Zero-Prefill Keep-Alive

Source: Publicaciones · **Status:** Reference demo · **Period:** 2026-04 (active)

Academic publication draft on a transport pattern for keeping connections useful without unnecessary prefill work.

The challenge

Standard keep-alive patterns prefill a session with speculative data so the connection looks busy. The prefill cost shows up at scale and the speculative data often has to be discarded on the first real exchange.

My role

I am drafting a publication on a transport pattern that keeps the connection useful without the prefill cost — the connection stays alive, the audit chain stays clean, and no speculative data is shipped over the wire.

What I did (3–5 decisions)

1. Connection-state primitive instead of prefill packets

Why: The keep-alive is encoded in a connection-state primitive rather than in speculative data. The receiver knows the connection is alive because the primitive says so; no payload needs to ship.

Tradeoff: The pattern requires both ends to opt in. A server that does not implement the primitive falls back to standard keep-alive with the prefill cost it already pays.

2. Audit chain stays honest

Why: No speculative data over the wire means no speculative data to invalidate when the first real exchange happens. The audit chain does not need a 'discard the prefilled bytes' step.

Tradeoff: The honest audit chain is more verbose than the dishonest one. Worth it for an operator who values reproducibility over hand-waving.

What changed

Prefill bytes shipped per keep-alive

Baseline: Variable, speculative

Zero — connection-state primitive only

Discard step in the audit chain

Baseline: Required — discard prefilled bytes on first real exchange

Not required

Overall tradeoffs

This is a draft; the public gist is pending. The reference demo lives in the operator's local transport harness until the publication is ready to ship.

Lessons

A keep-alive pattern that ships speculative data is not a keep-alive — it is a polite lie that compounds over the lifetime of the connection.

Stack

networking | transport | keep-alive | operations