Network Time Protocols
Internet-Draft
Intended status: Informational
Expires: 29 January 2026

G. McCollum
Cisco Systems

28 July 2025

Time Synchronization over QUIC
draft-mccollum-tsq-03

Abstract

This document proposes a modern, secure, and extensible time synchronization protocol designed to operate over the QUIC transport protocol. Known as TSQ (Time Synchronization over QUIC), this protocol aims to address the limitations of traditional NTP by leveraging QUIC's encryption, widespread UDP/443 acceptance, and multiplexed stream capabilities. TSQ is designed for contemporary deployment environments, including enterprise networks, cloud-native systems, containers, and mobile devices, where traditional UDP-based NTP struggles with security, scalability, or operational reliability.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 29 January 2026.

Copyright Notice

Table of Contents

1. Introduction

Time synchronization is foundational to modern computing. It underpins authentication systems, log correlation, distributed transactions, and more. NTP, the current standard, was designed in a different era and brings challenges related to security, deployment compatibility, and extensibility. TSQ is proposed as a new protocol built directly on top of QUIC, leveraging its modern transport features to provide secure, authenticated, and operationally-friendly time synchronization.

2. Scope and Goals

TSQ is intended to:

* Provide secure and authenticated time synchronization

* Support modern deployment scenarios
* Operate in environments where UDP/123 is blocked
* Be extensible and future-proof
* Scale for enterprise and cloud

TSQ is _not_ intended to:

* Replace NTP in ultra-precise or constrained devices
* Replace public NTP infrastructure without optimization

3. Protocol Overview

TSQ uses QUIC as its transport, establishing secure, short-lived connections. A typical exchange:

1. Client opens a QUIC connection to the TSQ server (UDP/443).
2. Client sends a TSQ Request with nonce and timestamp request.
3. Server replies with timestamps, echoed nonce, and metadata.
4. Client calculates RTT and adjusts clock accordingly.

4. Security and Threat Model

TSQ relies on QUIC’s handshake for mutual authentication, confidentiality, and replay protection. Optional Ed25519 or HMAC signatures can be added if auditability is required. By default, QUIC session integrity suffices.

5. Scalability Considerations

Short-lived connections, session resumption, and optional stateless design support scalability. TSQ is suitable for enterprise and cloud deployments.

6. Message Format (TLV)

TSQ Request

* Type: 0x01
* Nonce (16 bytes)

* Optional extensions

TSQ Response

* Type: 0x02

* Echoed Nonce

* Server Time

* Receive Timestamp

* Send Timestamp

* Optional metadata and signature

7. Use Cases

* Cloud/container infrastructure

* Mobile clients

* Firewalled enterprise networks

* High-precision timing visibility

8. Comparison to Existing Protocols

The following table highlights key differences between traditional NTP, NTS, and the proposed TSQ protocol:

Feature	NTP	NTS	TSQ
Transport	UDP	UDP+TLS	QUIC (UDP/443)
Encryption	No	Yes	Always
Extensibility	Low	Medium	High
Mobile Support	No	No	Yes
Precision Mode	No	No	Yes

Table 1

9. QUIC Overview

QUIC is a UDP-based transport protocol that provides multiplexed, stream-based delivery with built-in encryption and connection migration. It combines transport and cryptographic handshake layers, simplifying deployment behind firewalls. TSQ builds on QUIC to inherit its confidentiality, integrity, and path-resilience characteristics.

10. Clarification of Mobile Support

Mobile support in TSQ leverages QUIC's connection ID and migration capabilities. When a mobile device changes IP addresses (e.g. Wi-Fi to LTE), the QUIC connection remains valid as long as the connection ID is preserved, enabling uninterrupted time synchronization.

11. Clarification of Precision Mode

Precision Mode refers to TSQ' s optional support for fixed-length packets and constant-time processing. This reduces jitter introduced by variable processing times and packet sizes, improving accuracy in latency-sensitive environments.

12. Explanation of Server Time Field

The Server Time field in TSQ responses provides a consistent reference for clients even when timestamp echoing is used. Including it explicitly simplifies validation and debugging, particularly when clock discrepancies or replay concerns are involved.

13. QUIC Multiplexing and Port 443

TSQ can coexist with other protocols over QUIC on port 443 using Application-Layer Protocol Negotiation (ALPN). This avoids conflicts and enables unified endpoint services, allowing TSQ to blend with HTTPS or other QUIC-based services on the same port.

