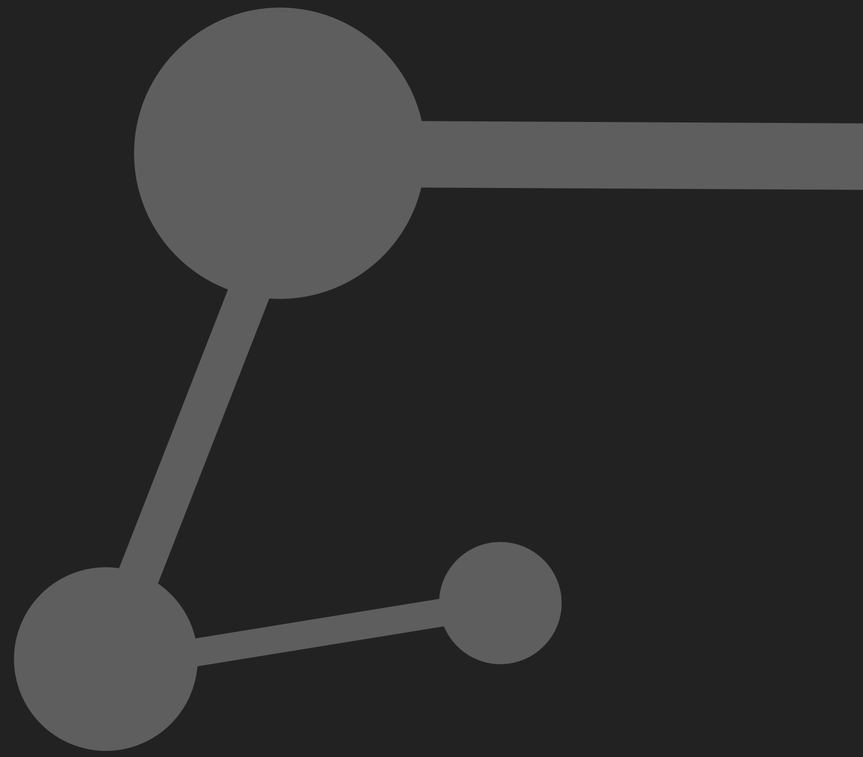
JARON VIËTOR

HOW MISTSERVER HANDLES SRT CONNECTIONS IN INDEPENDENT CHILD PROCESSES

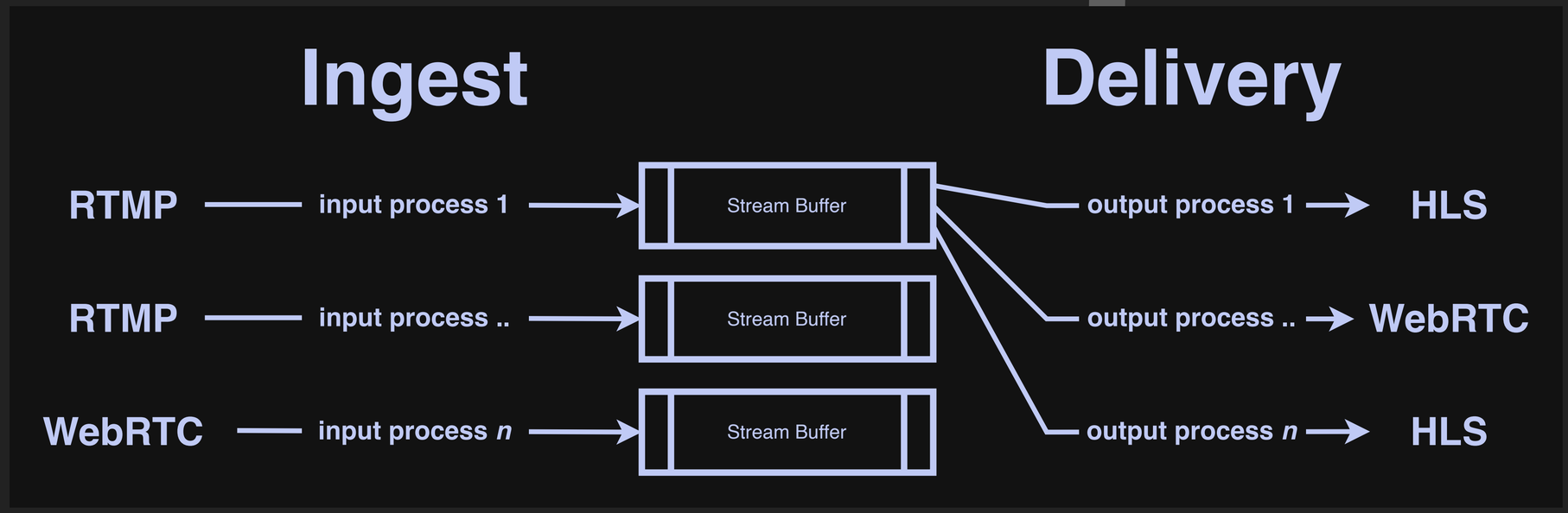
WHAT IS MISTSERVER?

