QUIC
Internet-Draft
Intended status: Standards Track
Expires: 3 September 2026

M. Munizaga
Ethereum Foundation
M. Seemann
Smallstep

2 March 2026

QUIC Alternative Server Address Frames
draft-munizaga-quic-alternative-server-address-00

Abstract

This document specifies an extension to QUIC to allow a server to advertise alternative addresses.

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/alternative-server-address/draft-munizaga-quic-alternative-server-address.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-munizaga-quic-alternative-server-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/alternative-server-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 3 September 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. A client can use this mechanism to keep redundant paths available or transparently move to a different local address. A server, in contrast, can not use alternative addresses as redundant paths and has no way to dynamically signal a preferred address. In some deployments, specifically peer to peer settings, adding this symmetry is useful.

This document specifies an extension to QUIC that allows a server to inform a client of alternative, possibly preferred, addresses.

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. A peer acting as a server would like to communicate to its peer its alternative addresses. The server peer does this for both redundancy (a peer may advertise a globally reachable relayed unicast address as a backup) and to signal preference (a peer may be using a proxy, and wish to migrate to a new proxy).

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.

TODO: Is the above NAT paragraph useful? Would it be better to leave this implied?

4. Negotiating Extension Use

alternative_address (0xff0969d85c):

Clients advertise their support of this extension by sending the alternative_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 ALTERNATIVE_V4_ADDRESS and ALTERNATIVE_V6_ADDRESS frames.

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. Server initiated Paths

In connections that use this extension, clients MUST NOT discard probing packets received from an unknown server address. Clients MUST validate the path per Section 9.1 of [QUIC-TRANSPORT].

TODO alternatively, should clients treat a server address identified by an alternative address frame as known, and accept probing packets from this address? This would require the server to know its address before hand, which could be annoying if the server is behind a NAT and initially reached over a relay.

6. Alternative Address Frames

A server uses the following frames to inform the client of an alternative address. The Preferred bit signals this address is preferred over the currently in-use server address. The Retire bit signals that this address is no longer an alternative address for this server (TODO what happens if the server sends a Retire bit on the current address?). Clients SHOULD close paths associated with addresses for which the Retire bit is set.

When the Retire bit is not set, clients SHOULD open a path to the provided address. If the Preferred bit is set, clients should migrate to or otherwise prioritze the path with the provided address.

The alternative address frames are defined as follows:

```
ALTERNATIVE_V4_ADDRESS Frame {
  Type (i) = 0x1d5845e2,
  Preferred (1),
  Retire (1),
  unused (6)
  Status Sequence Number (i),
  IPv4 Address (32),
  IPv4 Port (16),
}

ALTERNATIVE_V6_ADDRESS Frame {
  Type (i) = 0x1d5845e3,
  Preferred (1),
  Retire (1),
  unused (6)
  Status Sequence Number (i),
  IPv6 Address (128),
  IPv6 Port (16),
}
```

Following the common frame format described in Section 12.4 of [QUIC-TRANSPORT].

The sequence number space is common to the two frame types, and monotonically increasing values MUST be used when sending updates for a given IP and Port tuple.

TODO: Do we want a probing frame that identifies this path as preferred so it can be used to signal a request to migrate to this path? Do we want to reuse PATH_STATUS_BACKUP or PATH_STATUS_AVAILABLE to harmonize with the Multipath QUIC extension?

7. Frame properties

all 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 probe paths without an unused connection ID. The server MAY bundle a NEW_CONNECTION_ID frame with a alternative address frame. Likewise, the client should ensure the same to allow the server to probe new paths.

8. Interaction with the Multipath Extension for QUIC

This extension compliments the Multipath extension for QUIC by allowing the server to contribute more information to the client for alternative paths.

9. Security Considerations

9.1. Request Forgery Attacks

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

9.2. 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.

10. IANA Considerations

10.1. QUIC Transport Parameter

This document registers the alternative_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: alternative_address

Status: Provisional

Specification: This document

Change Controller: IETF (iesg@ietf.org)

Contact: Marco Munizaga (marco@marcopolo.io)

10.2. QUIC Frame Types

TODO

11. 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

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

Authors' Addresses

Marco Munizaga
Ethereum Foundation
Email: marco@marcopolo.io

Marten Seemann
Smallstep
Email: martenseemann@gmail.com