Using Zephyr RTOS for Professional Audio-over-IP

Zephyr Project Meetup: Winterthur, Switzerland

February 12, 2026

Peter Büchler

zhaw School of Engineering
InES Institute of Embedded Systems

NOSER ENGINEERING

Introduction

Peter Büchler

- Hardware and Software Engineer at Noser Engineering AG
- MSc Information and Communication Technologies (MSE), ZHAW SoE
- peter.buechler@noser.com

NOSER ENGINEERING

Motivation

Goal: Learn Zephyr through a hands-on project

Drivers for choosing Zephyr:

- Many robust drivers and modules with a high level of abstraction
- Supports complex embedded applications (filesystem, connectivity, cloud, security)
- Regulatory driver: Cyber Resilience Act (CRA)

Hands-on learning using a real project:

- Use case: “replacement of a discontinued Audio-over-IP module”

What is Audio-over-IP

Application:

- Professional use: broadcast, studios, live events
- Standards (e.g.): AES67, Dante, AVB, SMPTE ST 2110-30
- Low latency: 1–10 ms
- Synchronized transmission with consistent delay
- High-resolution, lossless, uncompressed audio
- Off-the-shelf solutions available as hardware modules

Not intended for:

- YouTube, Spotify, or internet radio streaming

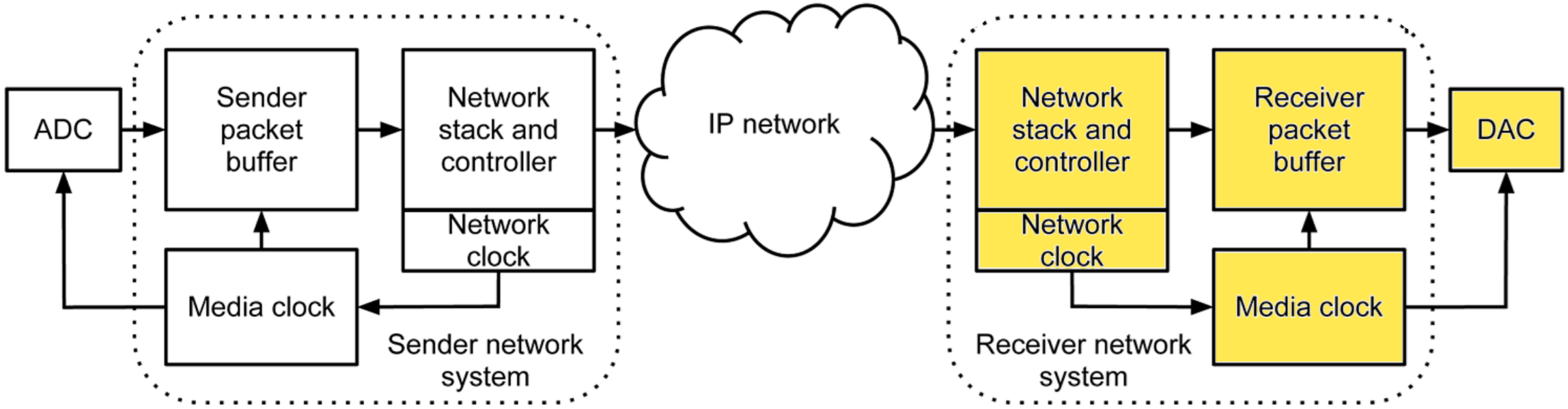
NOSER ENGINEERING

AES67

NOSER ENGINEERING

System Overview

- Time synchronization → PTP (Precision Time Protocol)
- Transport Protocol → RTP (Real-time Transport Protocol)
- Description Protocol → SDP (Session Description Protocol)
- Discovery Protocol → SAP (Session Announcement Protocol) or mDNS/Bonjour



NOSER ENGINEERING

PTP (Precision Time Protocol)

Challenges

- Networks are inherently asynchronous
- Variable delay and out-of-order packet delivery

Zephyr Support:

- Experimental PTP integration

	NTP	PTP
Purpose	Synchronize system time	High-precision time synchronization
Applications	IT, Server, PCs	Measurement systems, communication technology
accuracy	ms range	µs to ns range
Transport	UDP/IP	Layer 2

NOSER ENGINEERING

RTP (Real-time Transport Protocol)

