QUIC
Internet-Draft
Intended status: Standards Track
Expires: 18 April 2026

M. Munizaga
Ethereum Foundation
M. Seemann
Smallstep

15 October 2025

QUIC New Server Preferred Address
draft-munizaga-quic-new-preferred-address-00

Abstract

This document specifies an extension to QUIC to allow a server to request a migration to a new preferred address.

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://marcopolo.github.io/new-preferred-address/draft-munizaga-quic-new-preferred-address.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-munizaga-quic-new-preferred-address/.

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/MarcoPolo/new-preferred-address.

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 18 April 2026.

Table of Contents

1. Introduction

The QUIC transport protocol allows a client to migrate connections at any time to any new address (Section 9 of [QUIC-TRANSPORT]). This allows the connection to survive changes to the client's address. QUIC also allows a server to migrate to a different address, but only a single time, and only to an address specified at the start of a connection via the Server's Preferred Address (Section 9.6 of [QUIC-TRANSPORT]). For some applications, including those where the server and client are peers, limiting the server to only a single migration at the beginning is too limiting. This document specifies an extension to QUIC to allow a server to request a migration to a new preferred address.

This document defines a new transport parameter that indicates support of this extension and specifies a new frame type to inform the client of the server's new preferred address.

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

In peer to peer networks, the role of server and client is arbitrary. An endpoint may serve as a client in one connection and a server in another. Limiting connection migration to clients limits the flexibility of endpoints in this network. A peer in this network would like to migrate all of its connections, not just the ones it happens to be a client in.

While it is not the primary goal, this extension may also assist in NAT traversal by migrating to a dynamically chosen server address. A server could have a client connect over a relay, and later migrate to a direct connection after applying NAT traversal techniques. The specific NAT traversal techniques are out of scope of this document.

4. Negotiating Extension Use

new_preferred_address (0xff0969d85c):

Clients advertise their support of this extension by sending the new_preferred_address (0xff0969d85c) transport parameter (Section 7.4 of [QUIC-TRANSPORT]) with an empty value. Sending this transport parameter signals to the server that the client understands the NEW_PREFERRED_ADDRESS frame.

Servers MUST NOT send this transport parameter. A client that supports this extension and receives this transport parameter MUST abort the connection with a TRANSPORT_PARAMETER_ERROR.

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

5. New Preferred Address Frame

A server can use an NEW_PREFERRED_ADDRESS frame to request the client to migrate the connection to the provided server address. Upon receiving an NEW_PREFERRED_ADDRESS, the client MAY initiate migration. If the client does migrate it MUST adhere to the client behavior defined in Section 9.6 of [QUIC-TRANSPORT].

The NEW_PREFERRED_ADDRESS is defined as follows:

```
NEW_PREFERRED_ADDRESS Frame {
  Type (i) = 0x1d5845e2,
  Sequence Number (i),
  IPv4 Address (32),
  IPv4 Port (16),
  IPv6 Address (128),
  IPv6 Port (16),
}
```

Following the common frame format described in Section 12.4 of [QUIC-TRANSPORT], NEW_PREFERRED_ADDRESS frames have a type of 0x1d5845e2, and contain the following fields:

Sequence Number: A variable-length integer representing the sequence number assigned to the NEW_PREFERRED_ADDRESS frame by the sender so receivers can ignore obsolete frames. A sending endpoint MUST send monotonically increasing values in the Sequence Number field to allow obsolete NEW_PREFERRED_ADDRESS frames to be ignored when packets are processed out of order.

IPv4 and IPv6 Address and Port: Analogous to the preferred_address transport parameter, this frame contains an address and port for both IPv4 and IPv6. The four-byte IPv4 Address field is followed by the associated two-byte IPv4 Port field. This is followed by a 16-byte IPv6 Address field and two-byte IPv6 Port field.

NEW_PREFERRED_ADDRESS frames are ack-eliciting, and MUST only be sent in the application data packet number space.

The server SHOULD ensure that its peer has a sufficient number of available and unused connection IDs, as the client will be unable to migrate without an unused connection ID. The server MAY bundle a NEW_CONNECTION_ID frame with the NEW_PREFERRED_ADDRESS. Likewise, the client should ensure the same to allow the server to probe new paths.

6. Security Considerations

6.1. Request Forgery Attacks

The same considerations from Section 21.5 of [QUIC-TRANSPORT] apply here as well.

6.2. DDoS - Thundering herd

A malicious server could wait until it has received a large number of clients, and request a migration from all of them at the same time to a victim endpoint. If the clients all migrate at the same time, they may overload or otherwise negatively impact the victim endpoint.

Clients may mitigate this by randomly delaying the migration.

7. IANA Considerations

7.1. QUIC Transport Parameter

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

Value: 0xff0969d85c

Parameter Name: new_preferred_address

Status: Provisional

Specification: This document

Change Controller: IETF (iesg@ietf.org)

Contact: Marco Munizaga (marco@marcopolo.io)

7.2. QUIC Frame Types

This document registers one new value in the "QUIC Frame Types" registry established in Section 22.4 of [QUIC-TRANSPORT]. The following fields are registered:

Value: 0x1d5845e2

Frame Type Name: NEW_PREFERRED_ADDRESS

Status: Provisional

Specification: This document

Change Controller: IETF (iesg@ietf.org)

Contact: Marco Munizaga (marco@marcopolo.io)

8. 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.

TODOs

Questions

* Any new security conserations from allowing a dynamically chosen preferred address?

* Any new security conserations from allowing a deferred chosen preferred address?

Authors' Addresses

Marco Munizaga
Ethereum Foundation
Email: marco@marcopolo.io

Marten Seemann
Smallstep
Email: martenseemann@gmail.com