AVTCORE WG
IETF 116

Hybrid Meeting
Tuesday, March 28, 2023
00:00 - 02:00 Eastern Time
04:00 - 06:00 UTC
Session II, G314 - G315

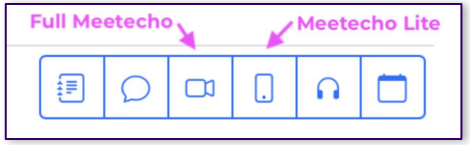
Mailing list: avtcore@ietf.org
Notes: https://notes.ietf.org/notes-ietf-116-avtcore
MeetEcho link: https://meetecho.ietf.org/conference/?group=avtcore

This session is being recorded

IETF 116 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

This session is being recorded

IETF 116 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 116 Yokahama

- Information about IETF 116
  https://www.ietf.org/how/meetings/116
- Agenda
  https://datatracker.ietf.org/meeting/agenda
- 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/116/faq/#covidmeasures

About this meeting

- Agenda: https://datatracker.ietf.org/doc/agenda-116-avtcore/
- Notes: https://notes.ietf.org/notes-ietf-116-avtcore
- Secretariat: mtd@jabber.ietf.org
- WG Chairs (Remote): Jonathan Lennox & Bernard Aboba
- Onsite: Harald Alvestrand
- Zulip Scribe: Jonathan Lennox
- Note takers: ?

Agenda

1. IETF 116 tips and resources, Note Well, Note Takers, Agenda Bashing, Draft status (Chairs, 10 min)
2. RTP Payload Format for SCIP (M. Faller, 10 min.)
   https://datatracker.ietf.org/doc/html/draft-ietf-avtcore-rtp-scip
3. RTP Payload Format for Essential Video Coding (EVC) (S. Wenger, 10 min)
   https://datatracker.ietf.org/doc/html/draft-ietf-avtcore-rtp-evc
4. RTP Payload Format for Volumetric Video Coding (L. Ilola, 10 min)
   https://datatracker.ietf.org/doc/html/draft-ietf-avtcore-rtp-v3c
5. RTP over QUIC (M. Engelbart, J. Ott, S. Dawkins, 30 min)
   https://datatracker.ietf.org/doc/html/draft-ietf-avtcore-rtp-over-quic
6. RTP Control Protocol (RTCP) Messages for Decoder Energy Reduction (S. Gudumasu, 10 min)
   https://datatracker.ietf.org/doc/html/draft-gudumasu-avtcore-decoder-energy-reduction
7. RTP Payload format for SFrame (P. Thatcher, 20 min)
   https://datatracker.ietf.org/doc/html/draft-ietf-sframe-enc
8. 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 9328: was draft-ietf-avtcore-rtp-vvc
  - RFC 9335: was draft-ietf-avtcore-cryptex
- RFC Editor Queue
  - draft-ietf-payload-vp9 (MISSREF)
- Approved-announcement-to-be-sent: AD Followup
  - draft-ietf-avtcore-rfc7983bis
  - draft-09 submitted, with editorial changes and an IANA note to update the TLS ContentType registry to point to 7983bis.
- IESG: AD Followup (3 DISCUSS positions)
  - draft-ietf-avtcore-rtp-scip
  - Ballot statements: https://datatracker.ietf.org/doc/draft-ietf-avtcore-rtp-scip/ballot/
  - Proposed changes to be discussed today.
- Waiting for AD Go-Ahead::Revised I-D Needed
  - draft-ietf-avtext-framemarking

Draft Status (cont'd)

