Pecision Time Protocol (PTP): IEEE 1588-2019

Cason Konzer

casonk@umich.edu

October 8, 2023

Outline

- Introduction:
  - Opening Discussion
  - Use Cases
- Clock Types:
  - Grandmaster, Master, Slave
  - Ordinary, Boundary, and Transparent Clocks
- Synchronization & Synotization:
  - Mean Path Delay
  - Mean Link Delay
  - Delay Asymmetry
- Best Master Clock Algorithm:
  - State Decision Algorithm
  - Data Set Comparison Algorithm

Opener

Use Cases

Why do we need a Time Protocol?

- We are building *time critical* *distributed & deterministic* systems.
  - Particle Colliders
  - Power Plants & Nuclear Reactors
  - Missile Systems & Flight Controllers
  - Automated Driver Assistance Systems (ADAS)
  - High Frequency Trading

Outline Review

- Introduction:
  - Opening Discussion
  - Use Cases
- Clock Types:
  - Grandmaster, Master, Slave
  - Ordinary, Boundary, and Transparent Clocks
- Synchronization & Synotization:
  - Mean Path Delay
  - Mean Link Delay
  - Delay Asymmetry
- Best Master Clock Algorithm:
  - State Decision Algorithm
  - Data Set Comparison Algorithm

Grandmaster, Master, Slave

Grandmaster, Master, Slave

- Grandmaster:
  - The ground truth time source, good to think of as a correction service provider.
- Master:
  - Provides time to slave clocks, is corrected based on the grandmaster.
- Slave:
  - Determines time as provided by a master.

Ordinary, Boundary, and Transparent Clocks

- Ordinary Clock:
  - Has a single PTP port.
  - Can act as either a *master or slave*.
- Boundary Clock:
  - Has a several PTP ports.
  - Can act as both a *master and slave*.
  - Any *master* corrected by a *grandmaster*.
- Transparent Clock:
  - Is neither a *master* or *slave*, but rather an *intermediate correctional device*.

Additional Note on Transparent Clocks

Transparent clocks measure what is called *residence time*, time spent traversing the device.

This is added to correctional fields of PTP event messages.

For *end-to-end* transparent clocks, this is all.

For *peer-to-peer* transparent clocks, ‘peer delay’ is additionally supported.

Outline Review

- Introduction:
  - Opening Discussion
  - Use Cases
- Clock Types:
  - Grandmaster, Master, Slave
  - Ordinary, Boundary, and Transparent Clocks
- Synchronization & Synotization:
  - Mean Path Delay
  - Mean Link Delay
  - Delay Asymmetry
- Best Master Clock Algorithm:
  - State Decision Algorithm
  - Data Set Comparison Algorithm

Mean Path Delay and Mean Link Delay

The mean delay measures the average transmission time between two devices.

- Mean Path Delay:
  * uses a delay request-response mechanism.
- Mean Link Delay:
  * uses a peer-to-peer delay mechanism.

Mean Path Delay

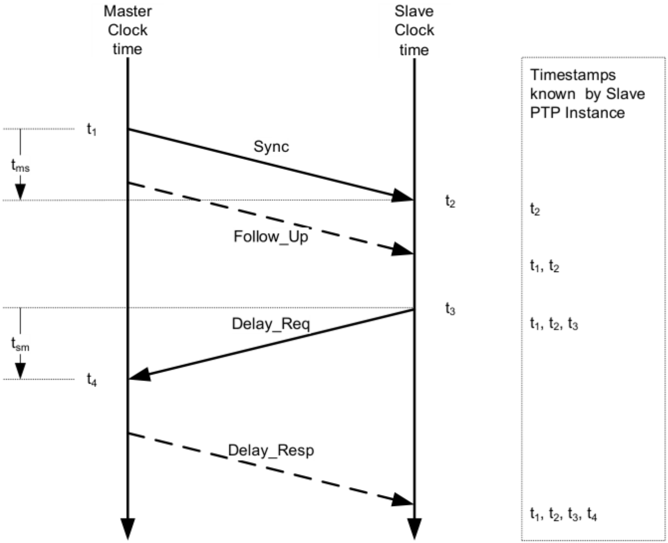
Mean Path Delay is measured as:

$$\frac{t_{m,s} + t_{s,m}}{2}.$$

Mean Path Delay is used for synchronization to master clocks. Master clocks distribute sync messages to their slaves.

This is a *distributed* correction service.

