

# AVTCORE WG

Virtual Interim Meeting

Thursday May 19, 2022

10:00 - 12:00 Pacific Time

17:00 - 19:00 UTC





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

Jabber Room: [avtcore@jabber.ietf.org](jabber:avtcore@jabber.ietf.org)

Meeting Link: <https://meetings.conf.meetecho.com/interim/?short=144e158a-fc0a-4d67-855c-0f50babca848>

# Virtual Interim Meeting Tips

**This session is being recorded**

- 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.

# About this meeting



- Agenda:  
<https://datatracker.ietf.org/doc/agenda-interim-2022-avtcore-02-avtcore-01/>
- Notes: Accessible from MeetEcho
- Jabber Room: [avtcore@jabber.ietf.org](mailto:avtcore@jabber.ietf.org)
- Secretariat: [mtd@jabber.ietf.org](mailto:mtd@jabber.ietf.org)
- WG Chairs: Jonathan Lennox & Bernard Aboba
- Jabber Scribe:
- Note takers:

# 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.

# IETF 114 Plans

- IETF 114 is a hybrid meeting.
- How many people are planning to attend in person?
- Please answer in the chat.

# Agenda



1. Note Well, Note Takers, Agenda Bashing, Draft status, IPR Declarations (Chairs, 10 min)
2. [Cryptex](#) (Sergio Garcia Murillo, 10 min)  
<https://datatracker.ietf.org/doc/html/draft-ietf-avtcore-cryptex>  
<https://github.com/juberti/cryptex/issues>
3. [Frame Marking RTP Header Extension](#) (Mo Zanaty, 5 min)  
<https://datatracker.ietf.org/doc/html/draft-ietf-avtext-framemarking>
4. [RTP Payload for SCIP](#) (Daniel Hanson, 10 minutes)  
<https://datatracker.ietf.org/doc/html/draft-hanson-avtcore-rtp-scip>
5. [RFC7983bis](#) (B. Aboba, 10 min)  
<https://datatracker.ietf.org/doc/html/draft-ietf-avtcore-rfc7983bis>
6. [QUIC congestion control for RTP](#) (D. Baldassin, 10 min)  
<https://datatracker.ietf.org/doc/html/draft-engelbart-rtp-over-quic>
7. [RTP over QUIC](#) (J. Ott, M. Engelbart, 15 min)  
<https://datatracker.ietf.org/doc/html/draft-engelbart-rtp-over-quic>
8. [WebTransport Liaison](#) (Will Law, 15 min)
9. [SDP for RTP over QUIC](#) (S. Dawkins, 10 min)  
<https://datatracker.ietf.org/doc/html/draft-dawkins-sdp-rtp-quic>
10. [RTP Payload for V3C](#) (Lauri Ilola, 10 min)  
<https://datatracker.ietf.org/doc/html/draft-ilola-avtcore-rtp-v3c>
11. [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)
- IETF Last Call Completed (April 5): Waiting for AD Go-Ahead:AD Followup
  - draft-ietf-avtcore-cryptex
- IETF Last Call Completed (May 19): Waiting for writeup
  - draft-ietf-avtcore-rtp-vcv
- Waiting for AD Go-Ahead:Revised I-D Needed
  - draft-ietf-avtext-framemarking
- WGLC completed: Revised I-D needed
  - draft-ietf-avtcore-rtp-scip (Completed May 8, 2022)
- Adopted
  - draft-ietf-avtcore-rtp-enc
  - draft-ietf-avtcore-rfc7983bis



# Cryptex IPR

- An IPR declaration has been submitted relating to draft-ietf-avtcore-cryptex:
  - <https://datatracker.ietf.org/ipr/5605/>
  - Holder legal name: Qualcomm Incorporated
- An IPR notice was sent to the WG on April 7, 2022:
  - <https://mailarchive.ietf.org/arch/msg/avt/erUF7hbJPzfQ8Zz9b0J9Q8rOV7U/>
- Are there objections to proceeding with this document?

# Completely Encrypting RTP Header Extensions and Contributing Sources (Cryptex)

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

<https://github.com/juberti/cryptex/issues>

Sergio Garcia Murillo

# Current Status

- Last call reviews :
  - SECDIR -
    - [Last Call Review of draft-ietf-avtc core-cryptex-05](#)
    - <https://github.com/juberti/cryptex/issues/46> [closed]
  - GENART
    - [Last Call Review of draft-ietf-avtc core-cryptex-05](#)
    - <https://github.com/juberti/cryptex/issues/47>
  - ARTART
    - [Last Call Review of draft-ietf-avtc core-cryptex-05](#)
    - <https://github.com/juberti/cryptex/issues/48>

# Current Status

- Last call reviews :
  - SECDIR
    - [Last Call Review of draft-ietf-avtc core-cryptex-05](#)
    - <https://github.com/juberti/cryptex/issues/46> [closed]
  - GENART
    - [Last Call Review of draft-ietf-avtc core-cryptex-05](#)
    - <https://github.com/juberti/cryptex/issues/47>
  - ARTART
    - [Last Call Review of draft-ietf-avtc core-cryptex-05](#)
    - <https://github.com/juberti/cryptex/issues/48>

# GENART Review

**Section 4:** *this document defines a new "a=cryptex" Session Description Protocol (SDP) [RFC4566] attribute to indicate support. Then the next sentence states that "This attribute takes no value". Why "no value"? The first statement already says a new "a=cryptex" attribute. It is confusing.*

- This document defines a new "a=cryptex" Session Description Protocol (SDP) [RFC4566] **property** attribute to indicate support. This attribute takes no value, and can be used at the session level or media level.
- This document defines a new "a=cryptex" Session Description Protocol (SDP) [RFC4566] attribute to indicate support. This attribute **is a property attribute and therefore** takes no value, and can be used at the session level or media level.

