

# AVTCORE WG

## IETF 115

Hybrid Meeting

Tuesday, November 8, 2022

09:30 - 11:30 London Time

Session I, Mezzanine 10-11

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

Notes: <https://notes.ietf.org/notes-ietf-115-avtcore>

MeetEcho link:

<https://www.conf.meetecho.com/conference/?group=avtcore&short=avtcore&item=1>

# IETF 115 Meeting Tips

## In-person participants





- Make sure to sign into the session using the Meetecho (usually the “Meetecho lite” client) from the Datatracker agenda
- Use Meetecho to join the mic queue
- *Keep audio and video off if not using the onsite version*
- **Wear masks unless actively speaking at the microphone.**



## Remote participants

- Make sure your audio and video are off unless you are chairing or presenting during a session
- Use of a headset is strongly recommended

# IETF 115 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.

# Resources for IETF 115 London

- Agenda  
<https://datatracker.ietf.org/meeting/agenda>
- Meetecho and other information:  
<https://www.ietf.org/how/meetings/115/preparation>
- If you need technical assistance, see the Reporting Issues page:  
<http://www.ietf.org/how/meetings/issues/>

# 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)

# 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.

# Reminder: IETF Mask Policy

- Masks must be worn in meeting rooms and are recommended for common areas but not required.
- In meeting rooms, masks may briefly be removed for eating and drinking, but that cannot be an excuse to leave them off for long periods.
- In meeting rooms, active speakers, defined as those who are at the front of the room presenting or speaking in the mic queue, can remove their mask while speaking.
- No exemptions for mask wearing, medical or otherwise, will be allowed.
- Masks must be equivalent to N95/FFP2 or better, and free masks will be provided.

<https://www.ietf.org/how/meetings/115/faq/#covidmeasures>

# About this meeting



- Agenda: <https://datatracker.ietf.org/doc/agenda-115-avtcore/>
- Notes: <https://notes.ietf.org/notes-ietf-115-avtcore>
- Secretariat: [mtd@jabber.ietf.org](mailto:mtd@jabber.ietf.org)
- WG Chairs: Jonathan Lennox & Bernard Aboba
- Zulip Scribe: Jonathan Lennox
- Note takers: ?



# Agenda



1. IETF 115 tips and resources, Note Well, Note Takers, Agenda Bashing, Draft status, CfAs (Chairs, 20 min)
2. [RTP Payload Format for SCIP](https://datatracker.ietf.org/doc/html/draft-ietf-avtcore-rtp-scip) (D. Hanson, 10 min)  
<https://datatracker.ietf.org/doc/html/draft-ietf-avtcore-rtp-scip>
3. [Sandbox Update](https://datatracker.ietf.org/doc/html/draft-ietf-avtcore-rtp-over-quic) (B. Aboba, 10 min)  
<https://datatracker.ietf.org/doc/html/draft-ietf-avtcore-rtp-over-quic>
4. [SFrame and RTP over QUIC](https://github.com/mengelbart/rtp-over-quic-draft/issues/29) (P. Thatcher, 15 min)  
<https://github.com/mengelbart/rtp-over-quic-draft/issues/29>
5. [RTP over QUIC](https://datatracker.ietf.org/doc/html/draft-ietf-avtcore-rtp-over-quic) (J. Ott, M. Engelbart, 20 min)  
<https://datatracker.ietf.org/doc/html/draft-ietf-avtcore-rtp-over-quic>
6. [SDP for RTP over QUIC](https://datatracker.ietf.org/doc/html/draft-dawkins-sdp-rtp-quic) (S. Dawkins, 15 min)  
<https://datatracker.ietf.org/doc/html/draft-dawkins-sdp-rtp-quic>
7. [Wrapup and Next Steps](#) (Chairs, 10 min)

