

Workgroup: Network Working Group
Internet-Draft:
draft-hurst-quick-http-data-offset-frame-01
Published: 13 August 2021
Intended Status: Experimental
Expires: 14 February 2022
Authors: S. Hurst
BBC Research & Development
An Offset Extension Frame For HTTP/3 Data

Abstract

This document specifies an optional extension frame type for HTTP/3 that extends the functionality of the DATA frame type to include an offset for the HTTP message payload. This is useful in situations where the HTTP/3 exchange is taking place over an unreliable transport mechanism.

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 February 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](#)
- [2. Conventions and Terminology](#)
- [3. DATA WITH OFFSET Extension Frame](#)
- [4. Realising HTTP Multipart Range Responses With HTTP/3 Binary Framing](#)
 - [4.1. Response Headers](#)
 - [4.2. Usage of DATA WITH OFFSET frame with HTTP Range Responses](#)
- [5. Negotiating Support For The DATA WITH OFFSET Frame](#)
- [6. Security Considerations](#)
- [7. IANA Considerations](#)
- [8. References](#)
 - [8.1. Normative References](#)
 - [8.2. Informative References](#)
- [Appendix A. Acknowledgements](#)
- [Appendix B. Changelog](#)
 - [B.1. Changes since draft-hurst-quic-http-data-offset-frame-00](#)
- [Author's Address](#)

1. Introduction

HTTP/3 [[QUIC-HTTP](#)] supports the transfer of HTTP semantics over the QUIC transport protocol [[RFC9000](#)]. In a conventional HTTP/3 message exchange, messages consist of a header field section sent as a single HEADERS frame, an optional HTTP message payload sent as a series of DATA frames, followed optionally by a trailer field section sent as a single HEADERS frame. Each DATA frame does not describe its position within the HTTP message payload; rather this is calculated from the position within the QUIC stream minus the overhead from HTTP/3 frame headers and the contents of the header field section.

In the case where the message exchange is taking place across a partially reliable or unreliable profile of [[RFC9000](#)], packet loss could result in a lack of synchronisation in the receiver between the perceived HTTP/3 DATA frame offset and the QUIC STREAM frame offset, potentially resulting in a corrupt HTTP representation at the receiver.

In addition, there are other use cases, such as HTTP multipart range requests, where the HTTP/3 payload offset has no direct mapping to the value calculated by the method described above.

This document introduces an extension frame type DATA_WITH_OFFSET which can be used to explicitly signal the offset in the original representation of the data being conveyed within the frame.

2. Conventions and Terminology

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 the variable-length integer encoding from [[RFC9000](#)]. The packet and frame diagrams in this document use the bespoke format specified in [[RFC9000](#)].

3. DATA_WITH_OFFSET Extension Frame

Based on the DATA frame defined in [[QUIC-HTTP](#)], the DATA_WITH_OFFSET frame conveys arbitrary, variable-length sequences of bytes at a defined offset of an HTTP representation. By carrying an explicit payload offset in the HTTP/3 frame header, the HTTP message payload offset is decoupled from the QUIC STREAM frame header offset value. The additional payload offset field takes the form of a variable-length integer, as shown in [Figure 1](#) below.

```
DATA_WITH_OFFSET Frame {  
  Type (i) = 0xd00,  
  Length (i),  
  Offset (i),  
  Data (..),  
}
```

Figure 1: DATA_WITH_OFFSET Frame

If its peer has indicated support for the DATA_WITH_OFFSET extension frame type (as described in [Section 5](#) below) a sender MAY choose to use either DATA frames or DATA_WITH_OFFSET frames to transmit an HTTP representation. Senders MUST NOT mix the use of DATA and DATA_WITH_OFFSET frames on the same QUIC stream (i.e. in the same HTTP message).

Author's Note: The author welcomes comments about relaxation of the requirement to not mix the usage of DATA and DATA_WITH_OFFSET frames in the same HTTP message.

[[QUIC-HTTP](#)] defines three stream types: control stream, request stream and push stream. The DATA_WITH_OFFSET frame is only permitted to appear on request streams and push streams. A DATA_WITH_OFFSET frame MUST NOT appear on control streams. If a DATA_WITH_OFFSET frame is received on a control stream, the recipient MUST respond with a connection error of type H3_FRAME_UNEXPECTED, as described in [[QUIC-HTTP](#)].

