Análise Comparativa de Performance entre Criptografia Clássica e Pós-Quântica no TLS 1.3

Luan Gustavo Mariano Marques[1], **André Ricardo Zavan**[1]

[1]Instituto Federal do Paraná – Campus Paranavaí (IFPR)
Paranavaí – Paraná – Brasil

20230006174@estudantes.ifpr.edu.br, andre.zavan@ifpr.edu.br

Abstract. *We compare X25519 and X25519MLKEM768 in Go 1.24 using handshake latency and captured bytes per connection. Three emulated network scenarios and runs of 1,000, 5,000 and 10,000 attempts yielded 96,000 successful TLS 1.3 handshakes. The hybrid had higher medians and captured volumes in all nine comparisons. At 10,000 attempts, median differences ranged from 0.742 to 2.168 ms, relative increases decreased from 46.98% to 2.54%, and volume increased around 100–115%. Serial structure motivated descriptive analysis with Vargha–Delaney A. Findings concern the tested environment, without establishing statistical significance or application-level effects.*

Resumo. *Comparam-se X25519 e X25519MLKEM768 no Go 1.24 pela latência do handshake e pelo volume capturado por conexão. Três cenários emulados e rodadas de 1.000, 5.000 e 10.000 tentativas totalizaram 96.000 handshakes TLS 1.3 bem-sucedidos. O híbrido apresentou maiores medianas e volumes nas nove comparações. Na rodada de 10.000, as diferenças entre medianas variaram de 0,742 a 2,168 ms; os aumentos relativos diminuíram de 46,98% para 2,54%, e o volume cresceu aproximadamente 100–115%. A estrutura serial motivou análise descritiva com A de Vargha–Delaney. Os resultados restringem-se ao ambiente avaliado, sem estabelecer significância estatística ou efeitos sobre aplicações.*

1. Introdução

O TLS 1.3 protege comunicações por meio de mecanismos de autenticação, estabelecimento de chaves e cifragem dos dados [Rescorla 2018]. Algoritmos clássicos de chave pública baseados em fatoração e logaritmos discretos são ameaçados pelo algoritmo de Shor quando executado em um computador quântico suficientemente capaz [Shor 1999]. O planejamento da migração deve considerar a duração necessária da confidencialidade e o tempo de transição, inclusive diante da possibilidade de coleta atual de tráfego para decifração futura [Mosca 2015].

O NIST padronizou o mecanismo de encapsulamento de chaves ML-KEM no FIPS 203 [National Institute of Standards and Technology 2024]. No Go 1.24, o pacote `crypto/tls` passou a oferecer o grupo híbrido `X25519MLKEM768`, que combina X25519 e ML-KEM-768 [The Go Team 2025]. Essa disponibilidade permite avaliar a configuração híbrida na biblioteca padrão, enquanto trabalhos correlatos empregam outras integrações TLS e diferentes procedimentos de medição [Paquin et al. 2020, Tzinos et al. 2022, Kampanakis and Childs-Klein 2024].

O objetivo deste estudo é comparar a latência do *handshake* e o volume médio capturado por conexão entre X25519 e X25519MLKEM768, sob três perfis de rede emulados. A pergunta de pesquisa é: como essas medidas diferem entre os modos na

implementação Go 1.24 e no ambiente avaliado? A contribuição consiste em quantificar as diferenças observadas, incluindo sua variação entre cenários e execuções, sem pressupor que sejam diretamente transferíveis a outras implementações ou aplicações.

A medição concentra-se no estabelecimento de uma nova sessão TLS, excluindo a conexão TCP e a transferência posterior de conteúdo. Esse recorte permite avaliar a etapa afetada pela mudança do mecanismo de estabelecimento de chaves. Já o *Time-To-Last-Byte* (TTLB) inclui a transferência de dados e responde a uma questão distinta: [Kampanakis and Childs-Klein 2024] mostram que o impacto relativo do handshake pode diminuir com o aumento do conteúdo transferido.

A escolha do handshake como objeto de análise também delimita a interpretação do custo. Em uma conexão que será reutilizada para muitas operações, a negociação inicial e a transferência subsequente constituem etapas diferentes; em novas conexões, essa negociação precisa ser realizada novamente. Assim, conhecer sua duração é relevante para compreender o estabelecimento da comunicação, mas não basta para estimar o tempo de uma requisição HTTP, chamada de serviço ou transferência de arquivo. A avaliação de aplicações permanece uma etapa posterior, coerente com a distinção entre handshake e TTLB discutida por [Kampanakis and Childs-Klein 2024].

Para responder à pergunta de pesquisa, o estudo organiza a comparação em três aspectos: a posição central e a região superior das latências, o volume médio observado por conexão e a variação desses resultados entre cenários e rodadas. Essa organização permite relatar tanto o padrão predominante quanto situações que não o acompanham, sem exigir que todas as métricas produzam uma conclusão idêntica.