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

RTP Payload Format for SCIP

draft-ietf-avtcore-rtp-scip

Dan Hanson

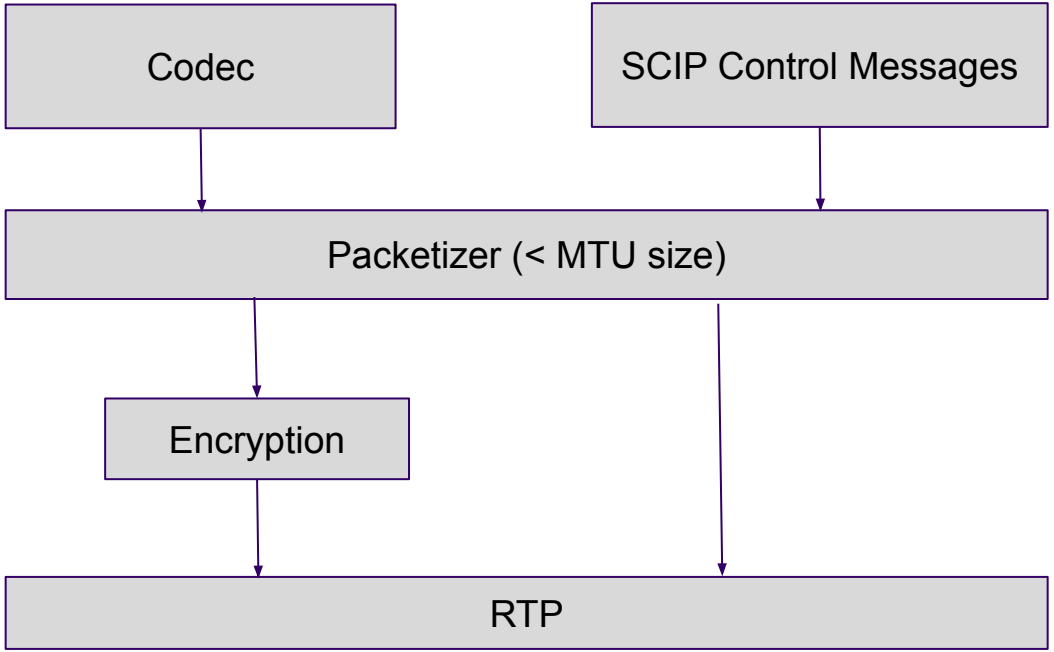
Mike Faller

RTP Payload for SCIP

- Preliminary draft 05 distributed to WG via email for review
  - Restated focus of document to be for network devices
    - Abstract, Introduction, Media Format, Payload Format sections
  - Re-worked Section 4 Payload Format
    - Added Figure to illustrate notionally how SCIP payloads are packetized to avoid IP fragmentation
    - Codecs payloads are packetized per their respective RFC
      - of the media codecs currently used by SCIP only H.264 has potential for fragmentation
    - SCIP Control Messages can be easily segmented by the SCIP Transport Layer since they contain an overall message length field
    - Explicitly stated that network devices MUST NOT filter or modify SCIP RTP packets
  - Other changes as discussed at Interim Meeting on February 23

RTP Payload for SCIP

SCIP RTP Packetization



RTP Payload for SCIP

- Next steps:
  - Formally submit rev 05 with comments received today
  - Request another review by IESG

RTP Payload Format for Essential Video Coding (EVC)

draft-ietf-avtcore-rtp-evc

S. Zhao

S. Wenger

Y. Lim

Summary, Status and Timeline

- -00 individual draft submitted 1/21, followed closely by -01
- -02, -03 submitted Feb 23, Mar 23. Two year delay while waiting for VVC payload (Now RFC 9328) to issue
- -03's normative content is complete and ready for WGLC
- Upon final review and based on private feedback, we will need a -04 version incorporating editorial improvements (including EVC characterization)
- Our working copy of -04 is ready and will be submitted ASAP.
- Media type review email msg drafted, ready to be sent
- We have no open questions to the WG, but thought a quick refresher may be helpful
- We request WGLC on -04

Changes since -01

- Media type registration
  - Largely based on lessons learned from RFC 9328
  - As EVC is basically a subset of VVC, so is the media type registration.
  - Related: corresponding changes/simplifications in the O/A section
- Alignment with RFC 9328
  - “Boilerplate”: congestion control, security
  - Feedback message support–only PLI and FIR supported per current implementation practice – yell if you think we need SLI or RPSI
  - Numerous editorial updates lifted from RFC 9328
- Changes in the packetization rules: mostly simplifications resulting from EVC’s lack of spatial scalability support

RTP Payload Format for Volumetric Video Coding (V3C)

draft-ietf-avtcore-rtp-v3c

L. Ilola

L. Kondrad

Editorial updates to the draft

- Couldn't get the new version uploaded in the system before the deadline (v0 -> v1)
- Below are the most notable proposed changes in text
- Full diff is available here
- Mostly just fixing typos and writing style

