

MPEG-TS - Why is it so hard?

GStreamer Conference 2026

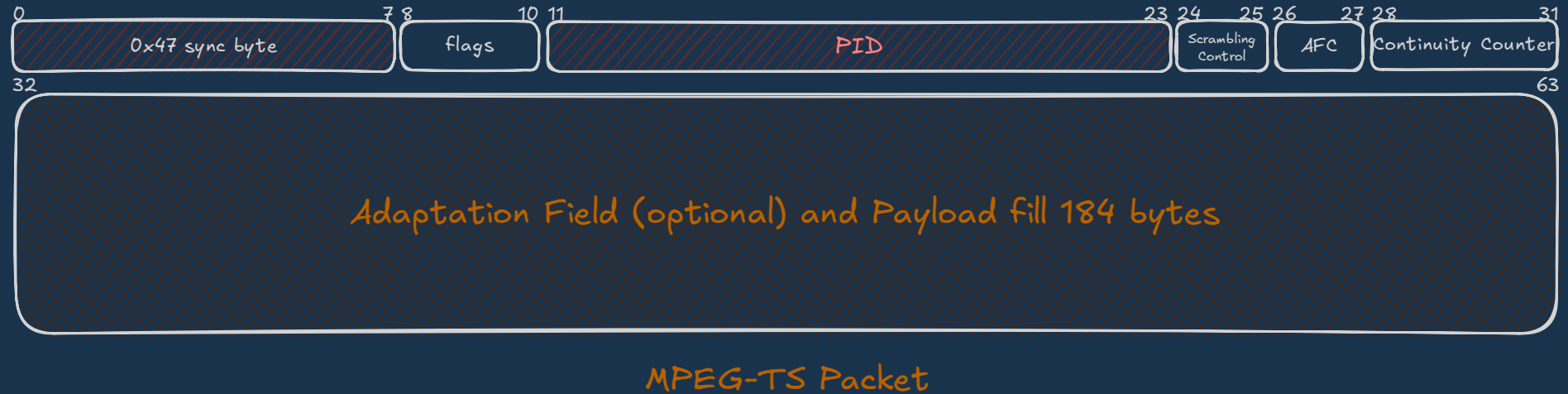
10 October 2026

Jan Schmidt <jan@centricular.com>

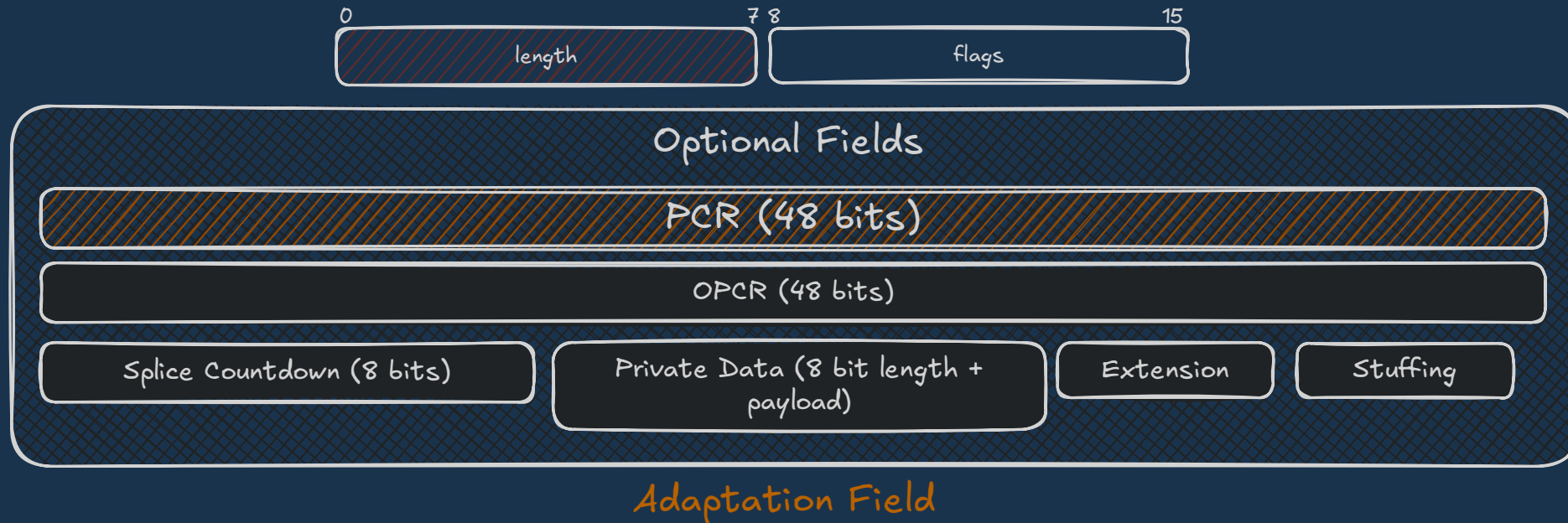
MPEG-TS?

