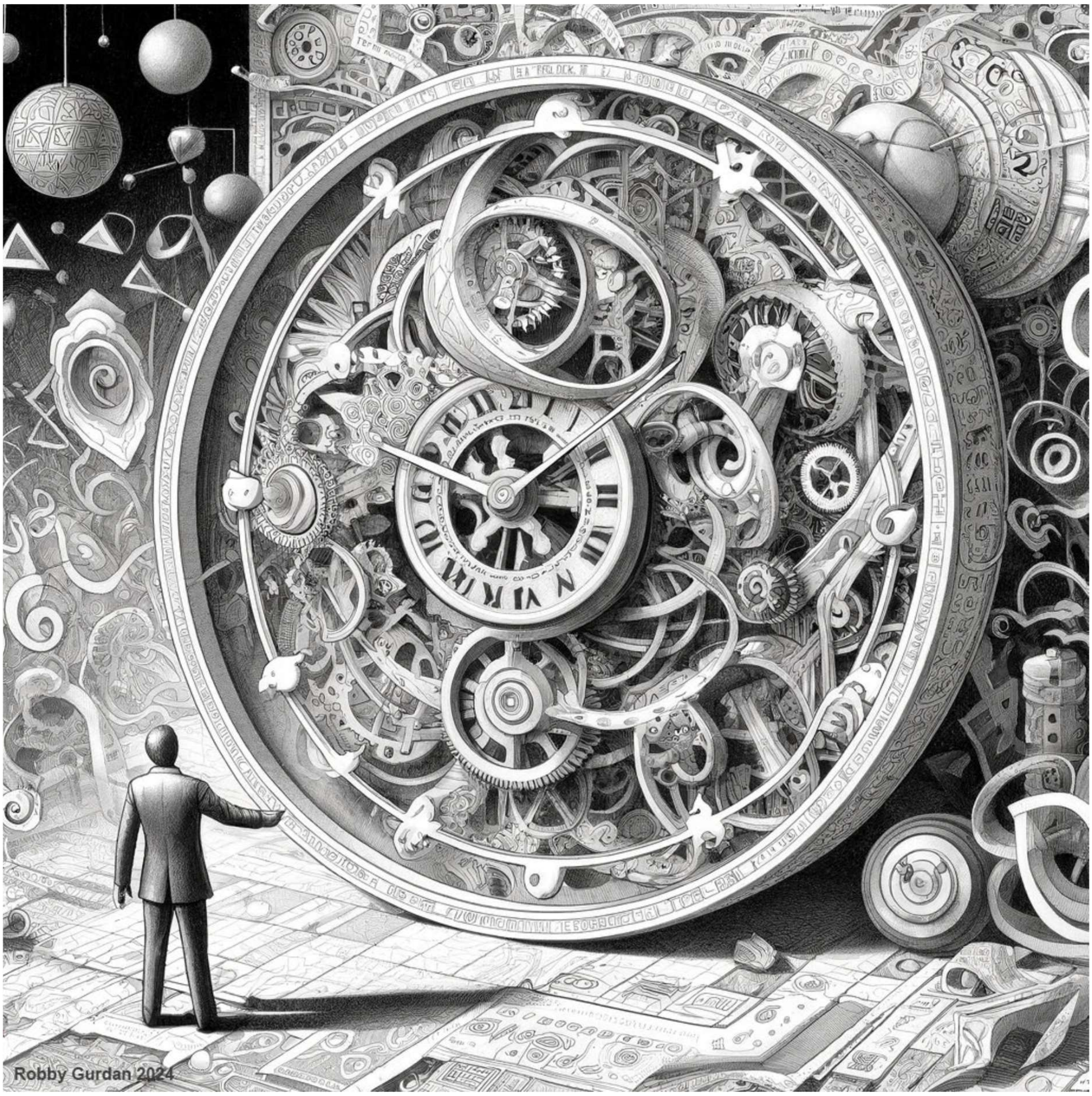
Understanding the essence of time

Part 4

Streaming Data and Media Clocks in time sensitive networking (Part 2)



In the first part of this series, we explored why streaming data requires independent media clocks and how they differ from PTP (Precision Time Protocol). We also examined various degrees of synchronicity, ranging from "similar" to "synchronous," and defined these terms.

In the second instalment we will delve into the mechanisms of media clock distribution and recovery within IEEE 1722 Streams, understand synchronisation processes, and consider additional critical factors for streaming solutions over Ethernet/ TSN.

Understanding Media Clock Transmission and Recovery in IEEE 1722

Within TSN standards, IEEE 1722 is crucial for managing real-time data streams over Ethernet networks. One of the key aspects of this protocol is the management of media clocks to maintain synchronization across network devices. This article explores the transmission of a media clock from a talker and its recovery by a listener within the IEEE 1722 framework, highlighting the precision and complexity involved.

What is a Media Clock?

