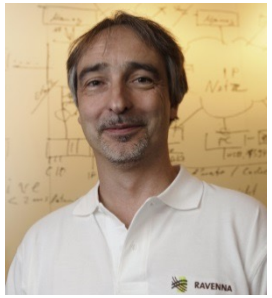
Comparison of Clocking & Synchronization in RTP, RAVENNA/AES67 & ST2110, AVB and IPMX:

"Do we really need PTP?"

MoIP Pavilion @ AES New York 2022

Andreas Hildebrand, ALC NetworX



Andreas Hildebrand, RAVENNA Technology Evangelist

- more than 25 years in the professional audio / broadcasting industry
- graduate diploma in computer science
- R&D, project & product management experience
- member of AES67 TG and ST2110 DG

ALC NetworX GmbH, Munich / Germany

- established 2008
- R&D center
- developing & promoting RAVENNA
- Partnerships with > 40 manufacturers

RAVENNA

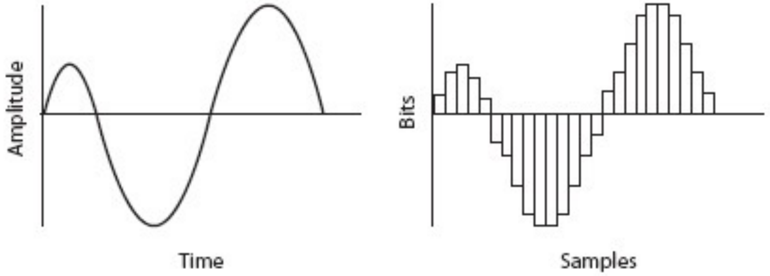
- IP media networking technology
- designed to meet requirements of professional audio / broadcasting applications
- open technology approach, license-free
- fully AES67- and SMPTE ST2110-compliant

RAVENNA
AES67 & ST 2110 built-in

Timing & Synchronization – General Requirements

- Media bit-transparency
  - → no sample rate conversion
  - → streams need to run on same media clock
- Concurrent operation of different sample rates on same network

Media clock
=
Sampling rate



- Determinable (low) end-to-end latency
- Time alignment between media streams

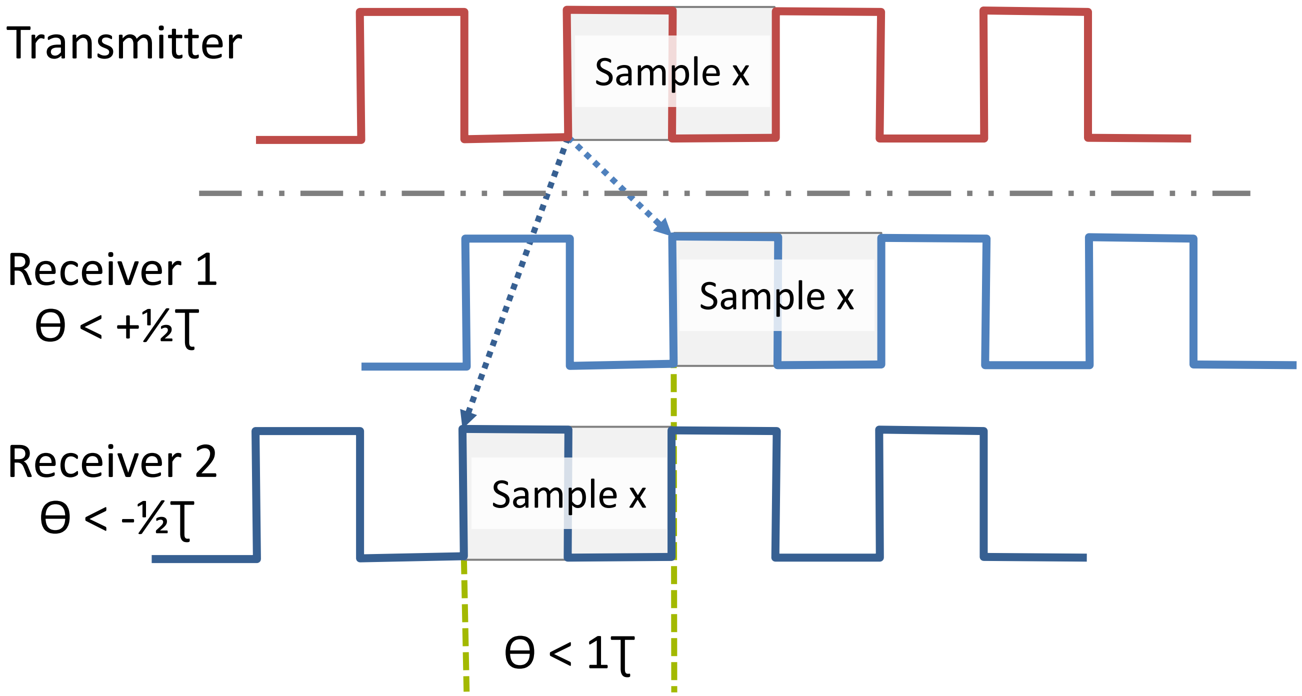
- Replacement for "house clock" distribution (word clock, black burst etc.)

⇨ Clock reassembly from stream data not applicable — > asynchronous <
⇨ Distribution of master clock beats not sufficient — > syntonous <
⇨ Common understanding of reference time required ("wall clock") — > synchronous <

RAVENNA
AES67 & ST 2110 *built-in*

Timing & Synchronization – Accuracy Requirements

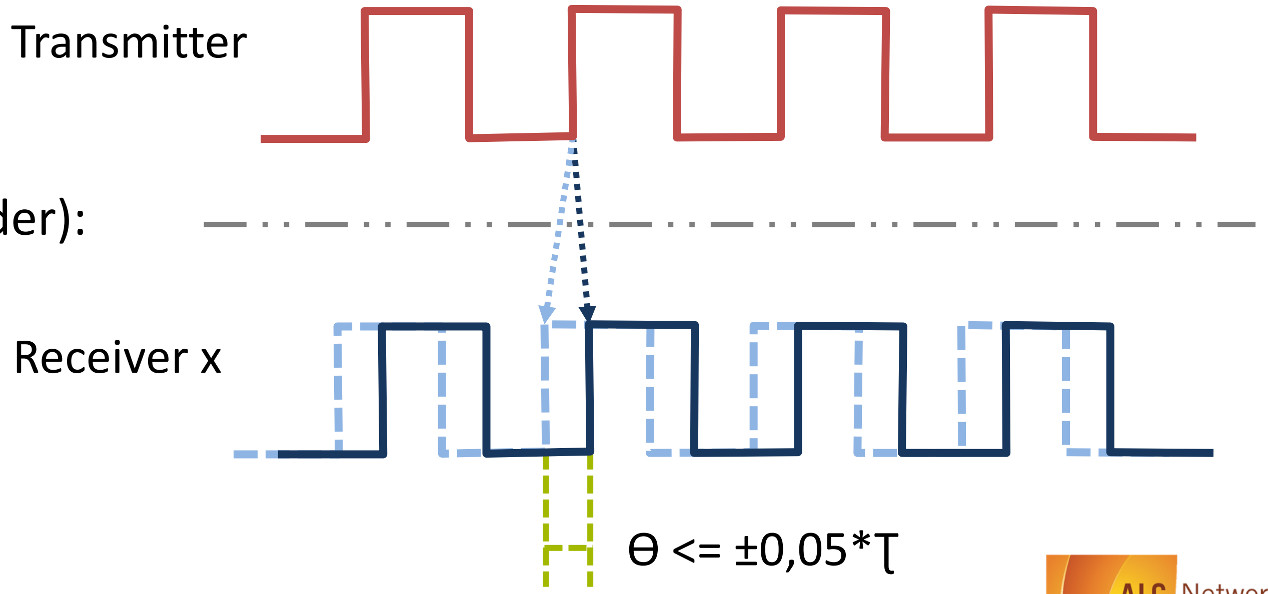
- Audio applications have highest time accuracy & precision demands:
  - ⇨ Sample accurate alignment of streams (± ½ sample)