# Draft Status



- Published
  - RFC 9071: was draft-ietf-avtcore-multi-party-rtt-mix
  - RFC 9134: was draft-ietf-payload-rtp-jpegxs
- RFC Editor Queue
  - draft-ietf-payload-vp9 (MISSREF)
  - draft-ietf-avtcore-cryptex (RFC-EDITOR)
    - IANA Action State: Rfc-Ed-Ack
  - draft-ietf-avtcore-rtp-vcv (RFC-EDITOR)
    - IANA Action State: Rfc-Ed-Ack

# Draft Status (cont'd)



- Waiting for AD Go-Ahead::Revised I-D Needed
  - draft-ietf-avtext-framemarking
- WGLC Completed, Waiting for WG Chair Go-Ahead : Proposed Standard
  - draft-ietf-avtcore-rtp-scip (WGLC completed May 8, 2022)
    - Gen-Art, Secdir and Art-Art reviews posted
    - Reviewers contacted to confirm status.
  - draft-ietf-avtcore-rfc7983bis (WGLC completed June 6, 2022)
    - WGLC announcement:  
[https://mailarchive.ietf.org/arch/msg/avt/k8r222c7e06\\_5XjYNueFXQWejQI/](https://mailarchive.ietf.org/arch/msg/avt/k8r222c7e06_5XjYNueFXQWejQI/)
    - Revised I-D submitted: draft-ietf-avtcore-rfc7983bis-06
- Adopted
  - draft-ietf-avtcore-rtp-over-quic
  - draft-ietf-avtcore-rtp-enc

# CfA on “Game State over RTP”



- Initial CfA Completed on May 8, 2022
- CfA summary:  
<https://mailarchive.ietf.org/arch/msg/avt/w80E9ihE4rrJyMzU6F9eOYgvYVE/>
- Two responses:
  - In favor:
    - Suhas Nandakumar:  
[https://mailarchive.ietf.org/arch/msg/avt/c\\_\\_oH22Gg-blmOAzVuYj3Ub\\_liU/](https://mailarchive.ietf.org/arch/msg/avt/c__oH22Gg-blmOAzVuYj3Ub_liU/)
  - In favor of adopting the RTP payload format (but not the format itself):
    - Stephan Wenger:  
<https://mailarchive.ietf.org/arch/msg/avt/R7il2WZ1k2xz1jljiQ-MNoPI9Xg/>
- IETF 114 Chair proposal
  - Proponents to provide a plan for obtaining responses outside the IETF (e.g. game developer forums)
  - WG to extend the CfA
- Current status: draft-01 submitted. Do authors wish to continue?

# CfA on “RTP Payload Format for V3C”

- CfA announcement:  
<https://mailarchive.ietf.org/arch/msg/avt/4SZNSxg6ljcAl00bNOUdwvCkYqM/>
- Draft available here:  
<https://datatracker.ietf.org/doc/html/draft-ilola-avtcore-rtp-v3c>
- CfA completed on October 31, 2022
- Replies (all affirmative)
  - [Stephan Wenger](#) (September 29)
  - [Shuai Zhao](#) (September 30)
  - [Lauri Ilola](#) (October 3)
  - [Yong He](#) (October 19)
  - [Ahmed Hamza](#) (October 31)

# CfA on “RTP Control Protocol (RTCP) Messages for Green Metadata”

- CfA announcement:  
<https://mailarchive.ietf.org/arch/msg/avt/ILvQ-P006O3llwyiAC-OQeeHpiU/>
- Draft available here:  
<https://datatracker.ietf.org/doc/html/draft-he-avtcore-rtcp-green-metadata>
- CfA runs until November 30, 2022
- Replies so far (all affirmative):
  - [Yong He](#) (October 31)
  - [Christian Herglotz](#) (November 2)

# RTP Payload Format for for SCIP

*Dan Hanson*

*Mike Faller*

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

# SCIP Draft RFC – Status

- Revision 03 was submitted October 17
  - Responded directly to reviewers regarding issues
  - Revised Abstract, Introduction, and Mapping to SDP sections
  - Submitted as XML format
  - Has the normative vs. informative reference issue for SCIP-210 been settled?
  - Request another review from area directorates?



