

SIPREC
Internet-Draft
Intended status: Informational
Expires: February 28, 2015

M. Yan
P. Kyzivat
Huawei
August 27, 2014

Overview for MSRP Recording based on SIPREC
draft-yan-siprec-msrp-recording-02

Abstract

SIPREC is capable of recording interactive text media that is transmitted via RTP. However that format is not commonly used for message or chat scenarios. There is also a need for recording text media carried via MSRP. One case of note is exchange of text between hearing-impaired users and emergence service bureaus. Also, recording support is needed for MSRP used in chat conferences and multimedia conferences.

This document describes how to achieve MSRP channel recording within the mechanism of SIP Recording (SIPREC).

Requirements Language

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 RFC 2119 [RFC2119].

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on February 28, 2015.

Copyright Notice

Copyright (c) 2014 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	3
1.1. EDITOR NOTES	3
2. Definitions	6
3. MSRP Recording Architecture	6
3.1. MSRP Client acts as SRC	6
3.2. MSRP Relay acts as SRC	7
3.3. MSRP Switch acts as SRC	7
4. MSRP Media Stream Mixing	8
5. MSRP Session Usage by the SRC	9
6. MSRP Session Usage by the SRS	9
7. File Transfer	9
8. Recording Chatrooms	10
9. Metadata	10
10. MIME Type for MSRP Recording	11
10.1. CPIM Extension Header - rs.Content	11
10.2. CPIM Extension Header - rs.Stream-ID	11
10.3. CPIM Extension Header - rs.Message-ID	12
10.4. CPIM Extension Header - rs.Nickname	12
10.5. CPIM Extension Header - rs.Unsupported-Type	12
10.6. CPIM Extension Header - rs.Size	12
11. Representation of CS MSRP Messages in the RS	12
11.1. Recording CS SEND Messages	13
11.2. Dropping CS SEND Messages	13
11.3. Recording NICKNAME Messages	14
11.4. Recording CS REPORT Messages	15
11.5. Recording CS Transaction Responses	15
12. Open Issues	15
13. IANA Considerations	15
14. Security Considerations	15
15. References	15
15.1. Normative References	15

15.2. Informative References	16
Authors' Addresses	17

1. Introduction

SIPREC is capable of recording interactive text media that is transmitted via RTP, as defined by [RFC4103]. However that format is not commonly used for message or chat scenarios. There is also a need for recording text media carried via MSRP. One case of note is exchange of text between hearing-impaired users and emergence service bureaus. Also, recording support is needed for MSRP used in chat conferences (as defined by [I-D.ietf-simple-chat]) and multimedia conferences (as defined by [RFC4597]).

Instant message media is carried by a variety of protocols such as IRC, MSRP and XMPP/JINGLE. The SIP based MSRP protocol (as defined by [RFC4975] and [RFC4976]) supports the delivery of messages and files from one SIP UA to another. When a SIPREC SRC is recording a CS that contains an MSRP channel, it may want to record the messages passing over that channel. To gain access to the messages, the SRC may act as an MSRP client, relay, or switch. The SRC needs to replicate and deliver the messages over an MSRP channel within a Recording Session (RS) to an SRS. The replicated content could be in Message/CPIM format containing plain text, HTML, images, etc. In this document, file delivering sessions have not yet been considered. Other instant message protocols, like IRC or XMPP, are out of scope.

This document describes how MRSP sessions are established between an SRC and SRS, and used for conveying the replicated MSRP Media, and also specifies metadata that describes the recorded MSRP sessions. A Recording Session employing MSRP is established using the normal procedures for establishing INVITE initiated dialogs [RFC3261] and uses SDP [RFC4566] for describing the media to be used during the session as described by the SIPREC Architecture [RFC7245].

1.1. EDITOR NOTES

This version addresses comments received on the -01 version, both on the mailing list and at IETF90. The following is my list of things to address:

- o Define a new MIME type that is used to wrap the CS MSRP messages that are being recorded. This allows the original message to be left as-is, so it is always clear what it was. While CPIM could be used for this, defining a new type will allow capturing other necessary metadata.

