HTTP 2.0 and WebSockets Coordination

Brian Raymor (Microsoft)

Greg Wilkins (Intalio)

Willy Tarreau (Exceliance)

Gabriel Montenegro (Microsoft)

Motivation

HyBi

- RFC6455 (websockets) defines
  - a (1) **negotiation** to switch protocols beyond HTTP 1.1 into a *websockets session,* plus any (2) **session maintenance**
  - **(3) binary framing**
- Ongoing work to define (4) **multiplexing** for *websockets*, along with (5) **flow control**

HTTPbis (HTTP 2.0)

- Current proposals define
  - a (1) **negotiation** to switch protocols beyond HTTP 1.1 into an *HTTP 2.0 session,* plus any (2) **session maintenance**
  - **(3) binary framing**
- Current proposals define (4) **multiplexing** for *HTTP 2.0* , along with (5) **flow control**

Let’s avoid this duplication/proliferation of protocols.

Coordination between HTTP 2.0 and Websockets

Alternative #1: Operate both over a **currently defined common layer** for framing&multiplexing

- Well, out of all this, the only thing currently defined in an RFC is *framing* for websockets (RFC6455).
- Would HTTP 2.0 adopt RFC6455 framing as is?
  - Current S+M proposal, but what to do about WS masking, WS variable length encodings, etc?
- Who would get to define multiplexing and have the other adopt it?

Alternative #2: Operate both over a **to-be-defined common layer**: "*HTTP/2.0 framing*"

- HTTP 2.0 will soon be chartered to define such a "2.0" layer.
- HTTP 2.0 defines it as a common layer that can be reused.
- "websockets 2.0" (some future version) would adopt this common layer to include multiplexing via the "*HTTP/2.0 framing*" proposal from HTTP 2.0

What is “*HTTP/2.0 framing*”

A separable protocol layer to be defined as part of HTTP/2.0 that will frame HTTP messages. It is likely that this framing layer will be derived from the current HTTP/2.0 proposals (SPDY framing or websockets/1.0) and will support message fragmentation and multiplexing.

Challenges:

- What to do about masking (required by Websockets 1.0)
- semantics for Ping, Close
- Closing channels/streams
- Initiation of stream/channel (client-only in WS 1.0 mux currently)
- Stream/Channel Prioritization (currently not in WS 1.0 mux)
- Flow Control (a hard thing to get right, fraught with tradeoffs)
- Encoding of channel/stream numbers (currently complex in WS 1.0 mux)
- Hop-to-hop nature of HTTP vs end-to-end nature of WS

Proposal: Alternative #2: Aim for only one future framing&multiplexing (“*HTTP/2.0 framing*”)

- HTTPbis WG:
  - defines *HTTP/2.0 framing*
  - Immediate application for HTTP 2.0
  - Make carrying websocket messages a requirement of HTTP/2.0 framing
- HyBi:
  - eventually recharters to work on “Websockets 2.0”
  - Websockets 2.0 uses *HTTP/2.0 framing*