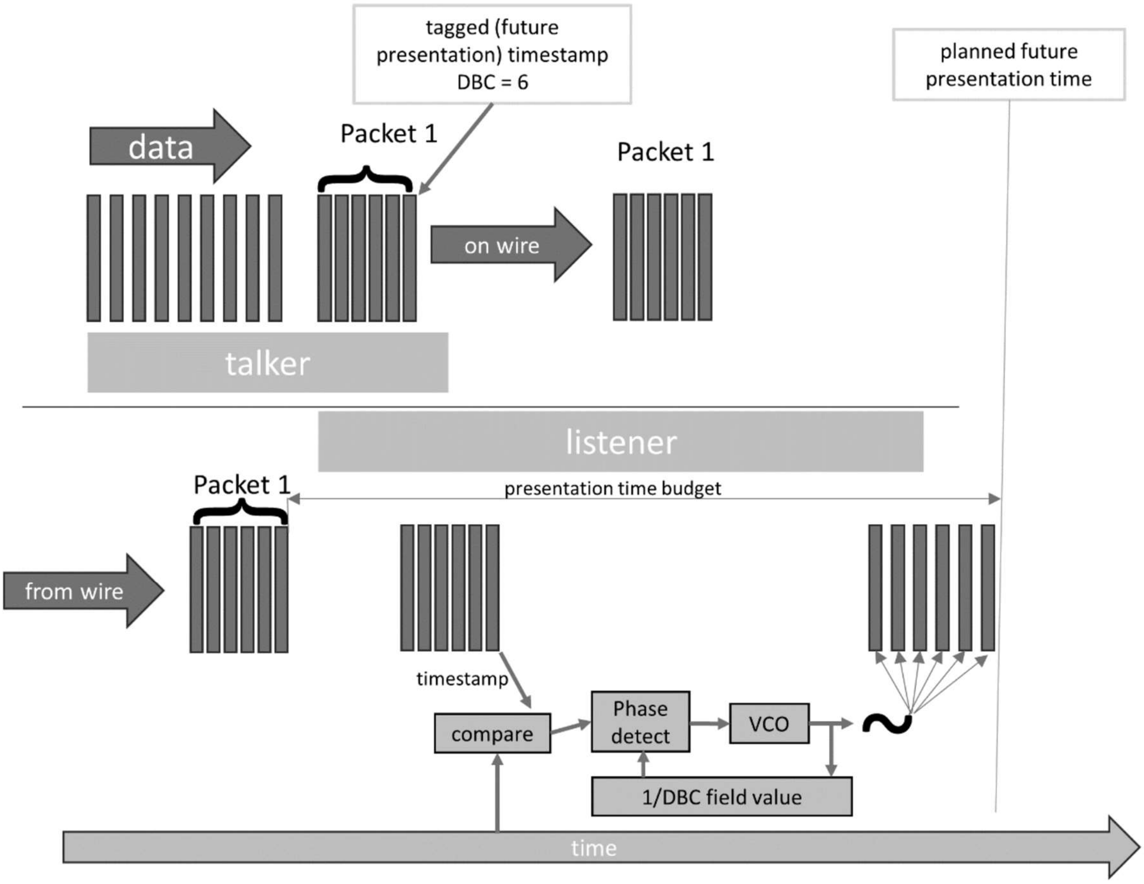
A media clock is vital for synchronization; it coordinates the timing of media stream presentations. For more details, please refer to Part 1 of this series.

Media Clock Transmission by the Talker

In IEEE 1722, the "talker" refers to the device sending streaming data. It embeds media clock information within the data packets it transmits within the IEEE 1722 header, which includes, among many other things, timing information tied to the media stream:

1. **Timestamps:** Each packet from the talker contains a timestamp based on the talker's PTP wall clock, plus a margin (e.g., 2ms) to set the "presentation time." This indicates when the listener should play out the data.
2. **Clock Identity:** Packets also carry a unique identifier for the talker's clock, indicating to listeners which PTP clock should be used to recover the media clock in case there are more than one.
3. **Stream Identification:** The stream ID in each packet uniquely marks the media stream, allowing listeners to associate packets with their respective streams and clocks correctly.

This continuous and consistent transmission allows listeners to maintain synchronization despite potential system-induced delays and jitter.



Pic 1 shows the principle how media clocks in 1722 are distributed and recovered.

Media Clock Recovery by the Listener

The primary role of a listener is to receive data packets from a talker. Upon receiving the data packets, the listener undertakes the critical task of recovering the media clock. This process involves several key steps:

1. **Media Clock Synchronization:** The listener uses the (AVTP) timestamps and the clock identity in the received packets to synchronize its media clock with the talker's media clock. By using a PLL mechanism to smooth out any larger jumps, a valid media clock can be successfully recreated at the listener side.
2. **Buffering and Timing:** The listener typically buffers the incoming packets to accommodate variations in network delay. The buffered data is then played out according to the recovered clock, ensuring that the playback is syntonized with the talker's timing. Using the presentation time, this mechanism can, therefore, achieve perfect synchronization of listeners across the network, provided that each listener maintains a positive presentation time budget at the time of receipt.