2. Fundamentação teórica

2.1. TLS 1.3, autenticação e estabelecimento de chaves

O TLS 1.3 separa as funções de autenticação, estabelecimento de chaves e proteção dos dados. Em um handshake completo com certificado e ECDHE, o cliente anuncia parâmetros e uma contribuição de estabelecimento de chaves no *ClientHello*; o servidor seleciona os parâmetros e responde com sua contribuição no *ServerHello*. As mensagens seguintes incluem o certificado, a comprovação de posse da chave de autenticação e a verificação da negociação. O material secreto estabelecido participa da derivação das chaves de tráfego, enquanto os dados da aplicação são protegidos por mecanismos simétricos [Rescorla 2018].

No ECDHE, as partes utilizam contribuições efêmeras para estabelecer o segredo. A propriedade de sigilo futuro protege sessões anteriores contra o comprometimento posterior da chave de autenticação, nas condições previstas pelo protocolo; ela não torna o estabelecimento clássico resistente a um adversário capaz de resolver o problema matemático subjacente [Rescorla 2018, Shor 1999]. Essa distinção é relevante porque a troca do mecanismo de estabelecimento de chaves não corresponde automaticamente à troca da assinatura utilizada no certificado.

Neste estudo, X25519 é o componente clássico de estabelecimento de chaves, enquanto ECDSA P-256 autentica o servidor nos dois modos. O experimento mantém essa autenticação comum para concentrar a comparação na configuração de estabelecimento de chaves. Não se avaliam assinaturas pós-quânticas, retomada de sessão ou transferência de conteúdo após a negociação.

2.2. Criptografia pós-quântica e ML-KEM

A criptografia pós-quântica busca mecanismos resistentes também a adversários com capacidade de computação quântica. A motivação não se limita ao momento em que

um ataque se torne viável: dados que precisam permanecer confidenciais por longos períodos podem ser coletados antes da migração. O planejamento deve relacionar o tempo de proteção necessário, o tempo de transição e o horizonte de comprometimento dos mecanismos atuais [Mosca 2015].

O ML-KEM, padronizado no FIPS 203 e derivado do CRYSTALS-Kyber, é um mecanismo de encapsulamento de chaves baseado em reticulados modulares. Sua segurança está relacionada à dificuldade do problema *Module Learning with Errors*; o padrão define os conjuntos ML-KEM-512, ML-KEM-768 e ML-KEM-1024 [National Institute of Standards and Technology 2024]. O presente estudo utiliza apenas ML-KEM-768, sem comparar os conjuntos de parâmetros.

O funcionamento de um KEM compreende três operações. A geração de chaves produz uma chave de encapsulamento e uma chave de desencapsulamento. A partir da primeira, o encapsulamento gera um segredo compartilhado e um texto cifrado; a parte que possui a chave de desencapsulamento utiliza esse texto para recuperar o segredo correspondente. Esse procedimento estabelece material secreto que pode ser usado na proteção da comunicação, e não cifra diretamente o conteúdo da aplicação [National Institute of Standards and Technology 2024].

No ML-KEM-768, a chave de encapsulamento possui 1.184 bytes e o texto cifrado, 1.088 bytes [National Institute of Standards and Technology 2024]. Esses tamanhos ajudam a explicar a motivação para medir o volume de dados, mas não devem ser confundidos com o total capturado em uma conexão. As mensagens TLS contêm outros campos, e a captura também registra cabeçalhos e controle do transporte.

2.3. TLS híbrido e condições de rede

O grupo `X25519MLKEM768`, disponível no pacote `crypto/tls` do Go 1.24, combina contribuições de X25519 e ML-KEM-768 no estabelecimento de chaves [The Go Team 2025]. A abordagem híbrida mantém um componente clássico durante a introdução do componente pós-quântico. Sua avaliação de desempenho deve considerar tanto a execução das operações quanto o material transmitido; a latência do handshake não equivale ao tempo isolado de processamento do ML-KEM.

[Paquin et al. 2020] mostram que tamanho de mensagens, custo computacional e condições de rede podem afetar de maneiras distintas a conclusão de uma conexão TLS. A distribuição das mensagens em múltiplos pacotes e a ocorrência de perdas são relevantes para interpretar o desempenho dos mecanismos avaliados pelos autores. Isso motiva a observação em diferentes perfis de rede, sem presumir que os mesmos efeitos ou limiares quantitativos se reproduzam em outra implementação.

Neste experimento, atraso, variação do atraso e perda são parâmetros do ambiente emulado. O atraso configurado na saída de uma interface não é, por si só, o RTT medido da conexão. Da mesma forma, um aumento de latência não demonstra isoladamente que houve retransmissão ou saturação da CPU. A emulação com `netem`, também utilizada por [Paquin et al. 2020], permite definir condições de execução; a identificação das causas de cada demora exigiria uma análise adicional dos eventos correspondentes.