- Media server software package
- In development since ~2009
- Free and Open Source (Unlicense)
- Focus on transmuxing and easy integration
- Developer-centric



MISTSERVER CONNECTIONS

- Separate process per connection
- ✘ Less efficient
- ✔ Isolation between clients



SRT: SECURE RELIABLE TRANSPORT

- Reference implementation: libsrt
- Haivision, circa 2012
- Open-sourced in 2017
- Sends MPEG-TS streams, preserving timing
- Fairly low latency
- Usage by the media industry increasing in recent years.

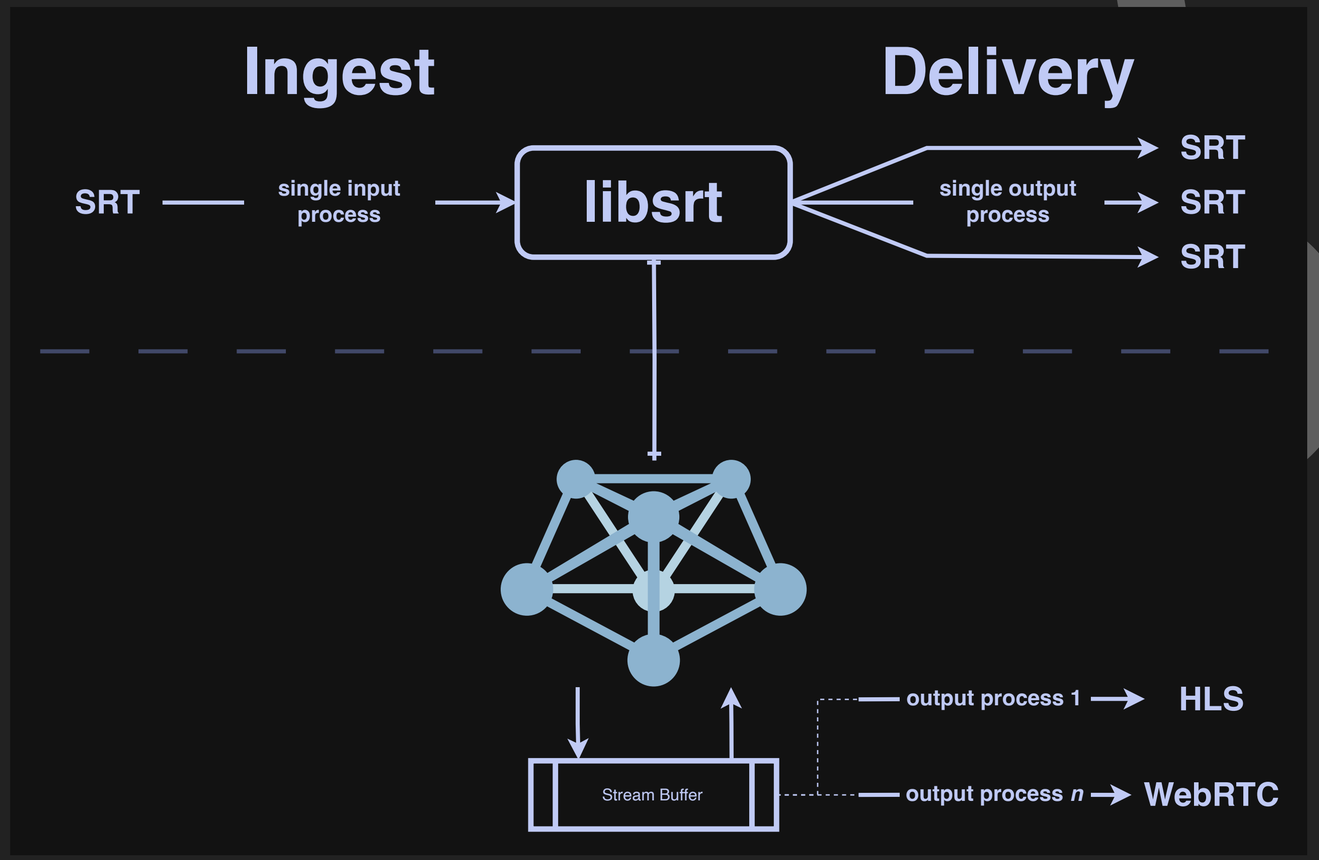
NO PROBLEM, RIGHT?

