RTCWeb data channel management
draft-marcon-rtcweb-data-channel-management-00

Abstract

The Real-Time Communication in WEB-browsers (RTCWeb) working group is
charged to provide protocols to support direct interactive rich
communication using audio, video, and data between two peers' web-
browsers. For the support of data communication, the RTCWeb working
group has in particular defined the concept of bi-directional data
channels over SCTP. How to transport application messages on these
data channels seems straightforward (i.e. they can be carried as SCTP
user messages), however it is yet to be decided how to establish and
manage these data channels. This document specifies a method for
this, which relies first on a lightweight and scalable out-of-band
negotiation of data channel configurations (within the SDP offer/
answer exchange) and second on the signaling of the configuration in
use in the SCTP user message itself. Once these configurations are
negotiated, further creations of data channels can occur purely in-
band by simply sending user messages, which avoids to define a new
in-band data channel protocol.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

[I-D.ietf-rtcweb-data-channel] provides use cases and requirements
for the definition of RTCWeb data channels, and outlines how the
Stream Control Transmission Protocol (SCTP) [RFC4960] encapsulated
within Datagram Transport Layer Security (DTLS) [RFC6347] can be used
for this purpose. While some of these requirements easily map to
existing capabilities of the SCTP protocol and extensions (e.g.
application messages can be carried as SCTP user messages), SCTP and
existing SCTP extensions do not natively support the following
requirements:

o data channel bidirectionality (this can be achieved by pairing one
   SCTP outbound stream and one SCTP inbound stream)

o data channel priority

o partial reliability of delivery, based on a maximum number of
   retransmissions

o general data channel setup

For setting up the SCTP association, the in-band SCTP association
initialization is assisted out-of-band by JSEP [I-D.ietf-rtcweb-jsep]
and the SDP Offer/Answer model [RFC3264]. For setting up each data
channel, several approaches can be considered:

1. a purely in-band data channel setup - such a protocol does not
    exist today.

2. a hybrid in-band / out-of-band data channel setup, where the in-
    band signaling relies on a new protocol defined on top of SCTP
    user messages. The proposal [I-D.jesup-rtcweb-data-protocol]
    follows this approach.

3. an out-of-band negotiation of data channel configurations,
    minimally assisted by some lightweight in-band signaling allowing
    further in-band creations of data channels.

This document describes the latter approach, preferred by the author
for the following reasons:

o Minimal need for SDP renegotiation: the initial offer/answer for
   establishing the SCTP association is often enough.

o Scalability of the SDP signaling: typically it is as light as a
   couple of attribute lines regardless of the number of data
   channels created in the session.

o Potential interoperability with other systems, due to the use of
   out-of-band signaling.

o Ability to create data channels purely in-band, once the data
   channel configurations are negotiated

o No need for a new in-band data channel control protocol.

o Speed: No control message overhead for the in-band creation of
   data channels: sending user messages automatically creates new
   data channels.

o Simplicity of implementation.

As a result, the proposal can easily cope with strenuous data
transmission scenarios, like:

o The transfer of a high number of files, eventually in parallel.

o The fast opening and closing of one data channel.

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:

   Agent: As defined in [RFC3264], an agent is the protocol
   implementation involved in the offer/answer exchange. There are
   two agents involved in an offer/answer exchange.

   Configuration: a fixed set of data channel parameters,
   constraining the configuration of some or all the data channels
   created on top of a given SCTP association.

   Data channel: A bidirectional channel consisting of paired SCTP
   outbound and inbound streams.

   In-band: transmission through the peer-to-peer SCTP association.

Out-of-band: transmission through the RTCWeb signaling path, using
JSEP [I-D.ietf-rtcweb-jsep] and the SDP Offer/Answer model
[RFC3264].

Peer: From the perspective of one of the agents in a session, its
peer is the other agent. Specifically, from the perspective of
the SDP offerer, the peer is the SDP answerer. From the
perspective of the SDP answerer, the peer is the SDP offerer.

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

4. Overview

This section provides an overview of the approach detailed in this
document.

A data channel configuration is an identified fixed set of data
channel parameters potentially applicable to some or all of the data
channels created during the session. These parameters include: order
of delivery, reliability of delivery, subprotocol.

Configurations are uniquely identified throughout the session, and
negotatied out-of-band between the endpoints. The configuration
concept is transparent to the application, which sets up and handles
each data channel individually. Whenever the application creates a
new data channel, the browser internally checks if the passed set of
parameters strictly matches an existing configuration, and if not
generates a new configuration identifier for this set. In the latter
case only does the browser trigger the application for an SDP
renegotiation.

In the SDP offer, the offerer associates to the m=application SDP
line that defines the SCTP association one attribute line per data
channel configuration.

