rs-sixty-seven

an open source framework for professional audio networks talking AES67

Marc Schettke

@marcschettke

www.schettke.com

Rust User Group Cologne / Open Source Audio Meeting Cologne, 15th November 2016

Why audio networks

- reduced cabling
- one physical infrastructure — many scenarios
- high channel count
  (even at high sampling freq.)

Requirements

- used by: broadcasters, studios, live venues, installations (congress halls, theme parks)
- uncompressed audio (i.e. 96kHz, 24 bit)
- low latency (few milliseconds, lower is better)

Every Audio Network

- synchronization (of end points)
- media clock generation and recovery
- encoding of audio data
- transport (network layer, QoS)

AES67

- Audio Engineering Society
- collects methods from existing network solutions
- not reinventing the wheel
- specifies the „minimum viable audionetwork“ (MVA)
- i.e. discovery as informative reference but not required

AES67 Synchronization

- PTP IEEE1588 v2
- master is elected and sends sync messages
- propagation delay is measured
- time offset to master is calculated
- syntonization and synchronization at end nodes

AES67 Transport

- IPv4 (because audio people usually are a little behind time)
- unicast or multicast
- DiffServ for QoS (1. clock, 2. media streams, 3. control data)
- RTP/RTCP over UDP as transport layer

AES67 Media Clock

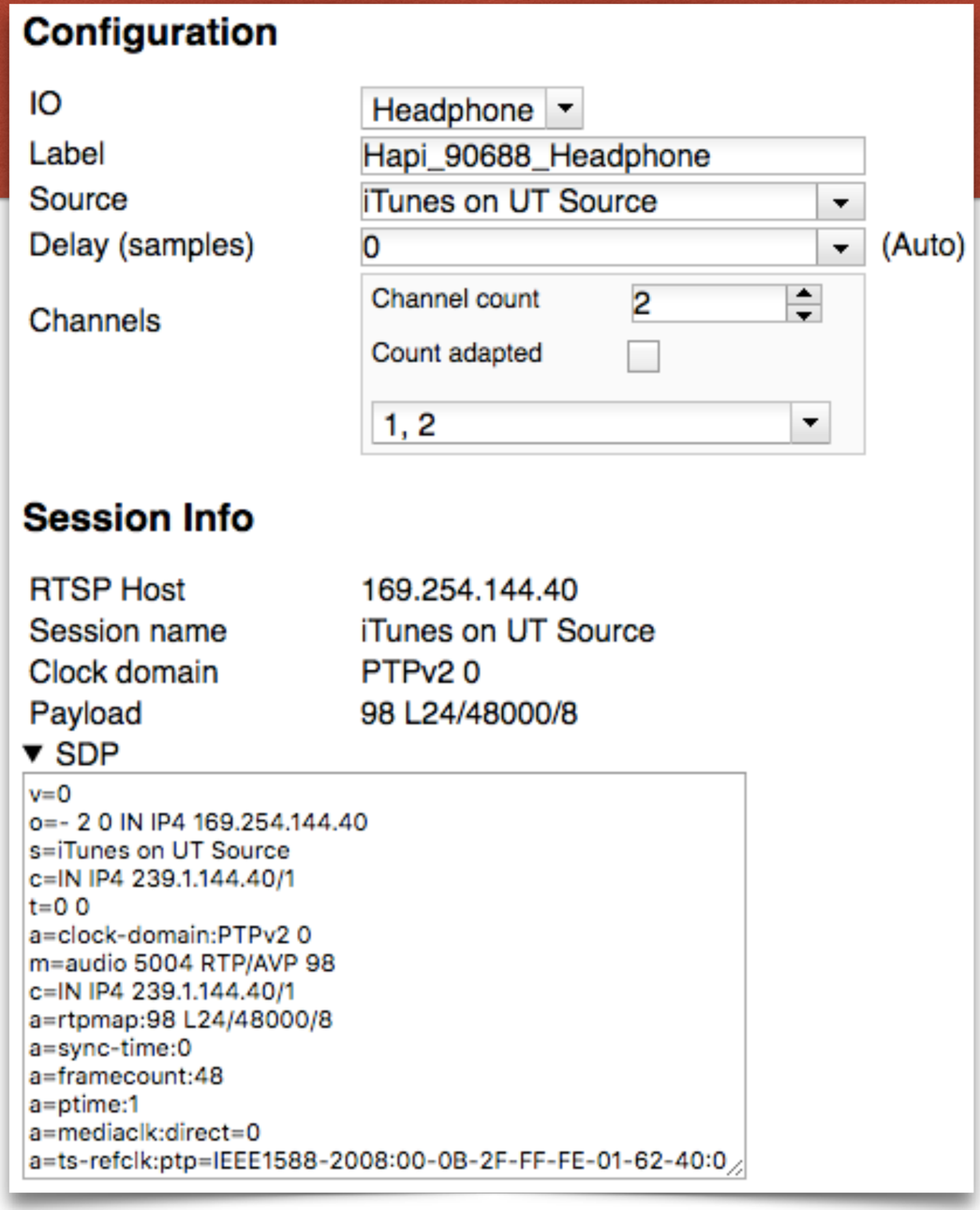
- media clock is derived from PTP time:
  - advances by one after sampling period
  - i.e. every 20.8 µs @ 48 kHz
  - referenced to UTC epoch
- rtp clock is media clock + offset
- 32 bit representation — overflows once a day