RAVENNA
AES67 & ST 2110 built-in

Timing & Synchronization – Accuracy Requirements

- Audio applications have highest time accuracy & precision demands:
  - ⇨ Sample accurate alignment of streams (± ½ sample)
    - @ 48 kHz: ± 10 µs
    - @ 96 kHz: ± 5 µs
    - @ 192 kHz: ± 2.5 µs
  - ⇨ "Distribution" of word clock reference (AES11 calls for ± 5% max jitter / wander):



RAVENNA
AES67 & ST 2110 built-in

Timing & Synchronization – Accuracy Requirements

- Audio applications have highest time accuracy & precision demands:

  ⇨ Sample accurate alignment of streams (± ½ sample)
  - @ 48 kHz: ± 10 µs
  - @ 96 kHz: ± 5 µs
  - @ 192 kHz: ± 2.5 µs

  ⇨ "Distribution" of word clock reference
  (AES11 calls for ± 5% max jitter / wander):
  - @ 48 kHz: ± 1 µs
  - @ 96 kHz: ± 500 ns
  - @ 192 kHz: ± 250 ns

Various Clocking & Synchronization Models

I E T F

RTP (RFC 3550)

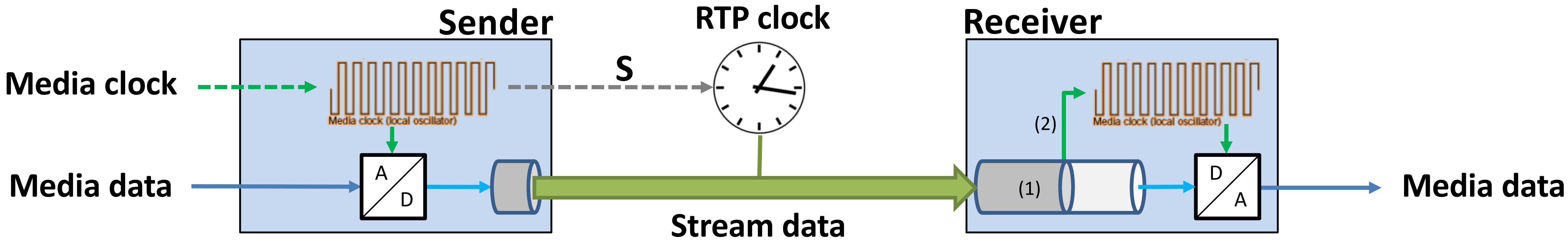
RAVENNA/AES67 & SMPTE ST 2110-10

Time Sensitive Networking (TSN)

AVB / TSN (IEEE 802.1)

IPMX

Generic RTP clocking (linear PCM audio) – asynchronous case (no "wall clock")



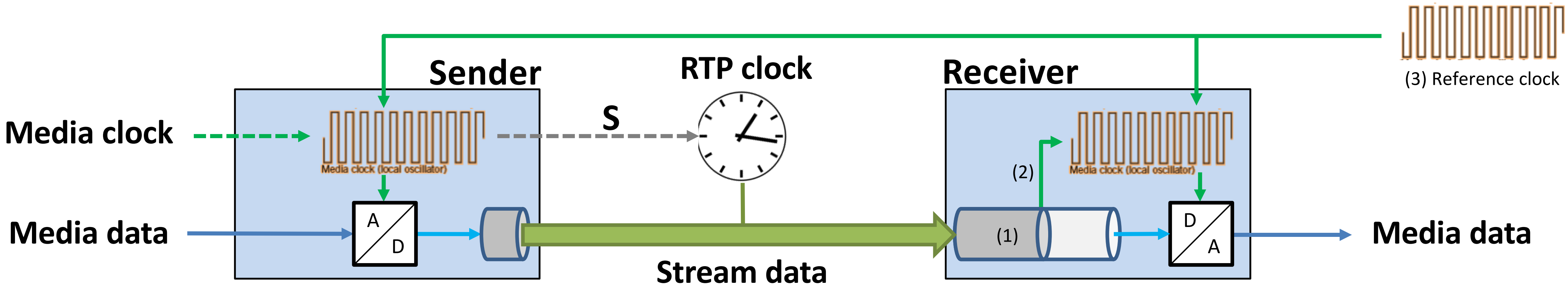
- (external or internal) media clock drives sampling
- RTP clock is derived from media (sampling) clock
  - Relation **S** (offset between media and RTP clock) is established on stream start-up
  - **S** may be random to defeat crypto attacks
  - This offset will be constant throughout the stream's lifetime
- Received samples will be stored in a (jitter/delay) buffer with configurable delay (1)
- Sender's media clock is recreated through adaptive algorithm observing receive buffer fill level (2)
- Samples are played out following recreated media clock

Generic RTP clocking (linear PCM audio) – asynchronous case (no “wall clock”)

Main benefits / drawbacks:

+ Any input signal (any media clock) can be transported and reasonably-well be resembled
  - Depending on quality of sender / receiver clock circuitry
- No concurrent processing of independent streams
  - SRC required
- No deterministic latency
  - No alignment between streams

Generic RTP clocking (linear PCM audio) – syntonous case (external "house clock")



- (external or internal) media clock drives sampling
- RTP clock is derived from media (sampling) clock
  - Relation **S** (offset between media and RTP clock) is established on stream start-up
  - **S** may be random to defeat crypto attacks
  - This offset will be constant throughout the stream's lifetime
- Received samples will be stored in a (jitter/delay) buffer with configurable delay (1)
- Sender and receiver media clocks may also be syntonized by external means (i.e., distribution of common reference clock) (3)

