ï»¿

Internet Engineering Task Force (IETF)
Request for Comments: 9287
Category: Standards Track
ISSN: 2070-1721

M. Thomson
Mozilla

August 2022

Greasing the QUIC Bit

Abstract

This document describes a method for negotiating the ability to send an arbitrary value for the second-most significant bit in QUIC packets.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc9287.

Table of Contents

1. Introduction

The version-independent definition of QUIC [QUIC-INVARIANTS] intentionally describes a very narrow set of fields that are visible to entities other than endpoints. Beyond those characteristics that are invariant, very little about the "wire image" [RFC8546] of QUIC is visible.

The second-most significant bit of the first byte in every QUIC packet is defined as having a fixed value in QUIC version 1 [QUIC].

The purpose of having a fixed value is to allow endpoints to efficiently distinguish QUIC from other protocols; see [DEMUX] for a description of a system that might use this property. As this bit can identify a packet as QUIC, it is sometimes referred to as the "QUIC Bit".

Where endpoints and the intermediaries that support them do not depend on the QUIC Bit having a fixed value, sending the same value in every packet is more of a liability than an asset. If systems come to depend on a fixed value, then it might become infeasible to define a version of QUIC that attributes semantics to this bit.

In order to safeguard future use of this bit, this document defines a QUIC transport parameter that indicates that an endpoint is willing to receive QUIC packets containing any value for this bit. By sending different values for this bit, the hope is that the value will remain available for future use [USE-IT].

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.

This document uses terms and notational conventions from [QUIC].

3. The Grease QUIC Bit Transport Parameter

The grease_quic_bit transport parameter (0x2ab2) is defined for QUIC version 1 [QUIC]. This transport parameter can be sent by both client and server. The transport parameter is sent with an empty value; an endpoint that understands this transport parameter MUST treat receipt of a non-empty value of the transport parameter as a connection error of type TRANSPORT_PARAMETER_ERROR.

An endpoint that advertises the grease_quic_bit transport parameter MUST accept packets with the QUIC Bit set to a value of 0. The QUIC Bit is defined as the second-most significant bit of the first byte of QUIC packets (that is, the value 0x40).

3.1. Clearing the QUIC Bit

Endpoints that receive the grease_quic_bit transport parameter from a peer SHOULD set the QUIC Bit to an unpredictable value unless another extension assigns specific meaning to the value of the bit.

Endpoints can set the QUIC Bit to 0 on all packets that are sent after receiving and processing transport parameters. This could include Initial, Handshake, and Retry packets.

A client MAY also set the QUIC Bit to 0 in Initial, Handshake, or 0-RTT packets that are sent prior to receiving transport parameters from the server. However, a client MUST NOT set the QUIC Bit to 0 unless the Initial packets it sends include a token provided by the server in a NEW_TOKEN frame (Section 19.7 of [QUIC]), received less than 604800 seconds (7 days) prior on a connection where the server also included the grease_quic_bit transport parameter.

> This 7-day limit allows for changes in server configuration. If server configuration changes and a client does not set the QUIC Bit, then it is possible that a server will drop packets, resulting in connection failures.

A server MUST set the QUIC Bit to 0 only after processing transport parameters from a client. A server MUST NOT remember that a client negotiated the extension in a previous connection and set the QUIC Bit to 0 based on that information.

An endpoint MUST NOT set the QUIC Bit to 0 without knowing whether

the peer supports the extension. As Stateless Reset packets (Section 10.3 of [QUIC]) are only used after a loss of connection state, endpoints are unlikely to be able to set the QUIC Bit to 0 on Stateless Reset packets.

3.2. Using the QUIC Bit

The purpose of this extension is to allow for the use of the QUIC Bit by later extensions.

Extensions to QUIC that define semantics for the QUIC Bit can be negotiated at the same time as the grease_quic_bit transport parameter. In this case, a recipient needs to be able to distinguish a randomized value from a value carrying information according to the extension. Extensions that use the QUIC Bit MUST negotiate their use prior to acting on any semantic.

For example, an extension might define a transport parameter that is sent in addition to the grease_quic_bit transport parameter. Though the value of the QUIC Bit in packets received by a peer might be set according to rules defined by the extension, they might also be randomized as specified in this document.

The receipt of a transport parameter for an extension that uses the QUIC Bit could be used to confirm that a peer supports the semantic defined in the extension. To avoid acting on a randomized signal, the extension can require that endpoints set the QUIC Bit according to the rules of the extension but defer acting on the information conveyed until the transport parameter for the extension is received.

Extensions that define semantics for the QUIC Bit can be negotiated without using the grease_quic_bit transport parameter. However, including both extensions allows for the QUIC Bit to be greased even if the alternative use is not supported.

4. Security Considerations

This document introduces no new security considerations for endpoints or entities that can rely on endpoint cooperation. However, this change makes the task of identifying QUIC more difficult without cooperation of endpoints. This sometimes works counter to the security goals of network operators who rely on network classification to identify threats; see Section 3.1 of [MANAGEABILITY] for a more comprehensive treatment of this topic.

5. IANA Considerations

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

Value: 0x2ab2

Parameter Name: grease_quic_bit

Status: Permanent

Specification: RFC 9287

Date: 2022-07-13

Change Controller: IETF (iesg@ietf.org)

Contact: QUIC Working Group (quic@ietf.org)

Notes: (none)

6. References

6.1. Normative References

[QUIC] 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/info/rfc9000>.

[QUIC-INVARIANTS] Thomson, M., "Version-Independent Properties of QUIC", RFC 8999, DOI 10.17487/RFC8999, May 2021, <https://www.rfc-editor.org/info/rfc8999>.

[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/info/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/info/rfc8174>.

6.2. Informative References

[DEMUX] Aboba, B., Salgueiro, G., and C. Perkins, "Multiplexing Scheme Updates for QUIC", Work in Progress, Internet-Draft, draft-ietf-avtcore-rfc7983bis-06, 5 August 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-avtcore-rfc7983bis-06>.

[MANAGEABILITY] Kuehlewind, M. and B. Trammell, "Manageability of the QUIC Transport Protocol", Work in Progress, Internet-Draft, draft-ietf-quic-manageability-18, 15 July 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-quic-manageability-18>.

[RFC8546] Trammell, B. and M. Kuehlewind, "The Wire Image of a Network Protocol", RFC 8546, DOI 10.17487/RFC8546, April 2019, <https://www.rfc-editor.org/info/rfc8546>.

[USE-IT] Thomson, M. and T. Pauly, "Long-Term Viability of Protocol Extension Mechanisms", RFC 9170, DOI 10.17487/RFC9170, December 2021, <https://www.rfc-editor.org/info/rfc9170>.

Author's Address

Martin Thomson
Mozilla
Email: mt@lowentropy.net