

Clock Reference Stream Improvements Proposal

Frank Bühren, Intel

August 2019

Issues with Current CRF Definition (1)

- PTP-related time stamps are absolute, but media clock is incremental
 - Have to “count” CRF PDUs and the number of contained time stamps, then multiply with timestamp_interval field to determine events elapsed
 - As opposed to the statement in 10.1

CRF is tolerant of lost packets since clocks free-wheel between defined clock points

a lost packet means I don't know how many events have elapsed, I am thus unable to maintain relation between media clock and PTP clock

- I could extrapolate and/or guess from seq_num how many packets I have missed, but that's more of a heuristic – or outright cheating.
- Being more tolerant to lost packets would resolve the conflict between not reserving bandwidth (10.3.1) and still running in the SR traffic class

Issues with Current CRF Definition (2)

- Media Clock is represented in events that, in turn, represent elapsed media time
 - Several specific media event types defined (audio samples, video lines, video pixels, etc.) but at the end of the day, they all represent **media clock time**.
- Why not introducing a generic “media clock” type that can be used for everything?
 - Has to address a few details to enable general use

Unrelated question:

- When would we ever need more than two timestamps in the same PDU? What for? Is anyone using non-linear extrapolation for time conversion?

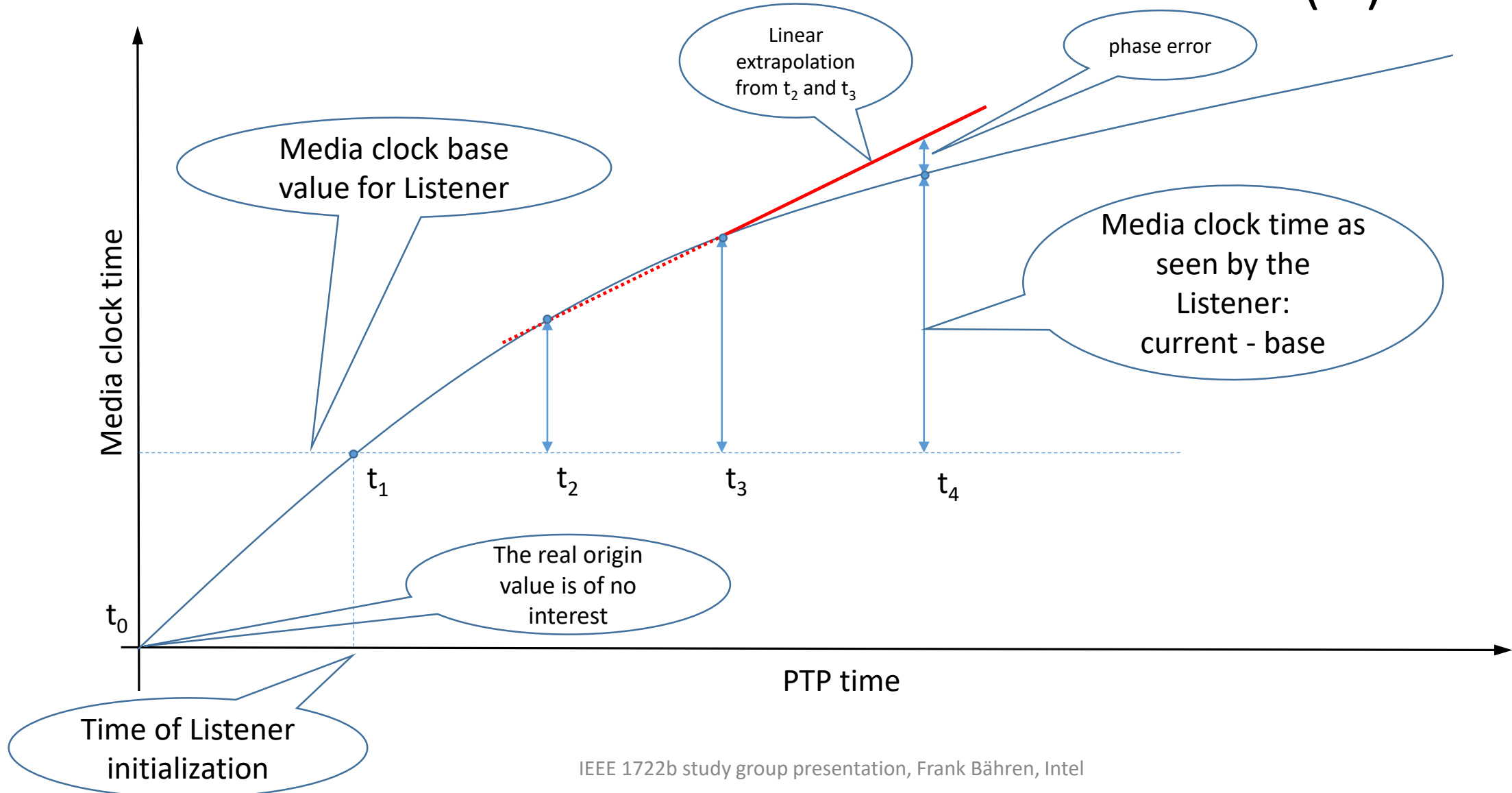
Introduction to Generic Media Time Type

- We still have “events”, but they are just time ticks
- Event durations that have an infinite fractional part can cause accumulating rounding errors, e.g. $48\text{kHz} \rightarrow 20.8\bar{3}\mu\text{s}$
Need to express the time span of an event in a fractional way, i.e. numerator and denominator, ideally in the irreducible form.
- There is often still an underlying idea of what the “event” represents (e.g. an audio sample interval), but it only shows in the event granularity, and not in a denoted “type”
- Often, but not always, an event granularity can be found that allows the use of the same media clock for multiple purposes (e.g. audio and video synchronization)

Introduction to Absolute Media Time (1)

- Instead of having only PTP-related time stamps in a CRF PDU, have one or more tuples of (PTP time stamp, media time stamp), both in absolute time
- Upon initialization/reset, CRF Listeners take first received media time stamp and pin this as local reference, only looking at the subsequent increments → The origin (t_0 where media time value = 0) doesn't really matter
- You can ignore/miss multiple intermediate PDUs and are still able to maintain PTP \leftrightarrow media clock relation
 - Maximum time span between PDUs that have to be received & processed is determined by phase error requirement
- Might require epoch version field rather than simple MR toggle

Introduction to Absolute Media Time (2)



Questions

- Is there value in adding a definition of a CRF with absolute time stamps for both PTP and media clock?
- Is there value in adding a generic “media clock time” CRF type?

Note: Both of these additions are orthogonal to each other.

- Can they be added to the standard in a backward-compatible way?