DISPATCH
Internet-Draft
Intended status: Standards Track
Expires: April 17, 2014
October 14, 2013

R. Ejzak
J. Marcon
Alcatel-Lucent

SDP-based WebRTC data channel negotiation
draft-ejzak-dispatch-webrtc-data-channel-sdpneg-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, where each data channel might be used to transport other protocols, called sub-protocols. Data channel setup can be done using either the in-band WebRTC Data Channel protocol or some external (in-band or out-of-band) negotiation. This document specifies how the SDP offer/answer exchange can be used to achieve such an external negotiation.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The RTCWeb working group has defined the concept of bi-directional data channels running on top of SCTP/DTLS. Each data channel consists of paired SCTP streams sharing the same SCTP Stream Identifier. Data channels are created by endpoint applications through the WebRTC API, and can be used to transport proprietary or well-defined protocols, which in the latter case can be signaled by the data channel "sub-protocol" parameter, conceptually similar to the WebSocket "sub-protocol". However, apart from the "sub-protocol" value transmitted to the peer, RTCWeb leaves open how endpoint applications can agree on how to instantiate a given sub-protocol on a data channel, and whether it is signaled in-band or out-of-band (or both). In particular, the SDP offer generated by the browser includes no channel-specific information.

This document defines SDP-based out-of-band negotiation procedures to establish WebRTC data channels for transport of well-defined sub-protocols.

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:

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

Data channel properties: fixed properties assigned to a data channel at the time of its creation. Some of these properties determine the way the browser transmits data on this channel (e.g., stream identifier, reliability, order of delivery...)

External negotiation: data channel negotiation based on out-of-band or in-band mechanisms other than the WebRTC data channel protocol.

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

In-band negotiation: data channel negotiation based on the in-band WebRTC data channel protocol defined in [I-D.ietf-rtcweb-data-protocol].

Out-of-band: transmission through the WebRTC signaling path.

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 identifier: the identifier of the outbound and inbound SCTP streams composing a data channel.

4. WebRTC Data Channels

This section summarizes how WebRTC data channels work in general.

A WebRTC application creates a data channel via the WebRTC Data Channel API, by providing a number of setup parameters (sub-protocol, label, reliability, order of delivery, priority). The application also specifies if the browser takes charge of the in-band negotiation using the WebRTC data protocol, or if the application intends to perform an "external negotiation" using some other in-band or out-of-band mechanism.

In any case, the SDP offer generated by the browser is per [I-D.ietf-mmusic-sctp-sdp]. In brief, it contains one m-line for the SCTP association on top of which data channels will run, and one attribute per protocol assigned to the SCTP ports:

```
m=application 54111 DTLS/SCTP 5000 5001 5002
c=IN IP4 79.97.215.79
a=sctpmap:5000 webrtc-datachannel 16
a=sctpmap:5001 bfcp 2
a=sctpmap:5002 t38 1
```

Note: A WebRTC browser will only create an sctpmap attribute for the webrtc-datachannel protocol, and will not create sctpmap attributes for other protocols such as bfcp or t38. This example shows the hypothetical power of the syntax to support multiplexing of SCTP associations for different protocols on the same DTLS connection.

Note: This SDP syntax does not contain any channel-specific information.

4.1. Stream identifier numbering

Independently from the requested type of negotiation, the application creating a data channel can either pass to the browser the stream identifier to assign to the data channel or else let the browser pick one identifier from the ones unused.

To avoid glare situations, each endpoint can moreover own an exclusive set of stream identifiers, in which case an endpoint can only create a data channel with a stream identifier it owns.

Which set of stream identifiers is owned by which endpoint is determined by convention or other means.

For data channels negotiated in-band, one endpoint owns by convention the even stream identifiers, whereas the other owns the odd stream identifiers, as defined in [I-D.ietf-rtcweb-data-protocol].

For data channels externally negotiated, no convention is defined by default.

4.2. Generic external negotiation

4.2.1. Overview

In-band negotiation only provides for negotiation of data channel transport parameters and does not provide for negotiation of sub-protocol specific parameters. External negotiation can be defined to allow negotiation of parameters beyond those handled by in-band negotiation, e.g., parameters specific to the sub-protocol instantiated on a particular data channel. See Section 5.1.2 for an example of such a parameter.

The following procedures are common to all methods of external negotiation, whether in-band (communicated using proprietary means on an already established data channel) or out-of-band (using SDP or some other protocol associated with the signaling channel).