**Section 6.3:** *what does "region" mean in the statement? "The decryption procedure is identical to that of [RFC3711] except for the region to decrypt" Do you mean the "header" to be encrypted by the scheme described in this document*

- Use **Encrypted Portion** as in RFC3711 instead of **region to encrypt**  
<https://github.com/juberti/cryptex/pull/49>

# ARTART Review (I)

## Section 5.2. Receiving

*"The implementation MAY stop and report an error if it considers use of this specification mandatory for the RTP stream." This reads oddly to me, as if it was originally written with 'may' rather than 'MAY'. I think what is meant is more like the following: Alternatively, in the presence of extensions but the absence of a matching value, an implementation MAY signal that it requires use of this specification by stopping and signalling an error.*

When receiving an RTP packet that contains header extensions, the "defined by profile" field MUST be checked to ensure the payload is formatted according to this specification. If the field does not match one of the values defined above, the implementation MUST instead handle it according to the specification that defines that value. ~~The implementation SHOULD stop and report an error if it considers use of this specification mandatory for the RTP stream.~~

Alternatively, if the implementation considers the use of this specification mandatory and the "defined by profile" field does not match one of the values defined above, it SHOULD stop the processing of the RTP packet and report an error for the RTP stream.

<https://github.com/juberti/cryptex/pull/50>

# ARTART Review (II)

**6.1 Packet Structure** *I think this diagram combines parts of diagrams taken from 3711 (Section 3.1 Figure 1) and 8285 (section 4.2). The latter is an example, and as such the "length=3" in the 6th line of the diagram doesn't really belong in something labelled generically "the SRTP packet is protected as follows", which seems to imply that what follows is a template for all such packets.*

- "length=3" is a typo, fixed as part of <https://github.com/juberti/cryptex/pull/45>

*A number of acronyms are not glossed at first use, e.g. SRTP, SSRC, CSRC. If anyone reading this RFC can be expect to be familiar with them perhaps that's OK...*

- I think we don't need to do define those anachronisms

*Is there a line break or two missing [in the plain text version]*

- Typo, fixed.

# Next Steps

- Resolution of open Github issues and merge pending PRS
  - [Issue 47](#): GENART Review [PR exist]
  - [Issue 48](#): ARTART Review [PR exist]
- Draft update



# Frame Marking RTP Header Extension

*Mo Zanaty*

<https://datatracker.ietf.org/doc/html/draft-ietf-avtext-framemarking>

# Last Call Reviews

- [GENART review](#)
  - Clarify experiment scope, time, goal, IANA changes.
- [OPSDIR review](#)
  - Clarify SDP reference.
- [SECDIR review](#)
  - Security considerations should reference RFC 8285 for header integrity protection.
- [ARTART review](#)
  - TID/LID wording in sections 3.3.2-4 can be clearer like 3.3.1.
  - Remove unenforceable requirement: "The header extension values MUST represent what is already in the RTP payload."
  - Security considerations for privacy implications and traffic analysis.
  - Nits on formatting, title (add "video"), duplicated text.

# Next Steps



- Confirm responses to Last Call review on list and with reviewers.
- Draft update (to version -14)

# RTP Payload Format for the SCIP Codec

*Daniel Hanson*

*and*

*Michael Faller*

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

# SCIP RFC Background, Purpose, and Issue



- The Secure Communication Interoperability Protocol (SCIP) began in 1994 in the U.S. and includes NATO and NATO partners
  - SCIP is an application layer protocol which uses RTP as transport for negotiating secure session capabilities
- Most commercial network administrators and security personnel are not aware of SCIP
  - Can result in the SCIP media subtype “scip” being removed from the SDP
  - Without the SCIP media subtype, secure session establishment cannot proceed
- The SCIP Draft RFC went to WGLC on April 8, 2022
  - Comments received from the AVTCORE group have been reviewed and some will be incorporated into a new draft SCIP RFC

# Overview of SCIP RFC

- Devices using the signaling in the SCIP Draft RFC are presently deployed in products used by US and NATO on nation, tactical, and commercial networks
- Two media subtypes “audio/scip” and “video/scip” have been registered with IANA as RTP Payload Format Media Types
- The SCIP RFC is needed to provide additional information for these media subtypes that is needed by OEMs of network equipment
- An example mapping for both audio/scip and video/scip is:

```
m=audio 50000 RTP/AVP 96
```

```
a=rtpmap:96 scip/8000
```

```
m=video 50002 RTP/AVP 97
```

```
a=rtpmap:97 scip/90000
```

# Comments Received during “Last Call” and Authors Reply (1 of 3)



- [BA] The term "end-to-end security" has a specific meaning as used in SFRAME that doesn't seem to be what is intended here. For example, my understanding is that SCIP handles video conferencing via an MCU that is trusted by the endpoints, as distinct from an untrusted SFU that has no access to cleartext payloads in SFRAME. Similarly, isn't it the case that audio conferencing is handled via a mixer that requires access to cleartext payloads?
- Reply – (Section 2) SCIP conferencing would be point-to-multipoint using a trusted multipoint control unit (MCU). SFRAME (draft-omara-sframe-03.txt) is not under consideration for SCIP
- Instead of "end-to-end security" the term "application-layer security applied to the RTP payload" might make more sense.
- Reply – Existing text is “... end-to-end security at the application layer” which is the intent. No change is planned at this time

# Comments Received during “Last Call” and Authors Reply (2 of 3)



- [BA] The term "end-to-end bit integrity" does not have the same meaning here as in SFRAME so it could be confusing to readers. Since the statements below make clear what is needed, it seems like it could be removed with no loss of clarity
- Reply – (Section 3) The text could be modified to be:  
“The "scip" media subtype indicates support for and identifies SCIP traffic that is being transferred using RTP. *Transcoding and lossy compression techniques SHALL NOT be performed on the SCIP RTP payload.*”
- [BA] Is it correct to say that SCIP handles audio/video multiplexing at the application layer?
- Reply – (Section 5) Audio/video multiplexing (combining media into a single UDP port) is not under consideration for SCIP at this time and would be beyond the scope of the SCIP RFC.
- [BA] For example, would the SDP ever indicate use of BUNDLE (RFC 9143) with SCIP?
- Reply – BUNDLE is not under consideration for SCIP at this time



