High Precision Clock Synchronization according to IEEE 1588 Implementation and Performance Issues

Prof. Hans Weibel
Zurich University of Applied Sciences
Institute of Embedded Systems (InES)
Technikumstrasse 9
CH-8401 Winterthur, Switzerland
Tel +41 52 267 75 52
Fax +41 52 268 75 52
hans.weibel@zhwin.ch

Abstract

A high precision time base is important for distributed systems in measurement and automation applications. The Precision Time Protocol (PTP) specified in IEEE 1588 [1] is able to synchronize distributed clocks with an accuracy of less than one microsecond. It is applicable in multicast capable network technologies such as Ethernet LANs. The mechanism combines high accuracy and fast convergence with low demand on clocks and on network and computing capacity.

The paper outlines the application areas of precise time synchronization. The operation of the mechanism as specified in IEEE 1588 is explained and the factors contributing to precision are analyzed. A real and portable implementation is presented. Its performance is evaluated with respect to synchronization behaviour, accuracy and stability under real conditions.

Keywords: Clock Synchronization Protocol, Time Dissemination, Ethernet

1 Time Synchronization Application Areas

We use watches and clocks to synchronize ourselves with other persons or procedures. The watch should be accurate – but how accurate depends on the application. Anyone wanting to catch a train should keep an eye on the time to within a minute. At sports events, a hundredth of a second can be decisive and drives in a packing machine need synchronization in the microsecond range.

Many technical systems have a time concept, either implicit or explicit. An implicit system time exists when no actual clock is available and the time behavior is determined by processes in the hardware and software. This often suffices in small, closed systems. An implicit system time is implemented e.g. by regular trigger events to every user, which signal the beginning of a time unit and then trigger the appropriate actions.

The system time is explicit when it is represented by a clock. This is usually necessary in complex systems. It separates the communication from the execution.

Not every clock is sufficiently accurate however. You have to check periodically whether the inaccuracy is tolerable and adjust the clock if necessary. As a reference, we use clocks which are more accurate than the ones we want to correct. All these processes are based largely on communication between better and worse clocks. Bad clocks or clocks, which may not deviate too greatly, need to be adjusted more often.

Two effects need to be observed when setting clocks: on the one hand, independent clocks run initially with a time offset. To synchronize them, the less accurate clock is set to the more accurate one (offset correction). It should also be noted that clocks do not always run at the same speed. Therefore constant control of the speed of the less accurate clock is necessary (drift correction).

PTP finds a lot of interest in different kinds of applications.

- Test and measurement: In many test and measurement systems, data is acquired by polling the sensors. Sampling timing heavily depends on application program timing and communication latency. A more flexible approach is to equip sensors with a synchronized clock, allowing a decoupling of communication and execution. An interesting example in this area is LAN eXtensions for Instrumentation (LXI). LXI is an instrumentation platform based on industry standard Ethernet technology designed to provide modularity, flexibility and performance to small- and medium-sized test systems. LXI will implement IEEE 1588, allowing triggering directly over the LAN, simplifying cabling. The LXI consortium [2] is about to develop a standard for LAN-based instrumentation.
- Industrial automation: Synchronization among multiple axes in a distributed system allows for accurate coordination of drives. The control loops of the individual drives have to be executed synchronously. Such multi-axis motion control applications require the jitter between the system clocks to be in the range of microseconds. The presence of such precise time information in terminal devices also enables distributed highly synchronous processes to be realized in instrumentation, and control applications or in all kind of machinery. For this reason high-precision clock synchronization as per the new Precision Time Protocol is considered the basis for many real-time automation protocols, such as ETHERNET Powerlink (EPSG [3]), PROFInet IRT (PNO [4]), CIPsync (ODVA [5]) and EtherCAT [6].
- Power industry: In power plants and substations, many electronic sensors are used to control and protect the equipment. Event time-stamping and data correlation facilitates applications including fault localization, network disturbance analysis, and detailed recording of events (exact sequence of events facilitates diagnosis). Synchronized sampling, event time stamping and other advanced functions require precise synchronization. Traditionally a separate synchronization network is used. Being able to transmit sync and data messages over the same network is a big advantage and saves a lot of cabling.
- Enhancement/replacement of existing time distribution networks: There are for example different standards issued by the Inter Range Instrumentation Group (IRIG), originally targeted for military applications. The most common version is IRIG-B, which encodes day of year, hour, minute, and second data amplitude modulated on an audio sine wave carrier. It is something like the “cable version” of a radio clock. An other common method is to send a high precision Pulse per Second (PPS) signal to every user over separate cabling. Both methods however, require a considerable amount of extra wiring work.
- Telecommunications: In telecommunication networks, system and service quality depends on accurate synchronization. Such networks are traditionally circuit switched and allow the distribution of clock signals over the transmission line. While the networks migrate more and more to packet switching, many traditional circuit switched services continue to exist. The emulation of circuits on a packet network is easy as long as both ends of the circuit have access to a common clock. IEEE 1588 can provide this clock. Current activities in this area include the rollout of Ethernet-based metropolitan area networks [7]. These multi-service networks carry packet data as well as emulated circuits for voice applications. An important telecommunication application is wireless networks. In cellular networks, the handover capability requires precise synchronization of all base transceiver stations. In digital TV broadcast systems (DVB-T),

