Requirements drafts issues

Gabriel Montenegro (Microsoft)
Salvatore Loreto (Ericsson)
Joe Hildebrand (Cisco)

Focus on major issues

Four issues to discuss today:

- #1 HTTP Compliance
- #8
  Support for binary data in addition to textual data.
- #7 Size of messages and/or total message size
- #9 Sub-Protocol support

#1 HTTP Compliance

- Based on the charter's clear text about the WG defining a mechanism between existing HTTP entities with as much compatibility as possible, and based on rough consensus on the mailing list:
  - HTTP (typically on ports 80 and/or 443) the WebSocket protocol will be HTTP compliant until the Upgrade exchange is completed.
  - The WG's focus is on leveraging existing HTTP-based infrastructure, although a future rechartering could investigate other alternatives.

Formal Declaration of Consensus

#6 HTTP Upgrade in relation to the WebSocket protocol

- In line with #1, HTTP compliance.

#8 Support for binary data in addition to textual data.

- Binary support will be required by JS APIs by the time the WG is done with the protocol, and it will be required before that by non-browser client code. Also called for by the charter's mention of a general protocol, and of non-browser scenario.

#7 Size of messages and/or total message size

- For binary (but also for text) it is problematic to not have a length indication.
- It is also preferable to have only one mechanism to indicate length. E.g., having both a length indication and a sentinel could result in two conflicting length indications.
- Better to have only one, whichever is more general: a length value so receivers can, in advance, know how much buffering they'll need, and, whether they're willing to process that message or not.
- Note: This size requirement does not apply to the *concatenation* of the individual chunks.

Potential Hums

- Should we support binary data?
- Should we support ONLY UTF-8 data?
- Do we want a different framing mechanism for UTF-8 strings and binary data?
- Should the chunk be unlimited in size?
- Should the concatenation of chunks into a frame be unlimited in size?
- Does the total length of the frame need to be known at the beginning of the frame? (e.g. “more” flag)

Potential Hums

- “Message” is a protocol unit with an end
- A message may be composed of one or more “frame”s
- Each frame has a length indication, encoded in a fixed number of bits (where that lengths is fixed in the specification to be written)

#9 Sub-Protocol support

- It should be possible to support other protocols by using the sub-protocol mechanism rather than http Upgrade.
- This would allow, say, XMPP to work from within the browser using the WebSockets support.
- There may be an advantage in the future as infrastructure (proxies, etc) start recognizing the WebSocket Upgrade token and blocking other tokens. This is not the case today, however.
- Discussion.