TECH NOTE :: IEEE1588:2008 Precision Time Protocol in Data Acquisition and Testing

Date: July 29, 2016
Author: Christof Salcher, Product Manager Test & Measurement, HBM
Status: public

Intro

This TECH NOTE explains the highly accurate time sync mechanism **Precision Time Protocol** or short **PTP,** its advantages and its use in **Test & Measurement** applications.

The following questions shall be answered:

- What is "time synchronous" and what is "real-time"?
- What applications are typical?
- How does PTP work, what are the benefits and what is necessary?

What does "time synchronous" mean?

We use clocks to be in time when meeting with other persons. In sport, some fragments of a second can be decisive to win or lose a race. In stock exchange some micro seconds in time difference on buying or selling might decide over some thousands of dollars (high-speed trading). The necessary accuracy of clocks of course depends on the application.

In Test & Measurement applications highly accurate time stamped signal inputs of any type observing the same physical process play an important role in qualifying and analysing measurement data online or in a post process mode in the same fragment of time.

When it comes to timing in general we also need to differentiate between **absolute and relative time**.

Absolute time is needed when measurement data needs to be mapped to a certain real life event, impact or activity or when two or more DAQ systems are not in the same network. An example where absolute time might be relevant is when the load influence of a train crossing a bridge shall be monitored and identification of the train makes sense for any further activities (overload warning). The absolute time is explicitly available when it is represented by a **clock**.

An **absolute time source** can be:

- PTPv2 Grandmaster Clocks
  - for lab use from company Meinberg is GPS based
  - for mobile use from company OMICRON is GPS based
- GPS sensors
  - Directly using the Precise Positioning Service (PPS) from the GPS sensor
  - taking over the protocol based absolute time when starting the DAQ job
- Network Time Protocol (NTP) master
  - publicly available in an internet access by for example NIST (http://tf.nist.gov/tf-cgi/servers.cgi)
  - distributed locally and based on GPS from companies Hopf or Meinberg
  - running on a PC and taking the absolute time from the operating system
- Terrestrial low frequency transmitted radio signal
  - Example here is DCF-77 (atomic clock) run by the PTB in Germany
- Simple Network Time Protocol (SNTP) time servers

In most of the Test & Measurement applications or processes it is fully OK to work on a **relative system time**. Especially when a test is reproducible and the relative time of the signals to each other matters the most. If needed at all absolute time might be part of the META data or integrated the file name itself, for example 2015-08-03_RLDA-test_Viper_No12.

Sometimes time accuracy is mixed up with reaction, time latency or **real-time**. Real-time ability can be reached by using for example real-time busses like EtherCAT, ProfiNET, EtherNET/IP or many other proprietary field busses. Always possible is leading inputs directly to dedicated analog output voltages with constant time latency.

What does “real-time” mean?
Real-time means deterministic behaviour – a “decision” or “reaction” needs to be done with a time given mainly used in control or automation tasks (sensor -> control algorithm -> reaction / actuator).

What is time latency?
Time latency is an aspect which has to be taken into account when it comes to design control algorithms or reaction needs to be executed in a given maximum time. For real-time control normally a fix and very low time latency from sensor to controller is needed. For non-deterministic protocols like Ethernet TCP/IP, CANbus or any PC based activity time latency is variable. Time latency also plays a role when data is streamed to a real-time controller just for monitoring purposes in case the time stamp send with the data value is not or cannot be considered.

What needs to be considered when solving real-time and high-speed analysis with one DAQ solution?
In Test & Measurement we deal with many different types of applications. One part focusses on time synchronous **Measurement** and **Data Analysis**. Examples for this are structural load test, vehicle testing and Road Load Data Acquisition (RLDA) or bridge monitoring. The other part deals with **Dynamic actuator based Lab Testing** focussing on simulation, stimulation, control or in-the-loop and data acquisition for the purpose of analysis does not play the major role. The following list gives you some aspects to be considered:

- Some applications use datarates up to 100 kS/s per channel and more for the purpose of data analysis. Real-time reaction is not the main criterion and also not possible with regular real-time busses. It would add complexity and would lower the degree of freedom.
- Highly accurate instruments work with 24 Bit Sigma Delta AD converters and filters causing phase shift and time delay. Real-time applications are focussing on determinism in the milliseconds range and need short reaction times. Modern DAQ solutions like QuantumX from HBM allow splitting sensor inputs to two signals for different purposes (1st real-time, 2nd high-speed time stamped and filtered data).
- Both disciplines “analysis” and “control” have a different character, workflow, purpose and often responsibility. Combining both aspects in one single solution can cause conflicts, i.e. driving a Dyno Test Stand or electric load and in parallel acquiring high-speed dynamic data from the system under test, the drivetrain. Therefore splitting responsibility and workflow makes sense – Control and Analysis.
- At the moment there is no world-wide established high performance real-time bus available. All solutions like ProfiNET RT, EtherCAT and many others are driven by global industrial players supporting their own technology in their market and mostly in specific applications. The **Time-Sensitive Networking** (TSN) Task Group is addressing an IEEE standardized real-time Ethernet bus, part of IEEE802.1AS to offer a global standard. DAQ solutions are open to link into many different real-time busses by using gateways.
- Real-time needs a master application running in real-time. Stopping this master for some reconfigurations is not an option.

The Precision Time Protocol - IEEE1588:2008 or PTPv2

Some Background
Many different timing mechanisms have been established in the last decades synchronizing devices to each other. Beginning with the classic “external clock” input, a digital signal which starts the sample and hold based analog to digital conversion. One of the first standardized Ethernet based time sync protocols is the “Networked Time Protocol” or short NTP, started in 1982 with a last major update in 1994 with NTP version 4 using a local or public timing master source.

IRIG B time codes are another option synchronizing distributed systems to each other. In this case satellite based receivers are often used as source for precision timing transferring the time information as direct analog or digitally coded signals. Work on these standards goes back to 1956.

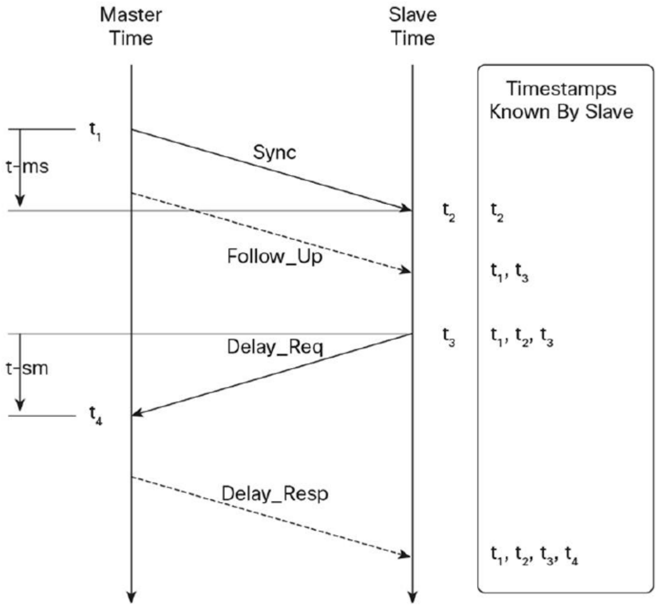
FireWire goes back to 1985 and has been standardized as IEEE1394**b** in 2002 offering an ease-of-use automatic time sync mechanism. All QuantumX modules offer two FireWire ports.

With the demand using the established Ethernet infrastructure in network communication between distributed systems a new approach was needed getting higher flexibility, at lower cost and also ease-of-use.

The only world-wide established standard in networking communication is Ethernet. From IT infrastructure in industry or private households – no doubt, Ethernet is the de facto world-wide standard when it comes to machine-2-machine or human-2-machine communication. Mobile devices like smart phones or even vehicles can be linked in via mobile telecom networks like LTE, EDGE or GSM.

The **Precision Time Protocol (PTPv2)** is based on Ethernet. Compared to NTP, PTPv2 is embedded in the physical layer and thus we talk about true hardware based time stamping for precise time synchronization of all participants in an Ethernet network. PTP is internationally standardized in IEEE1588, revised in 2008 with version 2 and is a network based time synchronization communication protocol which can be used to synchronize clocks of different device types providing time accuracy in the sub-microsecond range. All this speaks for demanding Test & Measurement applications with a new dimension in flexibility.

