Real-time text media handling in multi-party conferences
draft-hellstrom-mmusic-multi-party-rtt-01

Abstract

   This memo specifies methods for Real-Time Text (RTT) media handling
   in multi-party calls. The main solution is to carry Real-Time text
   by the RTP protocol in a time-sampled mode according to RFC 4103.
   The main solution for centralized multi-party handling of real-time
   text is achieved through a media control unit coordinating multiple
   RTP text streams into one RTP session.

   Identification for the streams are provided through the CSRC lists in
   the RTP packets and through the RTCP messages. This mechanism
   enables the receiving application to present the received real-time
   text medium 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 one is provided. Two 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.

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

   EDITOR NOTE: A number of alternatives are specified for discussion.
   A decision is needed which alternatives are preferred and then how
   the preferred alternatives shall be emphasized.

Status of This Memo

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. This memo specifies how the real-time text streams are handled in multi-party sessions.

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, 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. This specification builds on that description and indicates which protocol mechanisms should be used to implement multi-party handling of real-time text.

EDITOR NOTE: A number of alternatives are specified for discussion. A decision is needed which alternatives are preferred and then how the preferred alternatives shall be emphasized.

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 medium.

The central function is called the Focus UA and may be co-located in an advanced terminal including multi-party control functions, or it may be located in a separate location. 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.

3. Requirements on multi-party RTT

The following requirements are placed on multi-party RTT:

   The solution shall be applicable to IMS (3GPP TS 22.173), SIP based VoIP and Next Generation Emergency Services (NENA i3, ETSI TS 103 479, RFC 6443).

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

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

   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

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

   R7: 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 RFC 5194.

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

Conferencing procedures are documented in RFC 4579. Ref NENA i3 STA-010.2.

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

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

4. 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 UA SHOULD separate text from the different sources and identify and display them accordingly.

4.1. RTP Translator sending one RTT stream per participant

Within the RTP session, text from each participant is transmitted from the RTP media translator in a separate RTP stream, thus using the same destination address/port combination, 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 sources of the text in each RTP packet are identified by the SSRC parameters in the RTP packets, containing the SSRC of the initial sources of text.

A receiving UA 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.

The identification of the source is made through the RTCP SDES CNAME and NAME packets as described in RTP[RFC3550].

Pros:

This method has moderate overhead. 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 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.

Cons:

There may be RTP implementations not supporting the Translator model.

It is even most likely that this configuration is not supported by
current media declarations in sdp. RFC 3264 specifies in many places
that one media description is supposed to describe just one RTP
stream.

4.2. RTP Mixer indicating sources in CSRC-list

An RTP media mixer combines text from all participants except from
the receiving endpoint 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. The sources of the text in
each RTP packet are identified by the CSRC parameters in the RTP
packets, containing the SSRC of the initial sources of text. The
order of the CSRC parameters are the same as the order of the
redundant and primary data fields in the packet. If all redundancy
blocks in a packet are from the same source, then it is allowed to
use only one CSRC in the RTP packet. This method is described in RFC
7667, section 3.6.3 Media switching mixer.

A set of specific rules for the application of this method together
with RFC 4103 is needed.

The identification of the source can be made through the RTCP SDES
CNAME and NAME packets as described in RTP[RFC3550].

Also information provided through the notification according to RFC
4575 when the participant joined the conference provides suitable
information and a reference to the SSRC.

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

The ordered CSRC lists in the RFC 4103 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 need to have authority to decrypt the payload
in the 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.

Pros:

This method has moderate overhead.

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 stream. (normally primary and two redundant levels.

This method can be implemented with most RTP implementations.

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. Therefore it is not possible to know in which
stream to insert the missing text marker. It MAY be acceptable to
either indicate a general loss indication, or insert a loss marker in
all streams. Calculations of most likely source can however be made
from received RTP and RTCP contents so that the loss marker can be
inserted in the most likely struck stream.

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

4.3. 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 suitable
for application on RTP based conferences in draft-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 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.

4.4. 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 3.6.2
   Media mixing mixer.

