Abstract

The Web Real-Time Communication (WebRTC) working group is charged to provide protocols to support for direct interactive rich communication using audio, video, and data between two peers’ web-browsers. This document specifies an actual (minor) protocol for how the JS-layer dataChannel objects provide the data channels between the peers.

This minor protocol has applicability to other bidirectional uses of SCTP.
Table of Contents

1. Introduction .................................................. 3
2. Opening handshake ............................................ 3
3. Conventions ..................................................... 3
   3.1. Terminology ............................................. 3
4. Control Messages .............................................. 3
   4.1. Adding a Channel ......................................... 6
   4.2. Closing a Channel ....................................... 6
5. Sending and Receiving Data .................................... 7
6. Security Considerations ....................................... 7
7. IANA Considerations .......................................... 7
8. Acknowledgments .............................................. 7
9. References ..................................................... 7
   9.1. Normative References .................................... 7
   9.2. Informational References ................................. 8
Authors’ Addresses .............................................. 8
1. Introduction

The Data Channel Protocol is designed to provide, in the WebRTC context [I-D.ietf-rtcweb-overview], a generic transport service allowing Web Browser to exchange generic data in a bidirectional peer to peer fashion. As discussed in [I-D.jesup-rtcweb-data] the protocol uses Stream Control Transmission Protocol (SCTP) [RFC4960] encapsulated on Datagram Transport Layer Security (DTLS) [RFC6347] to benefit from their transport and security already standardized features.

2. Opening handshake

The opening handshake is based on the multimedia session description exchange that happens between the browsers, typically through a Web Server acting as the signaling service.

The draft [I-D.ietf-mmusic-sctp-sdp] defines the protocol identifier, ‘SCTP/DTLS’, and defines how to establish an SCTP association over DTLS using the Session Description Protocol (SDP).

The SCTP association is created with the number of streams specified by the application, and if not specified, then it shall default to 16 streams.

It is recommended that additional streams be available dynamically based on [RFC6525].

3. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3.1. Terminology

This document uses the following terms:

Association: An SCTP association.
Stream: A unidirectional stream of an SCTP association. It is uniquely identified by a stream identifier.

4. Control Messages

Data Channel Control Messages are sent to manage opening
bidirectional channels. A DATA_CHANNEL_OPEN is sent on the Stream that is intended to be used to send in that direction, and a response (DATA_CHANNEL_OPEN_RESPONSE) using the same structure is sent back on the Stream to be used for the other direction, with a reverse_direction_stream entry holding the Stream number the DATA_CHANNEL_OPEN was sent on. This allows association of the Streams that define the bidirectional channel.

Errors are returned by setting the first 16 bits of the msg_type_data field to a non-0 value. In this case the original sender of DATA_CHANNEL_OPEN shall close the channel.
Internet-Draft         data protocol P2P in RTCWEB            March 2012

0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Type = 0    | Reserved|U|B|E|    Length                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              TSN                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Stream Identifier S      |   Stream Sequence Number n    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        PPID = Data Channel Control            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| msg_type     | channel_type  |         flags                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| reverse_direction_stream    |   reliability_parameters      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   msg_type_data                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

'msg_type' contains:
DATA_CHANNEL_OPEN                     0
DATA_CHANNEL_OPEN_RESPONSE            1

'channel_type' contains:
DATA_CHANNEL_RELIABLE                 0
DATA_CHANNEL_RELIABLE_STREAM          1
DATA_CHANNEL_UNRELIABLE               2
DATA_CHANNEL_PARTIAL_RELIABLE_REXMIT  3
DATA_CHANNEL_PARTIAL_RELIABLE_TIMED   4

'flags' contains:
DATA_CHANNEL_FLAG_OUT_OF_ORDER_ALLOWED 0x0001
all other bits are reserved

'msg_type_data' contains:
for DATA_CHANNEL_OPEN:
a DOMString label for the data channel
for DATA_CHANNEL_OPEN_RESPONSE:
a 16-bit value for errors or 0 for no error