3. Trabalhos correlatos

3.1. Métricas e ambientes de avaliação

[Kwiatkowski et al. 2019] descrevem a avaliação de HRSS e SIKE com clientes Chrome/Canary e infraestrutura Cloudflare, motivada pelas diferenças de tamanho e custo computacional dos mecanismos. O uso de clientes e redes reais constitui um recorte distinto

do ambiente local emulado deste estudo. Esses mecanismos também diferem de ML-KEM-768, de modo que seus valores não representam estimativas diretas do desempenho do grupo avaliado aqui.

[Paquin et al. 2020] avaliam o tempo de conclusão do handshake com condições de rede emuladas por `netem`. [Tzinos et al. 2022] investigam handshake e throughput em um ambiente baseado em OQS/OpenSSL. Esses estudos motivam a observação de diferentes perfis de atraso e perda, mas utilizam implementações e mecanismos que não coincidem integralmente com os deste experimento.

[Kampanakis and Childs-Klein 2024] avaliam TTLB com `s2n-tls`, emulação de rede e diferentes volumes transferidos. Além da diferença de métrica, suas medições incluem o estabelecimento TCP, excluído da cronometragem adotada aqui. Essa distinção impede comparar diretamente tempos absolutos sem considerar os limites de cada medição.

3.2. Implementações e recorte da pesquisa

[Souvatzidaki and Limniotis 2025] utilizam OQS Provider e OpenSSL 3.4, com configuração de autenticação que inclui CRYSTALS-Dilithium. Essa escolha difere da autenticação clássica mantida neste trabalho. A Tabela 1 sintetiza as principais diferenças metodológicas, preservando a distinção entre medir o handshake e medir a transferência completa de conteúdo.

Tabela 1. Síntese dos recortes metodológicos dos trabalhos correlatos.

Trabalho	Métrica	Ambiente e implementação
Kwiatkowski et al. (2019)	Tempo de handshake	Clientes Chrome/Canary e rede real; infraestrutura Cloudflare.
Paquin et al. (2020)	Tempo de handshake	Rede emulada com netem; OQS/OpenSSL.
Tzinos et al. (2022)	Handshake e throughput	Rede emulada; OQS/OpenSSL e liboqs.
Kampanakis e Childs-Klein (2024)	TTLB	Netem em namespaces Linux; s2n-tls.
Souvatzidaki e Limniotis (2025)	Duração do handshake	Variação da perda; OQS Provider e OpenSSL 3.4.
Este trabalho	Mediana, P95 e volume capturado	Docker e netem; biblioteca padrão do Go 1.24.13.

Fonte: Elaborado pelo autor (2026), com base nos trabalhos citados.

O recorte deste estudo é a comparação entre dois grupos de estabelecimento de chaves na biblioteca padrão do Go, com autenticação comum e parâmetros de rede definidos. A contribuição não depende de demonstrar superioridade do Go ou ausência de pesquisas anteriores. Consiste em produzir evidências para essa implementação e esse procedimento, documentando as métricas de latência e o volume observado no ponto de captura.

4. Metodologia

4.1. Ambiente e configurações

A pesquisa é quantitativa e experimental. Os modos de estabelecimento de chaves e os perfis de rede são as condições comparadas; as respostas observadas são a latência e o volume médio capturado por conexão. Cliente e servidor foram executados em contêineres

Docker no mesmo notebook Lenovo IdeaPad 1 15IGL7, com Intel Celeron N4020 de dois núcleos e 4 GiB de RAM. Os binários foram compilados com Go 1.24.13, utilizando `crypto/tls` sem biblioteca criptográfica externa.

A comunicação ocorreu em rede bridge isolada, na sub-rede 10.10.0.0/24, com servidor em 10.10.0.2:8443. O servidor aceitou ambos os grupos; o cliente restringiu sua configuração a `tls.X25519` no baseline ou a `tls.X25519MLKEM768` no híbrido. Um certificado autoassinado com ECDSA P-256 foi mantido nos dois modos, com nome `tls-server` e endereço do servidor no SAN. O cliente confiou nesse certificado por meio de um `x509.CertPool` local, validando o nome configurado. Não foi habilitado cache de sessões; verificaram-se TLS 1.3, grupo negociado e ausência de retomada nas conexões bem-sucedidas.

A separação em serviços Docker manteve a mesma configuração de comunicação para os dois modos e permitiu aplicar a degradação na interface do cliente. O servidor aguardou conexões na porta definida, enquanto o cliente realizou tentativas sucessivas e registrou os resultados. Essa organização facilita a reprodução da configuração, mas os contêineres continuam compartilhando os recursos físicos do notebook; não constituem máquinas independentes.

4.2. Cenários e execução das rodadas

As condições de rede foram aplicadas com `tc netem` exclusivamente à saída da interface `eth0` do cliente. A Tabela 2 apresenta os parâmetros; a variação do atraso foi configurada com distribuição normal. S0 não recebeu degradação artificial. Os valores são parâmetros do emulador, não medições de RTT nem representações de uma rede real específica.

Tabela 2. Parâmetros de rede aplicados à saída do cliente.

Cenário	Atraso	Variação do atraso	Perda
S0	—	—	—
S1	20 ms	5 ms	0,5%
S2	80 ms	20 ms	2,0%

Fonte: Elaborado pelo autor (2026).