   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 [T.140]. 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 UA 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 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:

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. Therefore it is not possible to know in which
stream to insert the missing text marker. Calculations of most
likely source can however be made from recent history, so that it is
quite likely that the marker is inserted in the correct stream. Such
loss should however be rare, and a general warning that there might
have been text loss in the session might be acceptable.

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.

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

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

4.5. 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.

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 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.

4.6. 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.

This method is out of scope for further discussion here, because the foreseen applications use centralized model conferencing.

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 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 EENA specifications and does therefore not meet the requirements.

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

4.7. Mixing for conference-unaware user agents

Multi-party real-time text contents can be transmitted to conference-unaware user agents if source labeling 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 trhe 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 3.6.2 Media mixing mixer.

See Appendix A for an informative example of a procedure for presenting RTT to a conference-unaware UA.

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, even if automatic detection of suitable moments for switching source for presentation is made by the mixer.

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 MUST NOT be used as the main method, but only as the last resort for backwards interoperability with conference-unaware endpoints.

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

5. RTT bridging in WebRTC

Within WebRTC, real-time text is specified to be carried in WebRTC data channels as specified in draft-ietf-mmusic-t140-usage-data-channel. A few ways to handle multi-party RTT are mentioned briefly. They are explained and further detailed below.

5.1. 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 UA can compose a readable label for the presentation from this information.

Pros:

This is a straightforward solution.

Cons:

With a high number of participants, the overhead of establishing the high number of data channels required may be high.

5.2. 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.

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 [T.140]).

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

The UA 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 is
required.

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

6. Preferred multi-party RTT transport method

EDITOR NOTE: The recommendations here need to be validated, and the
proposed further studies performed.

For RTP transport of RTT, two methods for multi-party mixing and
transport for conference-aware parties stand out as fulfilling the
goals best is: "RTP Mixer indicating participants in CSRC".

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

7. Session control of 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 conference-aware model for RTT-transmission
shall only be applied if a capability exchange for conference-aware

real-time text transmission has been completed and a supported method for multi-party real-time text transmission can be identified.

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 UA to subscribe to conference notifications from the focus. The focus then sends notifications to the UA 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. The mechanism is described in detail in RFC 4575 [RFC4575].

Before a conference media server starts sending multi-party RTT to a UA, 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 according to this specification.

7.1. 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 conference-info provided in conference notifications.

A UA 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 UA 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 a UA implicitly declared RTT multi-party capability, it SHALL provide RTT according to the preferred method.

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

If the UA 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.

7.2. 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 and RFC 3841. Capability for the RTT Multi-party capability is then indicated by the media feature tag "rtt-mixer", with one or more of its possible values in a comma-separated list.

The possible values in the list are:

    rtp-translator

    rtp-mixer

    t140-mixer

    rtp-mesh

    multi-session

rtp-translator indicates capability for using the RTP-translator based coordination of multi-party text.

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

t140-mixer indicates capability for using the T.140 control code
source indicators in a mixer.

text-mixer indicates capability for using the fallback method with
text formatting for conference-unaware endpoints.

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

multi-session indicates capability for using separate point-to-point
RTP sessions between all participants.

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

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

;+sip.rtt-mixer="multi-session"

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 can
be used or the session dropped.

If more than one text media line 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.

7.3. 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-mixer", with one or more of its
possible values in a comma-separated list.

The possible values in the list are:

    rtp-translator

    rtp-mixer

    t140-mixer

    rtp-mesh

    multi-session

rtp-translator indicates capability for using the RTP-translator
based coordination of multi-party text.

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

t140-mixer indicates capability for using the T.140 control code
source indicators in a mixer.

text-mixer indicates capability for using the fallback method with
text formatting for conference-unaware endpoints.

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

multi-session indicates capability for using separate point-to-point
RTP sessions between all participants.

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.

Example: a=rtt-mixer: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.

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

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

The possible values in the list are:

   rtp-translator

   rtp-mixer

   t140-mixer

   rtp-mesh

   multi-session

rtp-translator indicates capability for using the RTP-translator
based coordination of multi-party text.

