

# Rumo à Blockchain Adaptiva para a Plataforma Hyperledger Fabric

Ericksulino Manoel de Araújo Moura<sup>1</sup>, Felipe Santiago Gama<sup>1</sup>,  
Glauber Dias Gonçalves<sup>1</sup>, Allan Edgard Silva Freitas<sup>2</sup>, André Soares<sup>1</sup>

<sup>1</sup>Universidade Federal do Piauí (UFPI), Teresina, Piauí, Brasil

<sup>2</sup>Instituto Federal da Bahia (IFBA), Salvador, Bahia, Brasil

{ericksulino, felipes4ntiago, ggoncalves, andre.soares}@ufpi.edu.br

allan@ifba.edu.br

**Abstract.** *Blockchain technology provides secure and decentralized solutions for data management. However, there are challenges for these solutions to perform efficiently in scenarios with variations in the transaction input rate. In this sense, adaptive strategies have been proposed aiming at near-optimal performance in throughput and latency, considering these variations. In this paper, we analyze the state-of-the-art aHLF and FabMAN strategies for dynamically adjusting block size and timeout in the context of the Hyperledger Fabric permissioned blockchain. We conduct performance analysis via experiments on a blockchain network and compare these strategies. The results show that aHLF responds more efficiently to high input rate scenarios, while FabMAN performs better in low input rate scenarios. Therefore, we analyze how a hybrid approach that explores both strategies can result in an effective adaptive blockchain for different scenarios.*

**Resumo.** *A tecnologia Blockchain proporciona soluções seguras e descentralizadas para o gerenciamento de dados. Porém, há desafios para que essas soluções tenham desempenho eficiente em cenários com variações na taxa de entrada de transações. Nesse sentido, estratégias adaptativas vem sendo propostas visando desempenho quase ótimo em vazão e latência, considerando essas variações. Neste artigo, analisamos as estratégias estado da arte aHLF e FabMAN para ajustar dinamicamente o tamanho e o tempo limite do bloco no contexto da blockchain permissionada Hyperledger Fabric. Conduzimos análises de desempenho via experimentos em uma rede blockchain e comparamos essas estratégias. Os resultados mostram que aHLF responde com maior eficiência aos cenários de alta taxa de entrada, ao passo que FabMAN tem melhor desempenho em cenários com taxa de entrada baixa. Por conseguinte, analisamos como uma abordagem híbrida que explora ambas as estratégias pode resultar em uma blockchain adaptativa eficaz para diferentes cenários.*

## 1. Introdução

Blockchain é uma tecnologia disruptiva, especialmente para os setores da indústria e serviços, pois oferece uma soluções robustas para o gerenciamento seguro e descentralizado de dados [Xu et al. 2019]. Por meio de blockchain, é possível o registro seguro

de transações entre partes distintas, tais como indivíduos ou organizações, mesmo na ausência de confiança mútua. Isto se tornou possível a partir da evolução e unificação de tecnologias como criptografia assimétrica, protocolos de consenso distribuído via comunicação par a par, e uma estrutura de blocos encadeados por resumos criptográficos (i.e. *hashes*) que são a essência da blockchain. Essas tecnologias possibilitam prover ao registro de transações imutabilidade, auditabilidade e consistência [Greve et al. 2018].

Em redes ditas públicas, onde em geral o consenso das transações a serem registradas é mediado por uma criptomoeda (e.g. Bitcoin e Ethereum), quaisquer nós podem participar da rede provendo cenários de larga escala onde o desempenho é prejudicado. Mas, em cenários envolvendo consórcios com um conjunto limitado de participantes é possível utilizar protocolos clássicos de consenso, como PBFT [Castro et al. 1999], RAFT [Ongaro and Ousterhout 2014] etc., os quais requerem que haja um gerenciamento dos participantes (i.e. *membership*) mas possibilitam um volume maior de transações, provendo melhor desempenho.

Esses cenários são ditos de blockchain em redes privadas (ou permissionadas) [Androulaki et al. 2018]. Dentre as plataformas para blockchain permissionada, o Hyperledger Fabric [Androulaki et al. 2018] é uma das plataformas para blockchains permissionadas mais populares atualmente<sup>1</sup> com recursos para a implantação de uma infraestrutura de rede privada entre organizações e desenvolvimento de aplicações no topo dessa rede. Nesse caso, os participantes da rede formam um consórcio e arcam com o custo da infraestrutura, buscando ganhos no desempenho em relação às redes blockchain públicas.

Desempenho é crucial para garantir que as plataformas blockchain sejam eficazes em aplicações industriais, onde é necessário registrar transações em um curto intervalo de tempo, garantindo alta vazão e baixa latência. A plataforma Hyperledger Fabric permite utilizar diferentes protocolos clássicos de consenso, definindo um canal como meio que nós previamente inscritos registram transações de acordo com a configuração do mesmo. Além do protocolo de consenso utilizado, parâmetros como o tamanho do bloco e o tempo limite do bloco são importantes influenciando no volume de transações registradas e interferindo na vazão e latência do registro de transações [Liu et al. 2021, Baliga et al. 2018].

Neste trabalho, investigamos o impacto das configurações desses parâmetros para diferentes taxas de transações por segundo (tps) no desempenho de redes Hyperledger Fabric. Realizamos uma análise comparativa entre duas estratégias estado da arte de ajuste dinâmico destes parâmetros: FabMAN e aHLF. A estratégia *FabMAN* [Roy and Ghosh 2024] foi desenvolvida especialmente para o Hyperledger Fabric, enquanto *aHLF* foi originada da abordagem adaptativa aPBFT [de Sá et al. 2013] para o PBFT, aqui adaptada para o Hyperledger Fabric. Ambas estratégias propõem diferentes abordagens no ajuste dos parâmetros tamanho e tempo limite do bloco.

