New RTP payloaders & depayloaders in Rust

GStreamer Conference 2024

7 October 2024

Sebastian Dröge <sebastian@centricular.com>

Centricular

gstreamer

What are RTP (de)payloaders?

- Elements that convert elementary or container streams from/to RTP packets
    - RTP: Used by WebRTC, RTSP, SIP and others
- Packetization, signalling of frame properties
- Handling of missing packets
- Implement the corresponding format-specific RTP mappings

Why rewrite the RTP (de)payloaders?

- Security and correctness
  - Very ad-hoc code for complicated bitstream parsing / constructing
  - Hard to follow
- Strange base class API
- Leaner design without historical baggage
- Clean slate for re-thinking the design

What was done?

- Sponsored by the Sovereign Tech Fund (STF)
  - Together with new RTP/RTCP manager and RTSP source
- New RTP packet parsing / construction library
- New base classes
- Mostly drop-in replacement
  - In some cases more features

What was done?

- Better adherence to the specs in various cases
- Consistent usage of reader/writer APIs for bitstream handling
- Unit tests
- All written in Rust

Status?

- **Merged:** AC3, AV1*, JPEG, KLV, MP2T, MP4A, MP4G, Opus, PCMA/U, VP8, VP9
- **Pending cleanup:** AMR, H264, H265, MP4V, MPA (+ robust), MPV, raw video, Vorbis

Status?

- Works in webrtcbin, gst-rtsp-server, rtspsrc
- What is left to be done
    - Cleanup
    - Compressed audio payloader base class
    - Improved discontinuity handling

Performance?

- Approximately as fast or faster than old code
  - E.g. JPEG / VP9 pay/depay of 100k 1920x1080 I420 frames
  - **Old:** 7.9s / 2.5s
  - **New:** 4.5s / 2.4s
- Not optimized at all yet!

Thanks!

https://gitlab.freedesktop.org/gstreamer/gst-plugins-rs/-/tree/main/net/rtp/

Give the new elements a try!

Centricular

gstreamer