July 24th, 2026

Exploring SCHC Compression of QUIC Frames

draft-sirohi-schc-quic-frame-compression

Authors: Samar Sirohi - Laurent Toutain

Presenter: Samar Sirohi

IETF 126, Vienna

IETF

Motivation and Main Challenge

Why?

QUIC frames carry transport metadata:

- Frame Type
- Stream ID
- Offset and Length
- ACK information

Potentially relevant for:

- Small packets
- Constrained or expensive links
- Predictable machine-to-machine traffic
- Space communication

But QUIC is already compact

Savings > Rule ID + framing + residue

Compression occurs inside the QUIC endpoint, before encryption.

Design Space

Dimension	Option A	Option B
Compression unit	One rule per frame	One rule for the full frame set
Rule context	Generic base rule	Traffic-specific, may change over time
QUIC encoding	Extension-frame container	Alternative payload style

State dependency

Packet-stateless

Safer under loss/reordering

Cross-packet state

Better compression, harder synchronization

Likely starting point

Traffic-specific rules, complete-frame-sequence compression, and packet-stateless reconstruction.

Illustrative Size Comparison

Complete QUIC payload size

Example	Native	Generic Container	Optimized Container	Alternative Syntax
Small ACK	5B	6B	5B	**3B**
STREAM + 64B data	69B	72B	68B	**65B**
ACK + STREAM	74B	77B	70B	**67B**

Generic compression often increases size.

Savings require:

- Known multi-byte fields
- Predictable frame sequences
- Bundling several frames under one rule
- Low encoding overhead

Conclusion and Future Direction

What we learned

- SCHC compression of QUIC frames is technically feasible.
- However, QUIC frame encoding is already highly compact.
- Meaningful gains require predictable traffic, frame bundling, or deeper QUIC integration.
- Therefore, frame compression is mainly a niche optimization for controlled deployments.

Where to focus next

- Prioritize compression of the outer packet headers: IP, UDP, and accessible QUIC header fields.
- This appears to offer a better compression-gain versus implementation-complexity trade-off.
- It can also reuse more of the existing SCHC compression architecture.

Required enabler

- Efficient dynamic rule management through SCHC CORECONF:
- low-overhead rule installation and updates;
- optimized RPCs;
- private SIDs.

Next step: outer QUIC compression supported by an efficient CORECONF management plane.

THE END

Thank you for listening