Workgroup: Internet Engineering Task Force
Internet-Draft:
draft-hellstrom-avtcore-multi-party-rtt-
solutions-03
Published: 8 August 2020
Intended Status: Informational
Expires: 9 February 2021
Authors: G. Hellstrom
Gunnar Hellstrom Accessible Communication

Real-time text solutions for multi-party sessions

Abstract

This document specifies methods for Real-Time Text (RTT) media handling in multi-party calls. The main discussed transport is to carry Real-Time text by the RTP protocol in a time-sampled mode according to RFC 4103. The mechanisms enable the receiving application to present the received real-time text media, separated per source, in different ways according to user preferences. Some presentation related features are also described explaining suitable variations of transmission and presentation of text.

Call control features are described for the SIP environment. A number of alternative methods for providing the multi-party negotiation, transmission and presentation are discussed and a recommendation for the main ones is provided. The main solution for SIP based centralized multi-party handling of real-time text is achieved through a media control unit coordinating multiple RTP text streams into one RTP stream.

Alternative methods using a single RTP stream and source identification inline in the text stream are also described, one of them being provided as a lower functionality fallback method for endpoints with no multi-party awareness for RTT.

Bridging methods where the text stream is carried without the contents being dealt with in detail by the bridge are also discussed.

Brief information is also provided for multi-party RTT in the WebRTC environment.

The intention is to provide background for decisions, specification and implementation of selected methods.

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 https://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 9 February 2021.

Copyright Notice

Table of Contents

1. Introduction

Real-time text (RTT) is a medium in real-time conversational sessions. Text entered by participants in a session is transmitted in a time-sampled fashion, so that no specific user action is needed to cause transmission. This gives a direct flow of text in the rate it is created, that is suitable in a real-time conversational setting. The real-time text medium can be combined with other media in multimedia sessions.

Media from a number of multimedia session participants can be combined in a multi-party session. The present document specifies how the real-time text streams can be handled in multi-party sessions. Recommendations are provided for preferred methods.

The description is mainly focused on the transport level, but also describes a few session and presentation level aspects.

Transport of real-time text is specified in RFC 4103 [RFC4103] RTP Payload for text conversation. It makes use of RFC 3550 [RFC3550] Real Time Protocol, for transport. Robustness against network transmission problems is normally achieved through redundant transmission based on the principle from RFC 2198 [RFC2198], with one primary and two redundant transmission of each text element. Primary and redundant transmissions are combined in packets and described by a redundancy header. This transport is usually used in the SIP Session Initiation Protocol RFC 3261 [RFC3261] environment.

A very brief overview of functions for real-time text handling in multi-party sessions is described in RFC 4597 [RFC4597] Conferencing Scenarios, sections 4.8 and 4.10. The present specification builds on that description and indicates which protocol mechanisms should be used to implement multi-party handling of real-time text.

Real-time text can also be transported in the WebRTC environment, by using WebRTC data channels according to [I-D.ietf-mmusic-t140-usage-data-channel]. Multi-party aspects for WebRTC solutions are briefly covered.

1.1. 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].

2. Centralized conference model

In the centralized conference model for SIP, introduced in RFC 4353 [RFC4353] "A Framework for Conferencing with the Session Initiation Protocol (SIP)", one function co-ordinates the communication with participants in the multi-party session. This function also controls media mixer functions for the media appearing in the session. The central function is common for control of all media, while the media mixers may work differently for each media.

The central function is called the Focus UA. Many variants exist for setting up sessions including the multipoint control centre. It is not within scope of this description to describe these, but rather the media specific handling in the mixer required to handle multi-party calls with RTT.

The main principle for handling real-time text media in a centralized conference is that one RTP session for real-time text is established including the multipoint media control centre and the participating endpoints which are going to have real-time text exchange with the others.

The different possible mechanisms for mixing and transporting RTT differs in the way they multiplex the text streams and how they identify the sources of the streams. RFC 7667 [RFC7667] describes a number of possible use cases for RTP. This specification refers to different sections of RFC 7667 for further reading of the situations caused by the different possible design choices.

The recommended method for using RTP based RTT in a centralized conference model is specified in [I-D.ietf-avtcore-multi-party-rtt-mix] based on the recommendations in this document.

Real-time text can also be transported in the WebRTC environment, by using WebRTC data channels according to [I-D.ietf-mmusic-t140-usage-data-channel]. Ways to handle multi-party calls in that environmnent are also specified.

3. Requirements on multi-party RTT

3.1. General requirements

The following general requirements are placed on multi-party RTT:

A solution shall be applicable to IMS (3GPP TS 22.173)[TS22173], SIP based VoIP and Next Generation Emergency Services (NENA i3 [NENAi3], ETSI TS 103 479 [TS103479], RFC 6443[RFC6443]).

The transmission interval for text should not be longer than 500 milliseconds when there is anything available to send. Ref ITU-T T.140 [T140].

If text loss is detected or suspected, a missing text marker should be inserted in the text stream. Ref ITU-T T.140 Amendment 1 [T140ad1]. ETSI EN 301 549 [EN301549]

The display of text from the members of the conversation shall be arranged so that the text from each participant is clearly readable, and its source and the relative timing of entered text is visualized in the display. Mechanisms for looking back in the contents from the current session should be provided. The text should be displayed as soon as it is received. Ref ITU-T T.140 [T140]

Bridges must be multimedia capable (voice, video, text). Ref NENA i3 STA-010.2. [NENAi3]

It MUST be possible to use real-time text in conferences both as a medium of discussion between individual participants (for example, for sidebar discussions in real-time text while listening to the main conference audio) and for central support of the conference with real-time text interpretation of speech. Ref (R7) in RFC 5194.[RFC5194]

It should be possible to protect RTT contents with usual means for privacy and integrity. Ref RFC 6881 section 16. [RFC6881]

Conferencing procedures are documented in RFC 4579 [RFC4579]. Ref NENA i3 STA-010.2.[NENAi3]

Conferencing applies to any kind of media stream by which users may want to communicate. Ref 3GPP TS 24.147 [TS24147]

The framework for SIP conferences is specified in RFC 4353 [RFC4353]. Ref 3GPP TS 24.147 [TS24147]

3.2. Performance requirements

The mixer performance requirements can be expressed in one number, extracted from the user requirements on real-time text expressed in ITU-T F.700, where it is stated that for "good" usability, text characters should not be delayed more than 1 second from creation to presentation. For "usable" usability the figure is 2 seconds. The main factor behind these limits is from when taking turns in a conversation gets disturbed by a delay of when a response gets visible to the receiving part. If that times get too long, the receiving part gets unsure if the previous utterance was well perceived and the receiving part maybe prepares for repetition. This is similar to the same effect in voice communication, where the usability limit is 400 ms delay.

Another important factor in a multi-party conference is the opportunity for a participant using real-time text to provide timely comments and get a chance to enter the discussion if the majority of participants use voice in the conference. A complicating factor when stating the requirements is that some transport methods do not cause a total delay, but instead an increasing jerkiness when the number of simultaneously sending participants is increased.

It should however be remembered that the expected number of participants sending real-time text simultaneously is low. Just as with voice or sign language, the capability of the participants to perceive utterances from more than one participant at a time is very limited. Therefore the normal case in multi-party situations is that one participant at a time is the main provider of text. Others might usually just provide very brief comments such as "yes" or "no" or "may I comment?". Only at very rare situations two participants provide more information simultaneously.

*The number of expected simultaneously transmitting users is different for different applications. In all cases, just one transmitting user is the normal case. Two simultaneously transmitting participants can occasionally be expected in emergency services, relay services, small unmanaged conferences and group calls and large managed conferences. Three simultaneously transmitting participants may appear occasionally in large unmanaged conferences. The following can therefore express the performance requirement.

*The mean delay of text passing the mixer introduced when only one participant is sending text should be kept to a minimum and should not be more than 400 ms.

*The mean delay of text passing the mixer should not be more than 1 second during moments when up to three users are sending text simultaneously.

*For the very rare case that more than three participants send text simultaneously, the mixer may take action to limit the introduced delay of the text passing the mixer to 7 seconds e.g. by discarding text from some participants and instead inserting a general warning about possible text loss in the stream.

4. RTP based solutions

4.1. Coordination of text RTP streams

Coordinating and sending text RTP streams in the multi-party session can be done in a number of ways. The most suitable methods are specified here with pros and cons.