4.2.2. Opening a data channel

In the case of external negotiation, the endpoint application has the option to fully control the stream identifier assignments. However these assignments have to coexist with the assignments controlled by the browser for the in-band negotiated data channels (if any). It is the responsibility of the application to ensure consistent assignment of stream identifiers.

When the application requests the creation of a new data channel to be set up via external negotiation, the browser creates the data channel locally without sending any DATA CHANNEL OPEN message in-band, and sets the data channel state to Connecting if the SCTP association is not yet established, or sets the data channel state to Open if the SCTP association is already established.

The application then externally negotiates the data channel properties and sub-protocol properties with the peer’s application.

[ASSUMPTION] The peer must then symmetrically create a data channel with these negotiated data channel properties. This is the only way for the peer's browser to know which properties to apply when transmitting data on this channel. The browser must allow data channel creation with any non-conflicting stream identifier so that both peers can create the data channel with the same stream identifier.

In case the external negotiation is correlated with an SDP offer/ answer exchange that establishes the SCTP association, the SCTP initialization completion triggers each endpoint's browser to change the data channel state from Connecting to Open.

Each application must ensure that a data channel is in the Open state both locally and at the peer prior to sending data. This document includes procedures for doing so that are specific to using SDP offer /answer for external negotiation.

[ACTION ITEM] Determine if these procedures are fully consistent with the data channel design and whether additional clarification is needed in data channel documents to ensure proper support of external negotiation.

4.2.3. Closing a data channel

When the application requests the closing of an externally negotiated data channel, the browser always performs an in-band SSN reset for this channel.

Depending upon the method used for external negotiation and the sub-protocol associated with the data channel, the closing might in addition be signaled to the peer via external negotiation.

5. SDP-based external negotiation

This section defines a method of external negotiation by which two WebRTC endpoints can negotiate data channel-specific and sub-protocol-specific parameters, using the out-of-band SDP offer/answer exchange.

5.1. SDP syntax

Two new SDP attributes are defined to support external negotiation of data channels. The first attribute provides for negotiation of channel-specific parameters. The second attribute provides for negotiation of sub-protocol-specific parameters.

5.1.1. SDP attribute for data channel parameter negotiation

Associated with the m line that defines the SCTP association for data channels (defined in Section 4), each SDP offer and answer includes an attribute line that defines the data channel parameters for each data channel to be negotiated. Each attribute line specifies the following parameters for a data channel: Stream Identifier, sub-protocol, label, reliability, order of delivery, and priority. The following is an example of the attribute line for sub-protocol "BFCP" and stream id "2" :

```
a=webrtc-DataChannel:5000 stream=2;label="channel 2"; \
  subprotocol="BFCP";max_retr=3
```

This line MUST be replicated without changes in the SDP answer, if the answerer accepts the offered data channel.

This line MUST be replicated without changes in any subsequent offer or answer, as long as the data channel is still opened at the time of offer or answer generation.

> Note: This attribute was defined in old version 03 of the following draft but was removed along with any support for SDP external negotiation in subsequent versions: [I-D.ietf-mmusic-sctp-sdp].
>
> Note: This document does not provide a complete specification of how to negotiate the use of a data channel to transport BFCP. Procedures specific to each sub-protocol such as BFCP will be documented elsewhere. The use of BFCP is only an example of how the generic procedures described herein might apply to a specific sub-protocol.

5.1.1.1. stream parameter

The ’stream’ parameter indicates the actual stream identifier within the association used to form the channel. Stream is a mandatory parameter.

```
stream-attr       =  "a=stream=" streamidentifier
streamidentifier      =  1*DIGIT
```

5.1.1.2. label parameter

The ’label’ parameter indicates the name of the channel. It represents a label that can be used to distinguish, in the context of the WebRTC API, an RTCDataChannel object from other RTCDataChannel objects. Label is a mandatory parameter.

```
label-attr       =  "a=label=" labelstring
labelstring      =  text
text             =  byte-string
```

5.1.1.3. subprotocol parameter

The ’subprotocol’ parameter indicates which protocol the client expects to exchange via the channel. Subprotocol is a mandatory parameter.

```
subprotocol-attr       =  "a=subprotocol=" labelstring
labelstring            =  text
text                   =  byte-string
```

[ACTION ITEM] The IANA registry to be used for the subprotocol parameter is still to be determined. It also needs to be determined what the relationship is to existing registries and how to reference already-existing protocols.

