

# AVTCORE WG

Virtual Interim

October 8, 2024





08:00 - 10:00 AM Pacific Time

Mailing list: [avt@ietf.org](mailto:avt@ietf.org)

Meeting info: <https://datatracker.ietf.org/meeting/interim-2024-avtcore-03/session/avtcore>

Notes: <https://notes.ietf.org/notes-ietf-interim-2024-avtcore-03-avtcore>

# 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/>(Privacy Policy)

# Participant Obligations

- When starting a presentation you **MUST** say if:
  - There is IPR associated with your draft
- When asking questions or commenting on a draft:
  - You **MUST** disclose any IPR your employer controls relating to the technology under discussion
- RFC 6701 “Sanctions Available for application to Violators of IETF PR Policy”
  - Describes potential consequences of violating these policies.

# Note really well

- IETF meetings, virtual meetings, and mailing lists are intended for professional collaboration and networking, as defined in the [IETF Guidelines for Conduct](#) (RFC 7154), the [IETF Anti-Harassment Policy](#), and the [IETF Anti-Harassment Procedures](#) (RFC 7776). 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-2024-avtcore-03/session/avtcore>

- Notes: <https://notes.ietf.org/notes-ietf-interim-2024-avtcore-03-avtcore>
- Secretariat: [mtd@jabber.ietf.org](mailto:mtd@jabber.ietf.org)
- WG Chairs: Jonathan Lennox & Bernard Aboba
- Zulip Scribe: Spencer Dawkins
- Note taker: Spencer Dawkins

# 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 9584: was draft-ietf-avtcore-rtp-enc
  - RFC 9607: was draft-ietf-avtcore-rtp-scip
- RFC Editor Queue
  - draft-ietf-avtext-lrr (AUTH48)
  - draft-ietf-payload-vp9 (AUTH48)
  - draft-ietf-avtext-framemarking (AUTH48)

# Draft Status (cont'd)

- Completed IETF Last Call (ended October 7, 2024)
  - draft-ietf-avtcore-rtp-payload-registry
- Completed WGLC (ended September 17, 2024)
  - draft-ietf-avtcore-rtp-j2k-scl ([WGLC Summary](#))
- [In WGLC \(to end October 22, 2024\)](#)
  - draft-ietf-avtcore-rtcp-green-metadata
- Adopted
  - draft-ietf-avtcore-rtp-haptics
  - draft-ietf-avtcore-rtp-over-quic
  - draft-ietf-avtcore-hevc-webrtc
  - draft-ietf-avtcore-rtp-volumetric-media-roi ([CFA summary](#))
- Expired
  - draft-ietf-avtcore-rtp-sframe

# AVTCORE GitHub Setup



- Organization created: <https://github.com/ietf-wg-avtcore>
- Recently adopted drafts can create or transfer repositories within the new hierarchy
  - Done:
    - draft-ietf-avtcore-rtp-haptics
    - draft-ietf-avtcore-rtp-j2k-scl
    - draft-ietf-avtcore-hevc-webrtc
  - Still to be transferred:
    - draft-ietf-avtcore-rtcp-green-metadata
    - draft-ietf-avtcore-rtp-sframe
    - draft-ietf-avtcore-rtp-volumetric-media-roi
- Once transferred, they should be added to the “Activity this week” e-mail
  - A pull request against Mark Nottingham’s repo

# Agenda



1. Note Well, Note Takers/, Agenda Bashing, Draft status (Chairs, 10 min)
2. [RTP Payload Format for sub-codestream J2K streaming](#) (P. Lemieux, 10 min)  
[draft-ietf-avtcore-rtp-j2k-scl](#)
3. [H.265 Profile for WebRTC](#) (B. Aboba, 15 min)  
[draft-ietf-avtcore-hevc-webrtc](#)
4. [RTP Payload Format for Volumetric Video Coding \(V3C\)](#) (L. Ilola, 10 min)  
[draft-ietf-avtcore-rtp-v3c](#)
5. [RTP over QUIC](#) (Mathis Engelbart, 20 min)  
[draft-ietf-avtcore-rtp-over-quic](#)
6. [RTCP Messages for Point Cloud Prioritization](#) (Mathis Engelbart, 15 min)  
[draft-engelbart-avtcore-rtcp-point-cloud-roi](#)
7. [Absolute Capture Time RTP Header Extension](#) (Harald Alvestrand, 10 min)  
[draft-alvestrand-avtcore-abs-capture-time](#)
8. [Wrapup and Next Steps](#) (Chairs, 10 min)