14. Next Steps

* Solicit feedback on sync behavior and design
* Improve trust and crypto model
* Implement prototype in QUIC
* Submit official Internet-Draft if interest grows

15. Acknowledgments

Thanks to contributors from the QUIC and NTP working groups for input on timing accuracy and protocol design.

16. Relationship to Roughtime

Roughtime [ROUGHTIME] is a secure time authentication protocol designed to provide coarse-grained time (typically accurate within a few seconds) using cryptographic proofs. It is particularly useful in scenarios where a system needs to verify that its clock is reasonably accurate before performing sensitive operations such as certificate validation.

Unlike TSQ, which aims to deliver secure and precise time synchronization suitable for continuous use in enterprise, mobile, and containerized environments, Roughtime focuses on bootstrapping trust in time from potentially untrusted states. Roughtime servers are stateless and optimized for high-throughput, low-complexity validation rather than tight synchronization.

As such, TSQ and Roughtime serve complementary purposes. Roughtime may help systems establish an initial level of temporal trust, while TSQ provides ongoing synchronization over a secure and extensible transport protocol.

17. Considerations

17.1. Cryptographic Considerations

TSQ relies on QUIC's TLS 1.3-based transport layer for encryption and authentication. These cryptographic operations are not eliminated, they are performed per packet at the transport layer, but they are not duplicated at the application level for each time exchange, unlike NTS for NTP, which often executes separate AEAD operations per request (e.g. decrypt cookie, authenticate request, encrypt new cookie, sign response).

While this offloading does not make QUIC "free" from a cryptographic standpoint, it enables session reuse and amortization of crypto costs over time. This can benefit TSQ in high-churn or mobile environments where maintaining lightweight, persistent connections is important. Future revisions aim to provide data-driven comparisons between TSQ and NTS in terms of cryptographic cost and performance.

17.2. Packet Symmetry

To reduce timing variance, TSQ recommends matching the sizes of request and response packets. Even though QUIC encrypts and pads its payloads, maintaining symmetry at the application layer helps minimize round-trip delay bias.

17.3. QUIC Datagram Considerations

TSQ may also use QUIC datagrams for lighter-weight synchronization. These are encrypted and authenticated using the same keys as QUIC streams but provide unreliable delivery. By using transport-level encryption, TSQ avoids additional per-packet cryptographic overhead at the application layer while preserving confidentiality and integrity.

17.4. Session Models and State

NTS establishes session state via cookie exchanges, while TSQ inherits QUIC's session management and connection reuse. QUIC enables stream multiplexing, bidirectional communication, and connection migration—capabilities useful in mobile and cloud-native environments. The key difference lies not in the existence of state, but in how that state is maintained and leveraged.

17.5. Deployment Assumptions

TSQ is designed for modern deployment scenarios such as enterprise, mobile, and cloud networks, especially those operating behind firewalls. It is not intended to replace the global NTP stratum hierarchy or public time pools directly but may complement them in environments where UDP-based NTP has limitations.

18. References

[RFC8915] Franke, D., Sibold, D., Teichel, K., Dansarie, M., and R. Sundblad, "Network Time Security for the Network Time Protocol", RFC 8915, <https://datatracker.ietf.org/doc/html/rfc8915>.

[RFC7384] Mizrahi, T., "Security Requirements of Time Protocols in Packet Switched Networks", RFC 7384, <https://datatracker.ietf.org/doc/html/rfc7384>.

[RFC9000] Iyengar, J. and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport", RFC 9000, <https://datatracker.ietf.org/doc/html/rfc9000>.

[RFC9221] Pauly, T., Kinnear, E., and D. Schinazi, "An Unreliable Datagram Extension to QUIC", RFC 9221, <https://datatracker.ietf.org/doc/html/rfc9221>.

[RFC9308] Khlewind, M. and B. Trammell, "Applicability of the QUIC Transport Protocol", RFC 9308, <https://datatracker.ietf.org/doc/html/rfc9308>.

[ROUGHTIME] Ladd, W. and M. Dansarie, "Roughtime", Work in Progress, Internet-Draft, draft-ietf-ntp-roughtime-14, <https://datatracker.ietf.org/doc/html/draft-ietf-ntp-roughtime-14>.

Appendix A. IANA Considerations

This document has no IANA actions.

Author’s Address

Garrett McCollum
Cisco Systems
Email: gmccollu@cisco.com