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

t140-mixer indicates capability for using the T.140 control code
source indicators in a mixer.

text-mixer indicates capability for using the fallback method with
text formatting for conference-unaware endpoints.

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

multi-session indicates capability for using separate point-to-point
RTP sessions between all participants.

Example: a=fmtp 96 98/98/98 cps=30;rtt-mixer=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.

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 implementations not
expecting a new fmtp-parameter.

Cannot guide SIP routing.

7.5. Preferred capability declaration method.

The preferred capability declaration method is the one with SDP
attributes because it is straightforward and partially usable also
for WebRTC.

8. Identification of the source of text

EDITOR NOTE: The text in the following sections need to be adapted
after recommendations for the main methods for coordination of RTT
has been selected. Details should be provided mainly for the
recommended method.

The main way to identify the source of text in the RTP based solution
is by the SSRC of the sending participant. It is included in the
CSRC list of the transmitted packets. Further identification that
may be needed for better labeling 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).

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. 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.

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.

9. Presentation of multi-party text

All session participants MUST observe the SSRC/CSRC field of incoming
text RTP packets, and make note of what 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.

9.1. 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 [T.140].

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.

9.2. Presentation details for multi-party aware UAs.

The multi-party aware UA 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.

9.2.1. 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, 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.

9.2.2. 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.

10. Presentation details for multi-party unaware UAs.

Multi-party unaware UA:s are prepared only for presentation of two
sources of text, the local user and a remote user. In order to
enable some multi-party communication with such UA, 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.

See Appendix A for an informative example of a procedure for
presenting RTT to a conference-unaware UA.

11. Transmission of text from each user

UAs participating in sessions with real-time text, SHOULD send SDES packets in RTCP giving values to appropriate identification fields.

The CNAME field SHALL be included in SDES packets.

The NAME field should be given a value that is suitable as an identifier of text from the user of the UA.

12. Robustness and indication of possible loss

This section discusses the means for robustness against loss of text that is already specified and their performance in the multi-party situation. means for reducing the risk for loss is discussed, as well as ways to detect in which stream loss has occurred.

TBD

13. Performance

This section discusses performance and performance limitations for the different transport solutions, and indicates which means for performance increase versus load limitations can be suitable to apply compared to the point-to-point case.

TBD

14. Security Considerations

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

15. IANA Considerations

EDITOR NOTE: TBD after decision of proposed preferences in the draft.

This document Introduces the TBD /SIP media tag/SDP media level attribute/ rtt-mixer, with a comma-separated parameter list containing the following possible values:

   rtp-translator

   rtp-mixer

   t140-mixer

   rtp-mesh

      multi-session

   rtp-translator indicates capability for using the RTP-translator
   based coordination of multi-party text.

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

   t140-mixer indicates capability for using the T.140 control code
   source indicators in a mixer.

   text-mixer indicates capability for using the fallback method with
   text formatting for conference-unaware endpoints.

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

   multi-session indicates capability for using separate point-to-point
   RTP sessions between all participants.

16. Congestion considerations

   The congestion considerations described in RFC 4103 are valid also
   for multi-party use of the real-time text RTP transport. A risk for
   congestion may appear if a number of conference participants are
   active transmitting text simultaneously, because this 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.

17. Acknowledgements

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

18. References

18.1. 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>.

   [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>.

   [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>.

   [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>.

   [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>.

   [T.140]   "Protocol for multimedia application text conversation",
             1998, <http://www.itu.int/rec/T-REC-T.140/en>.

18.2. Informative References

   [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>.

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

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

Appendix A. Mixing for a conference-unaware UA

   This informational appendix describes media mixer procedures for a
   multi-party conference server to format real-time text from a number
   of participants into one single text stream to a participant with a

terminal that has no features for multi-party text display. The
procedures are intended for implementations using ITU-T T.140 [T.140]
for the real-time text coding and presentation.

A.1. Short description

The media mixer procedures described here are intended to make real-
time text from a number of call participants be coordinated into one
text stream to a terminal originally intended for two-party calls. A
conference server is supposed to apply the procedures.