# High-performance JPEG 2000 RTP payload format

[draft-ietf-avtcore-rtp-j2k-scl](#)

P. A. Lemieux

**Start time: 08:10**

**End time: 08:20**

# Status



- <https://github.com/ietf-wg-avtcore/draft-ietf-avtcore-rtp-j2k-scl/issues>
  - No substantial comments
  - Only minor typos
- [WGLC summary posted.](#)
- Do we have a volunteer to act as Document Shepard?

# H.265 Profile for WebRTC

[draft-ietf-avtcore-hevc-webrtc](#)

B. Aboba

P. Hancke

**Start time: 08:20**

**End time: 08:35**

# For Discussion Today

- **Open Issues**

(<https://github.com/ietf-wg-avtcore/draft-ietf-avtcore-hevc-webrtc/issues>)

- [Issue 23/PR 24](#): tx-mode text
- [Issue 25/PR 28](#): NALU TID values
- [Issue 22](#): Issues with Receive-only codecs
- [Issue 27](#): Clarify requirement of handling symmetric/asymmetric levels of HEVC Offer/Answer

- Section 2 says:  
“tx-mode: Implementations SHOULD NOT include this parameter within SDP. If no tx-mode parameter is present, a value of "SRST" MUST be inferred. Implementations MUST support "SRST"; support for "MRST" and "MRMT" is OPTIONAL. Implementations that do not support "MRST" or "MRMT" MUST NOT include these tx-mode values in SDP.”
- The Chrome implementation includes the following fntp:  
a=fnTP:116 level-id=93;profile-id=1;tier-flag=0;tx-mode=SRST
- Including tx-mode rather than omitting it is less likely to result in interop problems with non-WebRTC implementations of H.265 that may not have the same interpretation when tx-mode is omitted.
- PR 24 Proposal: Change “SHOULD NOT” to “SHOULD”.

## Issue 25: NALU TID values

- The specification doesn't currently talk about packetization constraints on TID values in NAL units.
- RFC 7798 Section 4.4.2 states:  
"The value of TID MUST be the lowest value of TID of all the aggregated NAL units.  
Informative note: All VCL NAL units in an AP have the same TID value since they belong to the same access unit. However, an AP may contain non-VCL NAL units for which the TID value in the NAL unit header may be different than the TID value of the VCL NAL units in the same AP."
  - non-VCL NAL units (SPS, PPS, VPS, SEI, etc.) may be of interest to multiple operating points.
  - If an RTP packet includes VCL NAL units with a single TID as well as an AP containing non-VCL NAL units with multiple TIDs, how should a MANE or SFM forward the packet?
    - Proposal: Separate VCL NAL units and the AP containing non-VCL NAL units with different TIDs into distinct RTP packets.
      - AP containing non-VCL NAL units utilizes lowest TID, is forwarded to participants whose operating point contains that TID.

## PR 28: TID constraints

- Proposed text:

```
159 + [RFC7798] Section 4.4.2 describes how APs are carried within RTP payloads:
160 +
161 + "An AP consists of a payload header (denoted as PayloadHdr) followed
162 + by two or more aggregation units... The value of TID MUST be the
163 + lowest value of TID of all the aggregated NAL units.
164 +
165 + Informative note: All VCL NAL units in an AP have the same TID
166 + value since they belong to the same access unit. However, an AP
167 + may contain non-VCL NAL units for which the TID value in the NAL
168 + unit header may be different than the TID value of the VCL NAL
169 + units in the same AP."
170 +
171 + Within an RTP payload, VCL NAL units MUST NOT be aggregated with
172 + non-VCL NAL units with a lower TID value. Instead the non-VCL
173 + NAL units with a lower TID value MUST be packetized within a distinct
174 + RTP packet. This ensures that a MANE or SFM can forward VCL and
175 + non-VCL NAL units to the correct set of participants.
```