multiple transmitters operating at the same frequency need to be synchronized (in terms of both time and frequency).
- Aerospace and navigation: In telemetry, radar and sonar sonar systems, synchronized clocks are generally of importance. In our institute, we are currently elaborating a feasibility study of an “undersea positioning system”. It works similar as GPS, but uses buoys instead of satellites and subsonic waves instead of radio frequencies.
- Residential networks: In multi-channel sound systems where the speakers are connected via Ethernet, synchronous play-out of packetized audio is essential.

2 Synchronization Methods compared with PTP

There have been various previous methods of synchronizing clocks distributed over a network: the most common are the Network Time Protocol (NTP [8]) and the simpler Simple Network Time Protocol (SNTP [9]) derived from it. These methods are quite common in LANs or in the Internet and allow accuracy right down to the millisecond range. Another possibility is the use of radio signals of the GPS satellites. However, this requires relatively expensive GPS receivers for every clock and the appropriate antennae on the roof and the necessary cabling. Although this provides a high precision clock, it is often impractical for reasons of cost and effort.

3 The IEEE1588 Standard

This is where the Precision Time Protocol (PTP) described in IEEE 1588 comes in. It was developed with the following goals:
- Accuracy to at least microsecond and preferably nanosecond levels.
- Minimal network, computing and hardware resource requirements so that it can be applied to low-end as well as high-end devices.
- Applicable with minimal or no administration to systems defined by a single subnet or at most a few adjacent subnets of a networked system
- Applicable to common and inexpensive networks including but not limited to Ethernet.
- Applicable to heterogeneous systems where clocks of different capabilities can synchronize to each other in a well-ordered manner.
- Specified by a standard to promote interoperability and adoption by manufacturers.

The idea for PTP was originally born at the end of the 90s in the USA at Agilent Technologies out of the problem of creating clear time relations between measured values picked up at distributed points. The method developed there was submitted as a suggestion for standardization to the IEEE and then developed as the IEEE 1588 standard. In November 2002 the specification passed as a standard under the name “1588™ - IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems” [1].

On May 21, 2004, IEEE 1588 passed ballot to be dual labeled by both the IEEE and the IEC. In IEC, its number will be IEC 61588 [10].

During 2004, some *IEEE 1588 Task Forces* have been active to further refine and develop the technology and its application [11]. In the *Conformance and Interpretation Task Force*, discussions centered on issues such as interpretations of the standard, requirements for conformance, certification procedures or test sets and reference implementation.

On the *2004 Conference on IEEE 1588*, a plug-fest was arranged. Five IEEE1588 implementors participated. All nodes proved protocol interoperability and synchronized correctly. Precision of synchronization was in the range of +/-200ns between some nodes and even +/-50ns and less between the better ones.

4 PTP Operation

4.1 Best Master Clock Selection

The most precise clock in the network synchronizes all other clocks. There are two kinds of roles: master (the one, which synchronizes the others) and slaves (those being synchronized). In principle, any clock can play either the master or the slave role.