# Comments Received during “Last Call” and Authors Reply (3 of 3)



- [BA] What about RTP/RTCP multiplexing? Is this also handled at the application layer or is it negotiated in SDP?
- Reply – RTP/RTCP multiplexing (RFC 5761 – combining RTP and RTCP into a single UDP port) is independent of the SCIP protocol and is considered beyond the scope of the SCIP RFC
- [BA] What about interactions with feedback messages? In a SCIP video session would we see RTCP feedback negotiated?
- Reply – Extended RTCP feedback messages (RFC 8888) are independent of the SCIP protocol and are considered beyond the scope of the SCIP RFC

## Summary, Conclusions, and Questions

- Issues have occurred because OEMs of network equipment, network administrators and security personnel are unaware of SCIP and SDP contents necessary to establish an application layer secure session between SCIP devices
- The purpose of the SCIP RFC is to provide global access to information necessary to support SCIP
- “Last Call” ended on May 8, 2022
- Comments that have been received as a result of “Last Call” have been discussed and a revised draft will be published.
- What is the next step?
  - Directorate reviews

# RFC 7983bis

*Bernard Aboba*

*G. Salgueiro*

*C. Perkins*

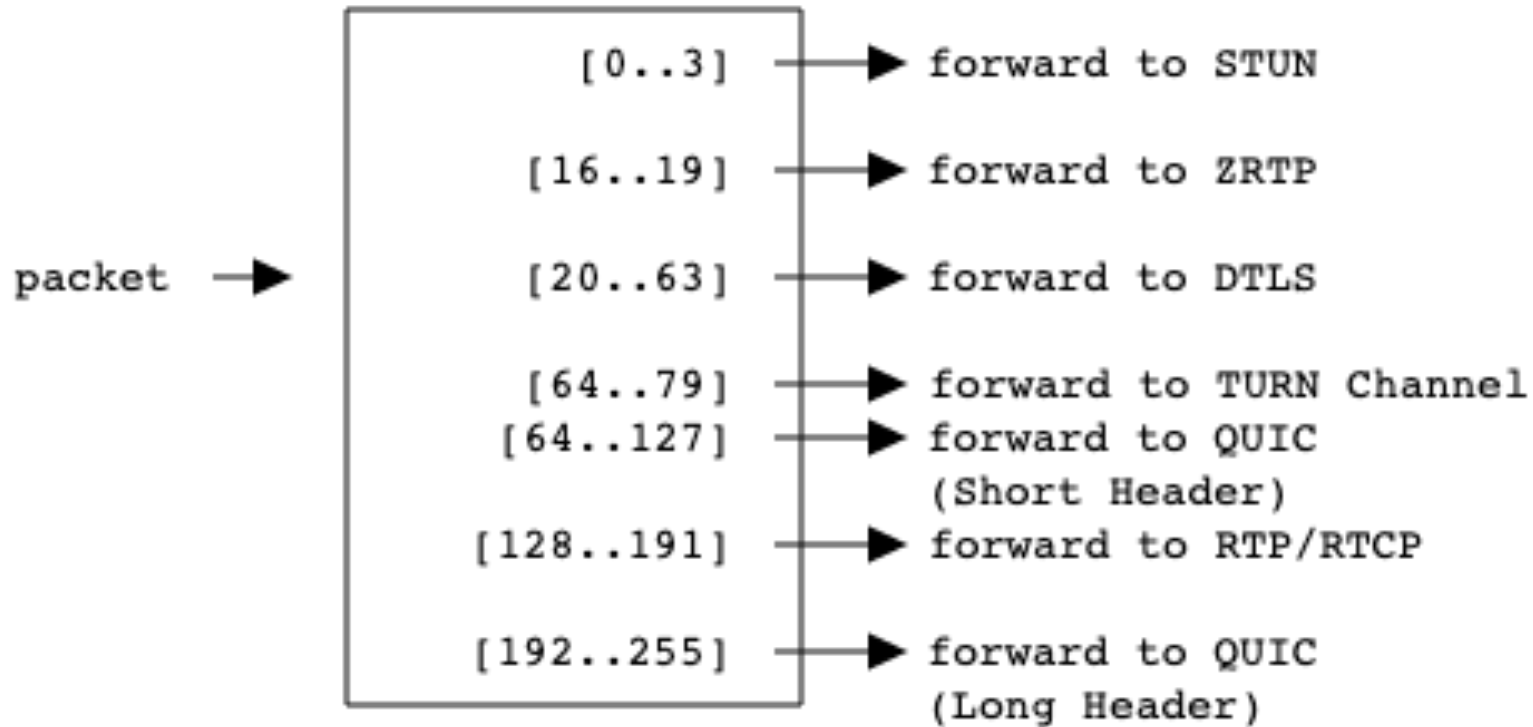
<https://datatracker.ietf.org/doc/html/draft-ietf-avtcore-rfc7983bis>

# RFC 7983bis



- Update to RFC 7983 Section 7, documenting QUIC multiplexing.
  - Description of multiplexing SRTP, SRTCP, STUN, TURN, DTLS, ZRTP and QUIC
  - Guidance on handling overlap between QUIC and TURN channels (not an issue in WebRTC).
  - Discussion of usage scenarios and multiplexing requirements
- Update to (D)TLS Content-Type Field IANA page to reference new RFC (no other change needed)

# RFC 7983bis (cont'd)



# RFC 7983bis-04



- Changes from -02 to -03:
  - Indication of compatibility with draft-ietf-quic-v2.
  - Reference updates:
    - RFC 9147 (DTLSv3)
- Changes from -03 to -04:
  - Addition of reference to draft-ietf-quic-bit-grease, removal of reference to draft-aboba-avtcore-quic-multiplexing.
  - Added paragraph in section 1:

The scheme described in this document is compatible with QUIC version 2 [I-D.ietf-quic-v2]. However, it is not compatible with QUIC Bit greasing, as defined in [I-D.ietf-quic-bit-grease]. Therefore, in situations where multiplexing is desired, QUIC Bit greasing MUST NOT be negotiated.

