Data Channels

Randell Jesup (randell@jesup.org)
Salvatore Loreto (salvatore.loreto@ericsson.com)
Michael Tüxen (tuexen@fh-muenster.de)

Outline

- Recent changes
  - draft-ietf-rtcweb-data-channel-06
  - draft-ietf-rtcweb-data-protocol-01
- Open Issues

Changes in the Latest Revision of the Data Channel IDs

- draft-ietf-rtcweb-data-channel-06 covers requirements, use-cases, the data transfer and the closing of data channels. The setting up of data channels is out of scope of this document.
- draft-ietf-rtcweb-data-protocol-01 focuses on the in-band negotiation of symmetric data channels (two way handshake)
- The results of the discussion at the last IETF are reflected.

Open Issue No 1:
DTLS Overhead

- Issue
  - The DTLS overhead depends on the Cipher Suite. Therefore taking the maximum of currently defined cipher suites is not future proof.
- Proposed Resolution
  - Just specify the IP Layer MTU

Open Issue No 2: Large Messages Block Other Channels

- Issue
  - The sending of a large message on a data channel blocks the sending of messages on other data channels.
- Proposed Resolution
  - All implementation must support the PPID based fragmentation and reassembly method for ordered reliable data channels. For other data channels the message size is limited to avoid the issue.
  - When SCTP level interleaving (as specified in draft-stewart-tsvwg-sctp-ndata) is available, it is used and the length restriction for unordered or unreliable data channels is removed.

Open Issues No 3: Message Size Limitations

- Issue
  - What are message size limitations?
  - For interoperability one needs a minimal upper limit.
- Proposed Resolution
  - Message size support for a minimum of 100 MB
    - WebSockets has a limit of ~2^71 or so, but removed any limit and said implementations can impose maximums to avoid DoS attacks (section 10.4)
  - If SCTP level interleaving is not supported, the message size limit of unordered or unreliable data channels is 10 KB

Open Issues No 4: Usage of the Nagle Algorithm

- Issue
  - The Nagle algorithm (which is enabled by default in SCTP) tries to reduce the number of packets by delaying small packets.
  - This does result in performance degradation, but not in interoperability problems.
- Proposed Solution
  - Disable Nagle Algorithm

Open Issues No 5: Initial Number of Streams

- Issue
  - When setting up an SCTP association, the initial number of outgoing streams (between 1 and 65535) is negotiated. During the lifetime of the association, this number can be increased (up to 65536 streams).
  - Implementing the increase of streams (above the SCTP stack) adds some complexity.
- Proposed Solution
  - No change: The complexity added is almost all needed anyways to handle blocking Opens while waiting for association init, and reporting errors if for any reason more channels aren't available.

Open Issues No 6:
Out of Band Negotiation

- Issue
  - It is not clear whether the same stream ID is used in both directions.
  - It is not clear whether the odd/even role applies.
  - It is not clear what the impact with in-band negotiation is.
  - JS libraries might use DataChannels, either in-band or out-of-band
- Proposed Solution
  - Clarify that the same stream ID is used in both directions.
  - The even/odd rule does not apply to out-of-band negotiated channels
  - Collisions between out-of-band-negotiated channels and in-band negotiated channels result in errors when detected
  - In-band and out-of-band negotiation must be able to be mixed to support applications using external JS libraries without needing a bunch of extra interfaces

Open Issues No 6: Out of Band Negotiation (cont)

- In the JS API, you should set up your side of the channel before sending the configuration out-of-band for the other side to install
- For adding out-of-band channels after initial O/A, collisions with in-band allocation MUST be avoided. This can be done by:
  - Not using in-band (including libraries)
  - Verifying that you're using the correct Oddness and an unused channel (requires tracking all in-use channels, including libraries)
  - Per the JS API, asking the stack/protocol for an unused channel of the correct Oddness