

This Internet-Draft will expire on 25 April 2024.

Copyright Notice

Copyright (c) 2023 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. Conventions and Definitions
3. Overview
 - 3.1. Example of use
4. Additional Addresses Transport Parameter
5. ADDITIONAL_ADDRESSES Frames
6. Security Considerations
7. IANA Considerations
8. References
 - 8.1. Normative References
 - 8.2. Informative References

Acknowledgments

Authors' Addresses

1. Introduction

The QUIC protocol specifies several techniques for network path migration. The client can migrate from one of its local addresses to another at any time after the handshake using connection migration. The server can transfer a connection to one of its other addresses shortly after the handshake by using the `preferred_address` transport parameter. However, it cannot advertise additional addresses that a client may use.

This limitation impacts several scenarios. For instance, a multihomed server that has access to several subnets cannot advertise all its addresses. In enterprise deployments where provider-assigned IPv6 Addresses are used to solve the multihoming problem [RFC8678], announcing several server addresses enables applications using QUIC to recover from provider failures. Also, a dual-stack server cannot advertise its other address so that a

client losing the address family used to establish the connection can migrate to the other address family.

This document proposes a QUIC frame and a QUIC transport parameter enabling a QUIC server to advertise additional addresses that can be used for a QUIC connection.

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. Overview

The ADDITIONAL_ADDRESSES frame proposed in this document enables a QUIC server to securely advertise additional addresses. The Additional Addresses transport parameter enables a QUIC client to indicate support for this frame.

These addresses can be used by the client to migrate to a new server address at any time after the handshake. When [MULTIPATH-QUIC] is used over a QUIC connection, the client can use these addresses to establish additional network paths.

When sending packets to a new server address, the client validates the address using Path Validation as described in [Section 8.2](#) of [QUIC-TRANSPORT]. When Preferred Address and Additional Addresses are used together, the client **SHOULD NOT** migrate to an additional address before acting on the preferred address indicated by the server.

3.1. Example of use

[Figure 1](#) illustrates an example of use for Additional Addresses in a QUIC deployment featuring a load balancer and a multihomed server making use of the Preferred Address mechanism.

First, the client sends its Initial packet to the load balancer, which forwards it to the first server IP. The client indicates support for this extension by using the dedicated transport parameter. The server answers to the QUIC connection opening and indicates its first IP as a preferred address and its second one as an additional address using the dedicated frame. When the handshake completes, the client validates the preferred address and migrates to it. Later during the connection, the client can validate the path towards the second server IP and can migrate to it.

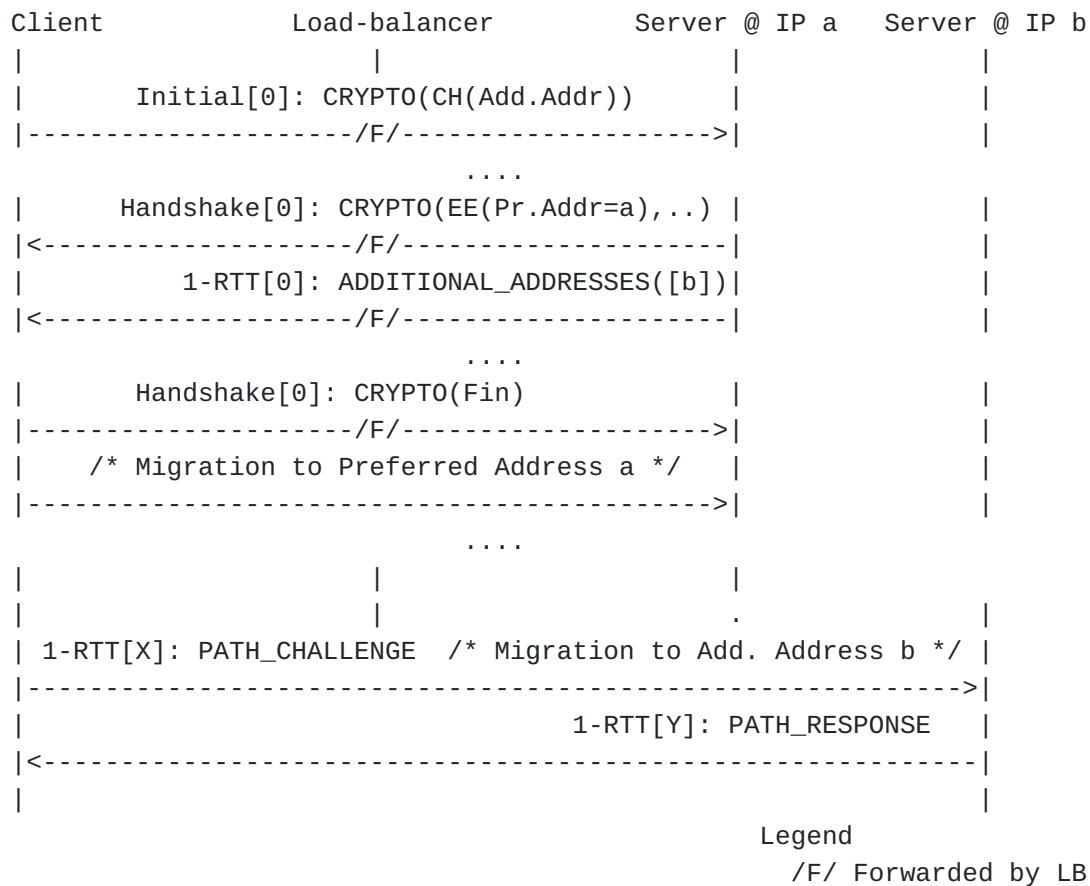


Figure 1: A server reached through a load-balancer uses Add. Address

4. Additional Addresses Transport Parameter

The following transport parameter is defined:

additional_addresses (TBD - experiments use 0x925addaXX): Indicates the support of the ADDITIONAL_ADDRESSES frame as defined in the -XX draft version of this document. This transport parameter **MUST NOT** be sent by a server.

