

Workgroup: MASQUE
Internet-Draft:
draft-pardue-masque-dgram-priority-01
Published: 26 July 2021
Intended Status: Experimental
Expires: 27 January 2022
Authors: L. Pardue
Cloudflare

HTTP Datagram Prioritization

Abstract

Application protocols using the QUIC transport protocol rely on streams, and optionally the DATAGRAM extension, to carry application data. Streams and datagrams can be multiplexed but QUIC provides no interoperable prioritization scheme or signaling mechanism itself. The HTTP Extensible Prioritization scheme describes how to prioritize streams in HTTP/2 and HTTP/3. This document adopts the scheme to support HTTP datagrams.

Note to Readers

RFC EDITOR: please remove this section before publication

Source code and issues list for this draft can be found at <https://github.com/LPardue/draft-pardue-masque-dgram-priority>.

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 27 January 2022.

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](#)
 - [1.1. Notational Conventions](#)
- [2. Signalling Datagram Priority](#)
 - [2.1. Datagram Urgency](#)
 - [2.2. Prioritization of Contexts](#)
 - [2.3. Reprioritization](#)
- [3. Client Scheduling](#)
- [4. Server Scheduling](#)
- [5. Retransmission Scheduling](#)
- [6. Security Considerations](#)
- [7. IANA Considerations](#)
- [8. References](#)
 - [8.1. Normative References](#)
 - [8.2. Informative References](#)
- [Appendix A. Acknowledgements](#)
- [Author's Address](#)

1. Introduction

Application protocols using the QUIC transport protocol [[QUIC](#)] rely on streams, and optionally the DATAGRAM extension [[QUIC-DATAGRAM](#)], to carry application data. Streams and datagrams can be multiplexed but QUIC provides no interoperable prioritization scheme or signaling mechanism itself. The HTTP Extensible Prioritization scheme [[I-D.ietf-httpbis-priority](#)] describes how to prioritize streams in HTTP/2 and HTTP/3. This document adopts the scheme to support HTTP datagrams [[HTTP-DATAGRAM](#)].

The Extensible Priorities scheme for HTTP describes how clients can send priority signals related to requests in order to suggest how a server allocates resources to serving responses. When the protocol is HTTP/2, responses are carried on streams. When the protocol is HTTP/3, responses are carried on QUIC streams.

While QUIC streams support multiplexing natively via use of a stream identifier, the QUIC DATAGRAM extension does not provide any such identifier. HTTP datagrams [[HTTP-DATAGRAM](#)] supports multiplexing

using a set of application-level identifiers that can be controlled and accessed by HTTP/3. One identifier relates to a request stream, the second, optional, identifier relates to an abstract context. [\[HTTP-DATAGRAM\]](#) does not, however, define any means for multiplexed datagram prioritization.

When the application protocol is HTTP/3, HTTP Datagrams can map directly to QUIC datagrams or they can be carried on streams using a DATAGRAM Capsule; see [Section 4.4](#) of [\[HTTP-DATAGRAM\]](#).

This document describes how the Extensible Priorities scheme applies to HTTP datagrams. Priority signals sent by clients, related to requests, can also be considered input to server scheduling decisions for HTTP datagrams mapped to QUIC datagrams.

1.1. Notational Conventions

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.

The term sf-integer is imported from [\[STRUCTURED-FIELDS\]](#).

2. Signalling Datagram Priority

The Extensible Prioritization scheme [\[I-D.ietf-httpbis-priority\]](#) provides a framework for communicating and acting upon priority parameters, using [\[STRUCTURED-FIELDS\]](#) formats. It defines the urgency and incremental parameters and provides guidance to implementers about how to act on these parameters, in combination with other inputs, to make resource allocation and scheduling choices. Urgency communicates the client-view of request importance, and incremental communicates how the client intends to process response data as it arrives. Parameters are communicated in HTTP headers or version-specific frames. A client omitting the urgency or incremental parameters can be interpreted by the server as a signal to apply default priorities. The core scheme is extensible, new parameters can be defined to augment the base ones.

This specification defines the datagram-urgency (du) extension parameter that operates in addition to the base urgency. There is no extension to the base incremental behavior; individual datagrams, even if belonging to the same identifier, are messages that are expected to be processed individually as they arrive.

2.1. Datagram Urgency

The datagram-urgency parameter (du) takes an integer between 0 and 7, in descending order of priority. This range matches the base urgency (u) parameter range; see Section 4.1 of [[I-D.ietf-httpbis-priority](#)].

