Virtual Interim Remote Meeting Tips

- Enter the queue with 👋, leave with 👋
- When you are called on, you need to enable your audio to be heard.
- Audio is enabled by unmuting 📈 and disabled by muting 🎤
- Video can also be enabled, but it is separate from audio.
- Video is encouraged to help comprehension but not required.
- Keep audio and video off unless you are chairing or presenting.
- Use of a headset is strongly recommended.
Note well

This is a reminder of IETF policies in effect on various topics such as patents or code of conduct. It is only meant to point you in the right direction. Exceptions may apply. The IETF’s patent policy and the definition of an IETF "contribution" and "participation" are set forth in BCP 79; please read it carefully.

As a reminder:

- By participating in the IETF, you agree to follow IETF processes and policies.
- If you are aware that any IETF contribution is covered by patents or patent applications that are owned or controlled by you or your sponsor, you must disclose that fact, or not participate in the discussion.
- As a participant in or attendee to any IETF activity you acknowledge that written, audio, video, and photographic records of meetings may be made public.
- Personal information that you provide to IETF will be handled in accordance with the IETF Privacy Statement.
- As a participant or attendee, you agree to work respectfully with other participants; please contact the ombudsteam (https://www.ietf.org/contact/ombudsteam/) if you have questions or concerns about this.

Definitive information is in the documents listed below and other IETF BCPs. For advice, please talk to WG chairs or ADs:

- **BCP 9** (Internet Standards Process)
- **BCP 25** (Working Group processes)
- **BCP 25** (Anti-Harassment Procedures)
- **BCP 54** (Code of Conduct)
- **BCP 78** (Copyright)
- **BCP 79** (Patents, Participation)
- [https://www.ietf.org/privacy-policy/](https://www.ietf.org/privacy-policy/) (Privacy Policy)
IETF meetings, virtual meetings, and mailing lists are intended for professional collaboration and networking, as defined in the [IETF Guidelines for Conduct](https://tools.ietf.org/html/rfc7154), the [IETF Anti-Harassment Policy](https://tools.ietf.org/html/rfc6941), and the [IETF Anti-Harassment Procedures](https://tools.ietf.org/html/rfc7776). If you have any concerns about observed behavior, please talk to the Ombudsteam, who are available if you need to confidentially raise concerns about harassment or other conduct in the IETF.

The IETF strives to create and maintain an environment in which people of many different backgrounds are treated with dignity, decency, and respect. Those who participate in the IETF are expected to behave according to professional standards and demonstrate appropriate workplace behavior.

IETF participants must not engage in harassment while at IETF meetings, virtual meetings, social events, or on mailing lists. Harassment is unwelcome hostile or intimidating behavior -- in particular, speech or behavior that is aggressive or intimidates.

If you believe you have been harassed, notice that someone else is being harassed, or have any other concerns, you are encouraged to raise your concern in confidence with one of the Ombudspersons.
About this meeting

- Meeting info: https://datatracker.ietf.org/meeting/interim-2023-avtcore-03/session/avtcore
- Notes: https://notes.ietf.org/notes-ietf-interim-2023-avtcore-03-avtcore
- Secretariat: mtd@jabber.ietf.org
- WG Chairs: Jonathan Lennox & Bernard Aboba
- Zulip Scribe: Jonathan Lennox
- Note takers: TBD
Draft Status

- Published
  - RFC 9071: was draft-ietf-avtcore-multi-party-rtt-mix
  - RFC 9134: was draft-ietf-payload-rtp-jpegxs
  - RFC 9328: was draft-ietf-avtcore-rtp-vvc
  - RFC 9335: was draft-ietf-avtcore-cryptex
  - RFC 9443: was draft-ietf-avtcore-rfc7983bis

- RFC Editor Queue
  - draft-ietf-payload-vp9 (837 days, MISSREF)

- IESG: AD Followup (272 days, 3 DISCUSS positions)
  - draft-ietf-avtcore-rtp-scip
  - Ballot statements: https://datatracker.ietf.org/doc/draft-ietf-avtcore-rtp-scip/ballot/
    - More discussion today.

