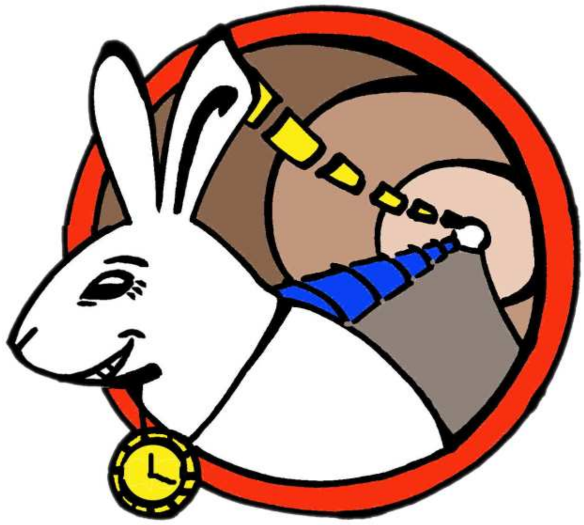
White Rabbit Specification:
Draft for Comments

Emilio G. Cota
Maciej Lipinski
Tomasz Włostowski
Erik van der Bij
Javier Serrano

May 2011



Revision History Table

Version	Date	Authors	Description
0.1	10/09/2010	E.G., M.L.	First draft for comments.
0.2	7/09/2010	T.W., M.L.	Added introduction about PTP.
0.3	8/09/2010	J.S., M.L.	Language&Style-related corrections, reference to Peter's paper.
0.4	14/09/2010	E.V.D.B., M.L.	1. Merged the FSM of the Slave and the FSM of the Master into single and linear FSM, 2. Changed WR managementIDs, 3. Added authors, this rev. table, explanatory figures and descriptions.
0.5	17/09/2010	E.V.D.B., J.S.	1. Added Table of Contents. 2. Re-ordered appendixes. 3. Fixed some typos.
1.0	26/04/2011	M.L.	1. Including feedback from comments. 2. Created WR PTP Profile. 3. Added modification to BMC. 4. Enabled WR Switch to be ”real” Boundary Clock. 5. Added figure clarifying WRPTP vs. PTP FSMs operation from Power On. 6. Mentioned (missing) modification to PTP FSM.
1.1	2/05/2011	J.S.& T.W.& M.L.	Clarity and typos.

Contents

1 Introduction

White Rabbit (WR) is a protocol developed to synchronize nodes in a packet-based network with sub-ns accuracy. The protocol results from the combination of IEEE1588-2008 (PTP)[1] with two further requirements: precise knowledge of the link delay and clock syntonization[1] over the physical layer.

A WR link is formed by a pair of nodes, master and slave (Figure 1). The master node uses a traceable clock to encode data over the physical layer, while the slave recovers this clock (syntonization) and bases its timekeeping on it. Absolute time synchronisation between master and slave is achieved by adjusting the clock phase and offset of the slave to that of the master. The *offset* refers to the clock (e.g. time defined in Coordinated Universal Time or International Atomic Time standards), while the *phase* refers to the clock signal (e.g. 125 MHz clock signal). The phase and offset adjustment is done through the two-way exchange of PTP sync messages, which are corrected to achieve sub-ns accuracy thanks to the precise knowledge of the link delay.