The value is encoded as an sf-integer. There is no default value.

This parameter indicates the sender's recommendation, based on the expectation that the server would transmit HTTP datagrams in the order of their datagram-urgency values if possible. The smaller the value, the higher the precedence. Omitting the datagram-urgency parameter is a signal to apply the value of the urgency parameter.

The following example shows a request for a CSS file with the urgency set to 0, any associated datagrams have the lower urgency of 2:

```
:method = GET
:scheme = https
:authority = example.net
:path = /style.css
priority = u=0, du=2
```

Endpoints MUST NOT treat reception of the datagram-urgency parameter, even if HTTP datagram support is not enabled.

The datagram-urgency parameter applies only to HTTP datagrams mapped to QUIC datagrams. Datagram capsules are sent on streams, so the base urgency parameter applies to them.

2.2. Prioritization of Contexts

The datagram-urgency parameter applies to all HTTP datagram contexts related to a request stream. Prioritization of individual contexts is not supported.

2.3. Reprioritization

Reprioritization is supported using the existing mechanisms defined in Section 6 of [[I-D.ietf-httpbis-priority](#)].

3. Client Scheduling

Clients MAY use datagram-urgency to make local processing or scheduling choices about HTTP datagrams related to the requests it initiates.

4. Server Scheduling

Priority signals are input to a prioritization process. Expressing priority is only a suggestion. The datagram-urgency parameter introduces new scheduling considerations on top of those presented in Section 10 of [[I-D.ietf-httpbis-priority](#)].

It is RECOMMENDED that, when possible, servers send higher urgency HTTP datagrams before lower urgency datagrams.

Where streams and datagrams have equal urgency and datagram-urgency, it is RECOMMENDED that servers alternate emitting HTTP datagrams and stream bytes. Where servers implement the recommendations in Section 10 of [[I-D.ietf-httpbis-priority](#)], alternating between datagram and stream data will result in fair scheduling. This recommendation holds whether stream are incremental or not.

It is RECOMMENDED that servers schedule DATAGRAM capsules the same as response data.

5. Retransmission Scheduling

Section 12 of [[I-D.ietf-httpbis-priority](#)] provides guidance about scheduling of retransmission data vs. new data. Since QUIC datagrams are not retransmitted, endpoints that prioritize QUIC stream retransmission data could delay datagrams. Furthermore, since DATAGRAM capsules are sent as stream data, they **are** subject to retransmission and could also delay native QUIC datagrams.

6. Security Considerations

There are believed to be no additional considerations to those presented in [[I-D.ietf-httpbis-priority](#)].

7. IANA Considerations

This specification registers the following entry in the HTTP Priority Parameters Registry

Name: datagram-urgency

Description: Priority of HTTP datagrams

Reference: This document

8. References

8.1. Normative References

[HTTP-DATAGRAM]

Schinazi, D. and L. Pardue, "Using Datagrams with HTTP", Work in Progress, Internet-Draft, draft-ietf-masque-h3-datagram-03, 12 July 2021, <<https://www.ietf.org/archive/id/draft-ietf-masque-h3-datagram-03.txt>>.

[I-D.ietf-httpbis-priority] Oku, K. and L. Pardue, "Extensible Prioritization Scheme for HTTP", Work in Progress, Internet-Draft, draft-ietf-httpbis-priority-04, 11 July 2021, <<https://www.ietf.org/archive/id/draft-ietf-httpbis-priority-04.txt>>.

[QUIC-DATAGRAM] Pauly, T., Kinnear, E., and D. Schinazi, "An Unreliable Datagram Extension to QUIC", Work in Progress, Internet-Draft, draft-ietf-quic-datagram-03, 12 July 2021, <<https://www.ietf.org/archive/id/draft-ietf-quic-datagram-03.txt>>.

[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>>.

[STRUCTURED-FIELDS] Nottingham, M. and P-H. Kamp, "Structured Field Values for HTTP", RFC 8941, DOI 10.17487/RFC8941, February 2021, <<https://www.rfc-editor.org/info/rfc8941>>.

8.2. Informative 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>>.

Appendix A. Acknowledgements

This document is inspired by discussion by many people across HTTP, QUIC and MASQUE WGs.

Author's Address

Lucas Pardue
Cloudflare

Email: lucaspardue.24.7@gmail.com