- Waiting for AD Go-Ahead::Revised I-D Needed (61 days)
  - draft-ietf-avtext-framemarking

- In IETF Last Call (Until October 4, 2023)
  - draft-ietf-avtcore-rtp-evc
Draft Status (cont’d)

- **Adopted**
  - draft-ietf-avtcore-rtp-over-quic
  - draft-ietf-avtcore-rtp-green-metadata
  - draft-ietf-avtcore-rtp-v3c

- **In call for adoption**
  - draft-aboba-avtcore-hevc-webrtc (completed on September 5, 2023)
IANA Registries

● A post to the W3C public-webrtc mailing list pointed out an issue with IANA RTP payload format type registrations: https://lists.w3.org/Archives/Public/public-webrtc/2023Aug/0033.html
  ○ RTP payload types registry is missing VP8, AV1, HEVC, VVC: https://www.iana.org/assignments/rtp-parameters/rtp-parameters.xhtml#rtp-parameters-2

● IANA mime-types registry (see “video”) is more complete: https://www.iana.org/assignments/media-types/media-types.xhtml

● Questions:
  ● Is this an issue?
  ● If so, what should we do about it?
Action items from IETF 117

- Working Group Last calls
  - V3C RTP payload format to be started
    - Open source implementation available
    - https://github.com/nokiatech/v3crtp
- Calls for adoption
  - SFrame RTP payload?
1. Note Well, Note Takers, Agenda Bashing, Draft status, IANA Registries (Chairs, 15 min)
2. RTP Payload Format for sub-codestream J2K streaming (P. Lemieux, 15 min)
3. HEVC Profile for WebRTC (B. Aboba, 15 min)
4. RTP Payload Format for SCIP (M. Faller, 15 min)
5. RTP over QUIC (M. Engelbart, J. Ott, S. Dawkins, 30 min)
6. RTP Payload format for SFrame (P. Thatcher, 15 min)
   draft-thatcher-avtcore-rtp-sframe
7. Wrapup and Next Steps (Chairs, 15 min)
High-performance JPEG 2000 RTP payload format
draft-lemieux-avtcore-rtp-j2k-scl
Dr. Pierre-Anthony Lemieux, Sandflow Consulting
pal@sandflow.com

Dr. David Taubman, UNSW
d.taubman@unsw.edu.au
JPEG 2000

ITU-T | ISO/IEC standard since 2001

Swiss army knife of image codecs
- Lossy to lossless coding
- Resolution and quality scalability
- Non-iterative optimal rate control
- High-dynamic range, multi-channel and sub-sampled images

Royalty-free

Widely used in high-performance applications (image and video)

Continuously maintained and improved

Medical
Geospatial
Archival
Cinema
Studio post-production
TV production over IP
Part 15: High-Throughput JPEG 2000 (HTJ2K)

New part of the JPEG 2000 family (published in 2019)
Opens the door to inexpensive and high-quality streaming

Decoding speed-up compared to legacy J2K

<table>
<thead>
<tr>
<th>1 bpp</th>
<th>2 bpp</th>
<th>4 bpp</th>
<th>Lossless</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.7x</td>
<td>8x</td>
<td>12x</td>
<td>28x</td>
</tr>
</tbody>
</table>

Coding efficiency loss compared to legacy J2K

<table>
<thead>
<tr>
<th>1 bpp</th>
<th>2 bpp</th>
<th>4 bpp</th>
<th>Lossless</th>
</tr>
</thead>
<tbody>
<tr>
<td>6%</td>
<td>6%</td>
<td>6%</td>
<td>8%</td>
</tr>
</tbody>
</table>
High-performance RTP payload for JPEG 2000

Intended for streaming of video signals encoded as a sequence of JPEG 2000 codestreams