ISO/IEC International Standards 23090-5 [ISO.IEC.23090-5] defines encoding and decoding processes of volumetric video which leverages 2D video coding technologies. V3C encoding of volumetric frame is achieved through a conversion of volumetric frame from its 3D representation to multiple 2D representations and a generation of associated data documenting such conversions and transformations. The associated data, also known as atlas data, is necessary to define how to reproject the 2D representations back into 3D volumetric frame.	ISO/IEC International Standards 23090-5 [ISO.IEC.23090-5] enables encoding and decoding processes of volumetric video which utilizes 2D video coding technologies and associated data. V3C encoding of volumetric frame is achieved through a conversion of volumetric frame from its 3D representation to multiple 2D representations and a generation of associated data.
Generally, it is useful to signal V3C parameter set out-of-band, because it describes what overall resources are needed to decode and reconstruct the associated V3C bitstream. Signalling it dynamically as part of an RTP stream might result in undefined behaviour when receiver does not have the required capabilities to decode the received video component sub-bitstreams or when reconstruction process relies on information that the receiver does not support.	

What's next?

- It would be good to get more feedback
  - Spec is available here, feel free to create issues and pull requests
- How can we progress the draft?
  - The authors believe that the draft as technically stable
  - Only editorial improvements have been introduced
- There are relevant 3GPP work items looking at volumetric media
  - For now only DASH and ISOBMFF based file delivery are considered
  - For real-time delivery, the V3C RTP payload format would need to reach mature status in IETF

RTP over QUIC

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

https://datatracker.ietf.org/doc/draft-dawkins-avtcore-sdp-rtp-quic/

https://datatracker.ietf.org/doc/draft-dawkins-avtcore-sdp-rtp-quic-issues/

Mathis Engelbart, Jörg Ott, Spencer Dawkins

Congestion Control

- #59: Avoid nested congestion control loops
- If you're using RTP, you're application-limited (right?)
- If you're using RTP over QUIC, you want to rely on signals from QUIC CC
- We will add guidance (guidance!) about using application-level CC
- We will reference #48 on QUIC feedback replacing (some of the) AVP/AVPF feedback
- We will add guidance (guidance!) about using appropriate QUIC CC

Congestion Control

- #67: Phrase discussion in terms of guidance, not normative should/musts
- This is a meta-issue - we'll handle this elsewhere

Congestion Control

- #68: Remove mentions about “disabling QUIC congestion control”, etc.
- We can note that a sender/QUIC stack **can** do this, acting on its own
- We're saying "don't do this", which is getting close to RFC 8085 territory
- We can say "if you break it, you own it"
- Do we need to say more?

Congestion Control

- #69: Explain why we aren’t making BCP recommendations for algorithms
- We'll note that all the RMCAT protocols are still Experimental
  - Much analysis focused on avoiding "self-congestion"
  - Less analysis focused on interaction with other congestion controllers
- We can reasonably expect new algorithms to be produced
- Providing guidance about what to look for in new algorithms is useful

Congestion Control

- #70: More clearly define what we mean by “QUIC CC”, etc.
- PR #73
- Two meanings (used interchangeably, causing confusion)
  - Anything that happens below the application level
  - What current QUIC implementations are likely to do
- Again, we're defining what to look for, rather than naming algorithms

STOP_SENDING (#45)

- STOP_SENDING allows receivers to ask senders to reset a stream

- Different use cases for RTP over QUIC
  - Cancel frames which are too late
  - Unsubscribing from SVC layers

- Requires receiver side knowledge (and possibly signaling) of what data will be sent on a stream
  - Exactly one frame per stream to avoid accidentally canceling a following frame
  - SVC layers need to be segregated in different streams to allow unsubscribing

STOP_SENDING (#45)

- One frame per stream could lead to opening many streams very fast for Codecs which produce small frames (e.g., audio)

- Sender may already have knowledge and may be able to make better decisions of when to cancel a stream/remove SVC layer

- Signaling could still define how senders could let receivers know what to expect on a given stream

MAX_STREAMS (#49)

- MAX_STREAMS limit could be exceeded in scenarios with many RTP streams that each send multiple frames concurrently

- Current draft says:

  The number of packets that have to be transmitted concurrently depends on several factors [...]. Receivers are responsible for providing senders with enough credit to open new streams for new packets at any time.