Para conduzir esta investigação, propomos uma metodologia de ajuste dinâmico, onde aplicamos estratégias específicas para otimizar o desempenho da rede em ciclos sucessivos. Em cada ciclo, medimos métricas de desempenho, como latência e vazão, e ajustamos os parâmetros de configuração da rede, como o tamanho do bloco e o tempo limite do bloco, conforme necessário para melhorar o desempenho no ciclo seguinte. Este trabalho contribui com (1) a análise de desempenho dessas estratégias, propondo uma

---

<sup>1</sup><https://www.ibm.com/topics/hyperledger>

metodologia para implementá-las em redes Hyperledger Fabric e (2) uma proposta de estratégia híbrida para adaptação eficiente dos parâmetros, visando melhorar rapidamente latências e vazões.

Observa-se que a estratégia aHLF se destaca nos cenários com taxa de entrada de transações altas, i.e., taxas de 40 TPS e superiores, alcançando a melhor latência (186 ms) e vazão (37.62 TPS) no quinto ciclo de adaptação. A estratégia FabMAN, por sua vez, apresenta vantagens em cenários iniciais com taxas baixa (10 TPS) e média (20 TPS), se adaptando mais rapidamente que aHLF nesses cenários ao obter as melhores latências (166 ms e 152 ms) e vazões (9.80 TPS e 19.41 TPS) respectivamente no terceiro e quarto ciclos de adaptação, mas enfrentando instabilidades refletidas em picos de altas latências e baixa vazão no cenário com taxa de entrada alta. Por conseguinte, analisamos a oportunidade de uma estratégia híbrida que utiliza ambas as estratégias, selecionando uma delas a depender do cenário de taxa de entrada, e observamos que é possível alcançar rapidamente esse desempenho quase ótimo, i.e., já no terceiro ciclo de adaptação, mesmo em diferentes cenários.

As próximas seções estão organizadas da seguinte forma: A Seção 2 apresenta uma revisão dos trabalhos relacionados a avaliação de desempenho de blockchain. A Seção 3 descreve as estratégias aHLF (baseado no aPBFT) e FabMAN, explicando os fundamentos e cálculos utilizados para prever os parâmetros *BatchSize* e *BatchTimeout*. A Seção 4 detalha o método da avaliação de desempenho. A Seção 5 apresenta os principais resultados obtidos e suas análises. Por fim, a Seção 6 apresenta nossas conclusões.

## 2. Trabalhos Relacionados

A avaliação de desempenho de plataformas blockchain e mecanismos de consenso possibilita compreender o impacto de parâmetros de configuração nas métricas de latência, vazão e eficiência do sistema. Os trabalhos descritos a seguir abordam desde análises experimentais tradicionais até propostas de ajuste dinâmico de parâmetros para otimização do Hyperledger Fabric.

O artigo de [de Sá et al. 2013] propõe o aPBFT para ajustar dinamicamente o *BatchSize* (BS) e o *BatchTimeout* (BT) no protocolo PBFT, melhorando o desempenho em sistemas distribuídos baseada na teoria de controle por feedback, com o objetivo de melhorar o desempenho do protocolo de consenso PBFT em ambientes dinâmicos. Em nosso estudo, baseamos o aHLF na estratégia proposta pelo autor, mas aplicando-a ao Hyperledger Fabric, e comparamos com o FabMAN para oferecer uma solução mais flexível e adaptativa para diferentes cenários de carga.

Por outro lado, [Shalaby et al. 2020] investigam como personalizar o comportamento da rede blockchain de acordo com os requisitos das aplicações, focando no desempenho do Hyperledger Fabric (HLF). O estudo analisou o impacto de parâmetros como o tempo limite do bloco, o tamanho do bloco e o número de pares endossantes em sete cenários experimentais. Em comparação, nosso artigo vai além ao comparar as estratégias aHLF e FabMAN, considerando o ajuste dinâmico de *BatchSize* e *BatchTimeout* e propondo uma estratégia híbrida para otimizar o desempenho da rede em diferentes cenários de carga.

No contexto de falhas transacionais, [Chacko et al. 2021] analisam as causas de erros no Hyperledger Fabric, definindo suas categorias e desenvolvendo o HyperLedger-

Lab para testar como parâmetros como tamanho do bloco e políticas de endosso influenciam essas falhas. Em contraste, nosso artigo não se foca apenas nas falhas, mas explora o ajuste dinâmico de parâmetros para otimizar a rede, com foco nas estratégias aHLF e FabMAN.

O estudo de [Liu et al. 2021] propõe uma estratégia adaptativa para ajustar dinamicamente o *BatchSize* (BS) no Hyperledger Fabric, voltada para aplicações de IoT, identificando gargalos na configuração estática da rede. No entanto, o estudo foca apenas no ajuste do *BatchSize*, enquanto nosso artigo expande a análise para incluir ambos os parâmetros (BS e BT) e propõe uma estratégia híbrida, proporcionando uma adaptação mais eficaz e flexível em diversos cenários de carga.

Em [Saeed et al. 2022], os autores analisam o desempenho do Hyperledger Fabric em um estudo de caso de E-votação, focando em latência, vazão e parâmetros como tamanho de bloco, taxas de transação e número de organizações. Embora o estudo ofereça recomendações para otimizar o sistema em cenários com muitas transações simultâneas, nosso artigo vai além, explorando o ajuste dinâmico de parâmetros e propondo uma estratégia híbrida com aHLF e FabMAN para otimizar o desempenho da rede em diferentes cenários operacionais.

Já [Wai and Thein 2023] analisam o Hyperledger Fabric como um framework de blockchain permissionado, destacando a influência dos parâmetros de configuração, como o tamanho do bloco e o tempo limite de lote, no desempenho do sistema. Utilizando a ferramenta Hyperledger Caliper, os autores medem a vazão de transações e a latência média. Em comparação, nosso artigo expande essa análise, ajustando dinamicamente tanto o *BatchSize* quanto o *BatchTimeout* e propondo uma estratégia híbrida entre aHLF e FabMAN para otimizar o desempenho em diferentes cenários de carga.