- Added note in Section 3:

Note: The demultiplexing of QUIC packets requires that QUIC Bit greasing [I-D.ietf-quic-bit-grease] not be negotiated.

# draft-ietf-quick-bit-grease Section 1



“The second-to-most significant bit of the first byte in every QUIC packet is defined as having a fixed value in QUIC version 1 [[QUIC](#)]. The purpose of having a fixed value is to allow intermediaries and endpoints to efficiently distinguish between QUIC and other protocols; see [[DEMUX](#)] for a description of a scheme that QUIC can integrate with as a result. As this bit effectively identifies a packet as QUIC, it is sometimes referred to as the "QUIC Bit".

Where endpoints and the intermediaries that support them do not depend on the QUIC Bit having a fixed value, sending the same value in every packet is more of liability than an asset. If systems come to depend on a fixed value, then it might become infeasible to define a version of QUIC that attributes semantics to this bit.

In order to safeguard future use of this bit, this document defines a QUIC transport parameter that indicates that an endpoint is willing to receive QUIC packets containing any value for this bit. By sending different values for this bit, the hope is that the value will remain available for future use [[USE-IT](#)].”

[[DEMUX](#)] references RFC 7983, not 7983bis.

# Next steps...



- Is the draft ready for WGLC?
- If not, what Issues remain?



# QUIC Congestion Control for RTP

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

<https://www.in.tum.de/fileadmin/w00bws/cm/papers/epiq21-rtp-over-quic.pdf>

David Baldassin

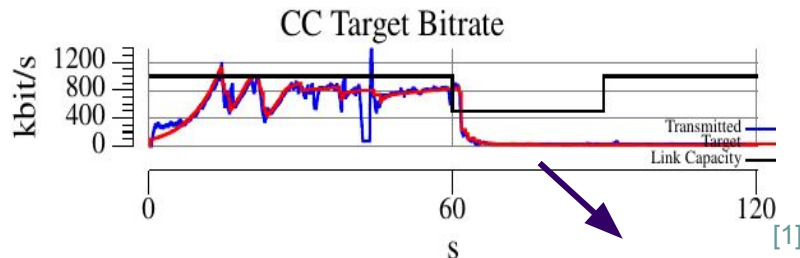
# Congestion control for real-time media over QUIC



- Paper investigates
  - SReAM over UDP
  - SReAM over QUIC with New Reno
  - SReAM over QUIC without New Reno
  - How to do avoid duplicate signaling between RTCP and QUIC statistics → SReAM w/o RTCP

# Our objective

- Investigate one observation from paper “When using SCReAM + NewReno and link capacity decreases, the transmitted bitrate drops to zero **and never recovers**”

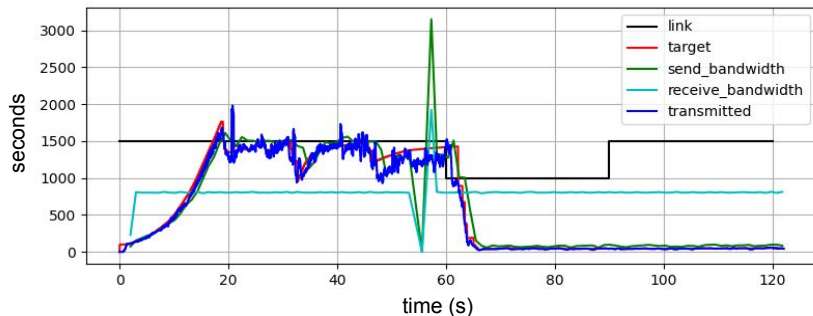


- Methodology
  - Same open source code
  - Same network emulation (tc and containers) with 30 seconds decrease of link capacity
  - Also consider constant rate scenario
  - Used a VBR video (visio conference capture with few movements)

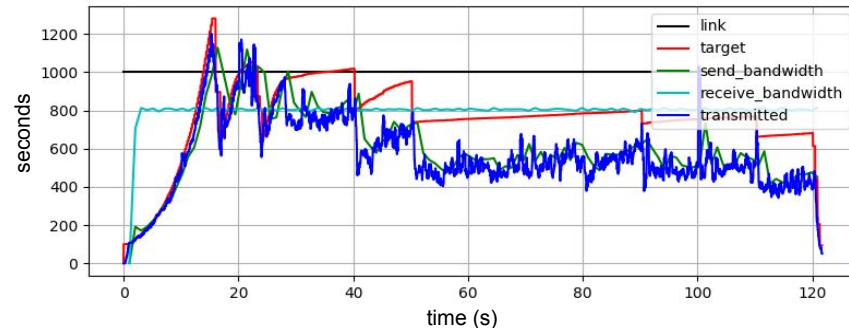
# Experimental results

- We observed the same behavior for SCReAM over New Reno
- We also observed it **randomly** for SCReAM over UDP

Initial rate at 1.5mpbs, changing to 1mpbs at 60s :

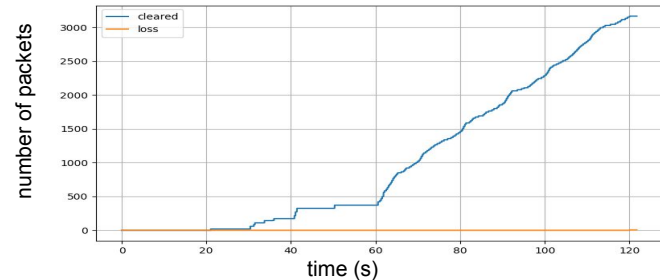
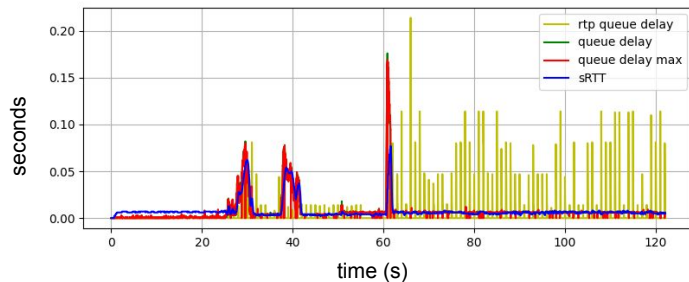