A receiving and presenting endpoint MUST separate text from the different sources and identify and display them accordingly.

4.1.1. RTP-based solutions with a central mixer

A set of solutions can be based on the central RTP mixer. They are described here and a preferred method selected.

4.1.1.1. RTP Mixer using default RFC 4103 methods

Without any extra specifications, a mixer would transmit with 300 milliseconds intervals, and use RFC 4103 [RFC4103] with the default redundancy of one original and two redundant transmissions. The source of the text would be indicated by a single member in the CSRC list. Text from different sources cannot be transmitted in the same packet. Therefore, from the time when the mixer sent one piece of new text from one source, it will need to transmit that text again twice as redundant data, before it can send text from another source. The jerkiness = time between transmission of new text is 900 ms. This is clearly insufficient.

Pros:

Only a capability negotiation method is needed. No other update of standards are needed, just a general remark that traditional RTP-mixing is used.

Cons:

Clearly insufficient mixer switching performance.

A bit complex handling of transmission when there is new text available from more than one source. The mixer needs to send two packets more with redundant text from the current source before starting to send anything from the other source.

4.1.1.2. RTP Mixer using the default method but decreased transmission interval

This method makes use of the default RTP-mixing method briefly described in Section 4.1.1.1. The only difference is that the transmission interval is decreased to 100 milliseconds when there is text from more than one source available for transmission. The jerkiness is 300 ms. The mean delay with two simultaneously sending participants is 250 ms, and with three simultaneously sending participants 500 ms. This is acceptable performance.

Pros:

Minor influence on standards

Can be relatively rapidly be introduced in the intended technical environments.

Can be declared in sdp as the already existing "text/red" format with a multi-party attribute for capability negotiation.

Cons:

The introduced jerkiness of new text from more than the required three simultaneously sending sources is high.

Slightly higher risk for loss of text at bursty packet loss than for the recommended transmission interval (300 ms) for RFC 4103.

When complete loss of packets occur (beyond recovery), it is not possible to deduct from which source text was lost.

A bit complex handling of transmission when there is new text available from more than one source. The mixer needs to send two packets more with redundant text from the current source before starting to send anything from the other source.

4.1.1.3. RTP Mixer with frequent transmission and indicating sources in CSRC-list

An RTP media mixer combines text from participants into one RTP stream, thus all using the same destination address/port combination, the same RTP SSRC, and one sequence number series as described in Section 7.1 and 7.3 of RTP RFC 3550 [RFC3550] about the

Mixer function. This method is also briefly described in RFC 7667, section 3.6.1 Media mixing mixer [RFC7667].

The sources of the text in each RTP packet are identified by the CSRC list in the RTP packets, containing the SSRC of the initial sources of text. The order of the CSRC parameters is with the SSRC of the source of the primary text first, followed by the SSRC of the first level redundancy, and then the second level redundancy.

The transmission interval should be 100 milliseconds when there is text to transmit from more than one source, and otherwise 300 ms.

The identification of the sources is made through the CSRC fields and can be made more readable at the receiver through the RTCP SDES CNAME and NAME packets as described in RTP[RFC3550].

Information provided through the notification according to RFC 4575 [RFC4575] when the participant joined the conference provides also suitable information and a reference to the SSRC.

A receiving endpoint is supposed to separate text items from the different sources and identify and display them accordingly.

The ordered CSRC lists in the RFC 4103 [RFC4103] packets make it possible to recover from loss of one and two packets in sequence and assign the recovered text to the right source. For more loss, a marker for possible loss should be inserted or presented.

The conference server needs to have authority to decrypt the payload in the received RTP packets in order to be able to recover text from redundant data or insert the missing text marker in the stream, and repack the text in new packets.

Even if the format is very similar to "text/red" of RFC 4103, it needs to be declared as a new media subtype, e.g. "text/rex".

Pros:

This method has low overhead and less complexity than the methods in Section 4.1.1.1, Section 4.1.1.2, Section 4.1.1.4 and Section 4.1.1.6.

When loss of packets occur, it is possible to recover text from redundancy at loss of up to the number of redundancy levels carried in the RFC 4103 [RFC4103] stream (normally primary and two redundant levels).

This method can be implemented with most RTP implementations.

The source switching performance is sufficient for well-behaving conference participants. The jerkiness is 100 ms.

Cons:

When more consecutive packet loss than the number of generations of redundant data appears, it is not possible to deduct the sources of the totally lost data.

Slightly higher risk for loss of text at bursty packet loss than for the recommended transmission interval for RFC 4103.

Requires a different sub media format, e.g. "text/rex". This takes a long time in standardisation and releases of target technical environments.

The conference server needs to be allowed to decrypt/encrypt the packet payload. This is however normal for media mixers for other media.

4.1.1.4. RTP Mixer using timestamp to identify redundancy

This method has text only from one source per packet, as the original RFC 4103 [RFC4103] specifies. Packets with text from different sources are instead allowed to be merged. The recovery procedure in the receiver will use the RTP timestamp and timestamp offsets in the redundancy headers to evaluate if a piece of redundant data should be recovered or not in case of packet loss.

In this method, the transmission interval is 100 milliseconds when text from more than one source is available for transmission.

Pros:

The format of each packet is equal to what is specified in RFC 4103 [RFC4103].

The source switching performance is sufficient. Text from five participants can be transmitted simultaneously with 500 milliseconds interval per source.

New text from five simultaneous sources can be transmitted within 500 milliseconds. This is sufficient.

Cons:

The recovery time in case of packet loss is long. With five simultaneously sending participants, it will be 1.5 seconds.

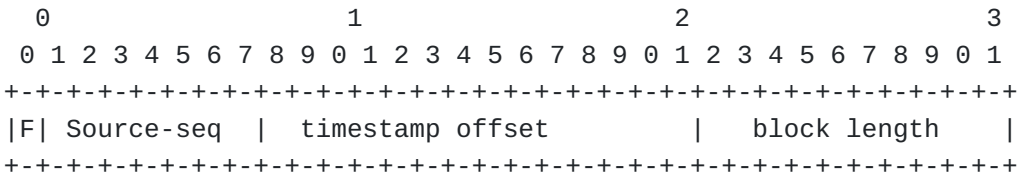
The recovery procedure is complex and very different from what is described in RFC 4103 [RFC4103].

It is not sure that this change can be regarded to be an update to RFC 4103. It may need a new media subtype.

4.1.1.5. RTP Mixer with multiple primary data in each packet and individual sequence numbers

This method allows primary as well as redundant text from more than one source per packet. The packet payload contains an ordered set of redundant and primary data with the same number of generations of redundancy as once agreed in the SDP negotiation. The data header reflects these parts of the payload. The CSRC list contains one CSRC member per source in the payload and in the same order. An individual sequence number per source is included in the data header replacing the t140 payload type number that is instead assumed to be constant in this format. This allows an individual extra sequence number per source with maximum value 127, suitable for checking for which source loss of text appeared when recovery was not possible.

The data header would contain the following fields:



Where "Source-seq" is the sequence number per source.

The maximum number of members in the CSRC-list is 15, and that is therefore the maximum number of sources that can be represented in each packet provided that all data can be fitted into the size allowable in one packet.

Transmission is done as soon as there is new text available, but not with shorter interval than 150 ms and not longer than 300 ms while there is anything to send.

A new media subtype is needed, e.g. "text/rex".

This is an SDP offer example for both traditional "text/red" and multi-party "text/rex" format:

```
m=text 11000 RTP/AVP 101 100 98
a=rtpmap:98 t140/1000
a=rtpmap:100 red/1000
a=rtpmap:101 rex/1000
a=fmtp:100 98/98/98
a=fmtp:101 98/98/98
```

Pros:

The source switching performance is good. Text from 15 participants can be transmitted simultaneously.

New text from 15 simultaneous sources can be transmitted within 300 milliseconds. This is good performance.

When more consecutive packet loss than the number of generations of redundant data appears, it is still possible to deduct the sources of the totally lost data, when next text from these sources arrive.

Cons:

The format of each packet is different from what is specified in RFC 4103 [RFC4103].

The processing time in standard organisation will be long.

A new media subtype is needed, causing a bit complex negotiation.

The recovery procedure is a bit complex.