# Actions and Questions

- What actions are required to resolve issues raised by SECDIR, GENART, and ARTART?
- [Chairs] Authors need to respond to the reviewers:
  - Quoting the review, indicate the changes that have been made in response to specific comments (CC'ing the WG and the relevant directorate).
  - Reviewer response indicating acceptance is desirable.
  - Review/response record will be cited in the Publication Request.

# RTP over QUIC Sandbox

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

Bernard Aboba

# Sandbox Update



- Curious observations
  - Glass-glass latency **considerably** higher than measured frame RTT.
    - Example: With AV1 with full-Hd @ 30 fps and 1 Mbps average target bitrate, glass-glass latency ~ 630 ms with frame RTTmin ~ 100 ms
  - Re-ordering not observed on the receiver.
  - Bandwidth consumption considerably lower than average target bitrate, except that...
    - VP9 with variable bitrate and SVC mode 'L1T3' sends at much higher than average target bitrate!

# Issue Found?

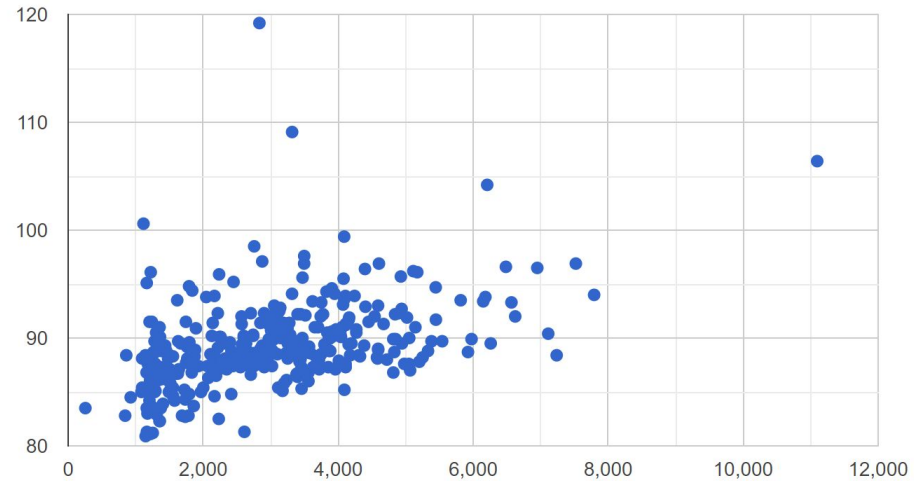


- `await writer.write(chunk)` promise returns once chunk is handed off to the QUIC send queue.
  - Prevents QUIC stack from multiplexing frames/streams.
  - Concurrent sending of P-frames & I-frames suppressed.
  - Result: re-ordering rarely observed on the receiver.
  - Reduces bandwidth usage by stretching out delivery times.
- Replace with `await writer.ready()`, `writer.write(chunk)`, `writer.close()`
  - P-frames and I-frames sent concurrently.
    - More rapid cwind growth at start?
  - Reordering observed at the receiver
    - P-frame(s) often arrive before first I-frame.
  - Bandwidth consumption closer to (but slightly larger than) average target bitrate
  - Glass-glass latency considerably lower
    - Particularly noticeable at high resolutions with complex codecs (e.g. AV1)
- Reference
  - <https://www.learnwithjason.dev/blog/keep-async-await-from-blocking-execution>

