

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: 21 August 2024

J. Yao
H. Li
M. Zhang
CNNIC
D. Keathley
J. Gould
VeriSign, Inc.
18 February 2024

**Extensible Provisioning Protocol (EPP) Transport over QUIC
draft-yao-regext-epp-quic-01**

Abstract

This document describes how an Extensible Provisioning Protocol (EPP) session is mapped onto a QUIC connection. EPP over QUIC (EoQ) leverages the performance and security features of the QUIC protocol as an EPP transport.

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 21 August 2024.

Copyright Notice

Copyright (c) 2024 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the [Trust Legal Provisions](#) and are provided without warranty as described in the Revised BSD License.

Table of Contents

- [1.](#) Introduction [2](#)
- [2.](#) Conventions Used in This Document [2](#)
- [3.](#) Session Management [3](#)
- [4.](#) Message Exchange [4](#)
- [5.](#) Data Unit Format [6](#)
- [6.](#) Transport Considerations [6](#)
- [7.](#) IANA Considerations [7](#)
 - [7.1.](#) Registration of an EoQ Identification String [7](#)
 - [7.2.](#) Registration of Port Number [7](#)
- [8.](#) Security Considerations [8](#)
- [9.](#) Acknowledgements [8](#)
- [10.](#) Normative References [8](#)
- [Appendix A.](#) Change History [9](#)
 - [A.1.](#) Change from 00 to 01 [9](#)
- Authors' Addresses [9](#)

1. Introduction

This document describes how the Extensible Provisioning Protocol (EPP [[RFC5730](#)]) is mapped onto the QUIC transport [[RFC9000](#)]. QUIC is a network protocol that is based on UDP and incorporates native encryption support using TLS [[RFC9001](#)]. Though based on UDP, QUIC provides connection semantics similar to other stateful protocols. This document discusses how EPP implementations can work with this and other features of QUIC while preserving the core EPP semantics.

2. Conventions Used in This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [[BCP14](#)] when, and only when, they appear in all capitals, as shown here.

3. Session Management

Mapping EPP session management facilities onto the QUIC service is accomplished with a combination of a QUIC connection and QUIC streams. An EPP session first requires creation of a QUIC connection between two peers, one that initiates the connection request and one that responds to the connection request. The initiating peer is called the "client", and the responding peer is called the "server". An EPP server MUST listen for QUIC connection requests on a standard UDP port assigned by IANA.

A successfully established QUIC connection is automatically secured by the native TLS support that QUIC provides.

Once the QUIC connection is established, the EPP client MUST then create a QUIC stream. The EPP server MUST return an EPP <greeting> to the client on that same stream once it is created. After reading the EPP <greeting> message, the EPP client sends EPP commands and receives EPP responses on the same stream. A QUIC stream corresponds to an EPP connection and an authenticated QUIC stream, via a successful EPP <login>, corresponds to an EPP session. This can also be referred to as a EoQ session.

An EPP session is normally ended by the client issuing an EPP <logout> command. A server receiving an EPP <logout> command MUST end the EPP session and close the QUIC stream.

EoQ connections are established as described in the QUIC transport specification [[RFC9000](#)]. During connection establishment, EoQ support is indicated by selecting the Application-Layer Protocol Negotiation (ALPN) token "eoq" in the crypto handshake.

A single QUIC connection may allow multiple QUIC streams. This means that a single EoQ connection may support multiple EoQ sessions. A server MAY limit the life span of an established EoQ session. EoQ sessions that are inactive for more than a server-defined period MAY be ended by a server closing the QUIC stream. A server MAY close EoQ sessions that have been open and active for longer than a server-defined limit. Once the last QUIC stream for a QUIC connection is closed, the server MAY end the QUIC connection immediately.

4. Message Exchange

With the exception of the EPP server greeting, EPP messages are initiated by the EPP client in the form of EPP commands. An EPP server **MUST** return an EPP response to an EPP command on the same QUIC stream that carried the command. If the QUIC stream is closed after a server receives and successfully processes a command but before the response can be returned to the client, the server **MAY** attempt to undo the effects of the command to ensure a consistent state between the client and the server. EPP commands are idempotent, so processing a command more than once produces the same net effect on the repository as successfully processing the command once.