The procedures may also be applied on a terminal for display of
multiple streams of real-time text in one area.

The intention is that text from each participant shall be displayed
in suitable sections so that it is easy to read, and text from one
active participant at a time is sent and displayed in real-time. The
receiving terminal is assumed to have one display area for received
text. The display is arranged by this procedure in a text chat
style, with a name label in front of each text section where switch
of source of the text has taken place.

When more than one participant transmits text at the same time, the
text from only one of them is transmitted directly to the receiving
terminals. Text from the other participants is stored in buffers in
the conference server for transmission at a later time, when a
suitable situation for switch of current transmitter can take place.

A.2. Functionality goals and drawbacks

The procedures are intended to make best efforts to present a multi-
party text conversation on a terminal that has no awareness of multi-
party calls. There are some obvious drawbacks, and a terminal
designed with multi-party awareness will be able to present multi-
party call contents in a more flexible way. Only two parties at a
time will be allowed to display added text in real-time, while the
other parties' produced text will need to be stored in the multi-
party server for a moment awaiting a suitable occasion to be
displayed. There are also some cases of erasure that will not be
performed on the target text but only indicated in another way. Even
with these drawbacks, the procedure provides an opportunity to
display text from more than two parties in a smooth and readable way.

This specification does not introduce any new protocol element, and
does not rely on anything else than basic two-party terminal
functionality with presentation level according to ITU-T T.140
[T.140]. It is a description of a best current practice for mixing

and presentation of the real-time text component in multi-party calls
with terminals without multi-party awareness.

The procedures are applicable to scenarios, when the conference focus
and a User Agent have not gone through any successfully completed
negotiation about conference awareness for the real-time text medium
neither on the transport level, nor on the presentation level.

A.3. Definitions

Active participant: Any user sending text, or being in a pending
period.

BOM Byte-Order-Mark, the Unicode character FEFF in UCS-16.

Buffer: A buffer intended for unsent text collected per
participant.

Contributing participants: The participants selected to contribute
to the text stream sent to the recipients.

By default all participants except the recipient are contributing
participants for transmission to the recipient.

Current participant: The participant for whom text currently is
transmitted to the recipient in real time.

Current Recipients: By default all participants.

Display Counter: A counter for the number of displayable
characters in a participant's buffer or in the current entry.
Used for controlling how far erasure may be performed.

Erasure replacement A character to be displayed when an erasure
was done, but the text to erase is not reachable on the multi-
party display. Default 'X'.

Message delimiter: Character(s) forming the end of an imagined
message. A configurable set of alternatives, consisting by
default of: Line Separator, Paragraph Separator, CR, CRLF, LF.

Pending period: A configurable time period of inactivity from a
participant, by default set to 7 seconds after each reception of
characters from that participant, evaluated as current time minus
time stamp of latest entered character.

Sentence delimiter: Characters forming end of sentence: A
configurable set of alternatives, by default consisting of: dot

'.', question mark '?' and exclamation mark '!' followed by a
space.

Label: A readable unique name for a participant, created by the
server from a suitable source related to the participant, e.g.
part of the SIP Display name, surrounded by the Label delimiters.
The label should have a settable maximum length, with 12 being the
default.

Label delimiters A configurable set of characters at the edges of
the Label, by default being a left bracket [at the leading edge
and a closing bracket] followed by a space at the trailing edge.

Line Separator Unicode UCS-16 2028. Used to request NewLine in
Real-Time Text.

Maximum waiting time: The maximum time any participant's text
shall be allowed to wait for transmission, by default set to 20
seconds.

Recipient: The terminal receiving the mixed text stream.

SGR Select Graphic Rendition, a control code to specify colours
etc.

Switch Reason: A set of reasons to switch Current Participant,
consisting of the following

-Waiting time higher for any other participant than the current
participant combined with any of the following states:

-A message delimiter was the latest transmitted item

-A sentence delimiter was the latest transmitted item

-A Pending Period has expired and still no text has been
transmitted

-The Maximum Waiting time has expired followed by a Word Delimiter
or an expired Time Extension.

