EBU
OPERATING EUROVISION AND EURORADIO

TECHNICAL
REVIEW

Using PTP for Time & Frequency in Broadcast Applications

Part 2: PTP Clock Characteristics

NOVEMBER 2019

Thomas Kernen, Mellanox

Nikolaus Kerö, Oregano
Systems

1 Introduction

The first part of this series gave a general overview of highly accurate packet-based time transfer in Ethernet networks using PTP, the Precision time Protocol [1] for the broadcasting industry [2], [3] explaining the message exchange and the Master election process. In this part we will describe the Master election process in more detail, focusing on the different states a PTP device, or to be more precise, a PTP port can assume. Of course, this paper is by no means intended to serve even as very basic guideline for actually designing a PTP stack, however, it is crucial to know your friends and enemies i.e. wanted and unwanted states devices in a PTP network can switch to transiently or remain in indefinitely.

After briefly touching on the concept of PTP ports the PTP states with their respective transitions are explained in detail for different classes of PTP devices. This paper concludes with comparing PTP enabled network devices and finally with describing the basic building blocks of a PTP software stack.

2 PTP Port vs. Physical Ports

IEEE1588 has been specified as a highly generic time transfer protocol to be deployed on arbitrary network architectures supporting at least some flavour of a multicast messaging mechanism i.e. allowing messages to be addressed to more than one receiver. Therefore, it can be mapped onto different networks using various transport protocols, with Ethernet, of course, being by far the most common and most widely used technology. In the strict context of IEEE1588, a PTP port is considered an entity capable of processing PTP messages. It is assumed to have two distinct interfaces one for processing general PTP messages and the other for dealing with PTP event messages i.e. messages carrying time information.

Each PTP port runs a single instance of the PTP protocol stack using a specific transport protocol. It is important to note that more than one PTP port can be mapped to a single physical port, in our case an Ethernet port. The advantage of this distinction between a physical and a PTP port becomes obvious when we consider a multi-port PTP device such as a Boundary Clock. Each PTP port can be configured individually with respect to all PTP parameters such as message rates, PTP domains or transport related data, A Boundary Clock, therefore, can link different PTP subnets with each other. Furthermore, this concept facilitates deploying PTP over multiple VLANs (Virtual Local Area Networks) without any restrictions.

A typical Boundary Clock configuration showing different assignments of PTP ports to physical ports is shown in Figure 1. In general, end devices known as Ordinary Clocks would operate using a single PTP port. However, if extended redundancy is considered the concept of multiple PTP ports comes in handy even for PTP Slave only devices. We will cover this specific topic in subsequent parts of this series.