The purpose of the DATA_WITH_OFFSET frame is only to assist in locating a particular slice of data carried as part of an HTTP message payload, and not as a means to send data out of order. Senders MUST send data in order, i.e. with increasing values in the Offset field. In cases where the underlying transport does not guarantee in-order delivery of HTTP/3 frames, the receiver MUST be prepared to deal with out-of-order reception of DATA_WITH_OFFSET frames.

4. Realising HTTP Multipart Range Responses With HTTP/3 Binary Framing

HTTP Range Requests, described in [[HTTP-SEMANTICS](#)], is an optional feature of HTTP that allows a client to request transfer of one or more subranges of a given representation. Despite the move to binary framing for HTTP in [[RFC7540](#)] and subsequently [[QUIC-HTTP](#)], multiple part HTTP Range Requests still rely on textual encoding - including boundary strings - which is inefficient. These boundary strings then preface additional HTTP headers for each body part, which always carry a Content-Range field, and may additionally carry a Content-Type field which is likely to be repeated across every body part. It is not possible to compress these headers using [[QUIC-QPACK](#)].

By using the DATA_WITH_OFFSET frame described in [Section 3](#), the network efficiency of multiple part range requests is improved by no longer needing to encode a boundary string into the representation response. Instead, the offset of each part of a representation is simply encoded in the header of a fresh HTTP DATA_WITH_OFFSET frame.

4.1. Response Headers

[[HTTP-SEMANTICS](#)] specifies how a server may respond to an HTTP multipart range request using the 206 (Partial Content) status code. The response message carries a Content-Type response header indicating the multipart/byteranges media type with its required boundary parameter. This boundary parameter allows each body part to carry its own header area containing a Content-Range header to describe what range of the selected representation this body part conveys, as well as a Content-Type header (if applicable) which describes the actual media type of the selected representation.

(Note that section 14.2 of [[HTTP-SEMANTICS](#)] describes several reasons why a server may choose to deliver a different selection of parts than what the client originally requested.)

Because a selected representation may only contain a single Content-Type header field with a single value, repeating this header field in every body part is highly inefficient. Moreover, the unbounded length of the boundary parameter further reduces transmission efficiency.

This specification modifies the syntax of the Content-Range header and explicitly defines it as a list-based field as per section 5.6.1 of [\[HTTP-SEMANTICS\]](#) that is carried in the first HEADERS block sent as part of an HTTP/3 response. In addition, when used on the same QUIC stream as DATA_WITH_OFFSET frames, this specification permits the Content-Range and Content-Type HTTP headers to appear in the HEADERS frame of a 206 (Partial Content) response, enabling it to indicate the MIME media type of the whole representation without needing to duplicate it for each body part.

```
Content-Range      = 1#range-item
range-item         = range-unit SP
                   ( range-resp / unsatisfied-range )

range-resp         = incl-range "/" ( complete-length / "*" )
incl-range         = first-pos "-" last-pos
unsatisfied-range  = "*" / complete-length

complete-length    = 1*DIGIT
```

Figure 2: ABNF for extended Content-Range

```
:status = 206
content-type = video/mp4
content-range = bytes 10000-17999/18879543, bytes 24000-41999/18879543
```

Figure 3: Range-Response header example

Implementations advertising support for the DATA_WITH_OFFSET frame as described in [Section 5](#) MUST be able to consume this overloaded form of the Content-Range HTTP response header.

A server MAY continue to use the method described in [\[HTTP-SEMANTICS\]](#) even if a client has expressed support for the DATA_WITH_OFFSET frame.

Author's Note: Is it possibly worth splitting this out into its own HTTP setting value?

4.2. Usage of DATA_WITH_OFFSET frame with HTTP Range Responses

The DATA_WITH_OFFSET frame may be used in conjunction with HTTP Range Requests, as described in [\[HTTP-SEMANTICS\]](#). When carrying data for a byte range response, the Offset field in the frame header MUST reflect the starting byte position of the frame's payload in the HTTP representation and not the offset within the HTTP/3 exchange. For example, for an HTTP/3 range request made with a request header of range: bytes=1000-1999, the first DATA_WITH_OFFSET frame in the

response MUST carry the value 1000 in the Offset field of the frame header.