Foram realizadas rodadas de 1.000, 5.000 e 10.000 tentativas por combinação de modo e cenário, com a mesma implementação e os mesmos parâmetros de rede. Cada rodada compreendeu seis casos, com tentativas consecutivas, sem divisão em blocos. A ordem foi S0, S1 e S2, com baseline seguido de híbrido em cada cenário, sem alternância ou aleatorização. Houve uma execução para cada quantidade de tentativas, em momentos distintos. Os dados foram analisados separadamente por rodada, sem agrupamento das amostras entre execuções.

A rodada de 10.000 foi adotada como referência para a apresentação detalhada por reunir mais observações por caso. Trata-se de uma escolha operacional para caracterizar mediana, P95 e dispersão, sem determinação de um tamanho amostral ótimo. As rodadas menores permitem comparações complementares; não demonstram que o aumento da amostra provoque convergência ou elimine efeitos do ambiente.

4.3. Medição e captura

Cada tentativa começou com uma nova conexão TCP, estabelecida por `net.DialTimeout` com limite de dez segundos. Após a conexão, o cliente configurou um prazo de dez segundos para o handshake TLS e mediu o tempo

imediatamente antes e depois da chamada a `Handshake()`. Assim, a duração exclui o estabelecimento TCP e a transferência de dados de aplicação, mas não isola o processamento criptográfico. As latências bem-sucedidas foram registradas em nanossegundos e convertidas para milissegundos na análise. Todas as tentativas tiveram seu estado registrado; falhas não seriam substituídas por novas tentativas.

Antes da coleta, um handshake verificou a disponibilidade do servidor, sem integrar a amostra. Em cada caso, a captura com `dumpcap` na interface do cliente, filtrada para TCP na porta 8443, foi iniciada e confirmada antes das tentativas. Ao final, aguardaram-se três segundos para registrar tráfego restante e encerrou-se a captura. A extração de campos com `tshark` ocorreu somente após os seis casos. Foram conferidos tentativas, sucessos, conexões, grupos negociados e contadores de descarte, preservando CSVs, capturas, registros do ambiente e cópia do código. Essa triagem verifica consistência, sem garantir isoladamente a captura de todos os bytes.

Os registros de latência identificam a tentativa e o modo, enquanto um arquivo adicional conserva estado de conclusão, horários, endereços e informações da sessão TLS. A distinção entre esses registros permite conferir quantas tentativas geraram uma latência válida, sem substituir falhas ou selecionar apenas uma parte conveniente da execução. Na análise apresentada, todas as tentativas previstas foram concluídas com sucesso.

4.4. Métricas e análise estatística

A mediana e o percentil 95 (P95) foram calculados com interpolação linear no índice $(n-1)q$, em que n é o número de observações e q a proporção do quantil. Foram mantidos todos os valores das conexões bem-sucedidas, sem exclusão de outliers. A mediana resume a posição central e o P95 descreve a região superior da distribuição observada, sem garantir o desempenho de conexões futuras.

O volume médio capturado é a soma de `frame.len` de todos os quadros do caso, dividida pelo número de conexões TCP observadas. A medida considera os dois sentidos do tráfego no cliente, incluindo cabeçalhos, controle TCP e retransmissões; não foi aplicado filtro de exibição TLS. As contagens e somas dos campos extraídos foram comparadas às dos arquivos PCAPNG. A medida caracteriza esse ponto de captura e não apenas o handshake ou o volume físico transmitido no enlace. Diferenças absolutas e relativas foram calculadas contra o baseline da mesma rodada e cenário, antes do arredondamento.

A ordem das observações foi examinada por autocorrelação dos valores e dos postos nas defasagens de 1, 5, 10, 50 e 100 tentativas [NIST/SEMATECH s.d.a]. São defasagens na sequência, não intervalos fixos em segundos. Aplicou-se também o teste de sequências acima e abaixo da mediana, bilateral, com aproximação normal sem correção de continuidade; empates com a mediana foram omitidos apenas desse diagnóstico [NIST/SEMATECH s.d.c]. As 18 verificações receberam correção de Bonferroni, com nível familiar de 5% [NIST/SEMATECH s.d.b].

Os diagnósticos encontraram estrutura serial em parte das séries. Por isso, não foram utilizados p-valores convencionais de Mann–Whitney U, cuja aplicação pressupõe observações independentes, apesar de dispensar normalidade [The SciPy Community s.d.]. A ausência de rejeição nas demais séries não foi tomada como prova de independência. Adotou-se uma comparação descritiva uniforme, sem declarar significância estatística entre os modos.