# Comparison

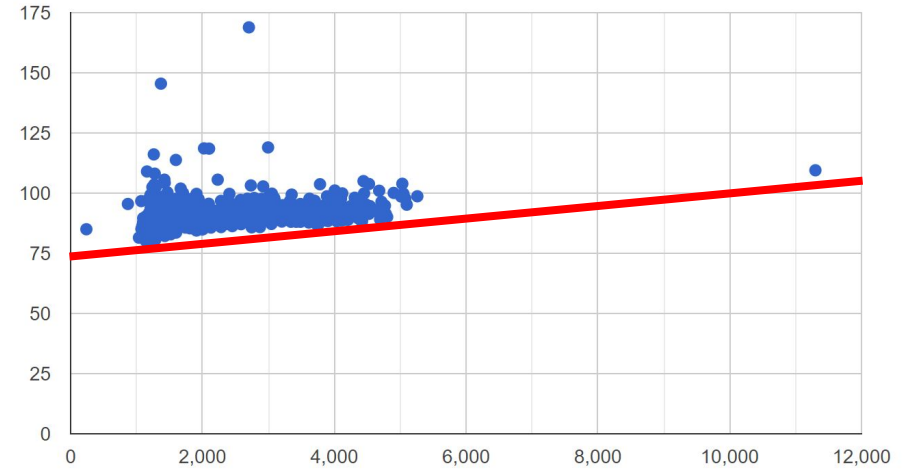


RTT (ms) versus Frame length



Before

RTT (ms) versus Frame length



After

# Pointers



- Tutorials
  - [Introduction to Webcodecs API](#)
  - [Introduction to WebTransport API](#)
- Latest version with BYOB reader, concurrent sending
  - Live: <https://webrtc.internaut.com/wc/wtSender7/> (requires Chrome Canary M108+)
  - GitHub repo: <https://github.com/aboba/wt-demo>
- Old version without BYOB or concurrent sending
  - Live: <https://webrtc.internaut.com/wc/wtSender2/>
- WebCodecs-only demo (no transport)
  - Live: <https://webrtc.internaut.com/wc/wcWorker2/>
  - GitHub repo: <https://github.com/aboba/wc-demo>

# SFrame and RTP over QUIC

<https://github.com/mengelbart/rtp-over-quic-draft/issues/29>

Peter Thatcher

# SFrame/SPacket #29

Open

aboba opened this issue on Aug 3 · 6 comments



aboba commented on Aug 3 · edited



The spec does not currently describe how [SFrame](#) is supported in RTP over QUIC, or what an RTP translator should do. One way to address this would be to add a section that would deal with the issues that come up when implementing [SFrame](#) in RTP over QUIC.



# Part 1: The scenario

It's simple

# RTP over QUIC

Let's you have big RTP packets

(Perhaps a whole video frame in one RTP packet in one QUIC stream)

# SFrame

Encrypts the media

# SFrame over RTP over QUIC

Put a video frame in an SFrame in a QUIC stream

Get a big RTP packet with an SFrame (opaque) payload

# Part 2: The problem

It's also simple

# Converting from Large to Small

A server may need to convert from large to small, and thus “**may need codec-specific knowledge** to packetize the payload of the incoming RTP packets in smaller RTP packets.”  
(I-D.draft-ietf-avtcore-rtp-over-quic-01 Section 4.1)

## But when using SFrame...

...only the endpoints have the key. So any middle box can only know about SFrame, not the cleartext payload. So ...

“may need codec-specific knowledge”

might really mean

“may need SFrame-specific knowledge”

## What about SPacket?

SPacket over RTP over QUIC has the same problem because a large RTP Packet using SPacket still needs to be re-packetized.

(With large RTP packets, there isn't much difference between SFrame and SPacket, since it's one frame per packet).



## What about RTP over QUIC datagrams?

Re-packetization can also be an issue when going from RTP over UDP to RTP over QUIC datagrams due to additional overhead.