An EPP client streams EPP commands to an EPP server on an established QUIC stream. A client **MUST NOT** distribute commands from a single EPP session over multiple QUIC streams. A client **MAY** establish multiple QUIC streams to support multiple EPP sessions with each session mapped to a single stream. A server **SHOULD** limit a client to a maximum number of QUIC streams per QUIC connection based on server capabilities and operational load.

EPP describes client-server interaction as a command-response exchange where the client sends one command to the server and the server returns one response to the client.

Each EPP data unit **MUST** contain a single EPP message. Commands **MUST** be processed independently.

A server **SHOULD** impose a limit on the amount of time required for a client to issue a well-formed EPP command in order to reduce the risk associated with a resource exhaustion attack. A server **SHOULD** end an EPP session and close the QUIC stream if a well-formed command is not received within the time limit.

A general state machine for an EPP server is described in [Section 2 of \[RFC5730\]](#). A general client-server message exchange using QUIC transport is illustrated in Figure 1. It shows the exchange over a single QUIC stream of a QUIC connection. Many QUIC streams may open and close during the life of a QUIC connection.

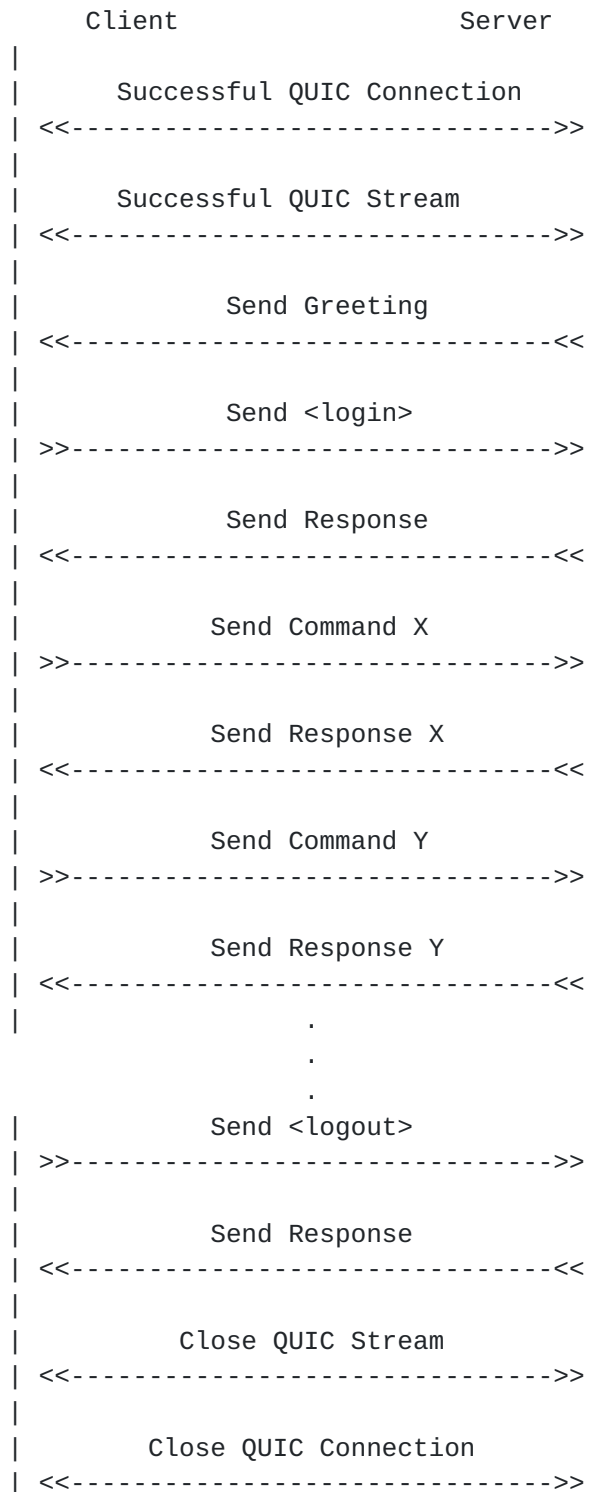
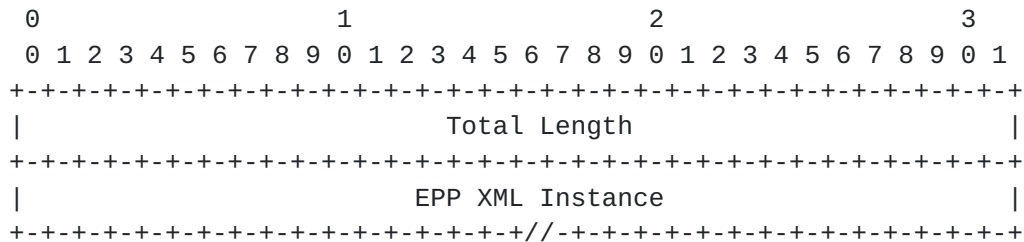


