GStreamer RTP Sessions in Rust

GStreamer Conference 2024

Matthew Waters

7 October 2024

Who Am I?

GStreamer developer and maintainer for over a decade

WebRTC, Vulkan, OpenGL

gstreamer

Centricular

RTP: Real-time Transport Protocol

An older relatively simple standard for encapsulating real time data, usually audio/video

RFC 3550: https://datatracker.ietf.org/doc/html/rfc3550

Used in

WebRTC - RTSP - SIP - XMPP - IPTV - RIST - And others

RTP Packet format

```
|    | 0   | 1   | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 0  | Version | P | X | CSRC Count    | M | Payload Type                    |
| 2  | Sequence Number                                                        |
| 4  | Timestamp                                                              |
| 6  | Timestamp (continued)                                                  |
| 8  | Sequence Identifier (SSRC)                                             |
| 10 | Sequence Identifier (SSRC) (continued)                                 |
|    | Contribution Source (CSRC) (optional)                                  |
|    | Contribution Source (CSRC) (continued)                                 |
|    | Profile specific Extension header ID (optional)                        |
|    | Extension length                                                       |
|    | Extension data                                                         |
|    | Extension data (continued)                                             |
|    | Payload data                                                           |
|    | Payload data (continued)                                               |
```

P = Padding, X = Extension, M = Marker

RTP Payload formats

Many

- Most Audio and Video Codecs have a defined RTP payload format
- Non audio/video formats are also possible, e.g. KLV, TTML, MIDI, ANC, etc
- Registry: https://www.iana.org/assignments/rtp-parameters/rtp-parameters.xhtml

RTCP: Real Time Control Protocol

Sister protocol to RTP allowing control and feedback messages

- Statistics from both the receiver and sender
  - Who is currently participating in this session?
- Limited to 5% of the session bandwidth
- Integral to Session management

RTCP Packet format

```
|     | 0     | 1     | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 0   | Version | P | Count             | Payload Type                    |
| 2   | length                                                            |
```

P = Padding

Prior Art: `rtpbin`

- A `GstBin` that handles most of the intricacies of a RTP session
  - By default includes `rtpjitterbuffer`, SSRC and payload demultiplexing
  - Extension points for Retransmission, FEC, etc

gstreamer

Centricular

`rtpbin` - Concerns

`rtpbin` - Concerns

- A large API surface
  - 30 properties, 26 signals, 7 action signals
  - Most owing to the vast range of use cases `rtpbin` attempts to support

`rtpbin` - Concerns

- A large API surface
  - 30 properties, 26 signals, 7 action signals
  - Most owing to the vast range of use cases `rtpbin` attempts to support
- Number of RTCP threads - one per session

gstreamer

Centricular

`rtpbin` - Concerns

- A large API surface
  - 30 properties, 26 signals, 7 action signals
  - Most owing to the vast range of use cases `rtpbin` attempts to support
- Number of RTCP threads - one per session
- Inter-stream synchronisation is a multi-element endeavour

`rtpbin` - Concerns

- A large API surface
  - 30 properties, 26 signals, 7 action signals
  - Most owing to the vast range of use cases `rtpbin` attempts to support
- Number of RTCP threads - one per session
- Inter-stream synchronisation is a multi-element endeavour
- SFU/MCU use cases generally require ‘wormhole’ elements to avoid graph loops