4.1.1.6. RTP Mixer with multiple primary data in each packet

This method allows primary as well as redundant text from more than one source per packet. The packet payload contains an ordered set of redundant and primary data with the same number of generations of redundancy as once agreed in the SDP negotiation. The data header reflects these parts of the payload. The CSRC list contains one CSRC member per source in the payload and in the same order.

The maximum number of members in the CSRC-list is 15, and that is therefore the maximum number of sources that can be represented in each packet provided that all data can be fitted into the size allowable in one packet.

Transmission is done as soon as there is new text available, but not with shorter interval than 150 ms and not longer than 300 ms while there is anything to send.

A new media subtype is needed, e.g. "text/rex".

SDP would be the same as in Section 4.1.1.6.

Pros:

The source switching performance is good. Text from 15 participants can be transmitted simultaneously.

New text from 15 simultaneous sources can be transmitted within 150 milliseconds. This is good performance.

Cons:

The format of each packet is different from what is specified in RFC 4103 [RFC4103].

A new media subtype is needed.

A new media subtype is needed, causing a bit complex negotiation.

The processing time in standard organisation will be long.

The recovery procedure is a bit complex [RFC4103].

When more consecutive packet loss than the number of generations of redundant data appears, it is not possible to deduct the sources of the totally lost data.

4.1.1.7. RTP Mixer with RFC 5109 FEC and RFC 2198 redundancy in the packets

This method allows primary data from one source and redundant text from other sources in each packet. The packet payload contains primary data in "text/t140" format, and redundant data in RFC 5109 FEC [RFC5109] format called "text/ulpfec". That means that the redundant data contains the sequence number and the CSRC and other characteristics from the RTP header when the data was sent as primary. The redundancy can be sent at a selected number of packets after when it was sent as primary, in order to improve the protection against bursty packet loss. The redundancy level is recommended to be the same as in original RFC 4103.

RFC 4103 says that the protection against loss can be made by other methods than plain redundancy, so this method is in line with that statement.

Transmission is done as soon as there is new text available, but not with shorter interval than 100 ms and not longer than 300 ms while there is anything to send (new or redundant text).

When more consecutive packet loss than the number of generations of redundant data appears, it is not possible to deduct the sources of the totally lost data.

The sdp can indicate the format as "text/red" with "text/ulpfec" redundant data in this way. with traditional RFC 4103 with "text/red" with "text/t140" as redundant data as a fallback.

```
m=text 49170 RTP/AVP 98 101 100 102
a=rtpmap:98 red/1000
a=fmtp:98 100/102/102
a=rtpmap:102 ulpfec/1000
a=rtpmap:100 t140/1000
a=rtpmap:101 red/1000
a=fmtp:101 100/100/100
a=fmtp:100 cps=200
```

The "text/ulpfec" format includes an indication of how far back the redundancy belongs, making it possible to cover bursty packet loss better than the other formats with short transmission intervals. For real-time text, it is recommended to send three packets between the primary and the redundant transmissions of text. That makes the transmission cover between 500 and 1500 ms of bursty packet loss. The variation is because of the varying packet interval between many and one simultaneously transmitting source.

The "text/ulpfec" format has a number of parameters. One is the length of the data to be protected which in this case must be the whole t140block.

Pros:

The source switching performance is good. Text from 5 participants can be transmitted within 500 ms.

Good recovery from bursty packet loss.

The method is based on existing standards. No new registrations are needed.

Cons:

When more consecutive packet loss than the number of generations of redundant data appears, it is not possible to deduct the sources of the totally lost data.

Even if the switching performance is good, it is not as good as for the method called "RTP Mixer with multiple primary data in each packet "Section 4.1.1.6. With more than 5 simultaneously sending sources, there will be a noticeable delay of text of over 500 ms, with 100 ms added per simultaneous source. This is however beyond the requirements and would be a concern only in congestion situations.

The recovery procedure is a bit complex [RFC5109].

There is more overhead in terms of extra data and extra packets sent than in the other methods. With the recommended two redundant generations of data, each packet will be 36 bytes longer than with traditional RFC 4103, and at each pause in transmission five extra packets with only redundant data will be sent compared to two extra packets for the traditional RFC 4103 case.

4.1.1.8. RTP Mixer with RFC 5109 FEC and RFC 2198 redundancy and separate sequence number in the packets

This method allows primary data from one source and redundant text from other sources in each packet. The packet payload contains primary data in a new "text/t140e" format, and redundant data in RFC 5109 FEC [RFC5109] format called "text/ulpfec". That means that the redundant data contains the sequence number and the CSRC and other characteristics from the RTP header when the data was sent as primary. The redundancy can be sent at a selected number of packets after when it was sent as primary, in order to improve the protection against bursty packet loss. The redundancy level is recommended to be the same as in original RFC 4103. The "text/t140e" format contains a source-specific sequence number and the t140block.

RFC 4103 says that the protection against loss can be made by other methods than plain redundancy, so this method is in line with that statement.

Transmission is done as soon as there is new text available, but not with shorter interval than 100 ms and not longer than 300 ms while there is anything to send (new or redundant text).

When more consecutive packet loss than the number of generations of redundant data appears, it is possible to deduct which sources lost data when new data arrives from the sources. This is done by monitoring the received source specific sequence numbers preceding the text.

This is an example of how can indicate the format as "text/red" with "text/t140e" as primary and "text/ulpfec" redundant data, with traditional RFC 4103 with "text/red" with "text/t140" as redundant data as a fallback.

```
m=text 49170 RTP/AVP 98 101 100 102 103
a=rtpmap:98 red/1000
a=fmtp:98 100/102/102
a=rtpmap:102 ulpfec/1000
a=rtpmap:103 t140/1000
a=rtpmap:100 t140e/1000
a=rtpmap:101 red/1000
a=fmtp:101 103/103/103
a=fmtp:100 cps=200
```

The "text/ulpfec" format includes an indication of how far back the redundancy belongs, making it possible to cover bursty packet loss better than the other formats with short transmission intervals. For real-time text, it is recommended to send three packets between the primary and the redundant transmissions of text. That makes the transmission cover between 500 and 1500 ms of bursty packet loss. The variation is because of the varying packet interval between many and one simultaneously transmitting source.

The "text/ulpfec" format has a number of parameters. One is the length of the data to be protected which in this case must be the whole t140block.

Pros:

The source switching performance is good. Text from 5 participants can be transmitted within 500 ms.

Good recovery from bursty packet loss.

The method is based on an existing standard for FEC.

When more consecutive packet loss than the number of generations of redundant data appears, it is possible to deduct the source of the lost data when new text arrives from the source.

Cons:

Even if the switching performance is good, it is not as good as for the method called "RTP Mixer with multiple primary data in each packet" Section 4.1.1.6. With more than 5 simultaneously sending sources, there will be a noticeable delay of text of over 500 ms, with 100 ms added per simultaneous source. This is however beyond the requirements and would be a concern only in congestion situations.

The recovery procedure is a bit complex [RFC5109].

There is more overhead in terms of extra data and extra packets sent than in the other methods. With the recommended two redundant generations of data, each packet will be 40 bytes longer than with traditional RFC 4103, and at each pause in transmission five extra packets with only redundant data will be sent compared to two extra packets for the traditional RFC 4103 case.

A new text media subtype "text/t140e" needs to be registered.

The processing time in standard organisation will be long.

4.1.1.9. RTP Mixer indicating participants by a control code in the stream

Text from all participants except the receiving one is transmitted from the media mixer in the same RTP session and stream, thus all using the same destination address/port combination, the same RTP SSRC and , one sequence number series as described in Section 7.1 and 7.3 of RTP RFC 3550 [RFC3550] about the Mixer function. The sources of the text in each RTP packet are identified by a new defined T.140 control code "c" followed by a unique identification of the source in UTF-8 string format.

The receiver can use the string for presenting the source of text. This method is on the RTP level described in RFC 7667, section 3.6.1 Media mixing mixer [RFC7667].

The inline coding of the source of text is applied in the data stream itself, and an RTP mixer function is used for coordinating the sources of text into one RTP stream.

Information uniquely identifying each user in the multi-party session is placed as the parameter value "n" in the T.140 application protocol function with the function code "c". The identifier shall thus be formatted like this: SOS c n ST, where SOS and ST are coded as specified in ITU-T T.140 [T140]. The "c" is the letter "c". The n parameter value is a string uniquely identifying the source. This parameter shall be kept short so that it can be repeated in the transmission without concerns for network load.