Significant improvements over RFC 5371

- Sub-codestream latency
- Resolution and quality scalability at the RTP packet level
- Code-block caching for screen content
- High-precision sender clock recovery
- Progressive-segmented frame (PsF) video support
- Explicit color space signaling

Meet the needs of emerging applications, e.g., SMPTE ST 2110
Payload structure

One video frame/field = one image = one JPEG 2000 codestream

Two kinds of payload

- Main RTP Packets contain codestream header information
- Body RTP Packets contain the rest of JPEG 2000 codestream
- Both types of RTP Packets have a 64-bit Payload Header
## Basic parameters

<table>
<thead>
<tr>
<th>Media type</th>
</tr>
</thead>
<tbody>
<tr>
<td>video/jpeg2000-scl</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Header fields</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Marker</td>
<td>Signals last RTP packet of a codestream</td>
</tr>
<tr>
<td>Timestamp</td>
<td>Presentation time of the video frame/field to which the codestream corresponds</td>
</tr>
<tr>
<td>Sequence number</td>
<td>Extended by ESEQ as in RFC 4175</td>
</tr>
</tbody>
</table>
What is sub-codestream latency?

Start transmitting parts of the encoded image before encoding of the full image is complete.

Very low latencies possible (32 video lines at 1080 lines video at 24 fps = 1.2 ms)
Enabling sub-codestream latency

Packet loss recovery depends on identifying resync points within the codestream.

JPEG 2000 PLM and PLT (packet length) marker segments cannot be used because they cannot be written until after the codestream is complete.

**NEW** POS and PID fields of the Payload Format identify resync information in each RTP packet (if a resync point exists):

- **POS (position):** byte offset of the resync point
- **PID (precinct ID):** identifies the precinct to which the resync point belongs
Resolution and quality scalability

J2K codestreams can be packetized according to discrete quality and resolution levels

QUAL and RES fields identify the quality and resolution levels present within each RTP packet

A receiver or network agent can select quality or resolution levels at the RTP packet level without parsing the codestream
High-precision clock recovery

RTP timestamp is low resolution compared to RTP packet rate

- 24 fps video @ 800 Mbps
- Timestamp rate: 24 Hz
- Packet rate: 67 kHz

PTSTAMP field indicates transmission time of each RTP packet in the same time base as RTP timestamp

- Increases clock recovery speed and resolution at the receiver
- Enabled by the sender using the P header field
Code-block caching

Improves bandwidth efficiency for mostly static video, e.g., screen content

Sender (for temporally static areas)
- only occasionally transmits code-blocks
- otherwise sends empty code-blocks

Receiver maintains a code-block cache
- updates the cache when it receives a non-empty code-blocks
- fetches from the cache when it receives an empty code-block
- standard JPEG 2000 decoder

Enabled by the C field and signaled in the media type

Code-block: coded samples belonging to a rectangular (spatial) region within one resolution level of one component
Intended as standards track RFC

Looking for feedback

Thank you and looking forward to your feedback!
HEVC Profile for WebRTC

draft-aboba-avtcore-hevc-webrtc

Bernard Aboba
Philipp Hancke
Open Issues

https://github.com/aboba/hevc-webRTC/issues

- **Issue 2/PR 10**: Out-of-band sprop-vps/sprop-sps/sprop-pps
- **Issue 3/PR 9**: tx-mode
- **Issue 4/PR 8**: max-cpb/max-dpb
- **Issue 5/PR 15**: max-fps/max-br
- **Issue 12/PR 16**: PACI packet
- **Issue 13**: RPSI RTCP feedback support
**Issue 2**: Out-of-band sprop-vps/sprop-sps/sprop-pps

- RFC 7742 Section 6.2 states:
  
  “sprop-parameter-sets: H.264 allows sequence and picture information to be sent both in-band and out-of-band. WebRTC implementations MUST signal this information in-band. This means that WebRTC implementations MUST NOT include this parameter in the SDP they generate.”

