Performance Metrics Subprotocol for QUIC
draft-kazuho-quic-perf-metrics-00

Abstract

This memo proposes a subprotocol that can be used to query the performance metrics of a QUIC path.

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 http://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 August 16, 2018.

Copyright Notice

Copyright (c) 2018 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 (http://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.
1. Introduction

Observation of upstream loss, reorder, and round-trip time is crucial to diagnosing issues on the network. With TCP, it is possible for an on-path device to make estimation of such metrics by observing the sequence number and the ACK packets being sent in clear. But with QUIC, packet number is never used twice and ACK is encrypted, hence such on-path observation has become difficult. There is also an ongoing discussion about encrypting the packet numbers to avoid ossification and also to minimize privacy concern. Such change will make observation even more challenging.

There have been proposals to include signals in each QUIC packet that conveys enough information for diagnosis but does not cause ossification nor privacy issues. However, it is difficult if not impossible to figure out an approach that will work well for the lifetime of a transport protocol.

This memo proposes an alternative solution. In the proposal, an on-path device willing to obtain the performance metrics of a QUIC path sends a query to the server, and the server responds with the necessary information to calculate such metrics.

There are three primary benefits in the approach:

- Observation becomes an active action rather than passive, giving the endpoints a chance to record observation attempts as well as rejecting undesirable ones.

- Observation becomes accurate due to the endpoints’ knowledge of what is being exchanged encrypted.

- Flexibility against issues (both performance- and privacy-related) that might arise in the future, since bits for observation no
longer exists hard-coded in each packet. The metrics protocol can evolve independently to the QUIC transport protocol.

2. Overview

An on-path device that is willing to query the performance metrics of a QUIC path sends a METRICS packet of subtype REQUEST to the server-side endpoint of the path.

When receiving the request packet, a QUIC server sends a response to the client (not to the observer). The response can be either a METRICS packet of subtype RESPONSE that contains the performance metrics, or a METRICS packet of subtype DENY that indicates the servers unwillingness to provide such information. An on-path device that sent the request will witness the METRICS packet containing the response and extract the necessary values.

3. METRICS Packet

A METRICS packet is used by a QUIC server and the on-path devices to communicate the performance metrics of a QUIC path.
Figure 1: METRICS packet format

The packet mimics a QUIC packet using the short packet header. But instead of the encrypted packet number and the payload, Preamble field directly follows the Connection ID field.

A Preamble field consisting of twenty (20) octets of zero indicates a METRICS packet.

Subtype field is used to identify the role of the METRICS packet. This document defines three subtypes: REQUEST, RESPONSE, REJECT.

Request UUID field contains an identifier that is used for correlating a performance metrics request and the response.
Omit Connection ID flag indicates if the Connection ID field is omitted.

Key Phase Bit and Type field is not used and SHOULD be set to the same value as those found in the ordinary QUIC packets being exchanged on the same path during the same time.

3.1. REQUEST Subtype

A METRICS packet of subtype REQUEST (0x0) is sent from an on-path observer to the QUIC server to query the performance metrics of a QUIC path that conveyed the packets it has observed.

The sender of the packet MUST fill the Request UUID field with a sequence of random octets so that the request and the packets sent in response can be correlated.

Figure 2 shows the payload format of the REQUEST subtype.

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Client IP Address (32/128)                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Client Port (16)       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Packet Fingerprints (*)                  ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: REQUEST subtype payload format

Client IP Address field and Client Port field contain the client-side tuple of a QUIC path.

Packet Fingerprints field contains a list of the packets for which the on-path device wants to obtain the metrics. Each element of the list is thirty-six (36) octets long, containing the first thirty-six (36) octets of the observed packets immediately following the Connection ID field (or the Type field if Connection ID is omitted). The list MUST contain two or more entries.

