louzt/quic-tunnel

Source: Private · **Status:** Active · **Period:** 2024-06 (active)

Multi-transport SSH fallback that survives hostile transit: layered QUIC, Hysteria2, gost, tls-direct, and direct SSH with CA pinning.

The challenge

My SSH sessions to the VPS were dropping roughly every hour on Mexican carrier paths. The carrier's traffic inspection was matching common SSH fingerprints and shaping the connection mid-session. Standard SSH tooling gave me no visibility into which transport was being shaped, and a single-transport tunnel is a single point of failure.

My role

I designed and shipped a layered transport fallback daemon that picks the best available path on connect and degrades through successive tiers when one path fails. The daemon is the transport layer my loust-vps bridge runs on.

What changed

Session survival

Baseline: Roughly hourly drops on Mexican carrier paths

Twelve-plus hours of continuous sessions

Reconnect time on drop

Baseline: 30+ seconds, manual

Under 1 second, automatic

New-peer setup time

Baseline: 1 minute

5 minutes (CA pin ceremony)

Stack

Go | QUIC | Hysteria2 | TLS | SSH