RTCWeb Working Group R. Jesup
Internet-Draft Mozilla
Intended status: Standards Track S. Loreto
Expires: December 11, 2012 Ericsson
M. Tuexen
Muenster Univ. of Appl. Sciences
June 9, 2012

WebRTC Data Channel Protocol
draft-jesup-rtcweb-data-protocol-01.txt

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.

Status of this Memo

Copyright Notice

Table of Contents

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.ietf-rtcweb-data-channel] the protocol uses Stream Control Transmission Protocol (SCTP) [RFC4960] encapsulated on Datagram Transport Layer Security (DTLS) [RFC6347] as described in [I-D.tuexen-tsvwg-sctp-dtls-encaps] to benefit from their already standardized transport and security features.

2. 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. 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.

Channel: A bidirectional channel consisting of two SCTP streams.

4. 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.

[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 SHOULD default to 16 streams.

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

5. Control Messages

Data Channel Control Messages are sent to manage opening bidirectional channels. A DATA_CHANNEL_OPEN_REQUEST message is sent on the Stream that is intended to be used to send in that direction, and a response (DATA_CHANNEL_OPEN_RESPONSE) 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_REQUEST was sent on. This allows association of the Streams that define the bidirectional channel. Finally, a DATA_CHANNEL_ACK is sent on the original Stream to complete the 3-way handshake.

Errors are returned by setting the error field of the DATA_CHANNEL_OPEN_RESPONSE message to a non-0 value. In this case the original sender of DATA_CHANNEL_OPEN_REQUEST shall close the channel.

5.1. DATA_CHANNEL_OPEN_REQUEST Message

This message is sent initially on the stream used for user messages using the channel.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Message Type |  Channel Type |            Flags              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Reliability Parameter     |          Priority             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
\                                                               /
|                             Label                             |
/                                                               \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Message Type: 1 byte (unsigned integer)
   This field holds the IANA defined message type for the the DATA_CHANNEL_OPEN_REQUEST message. The suggested value of this field for IANA is 0x00.

Channel Type: 1 byte (unsigned integer)
   This field specifies the type of the channel to be opened:
   DATA_CHANNEL_RELIABLE (0x00): The channel provides a reliable bi-directional communication channel.
   DATA_CHANNEL_RELIABLE_STREAM (0x01): TBD.

DATA_CHANNEL_UNRELIABLE (0x02): TBD.

DATA_CHANNEL_PARTIAL_RELIABLE_REXMIT (0x03): The channel provides a partial reliable bi-directional Communication channel. User messages will not be retransmitted more times than specified in the Reliability Parameter.

DATA_CHANNEL_PARTIAL_RELIABLE_TIMED (0x04): The channel provides a partial reliable bi-directional Communication channel. User messages might not be transmitted or retransmitted after a specified life-time given in milli-seconds in the Reliability Parameter. This life-time starts when providing the user message to the Javascript engine.

Flags: 2 bytes (unsigned integer)

This field contains the bitwise OR of the following flags:

DATA_CHANNEL_FLAG_OUT_OF_ORDER_ALLOWED (0x0001): If this flag is set, the channel does not need to preserve the message sequencing.

Reliability Parameter: 2 bytes (unsigned integer)

This field is ignored if a reliable channel is used. If a partial reliable channel with limited number of retransmissions is used, this field specifies the number of retransmissions. If a partial reliable channel with limited lifetime is used, this field specifies the maximum life-time in milli seconds.

Priority: 2 bytes (integer)

The priority of the channel.

Label: Variable Length (sequence of characters)

The name of the channel.

5.2. DATA_CHANNEL_OPEN_RESPONSE Message

This message is sent in response to an DATA_CHANNEL_OPEN_REQUEST message on the stream used for user messages using the channel. Messages with the error field set to non-0 values can be sent on any stream; it is suggested that they be sent (if possible) on an unused stream.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Message Type |     Error     |            Flags              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       Reverse Stream          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Message Type: 1 byte (unsigned integer)

This field holds the IANA defined message type for the the DATA_CHANNEL_OPEN_RESPONSE message. The suggested value of this field for IANA is 0x01.

Error: 1 byte (unsigned integer)
   TBD.
Flags: 2 bytes (unsigned integer)
   TBD.
Reverse Stream: 2 bytes (unsigned integer)
   The identifier for the incoming stream of the channel. The corresponding DATA_CHANNEL_OPEN_REQUEST message was received on this stream.