- libsrt -> reference implementation in c++
- upump-srt -> tied closely to upipe, partial
- gosrt -> written in go, partial

- MistServer -> Open Source!
- libsrt -> Open Source!

A MINOR INCONVENIENCE

- libsrt takes ownership of the underlying UDP sockets
- Manages them as an internal resource
- ☹ No separate process per connection!



POSSIBLE SOLUTIONS

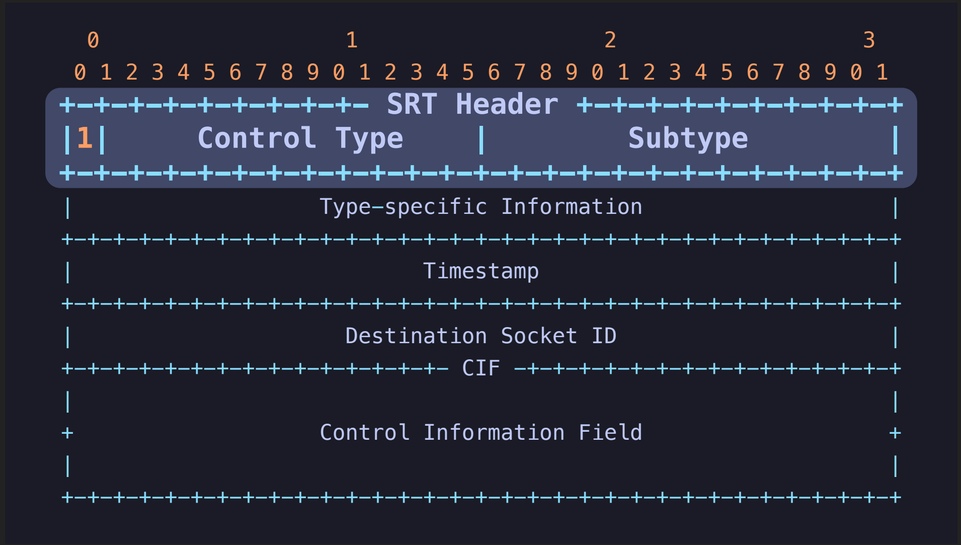
- Add support for forking binaries to libsrt
- Write our own SRT implementation

SO WE'RE STUCK. OR ARE WE?

- Unattractive options:
  - Modifying or forking libsrt
  - Custom implementation
- SRT’s future was uncertain
- Initial solution: ugly but functional
- Current situation: SRT ended up sticking around