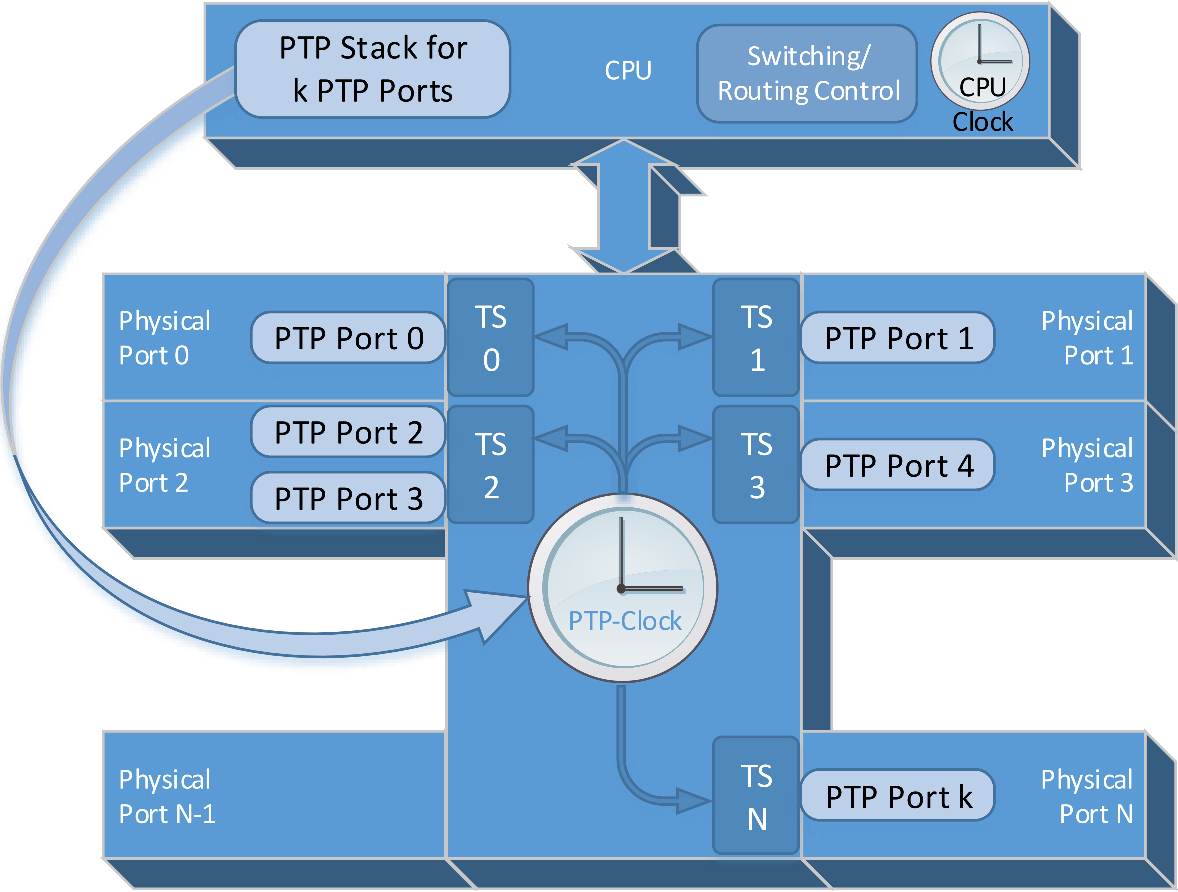

Figure 1: Typical PTP Boundary Clock Configuration

3 The Best Master Clock Algorithm revisited

As opposed to other common time transfer protocols such as NTP, all devices (PTP ports) within a PTP network assume their respective roles without any user interaction during normal operation. The network autonomously selects one device to become its Grand Master (GM). It is important to note that only one PTP device can become PTP Grandmaster at a time, while more than one PTP port may assume PTP Master role (in the case of a PTP Boundary Clock).

The selection process is governed by a series of PTP parameters describing the quality of the clock respective to the PTP port it's deriving its time information from. This data is communicated continuously via Announce messages. Every Slave, for example, knows whether the GM is deriving its time information from an external traceable time reference via a Global Navigation Satellite System (GNSS) link such as GPS, Galileo or GLONASS.

If all PTP ports receive Announce messages at the expected rate without any changes in the parameters contained in these messages, they remain in their respective state. Nevertheless, the BMCA is executed whenever an Announce message is received. Its content is compared with the local data and if these two datasets match 100%, no further action is taken.

Unless a PTP port is operating as a PTP Master its state can change only under two conditions: Either, if the data of the most recent Announce message differs from the previous message or, if no Announce message has been received for a predefined amount of time. The latter parameter is defined via the number of consecutive missing Announce messages. Together with the Announce message rate which must be specified by the user for every PTP port, the PTP protocol stack is able to calculate the timeout period.

The BMCA precisely defines how two datasets are to be compared. These could be the local dataset against an incoming Announce message or the content of two Announce message sent from different Masters during the actual election process. The datasets are compared value by value in a fixed precedence. This ensures that all PTP Ports reach the same conclusion i.e. uniformly select the same Best Master.