Como complemento, calculou-se o tamanho de efeito não paramétrico A de Vargha–Delaney, apresentado por [Arcuri and Briand 2011, p. 6–7], separadamente por rodada e

cenário:

$$\widehat{A}_{H,B} = \frac{\sum_{i=1}^{n_H}\sum_{j=1}^{n_B}\left[I(h_i > b_j) + \frac{1}{2}I(h_i = b_j)\right]}{n_H n_B}. \quad (1)$$

Aqui, h_i e b_j são as latências híbrida e baseline, n_H e n_B são os tamanhos das amostras e I vale 1 quando a condição é satisfeita e 0 caso contrário. Comparam-se todos os pares entre amostras: conta-se 1 quando o híbrido demora mais, 0,5 no empate e 0 quando demora menos, dividindo-se a soma pelo total de pares. A medida descreve a predominância de maiores latências híbridas quando superior a 0,5; não representa aumento percentual de latência. Seu uso descritivo não elimina a dependência serial nem demonstra equivalência quando próximo de 0,5.

5. Resultados e discussão

As três rodadas concluíram, respectivamente, 6.000, 30.000 e 60.000 tentativas, totalizando 96.000 handshakes bem-sucedidos. Os resultados são apresentados por execução, preservando as diferenças entre as quantidades de tentativas.

5.1. Rodada de 10.000 tentativas por caso

A Tabela 3 reúne latência e volume médio capturado. O híbrido apresentou maior mediana, P95 e volume nos três cenários. As diferenças entre medianas foram de 0,742 ms, 1,580 ms e 2,168 ms em S0, S1 e S2; os aumentos relativos foram de 46,98%, 6,51% e 2,54%. Portanto, o aumento absoluto cresceu entre esses cenários, enquanto sua proporção em relação ao baseline diminuiu.

Tabela 3. Latência e volume médio capturado: 10.000 tentativas por caso.

Cenário	Modo	Mediana (ms)	P95 (ms)	Volume médio (bytes/conexão)	Aumento do volume (%)
S0	Baseline	1,580	2,905	1.967,44	—
	Híbrido	2,323	4,043	4.239,63	115,49
S1	Baseline	24,271	32,993	2.274,50	—
	Híbrido	25,851	34,827	4.548,25	99,97
S2	Baseline	85,298	121,169	2.277,12	—
	Híbrido	87,467	123,591	4.554,24	100,00

Nota: O volume inclui cabeçalhos, controle TCP e retransmissões observadas, nos dois sentidos. O aumento toma como referência o baseline do mesmo cenário.

Fonte: Elaborado pelo autor (2026).

O P95 também aumentou nos dois modos de S0 para S2. As diferenças entre os modos foram de 1,138 ms, 1,834 ms e 2,421 ms, equivalentes a 39,17%, 5,56% e 2,00%. A redução percentual não indica desaparecimento do custo observado, pois ocorreu simultaneamente a diferenças absolutas maiores.

A Figura 1 mostra as distribuições. As caixas delimitam o primeiro e o terceiro quartis, com a mediana no interior; os bigodes alcançam as observações extremas dentro de $[Q_1 - 1{,}5\,IQR,\ Q_3 + 1{,}5\,IQR]$. Os demais valores são representados individualmente e foram mantidos na análise. Os intervalos interquartílicos não se sobrepõem em S0, mas se sobrepõem em S1 e S2. Os painéis têm escalas verticais próprias.

Quanto ao volume, o aumento relativo foi de 115,49% em S0 e aproximadamente 100% nos outros cenários. As diferenças absolutas foram próximas: 2.272,18, 2.273,75 e

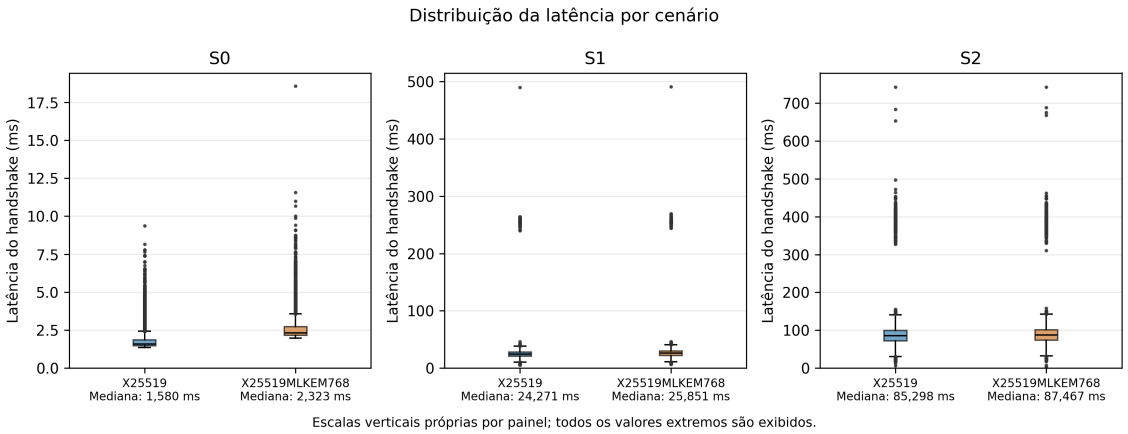


Figura 1. Latências por modo e cenário: 10.000 observações por grupo.

Fonte: Elaborado pelo autor (2026).