THE SECRET THIRD OPTION

- SRT specification available now
- Handshake packets quite easy to recognize

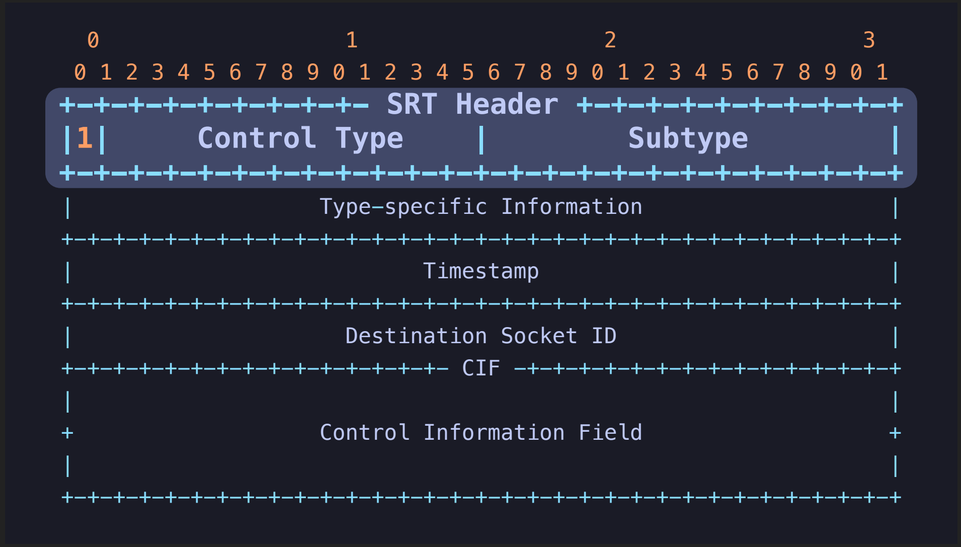


SRT Control packet structure

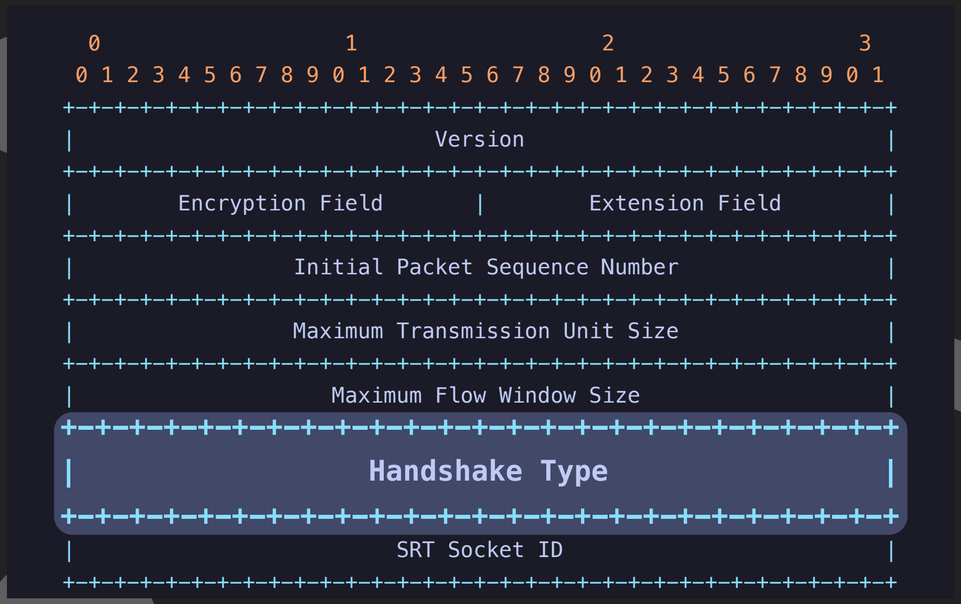
SRT Handshake packet structure

THE SECRET THIRD OPTION

- SRT specification available now
- Handshake packets quite easy to recognize



SRT Control packet structure



SRT Handshake packet structure

```
if (byte[0..3] == [0x80, 0x00, 0x00, 0x00]){
  isHandshake = true;
  if (dataSize >= 40){
    isRendezvous = byte[36..39] == [0x00, 0x00, 0x00, 0x00];
  }
}
```