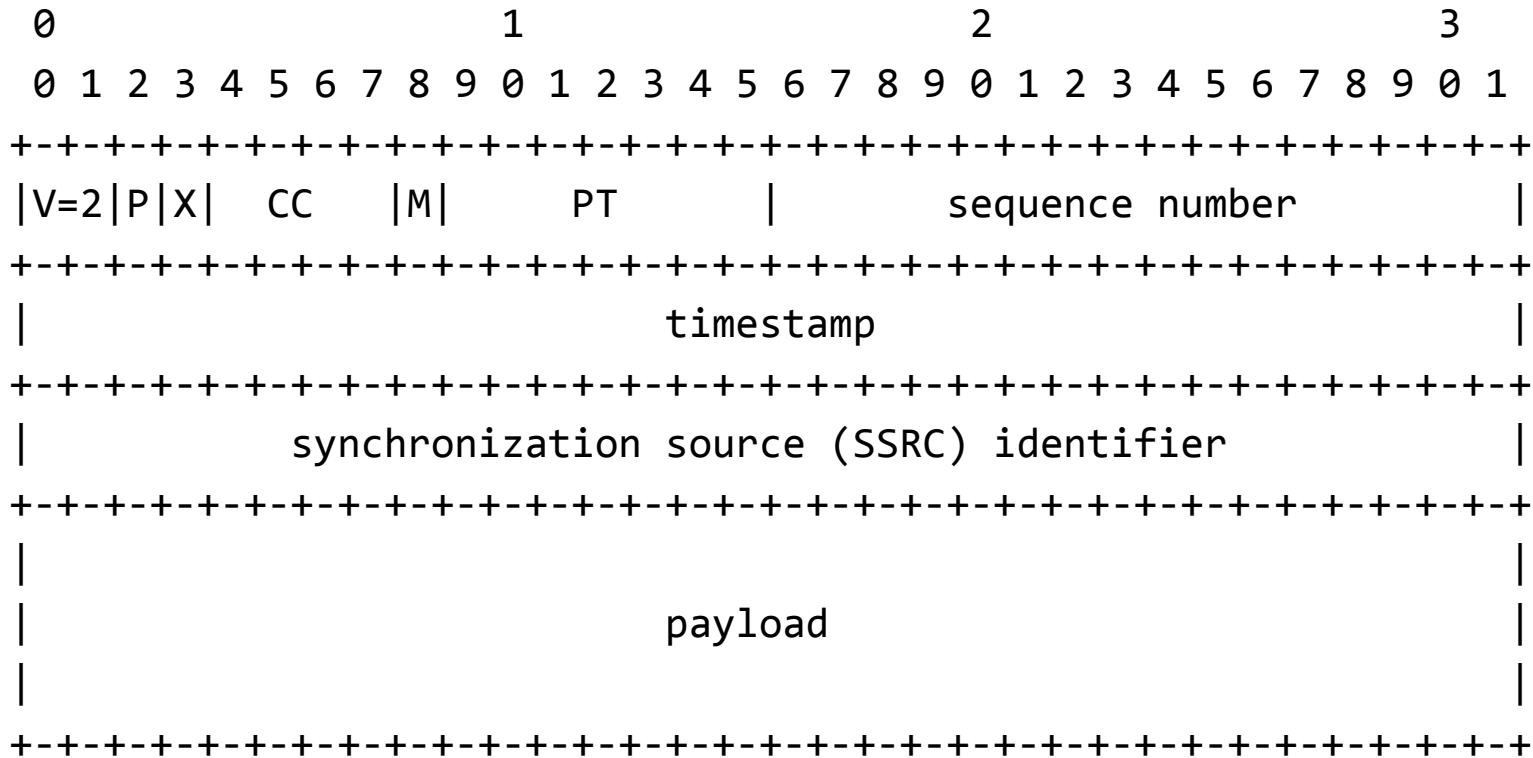
# Summary of the problem

1. RTP over QUIC allows for big RTP packets
2. MTU differences require re-packetizing
3. Re-packetizing is “codec-specific” (sframe-specific)
4. Problem cannot be solved purely on the endpoints

