EBU
OPERATING EUROVISION AND EURORADIO

TECHNICAL REVIEW

Using PTP for Time & Frequency in Broadcast Applications

Part 1: Introduction

MARCH 2018

Thomas Kernen, Mellanox

Nikolaus Kerö, Oregano Systems

The main purpose of an EBU Technical Review is to critically examine new technologies or developments in media production or distribution. All Technical Reviews are reviewed by 1 (or more) technical experts at the EBU or externally and by the EBU Technical Editions Manager. Responsibility for the views expressed in this article rests solely with the author(s).

To access the full collection of our Technical Reviews, please see: tech.ebu.ch/publications

If you are interested in submitting a topic for an EBU Technical Review, please contact: tech@ebu.ch

Abstract

As the broadcast industry continues to transform itself, leveraging new infrastructure and transport mediums for content production and delivery, the concept of the All-IP Studio is making inroads. Therefore a significant focus has been placed on Ethernet and IP transport as the basis of these new systems, including the transport of Video, Audio and Ancillary data with the likes of the SMPTE ST 2110 standards. This multipart series covers a specific aspect of this ongoing transformation, the transport and use of phase and frequency for the purpose of timing over a converged IP network.

In this series, we will start from the basics of the IEEE1588 Precision Time Protocol (PTP), its relationship to the broadcast industry and the related network requirements. As we build out from the basics, we will cover specific PTP design considerations, both from a network and end node perspective. This series will then focus on more advanced topics, such as PTP redundancy, where we will drill deeper into the technical details.

1. Introduction

A highly accurate yet reliable common notion of time is a crucial mandatory requirement for every broadcasting application. To process media content captured simultaneously by multiple cameras and audio recorders, all devices within a production chain need to be tightly and reliably synchronised to each other on a permanent basis. As long as SDI was used for media transport, accurate frequency transfer was a given, because SDI is an inherently synchronous communication medium. Every end device merely had to synchronize its local oscillator to the frequency provided by the SDI link, the black & burst or Tri-Level sync via a suitable PLL. However, accounting for delays caused by varying cable lengths within a studio often required manual calibration procedures to be executed regularly in order to provide precise phase alignment. Finally, absolute time transfer was implemented via time codes embedded within the video signal. Depending on the respective video standard, different SMPTE time codes have to be both provided and decoded. Within a facility where Sync Pattern Generators (SPGs) are used as a reference both for frequency and absolute time, these devices are usually linked to an external time reference such as GPS to provide time traceable signals.

As both frequency and time transfer are key services within every production environment, provisions had to be taken to ensure their continuous availability. One or more auxiliary SPGs were installed together with monitoring devices which will automatically trigger a switch-over in case of failure of the currently active SPG.

Although SDI is a well-known and proven technology in all respects with a huge installed base worldwide, it has several stringent bandwidth limits which are becoming ever more apparent with the advent of new high-resolution video formats, (4k/UHD, high video frame rate, high dynamic range, etc.). Consequently, the broadcasting industry's uptake of the all-IP studio is gaining more and more momentum, because it can provide both the highly scalable bandwidth and the flexibility required for modern media processing. Furthermore, to make best use of a single communication medium, Ethernet has to be used both for media transport and synchronization rather than having to provide additional infrastructure merely for transporting time and frequency to all devices within a studio. Considering that Ethernet is inherently an asynchronous medium, frequency transfer cannot be accomplished in the same way it is done with SDI. In general, only the clocks of two adjacent nodes which are directly connected via a physical link are synchronized with each other, without any preference on which of the two nodes will become the frequency source. Furthermore, the nodes remain phase locked only as long as data is actually transmitted over the link.

In the recent past, other application domains have been facing similar challenges when moving from legacy communication systems to Ethernet as their sole solution. The telecom industry and industrial automation are both typical examples coming to mind. The telecom industry moved from a fully established TDM (Time Division

Multiplex) infrastructure to Ethernet roughly at the same time as a variety of legacy field bus systems for real-time industrial communication were being replaced by Ethernet. A common notion of frequency and, in most cases, time was a crucial requirement for both telecom and industrial automation.