- o Need to further consider the need to track message timing. Can the timing of messages received by the SRS on the RS MSRP stream be considered a sufficient proxy for the timing of messages in the CS, or should we explicitly pass timestamps of messages as received on the CS? (The issue was raised but not decided.)
- o We need to clarify that there is no guarantee that messages received on the CS have been recorded.
- o It was agreed that there is no need to record the MSRP URIs that are used to establish the CS MSRP session.
- o It is important that we maintain a 1:1 consistency between MSRP MESSAGE-IDs used in recorded CS sessions and the MESSAGE-IDs used in the RS. But we should not violate MSRP by using the same MESSAGE-IDs. We came up with the idea of adding an SRC-specific prefix to the CS message ids to create unique ones for the RS. This should be done in a standard way so that the SRS can recover the original CS message ids, in order to support correlation across redundant SRCs.
- o Will need to work out the details of what happens when a CS MSRP session is terminated with an incomplete message. It will be necessary to send the incomplete message to the SRS, but must it appear to be incomplete within the SRS MSRP session?
- o There are a variety of reasons why the SRC may not want, or be able to, record individual messages in the CS session. (One example is because the message size is greater than the maximum indicated by the SRS. Another is because the mime type of the message is a type that the SRS did not indicate support for.) There should be a type of placeholder message that can be sent to the SRS to indicate a message has been dropped, why, and some key attributes about the message. The new SIPREC wrapper mime type could be designed to serve this purpose.
- o REPORT messages on the CS can't be sent directly on the RS. The new SIPREC wrapper mime type could also serve as a way to encapsulate those.

The primary change is to introduce a new wrapper MIME type ("application/msrp-recording") that is used in RS MSRP sessions for all CS MSRP messages that are to be recorded. This is used with SEND messages whether they have a CPIM wrapper or not. It also allows non-SEND messages from the CS to be sent intact in the RS for recording. And it provides a vehicle for carrying other data as needed.

Adding another layer of wrapper could substantially increase the total amount of data sent on the RS session, relative to what is present on the CS. I've tried to mitigate that via the details of the design. For SEND messages, only the body of the SEND message is wrapped. And From and To headers in this wrapper can be omitted in cases where that information is redundant. I've assumed that messages other than SEND should in general be infrequent enough that extra overhead when sending them isn't worth a lot of concern.

This wrapper can carry a DateTime header. This provides a mechanism to address the timestamp issues. I've left it as optional to use.

I clarified the non-guarantee of recording in the architecture section.

I've provided a special header in the wrapper to carry the MESSAGE-ID from SEND messages in the CS. And SEND messages will get a separate MESSAGE-ID on the RS MSRP session when sent to the SRS. This provides the SRS with enough information to solve the correlation problem when a message is incomplete in one CS MSRP session and is resumed on another. (The problem of reassembly is left to the SRS.)

The wrapper format includes a mechanism for the SRC to report dropped messages to the SRS.

The wrapper format also includes a mechanism for encapsulating CS REPORT messages for sending to the SRS.

I realized that this level of wrapping provides an opportunity to multiplex unrelated CS MSRP sessions on a single RS MSRP session. To allow this I've provided a way to include the session-id from the metadata, that identifies the particular CS MSRP session, as a value in the wrapper of the message sent on the RS. But I also made that optional when it is redundant. This gives a choice: multiplex but make the messages bigger, or create a separate RS MSRP session for each CS MSRP session and keep the messages smaller. I've included this as a trial balloon for discussion. I'm undecided about it.

The formatting of all of this could be better. But for now I just wanted to get the basic concepts down for review. Once the approach is reasonably well worked out I'll try to improve the formatting.

There are many places here where I am uncertain what normative strength to apply to individual requirements. I've indicated this inline for many of those. Please comment on this.

2. Definitions

(TBD...)