SRT handshake detection pseudocode

UDP SOCKETS

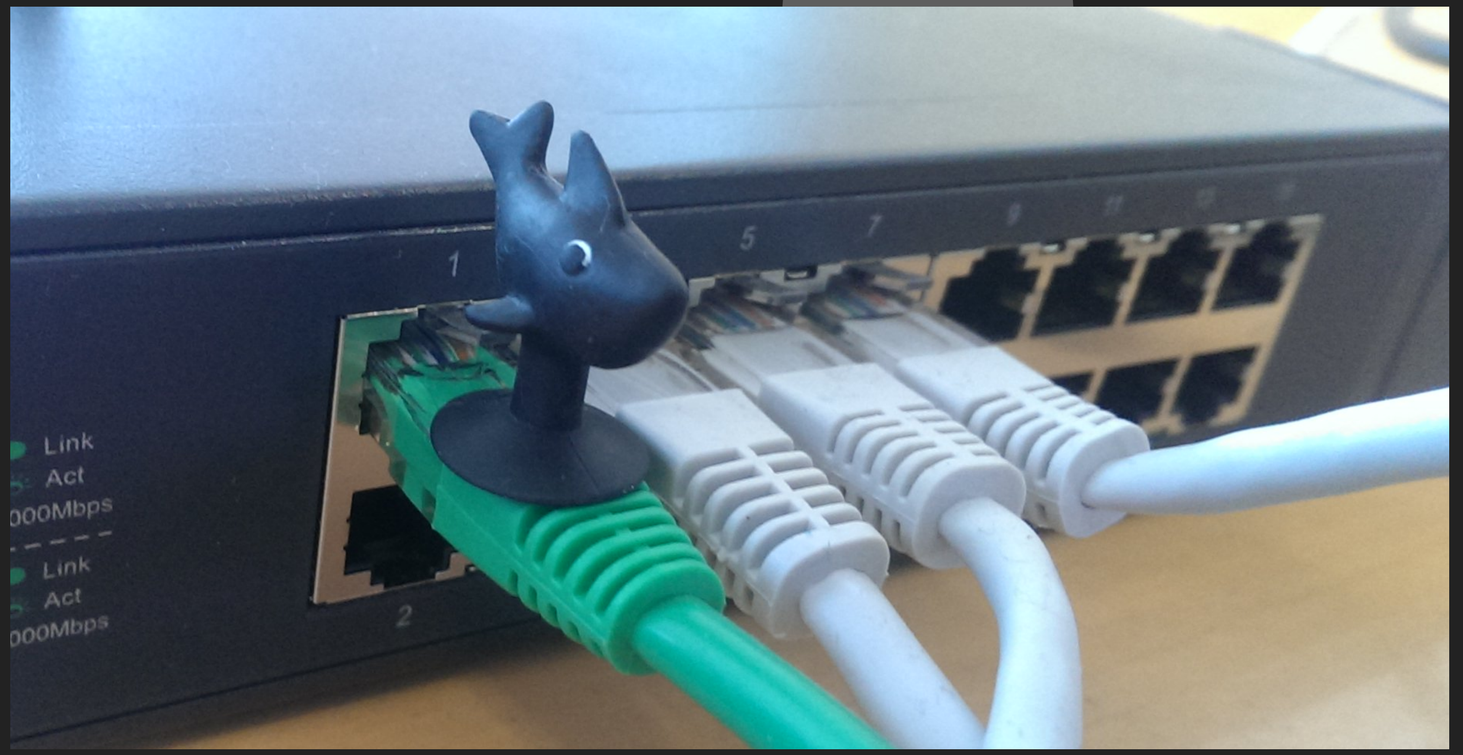
- UDP is a connection-less protocol
- UDP sockets *can* still be connected:
  - Connected UDP sockets receive all traffic from the connected peer

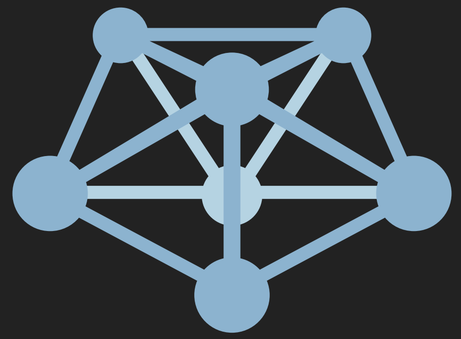
UDP SOCKETS

- UDP is a connection-less protocol
- UDP sockets *can* still be connected:
  - Connected UDP sockets receive all traffic from the connected peer

INTERCEPT THE HANDSHAKE OURSELVES

- Listen on the main port ourselves
- Spawn a new process, init libsrt, use a connected UDP socket to listen
- libsrt docs forbids this - but it works!

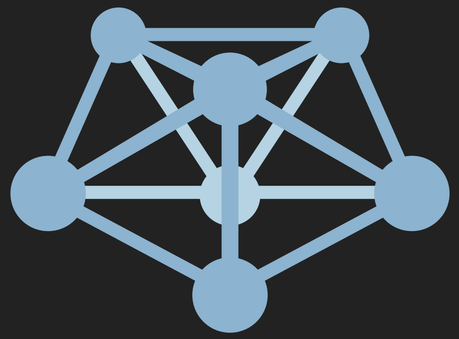




- ✔ STANDARD, UNMODIFIED LIBSRT
- ✔ MULTI-PROCESS ISOLATION!

BEST OF BOTH WORLDS!

QUESTIONS?



Email: jaron@mistserver.org

www.mistserver.org