

WebSocket multiplexing compression and HTTP/2.0

Takeshi Yoshino

tyoshino at google dot com

HyBi WG IETF 84 Vancouver

Integration Issues

- Per-frame compression locks fragmentation
 - Multiplexor needs to decompress, re-fragment and re-compress to fit quota
 - So, per-message?
- Per-frame extension data based multiplexing interleaves messages
 - Per-message compression is not applicable after multiplexing

Handle Everything on Per-message Basis

- It's time to give up ...
 - Using extension data on per-frame basis
 - Using RSV bits on per-frame basis
 - Using frame boundary information
- Let's use
 - Per-message compression
 - Multiplexing by encapsulation

Per-message compression

- Define how to compress “message”
- Friendly to multiplexing and intermediaries
 - Compressed messages are still safe to be re-fragmented
- Diff from the WG item is [a little](#)
 - draft-tyoshino-hybi-permessage-compression-00

Multiplexing by Encapsulation

- Changed on -03
- Encapsulate a frame into a binary message with channel ID
 - Only 1 byte additional overhead
- Friendly to intermediaries and post-mux compression
 - Encapsulating msgs are safe to be re-fragmented
 - Control frames can be inserted without interpreting multiplexing

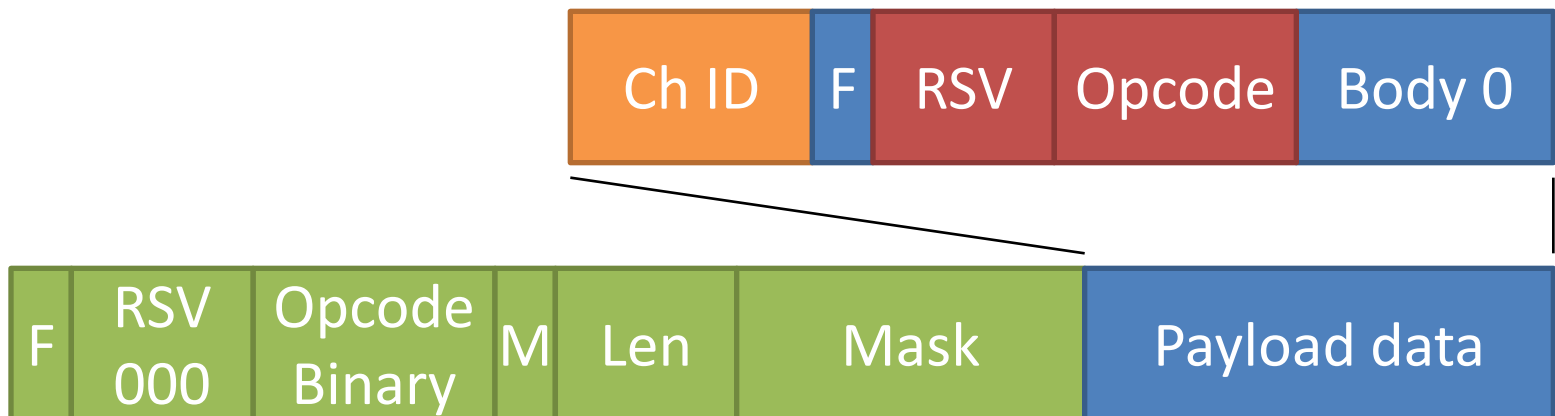
All the information of a message



Multiplexor (re-)fragments it according to quota



Encapsulate the chunked bodies into messages



Other Update

- Flow control
 - Pre-handshake quota
 - NewChannelSlot: AddChannelRequest throttling
- Compression
 - Decoupled compression algorithm and framing
 - Simple negotiation by parameter echo back

Multiplexing TODO

- Quota
 - 1 byte penalty / message
 - Spend quota for control frames
- Define multiplex error codes
- Define how to fallback to another physical connection

Can HTTP be Layered over WebSocket multiplexing?

- From SPDY, incorporate
 - header encoding for efficient HTTP transfer
 - “:scheme” header for protocol switching
- Unmask when safe
- ID space separation for server initiated stream
- Simplify number encodings
- Priority header