5.1.1.4. max_retr parameter

The ’max_retr’ parameter indicates the max times a user message will be retransmitted. The max_retr parameter is optional with default value unbounded.

```
maxretr-attr       =  "a=maxretr=" maxretrvalue
maxretrvalue       =  1*DIGIT
```

5.1.1.5. max_time parameter

A user messages will no longer be transmitted or retransmitted after a specified life-time given in milliseconds in the ’max_time’ parameter. The max_time parameter is optional with default value unbounded.

```
maxtime-attr       =  "a=maxtime=" maxtimevalue
maxtimevalue       =  1*DIGIT
```

5.1.1.6. unordered parameter

The ’unordered’ parameter indicates that DATA chunks in the channel MUST be dispatched to the upper layer by the receiver without any attempt to reorder. The unordered parameter is optional. If the unordered parameter is absent, the receiver is required to deliver DATA chunks to the upper layer in proper order.

5.1.2. Sub-protocol specific attributes

In the SDP, each data channel declaration MAY also be followed by other SDP attributes specific to the sub-protocol in use. Each of these attributes is represented by one new attribute line, and it includes the contents of a media-level SDP attribute already defined for use with this (sub)protocol in another IETF specification. Sub-protocol-specific attributes might also be defined for exclusive use with data channel transport, but should use the same syntax described here for other sub-protocol-specific attributes.

Each sub-protocol specific SDP attribute that would normally be used to negotiate the subprotocol using SDP is replaced with an attribute of the form "a=wdcsa:sctp-port:stream-id original-attribute", where wdcsa stands for "webrtc-DataChannel sub-protocol attribute", sctp-port is the sctp port number assigned for webrtc-DataChannel on the media line, stream-id is the sctp stream identifier assigned to this sub-protocol instance, and original-attribute represents the contents of the sub-protocol related attribute to be included.

```
a=webrtc-DataChannel:5000 stream=2;label="channel 2"; \
  subprotocol="MSRP";max_retr=3
a=wdcsa:5000:2 accept-types:text/plain
```

Thus in the example above, the original attribute line "a=accept-types:text/plain" is represented by the attribute line "a=wdcsa:5000:2 accept-types:text/plain", which specifies that this instance of MSRP being transported on the sctp association using port number 5000 and the data channel with stream id 2 accepts plain text files.

As opposed to the data channel setup parameters, these parameters are subject to offer/answer negotiation following the procedures defined in the sub-protocol specific documents.

The same syntax applies to any other SDP attribute required for negotiation of this instance of the sub-protocol.

Note: This document does not provide a complete specification of how to negotiate the use of a data channel to transport MSRP. Procedures specific to each sub-protocol such as MSRP will be documented elsewhere. The use of MSRP is only an example of how the generic procedures described herein might apply to a specific sub-protocol.

5.2. Procedures

5.2.1. Managing stream identifiers

For the SDP-based external negotiation described in this document, the initial offerer (in the context of a PeerConnection) owns by convention the even stream identifiers whereas the initial answerer owns the odd stream identifiers. This ownership is invariant for the whole lifetime of the PeerConnection, e.g. it does not change if the initial answerer sends a new offer to the initial offerer.

[ACTION ITEM] This convention is different from the convention currently defined for in-band negotiation, where even/odd assignment is determined by DTLS role. Since DTLS role cannot be determined until after the initial SDP offer/answer is complete, this convention cannot be used for external negotiation. It might be appropriate to change the convention for stream identifier assignment for in-band negotiation for consistency with external negotiation. Otherwise it might be necessary to prohibit simultaneous use of in-band and external negotiation for data channels.

5.2.2. Opening a data channel

The procedure for opening a data channel using external negotiation starts with the agent preparing to send an SDP offer. If a peer receives an SDP offer before getting to send a new SDP offer with data channels that are to be externally negotiated, or loses an SDP offer glare resolution procedure in this case, it must wait until the ongoing SDP offer/answer completes before resuming the external negotiation procedure.

The agent that intends to send an SDP offer to create data channels through SDP-based external negotiation performs the following:

- Creates data channels using stream identifiers from the owned set (see Section 5.2.1).

- As described in Section 4.2.2, if the SCTP association is not yet established, then the newly created data channels are in the Connecting state, else if the SCTP association is already established, then the newly created data channels are in the Open state.

- Obtains a new SDP offer from the browser.

- Determines the list of stream identifiers assigned to data channels opened through external negotiation.

- Completes the SDP offer with the webrtc-DataChannel and wdcsa attributes needed for each externally-negotiated data channel, as described in Section 5.1.

- Sends the SDP offer.

The peer receiving such an SDP offer performs the following:

- Applies the SDP offer. Note that the browser ignores data channel specific attributes in the SDP.

