The Transport-Info Response Header
draft-ohanlon-transport-info-header-00

Abstract

The Transport-Info header provides a mechanism to inform a client of the last-mile server’s view of the network transport related information such as current delivery rate and round-trip time. This information has a wide range of uses such as client monitoring and diagnostics, or allowing a client to adapt to current network conditions.

Note to Readers

_RFC Editor: please remove this section before publication_

Source code and issues for this draft can be found at https://github.com/bbc/draft-ohanlon-transport-info-header [1].

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 May 20, 2020.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.
1. Introduction

The Transport-Info header provides for relaying of transport protocol related information from a last-mile edge server entity to a client with the aim of informing the client of the server’s view on the transport state. The state of a connection is dependent upon information based upon packet exchanges during the transport processes. Firstly, there is information that is common to both client and server, such as the calculated round-trip time (RTT), although it may be measured using different packets at each end. Secondly, there is state information that exists only at each endpoint, such as the size of the congestion, and receive windows. Thus certain transport state information is only available at the server which can be useful to the client, for example, to calculate the current transport rate. This information may then be used to better inform a client of the state of the network path and make appropriate adaptations.

The information can also be utilised by a client to provide for application level client oriented metric logging to back-end systems.
for monitoring and analysis purposes. Such data could be utilised in a manner not unlike that proposed in [RFC4898].

This approach is directly applicable to TCP but also can be utilised with other related transport protocols, such as QUIC [I-D.ietf-quic-transport].

1.1. Motivation

This work is motivated in part by the fact that even in modern web browsers web applications are not currently able to obtain such low level information about their connections. Additionally some information is only available at the server, such as the size of the server congestion window. As a result clients often resort to application level measurements, to infer such things as the current delivery rate, but these are not always indicative of the performance of the transport layer.

There exist W3C specifications such as the Network Information API [network-info-api], which contains an attribute named "downlink" that purports to provide measurement of effective bandwidth "across recently active connections". However, in practice the downlink measurement appears to be a very rough estimate which is of little use for informing an application of dynamic network conditions. Furthermore, it currently has limited browser support.

1.2. 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.

This document uses the Augmented Backus-Naur Form (ABNF) notation of [RFC5234] with the list rule extension defined in [RFC7230], Appendix B. It includes by reference the DIGIT rule from [RFC5234] and the OWS and field-name rules from [RFC7230].

2. The Transport-Info Response Header

The Transport-Info header uses the proposed Structured Header draft [I-D.ietf-httpbis-header-structure]

Transport-Info = sh-list

Each member of the parameterised list represents an entry that contains a set of metrics reported by the edge server.
The list members identify the server that inserted the value, and MUST have a type of either sh-string or sh-token. Depending on the deployment, this might be a product or service name (e.g., ExampleEdge or "Example CDN"), a hostname ("edge-1.example.com"), and IP address, or a generated string.

Each member of the list can also have a number of parameters that contain metrics. While all but one of these parameters are OPTIONAL, edge servers are encouraged to provide as much information as possible.

- Exactly one parameter whose name is "ts", and whose value is an sh-float indicating the measurement timestamp in seconds since UNIX epoch.
- Optionally one parameter whose name is "alpn", and whose value is an sh-string representing the ALPN protocol identifier [alpn-ids].
- Optionally one parameter whose name is "cc_algo", and whose value is sh-string, conveying the name of congestion control algorithm used by the server for this connection.
- Optionally one parameter whose name is "cwnd", and whose value is a sh-integer, conveying the size of the server’s congestion window [RFC5681] in packets.
- Optionally one parameter whose name is "rcv_space", and whose value is a sh-integer, conveying the size of the receiver’s window in bytes.
- Optionally one parameter whose name is "dstport", and whose value is a sh-integer, conveying the server’s destination port of this connection for correlation of measurements between requests.
- Optionally one parameter whose name is "mss", and whose value is a sh-integer, conveying the size of the server’s Maximum Segment Size in bytes.
- Optionally one parameter whose name is "rtt", and whose value is an sh-float, in milliseconds, indicating the server’s estimate of the Round-Trip Time from its transport layer.
- Optionally one parameter whose name is "rttvar", and whose value is an sh-float, in milliseconds, indicating the server’s estimate of the variation of the Round-Trip Time [RFC6298] from its transport layer.
- Optionally one parameter whose name is "send_rate", and whose value is a sh-float, in kilobits per second, conveying the server’s calculation of the sending rate for this connection.

Here is an example of a header with a single set of metrics:

```plaintext
Transport-Info = ExampleEdge; ts=1567176968.69; alpn="h2"; cwnd=24; rtt=250; mss=1460; rttvar=10; dstport=12345
```
Whilst it is understood that such metrics may only provide an instantaneous view on the transport state, the Transport-Info header is designed to allow for delivery of multiple timestamped entries in a single header.