Constant rate of 1mpbs:



# SCReAM RTP queue

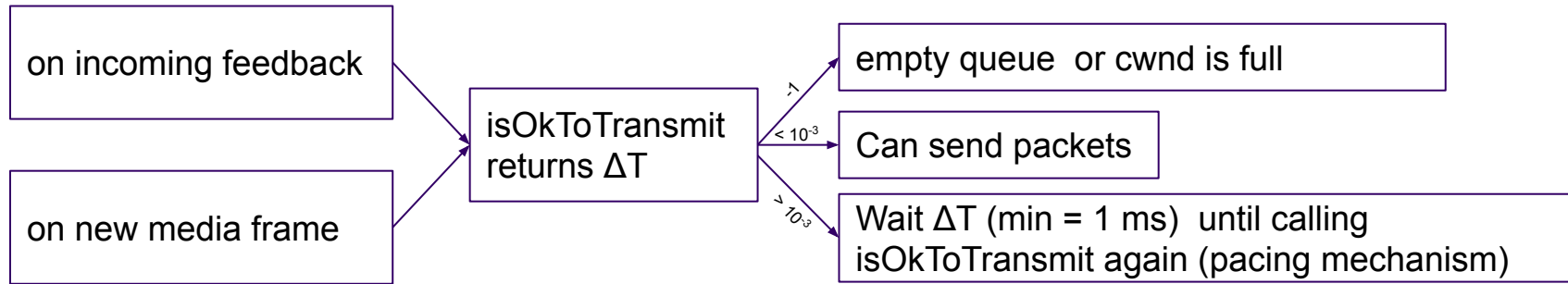
- According to RFC, SCReAM stores the RTP packets in a queue before sending
- The queue is cleared when it reaches the delay max value to mitigate long congestion events
- This mechanism explains the behavior under NewReno or UDP
  - Ok at  $t=60$  s when capacity decreases suddenly
  - **Why does it persist after  $t>90$ s?**



# SCReAM packet sending procedure



- OkToTransmit method called to check if we can send an RTP packet.
- in the implementation, the RTP packets were being queued and never sent, triggering SCReAM clearing queue mechanism

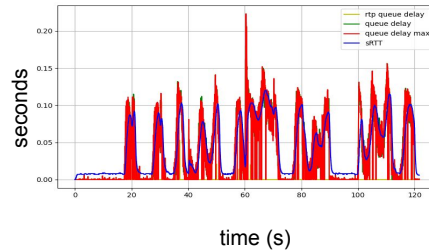
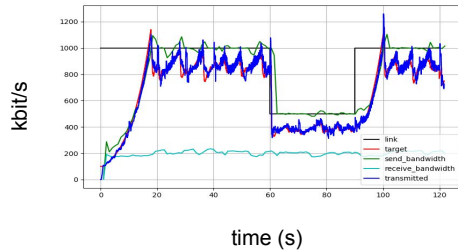


In implementation,  $\Delta T$  continuously  $> 1$  ms

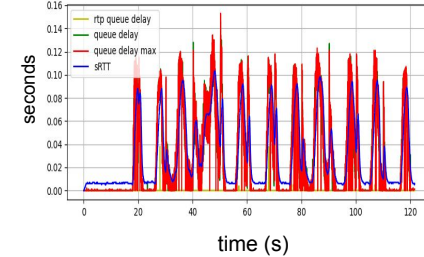
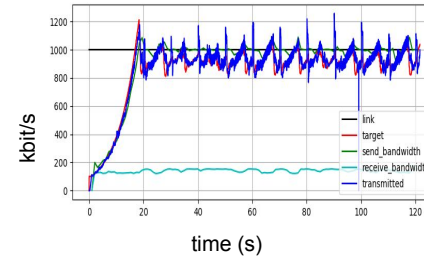
# New results : no more abnormal behavior



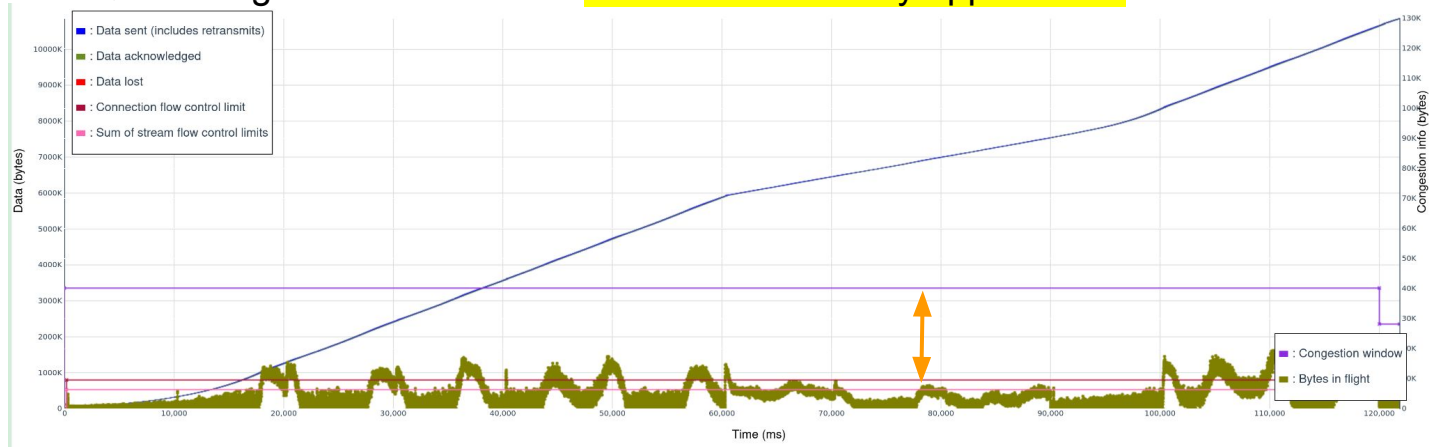
SCReAM and New Reno :