Generic RTP clocking (linear PCM audio) – syntonous case (external "house clock")

Main benefits / drawbacks:

- \+ All media clocks are following a common reference clock
  - o Same media clock for all streams
- \+ concurrent processing of independent streams possible
  - o No phase / time alignment
- \- No deterministic latency

RTP clocking w/ RTCP Sender Reports (linear PCM audio)*

1) With common reference clock (“wall clock”)
2) Without common reference clock (“wall clock”)

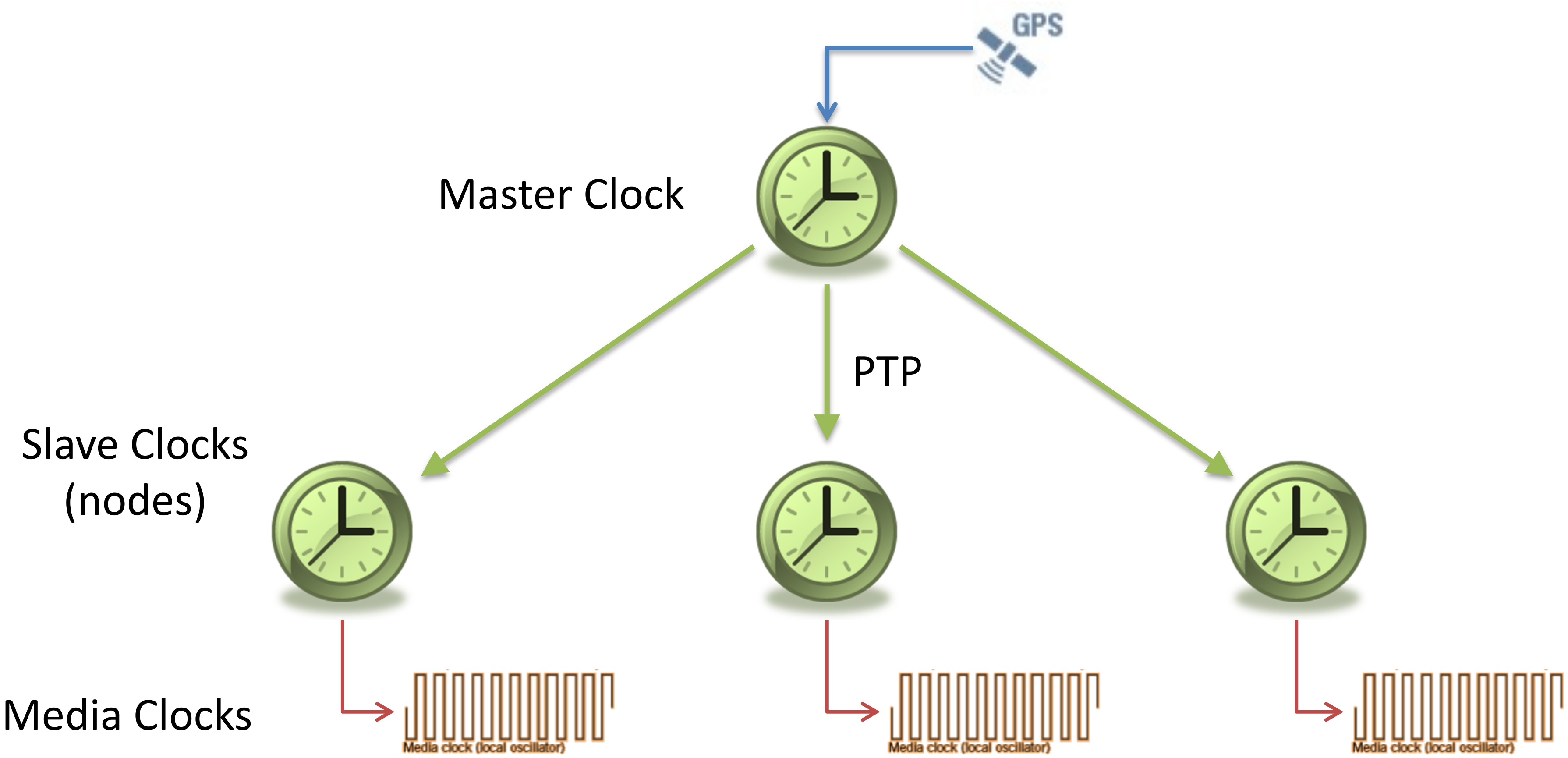
⇨ IPMX

** RTCP – RTP Control Protocol (part of RFC 3550)*

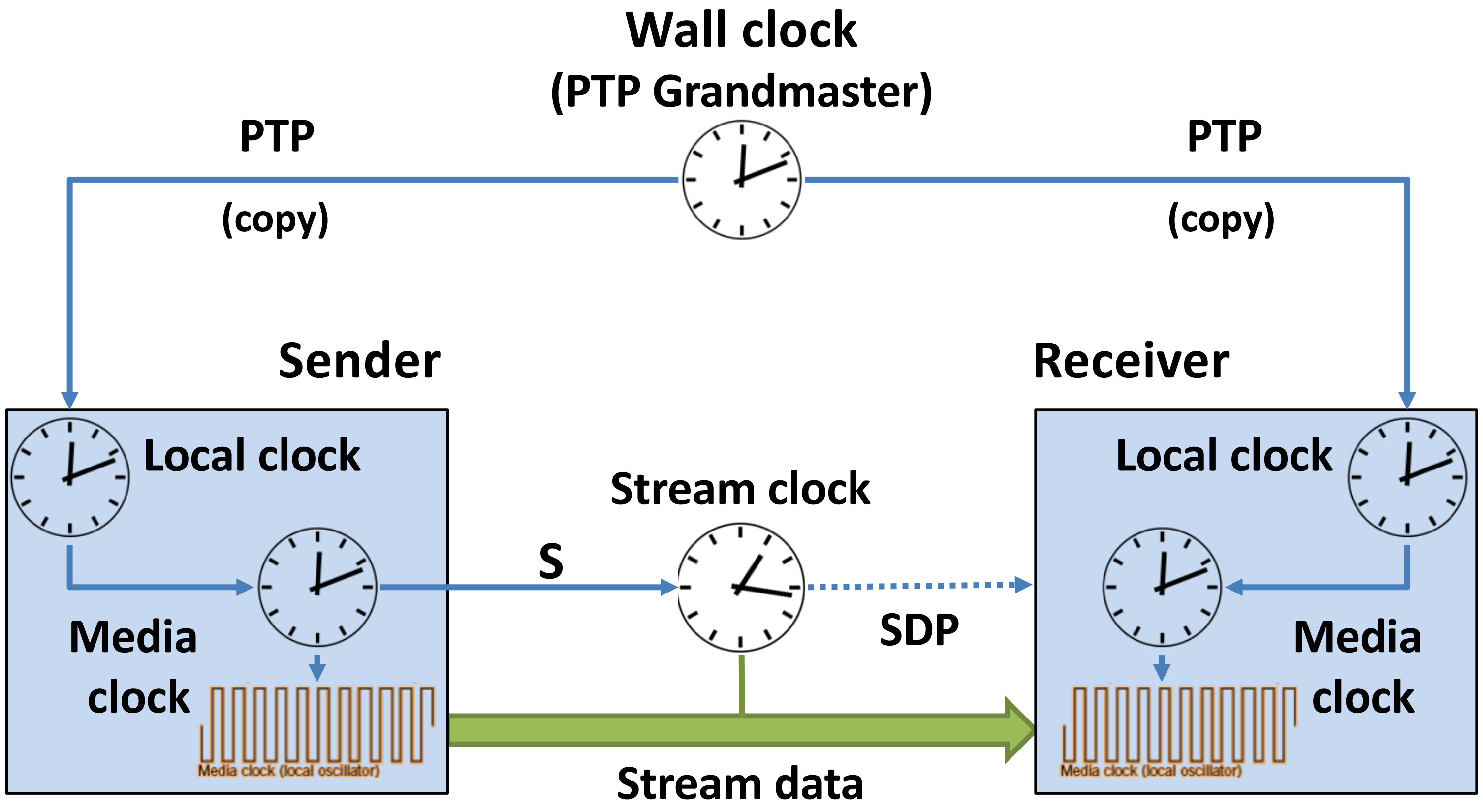
