WebSockets: Compression and Multiplexing

Ian Fette - ifette@google.com

Compression

- Spec currently defines deflate-stream extension (Section 9.2.1 of spec)
- deflate-stream takes all data after the handshake and compresses it.
- deflate-stream thus modifies the base framing (an intermediary must follow the compression to understand e. g. frame boundaries and sizes)
- deflate-stream negatively effected by masking (locality is lost because of masking)
- Current deflate-stream extension in spec has been questioned by multiple participants on the mailing list

Compression

- deflate-application-data ("deflate-frame") proposed as alternate to deflate-stream [1]
- Compresses the data contained in payload, doesn't affect framing
- State of compression dictionary / state not negatively affected by masking
- Experimental implementation shows better efficiency than deflate-stream [2]

[1] http://tools.ietf.org/html/draft-tyoshino-hybi-websocket-perframe-deflate-01
[2] http://www.ietf.org/mail-archive/web/hybi/current/msg07860.html

Compression

Open questions

- Allocation of RSV bits (useful for turning off per-frame compression, broader question around RSV bits and extensions generally)
- Coordinating with W3C/WHATWG (previously required deflate-stream, now forbids deflate-stream).
- Ability to enable/disable compression from JavaScript?
- Extension parameters in deflate-frame to control options (LZ77 window size, etc)

Multiplexing

- For large websites, very important to be able to re-use TCP connections
- Browsers have max number of connections that can be open e.g. to a proxy
- Theoretically possible to share a connection in client-side code, but similar to cooperative multitasking with all of its downfalls, and hard to program to
- Would like to move multitasking into the browser / protocol extension
- Proposed x-google-mux, want to figure out how to get more attention and move it forward as a spec [1]
  [1] https://docs.google.com/a/google.com/document/d/14DYRxmRTPVhV1GbVXlxg8KVrS1SEq4UhCNq6q8iNOws/view