- Background data flows may also use some of the credit and may need to be rate limited to avoid using **all** of the credit (only applies to unidirectional streams)

Disabled Datagrams (#13)

- This is pretty well discussed in the issue (thanks, Mathis and Lucas)
- We will work from the summary here, which is basically
  - If both sides don't support datagrams/provides zero credits for datagrams, you won't be using datagrams (duh),
  - So you're going to be using streams, and we can make suggestions that will help this work better.
- We will add details from other comments on the issue
- Lucas has raised a related question, about how to tell a QUIC implementation what to expect an incoming connection to do
  - This requires more analysis and discussion

Supported Topologies (#47)

- PR #72
- This resulted in a table of RFC 7667 topologies, like
  - RFC 7667 section number
  - RFC 7667 shortcut name
  - Supported by RTP-over-QUIC?
  - If "possibly", what are the qualifications?
  - If "no", what are the blocking factors?
- This could UPDATE RFC 7667, if that was The Right Thing To Do
- This could reasonably be its own draft, if we explain things in detail

AVP/AVPF Feedback (#48)

- PR #71
- This resulted in a table of RTCP packet and feedback types with
  - Document number
  - Can QUIC provide the same/similar feedback?
  - What is required to provide the same feedback

RTP Control Protocol (RTCP) Messages for Decoder Energy Reduction

draft-gudumasu-avtcore-decoder-energy-reduction

S. Gudumasu (InterDigital), F.Aumont (InterDigital), Edouard Francois (InterDigital), Christian Herglotz (FAU)

Background

- Energy Efficient Media Consumption (Green Metadata) was specified by MPEG in ISO/IEC 23001-11
- Two methods for decoder power reduction
  - Complexity metrics via SEI message
  - Interactive signaling
- Complexity metric signaling from encoder to decoder
  - Metrics related to coding tools (macro blocks, intra predicted blocks, filtering etc..)
- Interactive signaling (From decoder to encoder)
  - Spatial and temporal resolution request [1]
  - Coding tools-based request

[1] draft-ietf-avtcore-rtcp-green-metadata

Problem statement

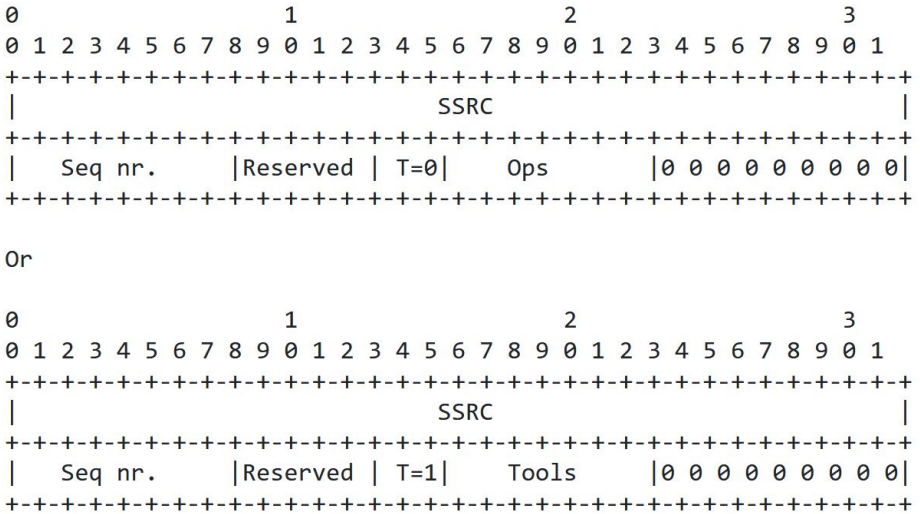
- Reducing **video decoder energy usage** during media consumption
  - How to signal the percentage of decoder power reduction required by the receiver?
  - How to signal the desired coding tools configuration to reduce decoder side power consumption by the receiver?
  - How to inform a receiver about changes to decoder power reduction percentage and/or the coding tools used by the sender/encoder?
  - Required signaling between endpoints to negotiate the above capabilities.