- MPEG *Transport Stream*
- Specified in ISO 13818-1
- But also: DVB, ATSC, Blu-Ray, and more "mappings"
- Designed for streaming over potentially lossy channels
- Compare with the other half of 13818-1: MPEG-**PS**
 - DVD, Video-CD, LaserDisc
- Oh, and used in HLS - but...

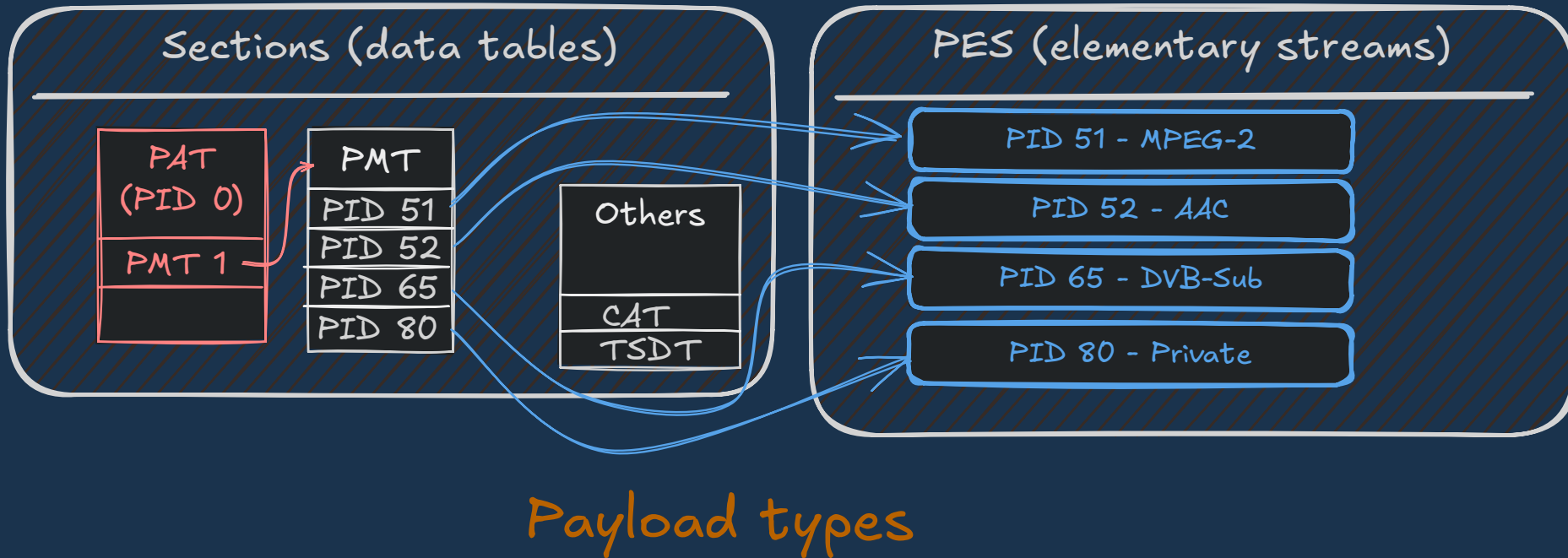
High Level Structure



Adaptation Field



Payloads



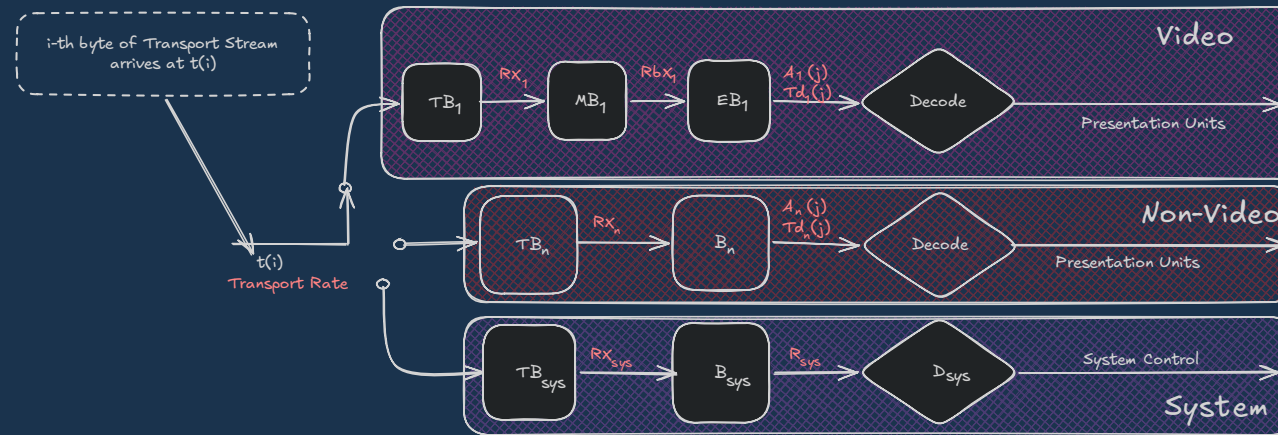
Seems simple enough, right?

- Wrap data in PES, stick into 188-byte packets, emit PAT/PMT, manage counters and CRCs.
- If all correct → valid transport stream. Right? Of course not or there'd be no talk.
- This is only the *syntax* layer. There's an entire *temporal* layer on top.
- The spec defines when data must arrive at the receiver, not just what packets look like.
- Other mux formats don't have that.

Scheduling

- "Compliant" MPEG-TS production