Com o foco em modelagem analítica, [Silva et al. 2023] propõem um modelo de Redes de Petri Estocásticas (SPN) para analisar o desempenho do Hyperledger Fabric, focando em configurações de blockchain e capacidade computacional. O estudo mostra como a modelagem ajuda a identificar configurações, como o tamanho do bloco, que afetam a latência devido ao enfileiramento de requisições. Em contraste, nosso estudo ajusta diretamente uma rede blockchain, modificando dinamicamente os parâmetros em tempo real, utilizando as estratégias aHLF e FabMAN para otimizar a latência e vazão.

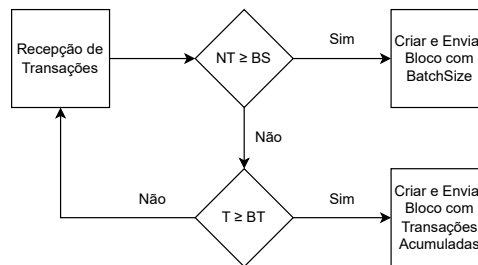
Por fim, [Roy and Ghosh 2024] propõe a abordagem FabMAN para gerenciar o livro-razão em expansão no Hyperledger Fabric, visando gestão de dados na Internet das Coisas (IoT). Os autores apresentam uma estratégia adaptativa que ajusta o “tamanho do bloco” e o “tempo limite do bloco” para otimizar o crescimento do livro-razão. A eficácia da abordagem é validada por meio de simulações e protótipos, mostrando melhorias significativas em comparação com sistemas tradicionais. Em nosso estudo, utilizamos a estratégia do autor para ajustar os parâmetros e comparamos sua eficácia com a do aHLF, propondo uma análise mais ampla e dinâmica das opções de ajuste de parâmetros no desempenho da rede.

Embora a maioria dos estudos sobre o Hyperledger Fabric (HLF) se concentre na análise de parâmetros de configuração e seu impacto no desempenho, poucos exploram a comparação de ajustes dinâmicos desses parâmetros. Este estudo se destaca ao comparar detalhadamente as estratégias aHLF, baseado no aPBFT [de Sá et al. 2013] e Fab-

MAN [Roy and Ghosh 2024], com foco no ajuste dinâmico de *BatchSize* e *BatchTimeout*. Ao avaliar o desempenho da rede em diferentes cenários de carga, nossa análise propõe formas de otimizar esses parâmetros para melhorar a latência e a vazão. A principal contribuição deste trabalho é fornecer diretrizes para escolher a estratégia mais eficiente, levando em consideração o contexto operacional e os requisitos de desempenho da rede.

### 3. Estratégias Adaptativas em Hyperledger Fabric

Nessa seção, apresentamos uma visão geral da blockchain Hyperledger Fabric. A seguir, descrevemos os procedimentos realizados pelas estratégias aHLF e FabMAN para prever os valores de tamanho do bloco abreviado por BS (*batchsize*) e tempo limite do bloco, abreviado por BT (*batchtimeout*). O processamento de transações no Hyperledger Fabric está centrado nos parâmetros *BatchTimeout* (BT) e *BatchSize* (BS), que regulam a formação de blocos.



**Figura 1. Fluxograma de Processamento de Transações no Hyperledger Fabric: Interação entre NT (Número de Transações), BT (*BatchTimeout*), BS (*BatchSize*) e Tempo (T).**

Figura 1 mostra o fluxo de transações para construção de blocos no Hyperledger Fabric, considerando BT e BS. O número de transações acumuladas atinge o valor de *BatchSize* (BS). O tempo de espera para o bloco excede o valor de *BatchTimeout* (BT). Quando uma dessas condições é satisfeita, um bloco é criado e enviado para validação pelos nós da rede. Esse fluxo ilustra como os principais parâmetros influenciam a formação e o envio de blocos na rede. A seguir, descrevemos como as estratégias aHLF e FabMAN realizam previsões e ajustes dinâmicos dos parâmetros BT e BS, com o objetivo de otimizar o desempenho da rede em diferentes cenários.

#### 3.1. Estratégia aHLF

A estratégia aPBFT [de Sá et al. 2013] utiliza uma abordagem adaptativa para ajustar dinamicamente os parâmetros BT e BS durante a execução. Essa adaptação é realizada por meio do monitoramento contínuo do desempenho do protocolo e as condições do sistema, propondo ajustes baseados em métricas coletadas. Nesta seção, apresentamos a estratégia adaptada do aPBFT na forma do aHLF para quaisquer consenso utilizado no HyperLedger Fabric.

O tempo médio de ordenação e execução (*Mean Time to Ordering and Execution*, MTE) é estimado com base na média móvel dos últimos  $W$  intervalos de tempo, conforme a Equação 1. Esse valor é usado como referência para ajustar os parâmetros de tempo e tamanho dos lotes.

$$\text{MTE} = (1 - \alpha) \cdot \text{MTE}_{\text{recente}} + \alpha \cdot \text{TE}_{\text{recente}} \quad (1)$$

onde  $\alpha = \frac{1}{W}$  e TE é o intervalo entre o início da ordenação e a conclusão da execução de um bloco.

O tempo médio entre chegadas de transações (*Mean Time Between Arrivals*, MTA) dado por:

$$\text{MTA} = \frac{t - t_0}{n_{\text{req}}} \quad (2)$$

onde  $t$  é o tempo atual,  $t_0$  o momento específico em que o replica primária assume sua função na rede, e  $n_{\text{req}}$  é o número de transações recebidas até  $t$ .

O ajuste do tamanho do bloco é realizado conforme a Equação 3, garantindo que o tamanho do bloco seja proporcional à taxa de entrada de transações observada:

$$\text{BS} = \left\lceil \frac{\text{MTE}}{\text{MTA}} \right\rceil \quad (3)$$

Já o tempo limite do bloco é ajustado para o valor de MTE, exceto nos casos em que  $\text{MTA} \geq \text{MTE}$ , situação em que BT é definido como zero, indicando que o sistema está subutilizado.