Proposal

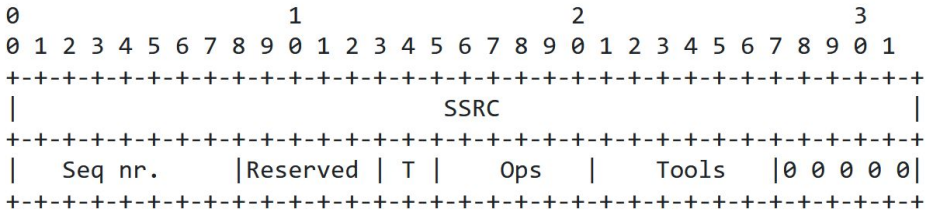
- Usage of the interactive signaling of metadata specified by MPEG in **ISO/IEC 23001-11** to:
  - Reduce decoder operations
  - Enable/disable coding tools
- Propose to extend RTCP feedback messages defined in AVPF [RFC4585][RFC5104] with 2 new RTCP feedback messages:
  - Decoder Operation Reduction Request (DORR)
  - Decoder Operation Reduction Notification (DORN)
- Can be used concurrently with the RTCP feedback messages defined in draft-ietf-avtcore-rtcp-green-metadata

RTCP feedback messages

- DORR RTCP message



- DORN RTCP message



SDP signaling

- Extension of rtcp-fb Attribute in SDP
  - **rtcp-fb-ccm-param =/ SP "dorr"** ; Decoder Operation Reduction Request
- Example

SDP Offer

```
v=0;
o=alice xxxxx
s=Offer/Answer
m=video 49170 RTP/AVP 98
a=rtpmap:98 H266/90000
a=fmtp:98 profile-id=1;
sprop-vps=<"video parameter sets data">;
sprop-sps="<"sequence parameter set data">;
sprop-pps=<"picture parameter set data">;
a=rtcp-fb:98 ccm fir
a=rtcp-fb:98 ccm dorr
```

SDP Answer

```
v=0;
o=alice xxxxx
s=Offer/Answer
c=xxxx
m=video 49170 RTP/AVP 98
a=rtpmap:98 H266/90000
a=rtcp-fb:98 ccm dorr
```

Next steps

- Gathering feedback
- Discussions in AVTCORE Reflector
- Update and Review
- Seek adoption when ready