The Precision Time Protocol (PTP) as defined in the IEEE1588-2008 standard [1] turned out to be best suited to provide highly accurate time and frequency transfer for a large variety of different applications. The protocol was deliberately specified in a highly generic manner offering ample room for adjusting its performance to the requirements of a specific application via PTP profiles. As of today, nine different profiles have been published with several more being currently specified. The broadcasting industry is relying on AES67 [2] and SMPTE ST 2059-2 [3] for clock synchronization, the former published by the Audio Engineering Society is intended primarily for audio applications while the latter is focusing on video applications for the All-IP studio. In contrast to the requirements of most other applications, the broadcasting industry has stringent timing requirements for a very diverse set of use cases. PTP shall provide time transfer equally reliably to both small networks, such as Outside Broadcasting trucks as well as large studios with thousands of devices communicating with each other.

2. Basic Principles of Time Transfer with PTP

If all nodes in a network have to be synchronised with each other according to the principles defined in IEEE1588, they need to exchange so-called event messages periodically. PTP follows a strict Master-Slave principle for transmitting time information. For the time being, let us assume that the network has already selected one node to become its Master. The synchronization technique relies on a simple principle: The Master transmits synchronization messages (*Sync_messages*) to all Slave nodes within the respective network on a regular basis (typically at least once every second). The content of these messages is basically the current time of the Master. Actually, it should be the very point in time (labelled as T_1) at which the Master starts sending the message via the physical channel. Every Slave, in turn, denotes the time at which it receives any such *Sync_message* on its local time scale (labelled as T_2). The difference between these two timestamps is the offset between the two clocks plus the transmission delay of the message via the physical channel.

$$T_2 - T_1 = Transmission\_Delay + Clock\_Offset$$

If the Master is not able to insert a timestamp into the *Sync_message* with sufficient accuracy while actually sending it (for details on effects deteriorating the accuracy see below), it will merely note the time at which the packet is sent over the network by drawing a timestamp from its accurate local clock while actually sending such a message and later on forward this time information by means of a corresponding *Follow_up_message* again to all its Slaves. The former method is referred to as one-step and the latter as two-step mode. It makes no difference at all for a Slave whether the Master operates in one- or two-step mode. It simply needs to retrieve T_1

from different messages, therefore, the support for both modes is mandatory for every Slave.

To calculate the transmission delay, the Slave performs a second time transfer procedure by sending a Delay Request packet (*Del_req_message*) noting the time when the transmission over the physical medium is initiated (labelled as T_3). The Master in turn will record the time when it has received such a packet (labelled as T_4) and will relay this data back to the querying Slave by sending a so-called *Del_resp_message*. This measurement cycle is continuously repeated to allow for filtering and account for topology changes. The difference of the two timestamps of the *Del_req_message* equals the clock offset minus the transmission delay:

$$T_4 - T_3 = Transmission_{Delay} - \ Clock_{Offset}$$

Now the Slave clock is able to calculate both the clock offset and the transmission delay using both timestamp differences:

$$Clock\_Offset = \frac{(\,T_2 - T_1) - (T_4 - T_3)}{2}$$

$$Transmission\_Delay = \frac{(\,T_2 - T_1) + (T_4 - T_3)}{2}$$

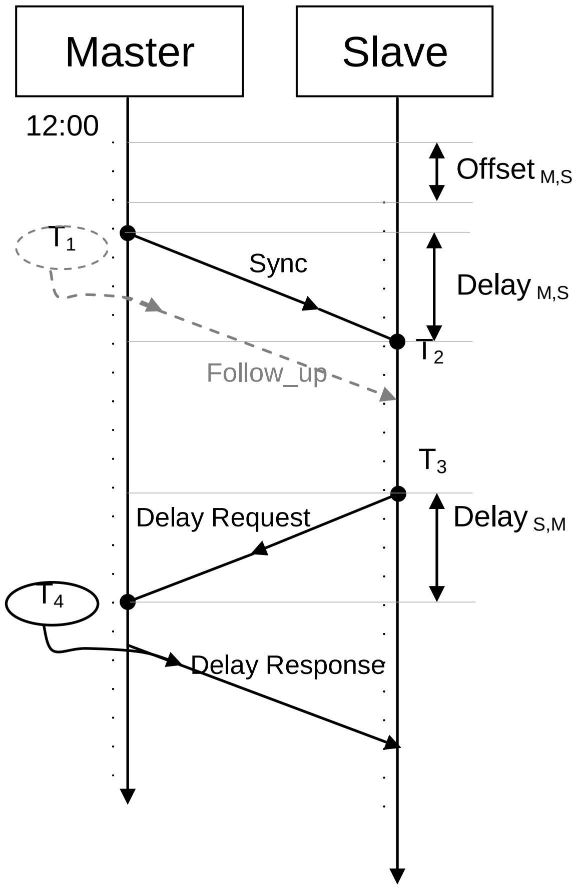

Figure 1: PTP Message flow

The message flow of the complete IEEE1588 synchronization process shown in Figure 1 refers to any end node as Ordinary Clock (OC) as opposed to devices within the network with PTP functionality (see below).

3. PTP Accuracy

The overall accuracy depends on a number of factors, the most obvious one being the precision with which the four timestamps can be taken. If sub-µs accuracy is required, a dedicated hardware module located in every PTP node is required. It scans all incoming and outgoing traffic as close as possible to the physical network interface and draws a timestamp from a dedicated hardware clock module upon detection of a PTP event message. This clock should have a reasonably high resolution, which modern systems can offer with less than 10 ns.

Within modern Ethernet networks, all nodes are connected to each other via intelligent network devices such as switches and/or routers performing Layer-2 or Layer-3 based packet switching, respectively. This technique has proven to be extremely efficient in terms of overall network throughput allowing every node to operate in full-duplex mode. With respect to accurate clock synchronization such active network components have one severe drawback: as shown above, the accuracy a Slave is able to synchronize to its Master relies on a known AND constant transmission time of the all PTP event messages. The time it takes for a standard network switching device to forward a network message is by no means constant. On the contrary, any network device will introduce so-called packet delay variations (PDVs) to a certain degree. Its value and distribution depends heavily on the architecture of the network elements used and, equally important, on the current load of the network. Whenever a message is received, part of its header information is analysed to decide to which of its output ports it has to be forwarded, and in case of Layer-3 devices the message will be modified during this process as well. This process introduces latency, which may vary considerably, depending on both the hardware architecture of the network switch itself and the current network loading condition the device has to handle, i.e. traffic from other ports to be forwarded to the same port as the PTP traffic at the same time.

To a certain extent, the effect of PDVs can be mitigated by the PTP Slave using complex (non)linear filters within the control loop it uses to adjust its local clock. Such filtering has proven to yield impressive results. As an additional measure, Quality of Service (QoS) techniques can be utilised by configuring a high priority queue for PTP traffic on every network device and tagging all PTP event messages accordingly. This method does reduce the impact of loaded network devices on synchronization accuracy; however, assigning QoS for PTP traffic may very well collide with other requirements within a given network.

4. Master Election Process

As mentioned before, PTP requires one node to act as a Master while all others revert to Slave mode. It is important to note that, the Master election is an autonomous process governed by the Best Master Clock [selection] Algorithm (BMCA), which ensures that the node with the "best" (most stable) local clock always becomes the PTP Master. It is triggered by two distinct events: Whenever the currently active Master becomes inoperable or when a node with a "better" (more accurate) clock is attached to the network. The data all nodes use to decide on the Best Master is conveyed via PTP Announce messages that the current Master has to send continuously to all Slaves. They contain information about the clock quality. As long as all nodes receive these at preconfigured intervals they will remain in Slave state. If they stop receiving Announce messages for a given period of time, all nodes will initiate the BMCA by starting to send Announce messages advertising their respective clock quality. They will compare the data of all Announce messages they have received from the other nodes with their own local data set. The node with the "best" clock will switch to Master mode.