The precision of a clock is categorized by the protocol in classes (stratum). The highest class is an atomic clock that has the stratum value 1. The selection of the best clock in the network is performed automatically using the "best master clock algorithm".

4.2 Clock Synchronization

PTP's operating principle is to exchange messages consecutively to determine the offset between master and slave but also the message transit delay through the network (see figure 1).

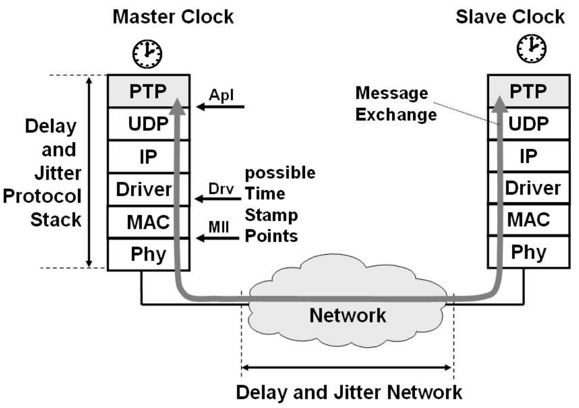

Figure 1: PTP message exchange

For the purpose of offset correction, the master cyclically transmits Sync messages to the slave clock at defined intervals (by default every 2 seconds). A time stamping mechanism determines the exact transmission time t1 as precise as possible and sends it down to the slave on behalf of a second message, the Follow_up message (see figure 2).

The slave clock measures the exact reception time t2 of the Sync message.

On reception of the Sync and the corresponding Follow_up message, the slave clock calculates the offset correction in relation to the master clock. The deviation between master and slave is the transit delay the Sync message has experienced when traveling through the network.

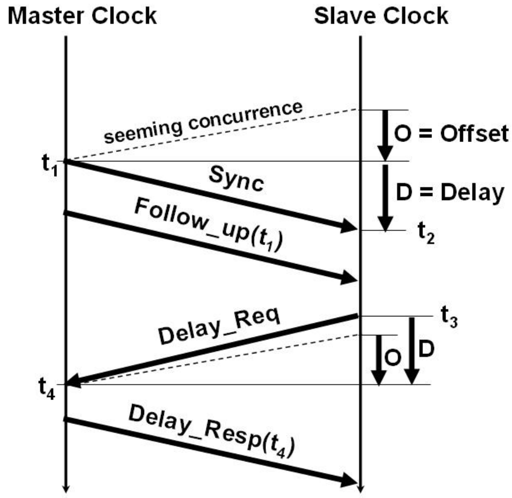

Figure 2: Offset and Delay Measurement

Furthermore, consecutive Sync measurements allow the slave's frequency drift to be compensated.

Another message exchange is required to measure the delay between slave and master. For this purpose, the slave clock sends a so-called Delay_Req packet to the master. The exact transmission and reception time t3 and t4 of this message are measured. The Delay_Resp message carries the measured value t4 to the slave, where now delay and offset are calculated out of the four time stamps t1, t2, t3, t4:

Delay + Offset = t2-t1
Delay - Offset = t4-t3
Delay = ((t2-t1) + (t4-t3)) / 2
Offset = ((t2-t1) - (t4-t3)) / 2

Note that the transit delay is assumed to be the same for both directions!

The delay measurement is performed irregularly and at larger time intervals (random value between 4 and 60 seconds by default). In this way, the master is not too heavily loaded if it has to synchronize a large number of slaves.

4.3 PTP Communication

PTP is based on IP multicast communication and is not restricted to Ethernet, but can be used on any network technology that supports multicasting. The scope of this paper is restricted to Ethernet.

PTP scales for a large number of PTP nodes because a master can serve many slaves with a single pair of Sync and Follow_up messages.

Multicast communication offers also the advantage of simplicity. IP address administration does not need to be implemented on the PTP nodes. For this reason, Delay_Req and Delay_Resp as well as the management messages, which are in fact point-to-point messages, use also multicast addressing. All other nodes but the wanted destination has to filter out these messages.