Constant rate scenario:

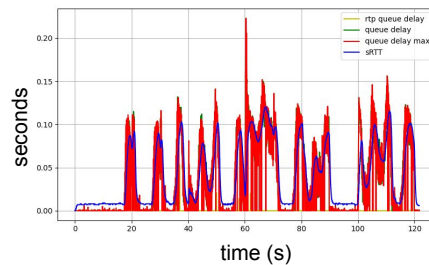
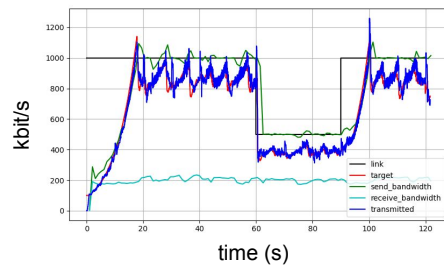


QUIC congestion statistics ⇒ **NewReno limited by application !**

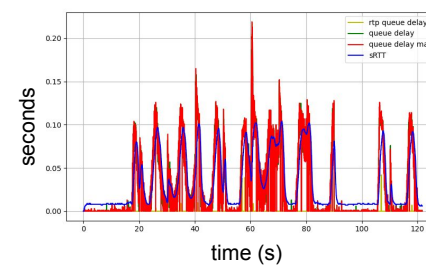
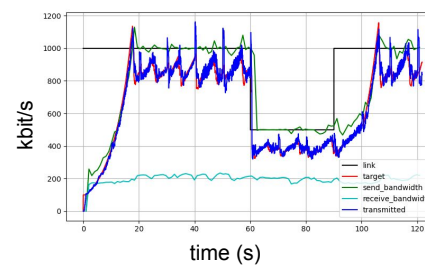


# New results

SCReAM and newreno :



SCReAM without newreno :





# Conclusion and Next Steps



Take-away:

1. Found a bug in the implementation
2. After fixing this bug, behavior of SCReAM+NewReno similar to SCReAM alone, and NewReno is always application limited.

Next steps:

- Additional tests in controlled environment
  - CC algorithms, different network scenarios and videos (CBR, VBR) and different QUIC implementations.
- Set up a new experiment to route webrtc traffic (sending from an unmodified webrtc client, chrome for example) through a QUIC tunnel and do network emulation.

# RTP over QUIC

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

Mathis Engelbart, Jörg Ott

# Encapsulation

- Earlier versions used QUIC Datagrams only
- Now added RTP over QUIC Streams
  - One Application Data Unit (ADU) per QUIC stream
  - Sender or Receiver Application can close stream if packet is no longer needed
- Common Encapsulation format:
  - Flow ID for demultiplexing different data streams
  - RTP packet
- How to handle RTCP with Stream-based RTP transmission
  - Employ the same statistics inference mechanisms
  - Carry additional RTCP packets also in one packet per stream
  - DO NOT use timing information from RTCP reception but QUIC stats instead
  - Should work without need for DATAGRAM support in a QUIC implementation

# RTCP

- Allow all RTCP by default
- Provide guidelines for how common RTCP packet types can be mapped to QUIC (RR, NACK, ECN, CCFB, XR)
  - Allow future specifications to provide mappings for other packet types
- SR and most application layer control packets cannot easily be mapped to QUIC
- Open Question: Only map RTCP to QUIC or include all data points from QUIC connection to improve statistics?

# Congestion Control

- Let sender choose how to do congestion control from different options:
- Congestion Control at QUIC layer
  - Let QUIC do congestion control, ideally low latency, delay-based
  - Expose bandwidth estimation from QUIC connection to application to allow dynamic codec target bitrate configuration
- Congestion Control at Application Layer
  - If no low-latency congestion control available in QUIC or bandwidth not exposed
  - Do congestion control/bandwidth estimation at application layer, e.g. using SCReAM, NADA, GCC
  - No true need for QUIC congestion control to be turned off (as per measurements so far): Reno seems feasible to work w/ application-limited traffic controlled by SCReAM or GCC
  - It MAY be useful to turn off QUIC congestion control (library implementation permitting)
- See section on RTCP for implementing congestion control feedback

# SDP Signaling

- Strawman SDP signaling included since -00 (less emphasis so far)
  - To reconcile with Spencer's draft (likely to move there)
  - <https://github.com/SpencerDawkins/sdp-rtp-quic>
- What to signal?
  - Initial QUIC connection establishment (who initiates, who accepts)
    - Known from TCP operation
    - Adding / removing flows to / from a connection
    - Zero-RTT support for connection setup?
      - To avoid media clipping
  - Media-to-Flow-ID mapping

# SDP Signaling (2)

- Issues to address
  - Choose STREAMs vs. DATAGRAMs
    - Receiver indicates capabilities to receive
    - Sender chooses
    - Baseline: DATAGRAM or STREAM?
  - Capability of statistics inference and export from QUIC
    - Turn off RTCP SR / RR
  - Congestion control choice can be done unilaterally
    - But feedback information must be sufficient
    - Can we negotiate just feedback (rather than CC algorithms)?

# Next steps

- Updates according to the points above
- Security considerations
- IANA registrations
  
- WG adoption?
  - Call for Adoption (CfA) to be issued once updated draft is submitted.



# W3C WebTransport WG Liaison

*Will Law*

# W3C WebTransport Request



- The W3C WebTransport WG has identified problems with bidirectional realtime audio/video communication over WebTransport, particularly: a client sending media to the server:
  - **Problem:** The client doesn't have enough information to know when it can reapply a multiplicative increase in the media send rate to recover from prior congestion.
  - **Requests:**
    - i. To know if RTP over QUIC can satisfy this use case.
    - ii. If so, what measurements can a ***browser make available to a JS client***, to assist with this problem.
    - iii. Will selectable congestion control be required? And if so, which algorithm(s)?