Although a PTP Master will send out Announce messages itself, it still must listen for, and process incoming Announce messages in the same way as described above. If upon such an event, it concludes that the Master having sent that Announce message has a better clock than itself, it must back off. Depending on its configuration it would become either a PTP Slave or switch to Passive state. This feature ensures that the Best Master will always take over regardless when it was attached to the network.

In general, every node within a PTP network can become the GM. This is particularly useful for applications where a common notion of time is far more important to maintain than the link to an external time reference. If, on the other hand, the network has to rely on precise absolute time, the network should be provisioned with several GMs each linked to one or more GNSS time sources. These would be configured in a way whereby one becomes the GM while the other(s) switch to Passive state. If the active GM fails, the remaining GMs will actively participate in the BMCA and one of the devices will assume the GM role. Such configurations imply that all other nodes shall not be able to assume the Master role. This can be accomplished by configuring the respective parameters in the Announce messages. In our example that would be the Clock Class parameter.

Boundary Clocks have to combine the data gathered by every of their respective ports during each BMCA round to reach a common conclusion as to which port should switch to Slave state based on the received time information, while all other ports will transition to Master providing time information.

Before we elaborate the state diagrams for different classes of PTP devices, the concept of PTP domains is briefly explained. Although most end users are unlikely to investigate the BMCA in its intricate details, getting familiar with the basic principles is tremendously helpful when analysing PTP related failures and performing network and services supervision.

4 PTP Domains

PTP devices communicating which each other do so using a common PTP domain. This is a user configurable parameter which is part of the header of every PTP message. The IEEE1588 standard explicitly states that a PTP port can operate in one and only one PTP domain at a time. Consequently, the PTP protocol stack running on a PTP port has to check for the domain number of every PTP message and will discard all PTP messages with the incorrect (different from the one the port is configured to operate at) domain number without processing their content any further.

5 PTP States Transitions of a Slave

Figure 2 shows the state diagram of a PTP Slave only device, i.e. a PTP port which will never assume PTP Master role, thus participating in the BMCA passively. This can be accomplished by setting the parameters for the Announce message accordingly for example by setting the Clock Class parameter to its lowest value, namely to 255.

After having initialized all PTP parameters with their default or pre-configured values the port enters the Listening state ready to receive Announce messages. it will remain in that state indefinitely, if no Announce messages are received. This can have several reasons: No PTP Master is present in the network, because all Master capable devices (ports) are either offline, disabled or unreachable. If this can be ruled out, the PTP domain settings should be verified.

As soon as the device selects a Master to synchronize to it will switch to the Uncalibrated state and begin to evaluate the data of the Sync messages while starting to send out Delay_Request messages. If it receives the correct answer from the Master via corresponding Delay_Response messages it finally switches to PTP Slave state, because it is now able to precisely calculate the offset and the transmission delay, allowing it to synchronize its local clock to the Master.

Although considered a transitory state, a PTP port can remain “Uncalibrated”. This indicates that the downstream time transfer is not functioning properly. This could mean either that Delay_Request messages are not forwarded correctly to the Master or not processed by it. Another reason could be that Delay_Response messages are either not transmitted by the Master or it is using an incorrect transport protocol. These messages could get dropped on their way from the Master to the Slave, however this is unlikely to happen and easy to detect using standard network monitoring tools. This may occur due to the Time to Live (TTL) of the IP encapsulated PTP packets causing them to be dropped before reaching their final destination.

The PTP port will remain in the Slave state indefinitely unless it either stops receiving Announce messages or receives Announce messages with different content. In the former case it will switch back to Listening state while the latter will immediately trigger a new BMCA round.