A receiving endpoint is supposed to separate text items from the different sources and identify and display them accordingly.

The conference server need to be allowed to decrypt/encrypt the packet payload in order to check the source and repack the text.

Pros:

If loss of packets occur, it is possible to recover text from redundancy at loss of up to the number of redundancy levels carried

in the RFC 4103 [RFC4103]stream. (normally primary and two redundant levels.

This method can be implemented with most RTP implementations.

The method can also be used with other transports than RTP

Cons:

The method implies a moderate load by the need to insert the source often in the stream.

If more consecutive packet loss than the number of generations of redundant data appears, it is not possible to deduct the source of the totally lost data.

The mixer needs to be able to generate suitable and unique source identifications which are suitable as labels for the sources.

Requires an extension on the ITU-T T.140 standard, best made by the ITU.

There is a risk that the control code indicating the change of source is lost and the result is false source indication of text.

The conference server need to be allowed to decrypt/encrypt the packet payload.

4.1.1.10. Mixing for multi-party unaware user agents

Multi-party real-time text contents can be transmitted to multi-party unaware user agents if source labelling and formatting of the text is performed by a mixer. This method has the limitations that the layout of the presentation and the format of source identification is purely controlled by the mixer, and that only one source at a time is allowed to present in real-time. Other sources need to be stored temporarily waiting for an appropriate moment to switch the source of transmitted text. The mixer controls the switching of sources and inserts a source identifier in text format at the beginning of text after switch of source. The logic of the mixer to detect when a switch is appropriate should detect a number of places in text where a switch can be allowed, including new line, end of sentence, end of phrase, a period of inactivity, and a word separator after a long time of active transmission.

This method MAY be used when no support for multi-party awareness is detected in the receiving endpoint.The base for his method is described in RFC 7667, section 3.6.1 Media mixing mixer [RFC7667].

See [I-D.ietf-avtcore-multi-party-rtt-mix] for a procedure for mixing RTT for a conference-unaware endpoint.

Pros:

Can be transmitted to conference-unaware endpoints.

Can be used with other transports than RTP

Cons:

Does not allow full real-time presentation of more than one source at a time. Text from other sources will be delayed.

The only realistic presentation format is a style with the text from the different sources presented with a text label indicating source, and the text collected in a chat style presentation but with more frequent turn-taking.

Endpoints often have their own system for adding labels to the RTT presentation. In that case there will be two levels of labels in the presentation, one for the mixer and one for the sources.

If loss of more packets than can be recovered by the redundancy appears, it is not possible to detect which source was struck by the loss. It is also possible that a source switch occurred during the loss, and therefore a false indication of the source of text can be provided to the user after such loss.

Because of all these cons, this method is not recommended and should be used as the main method, but only as fallback and the last resort for backwards interoperability with multi-party unaware endpoints.

The conference server need to be allowed to decrypt/encrypt the packet payload.

4.1.2. RTP-based bridging with minor RTT media contents reformatting by the bridge

It may be desirable to send text in a multi-party setting in a way that allows the text stream contents to be distributed without being dealt with in detail in any central server. A number of such methods are described. However, when writing this specification, no one of these methods have a specified way of establishing the session by sdp.

4.1.2.1. RTP Translator sending one RTT stream per participant

Within the RTP session, text from each participant is transmitted from the RTP media translator (bridge) in a separate RTP stream,

thus using the same destination address/port combination, the same payload type number (PT) but separate RTP SSRC parameters and sequence number series as described in Section 7.1 and 7.2 of RTP RFC 3550 [RFC3550] about the Translator function. The source of the text in each RTP packet is identified by the SSRC parameter in the RTP packets, containing the SSRC of the initial source of text.

A receiving and presenting endpoint is supposed to separate text items from the different sources and identify and display them in a suitable way.

This method is described in RFC 7667, section 3.5.1 Relay-transport translator or 3.5.2 Media translator [RFC7667].

The identification of the source is made through the SSRC. The translation to a readable label can be done by mapping to information from the RTCP SDES CNAME and NAME packets as described in RTP[RFC3550], and also through information in the text media member in the conference notification described in RFC 4575 [RFC4575].

The sdp exchange for establishing this mixing type can be equal to what is used for basic two-party use of RFC 4103 with just an added attribute for indicating multi-party capability.

```
m=text 49170 RTP/AVP 98 103
a=rtpmap:98 red/1000
a=fmtp:98 103/103/103
a=rtpmap:103 t140/1000
a=fmtp:103 cps=150
a=RTT-mix:RTP-translator
```

A similar answer including the same RTT-mix attribute would indicate that multi-party coding can begin. An answer without the same RTT-mix attribute could result in diversion to use of the mixing method for multi-party unaware endpoints Section 4.1.1.10 if more than two parties are involved in the session.

The bridge can add new sources in the communication to a participant by first sending a conference notification according to RFC 4575 [RFC4575] with the SSRC of the new source included in the corresponding "text" media member, or by sending an RTCP message with the new SSRC in an SDES packet.

A receiver should be prepared to receive such indications of new streams being added to the multi-party session, so that the new SSRC

is not taken for a change in SSRC value for an already established RTP stream.

Transmission, reception, packet loss recovery and text loss indication is performed per source in the separate RTP streams in the same way as in two-party sessions with RFC 4103 [RFC4575].

Text is recommended to be sent by the bridge as soon as it is available for transmission, but not less than 250 ms after a previous transmission. This will in many cases result in close to 0 added delay by the bridge, because most RTT senders use a 300 ms transmission interval.

It is sometimes said that this configuration is not supported by current media declarations in sdp. RFC 3264 [RFC3264]specifies in some places that one media description is supposed to describe just one RTP media stream. However this is not directly referencing an RTP stream, and use of multiple RTP streams in the same RTP session is recommended in many other RFCs.

This confusion is clarified in RFC 5576 [RFC5576] section 3 by the following statements:

"The term "media stream" does not appear in the SDP specification itself, but is used by a number of SDP extensions, for instance, Interactive Connectivity Establishment (ICE) [ICE], to denote the object described by an SDP media description. This term is unfortunately rather confusing, as the RTP specification [RFC3550] uses the term "media stream" to refer to an individual media source or RTP packet stream, identified by an SSRC, whereas an SDP media stream describes an entire RTP session, which can contain any number of RTP sources."

In most cases, it will be sufficient that new sources are introduced with a conference notification or RTCP message. However, RFC 5576 [RFC5576] specifies attributes which may be used to more explicitly announce new sources or restart of earlier established RTP streams.

This method is encouraged by draft-ietf-avtcore-multiplex-guidelines [I-D.ietf-avtcore-multiplex-guidelines] section 5.2.

Normal operation will be that the bridge receives text packets from the source and handles any text recovery and indication of loss needed before queueing the resulting clean text for transmission from the bridge to the receivers.

It may however also be possible for the bridge to just convey the packet contents as received from the sources, with minor adjustments, and let the receiving endpoint handle all aspects of recovery and indication of loss, even for the source to bridge path.

In that case also the sequence number must be maintained as it was at reception in the bridge. This mode needs further study before application.

Pros:

This method is the natural way to do multi-party bridging with RFC 4103 based RTT. Only a small addition is included in the session establishment to verify capability by the parties because many implementations are done without multi-party capability.

This method has moderate overhead in terms of work for the mixer, but high in terms of packet transmission rate. Five sources sending simultaneously cause the bridge to send 15 packets per second to each receiver.

When loss of packets occur, it is possible to recover text from redundancy at loss of up to the number of redundancy levels carried in the RFC 4103 [RFC4103] stream(normally primary and two redundant levels).

More loss than what can be recovered, can be detected and the marker for text loss can be inserted in the correct stream.

It may be possible in some scenarios to keep the text encrypted through the Translator.

Minimal delay. The delay can often be kept close to 0 with at least 5 simultaneous sending participants.

Cons:

There are RTP implementations not supporting the Translator model. They will need to use the fall-back to multi-party-unaware mixing. An investigation about how common this is is needed before the method is used.

The processing time in standard organisation will be long.

With many simultaneous sending sources, the total rate of packets will be high, and can cause congestion. The requirement to handle 3 simultaneous sources in this specification will cause 10 packets per second that is manageable in most cases, e.g. considering that audio usually use 50 packets per second.