- ISO 13818-1 defines the Transport Stream System Target Decoder (T-STD).
- It's a hypothetical decoder model — compliant receivers must behave as if they handle buffers a particular way

T-STD



- **Transport Rate arrival** - Based on PCR
- **TB_n** — Transport Buffer: 512 bytes, drains at RX_n
- **MB_n** — Multiplex Buffer: BS_{mux} bytes (profile/level) drains at R_{bx} (video only)
- **EB_n** or **B_n** — Elementary Stream Buffer: BS_{dec} bytes (codec), AU removed at DTS

Overflow or underflow at any buffer = unplayable stream.

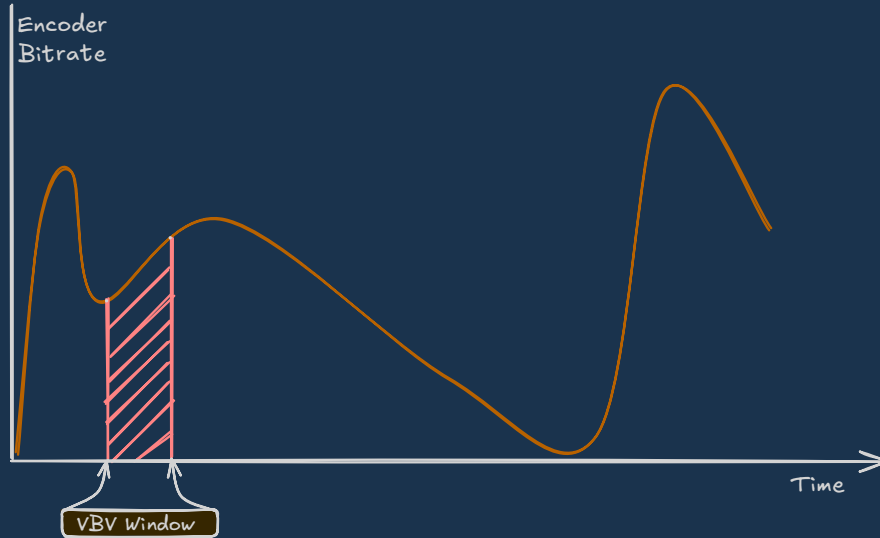
Let's talk about mpegtsmux

- GStreamer has produced MPEG-TS since ~2006. It works... But the output isn't always compliant, and receivers don't always handle it well. Why?
- Written as a PoC for an MPEG mapping for BBC's Dirac codec
- Not as a broadcast compliant muxer
 - But has improved over time