3. MSRP Recording Architecture

For consistency with the SIPREC Architecture [RFC7245] and the SIPREC Protocol [I-D.ietf-siprec-protocol] MSRP recording needs to deliver duplicated MSRP message content from the SRC to the SRS, with suitable descriptive metadata. The SRC may be associated with SIP UA (endpoint) with an MSRP client, or with a SIP B2BUA that accesses the media via an MRSP Relay. An SRC may also be associated with a SIP conference focus and an MSRP switch.

Note: The decision to record or not is a policy decision on the part of both the SRC and the SRS. Support for this specification provides no guarantee that any particular MSRP session, or message within a session, will be recorded. However MSRP recording is subject to the notification requirements called out in Section 6.1.2 of [I-D.ietf-siprec-protocol].

3.1. MSRP Client acts as SRC

[RFC4975] and [RFC4976] describe how an MSRP client communicates to another MSRP client via a SIP session. A MSRP client that has access to the MSRP content to be recorded may act as SRC. The MSRP client may send the replicated media to the SRS along with corresponding metadata.

If the MSRP client/SRC is aware the MSRP session needs to be recorded, it can initiate the establishment of a SIP RS by sending an INVITE to SRS, or vice-versa. The MSRP client/SRC is responsible for notifying the other MSRP client involved in the CS that the MSRP session is being recorded. The MSRP client/SRC is responsible for complying with request from recording aware UAs or through some configured policies indicating that the CS should not be recorded.

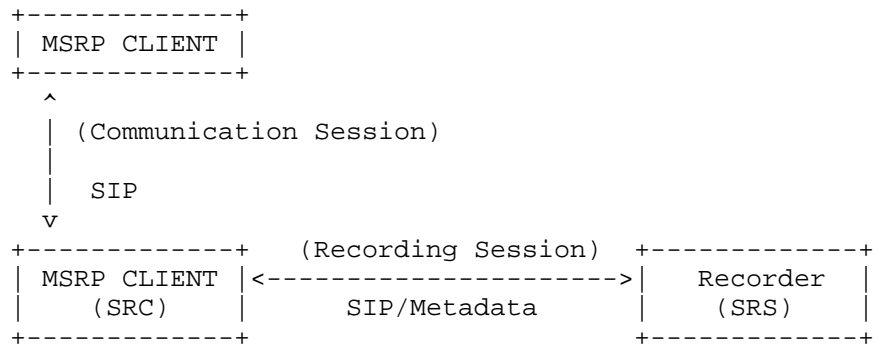


Figure 1: MSRP Client Acts as SRC

3.2. MSRP Relay acts as SRC

(TBD... RFC4976)

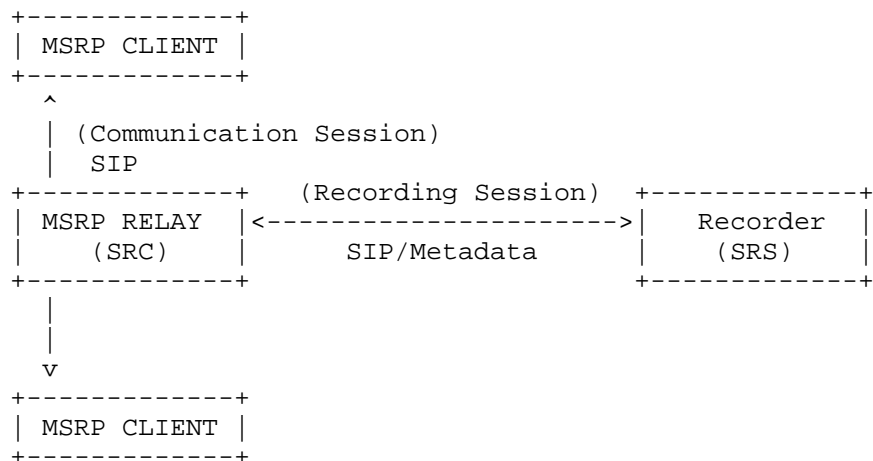


Figure 2: MSRP Relay Acts as SRC

3.3. MSRP Switch acts as SRC

(TBD... ietf-simple-chat)