## Issue 22: Issues with Receive-only codecs

- W3C WEBRTC WG discussed Send-only and Receive-only codecs at TPAC 2024.
  - WG agreed that API should conform to RFC 3264, Section 5.1:
    - sendrecv m-lines only include formats (codec + fmp parameters) that can be sent and received.
    - sendonly m-lines only include formats that can be sent.
    - recvonly m-lines only include formats that can be received.
  - Question: what fmp parameters are relevant for each codec?
- Tracking issue for WebRTC API changes:
  - [Issue 3006](#): Align spec w/codec direction decision
- [Issue 27](#) filed to clarify implications for H.265 support in WebRTC.

## Issue 27: Clarify requirement of handling

### symmetric/asymmetric levels of HEVC Offer/Answer

- H.265 proposal:
  - level-asymmetry assumed
  - Only advertise the highest supported level-id value for each profile & tier.
- sendrecv m-line
  - Offer includes highest level-id that the Offerer can decode.
  - Answer includes the highest level-id that the Answerer can decode.
- sendonly m-line
  - Offer includes highest level-id that the Offerer can encode.
  - Answer includes the highest level-id that the Answerer can decode.
- recvonly m-line
  - Offer includes highest level-id that the Offerer can decode.
  - Answer includes the highest level-id that the Answerer can encode.

## Issue 27: Clarify HEVC Offer/Answer (cont'd)

- With a fixed profile/tier, if the Offerer is able to decode/encode up to level 5.2, and the Answerer is able to decode/encode up to level 3.1, what level/levels should the Offer and Answer include for sendonly/sendrecv/recvonly?
  - sendrecv
    - Offer includes level 5.2 (maximum it can decode)
    - Answer includes level 3.1 (maximum it can decode)
  - sendonly
    - Offer includes level 5.2 (maximum it can encode)
    - Answer includes level 3.1 (maximum it can decode)
  - recvonly
    - Offer includes level 5.2 (maximum it can decode)
    - Answer includes level 3.1 (maximum it can encode)

## Issue 27: Clarify HEVC Offer/Answer (cont'd)

- With a fixed profile/tier, if the Offerer is able to decode/encode up to level 3.1, and the Answerer is able to decode/encode up to level 5.2, what level/levels should the Offer and Answer include for sendonly/sendrecv/recvonly?
  - sendrecv
    - Offer includes level 3.1 (maximum it can decode)
    - Answer includes level 5.2 (maximum it can decode)
  - sendonly
    - Offer includes level 3.1 (maximum it can encode)
    - Answer includes level 5.2 (maximum it can decode)
  - recvonly
    - Offer includes level 3.1 (maximum it can decode)
    - Answer includes level 5.2 (maximum it can encode)

## Issue 27: Clarify HEVC Offer/Answer (cont'd)

- With a fixed profile/tier, if the Offerer is able to encode up to level 3.1, but decode up to level 5.2, and the Answerer is able to decode/encode up to level 5.2, what level/levels should the Offer and Answer include for sendonly/sendrecv/recvonly?
  - sendrecv
    - Offer includes level 5.2 (maximum it can decode)
    - Answer includes level 5.2 (maximum it can decode)
  - sendonly
    - Offer includes level 3.1 (maximum it can encode)
    - Answer includes level 5.2 (maximum it can decode)
  - recvonly
    - Offer includes level 5.2 (maximum it can decode)
    - Answer includes level 5.2 (maximum it can encode)

