HTTP
Internet-Draft
Intended status: Standards Track
Expires: September 4, 2020

P. O'Hanlon
J. Gruessing
British Broadcasting Corporation

March 3, 2020

The Transport-Info HTTP Header
draft-ohanlon-transport-info-header-01

Abstract

The Transport-Info header provides a mechanism to transmit network transport related information such as current delivery rate and round-trip time, from a server or a client. 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 September 4, 2020.

Copyright Notice

Table of Contents

1. Introduction

The Transport-Info header provides for relaying of transport protocol related information from either a server or client entity with the aim of informing the sender'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 modern web browser-based web applications are not currently able to obtain such low level information about specific 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. However, these are not always indicative of the performance of the transport layer, and may not be sufficiently precise due to a couple of issues; Firstly, browser based timing is limited by the granularity of the JavaScript timers, which were reduced in the light of timing based side-channel attacks, although due to new mitigations such timer limits are currently of the order 5us-1ms. These limits can be an issue for higher rate connections and/or those with smaller transactions. Secondly, with flows where the content-length is unknown, such as with chunked transfer encoding, it is currently difficult to correctly measure the bandwidth in the browser as the even the fetch/streams APIs do not provide for sufficient information.

There exist W3C specifications such as the Network Information API [network-info-api], which provides estimates of metrics, including downlink rate and RTT, that are measured "across recently active connections", but are platform and browser dependent, with limited cross-browser support. In practice the downlink measurement is is generally of low accuracy and of little use for informing an application of dynamic network conditions, and the RTT measurement is also of low accuracy. However, it is implemented in Chrome and the utilisation of the API is now seen in a large proportion of websites, mainly due to adoption of the API by widely used libraries.

This information is already being sent by servers and clients so this document specifies a standard way for entities to encode and transport such information.

1.2. Use Cases

The header can be used to provide sender specific transport information that can inform a range of functions:

- Assist or drive the media quality selection algorithms for streaming media.
- Inform initial rate selection.
- Provide better bandwidth information for shorter requests (e.g. gRPC, audio) which are harder to measure.
    - Could be used to drive scheduling of different flows in systems such as Traefik.
- The RTT values are useful for informing the operation of latency sensitive applications.
- The RTTVAR could be used to provide an estimate of 'reliability' of rtt and bandwidth estimates.
- Inform client/browser media/data caching strategies.
- Use by intermediate nodes for traffic analytics and control.

1.3. 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 HTTP 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.

The list members identify either the server or client 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, implementations are encouraged to only provide as much information as necessary.

- Exactly one parameter whose name is "ts", and whose value is an sh-string indicating the measurement timestamp in [RFC3339] format.
- 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 for this connection.
- Optionally one parameter whose name is "cwnd", and whose value is a sh-integer, conveying the size of the 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 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 Maximum Segment Size in bytes.
- Optionally one parameter whose name is "rtt", and whose value is an sh-float, in milliseconds, indicating the 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 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 calculation of the sending rate for this connection.

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

```
Transport-Info = ExampleEdge; ts="2019-08-30T14:56:08.069Z";
                  alpn="h2"; send_rate="5100"
```

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 the header with multiple entries, utilising the structured header inner-list type:

```
Transport-Info = "edge-1.example.com"; ts="2019-08-30T14:56:08Z";
                  cwnd=24; rtt=50; mss=1452; rttvar=10; dstport=8065,
                 "edge-1.example.com"; ts="2019-08-30T14:57:08Z";
                  cwnd=23; rtt=55; mss=1452; rttvar=12; dstport=8065
```

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 requests for the resource at the same origin. This works with HTTP/2, and later, as all requests to the same origin usually utilise one TCP or QUIC connection. Whilst the HTTP priorities can affect the allocation of capacity between streams the header will still provide an estimate of the maximum available capacity. Likewise, in some cases with HTTP/2, and later, there may be multiple flows traversing the same transport connection to different origins if connection reuse (Section 9.1.1 of [RFC7540]) is utilised, which could have a similar effect on interpretation of the metrics to HTTP priorities, but may have privacy implications which are addressed in the privacy section Section 6.1.

2.1. Utilisation of Transport-Info header metrics

The metrics may be used directly to inform entities that receive the header. The calculation of the send rate maybe performed by the sender of the header and included in the send_rate parameter, or the receiver may calculate it as described below. The decision may depend upon a variety of factors including the privacy consideration of transporting any required parameters.