- Section 2 of draft-aboba-avtcore-hevc-webrtc contains similar advice:

  “sprop-sps, sprop-pps, sprop-vps, sprop-sei: H.265 allows sequence and picture information to be sent both in-band and out-of-band. WebRTC implementations MUST signal this information in-band. This means that WebRTC implementations MUST NOT include these parameters in the SDP they generate, and SHOULD silently ignore these parameters if they are received.”

- Do we have consensus to keep this text?
Issue 2/PR 10: IDR text

- Desire to avoid H.264 issues:
  - H.264 encoders that did not update the SPS/PPS ID even if those parameters changed.
  - Matching (and applying) in-band parameter sets to an IDR.

- PR 10 adds the following text to Section 2:

  “An IDR sent MUST always be preceded by the relevant parameter sets sent in a packet (not necessarily a separate packet) with the same RTP timestamp as the IDR.”
Issue 3/PR 9: tx-mode

- Currently the draft doesn't cover the tx-mode SDP parameter.
- PR 9 adds the following text to Section 2:

> “tx-mode: Implementations SHOULD NOT include this parameter within SDP. If no tx-mode parameter is present, a value of "SRST" MUST be inferred.”

- Remaining question: What happens if a WebRTC browser receives an Offer with “tx-mode: MRST” or “MRMT”?
  - Offer would most likely come from a gateway, not another browser.
  - Is it acceptable for setRemoteDescription to fail to apply such an Offer?
  - RFC 7798 Section 4.3 says: “Receivers MUST support all of SRST, MRST and MRMT.”
**Issue 4/PR 8: max-cpb/max-dpb**

- Section 2 says:

  “max-fps, max-cpb, max-dpb, and max-br:

  These parameters allow the implementation to specify that they can support certain features of H.265 at higher rates and values than those signaled by their level (set with level-id). Implementations MAY include these parameters in their SDP, but they SHOULD interpret them when receiving them, allowing them to send the highest quality of video possible.”

- PR 8 removes max-cbp and max-dpb.
Issue 5/PR 15: max-fps/max-br

- Should we also remove max-fps and max-br?
- RFC 7742 Section 6.1 says:
  - “VP8 encoders MUST limit the streams they send to conform to the values indicated by receivers in the corresponding max-fr and max-fs SDP attributes.”
  - Chromium lack of support for max-fr and max-fs has been unpopular. See: https://bugs.chromium.org/p/webrtc/issues/detail?id=2488 (34 stars)
- HEVC (in contrast to VP8) has profile/tier/level negotiation.
  - So parameters such as max-fps/max-br indicate additional receive constraints (for example, a receiver limited in maximum framerate/bandwidth beyond the indicated profile/tier/level-id). Example:
    - main profile 3.1 can by default receive 1280×720@33.7 fps.
    - max-fps only needed if the receiver cannot handle 33.7 fps @ 1280 x 720
    - max-br only needed if the receiver cannot handle 10 Mbps.
- PR 15 removes max-fps/max-br.
### Issue 12: PACI packet

- PACI extensions are defined in RFC7798 Section 4.4.4.
  - A new payload type 50 is introduced for the PACI packet.
  - Section 4.4.4.2 defines a payload header extension for Temporal Scalability Control Information (TSCI).
- In WebRTC implementations, TSCI information can also be communicated using RTP header extensions such as the Dependency Descriptor or Generic Frame Descriptor.
  - If TSCI is supported via negotiated RTP header extensions, what should implementations do with TSCI information received in PACI extensions?
Issue 12/PR 16: Add paragraph on TSCI

- PR 16 Proposed text:

"[RFC7798] Section 4.5 defines how Temporal Scalability Control Information (TSCI) is communicated using PACI Extensions defined in [RFC7798] Section 4.4.4.2. When WebRTC implementations negotiate the use of RTP header extensions containing TSCI information, such as the Dependency Descriptor [DD], the TSCI information contained in the RTP header extensions takes precedence and implementations MUST ignore TSCI information contained in the PACI."
**Issue 13: RPSI RTCP feedback support**