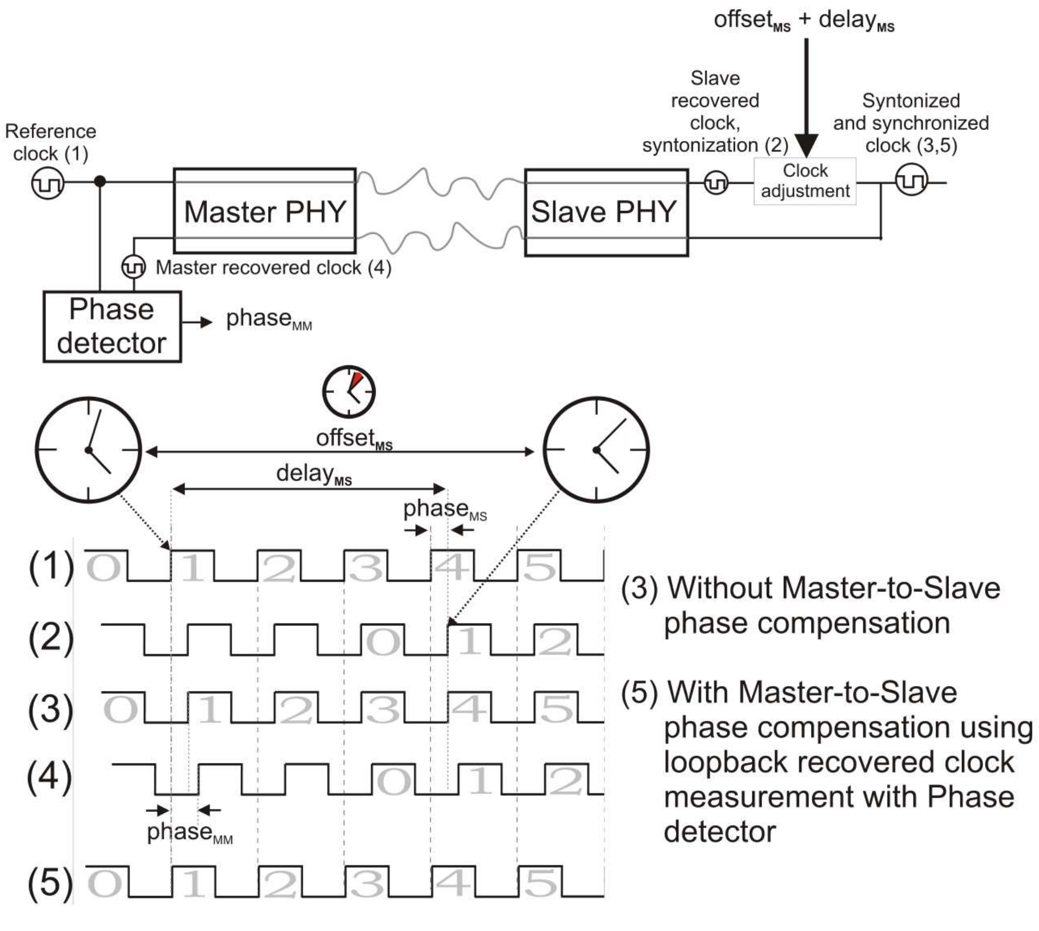

offset$_{MS}$ + delay$_{MS}$
Slave recovered clock, syntonization (2)
Syntonized and synchronized clock (3,5)
Reference clock (1)
Clock adjustment
Master PHY
Slave PHY
Master recovered clock (4)
Phase detector
phase$_{MM}$
offset$_{MS}$
delay$_{MS}$
phase$_{MS}$
(1) 0 1 2 3 4 5
(2) 0 1 2
(3) 0 1 2 3 4 5
(4) 0 1 2
phase$_{MM}$
(5) 0 1 2 3 4 5
(3) Without Master-to-Slave phase compensation
(5) With Master-to-Slave phase compensation using loopback recovered clock measurement with Phase detector

Figure 1: Synchronization and syntonization in a White Rabbit link.

[1] The adjustment of two electronic circuits or devices in terms of frequency.

The precise knowledge of the link delay is obtained by accurate hardware timestamps and calculation of the delay asymmetry (see section 4).

The described single-link synchronization can be replicated. Multi-link WR networks are obtained by chaining WR links forming a hierarchical topology. This hierarchy is imposed by the fact that a frequency traceable to a common grandmaster must be distributed over the physical layer, resulting in a *cascade* of master and slave nodes. As a result of this topology, a WR network consists of two kinds of WR network devices: *WR boundary clocks* (WR Switches) and *WR ordinary clocks* (WR Nodes), see section 6.2.

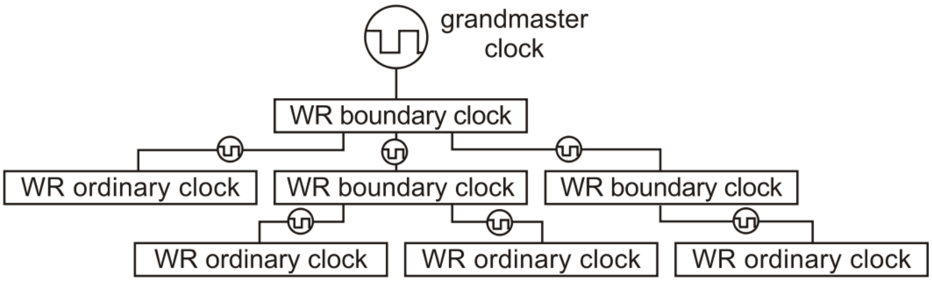


Figure 2: White Rabbit network; it forms a hierarchical topology.