mpegtsmux scheduling

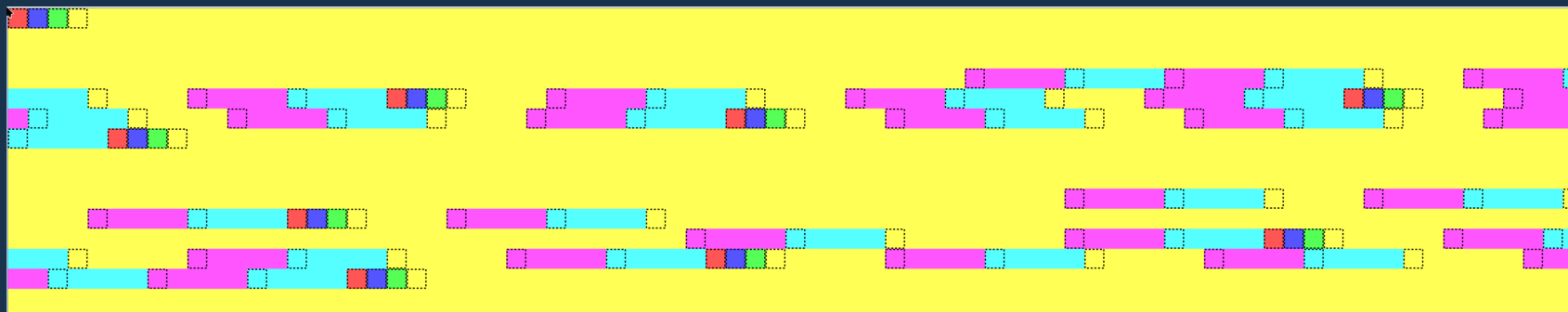
- Encoders with VBV control smooth output upstream
- Fixed PCR offset (~125 ms ahead of DTS) provides scheduling slack
- Outputting the "Earliest DTS first" buffer is good enough when bursts are bounded
- The VBV is key: size the encoder's VBV buffer so its output stays within the muxer's PCR offset window and nothing overflows.

Wait - VBV?

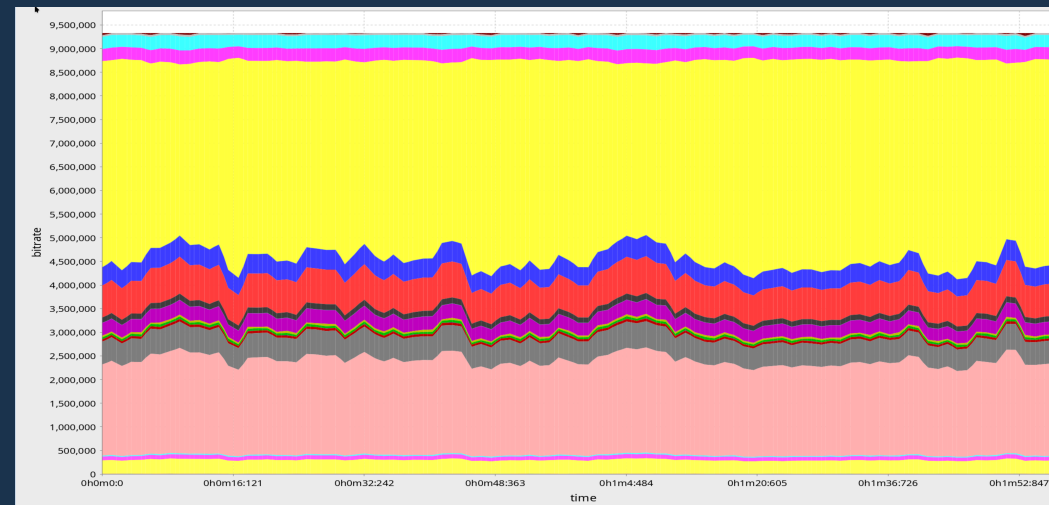
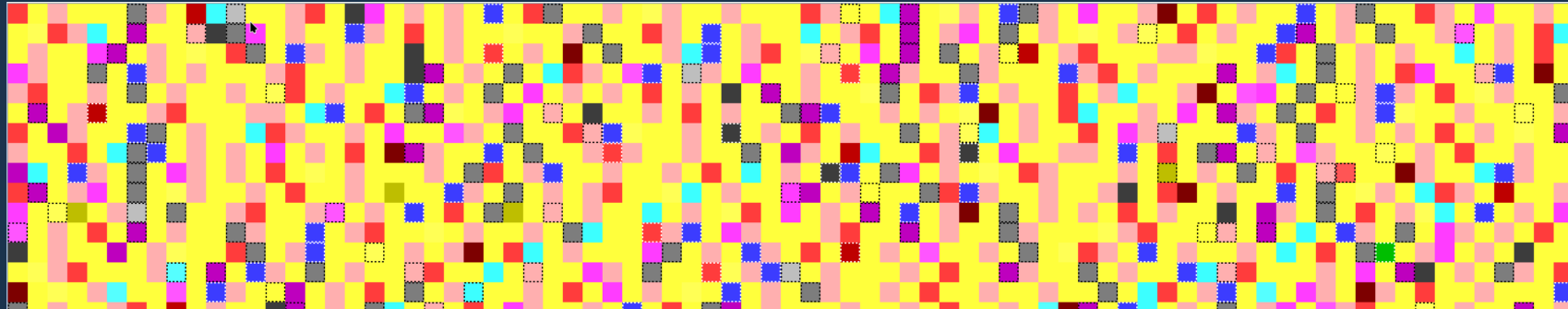