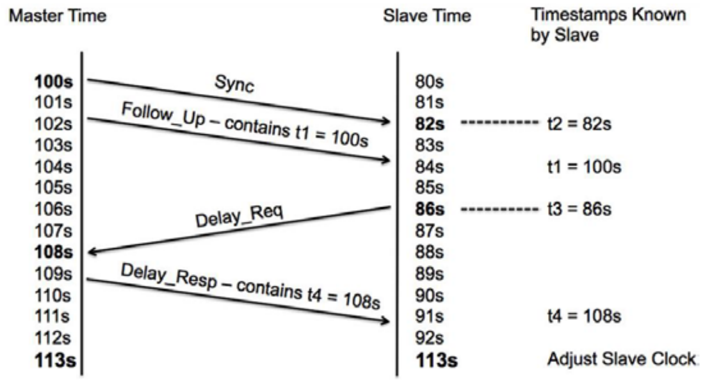
How does PTPv2 work?
PTPv2 is a relative time sync mechanism. One participant is selected to work as master clock which delivers time sync messages to all slaves. The sync process starts with a time sync telegram to the network. All participants (slaves) calculate the time difference (delay) between their local time and the given master clock and adapt step by step to a time difference less than 2 µs. For example, assume that the PTP source sends a PTP message advertising a time of 1:00:00 pm. The problem is that this message will take time to reach its destination. If the PTP packet took 1 second to reach its source, it would be 1:00:01 pm, when the source receives a PTP packet indicating 1:00:00 pm. So the network latency needs to be compensated, which is achieved through a series of messages exchanged between master and slave clocks, as shown in the next Figure.



1. The master clock sends the Sync message. The time that the Sync message leaves the master is time stamped as t1, which can be embedded in the Sync message itself (one-step operation) or sent in the Follow_Up message (two-step operation).
2. The slave receives the Sync message; t2 is the time that the slave receives the Sync message.
3. The slave sends the Delay_Req message, which is time stamped as t3 when it leaves the slave and time stamped as t4 when the master receives it.

4. The master responds with a Delay_Resp message that contains time-stamp t4.

Example: the master time initially is 100 seconds and the slave time is 80 seconds. This is how time would be adjusted in the slave.



All PTP participants need to be PTP capable including Ethernet switches but without the data sink (PC with DAQ software). The Clock in a DAQ module is named an **Ordinary Clock**. A Clock in an Ethernet Switch is named **Boundary Clock**. The Master is selected automatically if no **Grandmaster Clock** delivers absolute time information. This mechanism is called Best Master Clock Algorithm (BMC).

Some DAQ topologies are line or ring structured with many switches and thus Boundary Clocks with its own time control loops. As an alternative the so called **Transparent Clock** (TC) allows an **End-to-End** (E2E) sync control and follow-up messages. The introduction of transparent clocks allows for a far simpler solution to correcting for variable switch latency. The basic idea of the TC is to measure the time a PTP event message has spent in the switch (the so-called residence time). The residence time is reported to the receiver by the message itself or by a subsequent Follow_up or Delay_Resp message. For this purpose a new message field has been added, the so called Correction Field, which is a type of TimeInterval and can be used to accumulate residence time along the path of the message and may also be used for other kind of corrections? The key benefits are:

- No configuration required: Transparent clocks do not have to calculate and do not have to be considered in the calculation of the BMC algorithm, so they do not necessarily have to send or receive management messages.
- Quick reconfiguration of the network in case of faults.
- Faster setup times: in initialisation phase and after change in topology, transparent clocks do not have to re-synchronize to a master clock before they can be considered part of a valid synchronised path.
- Perfect for small configuration.

In contrast, **Transparent Clock** using **Peer-to-Peer** (P2P) scale well with the number of devices attached to the subnet and can recover rapidly to changes in network topology. So this mechanism offers a high scalability and its best suited for deeply cascaded topologies (large scale systems using many switches in daisy chain).

What are the advantages of PTPv2?