AES67 synchronization & media clocks

- All nodes are running local clocks
- Local clocks are precisely synchronized to a common wall clock via PTP
- Media clocks are generated locally from synchronized local clock

AES67 synchronization & media clocks



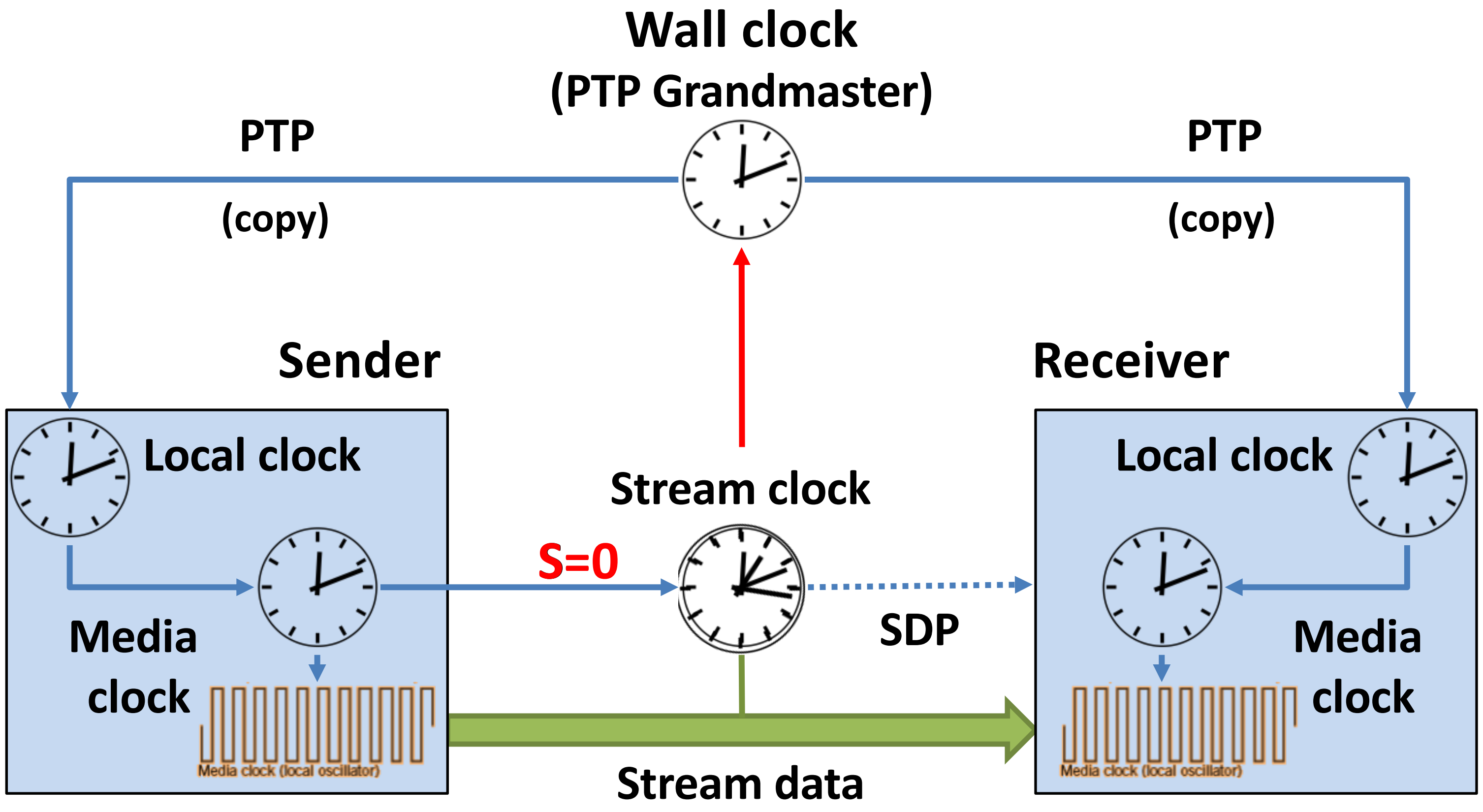
AES67 synchronization & media clocks



- Relation **S** is established on stream start-up
- **S** may be random to defeat crypto attacks
- This offset will be constant throughout the stream's lifetime
- The offset (**S**) will be conveyed via SDP (a=mediaclk:direct=<offset>)