4.1.2.2. Distributing packets in an end-to-end encryption structure

In order to achieve end-to-end encryption, it is possible to let the packets from the sources just pass though a central distributor, and handle the security agreements between the participants.

Specifications exist for a framework with this functionality for application on RTP based conferences in [I-D.ietf-perc-private-media-framework]. The RTP flow and mixing characteristics has similarities with the method described under "RTP Translator sending one RTT stream per participant" above. RFC 4103 RTP streams [RFC4103] would fit into the structure and it would provide a base for end-to-end encrypted rtt multi-party conferencing.

Pros:

Good security

Straightforward multi-party handling.

Cons:

Does not operate under the usual SIP central conferencing architecture.

Requires the participants to perform a lot of key handling.

Is work in progress when this is written.

4.1.2.3. Mesh of RTP endpoints

Text from all participants are transmitted directly to all others in one RTP session, without a central bridge. The sources of the text in each RTP packet are identified by the source network address and the SSRC.

This method is described in RFC 7667, section 3.4 Point to multi-point using mesh [RFC7667].

Pros:

When loss of packets occur, it is possible to recover text from redundancy at loss of up to the number of redundancy levels carried in the RFC 4103 [RFC4103] stream. (normally primary and two redundant levels.

This method can be implemented with most RTP implementations.

Transmitted text can also be used with other transports than RTP

Cons:

This model is not described in IMS, NENA and EENA specifications, and does therefore not meet the requirements.

Requires a drastically increasing number of connections when the number of participants increase.

4.1.2.4. Multiple RTP sessions, one for each participant

Text from all participants are transmitted directly to all others in one RTP session each, without a central bridge. Each session is established with a separate media description in SDP. The sources of the text in each RTP packet are identified by the source network address and the SSRC.

Pros:

When loss of packets occur, it is possible to recover text from redundancy at loss of up to the number of redundancy levels carried in the RFC 4103 [RFC4103] stream. (normally primary and two redundant levels.

Complete loss of text can be indicated in the received stream.

This method can be implemented with most RTP implementations.

End-to-end encryption is achievable.

Cons:

This method is not described in IMS, NENA and ETSI specifications and does therefore not meet the requirements.

A lot of network resources are spent on setting up separate sessions for each participant.

5. Preferred RTP-based multi-party RTT transport method

For RTP transport of RTT using RTP-mixer technology, one method for multi-party mixing and transport stand out as fulfilling the goals best and is therefore recommended. That is: "RTP Mixer using the default method but decreased transmission interval" Section 4.1.1.2

For RTP transport in separate streams or sessions, no current recommendation can be made. A bridging method in the process of standardisation with interesting characteristics is the end-to-end encryption model "perc" Section 4.1.2.2.

6. Session control of RTP-based multi-party RTT sessions

General session control aspects for multi-party sessions are described in RFC 4575 [RFC4575] A Session Initiation Protocol (SIP) Event Package for Conference State, and RFC 4579 [RFC4579] Session

Initiation Protocol (SIP) Call Control - Conferencing for User Agents. The nomenclature of these specifications are used here.

The procedures for a multi-party aware model for RTT-transmission shall only be applied if a capability exchange for multi-party aware real-time text transmission has been completed and a supported method for multi-party real-time text transmission can be negotiated.

A method for detection of conference-awareness for centralized SIP conferencing in general is specified in RFC 4579 [RFC4579]. The focus sends the "isfocus" feature tag in a SIP Contact header. This causes the conference-aware endpoint to subscribe to conference notifications from the focus. The focus then sends notifications to the endpoint about entering and disappearing conference participants and their media capabilities. The information is carried XML-formatted in a 'conference-info' block in the notification according to RFC 4575 [RFC4575]. The mechanism is described in detail in RFC 4575 [RFC4575].

Before a conference media server starts sending multi-party RTT to an endpoint, a verification of its ability to handle multi-party RTT must be made. A decision on which mechanism to use for identifying text from the different participants must also be taken, implicitly or explicitly. These verifications and decisions can be done in a number of ways. The most apparent ways are specified here and their pros and cons described. One of the methods is selected to be the one to be used by implementations of the centralized conference model according to this specification.

6.1. Implicit RTT multi-party capability indication

Capability for RTT multi-party handling can be decided to be implicitly indicated by session control items.

The focus may implicitly indicate muti-party RTT capability by including the media child with value "text" in the RFC 4575 [RFC4575] conference-info provided in conference notifications.

An endpoint may implicitly indicate multi-party RTT capability by including the text media in the SDP in the session control transactions with the conference focus after the subscription to the conference has taken place.

The implicit RTT capability indication means for the focus that it can handle multi-party RTT according to the preferred method indicated in the RTT multi-party methods section above.

The implicit RTT capability indication means for the endpoint that it can handle multi-party RTT according to the preferred method indicated in the RTT multi-party methods section above.

If the focus detects that an endpoint implicitly declared RTT multi-party capability, it SHALL provide RTT according to the preferred method.

If the focus detects that the endpoint does not indicate any RTT multi-party capability, then it shall either provide RTT multi-party text in the way specified for conference-unaware endpoint above, or refuse to set up the session.

If the endpoint detects that the focus has implicitly declared RTT multi-party capability, it shall be prepared to present RTT in a multi-party fashion according to the preferred method.

Pros:

Acceptance of implicit multi-party capability implies that no standardisation of explicit RTT multi-party capability exchange is required.

Cons:

If other methods for multi-party RTT are to be used in the same implementation environment as the preferred ones, then capability exchange needs to be defined for them.

Cannot be used outside a strictly applied SIP central conference model.

6.2. RTT multi-party capability declared by SIP media-tags

Specifications for RTT multi-party capability declarations can be agreed for use as SIP media feature tags, to be exchanged during SIP call control operation according to the mechanisms in RFC 3840 [RFC3840] and RFC 3841 [RFC3841]. Capability for the RTT Multi-party capability is then indicated by the media feature tag "rtt-mix", with a set of possible values for the different possible methods.

The possible values in the list may for example be:

rtp-mixer

perc

rtp-mixer indicates capability for using the RTP-mixer based presentation of multi-party text.

perc indicates capability for using the perc based transmission of multi-party text.

Example: Contact: <sip:a2@beco.example.com>

;methods="INVITE,ACK,OPTIONS,BYE,CANCEL"

;+sip.rtt-mix="rtp-mixer"

If, after evaluation of the alternatives in this specification, only one mixing method is selected to be brought to implementation, then the media tag can be reduced to a single tag with no list of values.

An offer-answer exchange should take place and the common method selected by the answering party shall be used in the session with that UA.

When no common method is declared, then only the fallback method for multi-party unaware participants can be used, or the session dropped.

If more than one text media section is included in SDP, all must be capable of using the declared RTT multi-party method.

Pros:

Provides a clear decision method.

Can be extended with new mixing methods.

Can guide call routing to a suitable capable focus.

Cons:

Requires standardization and IANA registration.

Is not stream specific. If more than one text stream is specified, all must have the same type of multi-party capability.

Cannot be used in the WebRTC environment.

6.3. SDP media attribute for RTT multi-party capability indication

An attribute can be specified on media level, to be used in text media SDP declarations for negotiating RTT multi-party capabilities. The attribute can have the name "rtt-mix".

More than one attribute can be included in one media description.

The attribute can have a value. The value can for example be:

   rtp-mixer

   rtp-translator

   perc

rtp-mixer indicates capability for using the RTP-mixer and CSRC-list based mixing of multi-party text.

rtp-translator indicates capability for using the RTP-translator based mixing

perc indicates capability for using the perc based transmission of multi-party text.

An offer-answer exchange should take place and the common method selected by the answering party shall be used in the session with that endpoint.

When no common method is declared, then only the fallback method for multi-party unaware endpoints can be used.

Example: a=rtt-mix:rtp-mixer

If, after evaluation of the alternatives in this specification, only one mixing method is selected to be brought to implementation, then the attribute can be reduced to a single attribute with no list of values.

Pros:

Provides a clear decision method.

Can be extended with new mixing methods.

Can be used on specific text media.

Can be used also for SDP-controlled WebRTC sessions with multiple streams in the same data channel.

Cons:

Requires standardization and IANA registration.

Cannot guide SIP routing.