2.277,12 bytes por conexão, respectivamente. Esses valores descrevem o conjunto de quadros capturados; não permitem atribuir toda a diferença exclusivamente aos componentes criptográficos ou identificar retransmissões como sua causa.

As Figuras 2 e 3 complementam a tabela ao mostrar os contrastes entre modos e cenários. A primeira evidencia o maior volume capturado no híbrido; a segunda mostra o crescimento do P95 em ambos os modos. As linhas da segunda figura apenas conectam os cenários avaliados: não representam uma série temporal nem valores medidos entre S0, S1 e S2.

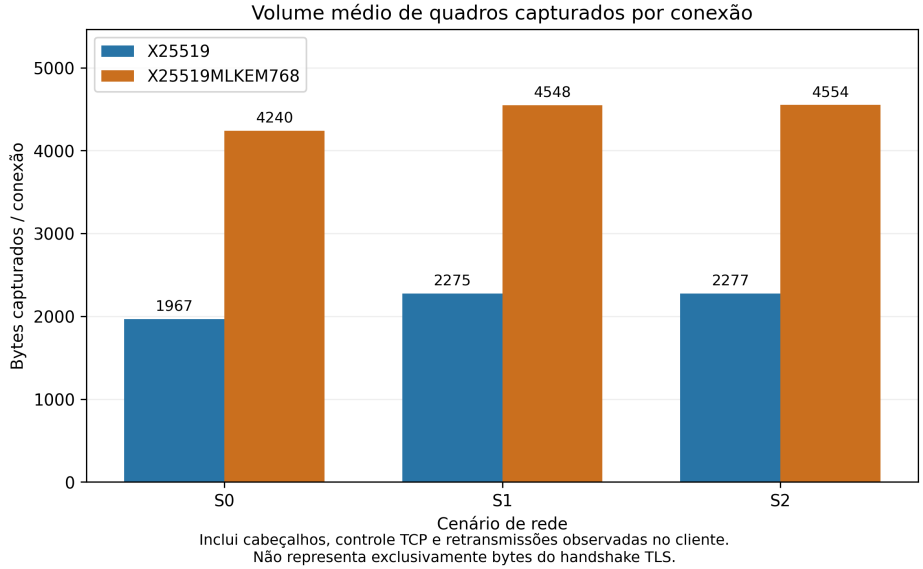


Figura 2. Volume médio capturado por conexão na rodada de 10.000.

Fonte: Elaborado pelo autor (2026).

A diferença entre bytes e latência é central para a interpretação. Os volumes medidos permitem descrever a quantidade adicional de dados observada, enquanto a latência resulta da execução e da comunicação sob cada perfil de rede. Por isso, o aumento de aproximadamente 100% no volume em S1 e S2 não implica aumento de 100% na duração do handshake. Na mesma rodada, os aumentos relativos de mediana nesses cenários foram de 6,51% e 2,54%.

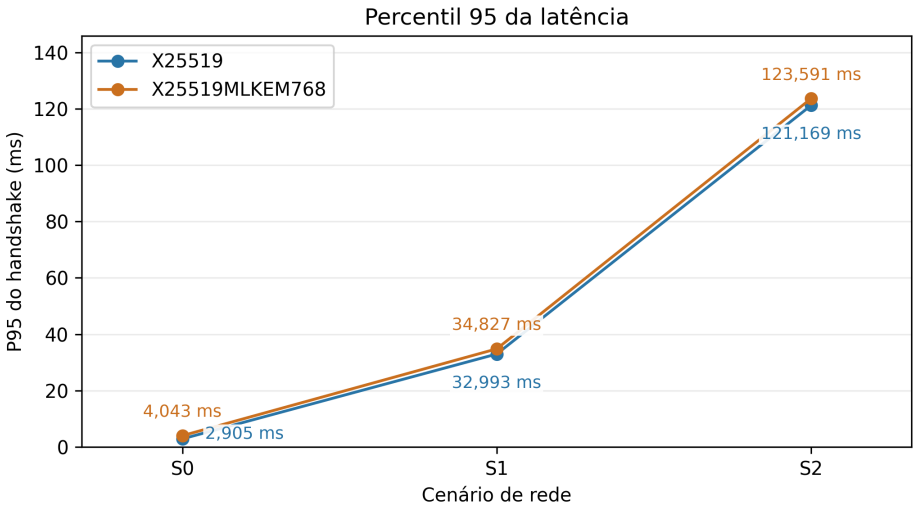


Figura 3. P95 da latência por modo e cenário na rodada de 10.000.

Fonte: Elaborado pelo autor (2026).

5.2. Comparação entre rodadas e tamanho de efeito

A Tabela 4 apresenta as diferenças relativas de latência e os valores de $\widehat{A}_{H,B}$ da Equação 1. As medianas foram superiores no híbrido nas nove combinações, com maior diferença relativa em S0 e menor em S2 em todas as rodadas. O volume médio capturado também foi superior no híbrido em todos os casos: próximo de 4.240 contra 1.967 bytes em S0 e aproximadamente o dobro do baseline em S1 e S2.

