

Análise e Otimização de Throughput no Protocolo QUIC: Um Resumo do Artigo “Understanding QUIC’s Throughput Speedbumps”

Arthur Mendes , Júlio César Gonzaga Ferreira Silva , Kaio Henrique , Lucas Cabral ,
Pedro Henrique Gaioso , Rafael Pereira Vilefort

¹ Pontifícia Universidade Católica de Minas Gerais (PUC Minas)

Resumo. O protocolo QUIC, alternativa moderna ao TCP+TLS, apresenta variações de throughput entre diferentes implementações. Com base no artigo “Understanding QUIC’s Throughput Speedbumps”, este trabalho investiga os principais gargalos de desempenho e propõe um modelo de arquitetura paralela, independente de implementação, para otimizar o protocolo.

Abstract. The QUIC protocol, a modern alternative to TCP+TLS, shows throughput variations across implementations. Based on “Understanding QUIC’s Throughput Speedbumps”, this work examines key performance bottlenecks and proposes an implementation-agnostic, parallel architecture model to optimize the protocol.

1. Motivação

O QUIC surgiu como alternativa moderna ao TCP+TLS, oferecendo conexões seguras, multiplexadas e de baixa latência no espaço do usuário. Essa flexibilidade acelerou a adoção, mas o *throughput* ainda costuma ficar abaixo do TCP e varia entre implementações, pois o protocolo roda como aplicação e cada pilha utiliza arquiteturas e bibliotecas distintas. O artigo “Understanding QUIC’s Throughput Speedbumps” busca explicar essa variabilidade e propor diretrizes para desempenho padronizado.

2. Objetivos

Identificar e classificar gargalos de *throughput* em pilhas QUIC e sugerir uma arquitetura de alto desempenho, agnóstica de implementação. Foram avaliadas mvfst, picoquic, picoquic-dpdk, MsQuic e quicly em Linux com enlaces de até 25 Gbps, observando impacto de design interno, bibliotecas criptográficas e estratégias de I/O.

3. Modelo

Propõe-se um *pipeline* paralelo de cinco estágios: I/O, UDP/IP, processamento

QUIC, criptografia e aplicação. Cada estágio roda de forma assíncrona em *threads* leves com memória compartilhada, aumentando o uso de núcleos e reduzindo esperas. A abordagem é independente de implementação e consolida boas práticas para maximizar *throughput*.

4. Resultados

Em servidores de 25 Gbps, mvfst e quicly registraram ganhos de $1,88\times$ e $1,36\times$; o quicly atingiu 15,4 Gbps com MTU 4000 bytes, superando TCP+TLS. O uso conjunto de GSO e GRO reduziu fortemente o custo de I/O, dobrando o *throughput* em alguns cenários. Após essas otimizações, a criptografia permaneceu o principal gargalo, chegando a 55% do tempo de CPU.

5. Conclusões e Trabalhos Futuros

As diferenças de desempenho entre pilhas QUIC são majoritariamente estruturais. O *pipeline* proposto padroniza práticas e viabiliza alto *throughput* de forma portátil. Futuro: criptografia em unidades maiores (nível de *flow*), aceleração híbrida com hardware e formalização de boas práticas de engenharia para QUIC de alta performance.