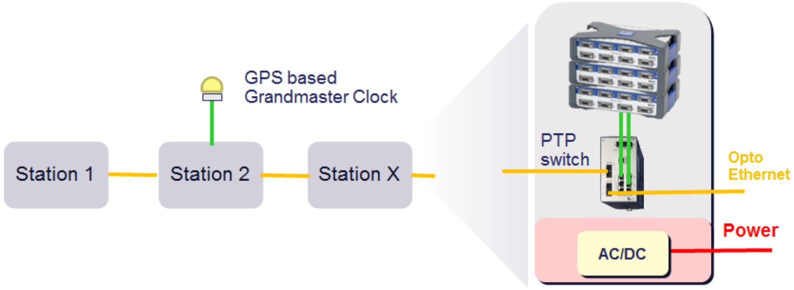
- It allows to time sync’ different device types from different vendors by a standardized protocol
- It allows large distances between DAQ modules
- It allows synchronizing different HBM product lines to each other. QuantumX, SomatXR and GENESIS Highspeed offers PTPv2 and enable data acquisition in all kind of fields:
    - Distributed, mobile, outdoor in harsh environment
    - lab or field testing with many hundred channels or up to some mega samples per second and channel
- High time accuracy in the sub-microsecond range between all participants when working with high data rates
- Using Ethernet as standard
    - Electrical: up to 100 m standard Ethernet cable
    - Optical: large distance (some miles) between modules and galvanic isolation
    - No need for extra sync lines

- Simple, administration free setup
  - automatic master selection
  - robust against master failures (smart slaves)
  - robust against topology changes
  - continuous time scale (no "jumping" time stamps, no roll over)
- Absolute timing if necessary
  - A Grandmaster Clock based on GPS can be integrated and serve as absolute time when one or several DAQ systems are not working in the same network but its data needs to be analysed in a quick.

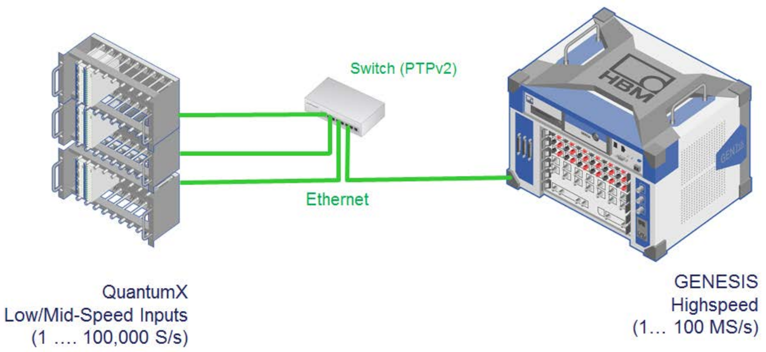
Typical Test & Measurement Applications using PTP Time Sync

Here is a list of some typical applications where PTP plays out all its benefits:

- Widely distributed data acquisition modules used for
  - Mobile testing of large scale ground vehicles (trains, construction): brakes, dynamics, structure, …
  - Lab testing of aircrafts: mechanical (iron bird), structural (durability)
  - Monitoring of large structures: bridges, towers or wind energy plants using GPS based absolute time
- Hybrid data acquisition systems used for
  - Electric or hybrid vehicle dyno test combining the high-speed data acquisition system GENESIS Highspeed acquiring voltage and current with 2 MS/s per channel with mid-speed QuantumX acquiring mechanical and thermal sensor data
  - Lab testing of aircrafts: electrical (copper bird)
  - Lab testing of machinery or generator: electrical, mechanical and thermal
  - Mobile vehicle testing or monitoring using cameras, wheel force transducers or other supplements to analog or digital vehicle bus data
  - In general combining different DAQ systems from different vendors for any purpose
- Mixed system architecture combining classic analog data acquisition systems with cameras or other information sources.



Picture: widely distributed DAQ system topology based on QuantumX



Picture: Hybrid DAQ system GENESIS and QuantumX

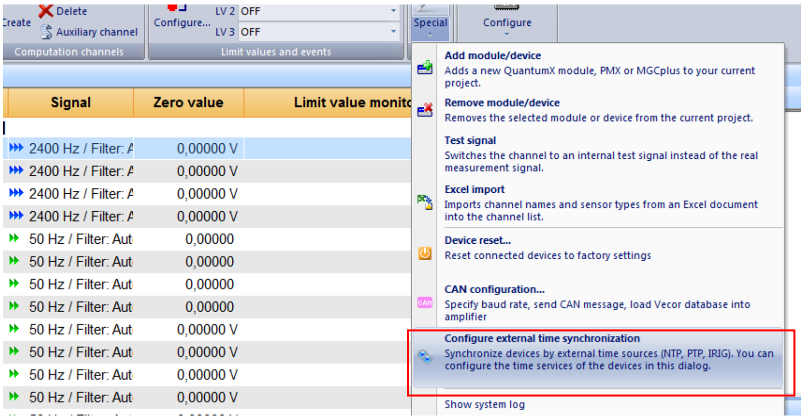
Parameterization of PTP in the DAQ Software

