White Paper Introduction to IEEE 1588 & Transparent Clocks

Caleb Gordon
2009

Abstract

In measurement and control systems there is often a need to synchronise distributed clocks. Traditionally, synchronisation has been achieved using a dedicated medium to convey time information, typically using the IRIG-B serial protocol. The Precision Time Protocol (standardised by IEEE 1588) has been designed as an improvement to current methods of synchronization within a distributed network of devices. PTP is a message based protocol that can be implemented across packet based networks including, but not limited to, Ethernet. Standard Ethernet switches introduce a variable delay to packets that inhibits path delay measurements. Transparent clocks (enhanced Ethernet switches) have been introduced to measure and adjust for packet delay, thus removing the negative effects that these variations cause.

TEKRON
www.tekroninternational.com

About Tekron

Tekron International is a leading developer of exceedingly accurate GPS clocks and time synchronization solutions for use in industrial applications.

Tekron GPS clocks are simple to install and use and are extremely rugged, attributes that are a prerequisite in the often extreme environments in which the clocks are installed.

Tekron GPS clocks have been installed in thousands of power stations & substations across the globe, where they prove invaluable in assisting power companies to operate efficiently, minimizing downtime and increasing the accuracy of control decisions.

With a Tekron GPS clock you can be confident that you can set it up and walk away.

Glossary

BMC Best Master Clock algorithm

IEEE Institute of Electrical and Electronics Engineers

IPv4 Internet Protocol Version 4 (also IPv6)

IRIG-B Inter-Range Instrumentation Group time code type B

ISCPS International Symposium on Precision Clock Synchronisation

PTP Precision Time Protocol (IEEE 1588)

SCADA Supervisory Control And Data Acquisition

Introduction

PTP

The ”Precision Clock Synchronization Protocol for Networked Measurement and Control Systems” (PTP/IEEE 1588) [3] is designed to synchronise clocks across packet based networks. PTP allows for synchronisation of distributed clocks to sub-microsecond accuracy with devices that may have differing precision, resolution, and stability.

The IEEE 1588 protocol was designed for low cost implementation in, but not limited to, Ethernet networks, with plug and play functionality for ease of installation. Synchronisation can be achieved with a minimum use of network resources and can be implemented in systems with minimal computing resources. Although PTP can be implemented over any packet based network, the major focus so far has been the development of PTP over UDP/ IPv4 (User Datagram Protocol over Internet Protocol version 4).

PTP has been designed as an improvement on current time synchronization technologies such as the network time protocol (NTP) and IRIG-B (a high-precision serial protocol). NTP allows for synchronisation of network distributed clocks to a precision in the order of hundreds of microseconds, which for many applications such as personal computer use is a sufficient level of accuracy.

The serial IRIG-B protocol is widely used when a higher level of precision (sub-microsecond) is required. IRIG-B must use a dedicated medium and typically delivers time derived from a high accuracy time source such as GPS. This can cause difficulties when a GPS signal cannot be accessed or the use of a dedicated medium proves to be costly.

Industrial field bus technology has recently moved towards Ethernet as a communication medium due to its prevalence, ease of installation and low cost. In many installations, Ethernet wiring will already be present, therefore simplifying and reducing the cost of setup. The ability of IEEE 1588 to use only a small component of total Ethernet network traffic makes it an ideal choice for the distribution of time among nodes of an industrial control environment.

Version 1 of the standard [1], published in 2002 is already being adopted by industrial automation and test and measurement communities. A full draft of version 2 of the standard has been approved and was published in May of 2008 [3].

Ethernet Switching

Ethernet switches enable a fully available, full-duplex communication path between devices connected in a network. Switches use address information contained within data packets to determine their correct destination and forward them to the appropriately addressed destinations. If multiple messages are due to exit a switch port at the same moment, the switch uses a buffer so that packets are not lost. In the event of the buffers becoming full, the switch will send pause frames to packet senders to delay transmission.

Transparent Clocks

The operation of PTP relies on a measurement of the communication path delay between the time source, referred to as a master, and the receiver, referred to as a slave. This process involves a message transaction between the master and slave where the precise moments of transmit and receive are measured — preferably at the hardware level. Messages containing current time information are adjusted to account for their path delay, therefore providing a more accurate representation of the time information conveyed.

The path delay measurement process of PTP involves the precision timing of two messages — a sync message and a delay request. The average path delay of the two messages gives the one-way delay. This however, assumes that the communication path is completely symmetric. This assumption does not hold in a switched network however, largely due to the buffering process within Ethernet switches.

PTP provides for transparent clocks to measure and account for this delay in a time-interval field within timing packets, thus making the switches temporally transparent to master and slave nodes. Transparent clocks must perform this operation very accurately and at the communication speed without introducing more delays. The end-to-end transparent clock forwards all messages just as a normal switch.