- Configurable, but fixed-size bit-buffer in the video encoder
- Encoder can output bitrate bursts within the window
 - I frames need more bits than P or B frames
- Large VBV = more VBR. Small VBV = more CBR but creates visible quality pulsing

mpegtsmux produced output



Better: Interleaving PES



PCR <-> DTS offset

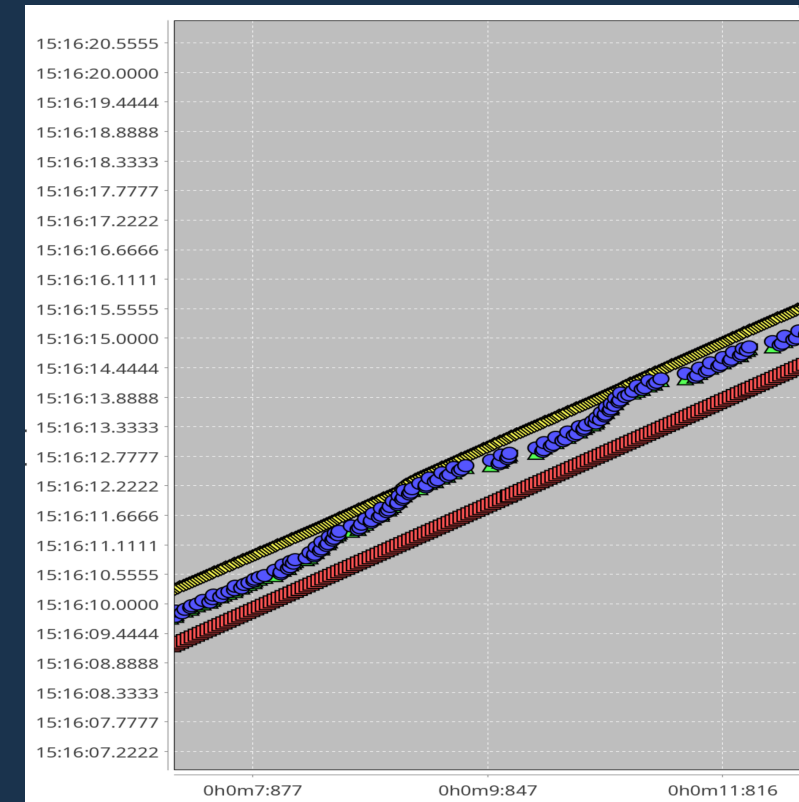
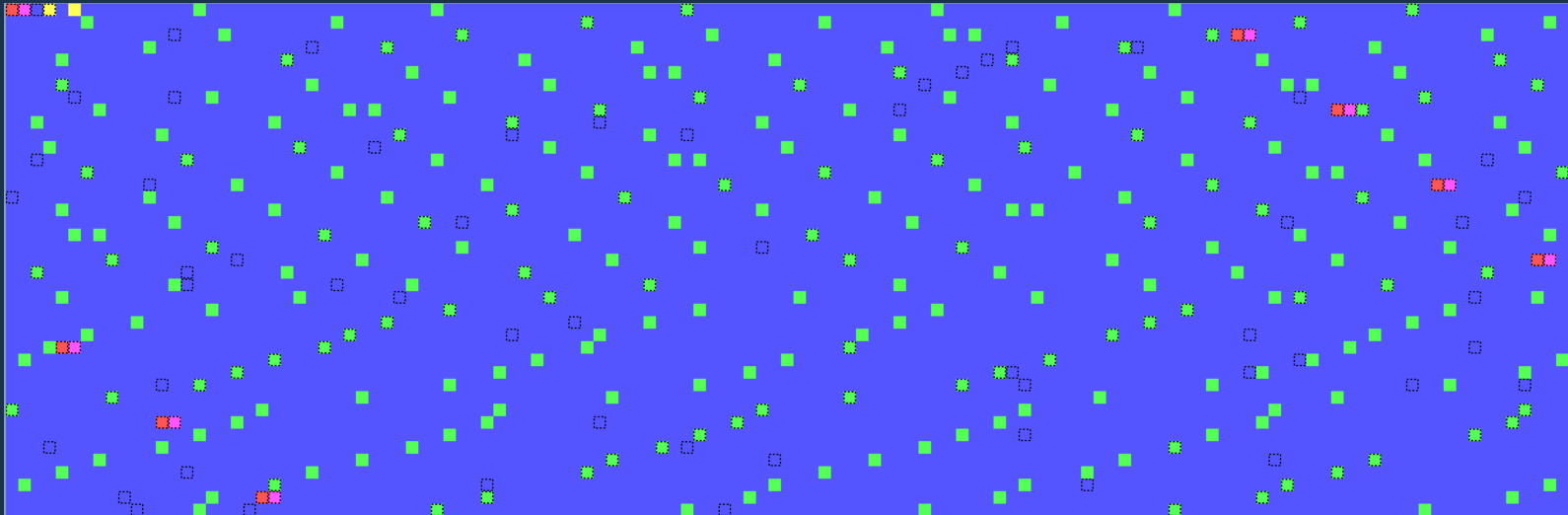
- PCR offset sets the end-to-end delay through the system
 - Low latency (PCR close to DTS) → less headroom → underflow risk
 - High latency (PCR ahead of DTS) → more freedom → current ~125 ms
 - Too High latency → T-STD buffers overflow
- End-to-end delay means *latency* for live