Mean Path Delay



[1]

[1][1] Delay request-response path length measurement

Mean Link Delay

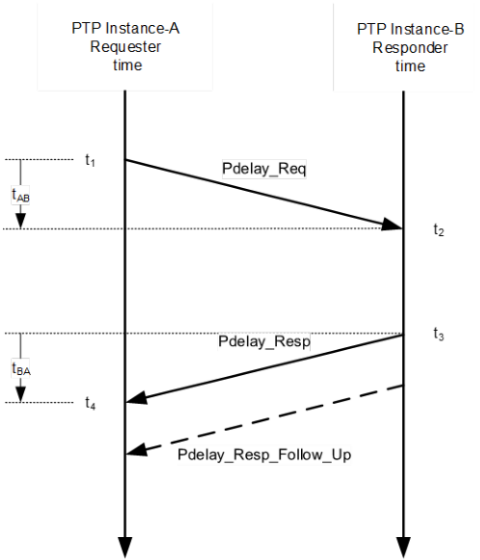
Mean Link Delay is measured as:

$$\frac{t_{res,req} + t_{req,res}}{2}.$$

Mean Link Delay allows for synchronization when Mean Path Delay is not supported e.g. transparent clocks.

This is an *on demand* correction service.

Mean Link Delay



[2]

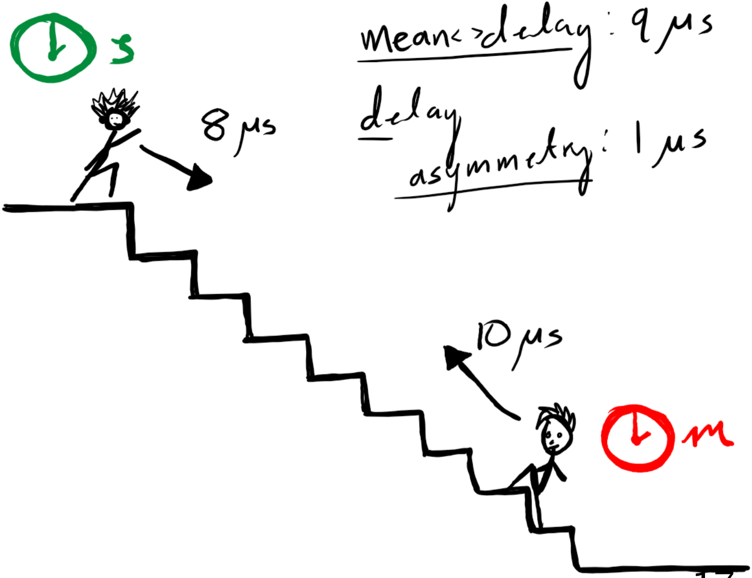
[2][1] Peer-to-peer delay link measurement

Delay Asymmetry

The delay asymmetry measures the difference in timing based on the direction in which a path is traversed.

- $< dA > \quad = \quad < delayAsymmetry >$
- $t_{m,s} \quad = \quad < meanPathDelay > + < dA >$
- $t_{s,m} \quad = \quad < meanPathDelay > - < dA >$
- $t_{res,req} \quad = \quad < meanLinkDelay > + < dA >$
- $t_{req,res} \quad = \quad < meanLinkDelay > - < dA >$

Delay Asymmetry



Outline Review

- Introduction:
  * Opening Discussion
  * Use Cases
- Clock Types:
  * Grandmaster, Master, Slave
  * Ordinary, Boundary, and Transparent Clocks
- Synchronization & Synotization:
  * Mean Path Delay
  * Mean Link Delay
  * Delay Asymmetry
- Best Master Clock Algorithm:
  * State Decision Algorithm
  * Data Set Comparison Algorithm

State Decision Algorithm

The state decision algorithm is used to determine if a clock shall change its state.

Each PTP port can exist is one of the following states for full and slave-only implementations respectively:

- Initializing (F, S)
- Faulty (F, S)
- Disabled (F, S)
- Listening (F, S)
- Pre-Master (F)
- Master (F)
- Passive (F)
- Uncalibrated (F, S)
- Slave (F, S)

Data Set Comparison Algorithm

The data set algorithm compares user configured, estimated and constructed clock attributes and outputs the better of the two clocks.

The following values are compared:

1. Priority1 (UC)
2. Class (C)
3. Accuracy (E)
4. Offset Scaled Log Variance (E)
5. Priority2 (UC)
6. Identity (C)

Outline Review

- Introduction:
  - Opening Discussion
  - Use Cases
- Clock Types:
  - Grandmaster, Master, Slave
  - Ordinary, Boundary, and Transparent Clocks
- Synchronization & Synotization:
  - Mean Path Delay
  - Mean Link Delay
  - Delay Asymmetry
- Best Master Clock Algorithm:
  - State Decision Algorithm
  - Data Set Comparison Algorithm

My experience

I have seen PTP implemented for automotive safety & ADAS.

Typically the PTP network encompass high fidelity reference sensors which are then used to calibrate less accurate series sensors.

- This is especially important for dynamic calibrations (less important for static calibrations)

References

[1] "IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems," in IEEE Std 1588-2019 (Revision of IEEE Std 1588-2008) , vol., no., pp.1-499, 16 June 2020, doi: 10.1109/IEEESTD.2020.9120376

Questions

QUESTIONS?

casonk@umich.edu