Understanding the essence of time

Part 3

Streaming Data and Media Clocks in time sensitive networking (Part 1)

Author: Robby Gurdan

Introduction

In applications where data streams are distributed across various devices, maintaining synchronization across those nodes of the payload is essential - any deviation in timing can lead to errors or glitches. Shared media clocks serve as the answer to this demand.

This article marks the third installment in a series exploring the critical role and management of time and clocks within time-sensitive networks. The initial segments that delve into the differences between local and distributed clocks and introduce the topic of gPTP, are available for download on the TSN Systems website at https://www.tsn.systems.

Media Clocks in time sensitive networking

As clock signals in networks cannot be sent directly to their sinks, an encapsulated clock mechanism and clock recovery mechanism are needed. The former establishes the clock-accurate transport throughout the network, and the latter ensures that each device in the network can reconstruct the original stream timing accurately, leading to a seamless media experience.

What is streaming data?

Streaming data refers to data that is continuously generated, transmitted, and processed in real-time or near-real-time. Unlike traditional batch data processing, where data is collected, stored, and analysed in large chunks at periodic intervals, streaming data involves the immediate handling of data as soon as it is generated. This approach allows for the instantaneous analysis and response to information, making it particularly useful in many automotive scenarios where time-sensitive decisions are critical.

Because of this continuous and real-time nature, buffering and re-clocking/-organizing data is not possible. Moreover, in some cases data processing must compute the data synchronously on different devices with at least a single-digit microsecond accuracy.

What is a media clock?

In general, media clocks serve as a fundamental component within digital media systems, playing a pivotal role in:

- Ensuring synchronization.
- Providing a standardized timing reference.
- Facilitating seamless integration and playback of varied media types.

These functions render media clocks indispensable in a modern digital streaming data environment. Since the dawn of networks, the digital media landscape has been increasingly shaped by shared real-time media, streaming control payloads, and sensor data. This evolution underscores the critical role of media clocks in maintaining consistency across a broad spectrum of platforms and technologies.

What is a distributed media clock?

Distributed media clocks are particularly vital for syncing streaming data across networks. Their key roles are:

- Keeping a consistent stream of data to prevent buffer issues.
- Providing a means to continuously adjust clock frequencies, thus allowing slave devices to effectively align with master clocks.
- Ensuring uniform clock frequencies for syntonous data synchronization.
- Ensuring consistent phase and frequency for synchronous data.

Achieving similar, syntonous, or synchronous clocks requires the master clock's frequency and, if necessary, phase to be transmitted swiftly and accurately through the network. However, in that approach, media clocks differ significantly from conventional clocks (e.g., wall clocks). In contrast and in addition to wall clocks, they must also be designed to:

- Suddenly and immediately alter their clock frequency.
- Adjust their phase and, in certain instances, even alter their base frequency.

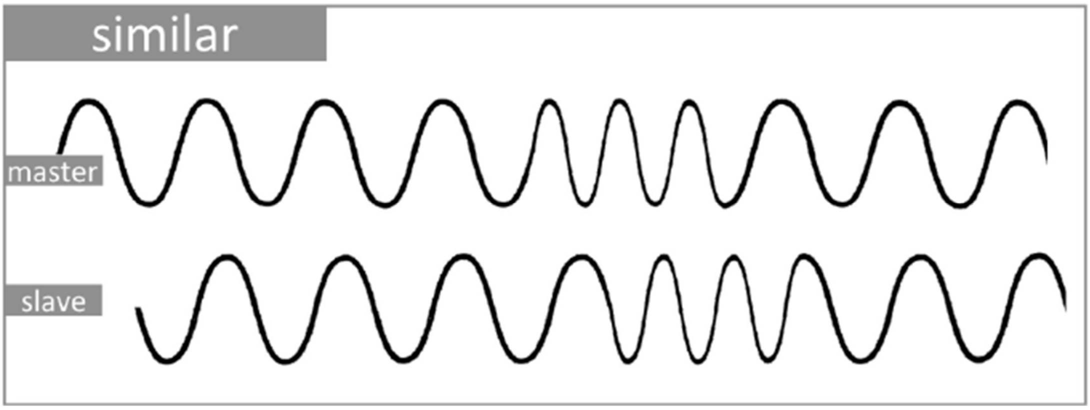
As a result, media clock slaves need to be much tighter coupled to media clock masters as PTP slaves must be to their masters.