As QuantumX supports different time sync mechanism, parameterization is needed the first time you set up the network. The default or AUTO system clock timing mechanism is FireWire. In addition to that you can select the following timing sources or mechanism:

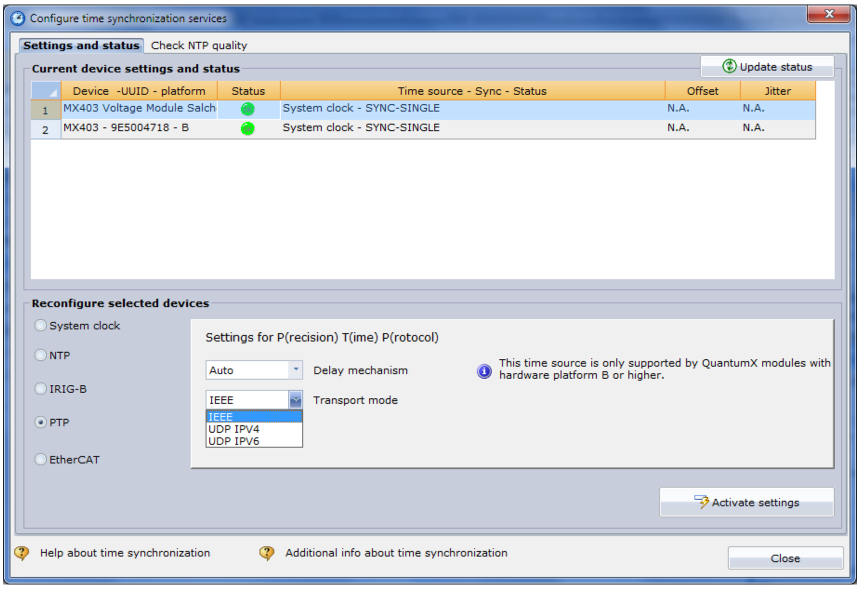
- PTPv2
- NTP
- IRIG-B
- EtherCAT

You can use the software **catmanEasy**, **MX Assistant** or **perception** parameterizing PTPv2.

catmanEasy: start catmanEasy, select the modules, go to Special in menu and select Configure external time synchronization

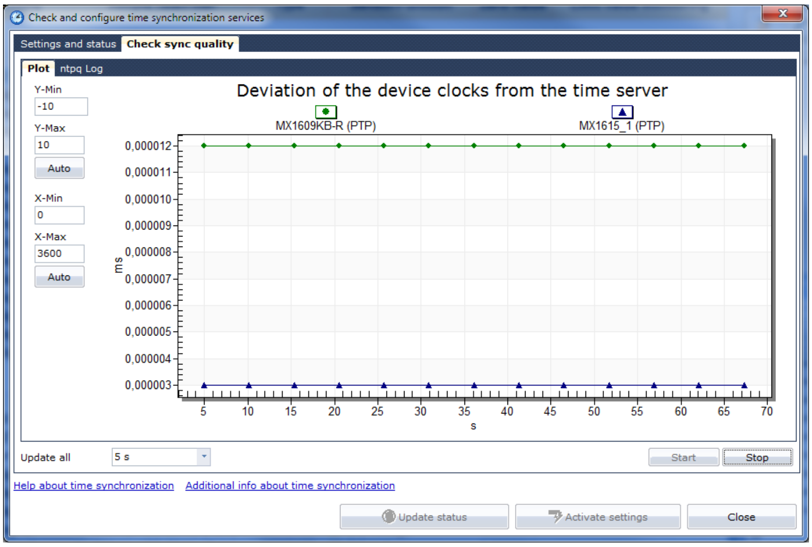


Select PTP and configure Delay mechanism and Transport mode (example E2E and UDP IPv4)