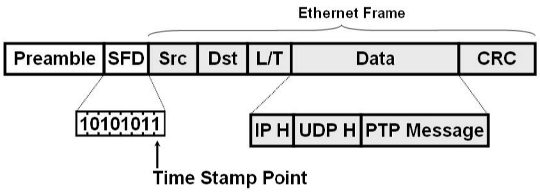

Figure 3: Ethernet encapsulated PTP messages

PTP is an application layer protocol, on top of UDP/IP/Ethernet (see figure 3). The use of port number 319, the so-called event port, is reserved for messages that have to be time stamped (i.e. Sync and Delay_Req). This facilitates the message detection logic. All other messages use port number 320.

4.4 The PTP Network

The precision of the protocol also depends on the latency jitter of the underlying network topology. Point-to-point links between master and slave provide the highest precision. Hubs impose very little network jitter (around 300 to 400 ns).

Switches are store and forward devices, which produce a load dependent delay. Under very low or no network load, switches have a very low processing time which is typically 2 to 10µs plus packet reception time, and have low latency jitter of e.g. about 0.4 µs.

However, a single queued maximum length packet imposes a delay for the following packet of about 122µs, and under high load conditions, more than one packet can be in the queue.

An important issue for the precision of the protocol is that latency is completely symmetric for both directions: from the master to the slave and vice versa. This can nearly never be guaranteed under higher network loads.

Prioritization of packets according to IEEE802.1p does not really solve the problem, because at least one long packet can be in front of a synchronization packet and so will impose up to 122µs to the jitter of transmission.

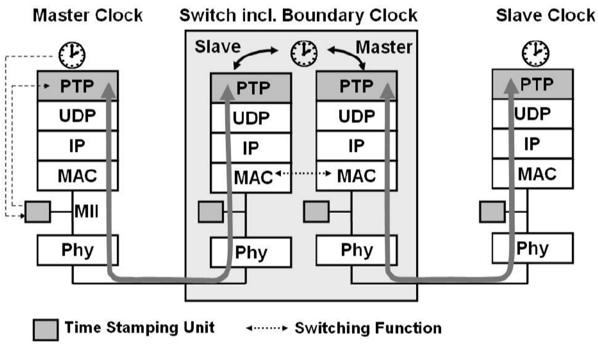

Figure 4: Boundary Clock

The solution for all these problems is the usage of so-called Boundary Clocks in switches (in contrast to the Ordinary Clock in end systems). A switch acting as a boundary clock is being synchronized from an attached master clock, and then the switch itself is acting as a master clock to all attached slaves. With this approach, all internal latencies and jitter in the switch are compensated and do not affect synchronization accuracy.

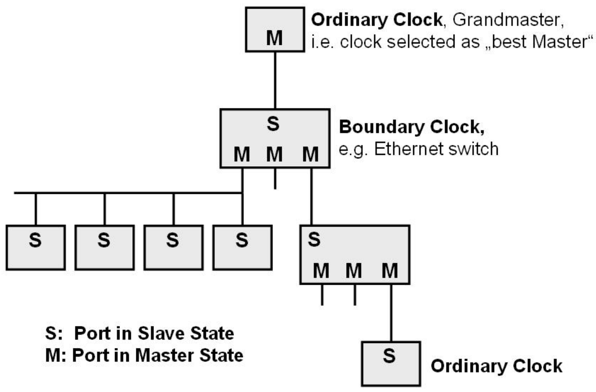

Figure 5: Network topology and clock types

5 A PTP Implementation

5.1 Time Stamping Methods

Synchronization accuracy directly depends on time stamp accuracy. As depicted in figure 1 there are different options to take time stamps:

- The most accurate method is to detect PTP frames with hardware assistance. Ingress and egress frames pass the Media Independent Interface (MII), where frames can easily be captured and decoded. Below the MII, data is 4B5B coded, scrambled, and therefore not directly interpretable. The accuracy of this method is limited by the Phy chip timing characteristics.