PTP Overview

PTP Operation

IEEE 1588 standardises the Precision Time Protocol (PTP). It defines the descriptors that characterise a clock, the states of a clock and the allowed state transitions. It defines network messages, fields and semantics, the datasets maintained by each clock and the actions and timing for all IEEE 1588 network and internal events. It also describes a suite of messages used for monitoring the system, specifications for an Ethernet-based implementation and conformance requirements and gives some implementation suggestions.

Message-Based Synchronisation

PTP is based upon the transfer of network datagrams to determine system properties and to convey time information. A delay measurement principle is used to determine path delay, which is then accounted for in the adjustment of local clocks. At start up, a master/slave hierarchy is created using what is called the Best Master Clock (BMC) algorithm to determine which clock has the best source of time. The BMC algorithm is then run continuously to quickly adjust for changes in network configuration. Synchronisation is achieved using a series of message transactions between master and slaves. There are five message types - Sync, Delay Request, Follow Up, Delay Response and Management - which are used for all aspects of the protocol. A sequence of message transactions takes place to synchronise a pair of clocks as shown in Figure 2.1.

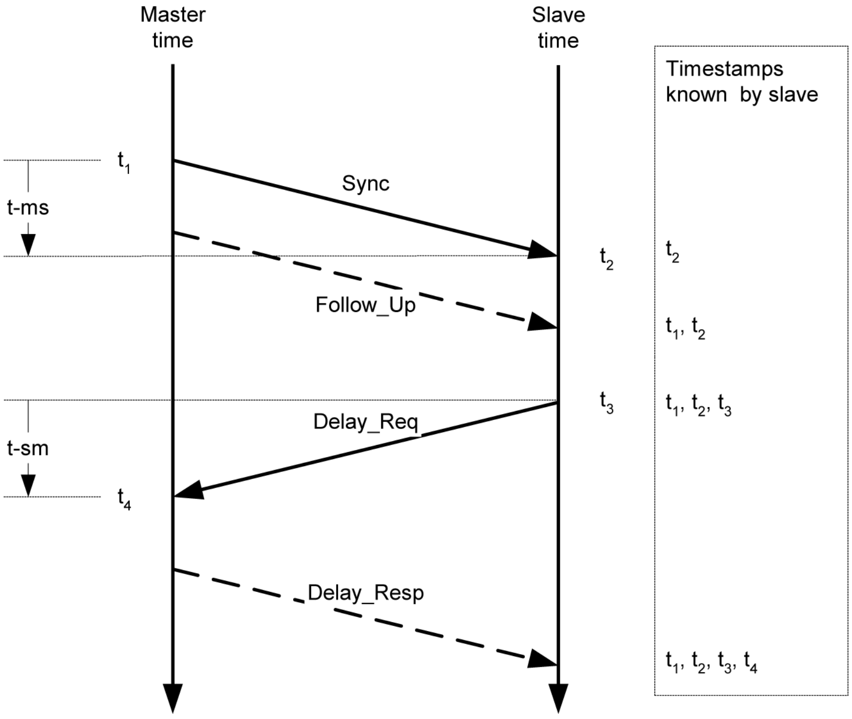

Figure 2.1: Master-Slave offset measurement

The message exchange process is as follows:

1. The master sends a Sync message to the slave and notes the time, $t_{1,}$ at which it was sent.
2. The slave receives the Sync message and notes the time of reception, t_2.
3. The master conveys to the slave the timestamp t_1 by either
    a. Embedding the timestamp t_1 in the Sync message (one-step). This requires some sort of hardware processing for highest accuracy and precision, or
    b. Embedding the timestamp t_1 in a Follow_Up message (two-step).
4. The slave sends a Delay_Req message to the master and notes the time, t_3, at which it was sent.
5. The master receives the Delay_Req message and notes the time of reception, t_4.
6. The master conveys to the slave the timestamp t_4 by embedding it in a Delay_Resp message.

After this message exchange the slave will have four timestamps from which can be determined both the slave offset (time offset by which the slave clock leads or lags the master) and the network delay (the time taken for packets to traverse the network link between the two nodes).