Furthermore, the BMCA can account for a better Master being connected to the network and wanting to take over. Such a device will evaluate the contents of the Announce messages from the current Master. If it decides to take over, it will start sending Announce message itself eventually causing the current Master to back off and all other nodes to start using it as their time reference.

With the BMCA, PTP has succeeded in defining a "configurationless" protocol which even provides fault tolerance, although this feature is limited to a sub-set of possible failure conditions such as recovery from Master failures. A detailed analysis on PTP fault tolerance with emphasis on broadcasting applications will be provided in subsequent issues of this document series.

5. PTP Aware Network Devices

IEEE1588 addressed the problem of PDVs caused by network devices by introducing two types of PTP-aware network devices, Transparent Clocks (TCs) and Boundary Clocks (BCs). The former act as normal network devices treating only PTP event messages in a special manner. A TC comprises an accurate clock allowing it to measure the time it requires to forward any given PTP event message. A timestamp is drawn from its clock upon reception of such a message and stored locally. If the message is re-transmitted via any other port of the TC, another timestamp is drawn. The first timestamp is retrieved and the difference between the two timestamps is calculated, which equates to the residence time of the packet. This information is either inserted into a *correction_field* within the *Sync_message* (*Del_req_message*) or stored and inserted into the respective field of the corresponding *Follow_up_message* (*Del_resp_message)*. The former method is referred to as one-step and the latter as two-step Transparent Clock.

It has to be noted that 1-step and 2-step devices are by no means mutually exclusive within a network. Any Ordinary Clock operating as a Slave has to be able to extract all data from the *correction_field* regardless whether it is contained in the *Sync_message* or in the *Follow_up_message* or in both of them. The latter would be the case, if both 1-step and 2-step TCs are used. To enable cascading of Transparent Clocks the respective residence times are accumulated rather than just inserted into the correction field. Every Slave is now able to account precisely for variations in the transmission time on a per packet basis in downstream as well as in upstream direction. If the correction field value is simply subtracted from the respective send time stamp, the subsequent delay calculation yields the overall transmission over the respective physical channels of all network segments between the Master and the Slave. This delay typically varies in the range of less than 1 ns per segment.

Boundary Clocks, on the other hand, are intended to partition time distribution within large networks effectively reducing the number of messages a single PTP Master node has to process. Rather than simply forwarding PTP messages from a given Master to all ports as TCs do, Boundary Clocks (BCs) terminate all incoming PTP traffic. The PTP event messages are used to synchronize a highly accurate local hardware clock of the BC to the Master attached to the respective port. Basically, a BC acts as a Slave synchronizing to the Master connected to this port. All other ports will generate *Sync_messages* using the time information of the local clock. To this end, each port of a Boundary Clock has to be capable of acting both as a PTP Master or Slave with all ports sharing the same internal clock. One port will assume the Slave role whilst all other ports will act as PTP Masters (or passive Master, if there is already a better Master in this part of the network). Rather than assuming these roles in a predefined way by means of static configuration, the role of every port will be determined dynamically by the BC itself.

To accomplish this, it will execute an extended version of the BMCA evaluating clock quality information contained in the *Announce_messages* on every port it’s receiving. If more than one port is receiving *Announce_messages,* the information on the clock quality is compared and the most accurate clock is selected as a Master. The respective port will switch to Slave state while all other ports will revert to Master state sending *Announce_messages* themselves, eventually causing all nodes connected to these ports to switch to Slave state. This mechanism supports cascading of BCs as well. Just as TCs, Boundary Clocks can be cascaded without any limitations with respect to the protocol.