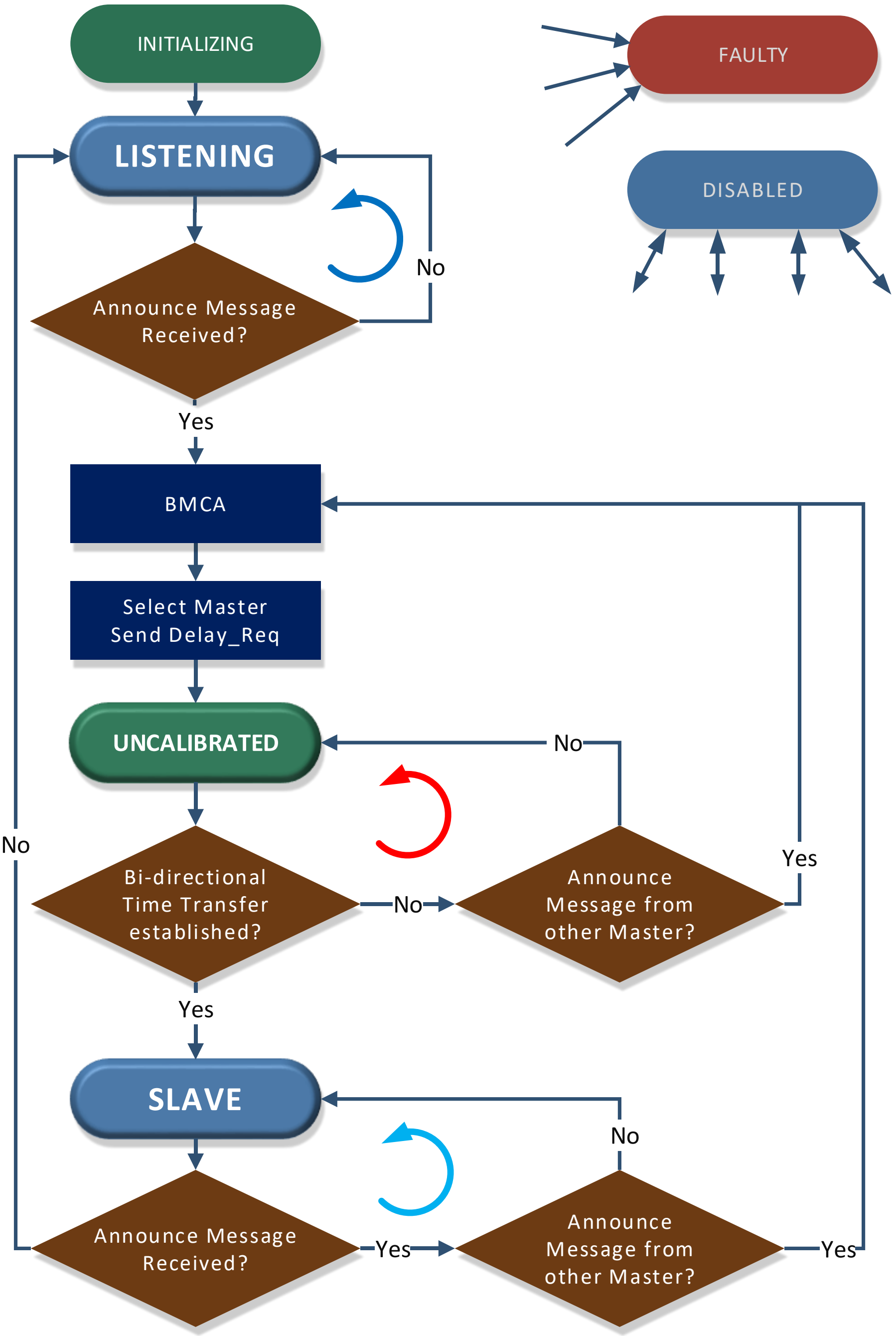

Figure 2: State diagram of a PTP Slave

6 PTP State Transitions of a PTP Grandmaster

Figure 3 shows the state diagram of a PTP port which acts as a PTP Grandmaster. Typically, these are directly connected to a GNSS reference. They either assume the Master role or become Passive rather than switching to Slave state in case a better Master enters the network. In most PTP implementations, more than one PTP GM capable device is deployed, however, from a PTP protocol standpoint, only one can be active while the other(s) are hot-standby devices. As all of them are attached to

GNSS sources, they announce the same clock quality. As a tiebreaker, the BMCA compares their clock Identities (a unique number assigned to every device) selecting the device with the highest clockID.