The link delay can be calculated as follows:

$$MasterSlave_{delay} = t_{ms} = t_2 - t_1$$

$$SlaveMaster_{delay} = t_{sm} = t_4 - t_3$$

In each case, the time differences refer to times taken from different clocks which may be offset from each other. However, if the assumption is made that the delay in one direction is the same as the delay in the opposite direction, then the two equations can be combined as follows:

$$Delay = \frac{(t_2 - t_1) + (t_4 - t_3)}{2}$$

From Figure 2.1, it can be seen that the slave clock offset (the time interval by which the slave leads the master) is given by:

$$Offset = t_2 - (t_1 + Delay)$$

Substituting from 2.1 above:

$$Offset = t_2 - (t_1 + {}^1\!/_2\,[(t_2 - t_1) + (t_4 - t_3)])$$

rearranging gives,

$$\begin{aligned} Offset &= t_2 - t_1 - {}^1\!/_2 t_2 + {}^1\!/_2 t_1 - {}^1\!/_2 t_4 + {}^1\!/_2 t_3 \\ &= {}^1\!/_2 (2 \times t_2 - 2 \times t_1 - t_2 + t_1 - t_4 + t_3) \\ &= \frac{(t_2 - t_1) - (t_4 - t_3)}{2} \end{aligned}$$

If two sets of Sync and Follow up messages are sent, then the drift between the two clocks (the phase change rate) can be found by comparing the ∆time between the successive sync messages.

$$Drift = \frac{\Delta time_{slave} - \Delta time_{master}}{\Delta time_{master}}$$

Best Master Clock Algorithm

The best master clock (BMC) algorithm is central to the operation of PTP. It specifies the method by which each clock determines the best master clock in its subdomain out of all clocks it can see, including itself. The decision is based upon the stratum (clock quality) number of the local clock (GPS and Atomic are stratum 1), the clock identifier/accuracy of the clock's time base, the stability of the local oscillator and the closest clock to the grand-master (based on the spanning tree algorithm). If there is a match between two ports in a subnet then the final decision is based upon the universally unique identifier (UUID) of the port. The algorithm was designed so that no negotiation has to occur between clocks, while ensuring that configurations with two masters, no masters or an oscillation between masters never occur.

Switch Delays

The majority of Ethernet switches on the market use a store-and-forward method to decide where to send individual packets. Incoming packets are stored in local memory while the MAC address table is searched and the cyclic redundancy field of the packet is checked before the packet is sent out on the appropriate port/s. This process introduces variations in the time latency of packet forwarding due to packet size, flow control, MAC (Media Access Control) address table searches, cyclic redundancy check (CRC) calculation and the delays associated with incoming and outgoing packet buffer queuing. The variations in these delays means that the assumption that packet delay is the same in each direction is invalid, thus rendering the path delay calculations of PTP inoperable. This issue has been compensated for with the use of two special switches, boundary clocks (version 1 and version 2) and transparent clocks (version 2) [3].

Components of a PTP Network

Version 1 of the standard published in 2002 [1] describes the devices needed to comprise a PTP network. Version 2 of the IEEE 1588 standard was published late 2008. The principal changes from version 1 of the standard are: a provision for higher accuracy, optional shorter frames, varied update rates (rather than 1 per second), unicast communications, rapid reconfiguration after network changes, fault tolerance (through extensions to allow for redundant systems) and the introduction of transparent clocks.

Figure 2.2 shows a possible PTP synchronisation network topology. The grandmaster clock is the primary time source, a boundary clock creates segmented synchronisation subdomains, and ordinary clocks synchronise to the boundary clock through end-to-end transparent clocks.

