Technical specification: Reliable RTT (Real-Time Text)

Abstract

This document describes a private Session Initiation Protocol (SIP) [IETF RFC 3261] header (P-header), used to achieve real-time sending of text using SIP Message requests, along with its applicability.

The formatting of the real-time text (RTT) messages is according to ITU-T Recommendation T.140.

This Technical Specification for RTT exchange between ICT systems (including networks, devices, services and protocols) is made available by the Swedish Post and Telecom Authority (PTS), www.pts.se. It is made freely available and without any claims on Intellectual Property Rights (IPRs), providing a common specification for RTT exchange. Its most recent version, approved by and published on PTS' homepage always supersedes any other, previous versions.

Table of Contents

1. Introduction and scope

This Technical Specification defines a way to provide reliable, real-time text, character-by-character conversation functionality using SIP MESSAGE [IETF RFC 3428] requests.
The purpose is to present the text typed by the sender as soon as possible on the device of the receiving party (unless otherwise preferred by the sender - e.g. an individual preference for word completion through predictive text input).
The method is typically implemented in the user agent client (UAC) by sending a MESSAGE request as soon as possible, after the detection of user input.
The difference from the standard behaviour is that the UAC doesn't wait for the user to hit the return key before sending, instead every typed character can be sent individually or as a group of characters.
The P-Header defined in this document is used to facilitate a simple negotiation procedure, where the UAC and user agent server (UAS) indicates support for this method of sending text by including the header.

2. Overall applicability

The SIP extension specified in this document makes no assumptions regarding the surrounding environment it operates in, except a SIP compliant UAC and UAS.
A UAS not implementing the mechanism described in this document can safely ignore the P-Header described in the document.

3. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [IETF RFC 2119].

4. Overview and advantage

This document describes a way to address the requirement to let characters appear in real-time as they are typed, instead of as a block of text after it is written (as typical IM- or SMS-applications would do), using only the SIP protocol. A private extension header is used to mimic this behaviour, using plain SIP Message requests.
The obvious advantage of this method is its simplicity. Since this method uses the SIP signalling path, instead of the speech path to transfer text, it also achieves a higher degree of reliability.

5. SIP Private Header

5.1. The P-Safe-Text header

The purpose of the P-Safe-Text header field is to provide an indication from a SIP UAC that it supports this way of sending text in real-time.

If used, the header MUST be present in the SIP INVITE request. It is also used for a SIP UAS to confirm the support for sending text this way. If used, this header field MUST occur in a final 200 OK Response.

5.1.1. P-Safe-Text Header Syntax

The syntax of the P-Safe-Text header is described as an augmented [IETF RFC 4234] Backus-Naur Form (BNF):

```
P-Safe-Text = "P-Safe-Text" HCOLON seqno
Seqno= 1*DIGIT
HCOLON is defined in [RFC3261]
DIGIT is defined in [RFC5234]
The parameter seqno SHALL be coded as a decimal value of an unsigned
integer.
```

Note that the P-Safe-Text header is not registered by IANA, and therefore no guarantee can be provided that other systems do not use the same name on a field for other purposes, causing a risk for interoperability problems.

6. SIP session establishment and modification

If the method for transmission of real-time text described in this specification is to be used in a SIP session, it must first be selected through a completed capability exchange during SIP session establishment or modification.

The P-Safe-Text header SHALL be included in the SIP transaction that carries the offer of the session by a UA which is prepared to use the method. The "seqno" parameter SHALL be set by the UA to an initial value one lower than the sequence number of the first real-time text fragment in a sequence that may be transmitted by the UA in the session.

Following such offer, if an answering UA selects to use this method for real-time text, it SHALL acknowledge the selection of the method by including the P-Safe-Text header in the SIP transaction carrying the answer. The "seqno" parameter SHALL be set by the answering UA to an initial value one lower than the sequence number of the first real-time text fragment in a sequence that may be transmitted by the answering UA in the session.

Real-time text use in a session MAY be added deleted or modified through a reINVITE transaction. The seqno parameters exchanged in a reINVITE transaction SHALL be used as initial values for the real-time text transmissions.

7. Sending and receiving SIP Messages

The UAC MUST send SIP MESSAGE Requests in an established SIP dialogue, to ensure that the sent messages gets delivered to the correct UAS.