## Issue 27: Clarify HEVC Offer/Answer (cont'd)

- With a fixed profile/tier, if the Offerer is able to encode up to level 5.2, but decode up to level 3.1, and the Answerer is able to decode/encode up to level 5.2, what level/levels should the Offer and Answer include for sendonly/sendrecv/recvonly?
  - sendrecv
    - Offer includes level 3.1 (maximum it can decode)
    - Answer includes level 5.2 (maximum it can decode)
  - sendonly
    - Offer includes level 5.2 (maximum it can encode)
    - Answer includes level 5.2 (maximum it can decode)
  - recvonly
    - Offer includes level 3.1 (maximum it can decode)
    - Answer includes level 5.2 (maximum it can encode)

# RTP Payload Format for Volumetric Video Coding (V3C)

[draft-ietf-avtcore-rtp-v3c](#)

L. Ilola

L. Kondrad

**Start time: 08:35**

**End time: 08:45**

# Changes since Vancouver



Version 7 available addressing WGLC and post-WGLC suggestions

## New issues closed

- [\[#24\]](#) Clarification of offer-answer considerations for partially or fully ignorant V3C receivers.
  - Questions were asked on how to handle session negotiation when the receiver is partially or fully ignorant of V3C coding.
  - V3C video components are coded 2D video bitstreams that can be received, decoded and displayed using legacy 2D video receivers. The content may in many cases not make much sense, but MIV (application of V3C) for example supports fall-back mechanism which allows to record full views in a single video bitstream.
  - There may be also other reasons why a receiver may want to receive 2D video streams even though they don't visually make sense to human eye.
  - As such, we propose to allow implementation flexibility depending on how the sender is set up and how the receiver understands V3C coding, rather than strictly enforcing all receivers to fully implement V3C decoding or not streaming 2D video components at all.
  - Spec text indicating such flexibility has been added

**Proposal to start new WGLC unless there are other issues that would need to be addressed.**

# RTP over QUIC

<https://datatracker.ietf.org/doc/html/draft-ietf-avtcore-rtp-over-quic>

Mathis Engelbart, Jörg Ott, Spencer Dawkins

**Start time: 08:45**

**End time: 09:05**

## RTCP in RoQ ([#226](#), [#227](#))

- #227 adds considerations for sending RTCP over streams or datagrams
- Queueing within QUIC stacks may lead to RTT measurements that are different from the network RTT as observed by QUIC
- Overhead cannot be computed deterministically:
  - Varying sizes of QUIC packet and frame headers
  - RTP/RTCP packet coalescing in QUIC packets
  - Fragmentation of RTP/RTCP packets into multiple stream frames
  - Retransmissions (if using QUIC streams)

## RTCP in RoQ: Estimating Overhead ([#226](#), [#227](#))

- IP: 20B (v4) / 40B (v6), UDP: 8B
- QUIC Short header packets:  $\geq 2B$ 
  - 1B header,  $\leq 20B$  connection ID, 1-8B sequence number
- QUIC STREAM frames:  $\geq 2B$ 
  - 1B type, 1-8B stream ID, optional: 1-8B offset + 1-8B length
- QUIC DATAGRAM frames:  $\geq 2B$ 
  - 1B type, optional: 1-2 B length
- RoQ Flow ID: 1-8B, every datagram and some stream frames
  
- Streams: 33-90B (v4) or 53-110B (v6)
  - $20|40 + 8 + (1+8+4) + (1+4+2+2) + 1 = 51 | 71B$  overhead
- Datagrams: 33-74B (v4) or 53-94B (v6)
  - $20|40 + 8 + (1+8+4) + (1+2) + 1 = 37 | 57B$  overhead

## RTCP in RoQ: Queueing/Priorities ([#226](#), [#227](#))

- RTP/RTCP packets may be queued in the QUIC stack and transmission timing cannot be controlled in the same way as with UDP
- When RTP/RTCP packets are sent on different QUIC streams, ordering of delivery is not defined
- RTP/RTCP packets may be transmitted on streams or datagrams and the relative transmission order between streams and datagrams is also undefined
- Streams for RTP and RTCP may be associated with different priorities and the impact of these priorities is not defined
- RTCP packets may reference RTP packets, e.g., highest sequence number sent in SR packets. If QUIC stacks cause substantial RTP packet reordering or delays, this may significantly confuse statistics calculated from RTCP