# Things to keep in mind

- WebTransport is a browser-based API that provides for:
  - Transport over HTTP/3 (QUIC reliable streams and datagrams)
    - Not tightly coupled to the QUIC stack.
    - Inter-process copy required for send/receive.
    - Path includes both WHATWG Streams queue (typically short) and QUIC send/receive queue.
    - API does not currently provide visibility into the transport layer: (e.g. no info on QUIC ACKs, ECN marking).

# Discussion in W3C and IETF

- Issues opened in W3C WebTransport WG:
  - [Issue 21](#): Access to congestion control and bandwidth estimation
  - [Issue 365](#): Pluggable Congestion Control
- IETF AVTCORE WG discussion relating to:
  - Performance of congestion control algorithms (this and previous meetings)
  - Metrics relevant to CC algorithms
    - RFC 8888
    - draft-engelbart-rtp-quic
  - [RMCAT WG work on CC algorithms](#)

# RFC 8888, Section 3

Based on an analysis of NADA [RFC8698], SReAM [RFC8298], Google Congestion Control [Google-GCC], and Shared Bottleneck Detection [RFC8382], the following per-RTP packet congestion control feedback information has been determined to be necessary:

**RTP Sequence Number:** The receiver of an RTP flow needs to feed the sequence numbers of the received RTP packets back to the sender, so the sender can determine which packets were received and which were lost. Packet loss is used as an indication of congestion by many congestion control algorithms.

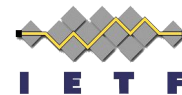
**Packet Arrival Time:** The receiver of an RTP flow needs to feed the arrival time of each RTP packet back to the sender. Packet delay and/or delay variation (jitter) is used as a congestion signal by some congestion control algorithms.

**Packet Explicit Congestion Notification (ECN) Marking:** If ECN [RFC3168] [RFC6679] is used, it is necessary to feed back the 2-bit ECN mark in received RTP packets, indicating for each RTP packet whether it is marked not-ECT, ECT(0), ECT(1), or ECN Congestion Experienced (ECN-CE). ("ECT" stands for "ECN-Capable Transport".) If the path used by the RTP traffic is ECN capable, the sender can use ECN-CE marking information as a congestion control signal.

**Every RTP flow is identified by its Synchronization Source (SSRC) identifier.** Accordingly, the RTCP feedback format needs to group its reports by SSRC, sending one report block per received SSRC.

As a practical matter, we note that host operating system (OS) process interruptions can occur at inopportune times. Accordingly, recording RTP packet send times at the sender, and the corresponding RTP packet arrival times at the receiver, needs to be done with deliberate care. This is because the time duration of host OS interruptions can be significant relative to the precision desired in the one-way delay estimates. Specifically, the send time needs to be recorded at the last opportunity prior to transmitting the RTP packet at the sender, and the arrival time at the receiver needs to be recorded at the earliest available opportunity.

# draft-engelbart-rtp-over-quic-02, Section 5.1



## 5.1. RTCP and QUIC Connection Statistics

Since QUIC provides generic congestion signals which allow the implementation of different congestion control algorithms, senders are not dependent on RTCP feedback for congestion control. However, there are some restrictions, and the QUIC implementation MUST fulfill some requirements to use these signals for congestion control instead of RTCP feedback.

To estimate the currently available bandwidth, real-time congestion control algorithms keep track of the sent packets and typically require a list of successfully delivered packets together with the timestamps at which they were received by a receiver. The bandwidth estimation can then be used to decide whether the media encoder can be configured to produce output at a higher or lower rate.

A congestion controller used for RTP over QUIC should be able to compute an adequate bandwidth estimation using the following inputs:

- `t_current`: A current timestamp
- `pkt_departure`: The departure time for each RTP packet sent to the receiver.
- `pkt_arrival`: The arrival time for each RTP packet that was successfully delivered to the receiver.
- The RTT estimations calculated by QUIC as described in [Section 5](#) of [\[RFC9002\]](#):
  - `latest_rtt`: The latest RTT sample generated by QUIC.
  - `min_rtt`: The minimum RTT observed by QUIC over a period of time
  - `smoothed_rtt`: An exponentially-weighted moving average of the observed RTT values
  - `rtt_var`: The mean deviation in the observed RTT values
- `ecn`: Optionally ECN marks may be used, if supported by the network and exposed by the QUIC implementation.

The only value of these inputs not currently available in QUIC is the `pkt_arrival`. The exact arrival times of QUIC Datagrams can be obtained by using the QUIC extension described in [\[draft-smith-quic-receive-ts-00\]](#) or [\[draft-huitema-quic-ts-05\]](#).

# Connection Statistics

<https://w3c.github.io/webtransport/>

```
dictionary WebTransportStats {  
  DOMHighResTimeStamp timestamp;  
  unsigned long long bytesSent;  
  unsigned long long packetsSent;  
  unsigned long long packetsLost;  
  unsigned long numOutgoingStreamsCreated;  
  unsigned long numIncomingStreamsCreated;  
  unsigned long long bytesReceived;  
  unsigned long long packetsReceived;  
  DOMHighResTimeStamp smoothedRtt;  
  DOMHighResTimeStamp rttVariation;  
  DOMHighResTimeStamp minRtt;  
  WebTransportDatagramStats datagrams;  
};
```

```
dictionary WebTransportDatagramStats {  
  DOMHighResTimeStamp timestamp;  
  unsigned long long expiredOutgoing;  
  unsigned long long droppedIncoming;  
  unsigned long long lostOutgoing;  
};
```

- Missing info
  - ecn, ACK info
  - latest\_rtt
  - pkt\_departure/pkt\_arrival

# Questions for the AVTCORE WG

- Can RTP over QUIC can satisfy the use-case of a client sending low latency A/V to a server over WebTransport?
- Do RTT measurements need to be adjusted?
  - QUIC stack measurements do not include application delays.
- Do departure/arrival times need to be adjusted?
  - pkt\_departure: when the packet leaves the QUIC send queue?
  - pkt\_arrival: when the packet arrives in the QUIC receive queue or when payload is consumed by reader.read()?
- Can a JS application ever implement sufficiently good congestion control (assuming it is supplied the correct stats) or will CC always need to be handled by the user-agent?
- Should the WebTransport connection in the browser be constructed with congestion control options, such as:
  - Default (optimized for throughput)
  - Low latency (optimized for realtime A/V from client -> server)
  - Circuit-breaker (application implemented CC with bounds)
  - Other ?