This necessitates a more dynamic updating mechanism than what's offered by protocols like gPTP, underscoring the need for media clocks to operate independently of PTP despite some level of interdependence.

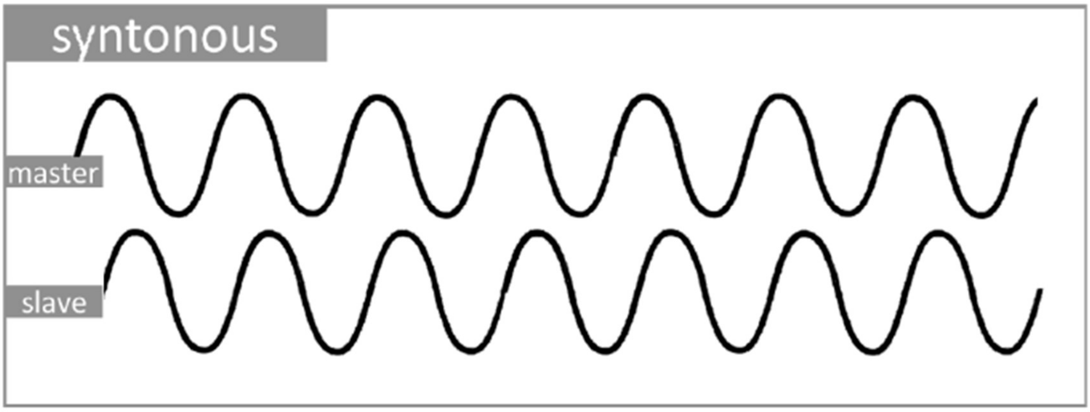
What means “synced”?

This concept lacks a clear definition and is applied in various contexts. To clarify, we can explore three distinct interpretations that encompass the majority of its applications, ranging from the broadest to the most specific forms of synchronization. Starting with the most basic understanding:

Similar:
This is the minimal requirement for any slave clock operating within a distributed media clock system. A slave clock that is "similar" tracks the frequency adjustments of its clock master. Although it may not maintain exact phase alignment, it remains sufficiently close to prevent buffer shortages or excesses, a necessity for all slave clocks by principle. However, the ultimate test of whether data is not in exact sync or frequency alignment, but merely similar is determined by examining the payload. In scenarios where the data stream is sample-based and the jiter of the slave clock causes the timing of sample processing to deviate by at least the duration of one sample in any direction, the system is neither in sync nor in frequency alignment. Nonetheless, if the buffers can accommodate this variation, a single source could still successfully transmit to a single sink without issue. Achieving this indicates the system is functionally synchronized.



Syntonous:
Syntonous timekeeping aligns with the concept of similar clocks but with a notable distinction that hinges on the granularity of the payload or the precision of its presentation. In the provided example below, this implies that jiter is minimized to the extent that no sample shift occurs, though a phase displacement in the resultant waveform may still be present. Consequently, while a syntonous sink's reconstructed waveform appears consistent, it could be temporally shifted, resulting in the stream data from various sinks not being presented simultaneously. It is important to note that the phase relationship is completely static and does not jiter/ wander with time. A syntonous systems often has the challenge that although the phase offset is static once locked, the offset is not guaranteed to be the same every time – if this was known it could be used with a feedback mechanism to effectively become synchronous! Syntonous clocks may be acceptable for some applications based on the amount and range of the offset.



Synchronous:
Finally, synchronous clocks would be both frequency and phase-accurate, so the slave's curve would perfectly match that of its master. Synchronous sinks' recovered clocks would look the same at each point in time and would enable each sink to present its data at the exact same moment.