If a PTP port is in Passive state, it will not transmit messages itself and will not respond to messages other than the PTP management messages. However, it will process all incoming Announce messages according to the rules of the BMCA.

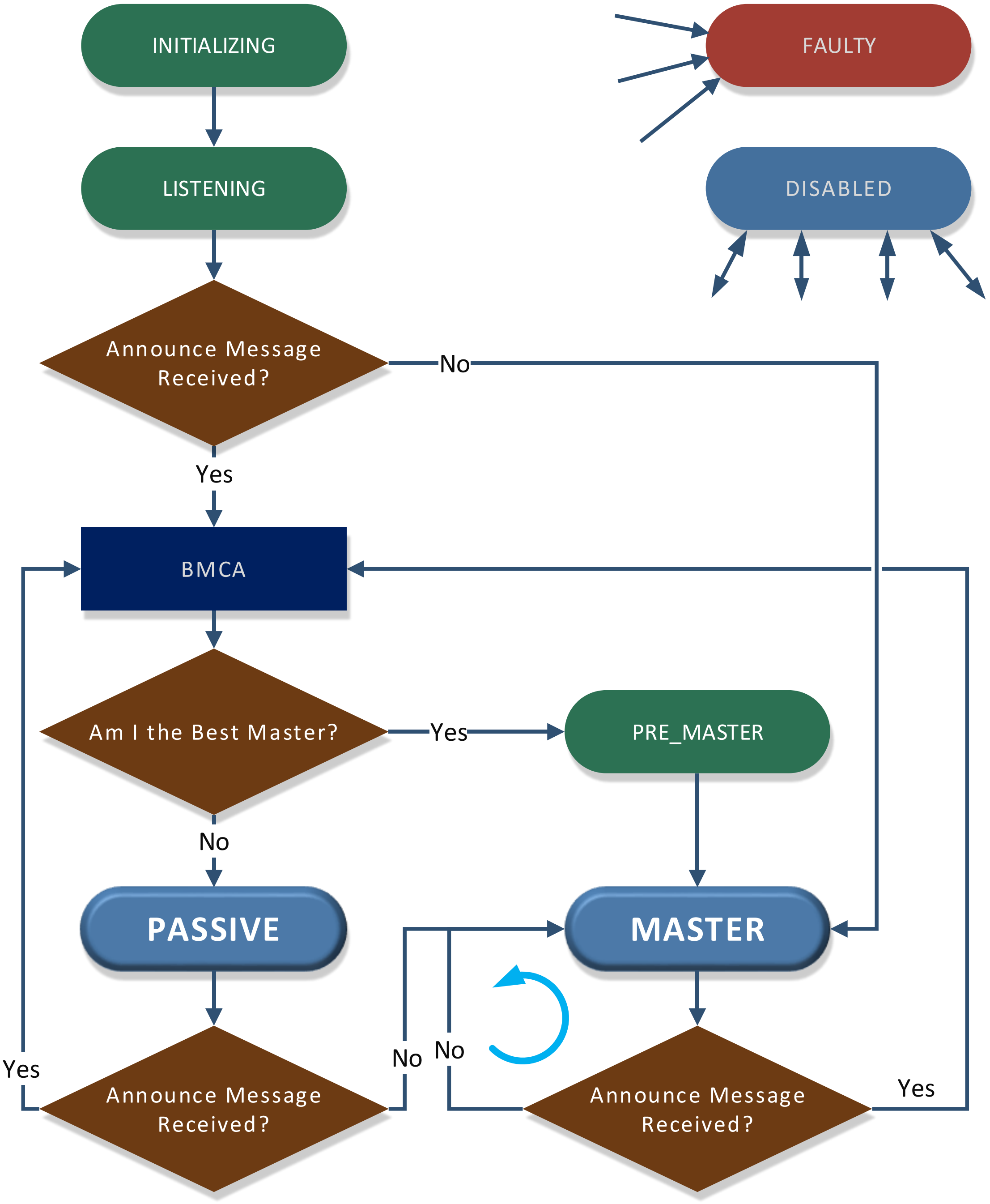

Figure 3: State diagram of a PTP Grandmaster

7 PTP State Transitions of a general PTP port

Finally, the state diagram of a general PTP port is shown in Figure 4. Such a port can assume both Master and Slave role. In case no (better) Master is present it will switch from Listening to Master. If a better Master enters the network, it will revert to Slave state.