Here is an example of header with multiple entries, utilising the structured header inner-list type:

```
Transport-Info = "edge-1.example.com"; ts=1567176968.69; alpn="h2"; cwnd=24; rtt=250; mss=1452; rttvar=10; dstport=123451,
 "edge-1.example.com"; ts=1567176969.97; alpn="h2"; cwnd=23; rtt=255; mss=1452; rttvar=12; dstport=123451
```

If the end points support HTTP/2, and later, another technique to increase temporal coverage for an ongoing session is for the client to issue additional HEAD or OPTION * requests for a resource at the same origin. This works with HTTP/2 and later as all requests to the same origin utilise one TCP or QUIC connection. Whilst the HTTP priorities can affect the allocation of capacity between streams, one use-case for the Transport-Info header is for information regarding sustained flows, such as media delivery, which tend consist of a known limited number of flows to the same origin so the priorities would not affect the calculations.

### 2.1. Utilisation of Transport-Info header metrics

In the case of TCP, calculation of the transport transmission rate is possible using the cwnd and rtt, and knowledge of the mss. The equation being as follows:

\[
\text{send\_rate} = 8 \times \text{send\_window} / \text{rtt}
\]

Where \( \text{send\_window} = \min (\text{cwnd} \times \text{mss}, \text{rcv\_space}) \)

If the mss is not available then it is possible to perform the calculation using an estimate of the mss, or a common value such as 1460 for IPv4. It understood there can be some variation for different network and tunnelled paths (e.g. 1452 for IPv4 PPPoE) as can been seen in recent studies [exploring-mtu], although the large proportion of mss values fall within a range 1220-1460. The send_window is preferably calculated using a minimum of the cwnd and rcv_space, but if the rcv_space is not available it may be approximated by just using the cwnd.

This equation maybe applied for other related window based transport protocols (e.g. QUIC [I-D.ietf-quic-transport]) with similar information, although it may need some modification.
The calculation of the send rate maybe performed at the server, or may be left to the client to calculate as and when required.

3. Server side behaviour

With most web server deployments an origin server sits behind some form of CDN, with varying levels of fan-out to a point where an edge server is connected on the last mile to clients. The Transport-Info header SHOULD only be inserted into an HTTP stream by the last hop edge server that is connected to clients so that it conveys information pertinent to the client’s direct transport path. Also the Transport-Info MUST not be cached.

_RFC Editor: please remove this section before publication_

The provision of the Transport-Info header is possible using a number of existing server systems that already provide support for such metrics, which currently utilise operating system support for tcp_info data structure which are available on Linux and BSD systems.

In terms of current implementations there is in-built support in Nginx/Openresty using its variables "var.tcpinfo_rtt" etc. Apache Traffic Server provides support using the TCPInfo plugin. Varnish provides access to "tcp_info" using their "vmod_tcp" module. Node.js has libraries such as "nodejs_tcpinfo" which provide support. Whilst most of the implementations do not provide access to the TCP MSS it is available via the underlying kernel tcp_info data structure so it would be fairly straightforward to provide access to such information.

4. Client side proxy considerations

In the case where a proxy services client requests, this proxy would be configured according to local policy as to whether it passes through, modifies or drops the Transport-Info header. This decision can depend on a number of factors, including the utility of the header given local network configuration, and also whether the header might reveal unwanted information to end clients, since the Transport-Info header would relate to the connection between the edge CDN node and the proxy.

5. IANA Considerations

This specification registers the following entry in the Permanent Message Header Field Names registry established by [RFC3864]:

- Header field name: Transport-Info
6. Security Considerations

The content of the Transport-Info is largely available through other techniques such as packet capture so it should not lead to security issues. Certain metrics, such as the cwnd, may be considered as less visible but since they are part of the transport layer they can be inferred. Any metrics that may be considered private should not be sent in the header, or sent only over an encrypted connection.

In the case where clients are connected via a proxy then organisations may wish modify or drop the header if they consider the it might reveal unwanted information to end clients.

If the header is delivered over a transport protocol whose content can be modified without detection then parties should be aware that the header could be maliciously modified to alter the metrics values which could result in the client making incorrect adaptations.

7. References

7.1. Normative References


7.2. Informative References

[alpn-ids]

[exploring-mtu]

[I-D.ietf-quic-transport]

[network-info-api]

[RFC4898]

[RFC5234]

[RFC7230]
7.3. URIs


Authors’ Addresses

Piers O’Hanlon
British Broadcasting Corporation

Email: piers.ohanlon@bbc.co.uk

James Gruessing
British Broadcasting Corporation

Email: james.gruessing@bbc.co.uk