- Analyzes the channel parameters and sub-protocol attributes to determine whether to accept each offered data channel.

- For accepted data channels, creates peer instances for the data channels with the browser using the channel parameters described in the SDP offer. Note that the browser is asked to create data channels with stream identifiers not "owned" by the agent.

- As described in Section 4.2.2, if the SCTP association is not yet established, then the newly created data channels are in the Connecting state, else if the SCTP association is already established, then the newly created data channels are in the Open state.

- Obtains a new SDP answer from the browser.

- Completes the SDP answer with the webrtc-DataChannel and wdcsa attributes needed for each externally-negotiated data channel, as described in Section 5.1.

- Sends the SDP answer.

The agent receiving such an SDP answer performs the following:

- Closes any created data channels (whether in Connecting or Open state) for which the expected webrtc-DataChannel and wdcsa attributes are not present in the SDP answer.

- Applies the SDP answer.

Any data channels in Connecting state are transitioned to the Open state when the SCTP association is established.

Each agent application must wait to send data until it has confirmation that the data channel at the peer is in the Open state. This occurs:

- At both peers when a data channel is created without an established SCTP association, as soon as the browsers report that the data channel transitions to the Open state from the Connecting state.

- At the agent receiving an SDP offer for which there is an established SCTP association, as soon as it creates an externally negotiated data channel in the Open state based on information signaled in the SDP offer.

- At the agent sending an SDP offer to create a new externally negotiated data channel for which there is an established SCTP association, when it receives the SDP answer confirming acceptance of the data channel or when it begins to receive data on the data channel from the peer, whichever occurs first.

5.2.3. Closing a data channel

When the application requests the closing of a data channel that was externally negotiated, the browser always performs an in-band SSN reset for this channel.

It is specific to the sub-protocol whether this closing must in addition be signaled to the peer via a new SDP offer/answer exchange.

The application must also close any data channel that was externally negotiated, for which the stream identifiers are not listed in an incoming SDP offer.

6. Security Considerations

To be completed.

7. IANA Considerations

To be completed.

8. Acknowledgments

The authors wish to acknowledge the borrowing of ideas from other internet drafts by Salvatore Loreto, Gonzalo Camarillo, Peter Dunkley and Gavin Llewellyn, and to thank Paul Kyzivat, Jonathan Lennox, Uwe Rauschenbach and Keith Drage for their invaluable comments.

9. References

9.1. Normative References

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

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.

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

[I-D.ietf-rtcweb-jsep] Uberti, J. and C. Jennings, "Javascript Session Establishment Protocol", draft-ietf-rtcweb-jsep-04 (work in progress), September 2013.

[I-D.ietf-rtcweb-data-channel] Jesup, R., Loreto, S., and M. Tuexen, "RTCWeb Data Channels", draft-ietf-rtcweb-data-channel-05 (work in progress), July 2013.

[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-04 (work in progress), June 2013.

[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>.

9.2. Informative References

[I-D.ietf-rtcweb-data-protocol] Jesup, R., Loreto, S., and M. Tuexen, "WebRTC Data Channel Protocol", draft-ietf-rtcweb-data-protocol-00 (work in progress), July 2013.

[RFC4975] Campbell, B., Mahy, R., and C. Jennings, "The Message Session Relay Protocol (MSRP)", RFC 4975, September 2007.

[RFC4976] Jennings, C., Mahy, R., and A. Roach, "Relay Extensions for the Message Sessions Relay Protocol (MSRP)", RFC 4976, September 2007.

[RFC5547] Garcia-Martin, M., Isomaki, M., Camarillo, G., Loreto, S., and P. Kyzivat, "A Session Description Protocol (SDP) Offer/Answer Mechanism to Enable File Transfer", RFC 5547, May 2009.

[RFC6135] Holmberg, C. and S. Blau, "An Alternative Connection Model for the Message Session Relay Protocol (MSRP)", RFC 6135, February 2011.

[RFC6714] Holmberg, C., Blau, S., and E. Burger, "Connection Establishment for Media Anchoring (CEMA) for the Message Session Relay Protocol (MSRP)", RFC 6714, August 2012.

Authors’ Addresses

Richard Ejzak
Alcatel-Lucent
1960 Lucent Lane
Naperville, Illinois 60563-1594
US

Phone: +1 630 979 7036
Email: richard.ejzak@alcatel-lucent.com

Jerome Marcon
Alcatel-Lucent
Route de Villejust
Nozay 91620
France

Email: jerome.marcon@alcatel-lucent.com