

MASQUE

HTTP Datagram Priorities

[draft-pardue-masque-dgram-priority](#)

IETF 111 – Virtual San Francisco – 2021-07

Lucas Pardue – lucaspardue.24.7@gmail.com

Background

RFC 9000 says stream multiplexing can have a significant effect on application performance if resources allocated to streams are correctly prioritized. QUIC does not provide a mechanism for exchanging prioritization information. Instead, it relies on receiving priority information from the application.

Draft-ietf-quic-datagram defines QUIC DATAGRAM frames. There is no multiplexing identifier but presumably DATAGRAM frames can be muxed with STREAM frames.

HTTP/3 doesn't define any priority signal, it punts to draft-ietf-httpbis-priority.

Draft-ietf-masque-h3-datagram defines datagram multiplexing but doesn't say anything about priority.

Bouncing around

- H3 Dgram Issue [#46](#) – The spec should discuss how h3-datagram works(or does not) with priority.

- Closed with a PR that says:

Prioritization of HTTP/3 datagrams is not defined in this document. Future extensions MAY define how to prioritize datagrams, and MAY define signaling to allow endpoints to communicate their prioritization preferences.

- HTTP Priorities [#1550](#) – How are DATAGRAM frames prioritized?

- Closed with a PR that says:

The priority scheme defined by this document considers only the prioritization of HTTP messages and tunnels ... Where HTTP extensions change stream behavior or define new data carriage mechanisms, they MAY also define how this priority scheme can be applied.



draft-pardue-masque-dgram-priority

Extends HTTP Priorities

Captures input from issues and meetings.

Realizes that this is untrodden territory so starts as Experimental.



Workgroup: MASQUE
Internet-Draft: draft-pardue-masque-dgram-priority-latest
Published: 18 July 2021
Intended Status: Experimental
Expires: 19 January 2022
Author: 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. ¶

Base HTTP Priorities recap

HTTP Extensible Prioritization for streams defines signals:

- *urgency* (“u”) - between 0 and 7. Smaller the value, higher the precedence
- *incremental* (“i”) - response can be processed incrementally (data as it arrives)

And some scheduling guidance:

- Expressing priority is only a suggestion.
- RECOMMENDED to respect urgency, serve in stream ID order.
- RECOMMENDED to respect incremental, round-robin incremental at same urgency

The proposal's design philosophy and assumptions

Start simple.

HTTP datagrams MAY have different priority from the related stream.

But all datagrams related to a stream have the same. No per-context prioritization.

Datagrams are “atomic”, so flows of datagrams are always logically incremental.

Streams and datagrams could compete - starvation is a real problem.

Flexibility with sensible defaults.

The proposal

Extend the priority scheme with a compatible parameter: *datagram-urgency* (“du”).

Identical to *urgency*, except that it applies to datagrams.

Omission of *datagram-urgency* is a signal to use the default. But there is no default value. Instead the default is to use the *urgency*.

Where stream and datagrams have the same urgency, default recommendation is to alternate between them.

Examples

Explicit different. Datagrams lower precedence.	Priority: u=1, du=2
Explicit same.	Priority: u=1, du=2
Implicit same. datagram-urgency inherits the value of 1	Priority: u=1
Implicit same via omission of both is legal. The urgency defaults to value of 3, datagram-urgency inherits the value of 3.	<intentionally blank>

Adopt this as a standalone draft?

Suspect datagram prioritization is not as easy as it seems.

Base QUIC and HTTP/3 drafts punt on priorities. Let's not delay HTTP Priorities or HTTP Datagram by trying to perfect prioritization in them.

Need to consider extension of HTTP Datagram to HTTP/2.

Need to consider prioritization of Datagram Capsules? Is that even going to be possible...

If worth adopting, should it be in MASQUE or HTTPbis

MASQUE

HTTP Priorities

draft-pardue-masque-dgram-priority

IETF 111 – Virtual SF – 2021-07

Lucas Pardue – lucaspardue.24.7@gmail.com

