

Comments on „60802-McCall-Clocks- Normative-Requirements-0223-v01.pdf“

Dragan Obradovic – Siemens AG

Definitions

~~Slave~~: Receiver
~~Master~~: Transmitter

- **10.2.13.2.1 updateSlaveTime():** Updates the global variable clockSlaveTime (see 10.2.4.3), based on information received from the SiteSync and LocalClock entities. It is the responsibility of the application to filter slave times appropriately (see B.3 and B.4 for examples).
- **10.2.4.3 clockSlaveTime:** The synchronized time maintained, at the slave, at the granularity of the LocalClock entity [i.e., a new value is computed every localClockTickInterval (see 10.2.4.18) by the ClockSlave entity]. The data type for clockSlaveTime is ExtendedTimestamp.
- **10.2.4.18 localClockTickInterval:** The time interval between two successive significant instants (i.e., “ticks”) of the LocalClock entity. The data type for localClockTickInterval is TimeInterval.

→ My interpretation:

- The state machine has to provide synchronized time at any significant instant of the LocalClock. → It defines a clock!
- It is not specified how. Only an example is given, but it is not normative.
- Using linear extrapolation can also be seen as filtering (this is a first order system)
- It is not explained how the “quality” of the synchronized time can be measured in the case of discontinuities (as in the provided example), which is needed for determining if the PTP-Instance is in the Synced state or not

Clock~~Slave~~Sync state machine

Slave: Receiver
Master: Transmitter

802.1AS-2020 10.2.13

Clock~~Slave~~ReceiverSync state machine

10.2.13.2 State machine functions

10.2.13.2.1 updateSlaveTime(): Updates the global variable clockSlaveTime (see 10.2.4.3), based on information received from the SiteSync and LocalClock entities. **It is the responsibility of the application to filter slave times appropriately** (see B.3 and B.4 for examples). As one example, clockSlaveTime can be:

- a) Set to syncReceiptTime at every LocalClock update immediately after a PortSyncSync structure is received, and
- b) Incremented by localClockTickInterval (see 10.2.4.18) multiplied by the rateRatio member of the previously received PortSyncSync structure during all other LocalClock updates.

If no PTP Instance is grandmaster-capable, i.e., gmPresent is FALSE, then clockSlaveTime is set to the time provided by the LocalClock. This function is invoked when rcvdLocalClockTickCSS is TRUE.

10.2.13.2.2 invokeApplicationInterfaceFunction (functionName): Invokes the application interface function whose name is functionName. For the ClockSlaveSync state machine, functionName is clockTargetPhaseDiscontinuity.result (see 9.6.2).

- The provided example is also a type of filtering as much as using a control loop is
- The difference is that in the provided example arbitrary jumps are allowed while in the control-loop example this is not the case.
- The measuring of the „quality“ of the synchronized time is in the provided example not clear (it could be done at the arrival of the next Sync message, but the observed error does not have any influence on the future calculation of the synchronized times)

Where do this analysis leads us to?

~~Slave~~: Receiver
~~Master~~: Transmitter

- This state machine triggers calculation of the synchronized time whenever it is needed!
- It does not specify how, but gives an example.
- By providing the synchronized time when needed, it is also a clock!
- It indirectly claims that the given example is not filtering and leaves filtering to the ClockTarget

- BUT: the calculation of the synchronized time can be done only by applying a computational scheme which can be seen as filtering (at least in the language used in this group)!

- Questions:
 - can we live with the statement that the calculation of the synchronized time is left to the application and not specified here (the state machine only triggers it)?
 - If we do not go into specifying the calculation, how do we judge the quality of the synchronized time (the 1 μ s requirement)? Or is it also left to the ClockTarget?
 - If the calculation is not specified, is the only normative requirement that the RR and MasterTimeEstimate have to be updated before forwarding them to the next element (again not specifying how)? Nothing else is needed in that case, in my opinion.