Waiting time: The time the first character in queue for
transmission from a participant has been waiting in a buffer for
transmission. The granularity shall be 0.3 Seconds or finer.

Word delimiter: Character forming end of word: space

Time extension: A configurable short extension time allowed after
the Maximum waiting time during which a suitable moment for
switching Current Participant is awaited, by default set to 7
seconds.

A.4. Presentation level procedures

The conference server applies these mixing procedures to text
transmitted to all call participants who have not gone through a
completed negotiation for conference awareness in real-time text
presentation.

All the participants and the conference server use real-time text
conversation presentation coding according to ITU-T T.140 [T.140]. A
consequence is that real-time text transmissions are UTF-8 coded,
with control codes selected from ISO 6429 [ISO 6429].

The description is from the conference server point of view.

A.4.1. Structure

The real-time text mixer structure described here is supposed to be
placed in the media path so that it is implemented with one mixer per
recipient. A mixer contains buffers for temporary storage of text
intended for the recipient. Each mixer has one buffer for each
contributing participant. A set of status variables is maintained
per buffer and is used in the mixer actions. The mixer logic decides
for each moment which participant?s buffer content is to be sent on
to the recipient. By default, the recipient does not contribute text
to its own mixer. Text transmitted by a participant is usually
displayed locally and will only cause confusion if it appears also in
received text.

If there is a reason, own text can be configured to be transmitted
also to the participants. That can enable a simplification of the
mixer design to have only one common set of buffers instead of a set
per recipient. That simplification will however hamper the flow of
the conversation severely and is therefore NOT RECOMMENDED.

A.4.2. Action on reception

This description of the mixer is valid per recipient.

Text from each contributing participant is checked for a set of
characteristics on reception.

Delete BOM: BOM characters are deleted.

Insert in buffer: Resulting text is put into the contributing
participant?s buffer in the receiving participant?s mixer.

Maintain a display counter: For each text character that will take
a position on the receiving display, a Display Counter for each
participant is increased by one.

There is one T.140 real-time text item that consists of two
characters, but is regarded to be a unit and therefore increase
the Display Counter with one only.That is CRLF.

Furthermore, the following control codes are regarded units that
shall not take any position on the receiving display and shall
therefore not increase the Display Counter:

0098 string 009C (SOS-ST strings)

ESC 0061 (INT)

009B Ps 006D (the SGR code, with special handling described below)

BEL (Alert in session)

See the section on control codes below for details.

Combination characters: Also note that it is possible to use
combination characters in Unicode. Such combination characters
contain more than one character part. They shall only increase
the Display Counter with one. The combination characters mainly
have components in the series 0300 ? 0361 and 20D0 ? 20E1.

Erasure: If the control code for erasure, BS, is received, the
following shall be done: If the Display Counter is 0, an Erasure
Replacement character, by default being ?X? is inserted in the
buffer instead of the erasure, to mark that erasure was intended
in earlier transmitted entries. (this matches traditional habits
in real-time text when participants sometimes type XXX to indicate
erasure they do not bother to make explicit). If the Display
Counter is >0, then the counter is reduced by one, and the erasure
control code BS put into the buffer.

Initial action in the session: BOM shall be sent initially to the
recipients in the beginning of the session.

Maintaining a waiting time per participant: The time that text has
been in the buffer is maintained as the waiting time for each
buffer. A granularity of 0.3 seconds is sufficient.

Storing time of reception for each character: Each character that is stored in a buffer shall be assigned with a time stamp indicating its time of reception. A granularity of 0.3 seconds is sufficient. This time stamp is used for calculation of idle time and waiting time in the evaluation of switch reasons.

Initial assignment of the Current Participant: The first contributing participant to send text in the session is assigned to be the Current Participant.