## RTCP in RoQ: Queueing ([#226](#), [#227](#))

- An alternative API could use callbacks from the QUIC stack to allow the application to generate "just in time" RTCP packets, but that is only possible if such an API is provided by the QUIC stack

## RoQ and Interop Tests (Details in [Wiki](#))

Client \ Server	mengelbart/roq	bbc/gst-roq	Lorenzo
mengelbart/roq	1,2,3,4,5,6	1,2,3**,4,5,6	2,4,6 *
bbc/gst-roq	1,2,3,4**,5,6	1,2,3,4,5,6 ?	
Lorenzo	1,3,5 *		1,2,3,4,5,6

1. C-S Datagrams
2. S-C Datagrams
3. C-S Single RTP packet per stream
4. S-C Single RTP packet per stream
5. C-S Multiple RTP packets per stream
6. S-C Multiple RTP packets per stream

\*: QUIC interop issue after key update

\*\* : mengelbart/roq can't read data after receiving RESET\_STREAM

## Next Steps

- Solve RTCP issue
  - Please read and comment on [#226](#), [#227](#)!
- Continue Interop tests
  - Please let us know of any other implementation and if you want to participate
- SDP

# RTCP Messages for Point Cloud Prioritization

[draft-engelbart-avtcore-rtcp-point-cloud-roi](#)

Mathis Engelbart, Jörg Ott, Lukasz Kondrad

**Start time: 09:05**

**End time: 09:20**

# Background: Point Clouds/G-PCC

- Data Structure used to represent three dimensional data
- List of points in three dimensional space
- Each point may be associated with zero or more attributes e.g., Color, Reflectance, ...
- Acquired by, e.g., LiDAR, Radar, Multiple Camera setups
- Example use case: 3D representation of a vehicle's surrounding environment
- ISO/IEC 23090-9: *Information technology — Coded representation of immersive media — Part 9: Geometry-based point cloud compression*
- [draft-engelbart-avtcore-rtp-gpcc](#)

# Motivation

- Depending on the number of points in a point cloud frame, the bandwidth may become very large
- Not all points are equally important for further processing
- Focusing on some regions or specific parameters of points within some regions is often sufficient for further processing
- The receiver might know better what parts should be prioritized
- [draft-engelbart-avtcore-rtcp-point-cloud-roi](#) defines RTCP feedback messages and header extensions to request/announce prioritization of certain regions and parameters

# Octree Encoding

- Tree data structure commonly used with point clouds
- Subdivides space in eight octants
- Every internal node in an octree has exactly eight children each representing one of the eight octants
- Binary representation can use one byte per node:
  - Each bit indicates presence of the child node
  - Leaf nodes encoded as zero-byte
- Optional: define bounding box of the octree encoding

# RTCP Messages and RTP Header extension for Point Cloud Prioritization

- Exchange parameters for real-time streamed point clouds
  - E.g., prioritization of regions, resolution, point attributes, ...
- RTCP messages consist of
  - Header: Flags to signal presence of parameters
  - Absolute/relative octree encoding to reference regions in a point cloud
  - Optional parameters for every referenced region
- Currently defined parameters
  - Priority: 1 Byte per region
  - Attributes: Bitset per region, semantics of every bit has to be signalled
  - Level-of-Detail: WIP

`<Header><{abs, rel}-octree>[<priority>][<attributes>][<level-of-detail>]`

## Signaling

- Define new rtcp-fb parameter “oer r” (octree encoded region request)
- Define new URI for RTP header extension  
“urn:ietf:params:rtp-hdext:octree-region”
- TODO: Define signalling for attribute bitmasks and possibly other parameters

# Absolute Capture Timestamp RTP Header Extension

[draft-alvestrand-avtcore-abs-capture-time](#)