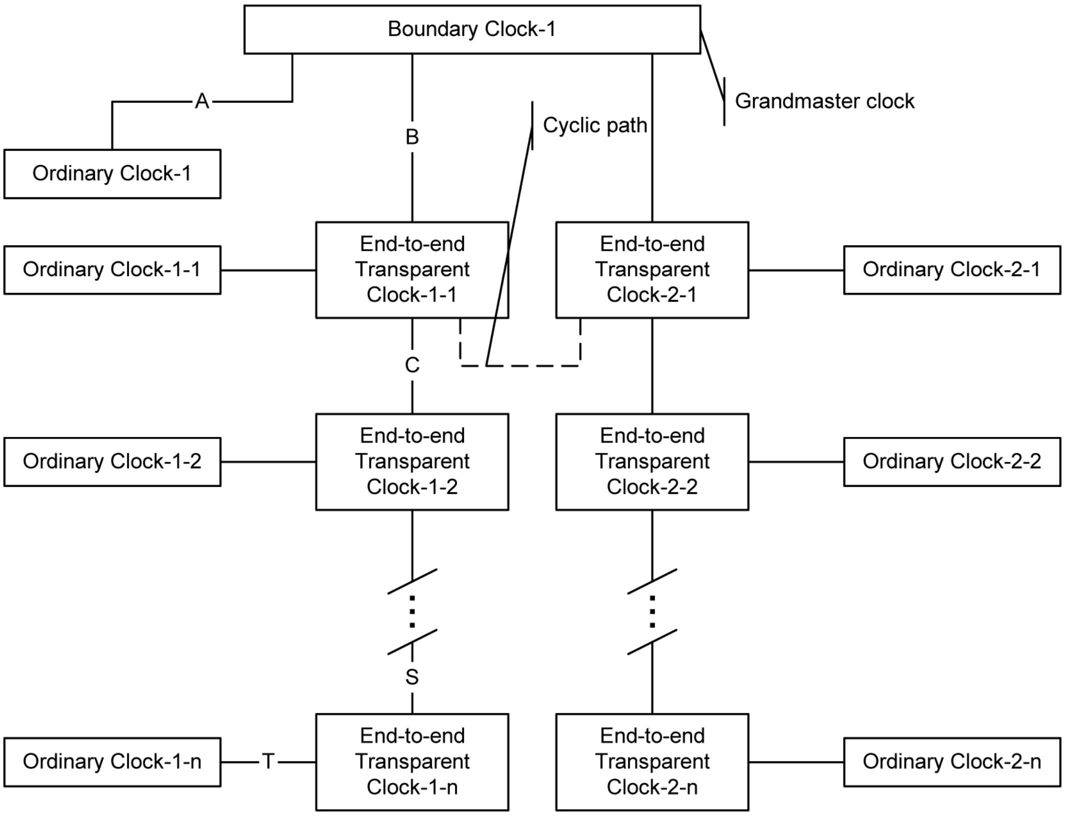

Figure 2.2: IEEE 1588 synchronisation network [3]

Grandmaster Clock

This is the primary reference source within a PTP subdomain, the “ultimate source of time for clock synchronisation using the PTP protocol” [1]. The Grandmaster clock will generally have a high-precision time source, which can be a GPS reference or an Atomic clock. If synchronisation is needed purely within a network and not to any external reference (such as UTC - Coordinated Universal Time), then the grandmaster clock could also free run.

Ordinary Clock

An ordinary clock is formally defined as a PTP clock with a single PTP port. It operates as a node within a PTP network, and can be selected as a master or slave within a segment according to the BCM algorithm. Ordinary clocks are the most populous device within a PTP network as they are generally used as the end nodes within a network connected to devices needing synchronisation. Ordinary clocks can come in various forms and with various interfaces to external devices.

Boundary Clock

Boundary clocks are defined within a PTP system to sit in place of standard network switches or routers. Boundary clocks are defined as PTP clocks “with more than a single PTP port, with each port providing access to a separate PTP communication path” [1]. The boundary clock acts as an interface between separate PTP domains intercepting and processing all PTP messages and passing all other network traffic. The BMC algorithm is used by the boundary clock to select the best clock any port can see. The chosen port is set as a slave and all other ports of the boundary clock are asserted as masters to their domain.

Transparent Clocks

Transparent clocks have been added to version 2 of the standard as an improved method of forming cascaded topologies. Rather than acting as a multi-port ordinary clock as boundary clocks do, transparent clocks update a newly introduced time-interval field within PTP event messages. This 64-bit time-interval correction field allows for switch delay compensation to a potential accuracy of less than a picosecond.

There are two types of transparent clocks, end-to-end and peer-to-peer. End-to-end transparent clocks update the time interval field for the delay associated with individual packet transfers, whereas peer-to-peer transparent clocks measure the line delay associated with the ingress transmission path and include this delay in the correction field also. Peer-to-peer transparent clocks can allow for faster reconfiguration after network topology changes.

State of the Art of Time Synchronisation

Comparison to other protocols

The Precision Time Protocol (PTP) has been designed as an improvement on current time synchronisation technologies. Table 3.1 provides a comparison of several major protocols.

Network Synchronisation

The Precision Time Protocol arose out of a need for greater synchronization over networks, particularly Ethernet. Several different research institutions were working in parallel to develop an improved method to NTP which is currently the most common network synchronisation protocol. Several of these research projects were incorporated into or have provided improvements to IEEE 1588.

	PTP	**NTP**	**GPS**	**IRIG-B**
Spatial Extent	A few subnets	Wide area	Wide area	Local
Communications	Network	Internet	Satellite	Dedicated
Target accuracy	Sub-microsecond	Few milliseconds	Sub-microsecond	Sub-microsecond
Style	Master/Slave	Peer ensemble	Client/server	Client/server
Resources	Small network message and computation footprint	Moderate network and computation footprint	Moderate computation footprint	Small computation footprint

Applications of PTP

PTP is useful in a wide range of applications of which a number of papers have been presented. Initially evolving out of a research program at Agilent Laboratories headed by Dr John Eidson, PTP was designed to satisfy the need for synchronisation in test and measurement environments but is also beneficial in various other industries. The major uses for a high level of synchronisation within an industrial motion-control environment are for sequencing event measurements, scheduling outputs, synchronising actuation, time-stamping logged data and coordinating events with a GPS time base [6]. Multistand printing presses such as that in Figure 3.1 process paper at up to 100 kph and therefore require a very high level of synchronisation.