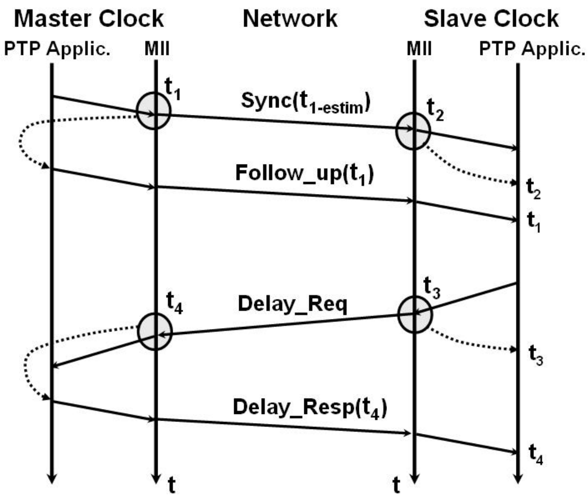

Figure 6: Time stamp transfer (circles indicate time stamping)

- Without any hardware assistance, the next best place for time stamping is the network driver. Egress frames are stamped at the very latest moment before the frame is handed over to the MAC controller. Ingress frames are stamped at the entry point of the network interface interrupt service routine. The accuracy of this method is limited by the operating platform timing characteristics (e.g. interrupt latency, CPU performance) and load dependent.
- Stamping on the application layer is best located at the socket interface. It does not require any modification to system software. The influence of protocol stack and load allows only moderate accuracy.

5.2 Hardware

In collaboration with Hirschmann Electronics, ZHW has developed an IEEE 1588 Evaluation Kit for Ordinary Clocks (EvOC). It consists of an embedded PC with a 300 MHz Geode CPU and the network interface on the main board. A special timing analyzer board (see figure 3) is connected to the PC/104 bus and includes a time stamping unit (TSU) and an adjustable clock. The clock can be monitored and compared with other clocks by the one pulse per second (1PPS) output.

The timing analyzer board has been developed for general-purpose Ethernet timing analysis. Its heart is a FPGA able to observe Ethernet frame streams and to take a time stamp if configurable filter criteria are met. In addition to the time stamp, a configurable selection of the frame's data is captured in order to correlate the time stamps and the respective packets on the application layer.

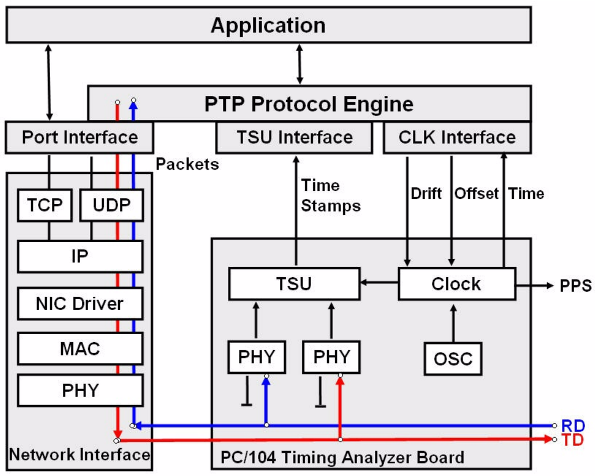

Figure 7: The IEEE 1588 evaluation kit

Because transmit and receive wire pairs are eavesdropped and decoded by two independent and identical Ethernet PHYs, both paths have the same hardware configuration. Unfortunately, path symmetry is not guaranteed, even if identical PHYs are used (see [12]).

5.3 Software

The software consists of Gentoo Linux (vanilla 2.4.27 kernel) and a PTP protocol engine from Hirschmann Electronics.

The PTP protocol engine sends and receives PTP messages over the OS's unmodified network interface. Using registers on the clock and time stamping board, time stamps and the current time can be accessed. The clock allows offset correction and drift compensation.

PTP messages can be time stamped by both the TSU and by the NIC driver. Since both time stamps use the same clock, a comparison of the time stamping methods is possible. The 1PPS output allows the synchronization to be checked.

6 Performance under real Conditions

6.1 Performance Criteria

The precision of the delay and offset calculations depend on the precision of the time stamps. They should reflect the send and receive time as precise as possible.