Dessa forma, a estratégia aHLF utiliza a estimativa de MTE e MTA de forma semelhante ao aPBFT e ajusta BS e BT a cada reconfiguração periódica do canal do Hyperledger Fabric. Assim, essa estratégia pode ser empregada para qualquer protocolo de consenso em uso no Hyperledger Fabric.

### 3.2. Estratégia FabMAN

A estratégia adaptativa FabMAN [Roy and Ghosh 2024] utiliza a ideia de *Exponential Weighted Moving Average* (EWMA) para prever os valores de taxa de entrada de transações ( $T_{\text{req}}$ ) e atraso da rede ( $N_{\text{latência}}$ ) com base nas observações anteriores. Essa abordagem ajusta dinamicamente os parâmetros de BS e BT para otimizar o desempenho da rede Hyperledger Fabric. O modelo matemático da estratégia utiliza as seguintes equações discutidas a seguir.

A estimativa da taxa de solicitação de transações ( $T_{\text{req}}$ ) é dada por

$$T_{\text{req}} = \frac{T_{\text{count}}}{BT} \quad (4)$$

onde  $T_{\text{count}}$  é o número de transações no bloco mais recente e  $BT$  é o tempo limite do bloco atual.

A estimativa da latência da rede ( $N_{\text{latência}}$ ) é dada por

$$N_{\text{latência}} = (TS_{\text{recente}} - TS_{\text{recente}-1}) - BT \quad (5)$$

onde  $TS_{\text{recente}}$  é o timestamp do bloco mais recente e  $TS_{\text{recente}-1}$  é o timestamp do bloco anterior.

O ajuste do tempo limite do bloco é dado por

$$BT_{novo} = T_{latência} - N_{latência} \quad (6)$$

onde  $T_{latência}$  é a latência máxima tolerável para a aplicação.

O ajuste do tamanho do bloco considera a seguinte condição. Se  $T_{req}$  exceder uma porcentagem ( $\alpha$ ) então, o tamanho do bloco será dobrado:

$$BS_{novo} = 2 \cdot BS_{atual} \quad (7)$$

Caso contrário, se  $T_{req}$  for menor que  $\frac{\alpha}{2}$ , o tamanho do lote será reduzido pela metade:

$$BS_{novo} = \frac{BS_{atual}}{2} \quad (8)$$

Ainda, caso nenhuma das condições ocorram o tamanho do bloco será mantido. Conforme ilustrado, o FabMAN ajusta dinamicamente os parâmetros para equilibrar vazão e latência, buscando o desempenho ideal do sistema. A aplicação dessas equações permite prever mudanças nas condições da rede e responder proativamente às demandas.

## 4. Metodologia de Avaliação

Esta seção descreve a metodologia utilizada para avaliar o desempenho do Hyperledger Fabric (HLF) sob diferentes configurações de parâmetros. Nossa abordagem envolve a realização de experimentos sistemáticos para analisar os efeitos das variáveis *BatchSize* (BS), *BatchTimeout* (BT) e taxa de transações por segundo (TPS) na latência média das transações.

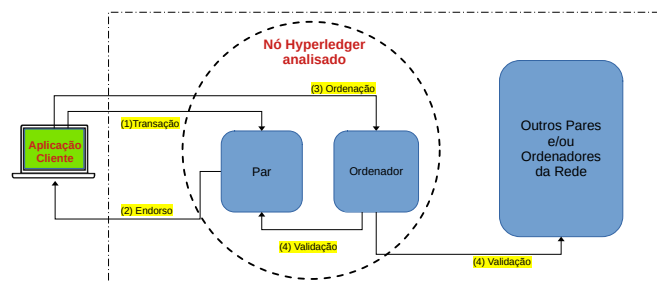
### 4.1. Configuração do Ambiente de Testes

Os experimentos foram realizados em um ambiente de teste configurado com três nós completos do Hyperledger Fabric, utilizando o HLF-PET [Moura et al. 2024] para gerar a carga de transações no sistema. Além disso, foi desenvolvido um código em Go para implementar as estratégias aHLF e FabMAN. Esse código permitiu prever automaticamente os valores dos parâmetros *BatchTimeout* (BT) e *BatchSize* (BS), além de aplicar de forma automatizada as modificações na configuração da rede durante os ciclos de adaptação.

Os nós da rede foram implantados em máquinas virtuais configuradas com especificações de hardware idênticas, cada uma equipada com quatro núcleos físicos, 8 GB de RAM e 100 GB de armazenamento. Essas máquinas executavam os serviços do Hyperledger Fabric, desempenhando as funções de ordenadores e validadores de transações. A comunicação entre os nós foi garantida por uma rede privada configurada para simular um ambiente de produção, com latência e largura de banda controladas. O protocolo de consenso utilizado foi o RAFT.

A Figura 2 ilustra a arquitetura geral do ambiente experimental baseado no Hyperledger Fabric, seguindo o modelo proposto por [Mendonça et al. 2023]. A estrutura inclui os seguintes componentes principais:

- Cliente de teste: Responsável por enviar transações e coletar métricas durante os experimentos.



**Figura 2. Componentes e fluxo de transações na plataforma Hyperledger Fabric.**

- Nós HLF: Cada nó do Hyperledger Fabric (HLF) é composto por um nó ordenador (orderer node), responsável por organizar as transações e gerar os blocos, e um nó par (peer node), que valida e armazena os blocos no ledger.

Para mitigar interferências externas, o ambiente de teste foi isolado, garantindo que os resultados fossem influenciados apenas pelas configurações das estratégias e das cargas de trabalho aplicadas. Este nível de controle assegura a validade e a confiabilidade das análises realizadas neste estudo.