Synchronisation is also very important within the telecommunications industry and this market has become a driving force in the progression of the standard. GSM, WCDMA, and CDMA2000 (common mobile phone standards) require frequency accuracy of 0.05 ppm (parts per million) at the air interface. CDMA2000 requires time synchronisation at the ± 3 µs level (± 10 µs worst case) and WCDMA TTD mode requires accuracy of ± 1.25 µs between base stations [18]. These requirements over large distances created the need for enhancements to the standard, specifically increased resolution and accuracy, corrections for asymmetry, and the use of shorter frames.

Figure 3.1: Multi-stand printing press [7]

The Audio Video Bridging Task Group (part of the Ethernet standards working group) has the goal of providing a standard that will allow for time-synchronised low latency streaming of audio and video data overwide area networks. IEEE 802.1AS incorporates IEEE 1588 as the standard to perform both time and frequency synchronisation [12]. Streaming audio and video is widely used over Ethernet networks today, where the variable delay that is introduced by the network is addressed by buffering and adaptive clock recovery. These techniques introduce a large amount of latency and the recovered clocks lack the time-of-day accuracy required to tightly synchronise the rendering of AV signals at different locations, such as required for digital speakers [8].

PTP can also be implemented across wireless Ethernet. This has applications such as location based LAN security, or proximity based control of industrial equipment. The high level of synchronisation provided by PTP could be useful in locating mobile nodes connected to a wireless LAN [14]. A hardware prototype presented in [13] shows that synchronization accuracy close to the nanosecond level could be achievable using hardware based packet timestamping.

Ethernet Switches

An Ethernet switch is used within networks to connect together different network technologies at the MAC (Media Access Control) level. Ethernet, Fibre Channel and ATM can all be connected together using a switch, which will connect packets to their appropriate destination. This packet interconnection can occur using information from one of the multiple OSI layers defined by the OSI reference model [5].

Layer-1 hubs are essentially repeater devices that recreate any signal sent to them on all of their ports. If multiple ports are trying to communicate at the same time then this will cause collisions. The Ethernet specification [2] allows for collisions to occur as Ethernet was originally defined to be used across a shared coaxial bus. Collisions should of course be avoided as they slow down the transfer of data.

Layer-2 switches allow for data packets to be sent to their appropriate destinations based upon Layer-2 addressing information, specifically the MAC address. Switches learn the MAC addresses of the devices connected on the different ports of the switch and send the data packet appropriately. Layer- 3 switches are termed routers and route data packets based on the packet’s Layer-3 address (the IP address in the case of Ethernet). Switches can therefore be used to create networks with a high number of network nodes with little performance degradation.

As noted, switches introduce a variable delay to packets as they are routed. This delay is mainly attributed to packet address lookup tables and input and output queuing. Most switches have a number of queues to which packets can be assigned depending on their types. The Quality of Service (QoS) field allows for packets to be classified into one of eight different priority levels [19], and packets can be assigned to different output queues depending on their priority level. A high QoS priority will reduce the packet latency to an extent but the variation is still greater than that required by PTP.

While the delay introduced to packets varies from switch to switch, comparisons can readily be made of packet delay variations introduced by various network mediums.

Table 3.2 displays data from [4] which shows the mean latency, and the peak-to-peak and standard deviation of the packet delay variation through a crossover cable, a hub, a switch and a router under no load. As can be seen, an increase in complexity of the device shows an increase in the packet delay and variation.

Crossover Cable	
Mean:	287.3 ns
Peak to Peak:	10.0 ns
Standard Deviation:	4.5 ns
Hub	
Mean:	659.8 ns
Peak to Peak:	60.0 ns
Standard Deviation:	12.1 ns
Switch	
Mean:	16.8 µs
Peak to Peak:	310.0 ns
Standard Deviation:	70.1 ns
Router	
Mean:	277.7 µs
Peak to Peak:	212.5 µs
Standard Deviation:	20.6 µs

Table 3.2: Packet delay distribution and statistics [4]

Under the presence of packet loading, a switch will exhibit a larger delay variation as can be seen in Table 3.3 (data from [4]). The standard deviation increases significantly with an increase in traffic due to the competition for output queues, and the increased size of packet address tables.