The length of each element (thirty-six (36) octets) has been chosen so that an endpoint in possession of the key used for encrypting the packet number can decrypt the packet number from the fingerprint, when a symmetric cipher that requires an initialization vector shorter than 33 octets is being used.
3.2. RESPONSE Subtype

A METRICS packet of subtype RESPONSE (0x1) is sent by a QUIC server to indicate the performance metrics of the QUIC path related to the packets that have been specified by the METRICS packet of subtype REQUEST.

The packet MUST echo the value of the Request UUID found in the METRICS packet of subtype REQUEST so that the on-path observer that sent the request can determine the response.

Figure 3 shows the payload format of the METRICS subtype.

```
+---------------------------------------------+ 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
| Number of Packets Sent (*) .................. |
| Number of Packets Lost (*) .................. |
| SRTT (32) ..................................... |
| RTTVar (32) .................................... |
| Distances (*) ................................... |
```

Figure 3: METRICS subtype payload format

Number of Packets Sent field contains the cumulative number of packets being sent on the path. Number of Packets Lost field contains the cumulative number of packets sent on the path but were deemed to be lost. The values are encoded using Variable-Length Integer Encoding defined in [QUIC-TRANSPORT].

SRTT field and RTTVar field indicates the smoothed round-trip time and the RTT variant, in the unit of microseconds, encoded as 32-bit unsigned integers.

Distances field contains a sequence of integers representing the distances between the packets specified by the request.

The field contains one less elements than the corresponding Packet Fingerprint field. Nth element of the Distances field corresponds to the distance between the Nth element and the N+1th element of the Packet Fingerprint field.

Distance between two packets (A and B) is defined as following:
o the integral part represents number of packets being sent between the two, incremented by one

o the sign part is "+" if A was sent before B, else "-"

As an example, if packet B was sent right after A, the distance between A and B is one "1". If packet B was sent two packets after A, the distance is "2". If packet B was sent right before A, the distance is "-1".

Each distance is converted to unsigned values using the following formula, then encoded into variable length using Variable-Length Integer Encoding defined in QUIC-TRANSPORT.

\[
\text{uint_distance} = \text{abs}(\text{distance} \times 2) + (\text{distance} < 0 ? 1 : 0)
\]

By observing the packet, the on-path device that sent the request can determine the losses or reorders between the packets it specified in the request.

3.3. DENY Subtype

A METRICS packet of subtype DENY (0x2) is sent by a QUIC server to indicate its unwillingness to provide performance metrics.

There is no payload for the subtype.

4. Server Behavior

A QUIC server MUST ignore a METRICS packet of subtype REQUEST if any of the following requirements is not being met.

o the destination IP address and port number match the server-side tuple of the QUIC path

o values of Client IP Address field and Client Port field match the client-side tuple of the QUIC path

o (unless omitted and permitted to be omitted during QUIC handshake) value of the Connection ID field matches that of the QUIC path

o Packet Fingerprint field contains two or more entries

o all the packet numbers being recovered from the entries of Packet Fingerprint field belong to the QUIC path
Once all the checks succeed, the server can send a METRICS packet of subtype RESPONSE, or notify the rejection the request by sending a packet of subtype DENY.

4.1. Server State

A QUIC server willing to let the on-path devices observe upstream loss and/or reorder ratio needs to calculate the distances of the packets being specified in the request.

A server can easily calculate the distances if it records the packet numbers of all the packets it has sent over a given path.

On the other hand, a server can calculate the distances by retaining very little state if it is implemented following the criteria shown below.

- record the packet number of the first packet sent after switching to the current path
- use a deterministic function (such as a keyed hash function) to determine when to skip a packet number as a mitigation against opportunistic ACK attacks
- record the packet numbers of packets that were exchanged on a prepared path (i.e. packet numbers of PATH_CHALLENGE and PATH_RESPONSE)

The distance can be calculated as a subtraction of two packet numbers, further subtracted by the number of skips and the number of packets used for preparing new paths.

5. Normative References

[QUIC-TRANSPORT]

Author’s Address

Kazuho Oku
Fastly

Email: kazuhooku@gmail.com