Actions on assignment of a Current Participant: When a participant becomes the Current Participant, the following initial actions shall be performed:

1. Scanning transmissions and timers for a Switch Reason is inactivated.

2. The Current Recipients are set so that all transmissions go to the new set of Current Recipients (See definition).

3. A Line Separator is transmitted if the switch reason was any other than a message delimiter.

4. The Label is transmitted

5. Any stored SGR code is transmitted

6. Scanning transmissions and timers for a Switch Reason is activated.

7. Text in the buffer is transmitted, recalculating and setting the waiting time for each transmitted character based on the time of reception of next character in the buffer. If a switch occurs during transmission from the buffer, the remaining buffer contents is maintained and transmission can continue next time this transmitter becomes the current participant. Any text entered into the buffer for the current participant is after that sent to the recipient until a Switch Reason occurs.

Actions on transmission and during the session: Transmissions are checked for control codes to act on at transmission as described below in the section about handling of control codes and such actions are performed. When the scanning of transmission and timers for a Switch Reason is active, the timers and the transmission to the recipient is analyzed for detection if a Switch Reason has occurred. See the definition of Switch Reasons for details.

Actions when a Switch Reason has occurred: If a Switch Reason has occurred, then the following actions shall be performed:

1. The Display Counter of the Current Participant is set to zero

2. If there is an SGR code stored for the Current Participant, a reset of SGR shall be sent by the sequence SGR 0 [009B 0000 006D].

3. A participant with the longest waiting time is assigned to be the Current Participant, and the procedure for assignment of a Current Participant described above is performed.

Handling of Control codes: The following control codes are specified by ITU-T T.140. Some of them require consideration in the conference server. Note that the codes presented here are expressed in UCS-16, while transmission is made in UTF-8 transform of these codes. Other sections specify procedures for handling of specific control codes in the conference server.

BEL 0007 Bell, provides for alerting during an active session.

BS 0008 Back Space, erases the last entered character.

NEW LINE 2028 Line separator.

CR LF 000D 000A A supported, but not preferred way of requesting a new line.

INT ESC 0061 Interrupt (used to initiate mode negotiation procedure).

SGR 009B Ps 006D Select graphic rendition. Ps is rendition parameters specified in ISO 6429.

SOS 0098 Start of string, used as a general protocol element introducer, followed by a maximum 256 bytes string.

ST 009C String terminator, end of SOS string.

ESC 001B Escape - used in control strings.

Byte order mark FEFF Zero width, no break space, used for synchronization.

Missing text mark FFFD Replacement character, marks place in stream of possible text loss.

Code for message border, useful, but not mentioned in T.140: New Message 2029 Paragraph separator

Handling of Graphic Rendition SGR: The following procedure shall be followed in order to let the participants control the graphic rendition of their entries without disturbing other participants? graphic rendition. The text stream sent to a recipient shall be monitored for the SGR sequence. The latest conveyed SGR sequence is also stored as a status variable for the recipient. If the SGR 0 code initiated from the current participant is transmitted, the SGR storage shall be cleared.

A.5. Display examples

The following pictures are examples of the view on a participant's display.

```
 _____
|        Conference       |         Alice          |
|_____|_____|
|                         |I will arrive by TGV.   |
|[Bob]:My flight is to    |Convenient to the main  |
|Orly.                    |station.                |
|[Eve]:Hi all, can we     |                        |
|plan for the seminar.    |                        |
|                         |                        |
|[Bob]:Eve, will you do   |                        |
|your presentation on     |                        |
|Friday?                  |                        |
|[Eve]:Yes, Friday at 10.|                        |
|[Bob]: Fine, wo          |We need to meet befo    |
|_____|_____|
```

Figure 2 : Alice who has a conference-unaware client is receiving the multi-party real-time text in a single-stream. This figure shows how a coordinated column view MAY be presented on Alice's device.

```
       _____
      |                                              |^|
      |[Alice] Hi, Alice here.                       | |
      |                                              | |
      |[Bob] Bob as well.                            | |
      |                                              | |
      |[Eve] Hi, this is Eve, calling from Paris.    | |
      |      I thought you should be here.           | |
      |                                              | |
      |[Alice] I am coming on Thursday, my           | |
      |      performance is not until Friday morning.| |
      |                                              | |
      |[Bob] And I on Wednesday evening.             | |
      |                                              | |
      |[Eve] we can have dinner and then take a walk | |
      |                                              | |
      | [Eve-typing] But I need to be back to        | |
      |    the hotel by 11 because I need            |-|
      |                                              |-|
      |_____|v|
      | of course, I underst                           |
      |_____|
```

   Figure 3 shows a conference view with real-time text preview. Bob?s
   text is buffering until a Current switch reason.

