



ERICSSON

WebRTC: RTP Usage WG Last Call Comments

draft-ietf-rtcweb-rtp-usage-14
Magnus Westerlund

Thanks



- › The authors likes to thank everyone that reviewed!
 - Olle E. Johansson
 - Jim Spring
 - Bernard Aboba
 - Suhas Nandakumar
 - Alex Eleftheriadis
 - Cullen Jennings
 - Roni Even
 - Ben Campbell

Outline



- › New Draft Version
 - Changes Introduced
- › Issues
 - Multi-Stream Optimisation
 - RTP Retransmission
 - Forward Error Correction (FEC)
 - Circuit Breakers Mandated
 - Mandate Signalling
 - Picture Loss Indication (PLI)
- › Clarification
 - Why T-rr-int provide best RTP/(S)AVP interop

Changes Proposed



- › A large number of editorials
 - Including reformulating so that SDP is not required
- › Section 4.1:
 - Replace RFC 2119 language with references to other sections of the draft where appropriate
 - Add references to where certain functionalities are defined
 - Reduced minimum RTCP reporting interval
($T_{min} = 360 / \text{Session BW in kbps}$)
 - › Changed from RECOMMENDED to REQUIRED
 - › Note that timeout is done on fixed (not the reduced minimum)
 - This is following RFC 3550

Changes Proposed



› Section 4.1:

– Clarified the requirement to ignore unknown RTCP packet types and header extensions:

- › “Ignore unknown RTCP packet types and RTP header extensions. This to ensure robust handling of future extensions, middlebox behaviours, etc., that can result in not signalled RTCP packet types or RTP header extensions being received. If a compound RTCP packet is received that contains a mixture of known and unknown RTCP packet types, the known packets types need to be processed as usual, with only the unknown packet types being discarded.”

Changes Proposed



› Section 4.3:

– An end-point that has signalled support for multiple RTP payload formats **MUST** be able to accept data in any of those payload formats at any time, unless it has previously signalled limitations on its decoding capability.

› Changed the **SHOULD** -> **MUST**

› Section 4.5:

– Clarified that RTP/RTCP MUX is

› **REQUIRED** to implement

› Used when negotiated in signalling

› Section 4.6:

– Same as above for Reduced Size RTCP

Changes Proposed



- › Section 5.2.2 (Reworded sentence):
 - The use of this encryption is RECOMMENDED, however usage of the encryption can be explicitly disabled through API or signalling.
- › Section 7.1, Last paragraph:
 - Proposal:
 - › Signalled bandwidth limitations, in case of SDP "b=AS:" or "b=CT:" from peer, MUST be followed when sending RTP packet streams. A WebRTC endpoint receiving media SHOULD signal its bandwidth limitations, these limitations have to be based on known bandwidth limitations, for example the capacity of the edge links when known.

Changes Proposed



- › Removed Previous Section 7.2:
 - RTCP Limitations for Congestion Control
 - Had turned into more guidance for designers of congestion control
 - Colin will provide this information to RMCAT in
 - › [draft-perkins-rmcat-rtp-cc-feedback](#)

Open Issue: Multi-Stream Optimisation



› Cullen on Section 4.1:

- [I-D.ietf-avtcore-rtp-multi-stream-optimisation] is RECOMMENDED.
- Cullen proposed MAY
- Authors suggestion is to leave as is:
 - › Provides significant benefit in multi SSRC sessions
 - › However, should clarify that it requires mutual support to use
 - › Table show reduction in average reporting interval in % compared to basic AVPF

SSRCs	1	2	3	4	5	8	12	16
Aggregation	-0.33	-1.50	-2.14	-2.38	-3.33	-8.3	-3.39	0.48
Report Groups	0.28	-2.23	-7.93	-15.27	-21.94	-43.3	-67.6	-78.9
Aggregation + Report Groups	0.89	-3.96	-10.2	-17.7	-25.8	-47.0	-70.4	-84.7

Open Issue: RTP Retransmission



- › What the draft says is (Section 6.1):
 - Receivers are REQUIRED to implement support for RTP retransmission packets [RFC4588]. Senders MAY send RTP retransmission packets in response to NACKs if the RTP retransmission payload format has been negotiated for the session, and if the sender believes it is useful to send a retransmission of the packet(s) referenced in the NACK. An RTP sender does not need to retransmit every NACKed packet.
- › This was discussed at the Stockholm Interim in June 2012
 - Followed by a call for input on the list
 - Authors implemented their view of the rough consensus
 - Text has been present since July 2012 (version -04)
- › Proposal from Authors: No change