Why is this still broken?

- Only broadcast systems care
- HLS and file playback don't: They only need sensible PTS
- It's a lot of work to get it right and **keep** it right

HLS example

One sample off the Internet



What will it take to fix it?

- Interleaving PES payloads
- Scheduling them into the mux correctly
- Passing VBV/HRD info from the encoder where possible
 - Parsing when not possible
- Modelling all the ES allowed bitrates and T-STD buffers
- Validation

Validating Streams

- Wrote most of a validator and presented that in 2019
- It's barely moved since then
 - I'd like to finish it, or better: RIIR

```
PID 0x0021 Overflow by 0.882282 sec at PCR 1483980567797 - Non-leaky B buffer at 3624 bytes of 3584. Next output @ PCR 1484004389400
PID 0x0021 Overflow by 0.881920 sec at PCR 1483980577569 - Non-leaky B buffer at 3714 bytes of 3584. Next output @ PCR 1484004389400
PID 0x0021 Overflow by 0.874797 sec at PCR 1483980769885 - Non-leaky B buffer at 3808 bytes of 3584. Next output @ PCR 1484004389400
PID 0x0021 Overflow by 0.874435 sec at PCR 1483980779657 - Non-leaky B buffer at 3898 bytes of 3584. Next output @ PCR 1484004389400
PID 0x0021 Overflow by 0.859433 sec at PCR 1483981184696 - Non-leaky B buffer at 3992 bytes of 3584. Next output @ PCR 1484004389400
PID 0x0021 Overflow by 0.859072 sec at PCR 1483981194468 - Non-leaky B buffer at 4082 bytes of 3584. Next output @ PCR 1484004389400
PID 0x0021 Overflow by 0.850799 sec at PCR 1483981417828 - Non-leaky B buffer at 4102 bytes of 3584. Next output @ PCR 1484004389400
PID 0x0021 Overflow by 0.844070 sec at PCR 1483981599507 - Non-leaky B buffer at 4196 bytes of 3584. Next output @ PCR 1484004389400
```

What else?

- Improve systems conformance
 - DVB and ATSC mandatory tables are mostly stubs
 - Need to add API to control those properly
- "Proper" API in general would be good

Blu-ray / AVCHD

- 192 byte packets
- Adds a 4-byte "Arrival Timestamp" to every packet

Questions? Comments?



Come talk to us!