6.4. Simplified SDP media attribute for RTT multi-party capability indication

An attribute can be specified on media level, to be used in text media SDP declarations for negotiating RTT multi-party capabilities. The attribute can have a name suitable for the selected method and no value. It would be selected and used if only one method for multi-party rtt is brought forward from this specification, and the other suppressed or found to be possible to negotiate in another way.

An offer-answer exchange should take place and if both parties specify rtt-mixing capability with the same attribute, the selected mixing method shall be used.

When no common method is declared, then only the fallback method for multi-party unaware endpoints can be used, or the session not accepted for multi-party use.

Example: a=rtt-mix-rtp-mixer

Pros:

Provides a clear decision method.

Very simple syntax and semantics.

Can be used on specific text media.

Cons:

Requires standardization and IANA registration.

If another RTT mixing method is also specified in the future, then that method may also need to specify and register its own attribute, instead of if an attribute with a parameter value is used, when only an addition of a new possible value is needed.

Cannot guide SIP routing.

6.5. SDP format parameter for RTT multi-party capability indication

An FMTP format parameter can be specified for the RFC 4103 [RFC4103]media, to be used in text media SDP declarations for negotiating RTT multi-party capabilities. The parameter can have the name "rtt-mix", with one or more of its possible values.

The possible values in the list are:

rtp-mixer

perc

rtp-mixer indicates capability for using the RTP-mixer based mixing and presentation of multi-party text using the CSRC-list.

perc indicates capability for using the perc based transmission of multi-party text.

Example: a=fmtp 96 98/98/98 rtt-mix=rtp-mixer

If, after evaluation of the alternatives in this specification, only one mixing method is selected to be brought to implementation, then the parameter can be reduced to a single parameter with no list of values.

An offer-answer exchange should take place and the common method selected by the answering party shall be used in the session with that UA.

When no common method is declared, then only the fallback method can be used, or the session denied.

Pros:

Provides a clear decision method.

Can be extended with new mixing methods.

Can be used on specific text media.

Can be used also for SDP-controlled WebRTC sessions with multiple streams in the same data channel.

Cons:

Requires standardization and IANA registration.

May cause interop problems with current RFC4103 [RFC4103] implementations not expecting a new fmtp-parameter.

Cannot guide SIP routing.

6.6. A text media subtype for support of multi-party rtt

Indicating a specific text media subtype in SDP is a straightforward way for negotiating multi-party capability. Especially if there are format differences from the "text/red" and "text/t140" formats of RFC4103 [RFC4103], then this is a natural way to do the negotiation for multi-party rtt.

Pros:

No extra efforts if a new format is needed anyway.

Cons:

None specific to using the format indication for negotiation of multi-party capability. But only feasible if a new format is needed anyway.

6.7. Preferred capability declaration method for RTP-based transport.

If the preferred transport method is one with a specific media subtype in sdp, then speciication by media subtype is preferred.

If this would not be the case, then the preferred capability declaration method would be the one with a specific SDP attribute for the selected mixing method Section 6.4 because it is straightforward.

6.8. Identification of the source of text for RTP-based solutions

The main way to identify the source of text in the RTP based solution is by the SSRC of the sending participant. In the RTP-mixer solution, this SSRC is included in the CSRC list of the transmitted packets. Further identification that may be needed for better labelling of received text may be achieved from a number of sources. It may be the RTCP SDES CNAME and NAME reports, and in the conference notification data (RFC 4575) [RFC4575].

As soon as a new member is added to the RTP session, its characteristics should be transmitted in RTCP SDES CNAME and NAME reports according to section 6.5 in RFC 3550 [RFC3550]. The information about the participant should also be included in the conference data including the text media member in a notification according to RFC 4575 [RFC4575].

The RTCP SDES report, SHOULD contain identification of the source represented by the SSRC/CSRC identifier. This identification MUST contain the CNAME field and MAY contain the NAME field and other defined fields of the SDES report.

A focus UA SHOULD primarily convey SDES information received from the sources of the session members. When such information is not available, the focus UA SHOULD compose SSRC/CSRC, CNAME and NAME information from available information from the SIP session with the participant.

Provision of detailed information in the NAME field has security implications, especially if provided without encryption.

7. RTT bridging in WebRTC

Within WebRTC, real-time text is specified to be carried in WebRTC data channels as specified in [I-D.ietf-mmusic-t140-usage-data-channel]. A few ways to handle multi-party RTT are mentioned briefly. They are repeated below.

7.1. RTT bridging in WebRTC with one data channel per source

A straightforward way to handle multi-party RTT is for the bridge to open one T.140 data channel per source towards the receiving participants.

The stream-id forms a unique stream identification.

The identification of the source is made through the Label property of the channel, and session information belonging to the source. The endpoint can compose a readable label for the presentation from this information.

Pros:

This is a straightforward solution.

The load per source is low.

Cons:

With a high number of participants, the overhead of establishing and maintaining the high number of data channels required may be high, even if the load per channel is low.

7.2. RTT bridging in WebRTC with one common data channel

A way to handle multi-party RTT in WebRTC is for the bridge combine text from all sources into one data channel and insert the sources in the stream by a T.140 control code for source.

This method is described in a corresponding section for RTP transmission above in Section 4.1.1.9.

The identification of the source is made through insertion in the beginning of each text transmission from a source of a control code extension "c" followed by a string representing the source, framed by the control code start and end flags SOS and ST (See ITU-T T.140 [T140]).

A receiving endpoint is supposed to separate text items from the different sources and identify and display them in a suitable way.

The endpoint does not always display the source identification in the received text at the place where it is received, but has the information as a guide for planning the presentation of received text. A label corresponding to the source identification is presented when needed depending on the selected presentation style.

Pros:

This solution has relatively low overhead on session and network level

Cons:

This solution has higher overhead on the media contents level than the WebRTC solution above.

Standardisation of the new control code "c" in ITU-T T.140 [T140] is required.

The conference server need to be allowed to decrypt/encrypt the data channel contents.

7.3. Preferred rtt multi-party method for WebRTC

For WebRTC, one method is to prefer because of the simplicity. So, for WebRTC, the method to implement for multi-party RTT with multi-party aware parties when no other method is explicitly agreed between implementing parties is: "RTT bridging in WebRTC with one data channel per source" Section 7.1.

8. Presentation of multi-party text

All session participants with RTP based transport MUST observe the SSRC/CSRC field of incoming text RTP packets, and make note of which source they came from in order to be able to present text in a way that makes it easy to read text from each participant in a session, and get information about the source of the text.

In the WebRTC case, the Label parameter and other provided endpoint information should be used for the same purpose.

8.1. Associating identities with text streams

A source identity SHOULD be composed from available information sources and displayed together with the text as indicated in ITU-T T.140 Appendix[T140].

The source identity should primarily be the NAME field from incoming SDES packets. If this information is not available, and the session is a two-party session, then the T.140 source identity SHOULD be

composed from the SIP session participant information. For multi-party sessions the source identity may be composed by local information if sufficient information is not available in the session.

Applications may abbreviate the presented source identity to a suitable form for the available display.

Applications may also replace received source information with internally used nicknames.

8.2. Presentation details for multi-party aware endpoints.

The multi-party aware endpoint should after any action for recovery of data from lost packets, separate the incoming streams and present them according to the style that the receiving application supports and the user has selected. The decisions taken for presentation of the multi-party interchange shall be purely on the receiving side. The sending application must not insert any item in the stream to influence presentation that is not requested by the sending participant.

8.2.1. Bubble style presentation

One often used style is to present real-time text in chunks in readable bubbles identified by labels containing names of sources. Bubbles are placed in one column in the presentation area and are closed and moved upwards in the presentation area after certain items or events, when there is also newer text from another source that would go into a new bubble. The text items that allows bubble closing are any character closing a phrase or sentence followed by a space or a timeout of a suitable time (about 10 seconds).

Real-time active text sent from the local user should be presented in a separate area. When there is a reason to close a bubble from the local user, the bubble should be placed above all real-time active bubbles, so that the time order that real-time text entries were completed is visible.

Scrolling is usually provided for viewing of recent or older text. When scrolling is done to an earlier point in the text, the presentation shall not move the scroll position by new received text. It must be the decision of the local user to return to automatic viewing of latest text actions. It may be useful with an indication that there is new text to read after scrolling to an earlier position has been activated.