No Traffic	
Mean:	16.8 µs
Peak to Peak:	310.0 ns
Standard Deviation:	70.1 ns
10% Load	
Mean:	17.9 µs
Peak to Peak:	121.4 µs
Standard Deviation:	11.5 µs
25% Load	
Mean:	19.6 µs
Peak to Peak:	122.6 µs
Standard Deviation:	17.6 µs
50% Load	
Mean:	48.0 µs
Peak to Peak:	122.8 µs
Standard Deviation:	50.9 µs

Table 3.3: Packet delay as a function of load [4]

Boundary Clocks

Version 1 of the IEEE 1588 protocol [1] describes one method of removing the negative effects that standard Ethernet switches have on the synchronisation process. Boundary clocks are switches that incorporate a clock and they create separate synchronisation domains by segmenting the synchronisation path from master clocks to slave clocks. Standard Ethernet messages are passed through the switch while synchronisation messages are used to synchronise the boundary clock. Boundary clocks are effective for networks in which slave clocks are not separated from their synchronisation masters by more than a few switch hops.

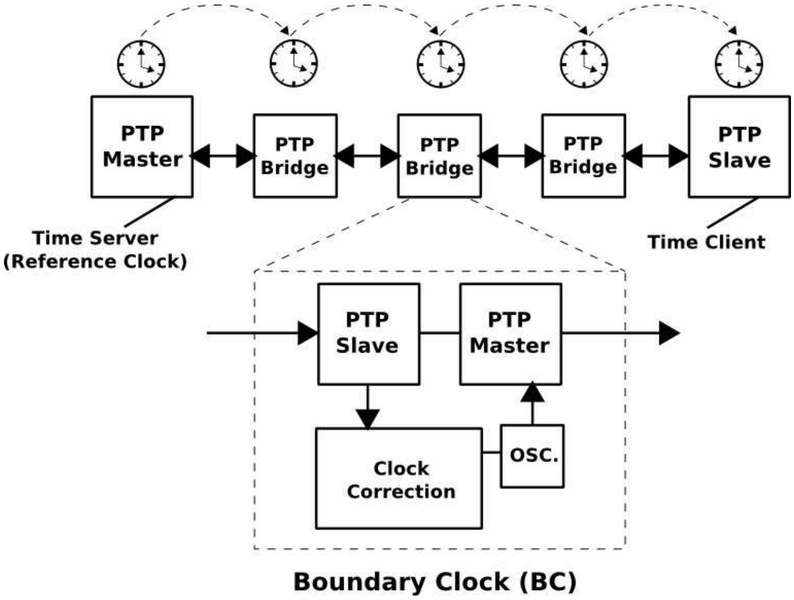

Figure 3.2: Cascaded boundary clocks

Networks frequently have topologies where switches are cascaded, such as in Figure 3.2. Each boundary clock uses a control loop to recover the local clock from its master, and when these control loops are cascaded non-linear errors accumulate [10]. This effect is due to two principal mechanisms: (1) the PTP message exchanges between different pairs of successive nodes in the chain are not necessarily synchronized, and (2) there may be gain peaking and noise generation in phase-locked-loop servos at successive nodes [9]. The effect of these can be seen in the following measurements [15].

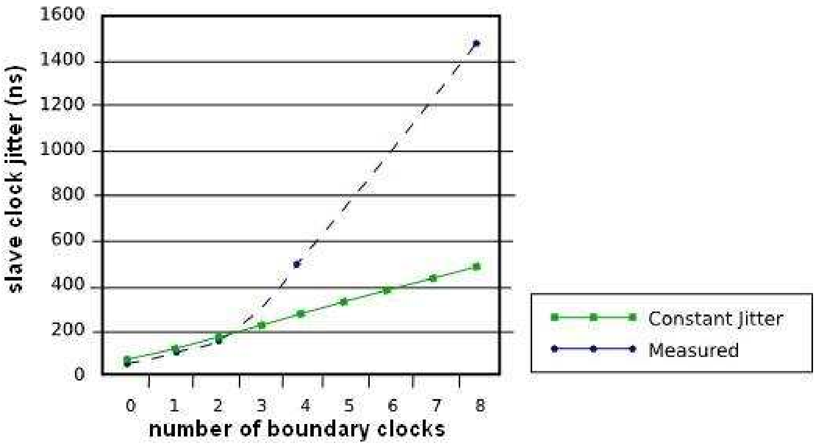

Figure 3.3: Boundary clock jitter [15]

Figure 3.3 displays the jitter experienced by a slave clock synchronizing to a master through different numbers of boundary clocks. It is seen that the precision of a synchronised slave clock diverges from the linear plot of accumulated errors. Boundary clocks also have the negative effect of increasing the reconfiguration time of the network. After a network topology change (which could occur if a master clock loses its GPS reference), the biggest effect on time to resynchronise is the BMC algorithm. During reconfiguration there is no synchronicity on the network as each clock's local time must be re-synchronised to that of its master. This reconfiguration time increases linearly with the number of hops.