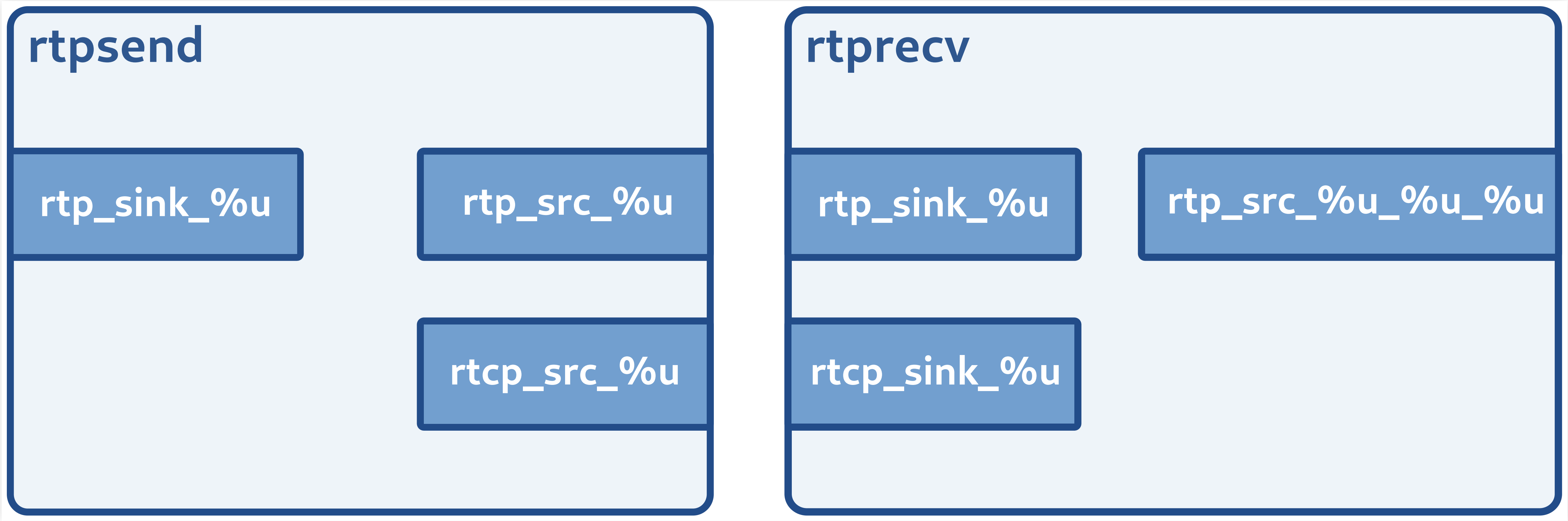
What if we did better?

What if we did better?

And in Rust?

Sponsored by the Sovereign Tech Fund

rtpsend + rtprecv



`rtpsend + rtprecv`

Features

- RTP/AVP profile
- RTP/AVPF profile
- PLI/FIR handling
- Reduced size RTCP
- rtcp-mux
- Jitterbuffer
- CNAME synchronisation
- Statistics reporting

`rtpsend` + `rtprecv`

- Two elements now
  - Removes the wormhole requirement for SFU/MCU use cases
- `rtpsend` and `rtprecv` are connected by using the same process unique `rtp-id` value

`rtpsend` + `rtprecv`

Reduced number of RTCP threads

- Uses a tokio runtime internally for RTCP timeouts for all `rtpsend` elements
- Actual RTCP pad push occurs from a tokio blocking thread pool

`rtpsend` + `rtprecv`

Internally sans-IO

- The session handling only operates on the inputs it receives
  - Even including retrieving current time
- Allows for intricate and precise unit testing
- Also reusability
- https://sans-io.readthedocs.io/

gstreamer

Centricular

New crate: `rtp-types`

- Parses and writes RTP packets
- RTP Header parsing in <2-3ns
- Generic and extensible writing
    - Could be used by Rust RTP payloaders to directly write from an incoming `GstBuffer` into an output `GstBuffer`
- Can also edit fixed RTP header fields in place

gstreamer

Centricular

New crate: `rtcp-types`

- Handles parsing and writing of RTCP packets
  - Sender Report
  - Receiver Report
  - Bye
  - SDES
  - App
- Other RTCP packet types are possible

gstreamer Centricular

Uses

- Optionally used by `rtspsrc2` with `USE_RTP2=1`
- Can already be used by simple RTP scenarios, i.e. most `gst-launch-1.0` pipelines

Demo

Demo

```
rtpsend name=rtp \
src ! vp8enc ! rtpvp8pay ! rtp.rtp_sink_0 \
rtp.rtp_src_0 ! udpsink host=127.0.0.1
```

Demo

```
rtpsend name=rtp \
src ! vp8enc ! rtpvp8pay ! rtp.rtp_sink_0 \
rtp.rtp_src_0 ! udpsink host=127.0.0.1
```

```
rtprecv name=rtp \
udpsrc caps=... ! rtp.rtp_sink_0 \
rtp. ! rtpvp8depay ! vp8dec ! autovideosink
```

Future

- Retransmissions - main requirement for WebRTC usage
- Forward Error Correction - secondary requirement for WebRTC usage
- RTCP XR
- TWCC: Transport Wide Congestion Control
- Other RTP scenarios

Thanks

Try it out!

- @ystreet00:matrix.org on Matrix
- https://discourse.gstreamer.org/u/ystreet00
- https://gitlab.freedesktop.org/ystreet
- ystreet00@floss.social on mastodon