The presentation area may become too small to present all text in all real-time active bubbles. Various techniques can be applied to provide a good overview and good reading opportunity even in such

situations. The active real-time bubble may have a limited number of lines and if their contents need more lines, then a scrolling opportunity within the real-time active bubble is provided. Another method can be to only show the label and the last line of the active real-time bubble contents, and make it possible to expand or compress the bubble presentation between full view and one line view.

Erasures require special consideration. Erasure within a real-time active bubble is straightforward. But if erasure from one participant affects the last character before a bubble, the whole previous bubble becomes the actual bubble for real-time action by that participant and is placed below all other bubbles in the presentation area. If the border between bubbles was caused by the CRLF characters (instead of the normal "Line Separator"), only one erasure action is required to erase this bubble border. When a bubble is closed, it is moved up, above all real-time active bubbles.

A three-party view is shown in this example .

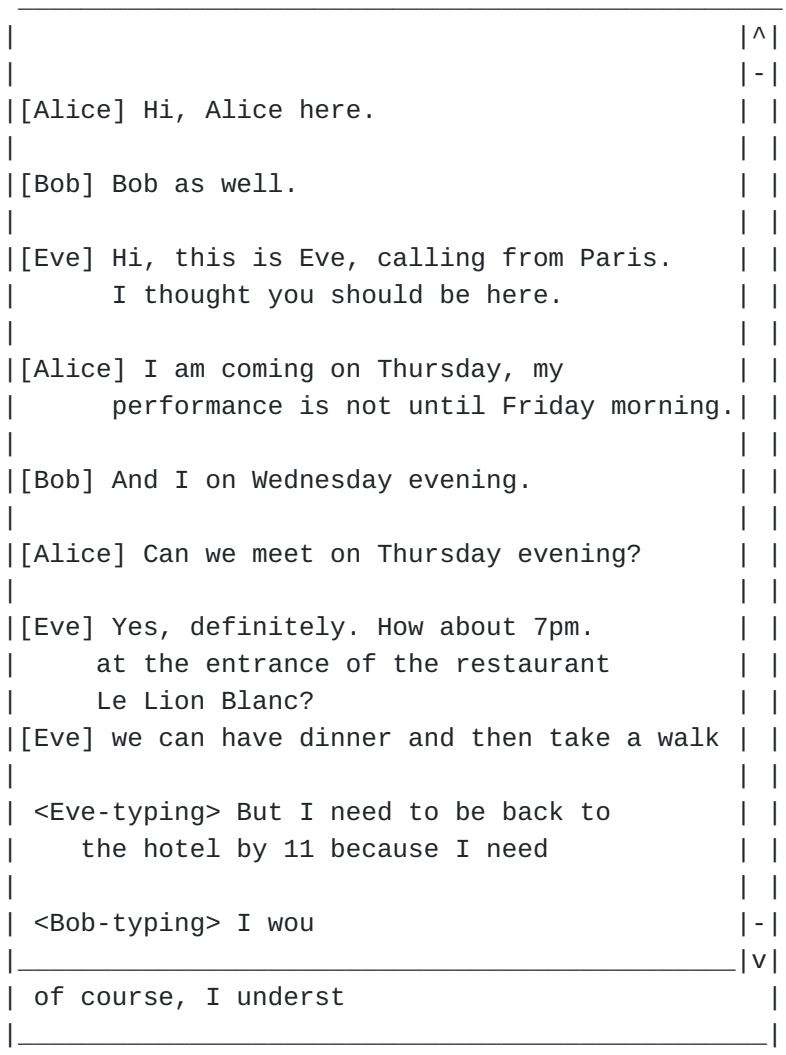


Figure 1: Three-party call with bubble style.

Figure 1: Example of a three-party call presented in the bubble style.

8.2.2. Other presentation styles

Other presentation styles than the bubble style may be arranged and appreciated by the users. In a video conference one way may be to have a real-time text area below the video view of each participant. Another view may be to provide one column in a presentation area for each participant and place the text entries in a relative vertical position corresponding to when text entry in them was completed. The labels can then be placed in the column header. The considerations for ending and moving and erasure of entered text discussed above for the bubble style are valid also for these styles.

This figure shows how a coordinated column view MAY be presented.

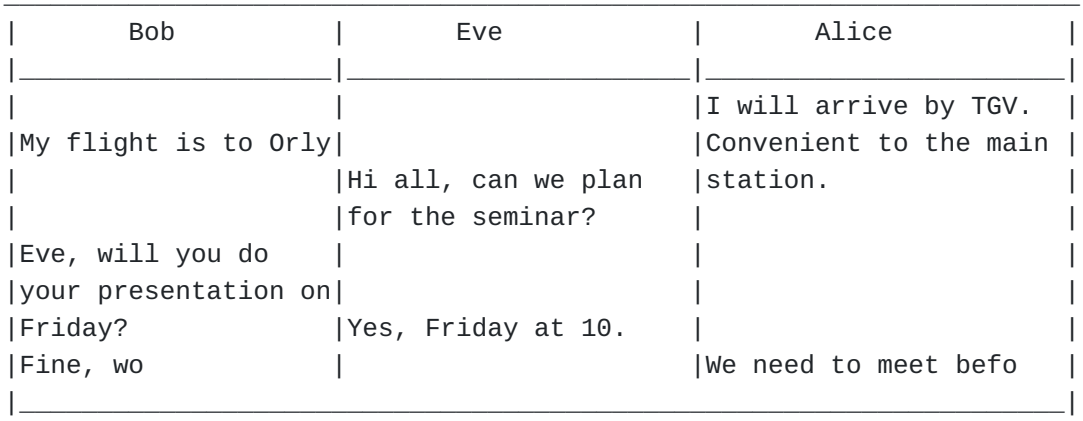


Figure 2: A coordinated column-view of a three-party session with entries ordered in approximate time-order.

9. Presentation details for multi-party unaware endpoints.

Multi-party unaware endpoints are prepared only for presentation of two sources of text, the local user and a remote user. If mixing for multi-party unaware endpoints is to be supported, in order to enable some multi-party communication with such endpoint, the mixer need to plan the presentation and insert labels and line breaks before lables. Many limitations appear for this presentation mode, and it must be seen as a fallback and a last resort.

A procedure for presenting RTT to a conference-unaware endpoint is included in [I-D.ietf-avtcore-multi-party-rtt-mix]

10. Security Considerations

The security considerations valid for RFC 4103 [RFC4103] and RFC 3550 [RFC3550] are valid also for the multi-party sessions with text.

11. IANA Considerations

The items for indication and negotiation of capability for multi-party rtt should be registered with IANA in the specifications where they are specified in detail.

12. Congestion considerations

The congestion considerations described in RFC 4103 [RFC4103] are valid also for the recommended RTP-based multi-party use of the real-time text transport. A risk for congestion may appear if a number of conference participants are active transmitting text simultaneously, because the recommended RTP-based multi-party transmission method does not allow multiple sources of text to contribute to the same packet.

In situations of risk for congestion, the Focus UA MAY combine packets from the same source to increase the transmission interval per source up to one second. Local conference policy in the Focus UA may be used to decide which streams shall be selected for such transmission frequency reduction.

13. Acknowledgements

Arnoud van Wijk for contributions to an earlier, expired draft of this memo.

14. Change history

14.1. Changes to draft-hellstrom-avtcore-multi-party-rtt-solutions-03

Modified info on the method with RFC 4103 format and sdp attribute "rtt-mix-rtp-mixer".

Increased the performance requirements section.

Inserted recommendations, with emphasis on ease of implementation and ease of standardisation.

14.2. Changes to draft-hellstrom-avtcore-multi-party-rtt-solutions-02

Added detail in the section on RTP translator model alternative 4.1.2.1.

14.3. Changes to draft-hellstrom-avtcore-multi-party-rtt-solutions-01

Added three more methods for RTP-mixer mixing. Two RFC 5109 FEC based and another with modified data header to detect source of completely lost text.

Separated RTP-based and WebRTC based solutions.

Deleted the multi-party-unaware mixing procedure appendix. It is now included in the draft draft-ietf-avtcore-multi-party-rtt-mix. Kept a section with a reference to the new place.

14.4. Changes from draft-hellstrom-mmusic-multi-party-rtt-02 to draft-hellstrom-avtcore-multi-party-rtt-solutions-00

Add discussion about switching performance, as discussed in avtcore on March 13.