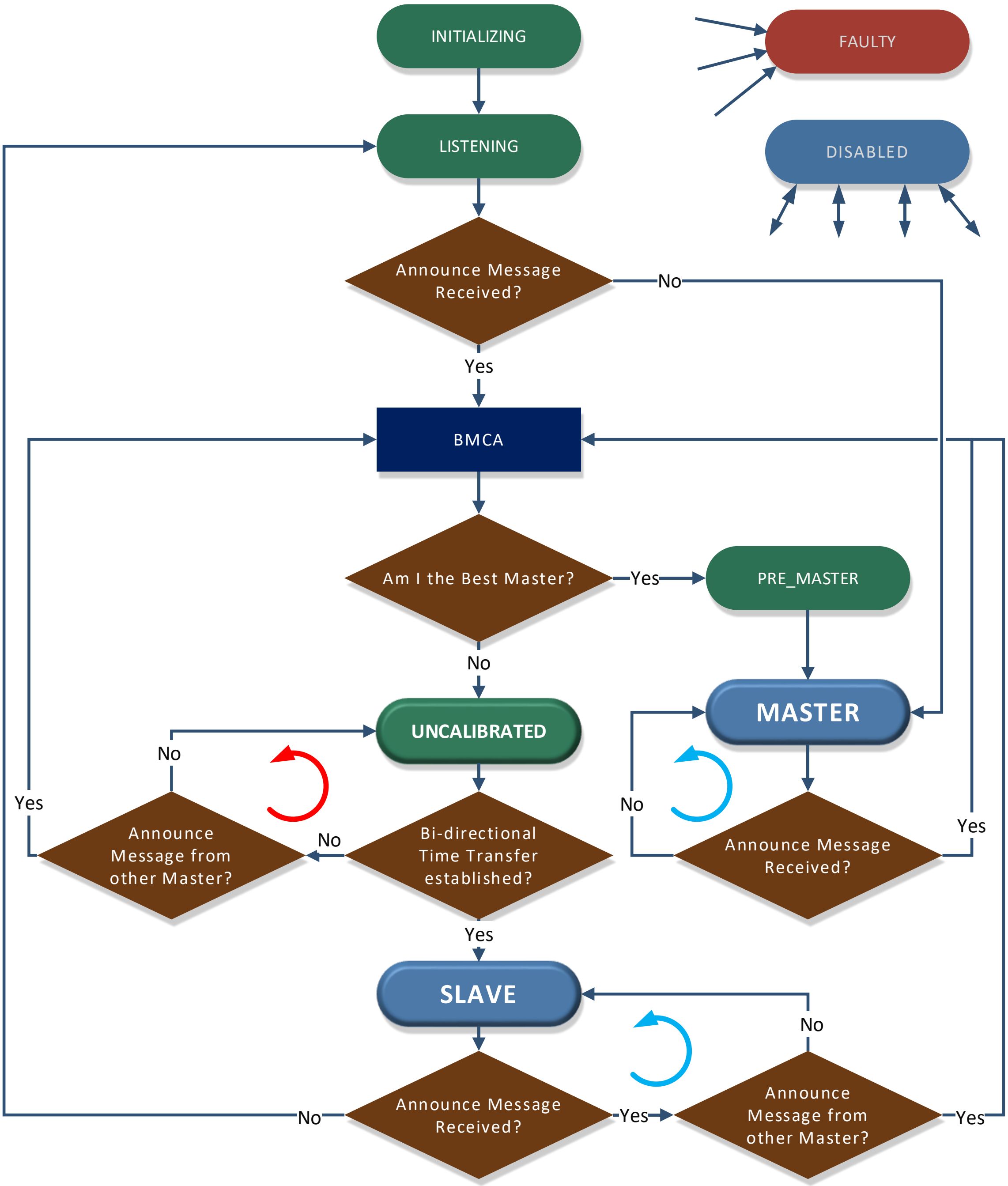

Figure 4: State diagram of a general PTP port

8 Final Remarks on PTP States

In Figures 3 and 4 respectively, a transitional state referred to as Pre-Master is shown. This state has been added in the protocol to avoid BMCAs loops that could occur under certain conditions during Master changes in networks with Boundary Clocks. This state is both entered and left unconditionally, effectively delaying the switching to the Master state.

If the current state of all devices (ports) in a PTP network is monitored, only Slave, Master or Passive should be reported. However, if such a query coincides with the election of a new Master, other states could be observed as well, yet they must be strictly transitory. If this is not the case, the state diagrams can serve as a starting point for an in-depth analysis.

Finally, all three state diagrams show two additional states: Disabled and Faulty. The former indicates that a PTP port has been disabled by the user while the latter marks that a fault condition has occurred. PTP ports in either of these states cease to process any PTP messages other than the PTP Management messages. Thus, their status and configuration can be still monitored, and its parameter set updated externally.

9 Building Blocks of a PTP Software Stack

Aside from hardware modules to be added to every Ethernet port in order to draw sufficiently accurate timestamps, a PTP protocol stack must be executed for every PTP port. Its main tasks are, of course, to process all incoming PTP messages, execute the BMCA and transmit messages in a timely manner. Fortunately, the latter requirement leaves some slack for the implementer, because according to IEEE1588 only the average message rate must be met, rather than requesting a highly constant rate with strict send times. Hence, a real time operating system is not mandatory for PTP to operate.

