

quic
Internet-Draft
Intended status: Standards Track
Expires: 14 May 2022

M. Thomson
Mozilla
10 November 2021

Greasing the QUIC Bit
draft-ietf-quic-bit-grease-02

Abstract

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

Discussion Venues

This note is to be removed before publishing as an RFC.

Discussion of this document takes place on the QUIC Working Group mailing list (quic@ietf.org), which is archived at <https://mailarchive.ietf.org/arch/browse/quic/> (<https://mailarchive.ietf.org/arch/browse/quic/>).

Source for this draft and an issue tracker can be found at <https://github.com/quicwg/quic-bit-grease> (<https://github.com/quicwg/quic-bit-grease>).

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 14 May 2022.

Internet-Draft

Greasing the QUIC Bit

November 2021

Copyright Notice

Copyright (c) 2021 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 Simplified BSD License text as described in Section 4.e of the [Trust Legal Provisions](#) and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	2
2.	Conventions and Definitions	3
3.	The Grease QUIC Bit Transport Parameter	3
3.1.	Clearing the QUIC Bit	3
3.2.	Using the QUIC Bit	4
4.	Security Considerations	4
5.	IANA Considerations	5
6.	References	5
6.1.	Normative References	5
6.2.	Informative References	5
	Author's Address	6

[1.](#) Introduction

QUIC [[QUIC](#)] intentionally describes a very narrow set of fields that are visible to entities other than endpoints. Beyond those characteristics that are defined as invariant [[QUIC-INVARIANTS](#)], very little about the "wire image" [[RFC8546](#)] of QUIC is visible.

The second-to-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 intermediaries and endpoints to efficiently distinguish between QUIC and other protocols; see [[DEMUX](#)] for a description of a scheme that QUIC can integrate with as a result. As this bit effectively identifies 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 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) 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 as a connection error of type `TRANSPORT_PARAMETER_ERROR`.

Advertising the `grease_quic_bit` transport parameter indicates that packets sent to this endpoint MAY set a value of 0 for the QUIC Bit. The QUIC Bit is defined as the second-to-most significant bit of the first byte of QUIC packets (that is, the value 0x40).

A server MUST respect the value it previously provided for the `grease_quic_bit` transport parameter if it accepts 0-RTT. A client MAY forget the value. In all other cases, only the presence or absence of the transport parameter in the current handshake is used to determine what values can be sent in the QUIC Bit.

[3.1.](#) Clearing the QUIC Bit

Endpoints that receive the `grease_quic_bit` transport parameter from a peer MAY set the QUIC Bit to any value in packets they send to that peer. Endpoints SHOULD set the QUIC Bit to an unpredictable value unless another extension assigns specific meaning to the value of the bit. All packets sent after receiving and processing transport parameters are affected, including Retry, Initial, and Handshake packets.

A client MAY also clear the QUIC Bit in Initial packets that are sent prior to receiving transport parameters from the server. A client can only clear the QUIC Bit if such packets include a token provided

by the server in a `NEW_TOKEN` frame on a connection where the server also included the `grease_quic_bit` transport parameter. To allow for changes in server configuration, clients SHOULD set the QUIC Bit if the token was provided more than 7 days prior.

[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. Endpoints MAY send a signal prior to this negotiation completing, but any value carried by the bit cannot be used until it is clear that the peer is using the extension.

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. Including both extensions allows for the QUIC Bit to be greased even if the alternative use is not supported.

Receiving 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.

[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.

[5.](#) IANA Considerations

This document registers the `grease_quic_bit` transport parameter in the "QUIC Transport Parameters" registry established in Section 22.2 of [[QUIC](#)]. The following fields are registered:

Value: 0x2ab2

Parameter Name: `grease_quic_bit`

Status: Permanent

Specification: This document.

Date: Date of registration.

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

Change Controller: IETF (iesg@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/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>>.

6.2. Informative References

- [DEMUX] Petit-Huguenin, M. and G. Salgueiro, "Multiplexing Scheme Updates for Secure Real-time Transport Protocol (SRTP) Extension for Datagram Transport Layer Security (DTLS)", [RFC 7983](#), DOI 10.17487/RFC7983, September 2016, <<https://www.rfc-editor.org/rfc/rfc7983>>.

Thomson

Expires 14 May 2022

[Page 5]

Internet-Draft

Greasing the QUIC Bit

November 2021

- [QUIC-INVARIANTS] Thomson, M., "Version-Independent Properties of QUIC", [RFC 8999](#), DOI 10.17487/RFC8999, May 2021, <<https://www.rfc-editor.org/rfc/rfc8999>>.
- [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/rfc/rfc8546>>.
- [USE-IT] Thomson, M. and T. Pauly, "Long-term Viability of Protocol Extension Mechanisms", Work in Progress, Internet-Draft, [draft-ietf-use-it-or-lose-it-04](#), 12 October 2021, <<https://datatracker.ietf.org/doc/html/draft-ietf-use-it-or-lose-it-04>>.

Author's Address

Martin Thomson
Mozilla

Email: mt@lowentropy.net