AES67 Encoding

- multiple samples are gathered according to packet time
- the payload is prepended with an RTP header
  - timestamp (media clock at first sample)

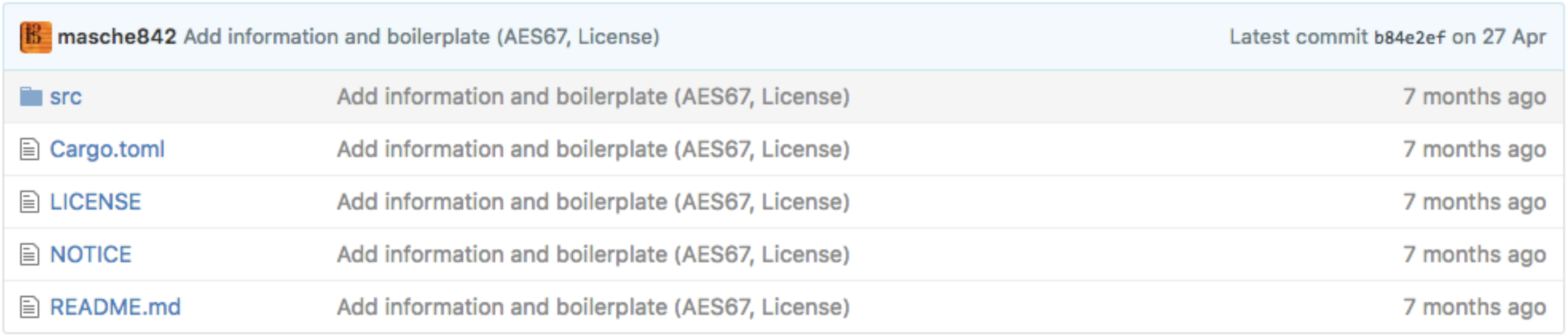
AES67 Metadata

- discovery and connection management (optional)
- link offset
- clock source
- uses SDP



rs-sixty-seven

- so far only a stub
- github.com/masche842/rs-sixty-seven



masche842 Add information and boilerplate (AES67, License) — Latest commit b84e2ef on 27 Apr

src	Add information and boilerplate (AES67, License)	7 months ago
Cargo.toml	Add information and boilerplate (AES67, License)	7 months ago
LICENSE	Add information and boilerplate (AES67, License)	7 months ago
NOTICE	Add information and boilerplate (AES67, License)	7 months ago
README.md	Add information and boilerplate (AES67, License)	7 months ago

Framework

- daemons that can be glued together easily
- example applications that work with professional audio equipment
- virtual soundcard for Linux?
- like https://github.com/AVnu/Open-AVB

Development

- use what's already available
  - ptpd
  - gstreamer
  - see https://groups.google.com/forum/#!topic/crc-mmbtools/uQ9s70yNr58
- test continuously against these libraries
- use a library first and replace with rust implementation

gstreamer

- http://gstreamer-devel.966125.n4.nabble.com/Ravenna-AES67-protocol-before-starting-the-journey-td4674772.html
- „[...] still, open source code lacks, so it is good that gstreamer supplies it.“
- gstreamer 1.6 (9/2015): support for PTP added
- https://gstreamer.freedesktop.org/projects/rust.html

openbroadcaster

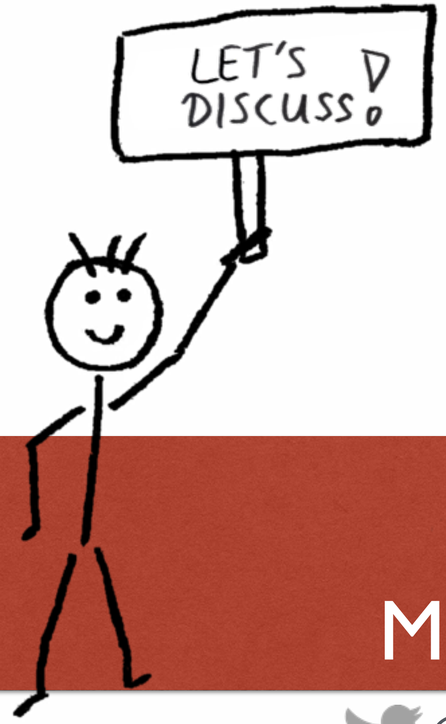
- https://obsproject.com/
- https://wiki.openbroadcaster.pro/Ravenna

rs-sixty-seven

- 100% Rust
- better out-of-the-box experience
- suitable for embedded devices
- foundation for the „real audio network stuff“
  - idiomatic signals from source to sink (i.e. microphone to DAW without interconnections)
- … but the journey has just started

rs-sixty-seven

github.com/masche842/rs-sixty-seven



Marc Schettke

@marcschettke

www.schettke.com

… and feel free to stop by and contribute :-)