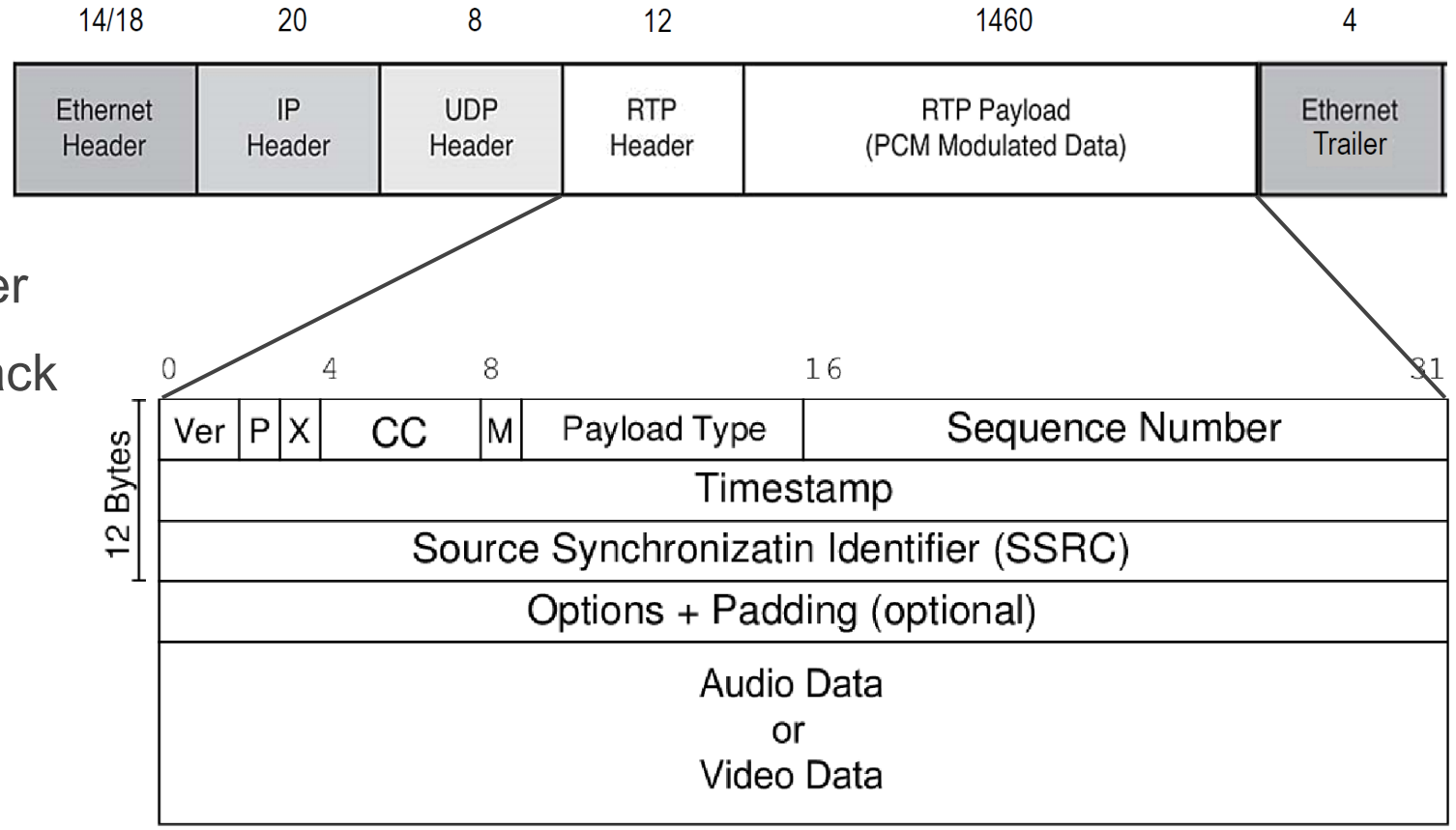
Purpose: Transports audio data in real time over IP networks

Properties:

- Streaming over UDP
- Supports multicast & unicast
- Sequence numbers for packet order
- Timestamp for synchronous playback

Zephyr Support:

- Network native API with multicast and zero-copy



NOSER ENGINEERING

SDP (Session Description Protocol)

Definition: A text file that describes all essential parameters of an audio stream.

Purpose:

- Receivers instantly know how to correctly receive the stream.
- Enables interoperability between different manufacturers.

Content:

- Stream name & URI
- Sample rate & sample format
- Number of channels
- IP address and port
- Clock/timing information

```
v=0
o=1 0 IN IP4 192.168.1.100
s=RAVENNA demo stream
t=0 0
a= ts refclk:ptp =IEEE1588 2008:00 60 6e ff fe 7c 23 0f:0
a= mediaclk:direct =0
m=audio 5004 RTP/AVP 98
a=rtpmap:98 L24/48000/2
c=IN IP4 239.3.14.142
a= recvonly
a=ptime:1
```


NOSER ENGINEERING

Discovery und Stream-Announcement

Goal: Automatically find devices and audio streams in the network.

Mechanisms:

- SDP (Session Description Protocol) → describes who is sending what
- SAP (Session Announcement Protocol) → announces streams in the network
- Bonjour / mDNS (DNS-SD) → enables local device discovery (e.g., studio equipment)

Application in AES67:

- Receivers automatically discover available streams
- Senders announce their streams, including sample rate, number of channels, and format

Zephyr Support:

- Network POSIX API with multicast

Zephyr

NOSER ENGINEERING

Hardware Evaluation

Criteria for Hardware Selection

- Large touch display
- Ethernet with PTP support
- Great Zephyr support
- Strong processing performance
- Large RAM for display and audio buffers
- Audio interface, preferably with codec

Reality Check

- Display configuration missing
- Quad-SPI PSRAM not supported
- Audio interface and codec not supported

NOSER ENGINEERING

Final Proof of Concept

Implemented Features

- Receive SAP and parse SDP
- Receive AES67 RTP audio streams
- PTP sample clock synchronization
- Touch display using LVGL

Hardware & Software Basis

- MCU: STM32H735 (~7% CPU load)
- RTOS: Zephyr 4.3.99 (main-branch)

NOSER ENGINEERING

Development Experience & Conclusion

What Worked Surprisingly Well

- Low CPU load (~7%)
- Stable multicast streaming
- LVGL UI performance
- Overall system stability

Pain Points / Lessons Learned

- Use main branch for new hardware
- Expect to patch drivers
- Maintain your own Zephyr fork
- Read issues, not only docs

“Zephyr RTOS: The perfect fit for modern, scalable, and connected embedded systems.”

NOSER ENGINEERING