#### 4.2. Processo Experimental

Os experimentos foram conduzidos utilizando cargas de 10 TPS, 20 TPS e 40 TPS, representando cenários de baixa, média e alta carga. Essas cargas foram escolhidas para avaliar como as estratégias se comportam em diferentes volumes de transações, desde otimizações rápidas em baixa carga até a capacidade de adaptação em alta carga. O processo experimental consistiu nas seguintes etapas:

1. **Configuração Inicial:** A configuração inicial da rede foi feita com parâmetros padrão. O *BatchTimeout* (BT) foi configurado para 2 segundos, e o *BatchSize* (BS) foi definido como 10 transações. Esses valores iniciais foram escolhidos para representar um ponto de partida equilibrado para as duas estratégias e permitir uma avaliação comparativa justa.
2. **Submissão de Transações:** Para cada carga fixa de TPS (10, 20 e 40), as transações foram enviadas de forma contínua ao longo de cada ciclo, com a taxa de envio ajustada de acordo com a carga definida. As transações foram geradas e enviadas constantemente durante todo o ciclo de adaptação, permitindo que as estratégias se ajustassem dinamicamente em resposta ao fluxo contínuo de transações, refletindo condições mais realistas de operação da rede blockchain.
3. **Coleta de Métricas:** Após o envio de cada conjunto de transações, métricas de desempenho como latência média e vazão (TPS alcançado) foram coletadas automaticamente. Para isso, foi desenvolvido um script em Go que monitora e registra as métricas em tempo real durante o experimento, assegurando a precisão e a consistência dos dados coletados para análise posterior.
4. **Cálculo dos Parâmetros Previstos:** As estratégias aHLF e FabMAN foram utilizadas para prever novos valores para o *BatchTimeout* (BT) e *BatchSize* (BS) com base nas métricas coletadas, utilizando suas respectivas metodologias de ajuste dinâmico. O cálculo das previsões seguiu as fórmulas e algoritmos descritos na Seção 3, levando em consideração os dados de latência e vazão obtidos em cada ciclo para ajustar os parâmetros de forma eficiente.



5. **Aplicação Automática dos Ajustes:** Os valores previstos de BT e BS foram aplicados automaticamente na rede a cada iteração. Isso garantiu que as estratégias pudessem ajustar dinamicamente os parâmetros conforme necessário, com base nos dados de desempenho mais recentes, proporcionando uma adaptação contínua e precisa às mudanças nas condições de carga.
6. **Ciclos de Adaptação:** O processo foi repetido por 10 ciclos consecutivos para cada carga fixa de TPS, com 100 transações enviadas por iteração. Esse número de ciclos foi escolhido porque, no contexto de blockchain, a adaptação ocorre com base no número de transações agrupadas em blocos. Dessa forma, 10 ciclos permitiram uma avaliação suficiente da capacidade das estratégias de se adaptarem e ajustarem os parâmetros ao longo do tempo, em resposta a diferentes condições de carga, sem sobrecarregar o processo de análise.

Este procedimento estruturado possibilitou a análise do impacto dos ajustes dinâmicos dos parâmetros *BatchTimeout* e *BatchSize* em diversos cenários de carga. Através dessa abordagem, foi possível avaliar a eficácia das estratégias aHLF e FabMAN em reduzir a latência e otimizar o desempenho da rede blockchain, ajustando-se de maneira eficiente às variações nas condições operacionais ao longo dos ciclos de adaptação.

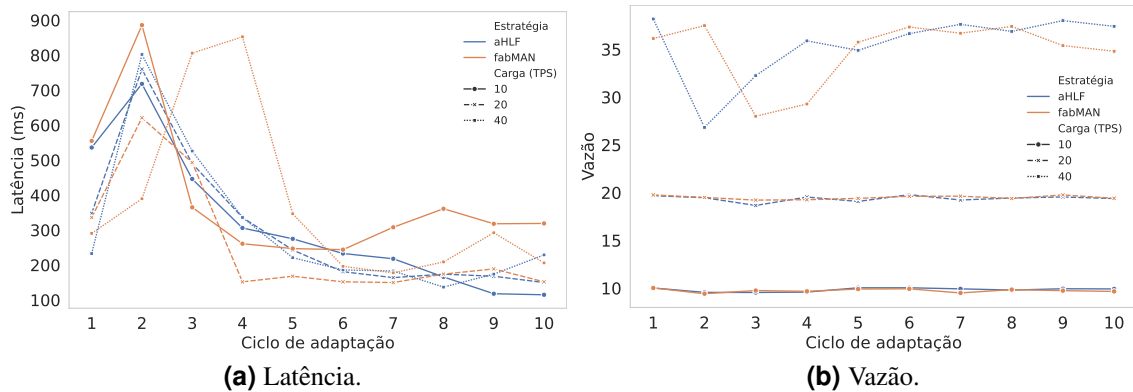
## 5. Resultados

Os resultados analisam o desempenho das estratégias aPBFT e FabMAN em diferentes cenários, considerando parâmetros como *BatchSize* (BS), *BatchTimeout* (BT), TPS e latência média. A Subseção 5.1 explora a relação entre latência e vazão, enquanto a Subseção 5.2 detalha a evolução dos parâmetros BS e BT. Por fim, a Subseção 5.3 apresenta uma análise preliminar de uma estratégia híbrida combinando o aHLF e o FabMAN para otimizar o desempenho da rede.

### 5.1. Desempenho das Estratégias: Análise da Latência Média e Vazão

Nesta subseção, avaliamos o desempenho das estratégias aHLF e FabMAN no Hyperledger Fabric, considerando duas métricas essenciais: latência média e vazão. A latência representa o tempo médio necessário para que uma transação seja confirmada e incluída em um bloco, sendo um fator crítico para aplicações que exigem tempos de resposta rápidos. A vazão, por sua vez, mede a quantidade de transações processadas por segundo, sendo um indicador fundamental da capacidade da rede.

