., and Nunna, S. K. (2024). Hierarchical federated learning with fog nodes: Enhancing efficiency in smart city networks. In *2024 OITS International Conference on Information Technology (OCIT)*, pages 749–753. IEEE.
- Ooi, M. P.-L., Sohail, S., Huang, V. G., Hudson, N., Baughman, M., Rana, O., Hinze, A., Chard, K., Chard, R., Foster, I., et al. (2023). Measurement and applications: Exploring the challenges and opportunities of hierarchical federated learning in sensor applications. *IEEE Instrumentation & Measurement Magazine*, 26(9):21–31.
- Sharma, P., Sharma, S. K., and Dani, D. (2025). Edge-assisted federated learning for anomaly detection in diverse iot network. *International Journal of Information Technology*, 17(5):3035–3045.
- Silva, B., Matos, R., Callou, G., Figueiredo, J., Oliveira, D., Ferreira, J., Dantas, J., Lobo, A., Alves, V., and Maciel, P. (2015). Mercury: An integrated environment for performance and dependability evaluation of general systems. In *45th dependable systems and networks conference (DSN)*, pages 1–4.
- Silva, L. G., Cardoso, I., Brito, C., Barbosa, V., Nogueira, B., Choi, E., Nguyen, T. A., Min, D., Lee, J. W., and Silva, F. A. (2023). Urban advanced mobility dependability: A model-based quantification on vehicular ad hoc networks with virtual machine migration. *Sensors*, 23(23):9485.
- Siriwardhana, Y., Porambage, P., Liyanage, M., Marchal, S., and Ylianttila, M. (2024). Shield-secure aggregation against poisoning in hierarchical federated learning. *IEEE Transactions on Dependable and Secure Computing*.
- Statista (2025). Number of internet of things (iot) devices connected worldwide from 2022 to 2023, with forecasts from 2024 to 2034. Accessed: 2025-02-04; Forecast shows increase from approximately 19.8 billion in 2025 to 40.6 billion by 2034.
- Wen, J., Zhang, Z., Lan, Y., Cui, Z., Cai, J., and Zhang, W. (2023). A survey on federated learning: challenges and applications. *International Journal of Machine Learning and Cybernetics*, 14(2):513–535.
- Zhang, H., Bosch, J., and Olsson, H. H. (2020). Federated learning systems: Architecture alternatives. In *2020 27th Asia-Pacific Software Engineering Conference (APSEC)*, pages 385–394.
- Zhao, T., Li, F., and He, L. (2022). Drl-based joint resource allocation and device orchestration for hierarchical federated learning in noma-enabled industrial iot. *IEEE Transactions on Industrial Informatics*, 19(6):7468–7479.