It should be pointed out that boundary clocks are still needed in IEEE 1588 synchronised networks as there will always be a need to translate between either protocol version or communication technologies. Improvements to the synchronisation methods of boundary clocks were recognised and results were published at the 2007 ISPCS conference [16]. The improvements revolve around altering the delay measurement process and the control loop for drift and offset measurement.

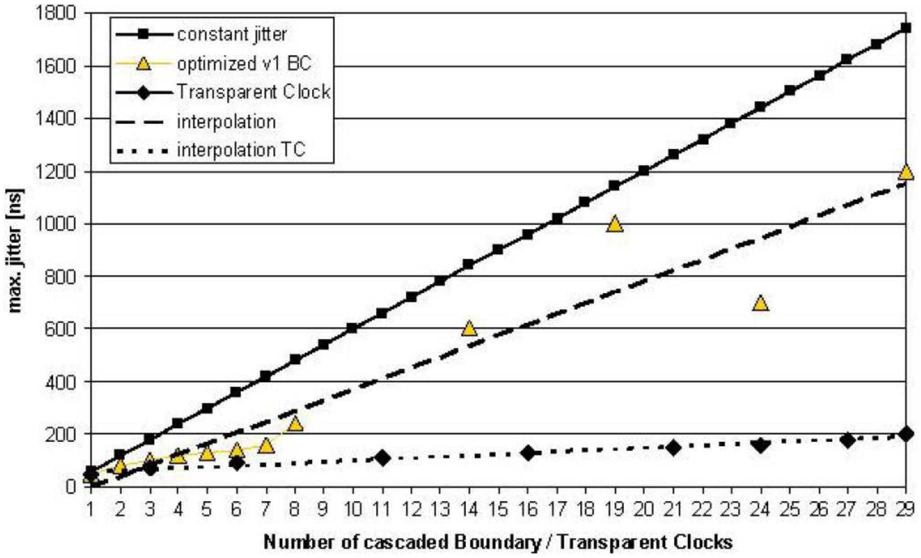

Figure 3.4: Improved boundary clock jitter and transparent clock jitter [16]

The reduction in jitter (Figure 3.4) through these adjustments is significant as it has reduced the jitter to below that of a linear addition. The jitter shown by a transparent clock over the same number of network hops is still much less.

Transparent Clocks

Transparent clocks, initially termed bypass clocks, were first introduced in 2004 by Siemens [11]. Simulations were performed in order to investigate the characteristics of boundary clocks compared to transparent clocks over a number of network hops. At the 2004 Conference on IEEE 1588 the first transparent clock design was presented [17]. This transparent clock was a pre-standard design that allowed for end-to-end transparency of version 1 sync and follow up messages. The introduction of transparent clocks allows for a far simpler solution to correcting for variable switch latency. The key benefits [17] of the introduction of transparent clocks are as follows:

- **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. It is however, possible for transparent clocks to have a data set accessible to other managed nodes on the network. This data set would contain information such as the number of ports, what delay mechanism is used (E2E or P2P), whether or not the local oscillator is syntonised (frequency synchronised) to any port and which port this is, and what the quality of the local oscillator is. However his information is not required by version 2 of the standard [3].

- **Faster setup times:** At initialisation and after a change in topology, transparent clocks do not have to resynchronise to a master clock before they can be considered part of a valid synchronised path.
- **Less interoperability problems:** Each ordinary clock will have slight differences in the way in which they implement the BMC algorithm. Clock properties such as the estimate of the clock variance parameter can change the way in which each clock calculates which clock is the best master.

End-to-end vs. Peer-to-peer

Each of these two methods has their advantages and disadvantages. End-to-end transparent clocks scale poorly with the number of devices connected to the subnet as the master "sees" all the slaves. This can introduce a 1:N topology where there is one master communicating with a large number of slaves. They are however, good for linear systems where there are lots of daisy chained clocks. In contrast, peer-to-peer clocks, scale well with the number of devices attached to the subnet and can recover rapidly to changes in network topology. They cannot resolve 1:N topologies as they cannot tell which line delay is being calculated and they must also maintain path delay measurements [6].

One-step vs. Two-step

PTP allows for two different types of timestamping methods, either one step or two-step. One-step clocks update time information within even messages (sync and delay-request) on-the-fly, while two-step clocks convey the precise timestamps of packets in general messages (follow-up and delay-response). A one-step end-to-end transparent clock updates for switch delay in sync and delay-request messages as they pass through the switch while a two-step transparent clock updates a field in the non time-critical general message.

The use of follow-up messages and delay response messages to convey precise timestamp and time-interval data requires more network bandwidth and extra processing. Tekron Transparent clocks are designed to operate as one-step clocks to maximise the efficiency of the protocol.