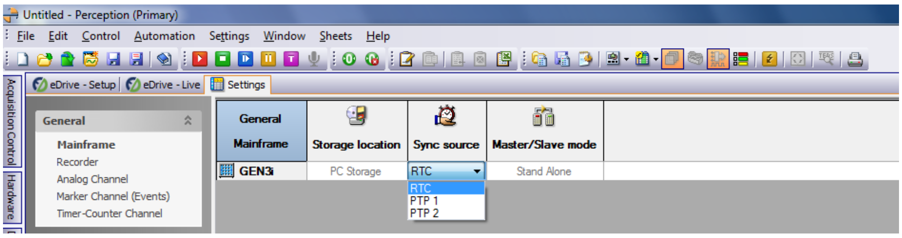
Timing Diagnosis

The software catman is able to analyse the overall time accuracy and deviation of the clocks from the timing master. As PTP timing *deamon* we use ptp4l. For this purpose please open the dialogue “Check sync quality”:



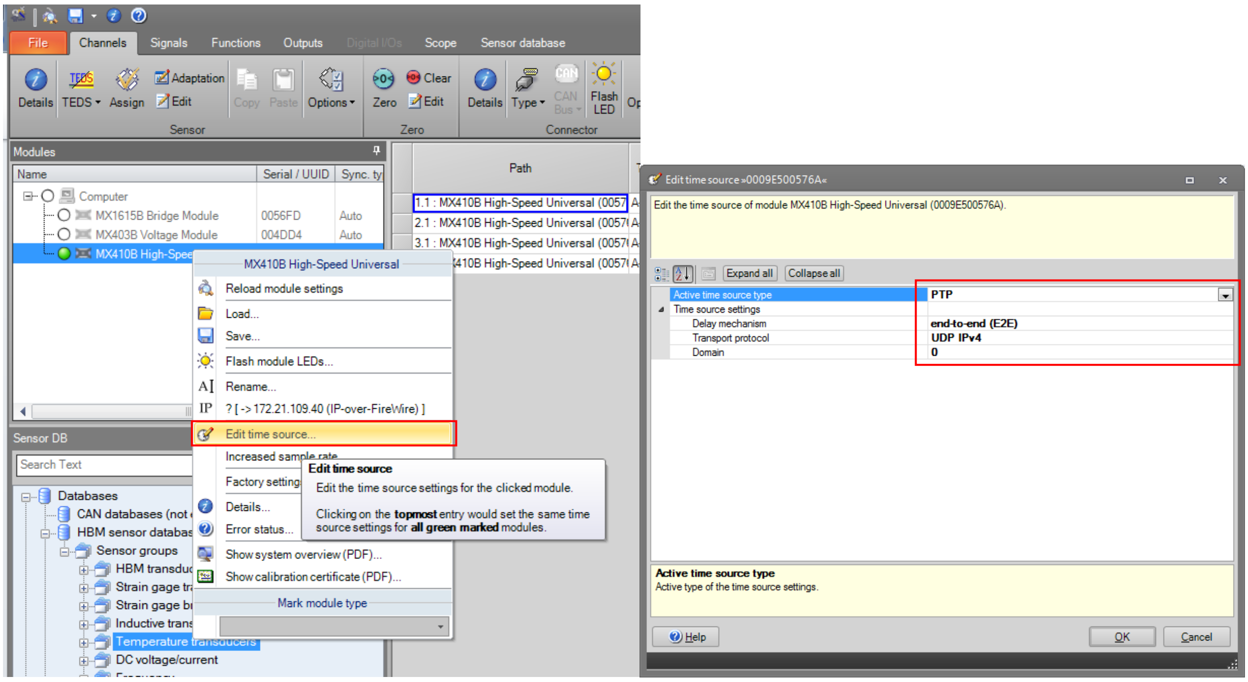
Perception is the GENESIS Highspeed software which allows integration of QuantumX.

Go to Settings and select the Sync Source.



MX Assistant: The MX Assistant is a for free tool for users who work with 3rd party software like Visual Studio .NET programming their own software or graphical programming in LabVIEW, or also CANape, DIAdem and so forth.

Start MX Assistant, right mouse click to module -> Edit time source -> select PTP -> parameterize according to your task.



Frequently Asked Questions

Is PTPv2 downwards compatible to PTPv1?