5. ADDITIONAL_ADDRESSES Frames

The server uses an ADDITIONAL_ADDRESSES frame (type=TBD - experiments use 0x925addaXX) to advertise the additional addresses that a client can use to reach it. This frame **MUST NOT** be sent by a client and can only appear in 1-RTT packets.

```
Additional Addresses {
  Type (i) = TBD,
  Sequence Number (i),
  Additional Addresses Count (i),
  Additional Address (..) ...,
}
```

Figure 2: ADDITIONAL_ADDRESSES Frame Format

Sequence Number: A variable-length integer indicating the sequence of the frame. The number is monotonically increasing within a QUIC connection and is chosen by the sender. It helps the receiver to order ADDITIONAL_ADDRESSES frames by recency. A receiver **SHOULD** ignore frames with a Sequence Number lower or equal to the highest Sequence Number received.

Additional Addresses Count: A variable-length integer indicating the number of additional addresses in the frame.

```
Additional Address {  
  Address Version (8),  
  IP Address (..),  
  IP Port (16),  
}
```

Figure 3: Additional Address Format

Address Version: An 8-bit value identifying the Internet address version of this address. The value 4 indicates IPv4 while 6 indicates IPv6.

IP Address: The address value. Its size depends on its version. IPv4 addresses are 32-bit long while IPv6 addresses are 128-bit long.

IP Port: A 16-bit value representing the port to use with this IP Address.

The ADDITIONAL_ADDRESSES frame is ack-eliciting. When a packet containing an ADDITIONAL_ADDRESSES frame is lost and its content is still relevant, the sender **MAY** retransmit the frame as is. Otherwise, sending a new frame with a new Sequence number is preferred.

The server can update the client on its additional addresses at any time by sending an ADDITIONAL_ADDRESSES frame. When a client is using one of these additional addresses and receives an ADDITIONAL_ADDRESSES frame not containing this address, it **SHOULD** stop using it in favor of another address.

6. Security Considerations

This document specifies a mechanism allowing servers to influence the IP addresses towards which clients send QUIC packets. In this case, a malicious server could cause a client to send packets to a victim. A countermeasure similar to [Section 21.5.3](#) of

[QUIC-TRANSPORT] is to limit the packets that are sent to a non-validated additional addresses.

Given that a server can provide additional addresses at any point in time, a malicious server could overload a client and direct it against many addresses. To alleviate this, a client can choose to limit the number of addresses it keeps track of and the frequency at which it considers them.

A client **MUST NOT** send non-probing frames to an additional address prior to validating that address. The generic measures described in [Section 21.5.6](#) of [QUIC-TRANSPORT] also remain applicable for further mitigation.

7. IANA Considerations

This document defines a new transport parameter for indicating support for additional addresses. The draft defines provisional identifiers for experiments. IANA will allocate the final identifiers.

The following entry in [Table 1](#) should be added to the "QUIC Transport Parameters" registry under the "QUIC" heading.

Value	Parameter Name.	Specification
TBD (experiments use 0x925addaXX)	additional_addresses	Section 4

Table 1: Addition to QUIC Transport Parameters Entries

The last byte of the experimental transport parameter ID is used by implementations to indicate the version of this document they support. For instance, the value 0x925adda01 indicates the support of the -01 version of this document.

The following entry in [Table 1](#) should be added to the "QUIC Frame Types" registry under the "QUIC" heading.

Value	Frame Type Name	Specification
TBD (experiments use 0x925addaXX)	ADDITIONAL_ADDRESSES	Section 4

Table 2: Addition to QUIC Frame Types Entries

The last byte of the experimental frame type is used by implementations to indicate the version of this document they support. For instance, the value 0x925adda01 indicates the support of the -01 version of this document.

8. References

8.1. 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>>.

8.2. Informative References

[MULTIPATH-QUIC] Liu, Y., , M. Y., De Coninck, Q., Bonaventure, O., Huitema, C., and M. Kühlewind, "Multipath Extension for QUIC", Work in Progress, Internet-Draft, draft-ietf-quic-multipath-06, 23 October 2023, <<https://datatracker.ietf.org/doc/html/draft-ietf-quic-multipath-06>>.

[RFC8678] Baker, F., Bowers, C., and J. Linkova, "Enterprise Multihoming using Provider-Assigned IPv6 Addresses without Network Prefix Translation: Requirements and Solutions", RFC 8678, DOI 10.17487/RFC8678, December 2019, <<https://www.rfc-editor.org/rfc/rfc8678>>.

Acknowledgments

We thank Quentin De Coninck and François Michel for their feedback and comments on the first version of this document.

Authors' Addresses

Maxime Piraux
UCLouvain & WEL RI

Email: maxime.piraux@uclouvain.be

Olivier Bonaventure
UCLouvain & WEL RI

Email: olivier.bonaventure@uclouvain.be