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:

```
send_rate = 8 * send_window / rtt

Where send_window = min (cwnd * mss, 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 is 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.

3. Server based 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 hop 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. The Transport-Info header MUST not be cached.

With respect to use in CORS [cors] enabled environments access to the header will be subject to restrictions in cross domain requests, which may be controlled through the inclusion of the Transport-Info header in the Access-Control-Request-Headers header.

The use of the header is expected to comply with data minimisation approaches where servers only send the necessary information on relevant flows.

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 the "tcp_info" data structure which is available on Linux and BSD based 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 based behaviour

The use of the header by a client is envisaged as a less common use-case since such information is generally less readily available on clients, and its general use might have privacy implications, although servers will be aware of most transport state already. We propose that use of the header could be controlled through the use of

the Allow-CH header [I-D.ietf-httpbis-client-hints]. The header can enable the server to make better informed decisions based upon client based transport information. In the case of non-browser clients which have access to transport information directly through operating system interfaces, this information can be relayed using the header. Whilst with browser based clients such information could be obtained through the use of the JavaScript Network Information API.

4.1. Client side proxy considerations

In the case where a client is configured to utilise a proxy directly, or through the use of the HTTP CONNECT pseudo-method, this proxy should 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 whether the flows are encrypted, 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
- Applicable protocol: http
- Status: standard
- Author/Change Controller: IETF
- Specification document(s): [this document]
- Related information:

6. Security Considerations

6.1. Privacy Considerations

The Transport-Info header provides information about a senders view of its network transport metrics, such as bandwidth and latency, to its receiver. This information may potentially be abused for such purposes as fingerprinting a user through their particular network metrics or a time series thereof. In some situations it might also be possible to infer location of users. This may also apply in the case where multiple users, or user identities, share a connection through the use of connection reuse mechanisms or otherwise.

However, these issues are not new and such information is already being shared by some servers and clients to arbitrary levels of accuracy. Furthermore, there are a number of other ways an attacker

can obtain such information. In the client side, in a browser, there exist a number JavaScript based techniques to measure the bandwidth and latency through existing network APIs such as the Network Information, the Resource Timing, and WebRTC. On the server side, or a non-browser client, there is no limit to the techniques that may be applied to obtain information about network flows.

6.2. Information control

Whilst such information may be available through other mechanisms we recommend that implementers minimise any potential privacy issues through the application of the following approaches: - The principle of data minimisation should be applied to any use of the header such that only information required for the purposes of the application be shared. - Any metrics deemed sensitive should apply an appropriate level of quantisation and noise to the values to a level that provides privacy whilst allowing for actual utility of the values. - Consideration of limits to the temporal update frequency of the metric values. - Any metrics that may be considered private should not be sent in the header, or should be appropriately protected. - Metrics should be sent over an encrypted connection.

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

[I-D.ietf-httpbis-client-hints]
Grigorik, I. and Y. Weiss, "HTTP Client Hints", draft-ietf-httpbis-client-hints-10 (work in progress), February 2020.

[I-D.ietf-httpbis-header-structure]
Nottingham, M. and P. Kamp, "Structured Headers for HTTP", draft-ietf-httpbis-header-structure-15 (work in progress), January 2020.

[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>.

[RFC3339] Klyne, G. and C. Newman, "Date and Time on the Internet: Timestamps", RFC 3339, DOI 10.17487/RFC3339, July 2002, <https://www.rfc-editor.org/info/rfc3339>.

[RFC3864] Klyne, G., Nottingham, M., and J. Mogul, "Registration Procedures for Message Header Fields", BCP 90, RFC 3864, DOI 10.17487/RFC3864, September 2004, <https://www.rfc-editor.org/info/rfc3864>.

[RFC5681] Allman, M., Paxson, V., and E. Blanton, "TCP Congestion Control", RFC 5681, DOI 10.17487/RFC5681, September 2009, <https://www.rfc-editor.org/info/rfc5681>.

[RFC6298] Paxson, V., Allman, M., Chu, J., and M. Sargent, "Computing TCP's Retransmission Timer", RFC 6298, DOI 10.17487/RFC6298, June 2011, <https://www.rfc-editor.org/info/rfc6298>.

[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>.

7.2. Informative References

[alpn-ids] "Application-Layer Protocol Negotiation (ALPN) Protocol ID", IANA , n.d., <http://www.iana.org/assignments/tls-extensiontype-values>.

[cors] van Kesteren, A., "Cross-Origin Resource Sharing", W3C , January 2014, <http://www.w3.org/TR/2014/REC-cors-20140116/>.

[exploring-mtu] Custura, A., Fairhurst, G., and I. Learmonth, "Exploring usable Path MTU in the Internet", Network Traffic Measurement and Analysis Conference , April 2018, <https://doi.org/10.23919/TMA.2018.8506538>.

[I-D.ietf-quic-transport] Iyengar, J. and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport", draft-ietf-quic-transport-27 (work in progress), February 2020.

[network-info-api] Grigorik, I., "Network Information API", W3C , September 2019, <http://wicg.github.io/netinfo/>.

[RFC4898] Mathis, M., Heffner, J., and R. Raghunarayan, "TCP Extended Statistics MIB", RFC 4898, DOI 10.17487/RFC4898, May 2007, <https://www.rfc-editor.org/info/rfc4898>.

[RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <https://www.rfc-editor.org/info/rfc5234>.

[RFC7230] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", RFC 7230, DOI 10.17487/RFC7230, June 2014, <https://www.rfc-editor.org/info/rfc7230>.

[RFC7540] Belshe, 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>.

7.3. URIs

[1] https://github.com/bbc/draft-ohanlon-transport-info-header

Appendix A. Acknowledgements

The authors would like to thank Craig Taylor, Lucas Pardue, Patrick McManus, and the IETF HTTP Working Group for feedback on the development of this document.

Appendix B. Changes

B.1. Since -00

- Issue 1 (HTTP Tunnels): Added text regarding the use of HTTP CONNECT.
- Issue 3 (Is sub-second resolution appropriate?): Changed from UNIC Epoch to RFC3339 time format.
- Issue 4 (Could this be used for both request and response?): Updated text to describe both server and client use, and their implications.
- Issue 5 (Privacy Implications): Added new Privacy Considerations section and updated security section
- Issue 9 (CORS considerations): Added text to address CORS usage.
- Issue 10 (Provide additional use-cases): Updated motivation and added use-cases section.

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