Figure 1: QUIC Client-Server Message Exchange

The EPP server MUST follow the "EPP Server State Machine" procedure described in [[RFC5730](#)].

5. Data Unit Format

The EPP data unit contains two fields: a 32-bit header that describes the total length of the data unit, and the EPP XML instance. The length of the EPP XML instance is determined by subtracting four octets from the total length of the data unit. A receiver must successfully read that many octets to retrieve the complete EPP XML instance before processing the EPP message. EPP Data Unit Format (one tick mark represents one bit position):



Total Length (32 bits): The total length of the EPP data unit measured in octets in network (big endian) byte order. The octets contained in this field MUST be included in the total length calculation. EPP XML Instance (variable length): The EPP XML instance carried in the data unit.

6. Transport Considerations

[Section 2.1 of \[RFC5730\]](#) describes considerations to be addressed by protocol transport mappings. This document addresses each of those considerations using a combination of features of the QUIC protocol itself and features of this document.

- * Command Order: QUIC guarantees ordered processing of data within each stream. [Section 2 of \[RFC9000\]](#) describes streams in detail.
- * Session Mapping: EPP session management utilizes QUIC streams and is described in [Section 3](#)
- * Stateful Nature: QUIC supports stateful communications between endpoints via connection IDs and long-lived streams within each connection. Sections [2](#) and [5](#) of [\[RFC9000\]](#) describe these features, respectively.
- * Frame Data Units: QUIC uses frames as one of its units of information when sending data over a stream. Part packets-frames of [\[RFC9000\]](#) describes frames, and packets, in detail.

- * Congestion Avoidance: QUIC provides various mechanisms to help achieve congestion avoidance. [[RFC9002](#)] describes these mechanisms in detail.
- * Reliability: QUIC uses message acknowledgement, packet retransmission, and other features to ensure reliability. Part packetization of [[RFC9000](#)] describes these features in detail.
- * Pipelining: Pipelining is allowed in EoQ. QUIC streams support sending multiple frames without waiting for responses from the other peer. This does not change the basic single command, single response operating mode of the core EPP protocol.

[7.](#) IANA Considerations

[7.1.](#) Registration of an EoQ Identification String

This document creates a new registration for the identification of EoQ in the "TLS Application-Layer Protocol Negotiation (ALPN) Protocol IDs" registry [[RFC7301](#)].

- * Protocol: EoQ
- * Identification Sequence: 0x65 0x6F 0x71 ("eoq")
- * Reference: This document

[7.2.](#) Registration of Port Number

The "Service Name and Transport Protocol Port Number Registry" contains an entry for EPP UDP/700 based on [[RFC6335](#)]. However, no known implementations of EPP over UDP exist. The entry will be reassigned to reference this draft.