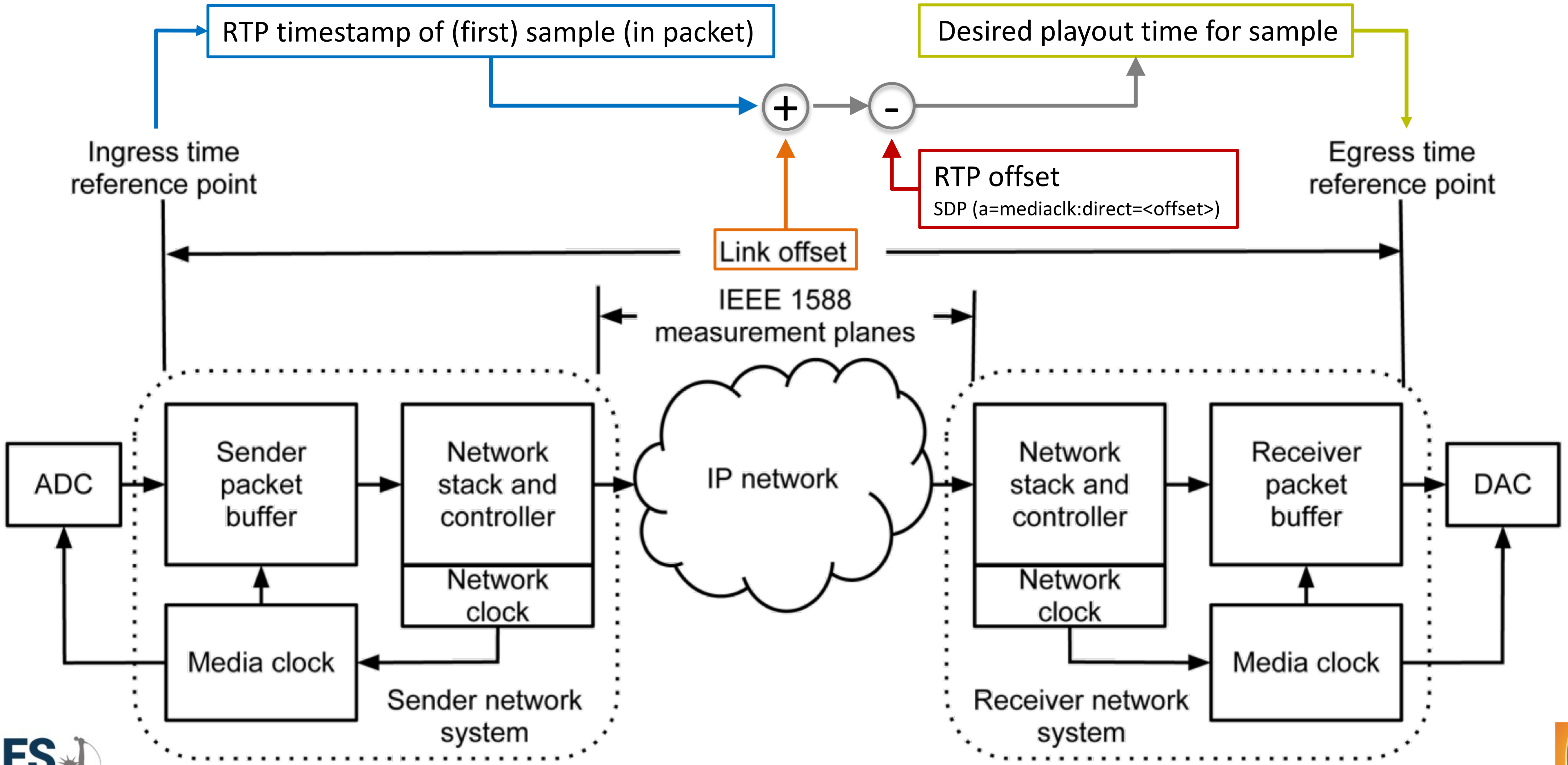
AES67 synchronization & media clocks

SMPTE ST 2110



- Relation **S** is established on stream start-up
- **S** may be random to defeat crypto attacks
- This offset will be constant throughout the stream's lifetime
- The offset (**S**) will be conveyed via SDP (a=mediaclk:direct=<offset>) **– must be "0" in ST2110**

AES67 synchronization - link offset (latency)

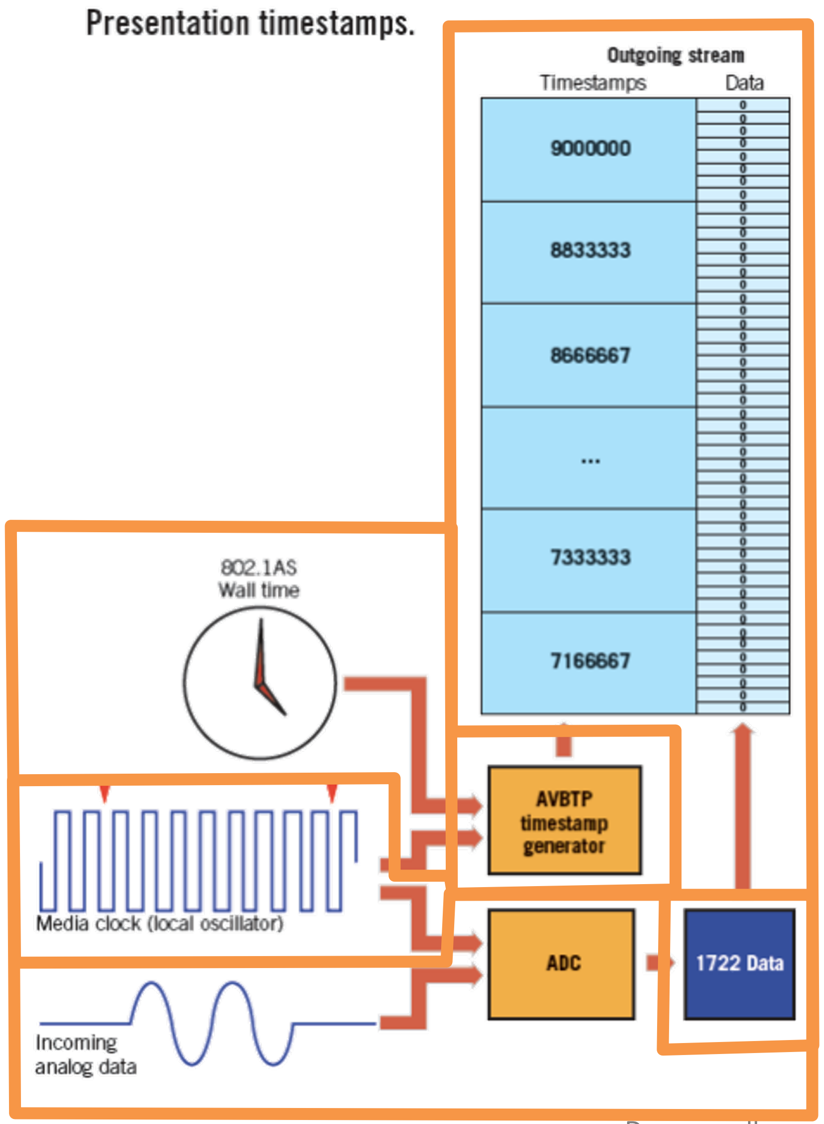


AES67 synchronization & media clocks

Main benefits / drawbacks:

- \+ All media clocks are following a common reference
  - Accurate reproduction of reference clock at all local nodes
  - Identical / phase-accurate media clocks for all streams
- \+ concurrent processing of independent streams
- \+ Deterministic latency
- \+ Sample-accurate alignment of streams possible
- \- Asynchronous input signals require SRC

