

RTCWeb Working Group
Internet Draft
Intended status: Informational
Expires: January 30, 2013

G. Mandyam
Qualcomm Innovation Center
Vijay Suryavanshi
Qualcomm
July 30, 2012

RTCWeb Data Stream and RTP Synchronization
draft-mandyam-rtcweb-data-synch-00.txt

Abstract

The RTCWeb working group in the IETF is tasked with developing standards that will ensure interoperability between web browsers establishing rich communications sessions. This working group is tasked with delivering the specifications necessary to establish real-time transport sessions between browsers (e.g. those based on real-time protocol, i.e. RTP). Moreover, the group is also tasked with providing a means for application data streaming between browsers (i.e. opaque data streaming). Much like RTP synchronization sources (SSRC's) can be temporally synchronized, there are use cases that require opaque data stream synchronization with the real-time communications stream between browsers in an RTCWeb session. This document provides some options for temporally associating an opaque data stream with a voice/video stream as part of RTCWeb communications.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79. This document may not be modified, and derivative works of it may not be created, and it may not be published except as an Internet-Draft.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/lid-abstracts.html>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on January 30, 2013.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction.....	2
2. SCTP-Based Data Streaming.....	3
2.1. SCTP Multi-Channel Impacts.....	4
3. Opaque Data Synchronization and In-band RTP Signaling.....	5
4. Security Considerations.....	7
5. IANA Considerations.....	7
6. Conclusions.....	7
7. References.....	7
7.1. Normative References.....	7
7.2. Informative References.....	8
8. Acknowledgments.....	9

1. Introduction

The RTCWeb effort seeks to define the necessary interoperability specifications required for real-time peer-to-peer communications sessions between browsers. These communications sessions normally involve multimedia data transmission (audio, video, or both). However, RTCWeb will also include the ability for web applications to initiate data streaming sessions between browsers.

One of the recommended transports for audio or video in RTCWeb sessions is real-time protocol (RTP) [I-D.-rtcweb-rtp-usage]. An RTCWeb session can include one or more RTP streams, each stream identified by an SSRC (synchronization source) included in the RTP frame header.

There has been some concern about whether existing mechanisms in RTP standards allow an RTP session endpoint to be able to render multiple SSRC's in a time-synchronized manner [I-D.-draft-alvestrand-rtcweb-msid]. As a result, several mechanisms have been proposed that would allow an RTCWeb endpoint to definitively determine which SSRC's are temporally synchronized and must be rendered as such.

Assuming the problem of associating temporally-synchronized SSRC's will be solved by one of the proposed mechanisms, there still can be cases where an SSRC may have a temporal relationship with application-generated data that would also be streamed as part of the RTCWeb session. An example is video overlay based on web touch events during a video telephony session. In this case, a web application detects an animation over the video preview window (based on the end user drawing an image using the device touch surface), and is required to send such information to the RTCWeb endpoint so that the animation can be rendered.

This document discusses three approaches to synchronization of data streams, along with associated recommendations.

2. SCTP-Based Data Streaming

[I-D.-jesup-rtcweb-data-protocol] describes an approach that could be adopted in RTCWeb for data streaming, leveraging the Stream Control Transmission Protocol (SCTP), and [I-D.ietf-mmusic-sctp-sdp] provides the necessary extensions to Session Description Protocol (SDP) to describe an SCTP stream. SDP is the mechanism by which multimedia sessions are described in RTCWeb, usually as part of the invite or call announce.

The m-line in the SDP message (as per [I-D.ietf-mmusic-sctp-sdp]) should include sufficient information to describe the SCTP session (e.g. plain SCTP, SCTP over DTLS, etc.). For example, an SDP message from an offerer at address xxx.xx.xx.xx using port yyyyy for SCTP communication, then a possible SDP offer would include

```
m=application yyyyy SCTP *
```

```
c=IN IP4 xxx.xx.xx.xx
```

If there is an additional RTP-based media source sent by the offerer that needs synchronization with the SCTP stream, the ideal case would be to leverage existing SDP grouping mechanisms. The mid attribute of RFC 5888 could potentially be leveraged:

```
c=IN IP4 xxx.xx.xx.xx
a=group:LS 1 2
m=application yyyyy SCTP *
a=mid:1
m=video zzzzz RTP/AVP
a=mid:2
```

There are some issues with this approach, as highlighted in [I-D.-draft-alvestrand-rtcweb-msid] (e.g. multiple SSRC's in each RTP stream). Nevertheless, SDP grouping can provide a sufficient solution to synchronizing the SCTP stream to an RTP stream as long as there is one SSRC per RTP stream. SDP grouping should also be applicable in the case where multiple SSRC's are part of the offer and are associated with a canonical name (CNAME), using the attribute guidelines of RFC 5576 (e.g. "a=ssrc:<ssrc-id>cname:<cname>" along with "a=mid:...").

