Network Working Group
Internet-Draft
Expires: June 11, 2013

O. Mazahir
J. Padhye
Microsoft
W. Chan
R. Peon
Google
R. Trace
Microsoft
S. Loreto
Ericsson
G. Montenegro
Microsoft

December 8, 2012

HTTP 2.0 Principles for Flow Control
draft-montenegro-httpbis-http2-fc-principles-01

Abstract

This document states the principles for flow control in HTTP 2.0.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

HTTP/2.0 introduces multiplexed streams over a given TCP connection. In HTTP 1.X, there is no interleaving of Request/Response pairs. Thus, any flow control issues are mostly left to the underlying TCP implementation. In HTTP 2.0, each Request/Response pair uses a separate stream, sharing the same TCP connection with other such pairs over different streams. All such streams will be vying for a common underlying resource of a single TCP connection. Given that this interaction among all the streams is not visible to the TCP implementation, handling the interaction among them has to be solved at the HTTP 2.0 multiplexing layer. There are issues of prioritization, head-of-line blocking and flow control. Perhaps the most complex aspect is that of flow control. It may be that flow control for HTTP 2.0 multiplexing will follow a path similar to what TCP's complex dynamics have followed throughout the years. In particular, TCP congestion control has seen a constant progress of improved specifications based on measurements and research of the networking community. What the TCP community recognized early on was that this was a hard problem. Thus, the best course of action was to agree on a minimal set of rules or principles (e.g., TCP "friendliness"). Many TCP congestion control algorithms are then possible as a (mostly) local implementation issue giving rise to TCP Reno, Tahoe, Vegas, CTCP, and many more.

Flow control for HTTP 2.0 multiplexing over TCP is also a complex issue. This document proposes (1) a set of principles aimed at preventing egregious behavior, while allowing for future and ongoing improvement of flow control algorithms, and (2) a simple flow control algorithm that could be implemented in the absence of better schemes (TBD). Other flow control algorithms with subsequent improvements should be specified in separate documents without encumbering nor delaying the base HTTP 2.0 specification. This is similar to how the myriad TCP congestion algorithms published so far have been specified separately from the base TCP documents.

The goal of this document is to propose additional text to the HTTP/2.0 specification. The starting point for HTTP/2.0, the SPDY [I-D.mbelshe-httpbis-spdy] protocol, does not have much language with respect to flow control. Hence, the text below is offered as a new section or sections within the HTTP/2.0 document.

2. Principles for Flow Control in HTTP 2.0 Multiplexing

Flow control for Multiplexing in HTTP 2.0 must follow these principles:

1. Flow control is hop by hop (where "hop" means an HTTP 2.0 hop), and not end-to-end.

2. Flow control is based on window update messages. It is essentially a credit-based scheme.

3. Flow control is directional and is determined by the receiver. Flow control MAY be declared by the receiver and MUST be heeded by the sender. For example, a client, a server or a proxy (in their role as a "receiver") independently advertise their flow control preference. The other side when operating as a "sender" must heed that preference.

4. Flow control in the direction towards the receiver can be OFF or ON as determined by the receiver. It is OFF if no flow control is advertised by the receiver, or if the receiver declares "infinite" credit to the sender.

5. HTTP 2.0 should only standardize the format of the window update message and its semantics. In particular, the algorithms used by the receiver to decide when to send window update messages, and how much to update the window by, are not mandated in the spec. The draft should, however, provide some illustrative examples.

NOTE: Whether flow control operates on a per-stream basis, on a per-session (per-TCP connection) basis or on both a per-stream and a per-session basis is TBD.

The spec will not define the algorithms the sender will use to manage priorities among streams and to minimize head of the line blocking. This is included for completeness, but is essentially independent of flow-control.

3. Acknowledgements

This document was produced using the xml2rfc tool [RFC2629].

4. References

4.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

[I-D.ietf-httpbis-p1-messaging]
Fielding, R. and J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", draft-ietf-httpbis-p1-messaging-21 (work in progress), October 2012.

[I-D.ietf-httpbis-p2-semantics]
Fielding, R. and J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content", draft-ietf-httpbis-p2-semantics-21 (work in progress), October 2012.

4.2. Informative References

[RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, June 1999.

[RFC6455] Fette, I. and A. Melnikov, "The WebSocket Protocol", RFC 6455, December 2011.

[I-D.mbelshe-httpbis-spdy]
Belshe, M. and R. Peon, "SPDY Protocol", draft-mbelshe-httpbis-spdy-00 (work in progress), February 2012.

[I-D.montenegro-httpbis-speed-mobility]
Trace, R., Foresti, A., Singhal, S., Mazahir, O., Nielsen, H., Raymor, B., Rao, R., and G. Montenegro, "HTTP Speed+ Mobility", draft-montenegro-httpbis-speed-mobility-02 (work in progress), June 2012.

[I-D.tarreau-httpbis-network-friendly]
Tarreau, W., Jeffries, A., and A. Croy, "Proposal for a Network-Friendly HTTP Upgrade", draft-tarreau-httpbis-network-friendly-00 (work in progress), March 2012.

Authors' Addresses

Osama Mazahir
Microsoft

Email: OsamaM@microsoft.com

Jitu Padhye
Microsoft

Email: padhye@microsoft.com

William Chan
Google

Email: willchan@chromium.org

Roberto Peon
Google

Email: fenix@google.com

Rob Trace
Microsoft

Email: Rob.Trace@microsoft.com

Salvatore Loreto
Ericsson

Email: salvatore.loreto@ericsson.com

Gabriel Montenegro
Microsoft

Email: Gabriel.Montenegro@microsoft.com