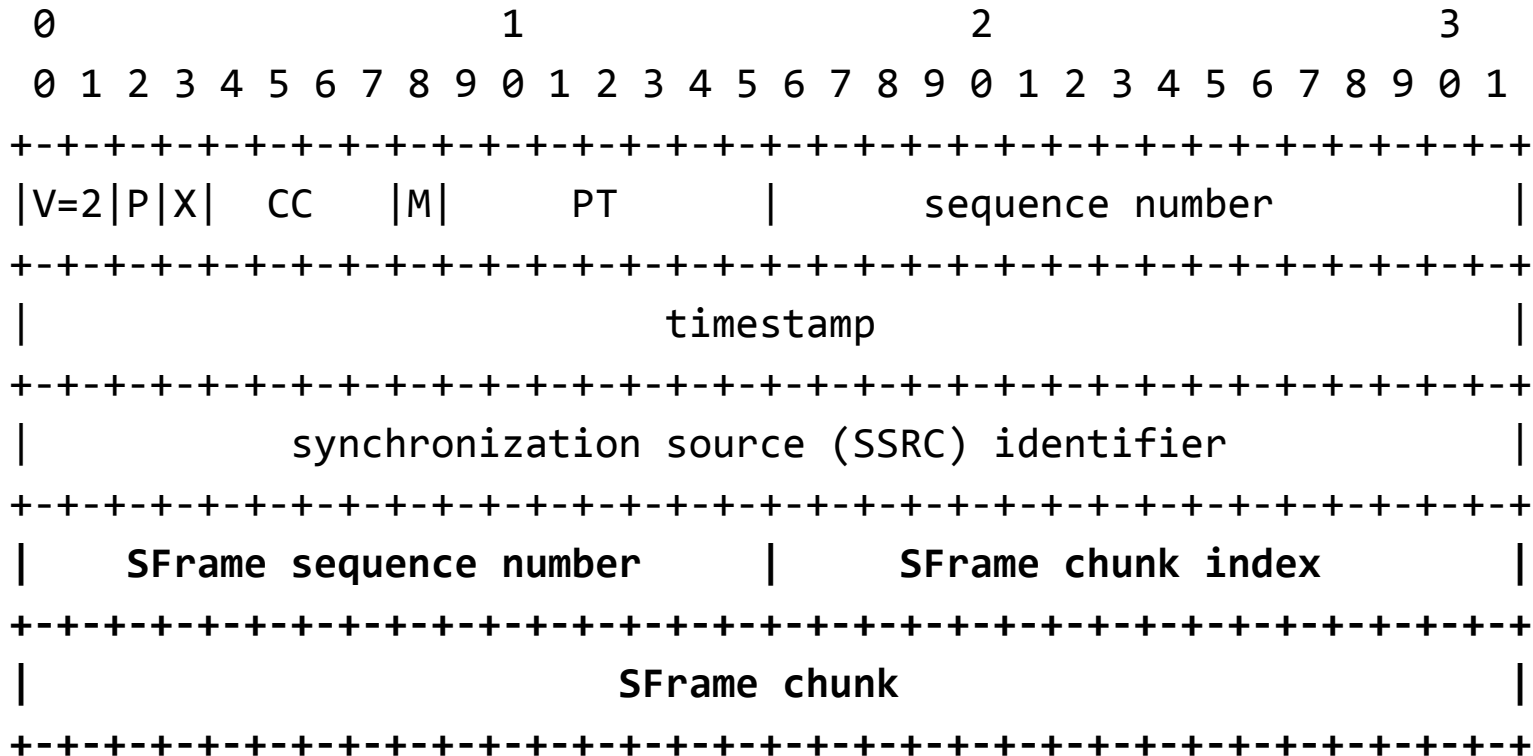
# Part 3: A Potential Solution

It's also simple

# RTP packet format



# RTP format + solution + SFrame



## Example

```
{seqnum=101, payload=ABCDEF}  
{seqnum=102, payload=GHIJKL}
```

Translates into (and back from):

```
{seqnum=1, sframe_seqnum=101, chunk_index=0, chunk=AB}  
{seqnum=2, sframe_seqnum=101, chunk_index=1, chunk=CD}  
{seqnum=3, sframe_seqnum=101, chunk_index=2, chunk=EF}
```

```
{seqnum=4, sframe_seqnum=102, chunk_index=0, chunk=GH}  
{seqnum=5, sframe_seqnum=102, chunk_index=1, chunk=IJ}  
{seqnum=6, sframe_seqnum=102, chunk_index=2, chunk=KL}
```

## Questions

Is this a direction worth pursuing for [Issue 29](#)?

