

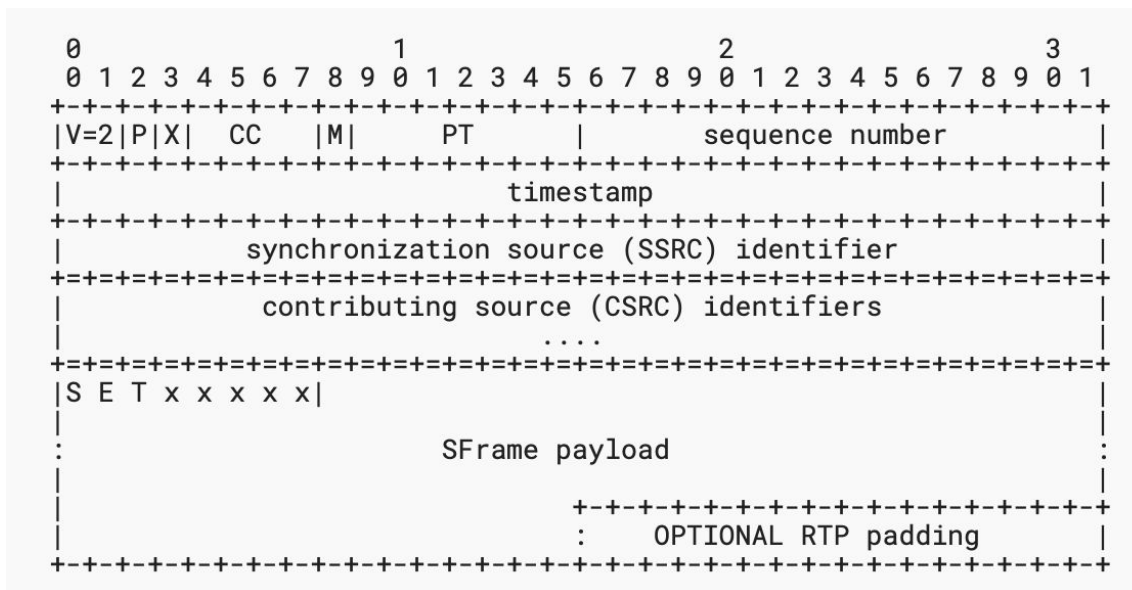
SFrame RTP Packetization

<https://www.ietf.org/archive/id/draft-ietf-avtcore-rtp-sframe-02.html>

Youenn Fablet

New published version (1/3)

A new T bit for raw (0) or packetized (1) content



New published version (2/3)

- Asymmetry between sender and receiver
 - Sender decides how to split a frame and whether to packetize before encryption
 - Receiver follows decision from sender
- Spec defines 4 algorithms
 - Sender generic generation of SFrame RTP packets
 - Per-frame SFrame sending ($T = 0$)
 - Per-packet SFrame sending ($T = 1$)
 - Receiver processing of a RTP packet

Per-ssrc key derivation (3/3)

- New spec section to allow derive keys for each SSRC
 - Use is optional, up to the application

```
ssrc_key = HKDF-Expand(HKDF-Extract(SSRC, base_key), "SFrame 1.0 RTP Stream", CipherSuite.Nh)
```

- SFrame ratcheting integration
 - Same as RFC 9605, with ssrc derived key as base key

W3C API side

- Sender side
 - Per-frame/per-packet SFrame sending
 - Application decides which flavour to use
 - Native support for both
 - JS support limited to per-frame sending
- Receiver side
 - Native support for all configurations
 - JS support limited to raw content for now

Implementation Efforts

- Libdatachannel prototyping
 - Conducted as part of Matter WG
 - Per-frame sending/receiving only for now
- Libwebrtc prototyping
 - Might start when possible
 - Might support both per-packet and per-frame

Next Steps

- Finalize document
 - Algorithm simplifications
 - RTP header extension generation
 - In case of per-frame encryption or refragmentation
 - Implementation feedback gathering