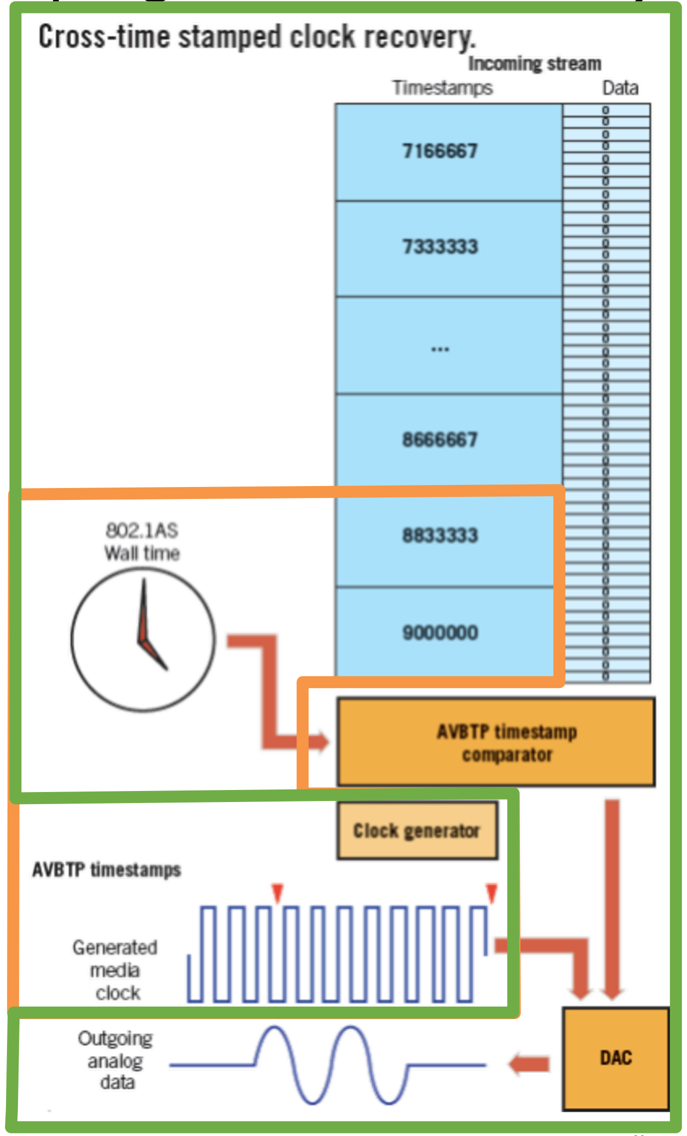
AVB timestamping & clock recovery system



Talker (sender):

- Audio is sampled with an external reference (sample) clock at the talker
- Clock edges are time-stamped with AS wall clock time
- N samples (as defined by DBC) and 1 referring time stamp w/ desired presentation time (usually sample time + 2 ms) are packetized

AVB timestamping & clock recovery system



Talker (sender):

- Audio is sampled with an external reference (sample) clock at the talker
- Clock edges are time-stamped with AS wall clock time
- N samples (as defined by DBC) and 1 referring time stamp w/ desired presentation time (usually sample time + 2 ms) are packetized

Listener (receiver):

- Receiver regenerates media clock from DBC value and subsequent time-stamp values (n clock edges within $t_2 - t_1$)
- De-packetized samples are buffered and played-out at their respective presentation time

AVB timestamping & clock recovery system

Main benefits / drawbacks:

- \+ Variable input clocking
- \+ Deterministic latency (presentation time)
- o (Precise) reassembly of input media clock
  - ▪ Depending on quality of sender / receiver clock circuitry
  - ▪ Frequency-locked, but most likely not phase-locked
- \- No concurrent processing of independent streams
  - o System-wide synchronization requires synchronization of media clocks to a reference stream
    - ⇨ Definition of CRF (*clock reference format*) in P1722-2016 (used by MILAN)

What is IPMX?

- IPMX defines a set of common, ubiquitous, open and **standards-based protocols** for **interoperability** over IP for the **ProAV** industry.
- IPMX is based on the proven SMPTE ST 2110 standard, but adapted / enhanced specifically for ProAV needs:
  - enhanced / “relaxed” timing requirements
  - supporting any bitrate with compressed or uncompressed streams
  - video format-agnostic
  - HDCP copy protection
  - network discovery and registration
  - I/O management
- The goal of IPMX is simple deployment, requiring only basic networking equipment (and knowledge?)

IPMX Clocking & Synchronization

- IPMX thrives to support the full range from
  - professional live production systems (aka SMPTE ST 2110),
  - over large synchronous / syntonous or asynchronous AV systems
  - down to a 1:1 connection between sender and receiver (i.e., HDMI source + display)
- Broadcast: devices locked to “house sync” (synchronous / syntonous operation with common reference clock)
- Non-broadcast / ProAV: asynchronous signal transport (with or without common reference cock)
- IPMX supports networks *WITH* and *WITHOUT* PTP
  - IPMX specifies what do in presence of PTP
  - IPMX specifies how to operate when PTP is not present
  - IPMX devices need to support both modes (!)

IPMX Clocking & Synchronization

Current draft confusing on several aspects:

- PTP shall be supported… (in case IPMX devices are to be integrated in broadcast applications)

  … but is not required (in “simple” applications w/o need for alignment among various senders /receivers)

  … and thus, needs not to be supported at all… (???)

- *“All IPMX Receiver should recover the Async signal timing and produce an output signal that is frequency locked to the IPMX Sender.“*

  … but mechanism of clock recovery from async stream sources not explicitly described

- Ambiguous stream alignment requirements
  - Refers back to SMPTE ST 2110 “D_{LO}” (aka AES67 “Link Offset”)