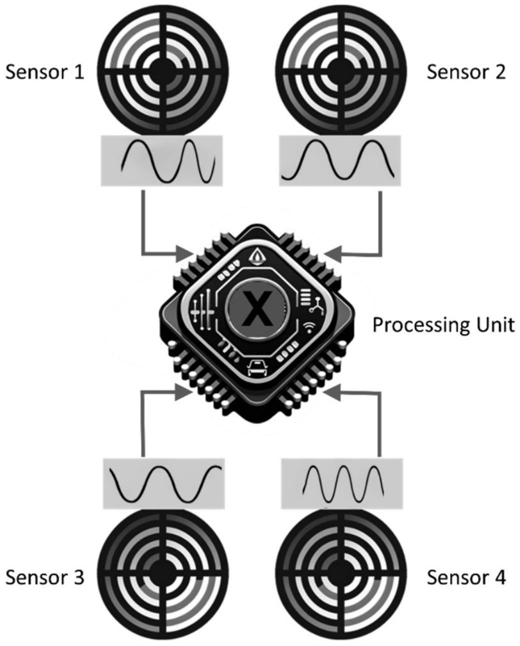
Challenges and Solutions

Distributing a talker's media clock to listeners introduces new challenges. For example, when four radar sensors connected to a processing unit each send an IEEE 1722 data stream with an embedded media clock from different sources, it necessitates re-clocking all streams, adding latency and complexity—undesirable in time-sensitive environments.

In the example to the right, we can see four Radar Sensors connected to a processing unit. Each radar Sensor sends out an IEEE 1722 data stream with an embedded Media Clock as discussed before.

Unfortunately, all four streams are coming from four different entities, resulting in the necessity to re-clock all streams, adding latency, complexity, and resources.

Especially in time sensitive environments this is not desirable.



Pic 2: Four radar sensors send data to a computing device.
Each of the sensors have an independent Media Clock.

A stream to clock them all ...

A simple solution involves multicasting a standard IEEE 1722 stream to all devices, allowing them to recover and synchronize their local media clocks, effectively making each device both a talker and a listener.

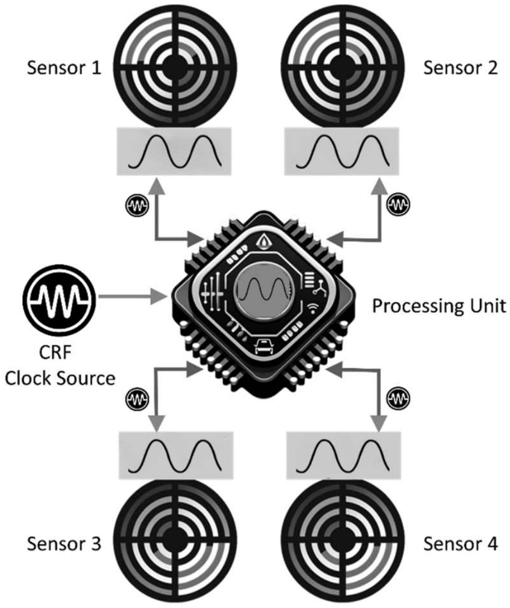
The disadvantage of this solution is, that some noise is sent unnecessarily throughout the network, as the payload of those sync streams is useless and consumes unnecessarily bandwidth. Furthermore, those streams need to be prioritized and reserved in the network. Nevertheless, for IEEE 1722-2011 the first 1722 Standard, this was the sole possibility to do exactly that.

CRF to the rescue ...

With the first revision of IEEE 1722 (IEEE 1722 - 2016) quite some new features were introduced. This revision introduced the Clock Reference Format (CRF), which eliminates the need to transmit redundant data for synchronization but is still enabling each device to adjust its media clock to a common beat. Prioritization and stream reservation are optional.

CRF streams contain, among other things:

- the sample's presentation time,
- the nominal sample rate,
- the number of samples in between 2 timestamps,
- may hold multiple timestamps in one packet,
- the media clock domain.



Pic 3: Four radar sensors send data to a computing device. Each of the sensors is synchronized through a CRF stream.

Talkers can use the distributed common clock within a system to synchronize stream data such that an AVTP Listener can receive streams from multiple Talkers and know that the streams are all synchronized with each other.

This allows use cases such as video switching, audio mixing, and synchronized sensor information to be designed with less complexity.

In a video system, the switching from one camera to another can be simplified if both cameras are genlocked. Likewise, the sensors of the above system can stream latency free and be processed without additional conversion if all devices are synchronized.

In time-sensitive networking with extremely low latency demands, this plays a vital role in achieving microsecond latencies from talker to listener while using zonal architectures and a shared data network.