For each data channel configuration in the offer that is accepted by
the answerer, the answerer echoes in the answer the configurations
supported and accepted. Once the offerer receives the answer and (in
case of an initial offer) the SCTP initialization is complete, each
data channel locally created using one of the accepted configurations
is signaled to the application as open for transmission.

Each created data channel is bound to to one negotiated
configuration.

By convention, the inbound and outbound streams of a data channel

have the same SCTP stream number. This stream number is selected by
the first endpoint sending a user message on this channel. Till this
happens, an open data channel has no assigned stream number.

Data channel messages are sent as SCTP user messages, preceded in the
DATA chunk User Data field by two bytes specifying data channel
configuration identifier as well as the message data framing type
(textual or binary).

A user message received on a stream number not assigned to any data
channel automatically opens a data channel, set up according to the
configuration signaled in the user message.

Closing a data channel is done in-band by resetting the Stream
Sequence Number (SSN) of the related SCTP inbound and outbound
streams, as defined in [RFC6525].

This proposal requires the registration of one SCTP Payload Protocol
Identifier.

5. **Data channel configuration and message properties**

For the proposal in this document, a data channel configuration is
characterized by the following properties:

o configuration identifier, a 12-bit integer unique across all the
   data channel configurations managed during the lifetime of an SCTP
   association.

o ownership: the configuration is owned by the endpoint which
   originated the very first offer that included this configuration,
   for a given SCTP association.

o reliability of delivery: full reliability (as per [RFC4960]) or
   partial reliability with max transmission lifetime (as per
   [RFC3758]) or partial reliability with max number of
   retransmissions.

o order of delivery: ordered or unordered

o subprotocol identifier

o subprotocol setup data, if applicable

For the proposal in this document, a data channel is characterized
from an endpoint viewpoint by the following properties:

o Configuration identifier. It can bind multiple data channels at a
   time.

o Label: local human-readable description of the data channel.

o Data channel priority

o SCTP stream number: common to the SCTP outbound stream and inbound
   stream composing the data channnel.

o state, which can have the following values:

   * Connecting: data channel opened locally, and awaiting opening
      acknowledgment by the peer.

   * Open: data channel available for bidirectional data
      transmission.

   * Closing: data channel closed locally, and awaiting closing
      acknowledgment by the peer.

   * Closed: data channel closed by the agent, and acknowledged as
      closed by the peer.

A message sent over a data channel inherits from the transmission
properties configured to this data channel: reliability and order of
delivery. In addition, the message is characterized by the following
message-specific property:

o transport format encoding: text or binary.

Note that for API simplification purpose, reliability, order of
delivery and payload protocol identifier are not configurable per
user message, but per data channel only.

The payload protocol identifier (PPID) field present in SCTP DATA
chunks is used to signal the data framing described in this document.
This value is to be obtained from the SCTP Payload Protocol
Identifiers registry managed by IANA.

6. **Procedures**

6.1. **Initialization**

The PeerConnection and underlying SCTP association are initialized
with N data channels, all in Closed state, and with respective
outbound and inbound stream numbers ranging from 0 to N-1.   The

number N can be specified by the application or else defaults to 16.

6.2. Opening a data channel out-of-band

An application creating a data channel providing some data channel
parameters to the agent's browser. If the subset of these parameters
composing a data channel configuration does not strictly match an
existing configuration, the browser assigns a new configuration
identifier to this subset, and binds it to the data channel. The
configuration identifier is generated incrementally, starting from
zero for each SCTP association. Identifiers of configurations
rejected by the answerer must never be used again.

In addition, if the application requests the creation:

o at a time where the endpoint is in a stable state with an SCTP
   association already set up, and if the match of configuration is
   successful, the browser then sets the data channel state to Open.

o Otherwise the browser sets the data channel state to Connecting.
   Moreover, unless the endpoint is in an init state and createOffer
   has not yet been called, the browser notifies to the application
   the need for an SDP renegotiation.

The created data channel has no assigned SCTP stream number yet. At
this stage though the user agent can anticipate a shortage of
available SCTP streams and send in-band the request to increase the
number of SCTP streams.

The new offer (if any) contains one 'm-line' for the SCTP assocation,
and one attribute line per data channel configuration. This list of
configurations must include the new configurations as well as all the
configurations successfully negotiated in previous offer/answer
exchanges for this SCTP association.

The peer's browser receiving the offer does the following:

1. for the data channels that are in Open state but which are bound
    to a configuration no longer present in the offer, change their
    state to Closed

2. for each newly offered configuration, the peer's browser then
    informs the application of a new offered data channel along with
    the configuration specifics. The application can accept this
    data channel (intended: configuration) by creating a new data
    channel using the configuration parameters. Not doing so will
    mean to reject the configuration in the answer.