Tabela 4. Diferenças relativas de latência e tamanho de efeito por rodada.

Tentativas por caso	Cenário	Mediana (%)	P95 (%)	$\widehat{A}_{H,B}$
1.000	S0	+33,07	-15,79	0,7494
1.000	S1	+13,57	+9,45	0,6742
1.000	S2	+1,45	+2,72	0,5236
5.000	S0	+39,19	+22,12	0,9579
5.000	S1	+6,77	+6,33	0,5988
5.000	S2	+2,96	+0,95	0,5327
10.000	S0	+46,98	+39,17	0,9021
10.000	S1	+6,51	+5,56	0,5803
10.000	S2	+2,54	+2,00	0,5229

Nota: As diferenças tomam como referência o baseline do mesmo cenário e rodada. Valores positivos indicam maior latência híbrida. A é descritivo e não constitui teste de significância.

Fonte: Elaborado pelo autor (2026).

O P95 foi maior no híbrido em oito das nove combinações. A exceção foi S0/1.000, com 5,107 ms no baseline e 4,300 ms no híbrido, embora a mediana permanecesse menor no baseline. Assim, o comportamento central e a região superior das distribuições nem sempre produziram a mesma ordenação. A repetição do sentido da diferença de medianas coexistiu com variações de magnitude e de P95 entre execuções; essas variações não podem ser atribuídas exclusivamente ao tamanho amostral.

5.3. Estrutura das séries e tamanho de efeito

Quanto aos diagnósticos, sete das 18 séries apresentaram evidência contra a ordenação aleatória após Bonferroni: os dois modos em S0 nas três rodadas e o híbrido em S1/10.000.

Esses resultados avaliam a sequência das medições, não a diferença entre modos. Fundamentam a interpretação descritiva estabelecida na Seção 4.4.

Uma série corresponde às latências de um modo em um cenário e uma rodada, ordenadas pelas tentativas. A inspeção dessa ordem procura associações entre medições próximas, que não são visíveis apenas pela mediana ou pelo boxplot. Em S0/10.000, as autocorrelações entre latências consecutivas foram de 0,408 no baseline e 0,495 no híbrido. Esses coeficientes descrevem associação dentro de cada série; não são percentuais de aumento da latência nem identificam a causa física da estrutura observada.

O tamanho de efeito foi superior a 0,5 nas nove comparações e diminuiu de S0 para S2 em cada rodada. Em S0/10.000, $\widehat{A}_{H,B} = 0{,}9021$ acompanhou diferença de mediana de 0,742 ms; em S2/10.000, $\widehat{A}_{H,B} = 0{,}5229$ acompanhou 2,168 ms. A predominância de maiores latências híbridas nas comparações entre pares pode, portanto, ser menor mesmo quando a diferença entre medianas em milissegundos é maior. As medidas descrevem aspectos distintos; valores próximos de 0,5 em S2 não demonstram equivalência.

Por exemplo, A igual a 0,9021 em S0/10.000 expressa a pontuação média das comparações entre todas as latências híbridas e todas as latências baseline, com meio peso para empates. Não significa que cada handshake híbrido tenha demorado mais, nem que a duração tenha aumentado 90,21%. A diferença de medianas informa uma magnitude em milissegundos, enquanto A informa a ordenação entre as amostras. A combinação dessas medidas permite descrever o resultado sem reduzir a avaliação a um único percentual.

5.4. Relação com a literatura e alcance dos resultados

A avaliação em diferentes condições de rede segue a motivação de [Paquin et al. 2020]: o tempo de estabelecimento TLS deve ser examinado considerando comunicação e operações criptográficas. Contudo, os mecanismos, implementações e perfis de rede diferem entre os estudos. Os resultados presentes não reproduzem diretamente um limiar de perda ou uma magnitude reportada na literatura; acrescentam uma caracterização específica do grupo híbrido na biblioteca padrão do Go.

A diminuição do aumento relativo entre cenários também precisa ser distinguida da discussão de TTLB de [Kampanakis and Childs-Klein 2024]. Neste trabalho, o denominador é a latência do handshake baseline de cada cenário; naquele recorte, a transferência de conteúdo compõe o tempo total avaliado. As duas análises mostram a importância de explicitar a métrica e sua referência, mas não autorizam transportar os percentuais de uma para a outra. A conclusão sustentada aqui é que as diferenças observadas dependem da medida escolhida e das condições da execução, e não que a configuração híbrida tenha impacto desprezível em qualquer aplicação.

5.5. Limitações da interpretação

Cliente e servidor compartilharam o mesmo computador, e cada modo ocupou um momento distinto da execução, com baseline sempre anterior ao híbrido. A ordem fixa limita a separação entre o efeito da configuração criptográfica e variações do ambiente. A estrutura serial também impede tratar automaticamente as numerosas conexões como réplicas independentes. As três quantidades de tentativas não substituem repetições independentes de uma mesma condição.