If not, what's the solution for SFrame/SPacket/MTU-mismatch?

# Next Steps

- Decide how to proceed
  - As a PR for I-D.draft-ietf-avtcore-rtp-over-quirk?
  - As a separate draft?
- Discuss design tradeoffs
  - SFrame sequence number vs. frame counter
  - chunk index vs. “first” bit
  - “last chunk” in marker bit vs. payload
  - Duplicate header extensions vs not
  - A header for forward compatibility or not



# SFrame Packetization (if needed)

# RTP over QUIC

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

Mathis Engelbart, Jörg Ott

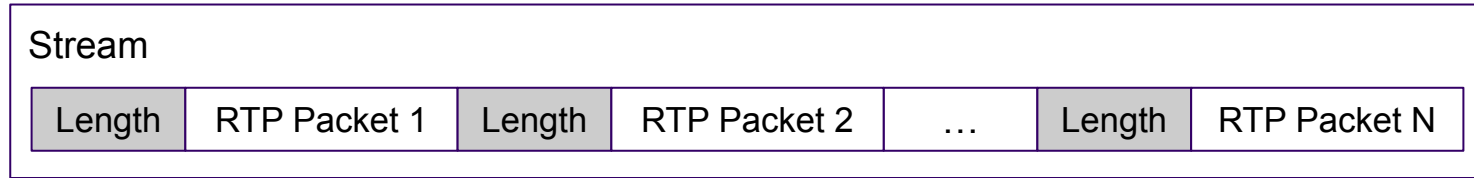
# New draft version -01

- Topology section to document considerations about different RTP topologies (unicast only, interop considerations for translators)
- Length field in QUIC streams
- Allow mixing streams and datagrams
- A couple of more smaller edits, see [full diff](#)

# Length Field



- Streams now include a length field and multiple RTP packets may be sent on the same stream:



- Allows sender-side segmentation of large ADUs into smaller packets
- Allows sending multiple ADUs per stream
- Problem: Receiver cannot abort stream when a packet is no longer needed, because it does not know which other packets follow on the same stream

# Issue #45: Canceling a stream



- The sender MAY send one or more ADUs per stream
  - Length fields in RTP/RTCP packet framing support this
  - Provides flexibility to the sender how to organize media transmission
- A stream may be canceled
  - if the ADU(s) are no longer needed (e.g., became obsolete)
- The sender may abort a stream: RESET\_STREAM
  - OK - because the sender knows which ADU(s) it put on the stream
- The receiver could also cancel a stream
  - Tricky - because the receiver doesn't really know what there is to come

# Issue #45: canceling a stream



- Possible solutions:
  - a. Accept that a receiver can't safely cancel streams
  - b. Require Single ADU (e.g., one media frame) per stream as in previous versions, but allow fragmentation into multiple RTP packets
  - c. Signaling to let receiver know what data will be sent on the same stream
  - d. Implicit from the application context?

# Issue #24/#31: Multiplexing



- Past discussions: How to signal the use of RTP-over-QUIC?
  - Which ALPN(s) to use?
  - Does the choice of an ALPN determine the multiplexing?
  - How much to do and define in this document?
- How to multiplex RTP-over-QUIC with non RTP streams and datagrams?
  - Prefixing explicit “session identifiers”?
  - How to ensure compatibility with other formats?
- Implications on the use of RTP-over-QUIC?
  - Want to remain lightweight and independent

# Issue #24/#31: Multiplexing



- Currently exploring an additional abstraction layer, to provide multiplexing
- Map RTP sessions to WebTransport sessions?
  - RTP/RTCP and other protocols can be multiplexed on same QUIC connection using multiple WebTransport sessions
  - No ALPN definition needed as this is managed by abstraction layer
  - Implication of using WebTransport: May run over HTTP/2 or any other protocol that will be defined for WebTransport (unless we require H3, but why would we?)
  - Naming: Is this still RTP over QUIC or should it be named RTP over WebTransport?
- BUT: Isn't this too limiting (given the implications of web transport)?
  - Currently discussing and preparing a PR for the draft