Note: for each new configuration, the offerer expectedly creates
one data channel or more, whereas the answerer creates one data
channel only. How the final data channel pairing (and SCTP stream
number assignment) is resolved is further explained in this
document.

Note: the answerer can only reject new configurations,
configurations previously negotiated cannot be removed from the
configuration list associated with the SCTP association.

The agent's browser receiving the answer does the following:

1. for the data channels not in Closed state and bound to a
    configuration no longer listed in the answer, change their state
    to Closed.

2. for each newly offered data channel configuration accepted by the
    answerer, change the state of any data channel bound to this
    configuration from Connecting to Open.

6.3. Opening a data channel in-band

Each user message sent in a data channel includes the identifier of
the configuration which this data channel is bound to. This
signaling allows to enable or speed up the opening of new data
channels in-band:

o   Case A: In a stable state, the local creation of a data channel
    with parameters mapping to a negotiated configuration creates the
    data channel in Open state immediately, and does not signal this
    to the peer. Some time later, when the application sends its
    first message on this data channel:

    1. the agent's browser selects for the lifetime of the data
        channel an SCTP outbound stream number not used by any
        channel.

    2. it then sends the user message over this SCTP stream.

    3. the peer's browser SACKnowledges the user message, using an
        SCTP stream of same stream number (expectedly unused). It
        then notifies the peer's application of the data channel
        opening.

o   Case B: Once the answerer has accepted a new offerer's
    configuration, and has subsequently opened a data channel bound to
    this configuration, the answerer's application may choose to send
    user messages on this channel immediately. The offerer receiving

this message should:

1. route this message to one of the Connecting-state data
    channels bound to the same configuration.

2. change its state to Open, and sets its SCTP outbound stream
    number (expectedly unused) to the SCTP inbound stream number
    of the message.

3. delivers the message to the application.

6.4. Closing a data channel

When the application requests to close a data channel, the agent's
browser initiates an in-band Stream Sequence Number (SSN) reset of
the related SCTP inbound and outbound streams. These streams are
then available for further reuse.

6.5. Sending and Receiving Data

To expose to upper layers an API similar to the Web Socket API
[WSAPI], the agent's browser needs to specify to the peer's browser
the framing type of each data channel message, amongst binary or text
(UTF-8).

In addition, each user message needs to carry the identifier of the
configuration which the data channel is bound to.

For the sending of a user message over an opened data channel, the
agent's browser:

1. converts the message data from UTF-16 to UTF-8 if provided by the
    application as a DOMString.

2. sets in the DATA chunk the Unordered bit, Payload Protocol
    Identifier and Stream Number as per data channel configuration.

3. constructs the User Data field as the framing type byte(s)
    followed by the (converted) message data

```
 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
+------------------------+------+------------------------------+
|        Config ID       | Type |          Payload Data        |
+------------------------+------+ - - - - - - - - - - - - - - -+
:                    Payload Data continued ...                :
+ - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -+
|                    Payload Data continued ...                |
+--------------------------------------------------------------+
```
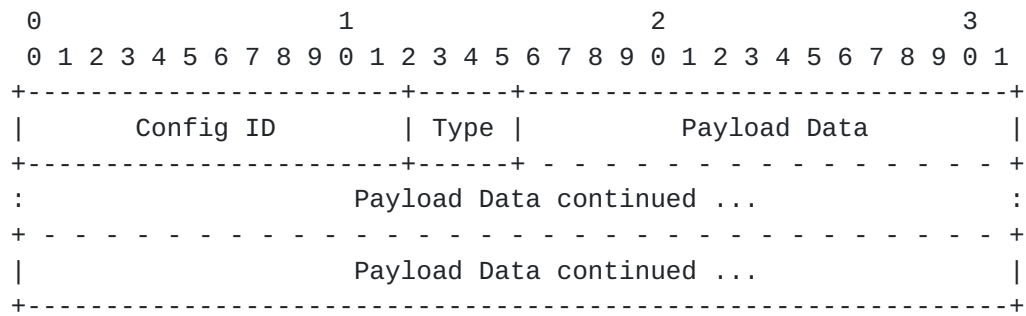
                      RTCWeb data framing

Config ID : 12 bits

   Identifies a data channel configuration negotiated out of band
   between the two endpoints.

      Note: to be considered if the Config ID should be included in
      all user messages.