- RFC 7798 Section 8.3 defines support for RPSI feedback in HEVC.
- “Use of RTP in WebRTC” RFC 8834 Section 5.1.4 says:
  - “Reference Picture Selection Indication (RPSI) messages are defined in Section 6.3.3 of the RTP/AVPF profile [RFC4585]. Some video-encoding standards allow the use of older reference pictures than the most recent one for predictive coding. If such a codec is in use, and if the encoder has learned that encoder-decoder synchronization has been lost, then a known-as-correct reference picture can be used as a base for future coding. The RPSI message allows this to be signaled. Receivers that detect that encoder-decoder synchronization has been lost SHOULD generate an RPSI feedback message if the codec being used supports reference-picture selection. An RTP packet-stream sender that receives such an RPSI message SHOULD act on that messages to change the reference picture, if it is possible to do so within the available bandwidth constraints and with the codec being used.”

- Clarifications in RFC 8082 Section 6.3.
  - “no implementations are known in conjunction of layered codecs. The current understanding is that the reception of an RPSI message on any layer indicating a missing reference picture forces the encoder to appropriately handle that missing reference picture in the layer indicated, and in all dependent layers.”
Issue 13: Reference Status in WebRTC

- libwebrtc hasn’t supported RPSI since 2017.
  - Initial RPSI implementation supporting VP8 and VP9 was removed.
  - H.264 implementation doesn’t support alternative reference frames.
  - AV1 RTP payload specification doesn’t mention RPSI.
  - Frame dependencies are provided in DD and GFD RTP header extensions
- Are there alternative ways to support alternative reference frames?
  - RFC 4585 Section 6.3.1.1 on PLI:
    - "The sender MAY react to a PLI by transmitting an intra-picture to achieve resynchronization (making this message effectively similar to the FIR message as defined in [6]); however, the sender MUST consider congestion control as outlined in Section 7, which MAY restrict its ability to send an intra frame."
  - Can a sender safely react to a PLI by switching to an alternative reference frame?
  - libwebrtc also supports the Loss Notification (LNTF) custom RTCP message.
- History: https://bugs.chromium.org/p/webrtc/issues/detail?id=10662
Comments? Next steps?

● ?
RTP Payload Format for SCIP


Dan Hanson
Mike Faller
Published **draft-ietf-avtcore-rtp-scip-06** on September 19

- **Key Points** section immediately after Abstract
  - SCIP can be considered a tunneling protocol
  - SCIP is an application layer protocol that uses RTP as a transport
  - Unlike most other IETF RTP Payload Format RFCs, SCIP does not have a discrete payload format
  - Reliable transport for SCIP control messages is implemented at the SCIP application layer
  - SCIP uses versioning mechanisms to ensure its evolution prevents ossification of the protocol
- Added example of listing ‘scip’ in preference order in SDP Offer/Answer
- Added link to older public version of SCIP-210: [https://www.iad.gov/SecurePhone/index.cfm](https://www.iad.gov/SecurePhone/index.cfm)
SCIP Draft 06 (2)

- Request another review by IESG
- IESG evaluation record
  - 3 Discuss
  - 1 Yes
  - 5 No Objection
  - 2 Abstain
  - 4 No Record
- Needs 5 more Yes or No Objection positions to pass
RTP over QUIC


Mathis Engelbart, Jörg Ott, Spencer Dawkins
Done

- Considerations for streams/datagrams/mixture of both #93 / PR#123
- Reasoning for not mandating a rate adaptation algorithm #116 / PR#121
- Reference RTCP subsections from tables #80 / PR#120
- Add multihop topologies considerations for RTCP #75 / PR#125
- Remove suppressing RTCP #117 / PR#124
- Published draft-ietf-avtcore-rtp-over-quic-06
STOP_SENDING, RESET_STREAM and Media Dependencies (#112)