6. PTP Profiles

Version 2.0 of the Precision Time Protocol as published in the IEE1588-2008 standard has been deliberately defined as a highly generic protocol, leaving ample room for tailoring it to the specific requirements of different application domains, which are more often than not mutually exclusive. The telecom industry, for example, operates large Metro area networks with scarce PTP support at network device level while time transfer in the closed confines of a power plant, factory floor, or a broadcasting studio can benefit from partial or full PTP support at network level, whilst having to cope with different constraints such as tighter accuracy requirements. Consequently, PTP can be customised for different use cases and application scenarios via PTP profiles. The IEEE1588-2008 standard provides detailed guidelines and rules on how to specify a PTP profile. Different standards organisations have defined a number of profiles to tailor PTP for their respective use cases and needs. So far, two PTP profiles have been published specifically for broadcasting applications: The Audio Engineering Society has defined time transport for audio applications with the AES67 standard, whilst SMPTE has been focusing on the requirements within broadcasting facilities.

Among other things, a profile may be used to specify sub-ranges for all message rates, which have been rather oversized in the original standard, enabling PTP to be deployed on low bandwidth as well as on high performance networks without consuming unacceptably high network resources. The transport protocol (i.e. IPv4, IPv6 etc.) together with mandatory or suggested network structures and topologies are typically specified in a protocol. Some profiles like the SMPTE ST2059-2 standard utilise PTP to transport application data to all nodes via PTP management messages, which in this case carry the synchronization metadata TLV (Type Length Value) data set. It contains information about local time zones as well as data related to the next daily jam.

7. Coming up next

In the next part of this series, we will further discuss how PTP messages are transported over an IP network, the implications of the broadcast PTP profiles, (SMPTE ST 2059-2 & AES67) and how the building blocks of a PTP node impact its accuracy. Furthermore, we will compare different PTP aware network devices with each other with respect to their respective usage. Special emphasis will be put on covering the PTP management mechanism as it is extensively used in broadcasting applications.

8. References

[1] IEEE 1588, “IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems,” IEEE Instrumentation and Measurement Society, Piscataway, NJ, 2008.

[2] AES-67-2015, AES standard for audio applications of networks - High-performance streaming audio-over-IP interoperability, Audio Engineering Society Inc.

[3] SMPTE ST 2059-2:2015, SMPTE Profile for Use of IEEE-1588 Precision Time Protocol in Professional Broadcast Applications, Approved March 19th 2015.

9. Author(s) biographies

	Thomas Kernen is a Software Architect for Mellanox's ethernet switch platform. His main area of focus is defining architectures that aid the broadcast industry in its move to an all-IP video infrastructure. He is a member of the IEEE Communications and Broadcast Societies and the Society of Motion Picture & Television Engineers (SMPTE). He is active within a number of trade and industry organizations including the SMPTE Standards. Prior to joining Mellanox, Thomas spent 11 years at Cisco during which he worked on their IPTV distribution and broadcast contribution portfolio and drove their entrance into the live media production market.
	After receiving a Master's Degree in Communication Engineering with distinction from the Vienna University of Technology, Nikolaus led the ASIC design division at the university's Institute of Industrial Electronics, successfully managing numerous research projects and industry collaborations. His research activities centred on distributed systems design, especially highly accurate and fault- tolerant clock synchronization. In 2001 he co-founded Oregano Systems Design & Consulting Ltd. as a university spin-off. While offering embedded systems design services to customers, Oregano successfully transferred Nick's research results into a complete product suite for highly accurate clock synchronization under the brand name syn1588®, for which Nick manages both development and marketing. He is an active member of the IEEE1588 standardization committee and the SMPTE 32NF standards group and holds frequent seminars on clock synchronization for both industry and academia.

Published by the European Broadcasting Union, Geneva, Switzerland

ISSN: 1609-1469

Editor-in-Chief: Patrick Wauthier

E-mail: wauthier@ebu.ch