5.3. DATA_CHANNEL_ACK Message

This message is sent in response to an DATA_CHANNEL_OPEN_RESPONSE message on the stream used for user messages using the channel. Reception of this message tells the opener that the channel setup handshake is complete.

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  Message Type |
   +-+-+-+-+-+-+-+-+
```

Message Type: 1 byte (unsigned integer)
   This field holds the IANA defined message type for the the DATA_CHANNEL_ACK message. The suggested value of this field for IANA is 0x02.

6. Procedures

6.1. Adding a Channel

When one side wants to add a channel, it picks an unused outgoing stream; if no unused streams are available a negotiation to increase the number is done. It then sends a DATA_CHANNEL_OPEN_REQUEST control message on the outgoing stream.

When an DATA_CHANNEL_OPEN_REQUEST is received on an incoming stream, an unused outgoing stream is picked; if no unused streams are available a negotiation to increase the number is done. A DATA_CHANNEL_OPEN_RESPONSE message is sent on the outgoing stream, with the Reverse Stream field set to the incoming stream the DATA_CHANNEL_OPEN_REQUEST message came in on.

When a DATA_CHANNEL_OPEN_RESPONSE is received, the Reverse Stream value is matched against all pending DATA_CHANNEL_OPEN_REQUEST messages. If no match can be found, the DATA_CHANNEL_OPEN_RESPONSE message SHOULD be ignored. If a match is found, then if the error

value is 0 a DATA_CHANNEL_ACK message is sent on the originator's outgoing Stream for the channel. If the error value is non-zero, the open failed, and the originator SHOULD close down the originally-selected outgoing stream and notify the application.

The channel_type and reliability_parameters fields of the DATA_CHANNEL_OPEN_REQUEST message MUST be used to set up the reverse side of the Data Channel so that both directions use the same options. So both directions are either reliable or use the PR-SCTP extension defined in [RFC3758] using the same policy and parameter.

6.2. Closing a Channel

Data Channels shall be closed by resetting the outgoing stream If an incoming stream is reset by the peer, an corresponding outgoing stream reset SHOULD be issued. If both streams of a channel are reset, the channel is closed and the streams are available for reuse for new channel opens.

6.3. 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.

Data may be sent before the 3-way handshake is complete; if so it must be sent with in-order delivery set in order to avoid race conditions cause by a handshake message being lost. This is an exception to the requirement to send all data using the channel reliability settings.

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

7. Security Considerations

To be done.

8. 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

9. Acknowledgments

The authors wish to thank ... for there invaluable comments.

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3758] Stewart, R., Ramalho, M., Xie, Q., Tuexen, M., and P. Conrad, "Stream Control Transmission Protocol (SCTP) Partial Reliability Extension", RFC 3758, May 2004.

[RFC4960] Stewart, R., "Stream Control Transmission Protocol", RFC 4960, September 2007.

[RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, January 2012.

[RFC6525] Stewart, R., Tuexen, M., and P. Lei, "Stream Control Transmission Protocol (SCTP) Stream Reconfiguration", RFC 6525, February 2012.

[I-D.ietf-mmusic-sctp-sdp]
Loreto, S. and G. Camarillo, "Stream Control Transmission Protocol (SCTP)-Based Media Transport in the Session Description Protocol (SDP)", draft-ietf-mmusic-sctp-sdp-01 (work in progress), March 2012.

[I-D.tuexen-tsvwg-sctp-dtls-encaps]
Jesup, R., Loreto, S., Stewart, R., and M. Tuexen, "DTLS Encapsulation of SCTP Packets for RTCWEB", draft-tuexen-tsvwg-sctp-dtls-encaps-00 (work in progress), March 2012.

10.2. Informational References

[I-D.ietf-rtcweb-overview]
Alvestrand, H., "Overview: Real Time Protocols for Brower-based Applications", draft-ietf-rtcweb-overview-03 (work in progress), March 2012.

[I-D.ietf-rtcweb-data-channel]
Jesup, R., Loreto, S., and M. Tuexen, "RTCWeb Datagram Connection", draft-ietf-rtcweb-data-channel-00 (work in progress), March 2012.

Authors' Addresses

Randell Jesup
Mozilla
US

Email: randell-ietf@jesup.org

Salvatore Loreto
Ericsson
Hirsalantie 11
Jorvas 02420
FI

Email: salvatore.loreto@ericsson.com

Michael Tuexen
Muenster University of Applied Sciences
Stegerwaldstrasse 39
Steinfurt 48565
DE

Email: tuexen@fh-muenster.de