PTP Version 1 primarily targeted test & measurement and industrial automation. It is a multicast protocol for use on a LAN with performance exceeding the capability of NTP.
PTP Version 2 or IEEE-1588-2008 is an enhancement to version 1. The versions are incompatible. Just to name some of the features of the newer PTPv2 standard:

- Multicast messages
- Two-step clock
- Peer-to-peer (P2P) or end-to-end (E2E) delay mechanism
- Sync Interval: 1, 2, 4, 8, 16, 32, 64 or 128 packets/second
- Delay Request Interval: 1, 2, 4, 8 or 16 seconds
- Support the Transport protocols: IPv4 and the modern IPv6.

What accuracy do I get when using PTPv2?

Time accuracy heavily depends on the type of network and its devices.
We recommend setting up a private LAN where all network participants fully support PTPv2. With this configuration time accuracy can go down to 100 nanoseconds device to device and its channels. Still you need to consider that different datarates and filters can cause a jitter in timing and phase delays.

What is the difference between hardware and software time stamping?

The main difference is in the synchronization accuracy that is achievable. With software based time stamping used for example in NTP you can see slave synchronization accuracies down to 100 microseconds in small networks but typically 1 ms. With hardware time stamping as implemented it is possible to achieve time synchronization accuracies down to 100 nanoseconds. However, in order to get this level of accuracy, the network topology such as switches and slave hardware must support hardware time stamping.

Can I also use a standard switch and if so what is the effect?

Using Non PTP capable switches is risky. Transferring PTP messages then depends on traffic and thus it's critical in the overall timing. In case the switch supports QoS this can be solved by a rule to higher prioritize PTP packages. In general we do not recommend using Switches without PTPv2 support. Worst effect is that PTP packets are lost and timing and thus required data is not any more reliable.

Common terms used in IEEE 1588:2008

Boundary clock: A boundary clock is a clock with more than a single PTP port, with each PTP port providing access to a separate PTP communication path. Boundary clocks are used to eliminate fluctuations produced by routers and similar network elements.

Clock: A device providing a measurement of the passage of time since a defined epoch. There are two types of clocks in 1588: boundary clocks and ordinary clocks.

Clock timestamp point: 1588 requires the generation of a timestamp on transmission or receipt of all 1588 Sync and Delay_Req messages. The point in the outbound and inbound protocol stacks where this timestamp is generated is called the clock timestamp point.

End-to-End Transparent Clock: Transparent clocks are an alternative to boundary clocks for multiport devices such as switches. The first type, the end-to-end transparent clock, forwards PTP event messages, but modifies the messages for the residence time for the message to propagate from an ingress port to an egress port. Corrections must be made for the propagation of both sync and Delay_Req messages.

External synchronization: It is often desirable to synchronize a single clock to an external source of time, for example to a GPS system to establish a UTC time base. This synchronization is accomplished by means other than those specified by 1588 and is referred to as external synchronization

Grandmaster clock: Within a collection of 1588 clocks one clock, the grandmaster clock, will serve as the primary source of time to which all others are ultimately synchronized.

Master clock: A system of 1588 clocks may be segmented into regions separated by boundary clocks. Within each region there will be a single clock, the master clock, serving as the primary source of time. These master clocks will in turn synchronize to other master clocks and ultimately to the grandmaster clock.

Message timestamp point: 1588 Sync and Delay_Req messages contain a distinguished feature, the message timestamp point, serving as a reference point in these messages. When the message timestamp point passes the clock timestamp point, a timestamp is generated that is used to compute the necessary corrections to the local clock.

Ordinary clock: An ordinary clock is a 1588 clock with a single PTP port.

PTP message: There are five designated messages types defined by 1588: Sync, Delay_Req, Follow-up, Delay_Resp, and Management.

Multicast communication: 1588 requires that PTP messages be communicated via a multicast. In this style of communication any node may post a message and all nodes in the same segment of a subdomain will receive this message. Boundary clocks define the segments within a subdomain.

Synchronized clocks: Two clocks are synchronized to a specified uncertainty if they have the same epoch and measurements of any time interval by both clocks differ by no more than the specified uncertainty. The timestamps generated by two synchronized clocks for the same event will differ by no more than the specified uncertainty.

-- end

Sources

- IEEE1588 spec