2.1. SCTP Multi-Channel Impacts

[I-D.-jesup-rtcweb-data-protocol] provides an SCTP-encapsulated control protocol for the RTCWeb data channel that takes advantage of the multistreaming capabilities of SCTP. SCTP allows for individual stream identifiers and associated sequence numbers for any given data chunk. This allows for flow control on individual streams within an SCTP session. Streams are also further identified by a label attribute as defined in [I-D.-jesup-rtcweb-data-protocol] as part of the logical channel request. Since the streams are dynamic, to associate an SCTP stream at any given instant in time with an RTP session is not straightforward. In addition, SCTP can be multihomed, i.e. endpoints can be associated with more than one IP address. Some of the current unresolved issues are:

- a. Should the SDP attribute describing the data channel stream be based on logical channel label or SCTP stream ID?
- b. What is the required receiver behavior if the data channel stream identifier provided in the SDP offer does not match with information sent in-band? Note that a comparable issue also exists for RTP streams using CNAME and SSRC.

In order to address these issues in a simpler manner, the following guideline is proposed for RTCWeb: the SDP grouping mechanism should not address individual streams within an SCTP session. In other words, once a temporal relationship is established between an RTP stream and an SCTP session, that relationship will apply to all streams in the SCTP session.

3. Opaque Data Synchronization and In-band RTP Signaling

Web application generated data may have a temporal relationship with an RTP-based media stream, but if is relatively infrequent and therefore requires much less throughput than the media stream itself it could make more sense to multiplex the application-specific data into the RTP stream. Sec. 5.3.1 of RFC 3550 describes the RFC Header Extension mechanism, by which an application-specific payload can be inserted into an existing RTP stream without affecting the media flow.

In the RTP header, and extension bit X can be set. This indicates the existence of an extension header.

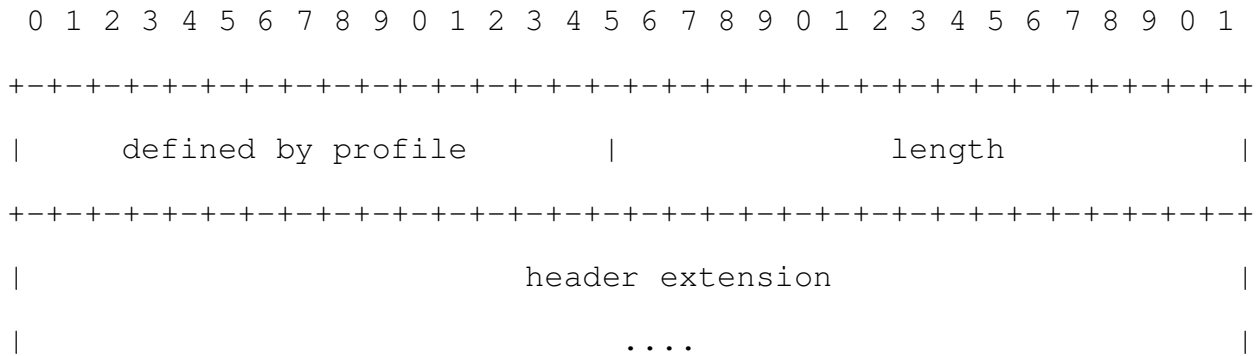


Figure 1 : RTP Extension Header

Referring to Figure 1, the value of 16-bit profile field in the extension header is implementation specific. This field could be used in place of the channel label in the SCTP-based data channel. Otherwise, this field can be ignored by the receiver.

The signaling of the use of an extension header as the means of opaque data transfer could be agreed upon by the two RTCWeb endpoints by means of an offer/answer protocol like SDP. The out-of-band signaling channel can be used to indicate to the receiver to create a data channel based on the RTP extension header. RFC 5576 can also be leveraged in this case using a new source-specific attribute 'data':

```
a=ssrc:<ssrc-id> data
```

The SDP exchange is not strictly required, however. This is because the SSRC of the RTP stream has already been negotiated, and the extension header is in fact really part of the RTP media stream data.