7.1. Frequency and transport of SIP MESSAGE

The UAC MUST never send more than 4 SIP MESSAGE requests per second and there MUST be at least 250ms between each SIP MESSAGE Requests sent. The UAS can send SIP MESSAGE Request using UDP or TCP following the procedures in RFC 3263 [RFC3263]. To avoid congestion, the UAC MUST wait for a SIP MESSAGE Response before sending the next SIP MESSAGE Request and the UAC SHOULD send SIP MESSAGE Request using TCP.

7.2. Content Type and character encoding

A SIP UAC utilizing the Reliable RTT way of sending a SIP Message Request MUST specify the Content Type Header to use media type 'text' and media sub-type 'plain'. The parameter “charset=UTF-8” SHALL be included. Contents

MUST be UTF-8 encoded. A SIP UAS receiving a SIP Message Request with this content type SHALL treat the text as entered character-by-character and SHALL NOT add any trailing CR-LF or other delimiter when displaying the text.

7.3. Content

The UAC SHOULD in each SIP MESSAGE include all characters created for transmission since the latest SIP MESSAGE request was sent. Consideration of the MTU for the current network and IP protocol used MUST however be taken into consideration.
The content SHALL follow the specification in ITU-T T.140, including Addendum 1 [ITU.T140.1998] regarding both text and control codes.
Receivers MUST be prepared to receive any text or control codes according to T.140 and act on the contents or ignore it according to what is specified in T.140. Note the preferred coding of new line and that erasure of new line SHALL be done with only one erasure action regardless of how the new line was coded in transmission.
The scope of erasure actions SHALL not be limited by new line or any other delimiters.

7.4. Actions against loss or duplication of text by the transmitter

The P-Safe-Text header SHALL be included in each SIP MESSAGE request sent by the method specified in the present document.
The value of the "seqno" parameter SHALL be incremented by one before the transmission.
In case of transmission failure indicated by a SIP final response, a retry MUST NOT be requested. Sufficient retries have normally already been performed by SIP procedures.

7.5. Actions against loss and duplication of text by the receiver

Each successfully received SIP MESSAGE is interrogated for presence of the P-Safe-Text field.
When present, the rules specified in this specification are applied.
There is no need to include the P-Safe-Text field in the response on the SIP MESSAGE.
The received "seqno" parameter value is compared to the previously received value.
If the value has been incremented by one since the previous received value, then no loss has occurred and the content can be presented without further actions.
If the value has been increased by more than one since the previous received value, loss of reception has occurred and it is likely that text is lost. The UTF-8 replacement character specified in T.140 Addendum 1 (a question mark in a rhombus) SHALL be inserted in the received text stream. When the replacement character is presented, it is allowed to replace it with an apostrophe " ' " if the display cannot represent the replacement character.

8. Interoperability

When a UA implements more methods for real-time text than the method specified in the present document, the following four procedures SHALL be applied:

1. If no other agreement has been made about the intention to use the real-time text medium, it SHALL be assumed that both real-time text methods are intended for the same purpose, in the same text conversation session. Thus for that case, a text method supported by

both Reliable RTT and the other method SHALL be selected by a session establishment or modification.

2. If more than one text stream is about to be established in the same session by another method than Reliable RTT, then that method SHOULD be selected for all text streams of that session.

3. Both the P-SafeText header and the capability indicator of another real-time text method SHOULD be present in the same session offer.

4. The method used MAY be re-negotiated, and RTT streams may be added by reINVITE during the session.

9. Miscellaneous

The UAC SHOULD use SIP via TCP when sending text to ensure the best transport reliability. Reliable RTT is fully compatible with [IETF RFC 3428], "SIP chat". The UAC can therefore in the same SIP dialogue use messages with Reliable RTT header mixed with messages without Reliable RTT header. This enables Reliable RTT interoperability with any SIP chat based system, also supporting use from mobile devices.

10. IANA Considerations

This Technical Specification includes no requests to the Internet Assigned Numbers Authority,IANA (see http://www.iana.org).

11. References

All [RFC(number)] references have been issued by the Internet Engineering Task Force (IETF). All IETF RFCs are freely available at www.ietf.org.