# Topologies (related to #29)



- Hopefully resolved by now for SFrame
- Checking further topologies considerations

# Next Steps



- Finish Multiplexing PR
- Reviewing for further detailed guidance
- Working further with Spencer on SDP signaling

# SDP for RTP over QUIC

*Spencer Dawkins*

[draft-dawkins-avtcore-sdp-rtp-quic](#) - a minimal specification

<https://github.com/SpencerDawkins/sdp-rtp-quic-issues> - includes discussion

# Background for this session

- [draft-dawkins-avtcore-sdp-rtp-quic](#)
  - a minimal SDP specification, tracking [draft-ietf-avtcore-rtp-over-quic](#)
  - Issues and PRs based on current text are welcome in this GitHub repo
- [Separate GitHub repo, used for broader issue tracking](#)
  - These issues aren't limited to [draft-ietf-avtcore-rtp-over-quic](#) scope
- Goal: update working group based on [draft-ietf-avtcore-rtp-over-quic-01](#)
  - Please feel free to provide feedback on the mailing list also
- **I'd like to request adoption of [draft-dawkins-avtcore-sdp-rtp-quic](#)**
  - Chairs asked if I wanted to request adoption at IETF 114 - I said "no"
  - I'd like to ask the working group questions about an **adopted** draft

# What AVP profiles to register

- This discussion has been **wild**
- The proposal history has been
  - "QUIC/RTP/SAVP", "QUIC/RTP/AVPF", and "QUIC/RTP/SAVPF"
  - ~~"QUIC/RTP/SAVP", "QUIC/RTP/AVPF", and "QUIC/RTP/SAVPF"~~
  - "UDP" prepended to /QUIC/RTP/AVPF"
  - "stream/", "dgram/" and "/shared" prepended to UDP/QUIC/RTP/AVPF
  - Also define TCP/QUIC/RTP/AVPF to allow use of TCP ICE paths?
  - Do people understand this last one? Do people agree?
- *This is **why** I'd like to adopt this draft, so chairs can make consensus calls*

# QUIC also does congestion control (#1)

- [draft-ietf-avtcore-rtp-over-quic-01](#) registers "rtp-quic" ALPN
  - Alternative: register "rtp-mux-quic"
  - So, the QUIC implementation at the other end will know this is RTP
- **Proposal is that SDP should carry ALPNs**
  - This allows use of multiple draft versions, experiments, etc.
- Question - do we also need AVP, or will everyone still use AVPF?
  - ***This loops back to [What AVP profiles to register ...](#)***

## How to ask for QUIC feedback - (#13)

- Want to allow implementations to ask QUIC for feedback
  - This is included in [draft-ietf-avtcore-rtp-over-quic-01](#)
- Goal: substitute transport layer feedback for RTCP feedback
  - Reduce overhead when possible
  - Mapping may require creativity (interpret lost packet as a NACK)
- What should SDP look like for this?
  - "enable-transport-layer-feedback", or something similar, or
  - name individual pieces of transport layer feedback
- Proposal at IETF 114 was "enable-transport-layer-feedback", or similar
  - This almost certainly needs more working group discussion

# Next Steps

- Request working group adoption of this draft and, if agreed, rename draft
- Agree on what AVP profiles we are defining (you might **think** I'm joking)
- Catch up on other [draft-ietf-avtcore-rtp-over-quic-01](#) SDP needs
- Submit updated draft
- Working further with Mathis, Jorg, and other working group participants



# Wrapup and Next Steps



- Action Items
  - Item 1
  - Item 2
- Next Steps
  - Step 1
  - Step 2

# Thank you

Special thanks to:

The Secretariat, WG Participants & ADs