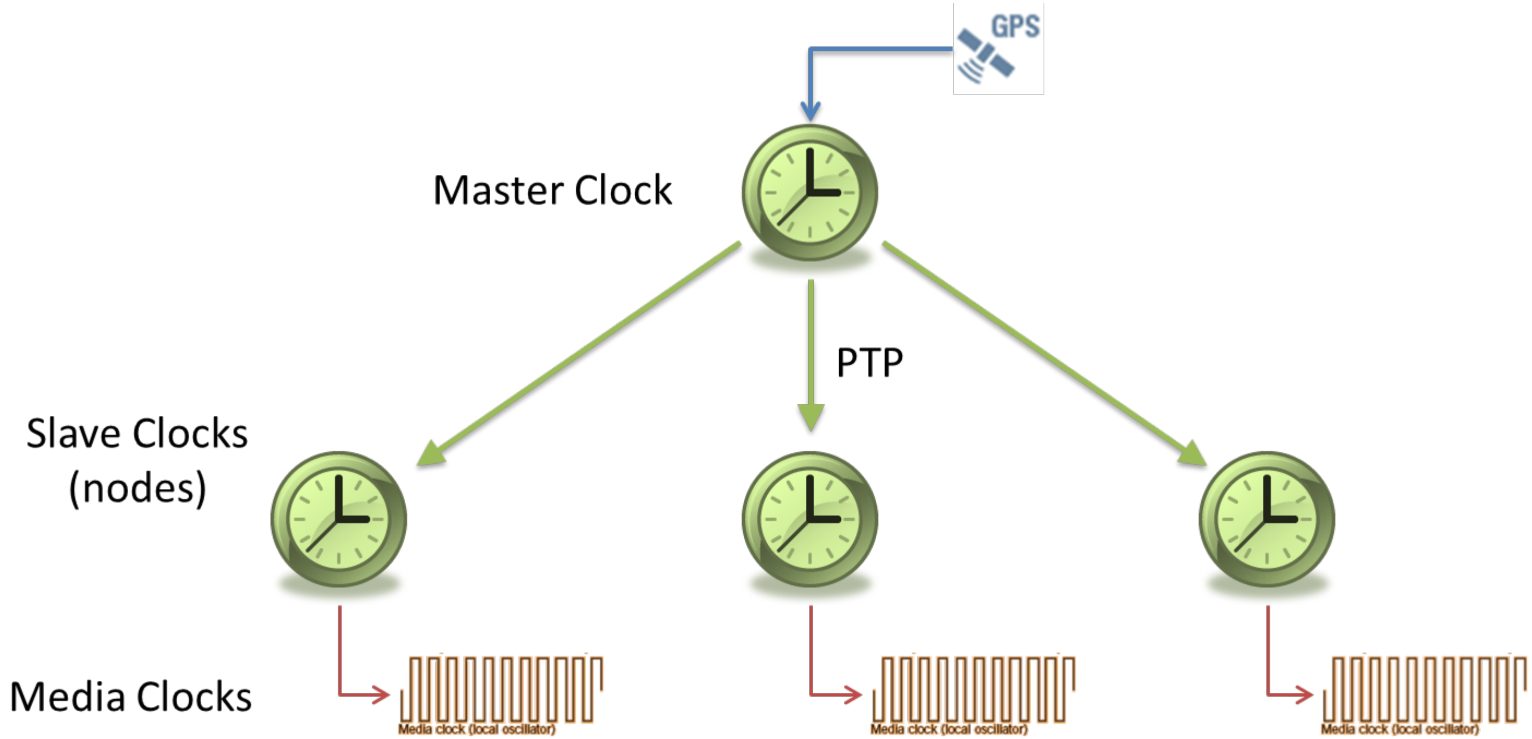
    … but works only if common reference clock (aka PTP) is present

    … or among streams originating from the same sender … (really?)

IPMX Clocking & Synchronization

IPMX basically defines 3 synchronization & clocking modes:

1) SMPTE ST 2110 (synchronous operation)
   - All input signals are synchronized by common reference clock (aka PTP)



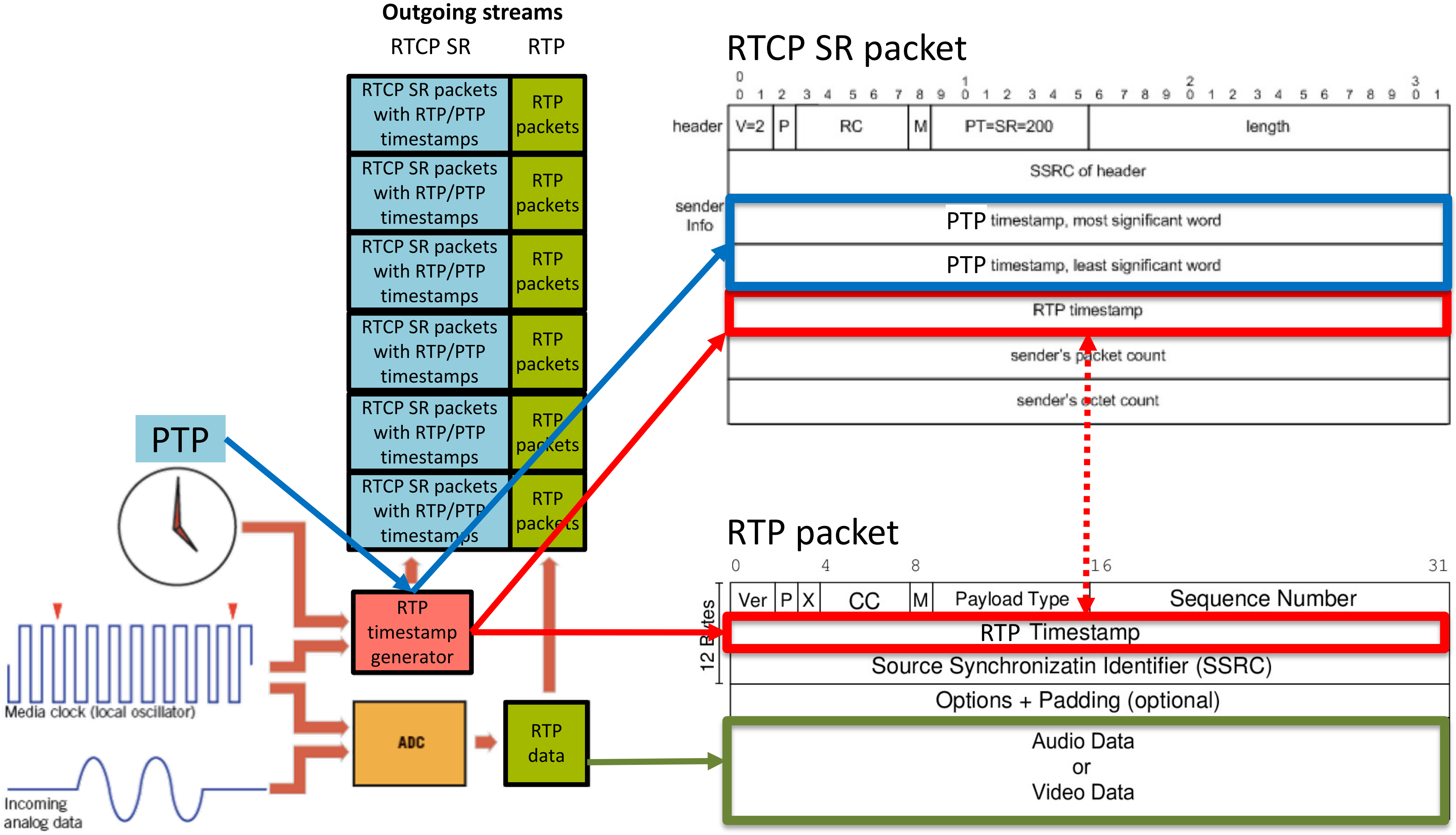
IPMX Clocking & Synchronization

IPMX basically defines 3 synchronization & clocking modes:

1) SMPTE ST 2110 (synchronous operation)
   - All input signals are synchronized by common reference clock (aka PTP)
2) RFC 3550 asynchronous operation (with common reference clock)
   - Input signals are asynchronous
   - Internal clocks are synchronized to common reference clock (aka PTP)
   - Requires RTCP SR with PTP/RTP timestamp pairs

   ⇨ Similar to AVB operation

IPMX Clocking & Synchronization



IPMX Clocking & Synchronization

IPMX basically defines 3 synchronization & clocking modes:

1) SMPTE ST 2110 (synchronous operation)
    - All input signals are synchronized by common reference clock (aka PTP)
2) RFC 3550 asynchronous operation (with common reference clock)
    - Input signals are asynchronous
    - Internal clocks are synchronized to common reference clock (aka PTP)
    - Requires RTCP SR with PTP/RTP timestamp pairs
3) RFC 3550 asynchronous (without common reference clock)
    - Input signals are asynchronous
    - Internal clocks are free-running
    - IPMX also requires RTCP with NTP/RTP timestamp pairs, but receivers do not have a reference for the NTP timestamp information (RFC 3550: NTP timestamp field may also be 0)
        - Relative difference of consecutive NTP timestamps could be used for receiver’s internal clock adjustment, but very unprecise

        ⇨ Better to be treated as RFC 3550 generic asynchronous RTP mode (receiver buffer fill level)

IPMX Clocking & Synchronization

… is actually trying to incorporate all 3 synchronization methods:

- RAVENNA/AES67 & SMPTE ST 2110-10
  - ⇨ media clock derived from common (PTP) reference clock

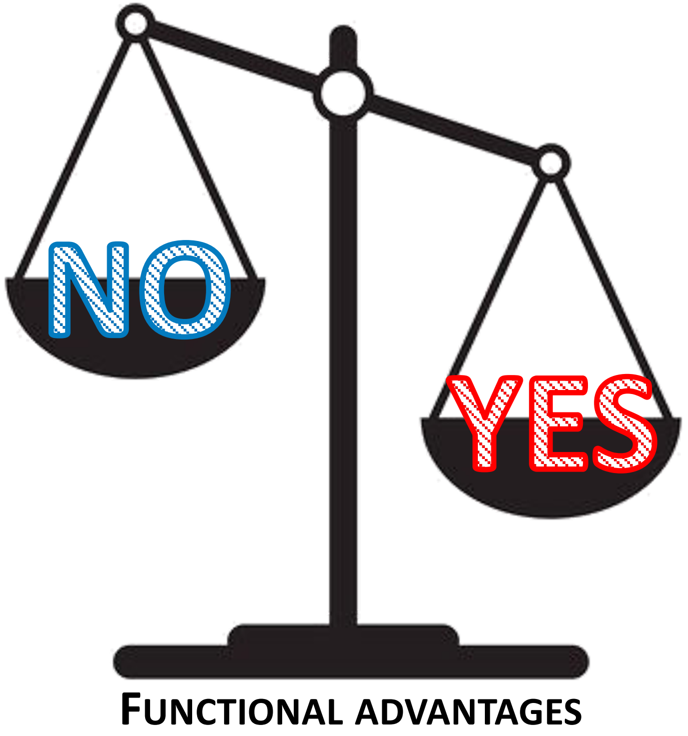
- AVB timestamping & clock recovery system
  - ⇨ free-running media clock w/ timestamping against common reference clock (RTP/PTP timestamp correlation via RTCP SR)

- RTP generic clocking
  - ⇨ free-running / asynchronous media clock with clock recovery from stream

Do we really need PTP?

PTP enables:

- Very precise clocking
- Synchronous operation of a complete system
- Transport of synchronous and asynchronous input signals
- Perfect alignment of any streams anywhere in the network
- But:
    - Most likely more complex network design



Without PTP:

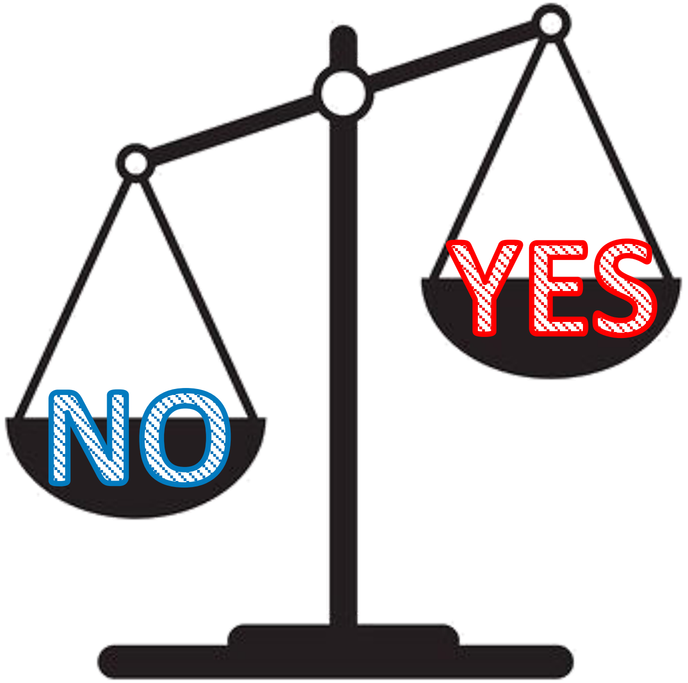
- Less complex network design
- But:
    - No synchronous operation at all (all streams / devices operate independently from each other)
    - No alignment of streams (except for streams originating from the same sender)

RAVENNA
AES67 & ST 2110 built-in

Do we really need PTP?

PTP enables:

- Very precise clocking
- Synchronous operation of a complete system
- Transport of synchronous and asynchronous input signals
- Perfect alignment of any streams anywhere in the network
- But:
    - Most likely more complex network design



EASE OF USE

Without PTP:

- Less complex network design
- But:
    - No synchronous operation at all (all streams / devices operate independently from each other)
    - No alignment of streams (except for streams originating from the same sender)

Questions?

More information…

RAVENNA / AES67 / SMPTE ST 2110 Resources:

www.ravenna-network.com/resources

www.aimsalliance.org (resources)

ravenna@alcnetworx.de

RAVENNA
AES67 & ST 2110 built-in

You've made it!

Contact information:

Andreas Hildebrand
ALC NetworX GmbH

ravenna@alcnetworx.de

CELEBRATING ~~10~~ 12 YEARS
RAVENNA
AES67 & ST 2110 built-in

www.ravenna-network.com

AES NEW YORK
AUDIO ENGINEERING CONVENTION

ALC NetworX