- * Service Name: epp
- * Port Number: 700
- * Transport Protocol(s): UDP
- * Assignee: IESG
- * Contact: IETF Chair
- * Description: EPP run over QUIC
- * Reference: This document

8. Security Considerations

EPP over QUIC provides the similar security with EPP over TCP with TLS. Some related security issues have been discussed in [RFC5734] and [RFC9000].

EoQ servers run the risk of a resource exhaustion attack by allowing the creation of unlimited QUIC streams per QUIC connection. Servers SHOULD limit a client to a maximum number of QUIC streams per QUIC connection based on server capabilities and operational load.

9. Acknowledgements

The authors wish to thank the following persons for their feedback and suggestions: Scott Hollenbeck.

10. Normative References

- [BCP14] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- Leiba, B., "Ambiguity of Uppercase vs Lowercase in [RFC 2119](#) Key Words", [BCP 14](#), [RFC 8174](#), May 2017.
- <https://www.rfc-editor.org/info/bcp14>
- [RFC5730] Hollenbeck, S., "Extensible Provisioning Protocol (EPP)", STD 69, [RFC 5730](#), DOI 10.17487/RFC5730, August 2009, <https://www.rfc-editor.org/info/rfc5730>.
- [RFC5734] Hollenbeck, S., "Extensible Provisioning Protocol (EPP) Transport over TCP", STD 69, [RFC 5734](#), DOI 10.17487/RFC5734, August 2009, <https://www.rfc-editor.org/info/rfc5734>.
- [RFC6335] Cotton, M., Eggert, L., Touch, J., Westerlund, M., and S. Cheshire, "Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry", [BCP 165](#), [RFC 6335](#), DOI 10.17487/RFC6335, August 2011, <https://www.rfc-editor.org/info/rfc6335>.
- [RFC7301] Friedl, S., Popov, A., Langley, A., and E. Stephan, "Transport Layer Security (TLS) Application-Layer Protocol Negotiation Extension", [RFC 7301](#), DOI 10.17487/RFC7301, July 2014, <https://www.rfc-editor.org/info/rfc7301>.

- [RFC9000] Iyengar, J., Ed. and M. Thomson, Ed., "QUIC: A UDP-Based Multiplexed and Secure Transport", [RFC 9000](#), DOI 10.17487/RFC9000, May 2021, <<https://www.rfc-editor.org/info/rfc9000>>.
- [RFC9001] Thomson, M., Ed. and S. Turner, Ed., "Using TLS to Secure QUIC", [RFC 9001](#), DOI 10.17487/RFC9001, May 2021, <<https://www.rfc-editor.org/info/rfc9001>>.
- [RFC9002] Iyengar, J., Ed. and I. Swett, Ed., "QUIC Loss Detection and Congestion Control", [RFC 9002](#), DOI 10.17487/RFC9002, May 2021, <<https://www.rfc-editor.org/info/rfc9002>>.

[Appendix A](#). Change History

[A.1](#). Change from 00 to 01

1. Added Dan Keathley and James Gould as co-authors and aligned the draft with EPP [RFC 5734](#).

Authors' Addresses

Jiankang Yao
CNNIC
4 South 4th Street, Zhongguancun, Haidian District
Beijing
Beijing, 100190
China
Phone: +86 10 5881 3007
Email: yaojk@cnnic.cn

Hongtao Li
CNNIC
4 South 4th Street, Zhongguancun, Haidian District
Beijing
Beijing, 100190
China
Email: lihongtao@cnnic.cn

Man Zhang
CNNIC
4 South 4th Street, Zhongguancun, Haidian District
Beijing
Beijing, 100190
China
Email: zhangman@cnnic.cn

Daniel Keathley
VeriSign, Inc.
12061 Bluemont Way
Reston, VA 20190
United States of America
Email: dkeathley@verisign.com
URI: <http://www.verisigninc.com>

James Gould
VeriSign, Inc.
12061 Bluemont Way
Reston, VA 20190
United States of America
Email: jgould@verisign.com
URI: <http://www.verisigninc.com>