RTP Payload Format for SFrame

draft-ietf-sframe-enc

Peter Thatcher

SFrame Architecture

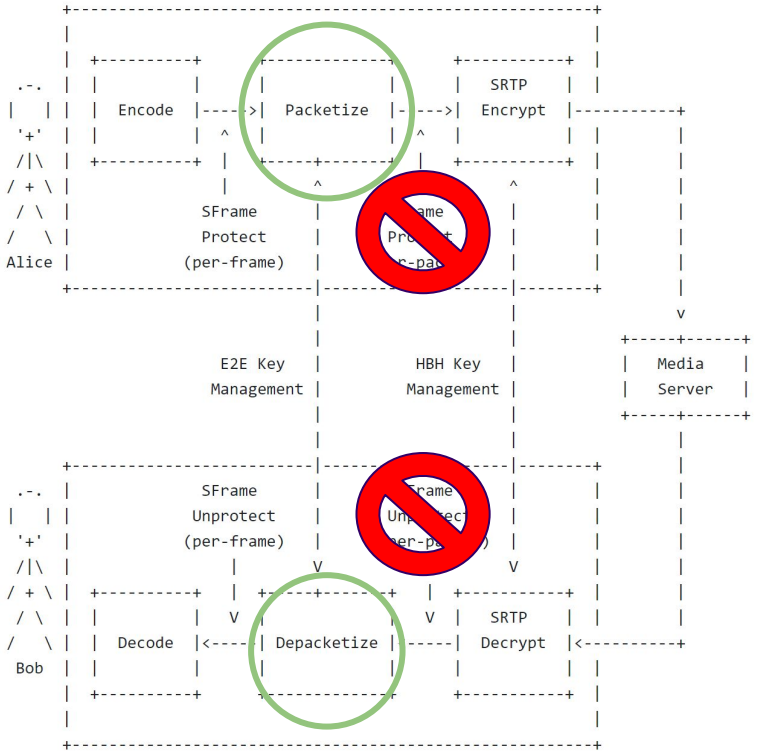
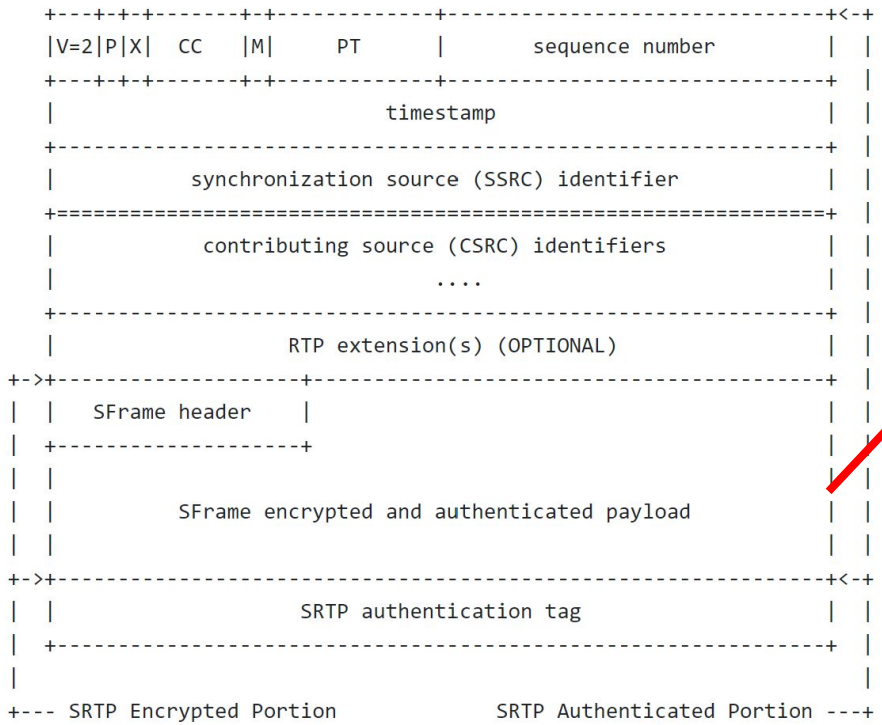