11.1. Normative references

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

[RFC3261]Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[RFC3263] Rosenberg, J. and H. Schulzrinne, "SIP: Locating SIP Servers", RFC 3263, June 2002

[RFC4234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 4234, October 2005.

[RFC3428] Campbell, B., Rosenberg, J., Schulzrinne, H.,Huitema, C., and D. Gurle, "Session Initiation Protocol (SIP) Extension for Instant Messaging", RFC 3428, December 2002.

[RFC5234] Crocker, D., Ed., and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.

[ITU.T140.1998] "Protocol for Multimedia Application Text Conversation", ITU-T Recommendation T.140, February 1998.

Appendix A (informative): SIP examples

A.1 Example of SIP Invite Request from “Bob” to “Alice” containing a P-Safe-Text header

```
INVITE sip:bob@biloxi.com SIP/2.0
Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bK776
Max-Forwards: 70
To: Bob <sip:bob@biloxi.com>
From: Alice <;sip:alice@atlanta.com>;tag=4234876723
Call-ID: a84b4c76e66710@pc33.atlanta.com
CSeq: 1 INVITE
Contact: <sip:alice@pc33.atlanta.com>
P-Safe-Text: 0
Content-Type: application/sdp
Content-Length: 134

----SDP----
```

A.2 Example of SIP 200 OK Response from “Alice” to an Invite Request, containing a PSafe-Text header

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bK776 ; received=192.0.2.1
To: Bob <sip:bob@biloxi.com>;tag=a6werc85cf
From: Alice <sip:alice@atlanta.com>;tag=4234876723
Call-ID: a84b4c76e66710@pc33.atlanta.com
CSeq: 1 INVITE
Contact: <sip:bob@192.0.2.4>
P-Safe-Text: 0
Content-Type: application/sdp
Content-Length: 123

----SDP----
```

A.3 Example of a sequence of SIP Message Requests in the established session with Content-Type-Header set to 'text/plain' containing the text “Hello” sent in real-time text fashion from “Bob” to “Alice”

A.3.1 A MESSAGE containing the real-time text sample “He” from “Bob”

```
MESSAGE sip:bob@biloxi.com SIP/2.0
Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bK776
Max-Forwards: 70
To: Bob <sip:bob@biloxi.com>;tag=a6werc85cf
From: Alice <sip:alice@atlanta.com>;tag=4234876723
Call-ID: a84b4c76e66710@pc33.atlanta.com
CSeq: 3 MESSAGE
Contact: <sip:alice@pc33.atlanta.com>
P-Safe-Text: 1
Content-Type: text/plain;charset=UTF-8
Content-Length: 2

He
```

A.3.2 Acknowledgement of the first text segment by “202 Accepted” from “Alice”

```
SIP/2.0 202 Accepted
Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bK776 ;received=192.0.2.1
To: Bob <sip:bob@biloxi.com>;tag=a6werc85cf
```

```
From: Alice <sip:alice@atlanta.com>;tag=4234876723
Call-ID: a84b4c76e66710@pc33.atlanta.com
CSeq: 3 MESSAGE
Contact: <sip:bob@192.0.2.4>
```

A.3.3 Next MESSAGE containing the real-time text fragment “llo” from “Bob” to “Alice”

```
MESSAGE sip:bob@biloxi.com SIP/2.0
Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bK776
Max-Forwards: 70
To: Bob <sip:bob@biloxi.com>;tag=a6werc85cf
From: Alice <sip:alice@atlanta.com>;tag=4234876723
Call-ID: a84b4c76e66710@pc33.atlanta.com
CSeq: 4 MESSAGE
Contact: <sip:alice@pc33.atlanta.com>
P-Safe-Text: 2
Content-Type: text/plain;charset=UTF-8
Content-Length: 3

llo
```

A.3.4 Acknowledgement of the second text segment by “202 Accepted”

```
SIP/2.0 202 Accepted
Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bK776 ;received=192.0.2.1
To: Bob <sip:bob@biloxi.com>;tag=a6werc85cf
From: Alice <sip:alice@atlanta.com>;tag=4234876723
Call-ID: a84b4c76e66710@pc33.atlanta.com
CSeq: 4 MESSAGE
Contact: <sip:bob@192.0.2.4>
```