A Figura 3a ilustra a variação da latência média em diferentes configurações de *BatchSize* (BS), *BatchTimeout* (BT) e *Taxa de Transações por Segundo*. Na carga de 10 TPS, ambas as estratégias iniciam com latências próximas de 540 ms e registram um pico no segundo ciclo, com 890 ms para o FabMAN e 800 ms para o aHLF. Após esse aumento, o aHLF reduz de forma consistente, encerrando em 110 ms no último ciclo, representando uma redução de 79,6% em relação ao valor inicial. O FabMAN também reduz a latência, mas com maior instabilidade, finalizando em 310 ms, uma queda de 42,6%. Vale destacar que o FabMAN se adapta mais rapidamente que o aHLF, alcançando a latência mais baixa em menos ciclos, apesar das oscilações. Em 20 TPS, ambas as estratégias atingem picos no segundo ciclo (860 ms para aHLF e 610 ms para FabMAN), mas seguem padrões distintos nos ciclos seguintes. O aHLF reduz de forma mais previsível e contínua, encerrando em 140 ms, com uma redução de 58,8%. O FabMAN, apesar de oscilar mais ao



**Figura 3. Comparação de Latência média e Vazão por Ciclo de adaptação entre as Estratégias aHLF e FabMAN.**

longo dos ciclos, também converge para 140 ms ao final, alcançando a mesma redução percentual. Em ambos os cenários (10 TPS e 20 TPS), o FabMAN se adapta mais rapidamente e atinge as melhores latências mais cedo, com o aHLF apresentando uma redução mais gradual, mas estável, ao longo dos ciclos.

Na carga de 40 TPS, o aHLF registra um pico de 800 ms no segundo ciclo antes de iniciar uma queda gradual até 160 ms no oitavo ciclo, subindo levemente para 210 ms no final, totalizando uma redução de 8,7%. Já o FabMAN apresenta oscilações mais acentuadas ao longo dos ciclos, atingindo um pico de 860 ms no quarto ciclo. Apesar de reduzir a latência para 180 ms no sétimo ciclo, o algoritmo não consegue manter essa melhora e volta a subir nos ciclos finais, encerrando em 200 ms, representando uma redução de apenas 28,6% em relação ao valor inicial. Esse comportamento indica que o FabMAN não é tão eficiente para cenários de alta carga, apresentando dificuldades em manter um desempenho estável, com variações expressivas que podem comprometer a previsibilidade do sistema. Assim, uma abordagem híbrida que combine a estabilidade do aHLF com a rápida adaptação do FabMAN pode oferecer um desempenho mais equilibrado e eficiente para o Hyperledger Fabric em diferentes cenários operacionais.

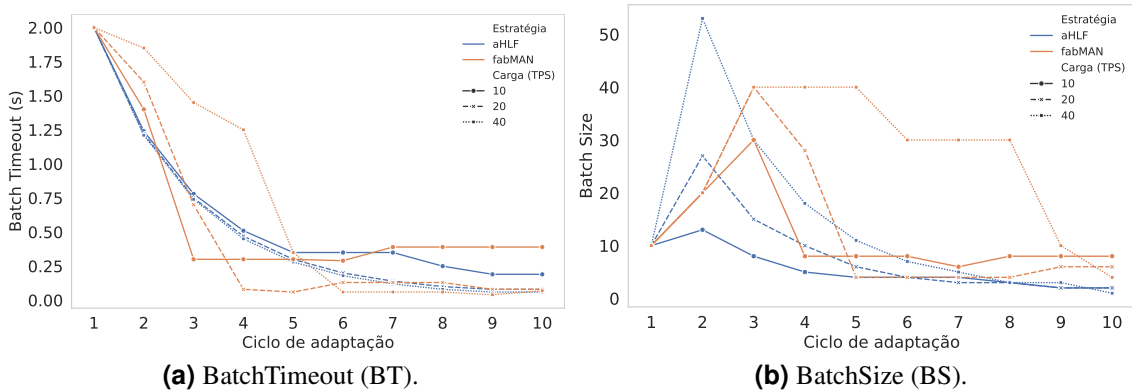
A Figura 3b ilustra a variação da vazão alcançada em cada cenário testado (10, 20 e 40 TPS), destacando as diferenças entre as estratégias e seus padrões de ajuste. Na análise da vazão, observa-se que, na carga de 10 TPS, ambas as estratégias iniciam com valores próximos, sendo 9 TPS para o aHLF e 8,8 TPS para o FabMAN. O aHLF mantém um comportamento consistente ao longo dos ciclos de adaptação, estabilizando próximo de 9,5 TPS nos ciclos finais. Já o FabMAN apresenta uma melhora gradual, ultrapassando 9 TPS no terceiro ciclo e mantendo-se nesse patamar até o final, embora sem atingir a mesma estabilidade do aHLF. Para a carga de 20 TPS, o aHLF inicia com cerca de 18 TPS, demonstrando uma melhora gradual e alcançando aproximadamente 19,5 TPS no quarto ciclo, onde estabiliza. O FabMAN, por sua vez, começa em torno de 17,5 TPS, apresentando um aumento inicial até atingir 18,8 TPS no terceiro ciclo, mas oscilando nos ciclos seguintes e encerrando com uma vazão inferior ao do aHLF.

Na carga de 40 TPS, o aHLF inicia com aproximadamente 32 TPS, aumentando gradativamente até atingir 36 TPS no quinto ciclo, estabilizando nesse valor até o final. O FabMAN começa com cerca de 30 TPS e apresenta oscilações significativas, alcançando

34 TPS no terceiro ciclo, mas sofrendo quedas nos testes seguintes e estabilizando em torno de 32 TPS no final. De forma geral, o aHLF apresentou maior consistência e estabilidade em todas as cargas avaliadas, adaptando-se melhor às condições e mantendo uma vazão crescente e estável. O FabMAN, embora tenha mostrado melhorias em algumas situações, apresentou maior instabilidade, especialmente em cargas mais altas, o que limita sua eficácia em contextos de maior demanda. Essa diferença de desempenho pode ser explicada pela capacidade do aHLF de adaptar-se de forma mais contínua e eficiente ao longo dos ciclos, o que resultou em uma vazão mais consistente, justificando seu desempenho quase ótimo nas condições de carga avaliadas.