- On receiving STOP_SENDING, a sender MUST send RESET_STREAM
- The sender MUST continue to send new media frames on other QUIC streams
- Any media frame that has already been sent on the QUIC stream that received the STOP_SENDING frame, MUST NOT be sent again on the new QUIC stream(s).
STOP_SENDING, RESET_STREAM and Media Dependencies (#112)

- Sender wants to send frames A and B on stream 1
  - Frame B depends on A
- Receiver sends STOP_SENDING, because A was not received in time
- STOP_SENDING arrives at sender before the sender transmits frame B
- MUST send RESET_STREAM on stream 1 and start on new stream
  - Provide notification to application that this has happened
- The application is in charge of what to do with "the next frame"
  - Drop B, because B is dependent on a lost or discarded frame
  - Treat STOP_SENDING as PLI for the new stream
  - Create an independent next frame (B' or other)
  - Transmit B on a new stream accepting the resulting decoding artifacts
Usefulness of CLOSE_STREAM and ENOUGH? (#113)

- CLOSE_STREAM and ENOUGH are like RESET_STREAM and STOP_SENDING, but include an offset.
- The offset specifies up to where a stream will be reliably retransmitted and processed.
- Could be useful to indicate that a stream should be reliable up to the end of an RTP packet so that a stream would not be dropped in the middle of a packet.
- The offset only makes the first half reliable, but in real-time transmission, we typically want to drop earlier pieces, instead of the later ones.
- Proposing "not yet" or "wontfix" labels in Github.

---

[Image: IETF logo]
Suppressing QUIC signaling in favor of RTCP signaling (#117 / PR#124)

- It turns out that we can't "suppress" QUIC feedback or RTCP feedback
- Use *enhance statistics* instead of *replacing* or *suppressing* RTCP feedback
- RoQ won’t require QUIC protocol updates and shouldn’t update RTCP feedback rules either
- But we still want to allow usage of QUIC state where useful, to enhance RTCP statistics and reduce RTCP bandwidth usage where possible
RTCP in Multihop Topologies (#75 / PR#125)

- Add considerations for RTCP enhancements in multihop topologies
- Non-RoQ participants may be unaware of QUIC usage and expect more RTCP feedback
- RoQ-participants may be unaware of non-RoQ protocols
- There are RFC7667 topologies that do not terminate RTP sessions
  - Translators may generate additional RTCP packets
  - Which of these topologies do we care about in 2023?
Next Steps

- [#76, #13]: Finish Error Codes
- [#65]: RTP and non-RTP traffic sharing multiple QUIC connections
- [#84]: Congestion control when sharing connections with non-RTP streams
- [#86]: Coalescing RTP packets in single QUIC packet
- [#87]: Real-time congestion controllers tied to RTP/usable in QUIC stacks?
- [#111]: Check SHOULD requirements for ROQ receivers
- [#114]: Reality check on RoC DATAGRAM use
- [#115]: Motivation: Exploiting Multiple Connections
- [#122]: Define how RoQ reacts to the absence of max_datagram_frame_size transport parameter
- [#126]: Review Appendix B for completeness
- [#127]: Normative References
- [#128]: Congestion Control and Rate Control
RTP Payload Format for SFrame

draft-ietf-sframe-enc
draft-thatcher-avtcore-rtp-sframe

Peter Thatcher
Changes Since Last Meeting

- Feedback was to allow "stage 1" (media-format-specific) packetization to output more than 1 RTP packet before feeding into "stage 2" (sframe-specific) packetization.
  - So I made that change
- Feedback was also to issue a “Call for Adoption”
  - So I renamed it to draft-ietf-avtcore-rtp-sframe (in Github).
Next Steps

- Issue a “Call for adoption”?
- If adopted, submit to IETF data tracker as AVTCORE document?
- Then what?
Thank you

Special thanks to:

The Secretariat, WG Participants & ADs