Atraso, variação e perda foram modificados conjuntamente; portanto, seus efeitos individuais não foram isolados. As durações incluem processamento e comunicação, sem permitir separar o tempo do ML-KEM dos demais componentes do handshake ou identificar um gargalo de CPU ou rede. O volume representa o ponto de captura virtual,

não uma medição direta do enlace físico. Não houve comparação entre bibliotecas TLS, transferência de conteúdo nem medição de TTLB. Esses limites restringem as conclusões ao procedimento e ao ambiente avaliados.

6. Conclusão

O experimento quantificou diferenças entre X25519 e X25519MLKEM768 na biblioteca padrão do Go. O híbrido apresentou maior mediana de latência e volume médio capturado nas nove combinações de rodada e cenário, e maior P95 em oito delas. Na rodada de 10.000, as diferenças entre medianas cresceram de 0,742 para 2,168 ms de S0 para S2, enquanto os aumentos relativos diminuíram de 46,98% para 2,54%. O volume capturado aumentou aproximadamente 100–115%.

O tamanho de efeito complementou essas medidas, mostrando menor predominância de maiores latências híbridas de S0 para S2 em todas as rodadas. Esse comportamento e a exceção do P95 em S0/1.000 mostram que a avaliação precisa considerar diferentes aspectos das distribuições. A maior quantidade de tentativas oferece mais observações da execução, mas não elimina as limitações do desenho experimental.

A contribuição consiste em caracterizar o custo observado da configuração híbrida, mostrando que um impacto percentual menor nos cenários degradados não significa ausência de diferença absoluta. A análise descritiva e o tamanho de efeito delimitam esses resultados às amostras, sem estabelecer significância estatística ou desempenho completo de aplicações. Trabalhos futuros podem realizar repetições independentes com ordem alternada ou aleatorizada dos modos, separar cliente e servidor em equipamentos distintos e incorporar transferência de dados.

Referências

Arcuri, A. and Briand, L. (2011). A practical guide for using statistical tests to assess randomized algorithms in software engineering. Technical Report 2010-10, Simula Research Laboratory.

Kampanakis, P. and Childs-Klein, W. (2024). The impact of data-heavy, post-quantum TLS 1.3 on the Time-To-Last-Byte of real-world connections. IACR Cryptology ePrint Archive, Paper 2024/176. Disponível em: `https://eprint.iacr.org/2024/176`. Acesso em: 30 set. 2026.

Kwiatkowski, K., Sullivan, N., Langley, A., Levin, D., and Mislove, A. (2019). Measuring TLS key exchange with post-quantum KEM. In *Workshop Record of the Second PQC Standardization Conference*. National Institute of Standards and Technology. Disponível em: `https://www.cs.umd.edu/~dml/papers/pqc_nist19.pdf`. Acesso em: 30 set. 2026.

Mosca, M. (2015). Cybersecurity in an era with quantum computers: will we be ready? Technical report, Institute for Quantum Computing, University of Waterloo, Waterloo, Ontario, Canada.

National Institute of Standards and Technology (2024). FIPS 203: Module-Lattice-Based Key-Encapsulation Mechanism Standard. Federal information processing standard, National Institute of Standards and Technology, Gaithersburg, MD.

NIST/SEMATECH (s.d.a). e-handbook of statistical methods: Autocorrelation. Seção 1.3.5.12. `https://www.itl.nist.gov/div898/handbook/eda/section3/eda35c.htm`. Acesso em: 29 set. 2026.

NIST/SEMATECH (s.d.b). e-handbook of statistical methods: Bonferroni's method. Seção 7.4.7.3. https://www.itl.nist.gov/div898/handbook/prc/section4/prc473.htm. Acesso em: 29 set. 2026.

NIST/SEMATECH (s.d.c). e-handbook of statistical methods: Runs test for detecting non-randomness. Seção 1.3.5.13. https://www.itl.nist.gov/div898/handbook/eda/section3/eda35d.htm. Acesso em: 29 set. 2026.

Paquin, C., Stebila, D., and Tamvada, G. (2020). Benchmarking post-quantum cryptography in TLS. In *Post-Quantum Cryptography — PQCrypto 2020*, pages 72–91. Springer.

Rescorla, E. (2018). RFC 8446: The Transport Layer Security (TLS) Protocol Version 1.3. Request for Comments 8446, Internet Engineering Task Force.

Shor, P. W. (1999). Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Review*, 41(2):303–332.

Souvatzidaki, K. and Limniotis, K. (2025). Post-quantum key exchange in TLS 1.3: Further analysis on performance of new cryptographic standards. *Cryptography*, 9(4):73.

The Go Team (2025). Go 1.24 Release Notes. https://go.dev/doc/go1.24. Acesso em: 27 set. 2026.

The SciPy Community (s.d.). SciPy reference: scipy.stats.mannwhitneyu. https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.mannwhitneyu.html. Acesso em: 29 set. 2026.

Tzinos, I., Limniotis, K., and Kolokotronis, N. (2022). Evaluating the performance of post-quantum secure algorithms in the TLS protocol. *Journal of Surveillance, Security and Safety*, 3:101–127.