QUIC
Internet-Draft
Intended status: Standards Track
Expires: 30 July 2026

M. Kozuka
Okayama University

26 January 2026

QUIC Extension for Server-Initiated Connection Migration
draft-kozuka-quic-server-migration-00

Abstract

This document specifies an extension of QUIC that allows the server to initiate connection migration.

About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at https://masa-koz.github.io/draft-kozuka-quic-server-migration/draft-kozuka-quic-server-migration.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-kozuka-quic-server-migration/.

Discussion of this document takes place on the QUIC Working Group mailing list (mailto:quic@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/quic/. Subscribe at https://www.ietf.org/mailman/listinfo/quic/.

Source for this draft and an issue tracker can be found at https://github.com/masa-koz/draft-kozuka-quic-server-migration.

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 30 July 2026.

Table of Contents

1. Introduction

RFC 9000 defines connection migration as a mechanism initiated by the client to change its IP address or network path without disrupting the QUIC connection. This mechanism is based on the assumption that a client may be behind a NAT and a server has a public address. However, there is a reversed situation where a client has a public address and a server is behind a NAT. This document specifies an extension that reverses this role: the server initiates migration.

2. Conventions and Definitions

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 BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Motivation

A key scenario is:

* The server is initially behind a NAT and cannot accept direct connections.

* The client has a public IP address.

* The server first establishes the QUIC connection via a relay server.

* Immediately after the handshake, the server migrates the connection to the client's public address, reducing latency.

This specification does not define how the server obtains the public address of the client. Possible methods include application-level signaling or external mechanisms, but these are out of scope.

4. Scope

* Reverse migration roles defined in RFC 9000.

* Maintain QUIC security guarantees.

* No new NAT traversal mechanism.

5. Negotiating Extension Use

allow_server_migration (0x3e764478):

Clients advertise their support of this extension by sending the allow_server_migration (0x3e764478) transport parameter (Section 7.4 of [QUIC-TRANSPORT]) with an empty value. Sending this transport parameter signals to the server that the client will accept the server-initiated migration.

Servers also send this parameter with an empty value. The server informs the client that it will initiate the connection migration by sending this parameter.

When this extension is negotiated, the server-initiated migration is only permitted and the client-initiated migration is prohibited.

An implementation that understands this transport parameter MUST treat the receipt of a non-empty value as a connection error of type TRANSPORT_PARAMETER_ERROR.

Endpoints MUST NOT remember the value of this extension for 0-RTT.

6. Connection Migration Procedure

The connection migration procedure is the same as (Section 9 of [QUIC-TRANSPORT]) except that the roles between the client and the server are reversed.

7. Security Considerations

TODO Security

8. IANA Considerations

8.1. QUIC Transport Parameter

This document registers the allow_server_migration transport parameter in the "QUIC Transport Parameters" registry established in Section 22.3 of [QUIC-TRANSPORT]. The following fields are registered:

Value: 0x3e764478

Parameter Name: allow_server_migration

Status: Provisional

Specification: This document

Change Controller: IETF (iesg@ietf.org)

Contact: Masahiro Kozuka (kozuka@okayama-u.ac.jp)

9. Normative References

[QUIC-TRANSPORT] 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/rfc/rfc9000>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

Acknowledgments

TODO acknowledge.

Questions

* Sould we extend this extension to allow both clients and servers to initiate connection migration?

* Any new security conserations from allowing servers to initiate connection migration?

Author’s Address

Masahiro Kozuka
Okayama University
Email: kozuka@okayama-u.ac.jp