

WG Working Group  
Internet-Draft  
Intended status: Informational  
Expires: 31 August 2025

C. Huitema  
Private Octopus Inc  
S. Nandakumar  
Cisco  
27 February 2025

QUIC Multimedia Performance  
draft-huitema-moq-qperfm-00

## Abstract

The QUIC multimedia performance protocol (QPERFM) updates the QUIC performance protocol (QPERF). QPERF provides a simple, general-purpose protocol for testing the performance characteristics of a QUIC implementation. QPERFM updates that protocol to simulate the traffic of multimedia application and measure performance in a multimedia oriented way.

## About This Document

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

Status information for this document may be found at  
<https://datatracker.ietf.org/doc/draft-huitema-moq-qperfm/>.

## 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 31 August 2025.

## Copyright Notice

Copyright (c) 2025 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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

## Table of Contents

1. Introduction
2. Conventions and Definitions
3. QUIC Perf protocol Extensions for Multimedia

- 3.1. Client side measurements
- 4. Security Considerations
- 5. IANA Considerations
- 6. References
  - 6.1. Normative References
  - 6.2. Informative References
- Acknowledgments
- Authors' Addresses

## 1. Introduction

The QUIC performance protocol (QUIC Perf) defined in [I-D.banks-quic-performance] provides a simple, general-purpose protocol for testing the performance characteristics of a QUIC implementation. This draft extends QPERF to allow simulation and performance measurement of a variety of multimedia applications.

The original QUIC Perf protocol was very simple. The client opens QUIC connection with the ALPN set to "perf", and then it opens bidirectional streams. The first 8 bytes sent by the client on each stream encode the size of the data that the server will send on the return stream. This can be used to measure batch performance, simply requesting a large amount of data and measuring how long it takes to get the result. It can also be used to measure transactional applications: open a large number of streams, require a small amount of data on each, and measure how long it takes to process that many query-reponse exchanges.

We extended this simple protocol to also test "real time" workloads, such as would be generated by "Media over QUIC" (moq). Instead of using the first 8 bytes of a bidirectional stream to merely specify a number of bytes, we use the first 16 bytes of the stream to specify the characteristics of a media stream. We set the first 4 bytes to a reserved value, so the server can distinguish QPERFM requests from QPERF requests.

## 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.

## 3. QUIC Perf protocol Extensions for Multimedia

The standard QPerf protocol uses bidirectional streams in a very simple way: the client opens a stream and starts sending data; the server reads the number of required bytes in the first 8 bytes of the client stream, and sends that many bytes to the client. We extend this protocol by using unidirectional streams and datagrams.

The extended QPerf protocol also uses bidirectional streams. The first 16 bytes sent by the client encode the type of response expected by the sender. The first 8 bytes use reserved values to differentiate these streams from the standard "batch" stream:

- \* The most significant 32 bits contain the value 0xFFFFFFFFD to indicate a "media" request, or 0xFFFFFFFFE to indicate a datagram request.
- \* The lower 32 bits contain the size of the frames.

The complete set of 16 bytes is defined as:

media request header {

```

    media or datagram mark (32),
    frame size (32),
    priority (8),
    frequency (8),
    number of frames (24),
    first frame size (24)
}

```

Upon receiving a request header, the server will start sending frames as specified by the frequency. If the client requested datagrams, the server will send datagrams as specified by the frequency. The first datagram (frame number 0) will be sent immediately. The other datagrams will be sent at:

```

datagram_send_time = first_datagram_send_time
                    + frame_number*1_second/frequency

```

Each datagram will carry a header and a payload, with a combined size set to the requested frame size. (The first frame size parameter is ignored for datagrams.) The first bytes of the datagram contain a header encoded as:

```

datagram header {
    request stream ID (i),
    frame number (i),
    datagram send time (64)
}

```

The datagram send time is the local time at the server, encoded in microseconds. When all datagrams have been sent, the server closes the media request stream.

If the client requested a "media" stream, the server will send the requested number of frames on the return side of the bilateral stream that carried the client request. The first frame contains "first frame size" bytes, while the other frames contain "frame size" bytes. The first frame is queued on the stream immediately. The next frames will be queued at:

```

frame_send_time = first_frame_send_time
                 + frame_number*1_second/frequency

```

The first 8 bytes of each frame carry the frame\_send\_time, set at the local time at which the server queued the frame, expressed in microseconds and encoded on 64 bits.

The client may issue a stop sending request for a specific media request stream. Upon receiving the request, the server will reset the stream, without sending any additional frame.

### 3.1. Client side measurements

The client can simulate complex scenarios by opening a series of streams and direct the server to send as many media streams. The client can parse the incoming data and extract statistics such as arrival time of different frames, latencies, etc.

## 4. Security Considerations

Since the performance protocol allows for a client to trivially request the server to do a significant amount of work, it's generally advisable not to deploy a server running this protocol on the open internet.

One possible mitigation for unauthenticated clients generating an unacceptable amount of work on the server would be to use client

certificates to authenticate the client first.

## 5. IANA Considerations

This document has no IANA actions.

## 6. References

### 6.1. Normative References

- [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

- [I-D.banks-quic-performance] Banks, N., "QUIC Performance", Work in Progress, Internet-Draft, draft-banks-quic-performance-00, 23 December 2020, <<https://datatracker.ietf.org/doc/html/draft-banks-quic-performance-00>>.

## Acknowledgments

This draft extends Nick Banks original specification of QUIC Perf, and incorporates some of the text in that original draft.

## Authors' Addresses

Christian Huitema  
Private Octopus Inc  
Email: [huitema@huitema.net](mailto:huitema@huitema.net)

Suhas Nandakumar  
Cisco  
Email: [snandaku@cisco.com](mailto:snandaku@cisco.com)