To comply with the requirements of a specific profile, a PTP stack must be able to establish PTP communication with the respective transport protocol. For SMPTE ST 2059-2 this entails Layer-3 communication via IPv4 or IPv6. Furthermore, specific message types must be sent via multicast (Announce, Sync) while others have to be transmitted either as multicast or unicast (Delay_Request/Response) and finally some are unicast only (Acknowledge to management messages).

In parallel, the PTP stack must continuously re-adjust the local clock to keep its offset as low as possible or at least within the requested limits of the PTP profile. For SMTPE ST 2059-2 this always means remaining below 1µs with respect to the Master. To this end, it must execute a dedicated control loop that modifies the rate of the local clock rather than simply update the absolute time value periodically. The latter approach would result in a non-monotonic time at the Slave with frequent jumps backward in time.

To cope with network impairments that result in Packet Delay Variations, the control loop must be complemented with a series of input filter stages. Besides dampening the network induced noise, they should also detect, and discard packets having faced excessive transmission delays. This technique combines both linear and non-linear filters (the latter relying on statistical data analysis), which has proven to be far superior to even complex linear filters when dealing with arbitrary PDVs, which unfortunately are neither evenly distributed nor follow a Gaussian curve. This is specifically the case if a network with several cascaded switches is overloaded transiently. Even in networks with a partial PTP support, this approach has its merits over basic implementations. An extensive analysis can be found in [4].

Finally, a professional PTP stack should provide efficient means for monitoring important PTP parameters such as actual message rates, number of lost PTP messages, state changes and, of course, the offset to the Master, to name but a few. Details in how to monitor SMPTE ST 2059-2 networks will be covered in the next parts of this series.