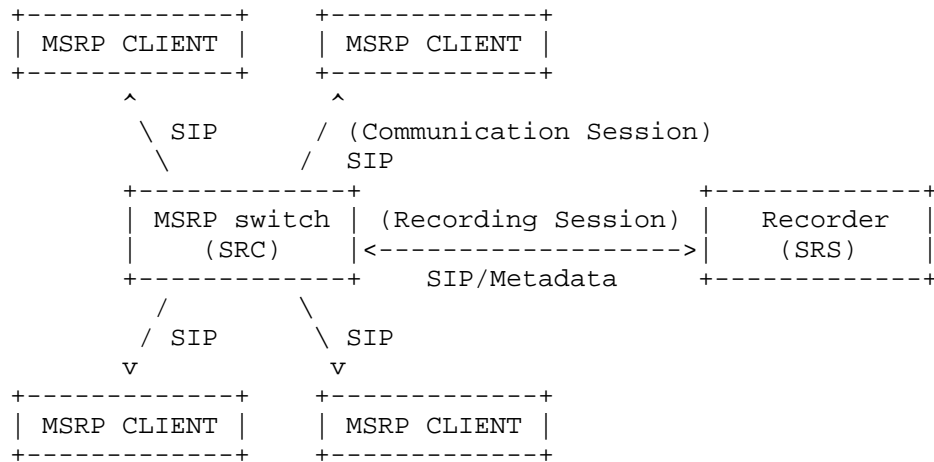


Figure 3: MSRP Switch Acts as SRC

4. MSRP Media Stream Mixing

[TODO: Revise this to cover multiplexing of unrelated media streams.]

Note: SIPREC metadata allows both the inclusion of multiple participants within a single <stream> element, and the mapping of multiple <stream> elements to a single MSRP m-line in the RS. These provide two ways to do very similar things.

Mapping multiple participants to a single <stream> is natural for a conference. It works well for MSRP chat sessions

By providing a way to specify the stream-id with an individual message on the RS, I've introduced a way to demux messages from multiple <stream>s that are mapped to the same MSRP m-line. This provides a way reduce the number of MSRP sessions in the RS. It also avoids confusion when an RS MSRP session is serially reused for distinct CS MSRP sessions.

I'm still considering whether it is good to have both of these mechanisms, or if one of them should be removed. Until I make a decision I haven't updated all the text that pertains to this.

Feedback on this will be appreciated.

As with RTP-based media, CS MSRP media streams from different participants may be mixed into a single RS media stream, or they may be conveyed as separate MSRP streams. In RTP, when media from

different participants is mixed, it is distinguished by CNAME and SSRC or CSRC. In MSRP, media from different participants is distinguished by wrapping the the message in a CPIM body, with the sender identified by the From header in the CPIM. If the SRC mixes MSRP media from multiple senders, then each message that isn't already in CPIM format SHOULD be embedded in a CPIM message, and the From and To headers of that CPIM wrapper SHOULD identify the sending and receiving participants for that message.

5. MSRP Session Usage by the SRC

[TODO: Revise this to cover multiplexing of unrelated media streams.]

When preparing to record a CS MSRP media stream, the SRC MUST choose a corresponding RS MSRP session. CS MSRP sessions that are being mixed share an RS MSRP session, while those that are not being mixed are assigned to unique RS MSRP sessions.

The RS MSRP session MAY be newly created, or a pre-existing RS MSRP session that is no longer in use MAY be repurposed. When an MSRP session is repurposed, the SRC communicates this change to the SRS via a change in the metadata. The SRC is responsible for ensuring that messages for the new session are not sent until the SRS has received the metadata describing this new session.

MSRP message flow on a RS MSRP session is always from the SRC to the SRS. The SRC generates SEND messages, and may receive REPORT messages. It does not receive SEND messages or send REPORT messages.