Ideally, a message-based Data Channel API from the WebRTC specification (see [W3C.WD-webrtc-20120530]) would be leveraged by the web application in such a way that the underlying user agent would multiplex application data onto an existing RTP stream using the RTP extension header. Borrowing from the JSEP messaging flow [I-D.-rtcweb-jsep], the PeerConnection setup will proceed as normal from the offerer perspective:

```
OffererJS->OffererUA: var pc = new PeerConnection(config, null);
OffererJS->OffererUA: pc.onIceCandidate = onIceCandidate;
OffererJS->OffererUA: pc.addStream(stream);
OffererJS->OffererUA: var offer = pc.createOffer(null);
OffererJS->OffererUA: pc.setLocalDescription("offer", offer);
```

... Answerer creates PeerConnection and sends answer

```
AnswererUA->OffererUA: <media>
// Send opaque data from Offerer to Answerer
```

```
OffererJS->OffererUA: var chan = pc.createDataChannel(10);
// Numeric label means opaque data to be sent with extension
header
```

```
OffererJS->OffererUA: chan.send("Some Payload");
AnswererUA->OffererUA: <media> with extension header
```

```
AnswererUA->AnswererJS: pc.ondatachannel = function({...});
// Answerer creates DataChannel listener on existing
PeerConnection based upon firing of onDataChannel event
```

```
OffererUA->OffererJS: datachannellistener.onmessage({});
```

Note that in the approach above, the creation of a data channel with a numeric label is what triggers the OffererUA to use the extension header. The numeric label can be directly sent as part of the profile field in the extension header (provided that the numeric label does not exceed 16 bits) The initial receipt of RTP data with an extension header triggers the onDataChannel event to fire from the AnswererUA.

3.1. Other Uses of the RTP Extension Header

Section 5.2 of [I-D.-rtcweb-rtp-usage] clearly discusses (but does not call for requiring) additional uses of the RTP extension header. These uses include a rapid synchronization feature (which allows timing metadata to be inserted into the RTP stream), client-to-mixer audio level, and mixer-to-client audio level. The profile space that may be consumed by these uses of the header extension can be avoided for RTCWeb logical data channels that also use the header extension.

4. Security Considerations

TBD.

5. IANA Considerations

TBD.

6. Conclusions

The ability to send and receive opaque data streams that are synchronized to existing RTP media sessions will greatly enhance RTCWeb. It will open up a several new possibilities for user interactions around telephony sessions (video or voice). The existing specifications in both the W3C and IETF do not address how such a feature would be implemented. This document provided two methods for achieving this feature that leveraged as much as possible the specifications currently under consideration in RTCWeb and the W3C.

7. References

7.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2234] Crocker, D. and Overell, P.(Editors), "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, Internet Mail Consortium and Demon Internet Ltd., November 1997.
- [RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.

- [RFC5576] Lennox, J., Ott, J., and T. Schierl, "Source-Specific Media Attributes in the Session Description Protocol (SDP)", RFC 5576, June 2009.
- [RFC5888] Camarillo, G. and H. Schulzrinne, "The Session Description Protocol (SDP) Grouping Framework", RFC 5888, June 2010.
- [RFC6222] Begen, A., Perkins, C. and D. Wing, "Guidelines for Choosing RTP Control Protocol (RTCP) Canonical Names", RFC 6222, April 2011.
- [RFC5576] Lennox, J., Ott, J., and T. Schierl, "Source-Specific Media Attributes in the Session Description Protocol (SDP)", RFC 5576, June 2009.
- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real Time Communications", RFC 3550, July 2003.
- [I-D.-rtcweb-rtp-usage] Perkins, C., Westerlund, M. and J. Ott, "Web Real-Time Communication (WebRTC): Media Transport and Use of RTP", draft-ietf-rtcweb-rtp-usage-03, June 2012.
- [W3C.WD-webrtc-20120530] Bergkvist, A., Burnett, D., Narayanan, A., and C. Jennings, "WebRTC 1.0: Real-time Communication Between Browsers", World Wide Web Consortium WD WD-webrtc-20120209, Editor's Draft, 30 May 2012.
- [I-D.-rtcweb-jsep] Uberti, J. and C. Jennings, "Javascript Session Establishment Protocol", draft-ietf-rtcweb-jsep-01, June 2012.

7.2. Informative References

- [I-D.-jesup-rtcweb-data-protocol] Jesup, R., Loreto, S. and M. Tuexen, "WebRTC Data Channel Protocol", draft-jesup-rtcweb-data-protocol-01, June 2012.
- [I-D.-draft-alvestrand-rtcweb-msid] Alverstand, H., "Cross Session Stream Identification in the Session Description Protocol", draft-alvestrand-rtcweb-msid-02, May 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.

8. Acknowledgments

This document was prepared using 2-Word-v2.0.template.dot.

Authors' Addresses

Giridhar Mandyam
Qualcomm Innovation Center
5775 Morehouse Drive
San Diego, CA 92121
USA

Email: mandyam@quicinc.com

Vijay Suryavanshi
Qualcomm Inc.
5775 Morehouse Drive
San Diego, CA 92121
USA

Email: vsuryava@qualcomm.com