# 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 some ocean boiling

# Background for this session

- [draft-dawkins-avtcore-sdp-rtp-quic](#)
  - a minimal SDP specification, tracking [draft-engelbart-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-engelbart-rtp-over-quic](#) scope
  - (Obviously, I refocus if AVTCORE adopts [draft-engelbart-rtp-over-quic](#))
- I'd like feedback from the group on some proposed resolutions for -01
  - Please feel free to provide feedback on the mailing list also
- My goal is to issue a -01 before IETF 114 and ask for adoption at IETF 114
  - (Assuming that we've adopted [draft-engelbart-rtp-over-quic](#) by then)

# What AVP profiles to register - Issue 5 (1)

- -00 registered three profiles
  - "QUIC/RTP/SAVP", "QUIC/RTP/AVPF", and "QUIC/RTP/SAVPF"
- My proposal is now to register one profile - "QUIC/RTP/AVPF"
- Rationale: secure profiles for QUIC/RTP are ambiguous
  - Is "QUIC/RTP/SAVPF" saying "double encryption" (QUIC and SRTP)?
  - If not - why do we need to register secure profiles?
- In -00, "QUIC/RTP/SAVPF" did NOT mean "double encryption"
  - Only QUIC encryption was used, until traffic reached a middlebox
  - The middlebox would know to use a secure profile for non-QUIC links

## What AVP profiles to register - Issue 5 (2)

- If secure QUIC/RTP profiles only have meaning at middleboxes, we could
  - Try to figure out what that meaning is - but I don't think we know now
  - OR explicitly signal middleboxes, so they'll know what to do
  - OR require middleboxes to forward QUIC/RTP/AVPF securely (BCP?)
  - OR address this, and any related issues, in an [RFC 7667](#)-bis
  - Note that Issue 8 asks the broader question about [RFC 7667](#)
- We can register QUIC/RTP/AVPF now (PR is [#9](#))
  - We can register secure profiles later if we need to do that

# RTP over streams, datagrams, or both?

- This shows up in two related issues
  - What will RTP be using? (Issue [8](#))
  - Do we need to signal this in SDP (Issue [3](#))
- Are we going to be able to agree on streams, or on datagrams?
  - [draft-engelbart-rtp-over-quick](#) includes both now, FWIW
- If we can't pick one for RTP, we'll need to signal this in SDP
  - The “streams/datagrams/both?” question also comes up in MOQ
- My proposal now is to include both “stream” and “dgram” in the SDP

# QUIC also does congestion control (Issue 1)

- Our experience with nested congestion controllers hasn't been good
  - Both QUIC and RTP applications perform congestion control
- To get a QUIC implementation to change its HTTP/3 behavior, we might
  - ~~○ Explicitly ask the QUIC implementation to do something media friendly~~
  - ~~○ Explicitly ask the QUIC implementation to do SCReAM~~
  - Explicitly ask a QUIC implementation for specific feedback information
- OR we could explore whether this is a problem in practice
  - In admittedly limited testing, SCReAM over NewReno hasn't been
  - If we find out in additional testing, we can come back to this one.

# RTP Payload for V3C

<https://datatracker.ietf.org/doc/draft-ilola-avtcore-rtp-v3c/>

<https://github.com/laurilo/draft-ilola-avtcore-rtp-v3c>

Lauri Ilola

Lukasz Kondrad

# Background

- The topic was first introduced in the AVTCORE virtual meeting (15/02/22)
- Visual volumetric video-based coding (V3C) aims to re-use the existing 2d video coding technologies and it is video codec agnostic.
- V3C encoder decomposes volumetric frame into multiple components
  - video components (geometry, occupancy, attribute) that can be encoded by any video codec
  - atlas (metadata) component
    - provide information how to re-project the video components back into volumetric video frame
    - is encoded using mechanisms defined in ISO/IEC 23090-5
    - high-level syntax is represented as atlas NAL units that are very similar to HEVC/VVC.



# V3C RTP overview

- Video components can be streamed according to respective RTP payload specifications (e.g. H.264 - RFC6184, H.265 - RFC7798, etc.)
- Atlas component is missing RTP payload format.
- Defining V3C RTP payload format for NAL unit based atlas data
  - Encapsulation of atlas NAL units into RTP packets
  - V3C specific payload format parameters
  - Grouping mechanism (e.g. video component streams and atlas component stream can be grouped according to RFC5888)
  - Bundling of RTP streams according to RFC8843

# Feedback on draft-00

- AVTCore feedback on draft-00 ([#1](#))
  - Align with VVC instead of HEVC RTP payload format - Done
  - Remove Multi Time Aggregation Packets - Done
  - Clean payload format parameter section - Done
- Public feedback
  - Missing acronyms ([#3](#)) - Done
- Feedback managed on [Github](#)

# Next Steps

- Suggestions and feedback is much appreciated
  - Can be done directly on [Github](#)
- Continue to update the draft based on feedback
- Looking for additional authors/contributors
  - preferably someone with experience from RTP payload formats
- Topic introduced to 3GPP

Any questions or comments?

# Wrapup and Next Steps

- Action items
- Next steps

# Thank you

Special thanks to:

The Secretariat, WG Participants & ADs