6. MSRP Session Usage by the SRS

[TODO: Revise this to cover multiplexing of unrelated media streams.]

The SRS MUST be able handle a case where an RS MSRP session is first used to record one CS MSRP session and then is repurposed to record a different CS MSRP session. The SRS learns of this change via a change in the metadata.

MSRP message flow on a RS MSRP session is always from the SRC to the SRS. The SRS receives SEND messages, and sends REPORT messages. It does not generate SEND messages or receive REPORT messages.

7. File Transfer

A mechanism for doing file transfer via MSRP is specified in [RFC5547]. If this mechanism is used in the CS, then the SRC MAY use it in the RS to record those files. In turn, the SRS MAY choose to

accept some or all of those file transfer requests, or MAY reject them.

Both file push and file pull operations are defined. If the SRC chooses to record a file transfer, whether it is initiated in the CS via a push operation or a pull operation, within the RS the SRC MUST initiate the transfer with a push operation in an SDP Offer.

(SRS initiation of a file transfer is out of scope of this document.)

It is possible that the SRC may support file transfer while the SRS does not. If the SRC sends an SDP offer to the SRS containing an m-line initiating a file transfer, and the SRS sends an answer accepting the MSRP session, but fails to include a matching file-transfer-id, then the SRC MUST NOT send the content of CS MSRP file transfer session to the SRS.

8. Recording Chatrooms

An CS MSRP session might involve a chatroom. The SRC discovers this by observing use of the features defined in [I-D.ietf-simple-chat]

When the CS MSRP session involves a chatroom, the SRC SHOULD [MUST?] indicate this in the corresponding RS MSRP session. The key unique features of chatrooms are nicknames and private messages. If either of these features is indicated in an SDP 'chatroom' attribute in the CS, then this MAY also be indicated in the RS SDP.

Requests for nickanmes in the CS via the NICKNAME message are reported to the SRS using the mechanism described in Section 11.3.

When messages are sent by sources that have had a nickname assigned, the nickname is conveyed to the SRS using the mechanism described in Section 10.4.

Private messages used in a chatroom are identified in the CS via a CPIM wrapper with a To header that identifies the intended recipient(s) rather than the URI of the chatroom itself. This information is retained when the message is forwarded to the RS, while the chatroom URI is also conveyed using the "To" header of the "application/msrp-recording" wrapper, as described in section Section 11.

9. Metadata

The metadata defined in [I-D.ietf-siprec-metadata] can be used without change to describe MSRP streams.

10. MIME Type for MSRP Recording

The document defines a new MIME type "application/msrp-recording" as an extension to type "application/cpim". This type includes new headers for carrying details about the wrapped message. The new headers are all identified by a namespace prefix of "rs.".

[Note: I found the details of how to make an application-specific extension to CPIM to be vague in RFC3862. I'm uncertain if extension headers must be referenced with a prefix, but that is my best guess. The details need more research.]

10.1. CPIM Extension Header - rs.Content

The value of the "rs.Content" header is a token identifying the sort of content contained in the body of this message. The following types of content are defined:

- o send
- o drop
- o msrp

At most one "rs.Content" header may be present in a message. If no "rs.Content" header is present, then "rs.Content: send" is implied.

The 'send' token indicates that the content of the message contains all or a fragment of the body of an MSRP SEND message.

The 'drop' token indicates that the content of a SEND message in the CS is not being sent to the RS for recording.

The 'msrp' token indicates that the body of the message contains a complete MSRP message from the CS. This form MAY be used to convey REPORT messages, NICKNAME messages, and transaction responses.

10.2. CPIM Extension Header - rs.Stream-ID

The value of the "rs.Stream-ID" header is the stream-id used in the SIPREC metadata to identify the stream that this message belongs to. This header MAY be omitted if the SIPREC metadata associates exactly one stream with this MSRP session. If present, the value MUST match the stream-id of exactly one of the streams associated with this MSRP session.