A.6. Summary of configurable parameters

   A number of configurable parameters are described in this
   specification. This table provides a summary of the parameters on
   presentation level. A service provider implementing a multi-party
   service may want to set specific values on these parameters to adapt
   the characteristics of the service. It is possible to control them
   per recipient, if desired.

   Parameter: Current Recipients

   Purpose: Control if participant shall get their own text.

   Possible values: Exclude or Include Current Participant

   Default value: Exclude

   Comment: Own transmissions are usually displayed sufficiently locally

   Parameter: Erasure replacement

   Purpose: Character to show erasure, when erasure cannot be done

Possible values: Character

Default value: X

Comment: May need to have other value for other than Latin script.

Parameter: Message delimiter

Purpose: Detection of suitable place in text for switching Current
Participant

Possible values: List of Unicode editing codes

Default value: Line Separator, Paragraph Separator, CR, CRLF, LF

Comment: Other than Latin based scripts may have other conventions

Parameter: Pending period

Purpose: Inactivity timer for detection of time to Switch Current
Participant

Possible values: Time in seconds

Default value: 7

Comment: Longer times may cause inefficient transmission. Shorter
time may cause unwanted switching cutting lines of thought
inconveniently

Parameter: Sentence delimiter

Purpose: Characters forming end of sentence

Possible values: List of delimiters.

Default value: . or ? or ! followed by a space

Comment: Used for deciding on a position in the text to switch
Current Participant according to configured logic.

Parameter: Label length

Purpose: Length of label put in front of or above entry.

Possible values: Number of characters

Default value: 12

Comment: Includes any surrounding characters

Parameter: Label delimiters

Purpose: Set of characters at the edges of the label

Possible values: Two strings. One in the beginning, one after.

Default value: [] followed by a space

Comment: It may be valid to include a Line Separator instead of the space

Parameter: Maximum waiting time

Purpose: The maximum time any participant?s text shall be allowed to wait for transmission

Possible values: Seconds

Default value: 20

Comment After this time a Switch will be forced within the Time Extension

Parameter: Word delimiter

Purpose: Delimiter for words

Possible values: List of characters

Default value: Space

Comment: Used for detection of suitable switch position if Maximum Waiting time has passed.

Parameter: Time extension

Purpose: Time for maximum further waiting for a Switch Reason

Possible values: Time in seconds

Default value: 7

Comment: After this time a Switch is forced.

A.7. References for this Appendix

    [T.140] ITU-T T.140 Application protocol, text conversation
    (including amendment 1.)

    [RFC 4103] IETF RFC 4103 RTP Payload for text conversation

    [RTP] IETF RFC 3550 RTP: A Transport Protocol for Real-Time
    Applications.

    [RFC 4579] IETF RFC 4579 SIP Call Control ? Conferencing for user
    agents.

    [ISO 6429] ISO 6429 Control functions for coded character sets.

    [UTF-8] IETF RFC 3629 UTF-8, a transformation format of ISO 10646

    [Unicode] The Unicode Consortium, "The Unicode Standard ? Version
    4.0?

    [ISO 10?646-1] ISO 10?646 Universal multiple-octet coded character
    set (UCS)

    [UCS-16] See ISO 10?646-1

A.8. Acknowledgement

   This appendix was developed with funding in part from the National
   Institute on Disability and Rehabilitation Research, U.S. Department
   of Education,RERC on Telecommunications Access,?grant # H133E090001?.
   However, the contents do not necessarily represent the policy of the
   Department of Education, and you should not assume endorsement by the
   Federal Government.

Author's Address

   Gunnar Hellstrom
   Omnitor
   Esplanaden 30
   Vendelso SE-136 70
   SE

   Phone: +46 708 204 288
   Email: gunnar.hellstrom@omnitor.se
   URI: www.omnitor.se