For HTTP response messages carrying a set of byte ranges, a DATA_WITH_OFFSET frame MUST NOT carry data for more than one contiguous byte range within that set. An individual byte range MAY be carried over multiple instances of the DATA_WITH_OFFSET frame.

5. Negotiating Support For The DATA_WITH_OFFSET Frame

The DATA_WITH_OFFSET frame described in [Section 3](#) is an optional extension to the regular HTTP/3 protocol specification and, as such, usage of the frame type must be negotiated as described in section 9 of [\[QUIC-HTTP\]](#). For a conventional HTTP/3 connection, this is done using HTTP/3 SETTINGS frames carried on the control streams.

This specification defines the following setting:

SETTINGS_ENABLE_DATA_WITH_OFFSET_FRAME (0xd00): A boolean value with a default value of 0 (false). Any non-zero value is true.

Author's Note: It is intended that the value of the H3 SETTINGS frame should mirror the value of the frame to indicate which version of the DATA_WITH_OFFSET frame it understands, should subsequent revisions of this draft change the frame type.

An endpoint that implements this specification SHOULD send a SETTINGS_ENABLE_DATA_WITH_OFFSET_FRAME setting at the beginning of the connection to indicate that it is able to process DATA_WITH_OFFSET frames from its peer.

An endpoint MUST NOT send a DATA_WITH_OFFSET frame unless it has received a positive (i.e. non-zero) SETTINGS_ENABLE_DATA_WITH_OFFSET_FRAME setting from its peer.

6. Security Considerations

This document introduces no new security considerations beyond those discussed in [\[QUIC-HTTP\]](#).

7. IANA Considerations

This specification registers a new frame type in the "HTTP/3 Frame Type" registry ([\[QUIC-HTTP\]](#)).

Frame Type	Value	Specification
DATA_WITH_OFFSET	0xd00	Section 3

Table 1: Registered HTTP/3 Frame Type

Author's Note: The final, intended value of the frame type is 0xd0f, but in order to allow for this extension to naturally evolve and allow for the frame format to change, it starts at 0xd00 and subsequent revisions of this extension can take incrementally higher frame type values between 0xd00 and 0xd0e.

This specification registers a new setting in the "HTTP/3 Settings" registry ([[QUIC-HTTP](#)]).

Setting	Value	Specification	Default
SETTINGS_ENABLE_DATA_WITH_OFFSET_FRAME	0xd00	Section 5	0

Table 2: Registered HTTP/3 Settings

8. References

8.1. Normative References

- [**HTTP-SEMANTICS**] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "HTTP Semantics", Work in Progress, Internet-Draft, draft-ietf-httpbis-semantics-17, <<https://datatracker.ietf.org/doc/html/draft-ietf-httpbis-semantics-17>>.
- [**QUIC-HTTP**] Bishop, M., Ed., "Hypertext Transfer Protocol Version 3 (HTTP/3)", Work in Progress, Internet-Draft, draft-ietf-quic-http-34, <<https://datatracker.ietf.org/doc/html/draft-ietf-quic-http-34>>.
- [**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>>.
- [**RFC7540**] Belshé, M., Peon, R., and M. Thomson, Ed., "Hypertext Transfer Protocol Version 2 (HTTP/2)", RFC 7540, DOI 10.17487/RFC7540, May 2015, <<https://www.rfc-editor.org/info/rfc7540>>.
- [**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>>.
- [**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/info/rfc9000>>.

8.2. Informative References

[QUIC-QPACK] Crasic, C., Ed., Bishop, M., Ed., and A. Frindell, Ed., "QPACK: Header Compression for HTTP/3", Work in Progress, Internet-Draft, draft-ietf-quic-qpack-21, <<https://datatracker.ietf.org/doc/html/draft-ietf-quic-qpack-21>>.

Appendix A. Acknowledgements

The author would like to thank the following for their contributions to the design described in the present document: Lucas Pardue, Richard Bradbury and David Waring.

I am also grateful for Chris Poole's helpful review comments.

Appendix B. Changelog

RFC Editor's Note: Please remove this section prior to publication of a final version of this document.

B.1. Changes since draft-hurst-quic-http-data-offset-frame-00

*Update reference to QUIC transport I-D to [[RFC9000](#)].

*Update reference to [[HTTP-SEMANTICS](#)] I-D.

Author's Address

Sam Hurst
BBC Research & Development

Email: sam.hurst@bbc.co.uk