10.3. CPIM Extension Header - rs.Message-ID

The value of the "rs.Message-ID" header carries the value of the "Message-ID" from the MSRP SEND message. At most one "rs.Message-ID" header may be present in a message. It MUST be present when the "rs.Content" value is 'send' or 'drop', and MUST NOT be present in other cases.

10.4. CPIM Extension Header - rs.Nickname

The value of the "rs.Nickname" header carries the nickname of the sender of the MSRP SEND message. At most one "rs.Nickname" header may be present in a message. It MAY be present when the "rs.Content" value is 'send' or 'drop', and MUST NOT be present in other cases.

10.5. CPIM Extension Header - rs.Unsupported-Type

The value of the "rs.Unsupported-Type" header carries a content-type from a CS MSRP SEND message that is not supported by the SRS. It may be the outermost type, or the type of a component of a container type. Any number of "rs.Unsupported-Type" headers may be present in a message. It MAY be present when the "rs.Content" value is 'drop', and MUST NOT be present in other cases.

10.6. CPIM Extension Header - rs.Size

The value of the "rs.Size" header carries the integer size of an CS MSRP SEND message. At most one "rs.Size" header may be present in a message. It MAY be present when the "rs.Content" value is 'drop', and MUST NOT be present in other cases.

11. Representation of CS MSRP Messages in the RS

When CS MSRP messages are being recorded, the SRC encapsulates them in the wrapper type "application/msrp-recording". This wrapper type is used to encapsulate the basic MSRP SEND message content, and also to send CS MSRP control messages that should be recorded. It also provides the means for conveying per-message metadata.

The CPIM From and To headers of the wrapper are optional. They MUST be supplied when the proper value cannot be determined by other means:

- o The From header may be omitted if the metadata for the stream indicates that there is only one possible sender, or if the message being encapsulated contains a CPIM From header with the proper value.

- o The To header may be omitted if the metadata for the stream indicates that there is only one possible receiver, or if the message being encapsulated contains a CPIM To header with the proper value.

The CPIM DateTime header MAY be included. If included, it SHOULD indicate the time that the corresponding CS message was sent or received by the SRC.

11.1. Recording CS SEND Messages

When the SRC wishes to record a SEND message from the CS it rewraps the message, taking body from the CS SEND message, placing that into the body of a new "application/msrp-recording" message, and then sending that with a SEND message in the corresponding RS MSRP session.

The SRC MAY retain the fragmentation present in the CS, mapping one CS SEND message to one RS SEND message. Or it MAY merge CS message fragments and/or re-fragment CS SEND message fragments. If a received fragment ends with a continuation-flag of "#", then last fragment sent on the RS MUST also end with a continuation-flag of "#".

Each SEND message fragment MAY, but need not, contain a "rs.Content: send" header.

Each SEND message fragment MUST contain an "rs.Message-ID" header identifying the Message-ID from the corresponding CS MSRP SEND message. (The resulting RS MSRP SEND message will also contain a Message-ID in the RS. This is a distinct value.)

If the SRC knows that the sender of the message on the CS has an associated Nickname [I-D.ietf-simple-chat], then the SRC SHOULD insert an "rs.Nickname" header containing the nickname.

11.2. Dropping CS SEND Messages

[QUESTION: Do we need a way for the SRS to indicate a desire (or not) to receive indications of dropped messages?]

The SRC might decide not to record selected SEND messages from the CS MSRP session. When doing so it MAY send a 'drop' message as an indicator that a message has been dropped. The following considerations apply when deciding whether to send a 'drop' message:

- o While the SRC is honoring a request within the CS to disable recording, it SHOULD [MUST?] NOT send 'drop' messages for CS SEND messages.
- o If the total size from the Byte-Range of the initial fragment of a SEND message in the CS is acceptable for the CS, but exceeds the max-size for the RS session, then the SRC SHOULD send a 'drop' message, and SHOULD include an "rs.Size" header indicating the total size of the message.
- o If a SEND message in the CS contains a continuation fragment, with a Byte-Range indicating that the total message will exceed the max-size for the RS session, then the SRC SHOULD send a 'drop' message, and SHOULD include an "rs.Size" header indicating the total size of the message.
- o If a SEND message has a content type accepted by the 'accept-types' and 'accept-wrapped-types' attributes of the CS but is not accepted by the 'accept-types' or 'accept-wrapped-types' attributes of the RS, then the SRC SHOULD send a 'drop' message. The 'drop' message SHOULD contain an "rs.Unsupported-Type" header identifying the type that is not supported. (When a multipart body is present, the SRC MAY include multiple "rs.Unsupported-Type" headers identifying multiple types.) The SRC MAY choose to send a limited number of 'drop' messages for particular stream - either in total or per unacceptable type.

When a 'drop' message is sent:

- o it MUST be terminated with a continuation-flag of "#";
- o additional fragments with the same CS Message-ID MUST NOT be sent on the RS.

11.3. Recording NICKNAME Messages

The SRC SHOULD forward NICKNAME messages in the CS to the SRS.

[QUESTION: Do we need a way for the SRS to indicate a desire (or not) to receive CS Transaction Error messages?]

To forward a NICKNAME message from the CS to the RS, the SRC places the entire NICKNAME message into the body of a new "application/msrp-recording" message, and then sends that with a SEND message in the corresponding RS MSRP session.

11.4. Recording CS REPORT Messages

The SRC SHOULD [MUST?] forward CS Failure Report messages on the RS.

The SRC MAY [SHOULD?] forward CS Success Report messages on the RS.

[QUESTION: Do we need a way for the SRS to indicate a desire (or not) to receive CS REPORT messages?]

To forward a REPORT message from the CS to the RS, the SRC places the entire REPORT message into the body of of a new "application/msrp-recording" message, and then sends that with a SEND message in the corresponding RS MSRP session.

11.5. Recording CS Transaction Responses

The SRC SHOULD [MUST?] forward CS transaction responses indicating errors to the SRS.

The SRC MAY, but SHOULD NOT forward CS transaction responses indicating success to the SRS. An exception is success responses to NICKNAME messages, which MAY [SHOULD?] be passed to the SRS.

[QUESTION: Do we need a way for the SRS to indicate a desire (or not) to receive CS transaction response messages?]

To forward a transaction response from the CS to the RS, the SRC places the entire transaction response message into the body of of a new "application/msrp-recording" message, and then sends that with a SEND message in the corresponding RS MSRP session.

12. Open Issues

13. IANA Considerations

[TODO: Register application/msrp-recording.]

14. Security Considerations

Not explicitly covered in this version.

15. References

15.1. Normative References

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

- [RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.
- [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.
- [RFC7245] Hutton, A., Portman, L., Jain, R., and K. Rehor, "An Architecture for Media Recording Using the Session Initiation Protocol", RFC 7245, May 2014.
- [I-D.ietf-siprec-metadata]
R, R., Ravindran, P., and P. Kyzivat, "Session Initiation Protocol (SIP) Recording Metadata", draft-ietf-siprec-metadata-16 (work in progress), August 2014.
- [I-D.ietf-siprec-protocol]
Portman, L., Lum, H., Eckel, C., Johnston, A., and A. Hutton, "Session Recording Protocol", draft-ietf-siprec-protocol-14 (work in progress), August 2014.
- [I-D.ietf-simple-chat]
Niemi, A., Garcia, M., and G. Sandbakken, "Multi-party Chat Using the Message Session Relay Protocol (MSRP)", draft-ietf-simple-chat-18 (work in progress), January 2013.

15.2. Informative References

- [RFC4103] Hellstrom, G. and P. Jones, "RTP Payload for Text Conversation", RFC 4103, June 2005.
- [RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.
- [RFC4597] Even, R. and N. Ismail, "Conferencing Scenarios", RFC 4597, August 2006.

Authors' Addresses

Michael Yan
Huawei

Email: michael.yan@huawei.com

Paul H. Kyzivat
Huawei

Email: pkyzivat@alum.mit.edu