Figure 2

Source: draft-ietf-sframe-enc-01, Page 7

draft-ietf-sframe-enc-01



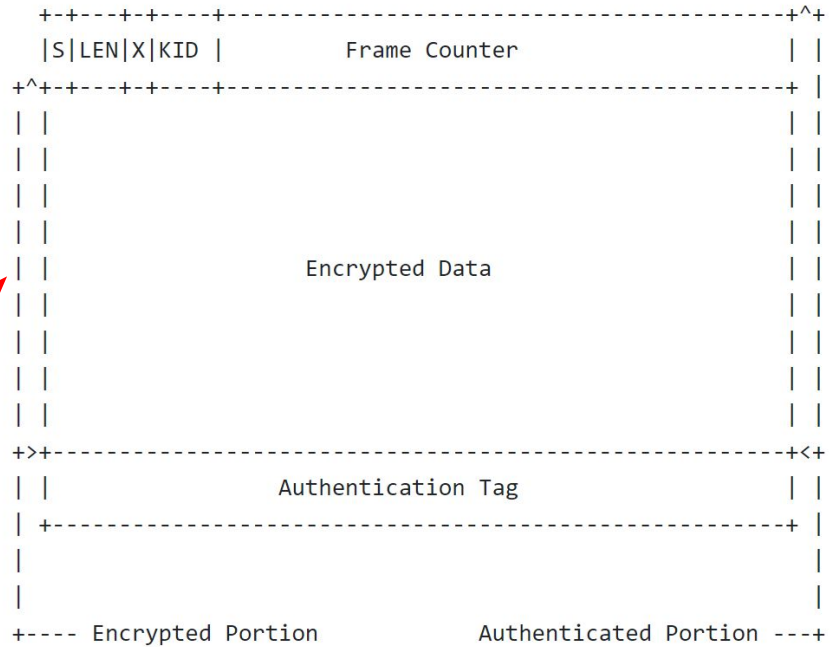
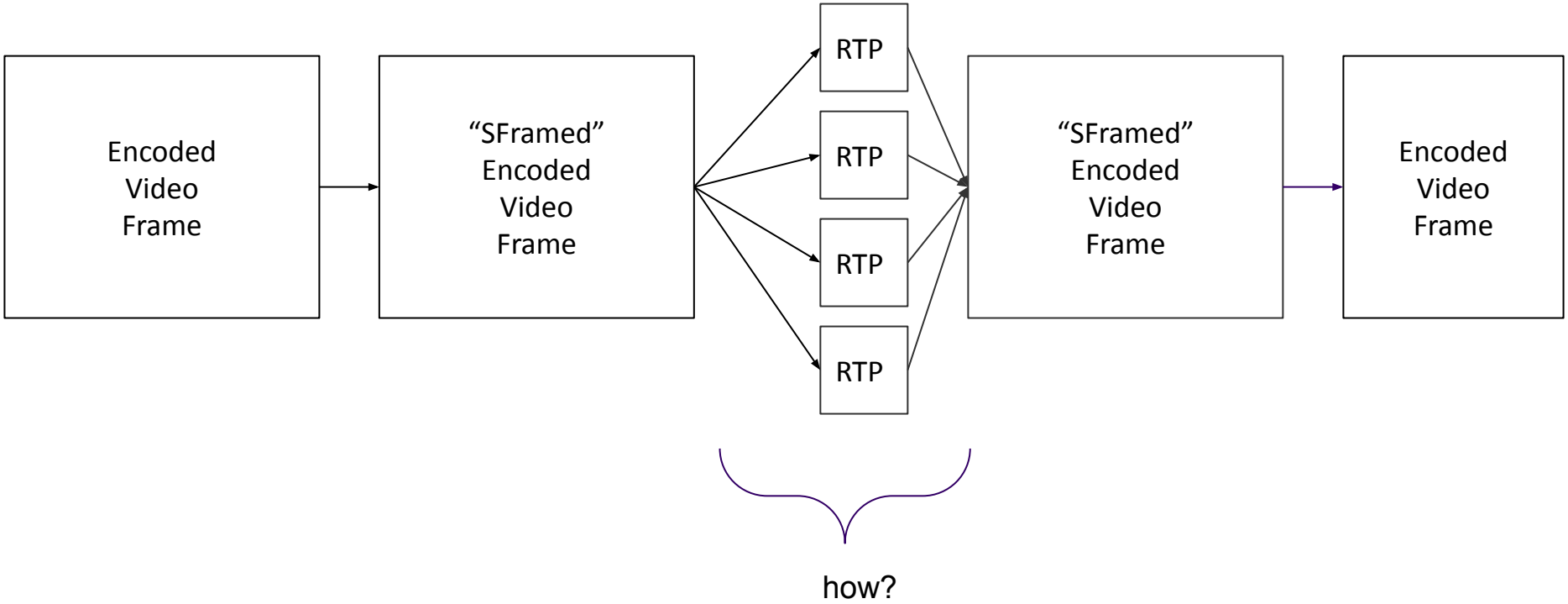