Open Issue: FEC



- › The draft currently says:
 - There are several block-based FEC schemes that are designed for use with RTP independent of the chosen RTP payload format. At the time of this writing there is no consensus on which, if any, of these FEC schemes is appropriate for use in the WebRTC context. Accordingly, this memo makes no recommendation on the choice of block-based FEC for WebRTC use.
- › This has been the state of the draft for quite some time
 - Last time it was raised as potential issue prior to the WGLC was in August last year by Bernard Aboba.
 - No one have made any proposal for what FEC should be recommended or MTI

Open Issue: FEC



- › If you want a FEC mechanism as recommended or MTI, please propose one!
 - RFC 2733/5109/6015/SMPTE 2022-1 (The XOR bases ones):
 - › Does not work with single RTP session (i.e. Full Bundle) unless:
 - extensions required: draft-lennox-payload-ulp-ssrc-mux-00
 - Use RFC 2198 (Audio: ok, Video: Inefficient and interoperability?)
 - › Use separate RTP session for the FEC RTP streams
 - Raptor over RTP (RFC 6682)
 - Other Block FEC algorithms lacks RTP specification
 - › Reed-Solomon
 - › LDPC
 - › Raptor-Q
 - Please consider IPR for all of these

Open Issue: Circuit Breakers



- › Currently mandates **draft-ietf-avtcore-rtp-circuit-breakers-05**
 - Even when proprietary congestion control are implemented and used
- › Intention that Circuit breaker will provide a fail safe to protect any network against persistent congestion
 - A mechanism that would allow us to publish RTCWEB media specification prior to RMCAT finishing their work
 - Transport folks don't like multi Mbps transmitting endpoints without any type of congestion control on the Internet

Open Issue: Circuit Breakers



- › Worries raised that it will trigger unnecessary and stop the session
 - Requiring user restart, or
 - Reducing bit-rate with a least a factor of 10
- › However there has been evaluation results presented:
 - In Berlin AVTCORE: Results for ADSL links
 - In Vancouver: LTE Evaluation
 - Circuit breaker draft references academic papers
 - › Some also available here:
<http://www.csperkins.org/publications/index.html>
- › Circuit breaker work in AVTCORE have not concluded
 - If you have worries contribute there!

Open Issue: Circuit Breakers



- › Circuit Breaker triggers when:
 1. Media Timeout, i.e. broken forward path no RTP packets received
 2. RTCP Timeout, i.e. no RTCP reports received from receiver
 3. Congestion, i.e. Loss rate => in TCP would get less than 1/10 over same path
 4. Media Usability, i.e. the user of the RTP packets gets no utility out of the packets
- › Triggers after 2 or 3 consecutive regular reporting intervals, i.e. ~ 5-18 seconds
- › When RMCAT is done what is mandated in WebRTC will be revisited
 - Potential to replace circuit breakers then!
- › Author's Proposal: No change, contribute to circuit breaker

Open Issue: Mandate Signalling



- › Cullen raised that he like to require signalling of extensions before usage:
 - Some Extensions require negotiation prior to use
 - › e.g. RTP/RTCP multiplexing
 - Other are RTP/RTCP extensions that provide functionality but doesn't cause interoperability failures:
 - › Draft currently says SHOULD signal in these cases
 - Motivation:
 - › Legacy, middleboxes or other gateways that don't remove what has not been signalled
 - › Postel's law - "Be conservative in what you do, be liberal in what you accept from others"
 - › Robust endpoints do ignore unknown extensions
 - Agree that signalling is encouraged and expected,
 - › However, end-points must cleanly ignore unsignalled extensions if they occur

Open Issue: Picture Loss Indication (PLI)



- › Suhas said:
 - "Should we add the requirement for the WebRTC end-point sending PLI messages?"
- › Unclear what is proposed here?
- › Current implementation requirements are:
 - WebRTC senders **MUST** understand and react to PLI feedback messages as a loss tolerance mechanism.
 - Receivers **MAY** send PLI messages.
- › Author's Proposal is to not change anything

Clarification: RTP/SAVP compatibility



- › Suhas asked about the motivation for why $t\text{-rr-int} = 4$ was said to provide maximum interoperability between RTP/SAVP and RTP/SAVPF