Contact

I welcome feedback on this paper.

If you would like to discuss this paper or would like information on Tekron's IEEE 1588 products, please feel free to contact us at the email address below or through our website.

- Caleb Gordon
Design Engineer
Tekron International

PO Box 31 285, Lower Hutt 5040
Phone: +64 4 566 7722
Email: information@tekroninternational.com
Fax: +64 4 569 9272
Web: www.tekroninternational.com

Bibliography

[1] IEEE Std. 1588 - 2002 IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems. *IEEE Std 1588-2002* (2002), i–144.

[2] IEEE Std 802.3 - 2005 Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications - Section Three*. IEEE Std 802.3-2005 (Revision of IEEE Std 802.3-2002 including all approved amendments) Section 3* (2005), 1–387.

[3] Draft Standard for a Precision Clock Synchronization Protocol for Net- worked Measurement and Control Systems. IEEE Unapproved Draft Std P1588/D2.2, Mar 2008 (2008).

[4] COSART, L. Precision Packet Delay Measurements Using IEEE 1588v2. *Precision Clock Synchronization for Measurement, Control and Communication, 2007. ISPCS 2007. IEEE International Symposium on* (Oct. 2007), 85–91.

[5] DAY, J., AND ZIMMERMANN, H. The OSI reference model. *Proceedings of the IEEE 71, 12* (Dec. 1983), 1334–1340.

[6] EIDSON, J. IEEE-1588 Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems - A Tutorial. *Proceedings of the Workshop on IEEE 1588* (2005).

[7] EIDSON, J., AND LEE, K. Sharing a common sense of time. *Instrumentation & Measurement Magazine, IEEE 6, 1* (Mar 2003), 26–32.

[8] GARNER, G., FENG, F., DEN HOLLANDER, K., JEONG, H., KIM, B., LEE, B.-J., JUNG, T.-C., AND JOUNG, J. IEEE 802.1 AVB and Its Application in Carrier-Grade Ethernet [Standards Topics]. *Communications Magazine, IEEE 45, 12* (December 2007), 126–134.

[9] GEOFFREY M. GARNER, SIHAI WANG, J. C. Improvements to Boundary Clock Based Time Synchronization through Cascaded Switches. *Proceedings of the 2006 Conference on IEEE 1588 Standard for a Precision Clock Synchronization Protocol*.

[10] HANS WEIBEL, D. M. Pre-Standard Prototype Implementation of an End-to-End Transparent Clock. *Proceedings of the 2006 Conference on EEE 1588 Standard for a Precision Clock Synchronization Protocol* (2006).

[11] JASPERNEITE, J., SHEHAB, K., AND WEBER, K. Enhancements to the time synchronization standard IEEE-1588 for a system of cascaded bridges. *Factory Communication Systems, 2004. Proceedings. 2004 IEEE International Workshop on* (Sept. 2004), 239–244.

[12] KANG, S.-H., LEE, J.-W., EOM, J.-H., KWON, Y.-S., AND KIM, S.H. Time Synchronization and Resource Reservation for Time Sensitive

[13] Applications in Bridged Local Area Networks. *Advanced Language Processing and Web Information Technology, 2007. ALPIT 2007. Sixth International Conference on* (Aug. 2007), 364–369.

[14] LOSCHMIDT, P., GADERER, G., AND SAUTER, T. Clock Synchronization for Wireless Positioning of COTS Mobile Nodes. *Precision Clock Synchronization for Measurement, Control and Communication, 2007. ISPCS 2007. IEEE International Symposium on* (Oct. 2007), 64–69.

[15] MOHL, D. Experiences with IEEE 1588 in Higher Cascaded Ethernet Networks: Evaluation of Network Elements. *Proceedings of the 2005 Conference on IEEE 1588 Standard for a Precision Clock Synchronization Protocol* (2005).

[16] MOHL, D., AND RENZ, M. Improved synchronization behavior in highly cascaded networks. *Precision Clock Synchronization for Measurement, Control and Communication, 2007. ISPCS 2007. IEEE International Symposium on* (Oct. 2007), 96–99.

[17] NYLUND, S., AND ØYVIND HOLMEIDE. IEEE 1588 Ethernet switch transparency - No need for Boundary Clocks. *Proceedings of the 2004 Conference on IEEE 1588 Standard for a Precision Clock Synchronization Protocol*.

[18] RODRIGUES, S., AND PIETILAINEN, A. IEEE 1588 Telecom Applications Tutorial. In *Proceedings of the 2005 Conference on IEEE 1588 Standard for a Precision Clock Synchronization Protocol (2005).*

[19] SEIFERT, R. *The Switch Book*. Wiley, 2000.