## 5.2. Adaptação dos Parâmetros BS e BT

Nesta subseção, analisamos como as estratégias aHLF e FabMAN ajustaram dinamicamente os parâmetros *BatchTimeout* (BT) e *BatchSize* (BS) ao longo dos ciclos de adaptação realizados. Esses ajustes são fundamentais para a formação de blocos e têm impacto direto no desempenho e eficiência da rede blockchain.



**Figura 4. Evolução dos parâmetros *BatchTimeout* (BT) e *BatchSize* (BS) ao longo dos ciclos de adaptação para as estratégias aHLF e FabMAN.**

Os ajustes de *BatchTimeout* (BT) (Figura 4a) mostram que o FabMAN apresenta reduções mais abruptas, enquanto o aHLF segue uma abordagem gradual e estável. Em 10 TPS, o FabMAN reduz o BT até 0,26 segundos no ciclo 3 e mantém-se estável até um aumento para 0,40 no ciclo 7, enquanto o aHLF reduz progressivamente, finalizando em 0,24 segundos. Em 20 TPS, ambas as estratégias convergem para 0,10 segundos no final, mas o FabMAN oscila levemente antes da estabilização. Já em 40 TPS, o FabMAN atinge 1,25 segundos no ciclo 4 antes de cair para 0,10 segundos, enquanto o aHLF mantém uma queda constante, finalizando no mesmo valor.

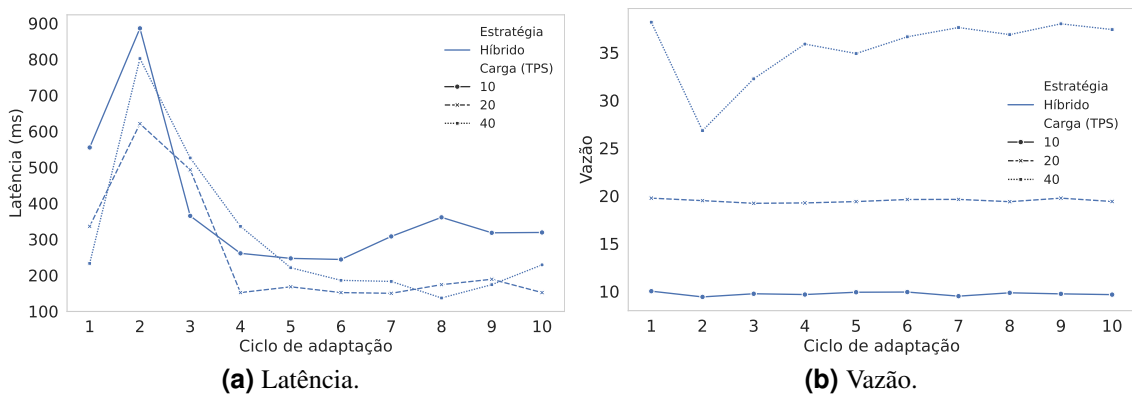
Nos ajustes de *Batch Size* (BS) (Figura 4b), o FabMAN adota uma abordagem mais agressiva no início, aumentando o BS rapidamente antes de reduzi-lo de forma brusca, enquanto o aHLF ajusta os valores de maneira mais progressiva e estável. Em 10 TPS, o FabMAN sobe para 30 no terceiro ciclo, mas cai para 8 no quarto e mantém esse valor, enquanto o aHLF reduz gradativamente até 2. Em 20 TPS, o FabMAN atinge 40 no ciclo 3 e depois reduz para 5, estabilizando-se em 6 nos ciclos finais, enquanto o aHLF sobe para 27 no início e depois decai continuamente até 2. Em 40 TPS, o FabMAN mantém 40 até o ciclo 5, reduz para 30 no ciclo 6 e encerra em 4, enquanto o aHLF tem um pico inicial de 55 e segue uma queda progressiva até 1.

A análise indica que o FabMAN, apesar de sua variabilidade em cenários de alta carga, ajusta os parâmetros e converge para valores finais semelhantes ao aHLF. Por outro lado, o aHLF se destaca pela estabilidade e previsibilidade dos ajustes. Enquanto o FabMAN adota uma abordagem mais agressiva, o aHLF mantém uma adaptação mais consistente. Esses padrões sugerem que uma estratégia híbrida, unindo a estabilidade do aHLF com a resposta adaptativa do FabMAN, pode otimizar o desempenho do Hyperledger Fabric em diferentes condições operacionais.

### 5.3. Estratégia Híbrida

Dado que os experimentos indicaram que o FabMAN apresentou melhor desempenho em cargas mais baixas (10 e 20 TPS), enquanto o aHLF foi mais eficiente em cenários de alta carga (40 TPS), propomos uma estratégia híbrida que combina os pontos fortes de ambas as abordagens. Esta estratégia poderia adaptar-se dinamicamente, utilizando o FabMAN em cenários de menor carga e o aHLF em situações de maior carga, oferecendo uma solução otimizada para diferentes condições operacionais no futuro.

A Figura 5 apresenta a variação da latência média e da vazão ao longo dos ciclos de teste considerando essa estratégia híbrida. Os resultados indicam que, ao combinar as estratégias de forma adaptativa, é possível obter um desempenho mais equilibrado, reduzindo a latência sem comprometer a estabilidade da rede. Além disso, os ajustes dinâmicos do *BatchSize* (BS) e *BatchTimeout* (BT) se mantêm alinhados com as condições ideais de cada cenário, garantindo maior eficiência na formação de blocos e na velocidade de processamento das transações.



**Figura 5. Análise da Latência Média e Vazão por Ciclo de adaptação na Estratégia Híbrida**

Esse modelo híbrido serve como base para estudos futuros, onde se pretende validar sua eficácia em diferentes configurações de rede, explorando a viabilidade de uma implementação automatizada que selecione dinamicamente a melhor estratégia conforme a carga de transações da rede blockchain.