Harald Alvestrand

**Start time: 09:20**

**End time: 09:30**

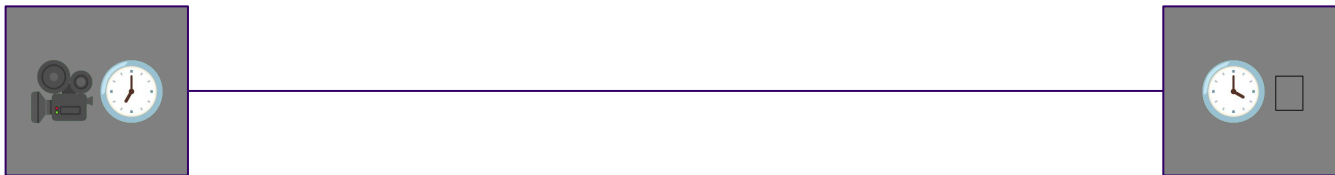
# Problem to be solved

- How does the displaying system know the time at which a frame was captured?
  - The clocks may be out of sync
  - There may be delay before sending the media
  - There may be network delays
  - There may be multiple hops in the path
- Solution proposed: Abs-capture-timestamp
  - [draft-alvestrand-avtcore-abs-capture-time](#)
  - <https://webrtc.googlesource.com/src/+refs/heads/main/docs/native-code/rtp-hdext/abs-capture-time>

# Simple system

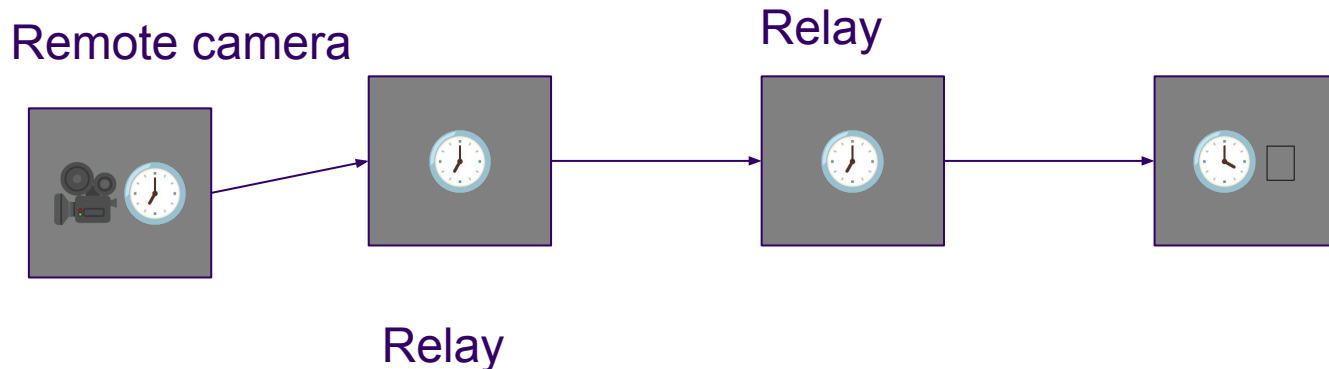


RTP provides NTP timestamps and RTT estimates



This is enough to “translate between the clocks”

# Complex system



Each element can estimate delay and clock skew for one hop.  
No element has the full picture.



# Experience from experimentation

- Not every packet needs this. “Once in a while” is enough.
- Clock drift is not directly addressed. Depends on regular updates “once in a while”.
- Enough to get synchronization of media over diverse paths working.

# Wrapup and Next Steps

Chairs

**Start time: 09:30**

**End time: 09:40**

# Action Items

1. [draft-ietf-avtcore-rtp-j2k-scl](#)
  - a. Chairs & authors: find a document shepard
2. [draft-ietf-avtcore-rtp-v3c](#)
  - a. Chairs: start 2nd WGLC before IETF 121
  - b. Authors: *move this draft into the AVTCORE organization*
3. [draft-alvestrand-avtcore-abs-capture-time](#)
  - a. Participants: Comment in github
  - b. Chairs: Call for adoption after IETF 121

# Thank you

Special thanks to:

The Secretariat, WG Participants & ADs