The slave's offset and delay calculation is based on the difference of time stamps taken at two different places (i.e. master and slave node). Therefore, the two clocks should use the same scale, i.e. the same tic interval. This is achieved by drift compensation: the slave clock's rate is accelerated or slowed down. A slightly different tic interval will degrade the result.

It is assumed that the message transit delay is the same for both directions. At a first glance, this is the case in an Ethernet link. However, going into the details, there are a few non-idealities: Cables used for Ethernet have a minor asymmetry by design, to reduce far end cross talk (FEXT). Ethernet transceivers have asymmetric transmit and receive paths. If their timing characteristics are clearly specified within a small range, the asymmetry can be taken into account by calculation as inbound and outbound latency correction constants.

On the long run, conditions may change due to reconfiguration (leading to a very different delay) or environmental conditions (temperature). How fast the clocks can react depends on the frequency of sync and delay measurement and the dynamic behaviour of the servos controlling the slave clock.

To sum it up, performance depends on:

- the communication channel's symmetry (i.e. same delay in both directions and constant over a longer period of time)
- drift compensated clocks (i.e. adjusted time base in master and slave clocks)
- time stamp accuracy
- time stamp resolution
- sync interval
- clock stability
- clock control loop characteristics

6.2 Measurement Results

The setup consists of a master and a slave connected via a 100 Base-TX hub. The measurement conditions are:

- Sync interval is 2s
- The two clocks used in the tests are equipped with cheap quartzes as are used for Ethernet interfaces (50 MHz, ±50 ppm, they drifted away from each other approx. 9 µs/s when running free)
- Temperature: constant (room temperature)

Figure 8 shows the offset of the slave clock referenced to the master clock during start-up as determined by comparing the PPS signals of the two nodes.

During the Best Master Clock Selection, both clocks are running free. Their time difference is much more than the diagrams's scaling allows to display. The first Sync messages correct the slave's time. Because the two clocks do not oscillate exactly on the same frequencies, the slave drifts away again. After two sync intervals, the drift is compensated.

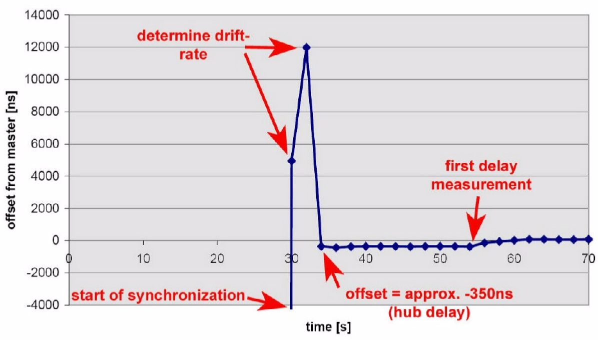

Figure 8: Synchronization behaviour on start-up

A delay calculation has not happened yet (Delay_Req messages are transferred at a much lower rate than Sync messages). During this phase, the one way delay is assumed to be zero. Up until the time a delay calculation is made, the slave's offset from the master is due to the „network" delay. This „network" delay is constant at approximately 350 ns and represents the one way delay caused mainly by the hub. The „network" delay will be fully compensated as soon as the first delay calculation is made.

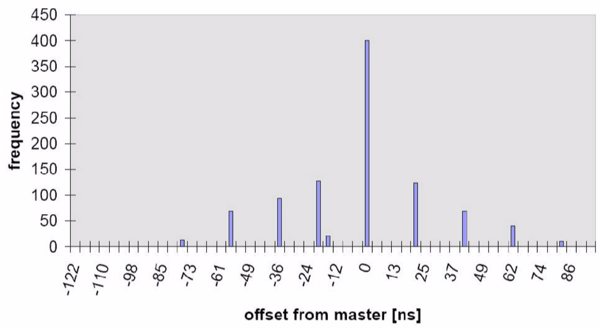

Figure 9: Synchronization accuracy after start-up

The histogram in figure 9 shows the difference between the 1PPS signals of master and slave as measured under stable conditions. The significant variation is within ±80 ns. The values clearly show the 20 ns quantization interval of the measurement equipment.

The observed fluctuation can easily be explained as the combination of the hub jitter, the transceiver's jitter and the time stamp's resolution.