10 Improving Accuracy: Transparent and Boundary Clocks

To consistently and reliably reach accuracies of well below 1µs without imposing any constraints on the network load as well as the network topology, deploying PTP-enabled network devices must be considered. IEEE1588 defines two different solutions to cope with PDVs: Transparent and Boundary Clocks. The principal function of both has been described in part 1; here we want to compare them viewed from an end user’s perspective. It must be pointed out that with respect to accuracy both devices yield identical results, if the respective hardware modules (packet scanning and time stamping units) have the same performance.

Transparent Clocks forward PTP messages following the same layer 2 and layer 3 rules as would apply for any other traffic. By measuring the residence time of every PTP event message and inserting this information into the message, they enable PTP Slaves to account for any variation very effectively. They do not participate in the BMCA.

Boundary Clocks, on the other hand, can be considered as active PTP devices. They process all incoming PTP traffic according to the PTP protocol rather than forwarding it according to the respective IP rules, and they do participate in the BMCA. By re-generating the time information provided by the Best Master and redistributing it to all Slaves attached to them, they can be viewed as a means to build a hierarchical time transfer architecture.

Consequently, the overall (network wide) PTP traffic is greatly reduced, since PTP messages are exchanged only between two adjacent devices rather than being transmitted through the complete network, which is the case with Transparent Clocks.

There are, however, a few caveats to consider when choosing Boundary Clocks in preference to Transparent Clocks. Being active PTP devices with multiple PTP ports that all perform the BMCA, they must be monitored far more closely than Transparent Clocks, for whom supervision is fairly straightforward to accomplish.

Furthermore, the behaviour and performance of the control loops implemented in Boundary Clocks should be carefully evaluated during their deployment, especially when large networks are commissioned, where several Boundary Clocks must be cascaded. Of special interest in such cases is their behaviour during a Master failure. Typically, PTP Grandmasters would not be co-located to improve the resilience of the system against malicious attacks. Thus, the time information would suddenly be provided from the "opposite end" of the network, which could very well cause settling effects in the cascaded control loops of the Boundary Clocks to introduce transient noise. A more detailed analysis can be found in [5] and [6],

11 Still to come

The next parts of this series will cover network related aspects with special emphasis on broadcasting requirements, focus on redundancy and highlight some aspects of securing time transfer.

12 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.

[4] N. Kerö, A. Puhm, T. Kernen and A. Mroczkowski, "Performance and Reliability Aspects of Clock Synchronization Techniques for Industrial Automation," in *Proceedings of the IEEE*, vol. 107, no. 6, pp. 1011-1026, June 2019.

[5] N. Kerö, T. Kernen and T. Müller, "PTP Deployment in Large Networks – Traps and Pitfalls," *SMPTE 2014 Annual Technical Conference & Exhibition*, Hollywood, CA, USA, 2014, pp. 1-16.

[6] N. Kerö, T. Kernen, T. Müller and M. Schimandl, "Optimizing Large Media Networks for Highly Accurate Time Transfer via PTP," in SMPTE Motion Imaging Journal, vol. 125, no. 2, pp. 30-44, March 2016.

13 Author(s) biographies

	Thomas Kernen is currently a Staff Architect with Mellanox Technologies Ltd. Prior to joining Mellanox, he spent over 20 years in the IP industry, including driving Cisco's entry into live media production, co-founding Internet Service Providers, Telecom carriers, and architecting Fibre to the Home networks. He has authored over 20 publications in leading journals. He holds six patents that cover both network and video coding optimizations for media transport and delivery. His current interests include defining architectures for transforming the broadcast industry to an All-IP infrastructure. Thomas is also a member of the IEEE Communications and Broadcast Societies. He currently serves as the Co-Chair of the Society of Motion Pictures and Television Engineers (SMPTE) 32NF Committee. He is also a frequent speaker at leading events, such as the SMPTE Annual Technical Conference, the National Association of Broadcasters, IBC, and the European Broadcasting Union Network Technology Seminar.
	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