Multi-party real-time text

status update for IETF AVTCORE 108 July 30 2020

Real-time text solutions for multi-party sessions
draft-hellstrom-avtcore-multi-party-rtt-solutions-02

RTP-mixer formatting of multi-party Real-time text
draft-ietf-avtcore-multi-party-rtt-mix-07

Gunnar Hellström, GHAccess
gunnar.hellstrom@ghaccess.se

Requirements for multi-party real-time text

- Real-time text is text transmitted while it is created. Rapid, No waiting.
- Enables smooth conversation, just like voice and video.
- Useful for all. Enforced by regulation for accessibility reasons.
- Multi-party was envisioned from the beginning in presentation standard ITU-T T.140
- But it was not clearly explained how, when specified for SIP in RFC 4103.
- Urgent need for implementing in NG9-1-1 emergency services in North America.
- Required by European Accessibility Act.
- Main current implementation technologies needing upgrade are 3GPP IMS, NG emergency services, VRS
- Specified now with goal to be easily implemented in existing technologies.

draft-hellstrom-avtcore-multi-party-rtt-solutions-02

- Informational, used as background for specification of the standards track draft
- Requirements.
- Transport solutions with pros and cons
  - Single stream RTP-based. text/t140, text/red and new redundant multi-party format text/rex
  - Multi-stream RTP text/t140 and text/red
  - WebRTC t140 data channel
- Session control solutions
  - SDP
  - Centralized SIP conference
- Negotiation solutions with pros and cons
  - SDP attribute
  - FMTP parameter
  - Media tag
- Presentation aspects
- Can be used by developers and in future work

RTP-mixer formatting of multi-party Real-time text
draft-ietf-avtcore-multi-party-rtt-mix-07

- Standards track
- Main focus on RTP-based solutions for centrally controlled SIP conferences
- Brief info on WebRTC, gateways, security....
- Good text presentation requires mult-party aware actions by the receiver
- Now contains three RTP mixing solutions. May be decreased to two.
- #1. Mixing for multi-party unaware endpoints. Low functionality fallback.
- #2. Mixing of RFC 4103 text/red format, one source per packet in CSRC.
- #3. Mixing with new "text/rex" format, with many sources per packet.
- Recent move from high performance solutions (#3 and multi-stream RTP) to rapid and easy implementation in current technologies (#2)

Balance between performance and ease of implementation

Key performance figure: Introduced mean jerkiness in ms at specified number of simultaneously sending participants. 1 second is accepted in 2-party calls (F.700) Likely the same for multi-party. Example performance per solution:

#1: Varying, usually many seconds, depends on currently presented source indicating suitable place in text for switch (New line, full stop, long pause ...)

#2: 150 ms at 2 simultaneously sending sources, 300 ms at 3 sources. 750 ms at 4 sources

#3: 150 ms at 2 - 15 simultaneously sending sources.

RTP multi-stream: 150 ms at 2-5 simultaneously sending sources. Avoid more!

How many user can be expected to send simultaneously?

- Similiar expectations as for voice in conference: Usually just one source sending.
- Occasionally a second source sending very brief comment or asking for the floor: "Yes", "No", "I agree", "I want to comment"
- Possibly sometimes also the second source sending a bit longer info when getting anxious, e.g. in emergency calls: "Yes, I will clarify but please send an ambulance NOW!"
- One application can be with a language translation service, but then one of the links to the service is usually voice, while the other may be in real-time text.
- Very rarely more than these two simultaneously sending parties.
- Conclusion: Good performance at 3 simultaneously sending sources is sufficient.

Ease of implementation

#1: A bit complex for the mixer. (But needed for interop with current base)

#2: Easy for the mixer. Easy for current endpoints. Same packet format as for RFC 4103. Just added sdp attribute negotiation and sorting received text per CSRC source.

#3: New alternative packet format "text/rex". More complex for mixer. More complex sdp negotiation. More complex for the endpoint. Likely longer time than acceptable to get added to 3GPP and GSMA specification releases.

Advice wanted: Keep 3 solutions or reduce to 2?

- Solution #1 needed for current endpoints
- But #2 and #3 may seem to be overkill to keep both in current draft.
- #2 seems needed for its timeline, and sufficient in performance.
- #3 seems not needed, just good to have for performance beyond requirements.
- Proposal: Move on with just #1 for multi-party unaware enpoints and #2 with RFC 4103 for multi-party negotiated by a=rtt-mix-rtp-mixer.
- Future use may also take other directions: E.g. more use of T140 in WebRTC data channel briefly still mentioned in the draft.

Thanks

Comments are welcome

Gunnar Hellstrom

gunnar.hellstrom@ghaccess.se