More results and details can be found in [13] and [14].

7 Proposed Technical Extensions

The *IEEE 1588 Task Force on Technical Extensions* has actively discussed possible enhancements and extensions to the 2002 version of the standard. The objective is to meet new requirements and allow for new application areas. Backward compatibility to existing implementations is imperative.

At the *2004 Conference on IEEE 1588*, a proposed Project Authorization Request (PAR) has been discussed. A PAR is the authorization of work by IEEE and required for a new project or substantial enhancement of an existing standard. It was decided that a PAR should be opened.

The new version shall include issues such as:

- Accuracy: Synchronization accuracy depends on many factors. Link symmetry and time stamp accuracy are most important. A control loop allows the slave's drift (mainly caused by ambient temperature change) to be compensated. The stability of the oscillators and the dynamics of the control loop affect the fluctuations as well. A challenging objective is to meet nano and sub-nanosecond requirements. High quality oscillators, shorter sync intervals (i.e. increased message rate), extended time stamp resolution and highly symmetrical transmission links are important ingredients.
- Fault tolerance: A master clock may fail to deliver the specified performance (deviation or total failure) and communication may fail as well. For robust performance a redundant master has to be put into operation without loss of synchronization. Reconfiguration mechanisms supposedly change the topology and the related message delays.
- SNMP management: The 2002 version of the standard specifies some messages for system management purposes. For Ethernet implementations SNMP is the appropriate choice.
- Variable Ethernet headers: Time stamping units have to take into account that PTP messages may occur at different offsets and may have other values in certain environments. Tagged Ethernet frames or future IPv6 packets have to be considered.
- PTP bridges: In daisy chain and ring topologies with many Boundary Clocks in sequence, the behavior of concatenated control loops could lead to significant loss of precision. Therefore, two PTP bridge concepts have been proposed which have both the same major goal: applying a single end-to-end control loop between slave and its synchronizing master in order to achieve better synchronization performance.
- Mapping PTP to other network technologies: While PTP can be applied in any multicast-capable network, the 2002 version of the standard defines an Ethernet implementation only (protocol stack is PTP, UDP, IP, and Ethernet). Mapping PTP to other network technologies, some details such as message on-the-wire format, PTP message format, addressing, and time stamp points have to be speci-

fied. Candidates are wireless LANs, layer 2 Ethernet multicast (without UDP/IP), packet-switched telecom networks, DeviceNet and others.

- Security: In open network environments, the sync mechanism has to be protected from malicious PTP clocks. By injecting PTP messages, denial of service attacks would be easy to achieve.

8 Conclusion

The Precision Time Protocol standardized in IEEE1588 reaches synchronization accuracy within the sub-microsecond range and has further potential for higher precision. It is suited for applications which need a time synchronization of distributed clocks of highest accuracy in a limited network domain.

Many manufacturers have already begun the development of appropriate components and have already started to evaluate their first prototypes.

References

[1] IEEE Std 1588™-2002: "IEEE Standard for a precision Clock Synchronization Protocol for Networked Measurement and Control Systems"

[2] http://www.lxistandard.org/

[3] http://www.ethernet-powerlink.com/

[4] http://www.profibus.com/

[5] http://www.odva.org/

[6] http://www.ethercat.com/

[7] http://www.metroethernetforum.org/

[8] RFC 1205: "Network Time Protocol (Version 3) Specification, Implementation and Analysis"

[9] RFC 2030: "Simple Network Time Protocol (SNTP) Version 4 for IPv4, IPv6 and OSI"

[10] IEC 61588 Ed.1: "Precision Clock - Synchronization Protocol for Networked Measurement and Control Systems"

[11] http://ieee1588.nist.gov/

[12] Thomas Müller, Alexander Ockert, Hans Weibel: "PHYs and Symmetrical Propagation Delay", 2004 Conference on IEEE 1588, Sept. 27-29, Gaithersburg

[13] Hans Weibel, Dominic Béchaz: "IEEE 1588 Implementation and Performance of Time Stamping Techniques", 2004 Conference on IEEE 1588, Sept. 27-29, Gaithersburg

[14] http://ines.zhwin.ch/ieee1588/