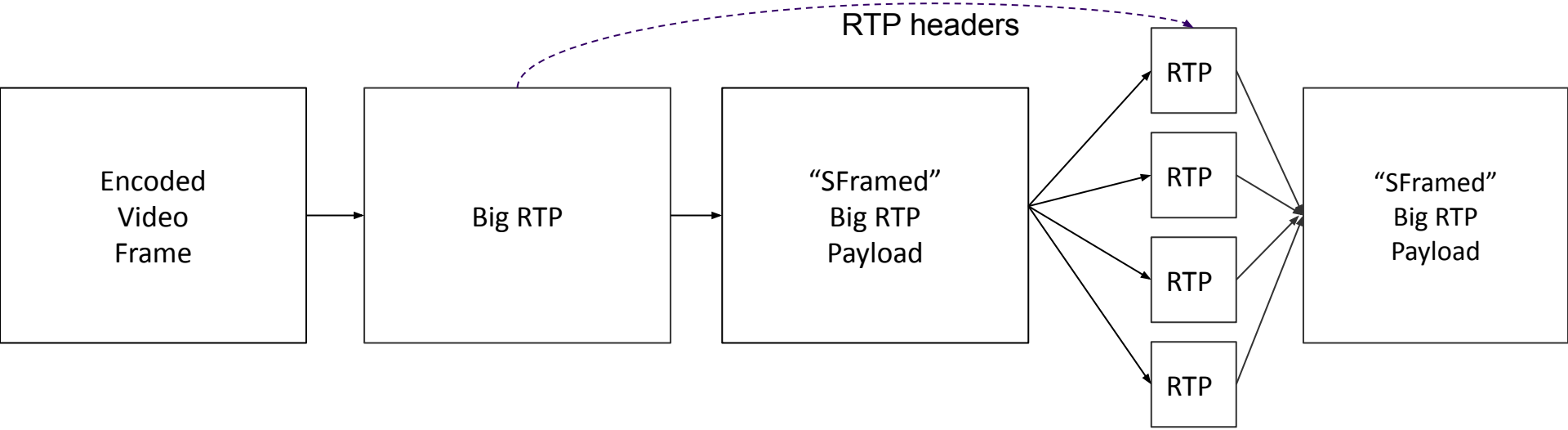
Figure 1: SRTP packet with SFrame-protected payload

Sources: Page 4, Page 8

Problem: 1 frame to N packets (and back)



Solution: packetize twice



Questions with answers

1. What goes in the RTP header?
2. What goes in the RTP payload?
3. How does packetization work?
4. How does depacketization work?

SFrame RTP Payload Format Proposal

(from Youenn, Peter, Lennox)

1. What goes in the RTP header?
   a. Payload Type = An SFrame Payload Type (one for each clockrate needed)
      For example, with SDP:

```
m=audio 50000 RTP/SAVPF 96
a=rtpmap:96 sframe/48000
m=video 50002 RTP/SAVPF 97
a=rtpmap:97 sframe/90000
```

   b. SSRC, timestamp, marker bit, header extensions: From “Big RTP”
   c. sequence number: incremented like normal

SFrame RTP Payload Format Proposal

(from Youenn, Peter, Lennox)

2. What goes in the RTP payload?
   a. The Payload Type from the “Big RTP” header
   b. A *fragment* of the “SFramed” “Big RTP” Payload
   c. An optional intra-frame sequence number (AKA fragment ID) or intra-frame offset (AKA fragment offset)

SFrame RTP Payload Format Proposal

(from Youenn, Peter, Lennox)

3. How does packetization work?
    a. Packetize with the codec's packetizer into one "Big RTP" packet
    b. "SFrame" the "Big RTP" payload
    c. Fragment the "SFramed" "Big RTP" payload
    d. Create RTP packets from the fragments
        i. The fragment is the payload
        ii. Copy SSRC, timestamp, header extensions, marker bit from "Big RTP" packet
        iii. Use the Payload Type for SFrame

SFrame RTP Payload Format Proposal

(from Youenn, Peter, Lennox)

4. How does depacketization work?
    a. Collect the RTP packets for a given SFrame
    b. Concatenate the SFrame fragments from the RTP payloads
    c. Remove the prefixed “Big RTP” Payload Type
    d. Unwrap/decrypt SFrame into “Big RTP” Payload
        i. Add the RTP header from the first RTP packet
        ii. Replace the Payload Type with the “Big RTP” Payload Type
    e. Pass the newly created “Big RTP” to the codec-specific depacketizer

Questions not yet answered

- How does the depacketizer know which packets go with which SFrame?
  - We probably need a “Frame ID”, along with a “this is the first first fragment”, and “this is last fragment”
- Do *all* RTP header extensions go in *all* RTP packets or just the first?
  - Probably necessary: MID
  - Probably not necessary: video rotation
- Does the “Big RTP” Payload Type go in *all* RTP packets or just the first?
- How do we indicate the presence of optional data?
  - There is a bit in front of the “Big RTP payload type” available
- What sequence number should be passed to the codec-specific depacketizer?
  - Does a depacketizer care?

The Big Question

If we make a draft that matches this, will it be adopted?

Discussion

- Your comments go here!

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