Added that a decrease of transmission interval to 100 ms increases switching performance by a factor 3, but still not sufficient.

Added that the CSRC-list method also uses 100 milliseconds transmission interval.

Added the method with multiple primary text in each packet.

Added the timestamp-based method for rtp-mixing proposed by James Hamlin on March 14.

Corrected the chat style presentation example picture. Delete a few "[mix]".

14.5. Changes from version draft-hellstrom-mmusic-multi-party-rtt-01 to -02

Change from a general overview to overview with clear recommendations.

Splits text coordination methods in three groups.

Recommends rtt-mixer with sources in CSRC-list but referenes to its spec for details.

Shortened Appendix with conference-unaware example.

Cleaned up preferences.

Inserted pictures of screen-views.

15. References

15.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

15.2. Informative References

[EN301549] ETSI, "EN 301 549. Accessibility requirements for ICT products and services", November 2019, <https://www.etsi.org/deliver/etsi_en/301500_301599/301549/03.01.01_60/en_301549v030101p.pdf>.

[I-D.ietf-avtcore-multi-party-rtt-mix] Hellstrom, G., "RTP-mixer formatting of multi-party Real-time text", Work in Progress, Internet-Draft, draft-ietf-avtcore-multi-party-rtt-mix-06, 11 June 2020, <https://tools.ietf.org/html/draft-ietf-avtcore-multi-party-rtt-mix-06>.

[I-D.ietf-avtcore-multiplex-guidelines]
Westerlund, M., Burman, B., Perkins, C., Alvestrand, H., and R. Even, "Guidelines for using the Multiplexing Features of RTP to Support Multiple Media Streams", Work in Progress, Internet-Draft, draft-ietf-avtcore-multiplex-guidelines-12, 16 June 2020, <https://tools.ietf.org/html/draft-ietf-avtcore-multiplex-guidelines-12>.

[I-D.ietf-mmusic-t140-usage-data-channel]
Holmberg, C. and G. Hellstrom, "T.140 Real-time Text Conversation over WebRTC Data Channels", Work in Progress, Internet-Draft, draft-ietf-mmusic-t140-usage-data-channel-14, 10 April 2020, <https://tools.ietf.org/html/draft-ietf-mmusic-t140-usage-data-channel-14>.

[I-D.ietf-perc-private-media-framework]
Jones, P., Benham, D., and C. Groves, "A Solution Framework for Private Media in Privacy Enhanced RTP Conferencing (PERC)", Work in Progress, Internet-Draft, draft-ietf-perc-private-media-framework-12, 5 June 2019, <https://tools.ietf.org/html/draft-ietf-perc-private-media-framework-12>.

[NENAi3] NENA, "NENA-STA-010.2-2016. Detailed Functional and Interface Standards for the NENA i3 Solution", October 2016, <https://www.nena.org/page/i3_Stage3>.

[RFC2198] Perkins, C., Kouvelas, I., Hodson, O., Hardman, V., Handley, M., Bolot, J.C., Vega-Garcia, A., and S. Fosse-Parisis, "RTP Payload for Redundant Audio Data", RFC 2198, DOI 10.17487/RFC2198, September 1997, <https://www.rfc-editor.org/info/rfc2198>.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, DOI 10.17487/RFC3261, June 2002, <https://www.rfc-editor.org/info/rfc3261>.

[RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, DOI 10.17487/RFC3264, June 2002, <https://www.rfc-editor.org/info/rfc3264>.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550, July 2003, <https://www.rfc-editor.org/info/rfc3550>.

[RFC3840] Rosenberg, J., Schulzrinne, H., and P. Kyzivat, "Indicating User Agent Capabilities in the Session Initiation Protocol (SIP)", RFC 3840, DOI 10.17487/RFC3840, August 2004, <https://www.rfc-editor.org/info/rfc3840>.

[RFC3841] Rosenberg, J., Schulzrinne, H., and P. Kyzivat, "Caller Preferences for the Session Initiation Protocol (SIP)", RFC 3841, DOI 10.17487/RFC3841, August 2004, <https://www.rfc-editor.org/info/rfc3841>.

[RFC4103] Hellstrom, G. and P. Jones, "RTP Payload for Text Conversation", RFC 4103, DOI 10.17487/RFC4103, June 2005, <https://www.rfc-editor.org/info/rfc4103>.

[RFC4353] Rosenberg, J., "A Framework for Conferencing with the Session Initiation Protocol (SIP)", RFC 4353, DOI 10.17487/RFC4353, February 2006, <https://www.rfc-editor.org/info/rfc4353>.

[RFC4575] Rosenberg, J., Schulzrinne, H., and O. Levin, Ed., "A Session Initiation Protocol (SIP) Event Package for Conference State", RFC 4575, DOI 10.17487/RFC4575, August 2006, <https://www.rfc-editor.org/info/rfc4575>.

[RFC4579] Johnston, A. and O. Levin, "Session Initiation Protocol (SIP) Call Control - Conferencing for User Agents", BCP 119, RFC 4579, DOI 10.17487/RFC4579, August 2006, <https://www.rfc-editor.org/info/rfc4579>.

[RFC4597] Even, R. and N. Ismail, "Conferencing Scenarios", RFC 4597, DOI 10.17487/RFC4597, August 2006, <https://www.rfc-editor.org/info/rfc4597>.

[RFC5109] Li, A., Ed., "RTP Payload Format for Generic Forward Error Correction", RFC 5109, DOI 10.17487/RFC5109, December 2007, <https://www.rfc-editor.org/info/rfc5109>.

[RFC5194] van Wijk, A., Ed. and G. Gybels, Ed., "Framework for Real-Time Text over IP Using the Session Initiation Protocol (SIP)", RFC 5194, DOI 10.17487/RFC5194, June 2008, <https://www.rfc-editor.org/info/rfc5194>.

[RFC5576] Lennox, J., Ott, J., and T. Schierl, "Source-Specific Media Attributes in the Session Description Protocol

(SDP)", RFC 5576, DOI 10.17487/RFC5576, June 2009, <https://www.rfc-editor.org/info/rfc5576>.

[RFC6443] Rosen, B., Schulzrinne, H., Polk, J., and A. Newton, "Framework for Emergency Calling Using Internet Multimedia", RFC 6443, DOI 10.17487/RFC6443, December 2011, <https://www.rfc-editor.org/info/rfc6443>.

[RFC6881] Rosen, B. and J. Polk, "Best Current Practice for Communications Services in Support of Emergency Calling", BCP 181, RFC 6881, DOI 10.17487/RFC6881, March 2013, <https://www.rfc-editor.org/info/rfc6881>.

[RFC7667] Westerlund, M. and S. Wenger, "RTP Topologies", RFC 7667, DOI 10.17487/RFC7667, November 2015, <https://www.rfc-editor.org/info/rfc7667>.

[T140] ITU-T, "Recommendation ITU-T T.140 (02/1998), Protocol for multimedia application text conversation", February 1998, <https://www.itu.int/rec/T-REC-T.140-199802-I/en>.

[T140ad1] ITU-T, "Recommendation ITU-T.140 Addendum 1 - (02/2000), Protocol for multimedia application text conversation", February 2000, <https://www.itu.int/rec/T-REC-T.140-200002-I!Add1/en>.

[TS103479] ETSI, "TS 103 479. Emergency communications (EMTEL); Core elements for network independent access to emergency services", December 2019, <https://www.etsi.org/deliver/etsi_ts/103400_103499/103479/01.01.01_60/ts_103479v010101p.pdf>.

[TS22173] 3GPP, "IP Multimedia Core Network Subsystem (IMS) Multimedia Telephony Service and supplementary services; Stage 1", 3GPP TS 22.173 17.1.0, 20 December 2019, <http://www.3gpp.org/ftp/Specs/html-info/22173.htm>.

[TS24147] 3GPP, "Conferencing using the IP Multimedia (IM) Core Network (CN) subsystem; Stage 3", 3GPP TS 24.147 16.0.0, 19 December 2019, <http://www.3gpp.org/ftp/Specs/html-info/24147.htm>.

Author's Address

Gunnar Hellstrom
Gunnar Hellstrom Accessible Communication
Esplanaden 30
SE-136 70 Vendelso
Sweden

Phone: +46 708 204 288
Email: gunnar.hellstrom@ghaccess.se