'reliability_parameters' contains (as unsigned 16-bit numbers):
for DATA_CHANNEL_PARTIAL_RELIABLE_REXMIT:
the maximum number of retransmits or 0 for none
for DATA_CHANNEL_PARTIAL_RELIABLE_TIMED:
the maximum time in ms to attempt to deliver the data
or 0 for no attempts.
Figure 1: Data Channel control messages: OPEN and OPEN_RESPONSE Messages

Note that the forward and reverse-direction Stream numbers may be different if both sides attempt to open the same Stream number at the same time, and the protocol does not require they be the same.

Control messages shall be sent on the relevant Streams, and shall be sent using reliable, in-order delivery.

4.1. Adding a Channel

When one side wants to add a channel, it picks an unused Stream, and if need be, negotiates an increase in the number of Streams available. It then sends a DATA_CHANNEL_OPEN control message on the Stream, with reverse_direction_stream set to 0, though that value MUST be ignored by the recipient.

When a DATA_CHANNEL_OPEN is received on a Stream, an unused Stream is picked, and if need be, negotiates an increase in the number of Streams available. A DATA_CHANNEL_OPEN_RESPONSE is sent on the Stream, with reverse_direction_stream set to the Stream the DATA_CHANNEL_OPEN request came in on.

When a DATA_CHANNEL_OPEN_RESPONSE is received, the reverse_direction_stream value is matched against all pending DATA_CHANNEL_OPENs. If no match can be found, the DATA_CHANNEL_OPEN_RESPONSE should be ignored. If a match is found, then if the error value is 0 the bidirectional Data Channel is now established and ready for use. If the error value is non-zero, the open failed, and the originator should close down the originally-selected Stream and notify the application.

An alternative to embedding the control messages on the Streams used for data transfer would be to put them on a reserved Stream (0 for example), and add an additional field for the forward_direction_stream.

The channel_type and reliability_parameters fields of the DATA_CHANNEL_OPEN message MUST be used to set up the reverse side of the Data Channel so that both directions use the same options.

4.2. Closing a Channel

Data Channels shall be closed by doing an SCTP Stream Sequence Number reset, which should guarantee that all data sent before closing is delivered or has been abandoned (for partially-reliable Data Channels).
5. Sending and Receiving Data

Data shall be sent using PPID’s other than the Data Channel Control PPID. These PPID’s should be registered with IANA via (TBD). The meaning of these data PPIDs and the format of the data shall be specific to the usage of this protocol, and typically shall be provided to the higher layers to allow proper decoding of the data.

For WebRTC, data PPID’s for DOMStrings and binary data blobs shall be created.

All data sent on a Data Channel in both directions MUST be sent over the underlying Stream using the reliability defined when the Data Channel was opened.

It is recommended that message size be kept within certain bounds (TBD).

6. Security Considerations

To be done.

7. IANA Considerations

This document also defines three new SCTP Payload Protocol Identifiers (PPIDs). RFC 4960 [RFC4960] creates the registry from which these identifiers have been assigned. The following values have been reserved:

WebRTC Control - #To Be Assigned
DOMString - #To Be Assigned
Binary Data - #To Be Assigned

8. Acknowledgments

Many thanks for comments, ideas, and text ... (TBD)

9. References

9.1. Normative References


9.2. Informational References

Alvestrand, H., "Overview: Real Time Protocols for Browser-based Applications", draft-ietf-rtcweb-overview-02 (work in progress), September 2011.


Authors’ Addresses

Randell Jesup
Mozilla
USA
Email: randell-ietf@jesup.org

Salvatore Loreto
Ericsson
Hirsalantie 11
Jorvas 02420
Finland
Email: salvatore.loreto@ericsson.com
Michael Tuexen
Muenster University of Applied Sciences
Stegerwaldstrasse 39
Steinfurt 48565
Germany

Email: tuexen@fh-muenster.de