Type : 4 bits

   Defines the data framing type of "Payload data". If an unknown
   type is received, the receiving endpoint must reject the user
   message. The following values are defined:

   * x0 denotes a binary frame

   * x1 denotes a UTF-8 encoded text frame

   * x2-xFFF are reserved for further use

An agent's browser receiving a user message on a data channel behaves
as follows:

o If the configuration identifier in the message does not map to any
   negotiated configuration, or if the data channel (identified by
   the message stream number) is in Closing or Connecting state,
   reject (or discard?) the message.

o Otherwise (i.e. supported configuration identifier and Open
   state):

      If the endpoint has just sent the very first user message on
      this data channel and has not yet received the SACK, it means
      that both endpoints attempt to dynamically create a data
      channel by the sending of a user message. In this case, if the
      endpoint has ownership of the signaled configuration, the

browser must discard (reject?) the message.

> Note: another way of avoiding this conflict is to state by
> convention that the endpoint which initiated the offer for
> the SCTP association establishment owns all the even stream
> numbers, while the other endpoint owns all the odd stream
> numbers.

Otherwise deliver the message.

o Otherwise (i.e. supported configuration identifier and Closed
   state):

   If a Connecting-state data channel exists with no assigned
   stream number, open it with a stream number set to the message
   stream number.

   If not, open a new data channel with a stream number set to the
   message stream number

   Finally, deliver the message.

6.6. Out-of-band signaling

To signal the data channel configurations intended for use during the
lifetime of an SCTP association, the agent completes the SDP <fmt>
section of the m=application SDP line defined for the SCTP
association. For each data channel configuration previously
negotiated or newly added at the time of offer generation, the
agent's browser:

o must specify: configuration identifier.

o may specify: order of delivery, reliability of delivery,
   subprotocol, subprotocol configuration data.

As an example in the offer (line split for readability):

```
m=application 54111 DTLS/SCTP 5000
a=sctpmap:5000 webrtc-DataChannel 16
a=sctp-protocol:5000 config=1;label="channel 1";subprotocol="chat";
a=sctp-protocol:5000 config=2;label="channel 2";
subprotocol="file transfer";max_retr=3
```

data channel configuration setup in SDP offer

An in the returned answer (line split for readability):

```
m=application 54111 DTLS/SCTP 5000
a=sctpmap:5000 webrtc-DataChannel 16
a=sctp-protocol:5000 config=2;label="channel 2";
subprotocol="file transfer";max_retr=3
```

                data channel configuration setup in SDP answer

   In reply to this offer, the peer constructs in the answer the data
   channel configuration list of the SCTP association as follows:

   o drop any unwanted or unsupported data channel configuration

   o echo the other configurations, and preserve the following
      properties at least: configuration identifier, subprotocol (if
      any). Specifies also the peer-specific properties (subprotocol
      setup data).

   This may need to be specified via MMUSIC.

7. Security Considerations

   To be completed.

8. IANA Considerations

   To be completed.

9. Acknowledgments

   The authors wish to thank Richard Ejzak, ... for their invaluable
   comments.

10. References

10.1. Normative References

   [I-D.ietf-rtcweb-jsep]
              Uberti, J. and C. Jennings, "Javascript Session
              Establishment Protocol", draft-ietf-rtcweb-jsep-02 (work
              in progress), October 2012.

   [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.

10.2. Informative References

   [I-D.ietf-rtcweb-data-channel]
               Jesup, R., Loreto, S., and M. Tuexen, "RTCWeb Datagram
               Connection", draft-ietf-rtcweb-data-channel-02 (work in
               progress), October 2012.

   [I-D.jesup-rtcweb-data-protocol]
               Jesup, R., Loreto, S., and M. Tuexen, "WebRTC Data Channel
               Protocol", draft-jesup-rtcweb-data-protocol-03 (work in
               progress), September 2012.

   [ITU.T140.1998]
               "Protocol for Multimedia Application Text Conversation",
               ITU-T Recommendation T.140, February 1998.

   [RFC3264]   Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model
               with Session Description Protocol (SDP)", RFC 3264,
               June 2002.

   [WebRtcAPI]
               Bergkvist, A., Burnett, D., Narayanan, A., and C.
               Jennings, "WebRTC 1.0: Real-time Communication Between
               Browsers", World Wide Web Consortium WD WD-webrtc-
               20120821, August 2012,
               <http://www.w3.org/TR/2012/WD-webrtc-20120821>.

   [WebSocketAPI]
               Hickson, I., "The WebSocket API", World Wide Web
               Consortium LastCall WD-websockets-20120809, August 2012,
               <http://www.w3.org/TR/2012/WD-websockets-20120809>.

Author's Address

    Jerome Marcon
    Alcatel-Lucent
    Route de Villejust
    Nozay 91620
    France

    Email: jerome.marcon@alcatel-lucent.com