## 6. Considerações Finais

Este estudo apresentou uma análise detalhada de duas estratégias, aHLF (baseado no aPBFT) e FabMAN, aplicadas ao Hyperledger Fabric para a previsão e ajuste dinâmico dos parâmetros *BatchSize* (BS) e *BatchTimeout* (BT). A abordagem experimental avaliou

o impacto dessas configurações em diferentes cenários de taxa de transações por segundo (TPS), analisando métricas como latência e vazão para determinar qual estratégia oferece melhor desempenho em diferentes condições operacionais. Os resultados indicaram que nenhuma das estratégias, quando aplicada isoladamente, é ideal para todos os cenários testados. O aHLF demonstrou maior estabilidade e previsibilidade, promovendo ajustes graduais nos parâmetros e garantindo uma redução consistente da latência ao longo dos ciclos de adaptação, especialmente em cenários de alta carga. Já o FabMAN, com sua abordagem mais agressiva, foi eficaz em cenários de baixa e média carga, mas apresentou oscilações em altas taxas de transações, comprometendo a estabilidade.

Na análise da vazão, o FabMAN inicialmente superou o aHLF em altas taxas de transação, demonstrando maior agilidade na adaptação aos primeiros ciclos de ajuste. No entanto, essa vantagem não foi sustentada ao longo dos testes, apresentando flutuações que comprometeram a previsibilidade do desempenho. O aHLF, por outro lado, manteve uma vazão mais estável e sem variações bruscas, mostrando-se mais indicado para redes blockchain que exigem previsibilidade a longo prazo. O desempenho quase ótimo do aHLF foi justificado pela sua capacidade de adaptação contínua e estável. A análise das diferenças entre as estratégias revelou que o aHLF seguiu um modelo conservador, enquanto o FabMAN aplicou ajustes rápidos, gerando volatilidade. Embora as estratégias tenham se comportado de maneira diferente, ambas convergiram para valores finais semelhantes. A análise sugere que um modelo híbrido, combinando a estabilidade do aHLF com a rapidez do FabMAN, seria a abordagem mais eficiente, equilibrando a redução da latência com a previsibilidade. Como continuidade deste trabalho, propõe-se o desenvolvimento e validação deste modelo híbrido, além da aplicação das estratégias em redes blockchain com diferentes arquiteturas e cenários, para aprimorar ainda mais o desempenho do Hyperledger Fabric.

## Referências

- Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., Enyeart, D., Ferris, C., Laventman, G., Manevich, Y., et al. (2018). Hyperledger fabric: a distributed operating system for permissioned blockchains. In *Proceedings of the Thirteenth EuroSys Conference*, page 30. ACM.
- Baliga, A., Subhod, I., Kamat, P., and Chatterjee, S. (2018). Performance evaluation of the quorum blockchain platform. *arXiv preprint arXiv:1809.03421*.
- Castro, M., Liskov, B., et al. (1999). Practical byzantine fault tolerance. In *OsDI*, volume 99, pages 173–186.
- Chacko, J. A., Mayer, R., and Jacobsen, H.-A. (2021). Why do my blockchain transactions fail? a study of hyperledger fabric. In *Proceedings of the 2021 international conference on management of data*, pages 221–234.
- de Sá, A. S., Silva Freitas, A. E., and de Araújo Macêdo, R. J. (2013). Adaptive request batching for byzantine replication. *ACM SIGOPS Operating Systems Review*, 47(1):35–42.
- Greve, F., Sampaio, L., Abijaude, J., Coutinho, A. A., Brito, I., and Queiroz, S. (2018). Blockchain e a Revolução do Consenso sob Demanda. In *Proc. of SBRC Minicursos*.

- Liu, C.-M., Badigineni, M., and Lu, S. W. (2021). Adaptive blocksize for iot payload data on fabric blockchain. In *2021 30th Wireless and Optical Communications Conference (WOCC)*, pages 92–96. IEEE.
- Mendonça, R., Moura, E., Gonçalves, G., Vieira, A., and Nacif, J. (2023). Comparação e análise de custo e desempenho entre nós de redes blockchain permissionadas e públicas. In *Anais do XLI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 141–154, Porto Alegre, RS, Brasil. SBC.
- Moura, E., Melo, C., Gonçalves, G., Silva, F., and Soares, A. (2024). Uma ferramenta de avaliação de desempenho para plataforma blockchain hyperledger fabric: Hlf-pet. In *Anais do II Colóquio em Blockchain e Web Descentralizada*, pages 8–13, Porto Alegre, RS, Brasil. SBC.
- Ongaro, D. and Ousterhout, J. (2014). In search of an understandable consensus algorithm. In *2014 USENIX annual technical conference (USENIX ATC 14)*, pages 305–319.
- Roy, U. and Ghosh, N. (2024). Fabman: A framework for ledger storage and size management for hyperledger fabric-based iot applications. *IEEE Transactions on Network and Service Management*.
- Saeed, S. H., Hadi, S., and Hamad, A. H. (2022). Performance evaluation of e-voting based on hyperledger fabric blockchain platform. *Revue d’Intelligence Artificielle*.
- Shalaby, S., Abdellatif, A., Al-Ali, A., Mohamed, A. M., Erbad, A., and Guizani, M. (2020). Performance evaluation of hyperledger fabric. *2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT)*, pages 608–613.
- Silva, F., Gonçalves, G., fé, I., Feitosa, L., and Soares, A. (2023). Avaliação de desempenho de blockchains permissionadas hyperledger orientada ao planejamento de capacidade de recursos computacionais. In *Anais do XLI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 71–84, Porto Alegre, RS, Brasil. SBC.
- Wai, K. and Thein, N. (2023). Performance analysis on block size valuation of hyperledger fabric blockchain. *2023 IEEE Conference on Computer Applications (ICCA)*, pages 50–55.
- Xu, X., Weber, I., and Staples, M. (2019). *Architecture for blockchain applications*. Springer.