QUIC
Internet-Draft
Intended status: Standards Track
Expires: 4 September 2025

M. Seemann

C. Huitema
Private Octopus Inc.

3 March 2025

QUIC Address Discovery
draft-ietf-quic-address-discovery-00

Abstract

Unless they have out-of-band knowledge, QUIC endpoints have no information about their network situation. They neither know their external IP address and port, nor do they know if they are directly connected to the internet or if they are behind a NAT. This QUIC extension allows nodes to determine their public IP address and port for any QUIC path.

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

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/quicwg/address-discovery.

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/.

Copyright Notice

Table of Contents

1. Introduction

STUN ([RFC8489]) allows nodes to discover their reflexive transport address by asking a remote server to report the observed source address. While the QUIC ([RFC9000]) packet header was designed to allow demultiplexing from STUN packets, moving address discovery into the QUIC layer has a number of advantages:

1. STUN traffic is unencrypted, and can be observed and modified by on-path observers. By moving address discovery into QUIC's encrypted envelope it becomes invisible to observers.

2. When located behind a load balancer, QUIC packets may be routed based on the QUIC connection ID. Depending on the architecture, not using STUN might simplify the routing logic.

3. If QUIC traffic doesn't need to be demultiplexed from STUN traffic, implementations can enable QUIC bit greasing ([RFC9287]).

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. Negotiating Extension Use

Endpoints advertise their support of the extension by sending the address_discovery (0x9f81a176) transport parameter (Section 7.4 of [RFC9000]) with a variable-length integer value. The value determines the behavior with respect to address discovery:

* 0: The node is willing to provide address observations to its peer, but is not interested in receiving address observations itself.

* 1: The node is interested in receiving address observations, but it is not willing to provide address observations.

* 2: The node is interested in receiving address observations, and it is willing to provide address observations.

Implementations that understand this transport parameter MUST treat the receipt of any other value than these as a connection error of type TRANSPORT_PARAMETER_ERROR.

When using 0-RTT, both endpoints MUST remember the value of this transport parameter. This allows sending the frame defined by this extension in 0-RTT packets. If 0-RTT data is accepted by the server, the server MUST NOT disable this extension or change the value on the resumed connection.

4. Frames

This extension defines the OBSERVED_ADDRESS frame.

4.1. OBSERVED_ADDRESS

```
OBSERVED_ADDRESS Frame {
    Type (i) = 0x9f81a6..0x9f81a7,
    Sequence Number (i),
    [ IPv4 (32) ],
    [ IPv6 (128) ],
    Port (16),
}
```

The OBSERVED_ADDRESS frame contains the following fields:

Sequence Number: A variable-length integer specifying the sequence number assigned for this OBSERVED_ADDRESS frame. The sequence number MUST be monotonically increasing for OBSERVED_ADDRESS frames in the same connection. Frames may be received out of order. A peer SHOULD ignore an incoming OBSERVED_ADDRESS frame if it previously received another OBSERVED_ADDRESS frame for the same path with a Sequence Number equal to or higher than the sequence number of the incoming frame.

IPv4: The IPv4 address. Only present if the least significant bit of the frame type is 0.

IPv6: The IPv6 address. Only present if the least significant bit of the frame type is 1.

Port: The port number, in network byte order.

This frame MUST only appear in the application data packet number space. It is a "probing frame" as defined in Section 9.1 of [RFC9000]. OBSERVED_ADDRESS frames are ack-eliciting, and SHOULD be retransmitted if lost. Retransmissions MUST happen on the same path as the original frame was sent on.

An endpoint MUST NOT send an OBSERVED_ADDRESS frame to a node that did not request the receipt of address observations as described in Section 3. A node that did not request the receipt of address observations MUST close the connection with a PROTOCOL_VIOLATION error if it receives an OBSERVED_ADDRESS frame.

5. Address Discovery

An endpoint that negotiated (see Section 3) this extension and offered to provide address observations to the peer MUST send an OBSERVED_ADDRESS frame on every new path. This also applies to the path used for the QUIC handshake. The OBSERVED_ADDRESS frame SHOULD be sent as early as possible.

For paths used after completion of the handshake, endpoints SHOULD bundle the OBSERVED_ADDRESS frame with probing packets. This is possible, since the frame is defined to be a probing frame (Section 8.2 of [RFC9000]).

Additionally, the sender SHOULD send an OBSERVED_ADDRESS frame when it detects a change in the remote address on an existing path. This could be indicative of a NAT rebinding. However, the sender MAY limit the rate at which OBSERVED_ADDRESS frames are produced, to mitigate the spoofed packets attack described in Section 6.2.

6. Security Considerations

6.1. On the Requester Side

In general, nodes cannot be trusted to report the correct address in OBSERVED_ADDRESS frames. If possible, endpoints might decide to only request address observations when connecting to trusted peers, or if that is not possible, define some validation logic (e.g. by asking multiple untrusted peers and observing if the responses are consistent). This logic is out of scope for this document.

6.2. On the Responder Side

Depending on the routing setup, a node might not be able to observe the peer’s reflexive transport address, and attempts to do so might reveal details about the internal network. In these cases, the node SHOULD NOT offer to provide address observations.

On-path attackers could capture packets sent from the requester to the responder, and resend them from a spoofed source address. If done repeatedly, these spoofed packets could trigger the sending of a large number of OBSERVED_ADDRESS frames. The recommendation to only include OBSERVED_ADDRESS frames in packets sent on the same path over which the address was observed ensures that the peer will not receive the OBSERVED_ADDRESS frames if the addresses are not valid, but this does not reduce the number of packets sent over the network. The attack also has the effect of causing spurious detection NAT rebinding, and is a variant of the replacement of addresses of packets mentioned in Section 21.1.1.3 of [RFC9000]. QUIC

implementations are expected to have sufficient protection against spurious NAT rebinding to limit the incidental traffic caused by such attacks. The same protection logic SHOULD be used to prevent sending of a large number of spurious OBSERVED_ADDRESS frames.

7. IANA Considerations

TODO: fill out registration request for the transport parameter and frame types

8. References

8.1. Normative References

[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>.

[RFC8489] Petit-Huguenin, M., Salgueiro, G., Rosenberg, J., Wing, D., Mahy, R., and P. Matthews, "Session Traversal Utilities for NAT (STUN)", RFC 8489, DOI 10.17487/RFC8489, February 2020, <https://www.rfc-editor.org/rfc/rfc8489>.

[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/rfc/rfc9000>.

8.2. Informative References

[I-D.pauly-quic-address-extension] Pauly, T., Wood, C. A., and E. Kinnear, "QUIC Address Extension", Work in Progress, Internet-Draft, draft-pauly-quic-address-extension-00, 11 March 2019, <https://datatracker.ietf.org/doc/html/draft-pauly-quic-address-extension-00>.

[RFC9287] Thomson, M., "Greasing the QUIC Bit", RFC 9287, DOI 10.17487/RFC9287, August 2022, <https://www.rfc-editor.org/rfc/rfc9287>.

Acknowledgments

Unbeknownst to the authors, the idea of moving address discovery into QUIC was conveived of before in [I-D.pauly-quic-address-extension].

Authors' Addresses

Marten Seemann
Email: martenseemann@gmail.com

Christian Huitema
Private Octopus Inc.
Email: huitema@huitema.net