Some applications need WR and IEEE1588-2008 nodes to coexist. Examples of this are networks where the need for highly accurate time synchronisation is concentrated on a certain group of nodes. For this purpose the WR protocol enables WR nodes to defer to IEEE1588 behaviour when not connected to another WR node.

2 Precision Time Protocol

The IEEE1588-2008 standard, known as Precision Time Protocol (PTP), is repeatedly referenced in this document. Knowledge of basic PTP concepts is required to read this specification, therefore they are explained in this section.

PTP is a packet-based protocol designed to synchronize devices in distributed systems. The standard defines two kinds of messages which are exchanged between *PTP nodes*: *event messages* and *general messages*. Both, the time of transmission and the time of reception of event messages are *timestamped*. General messages are used by PTP nodes to identify other PTP nodes, establish clock hierarchy and exchange data, e.g. timestamps, settings or parameters. PTP defines several methods for node synchronization. Figure 3 presents the messages used when the *delay request-response mechanism* (with a *two-step clock*) is used, which is the case in White Rabbit. For simplicity reasons, a PTP node is considered an *ordinary clock* in the remainder of this section; such clocks have only one port.

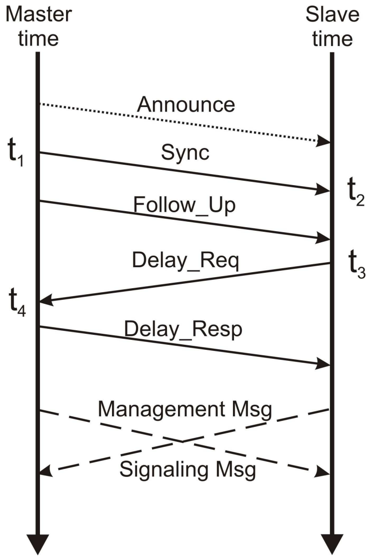


Figure 3: PTP messages used by WRPTP.

An *Announce Message* is periodically broadcast by the PTP node which is in the Master state. The message carries information about its originator and the originator's clock source quality. This enables other PTP nodes receiving the announce message to perform the Best Master Clock (BMC) Algorithm. This algorithm defines the role of each PTP node in the PTP network hierarchy; the outcome of the algorithm is the recommended next state of the PTP node and the node's synchronization source (grandmaster). In other words, a PTP node decides to which other PTP node it should synchronize based on the information provided in the announce messages and using the BMC algorithm. A PTP node which is in the SLAVE state synchronizes to the clock of another PTP node. A PTP node which is in the MASTER state is regarded as a source of synchronization for the other PTP nodes. The full PTP state machine with state descriptions is included in Appendix A.

Sync Messages and *Delay_Req Messages* are timestamped (t_1, t_2, t_3, t_4) and these timestamps are used to calculate the offset and the delay between the nodes exchanging the messages. *Follow_UP Messages* and *Delay_Resp Messages* are used to send timestamps between Master and Slave (in the case of a two-step clock).

Management Messages are used only for configuration and administrative purposes. *Signaling Messages* are used for communication between clocks for optional or experimental features of the PTP standard as well as implementation-specific mechanisms. Both, Management and Signaling messages, are not essential for PTP synchronization.

The flow of events in the PTP delay request-response (two-step clock) mechanism is the following (simplified overview):

1. The master sends Announce messages periodically.
2. The slave receives the Announce message and uses the BMC algorithm to establish its place in the network hierarchy.
3. The master periodically sends a Sync message (timestamped on transmission, t_1) followed by a Follow_UP message which carries t_1.
4. The slave receives the Sync message sent by the master (timestamped on reception, t_2).
5. The slave receives the Follow_Up message (which carries timestamp of Sync transmission time, t_1) sent by the master .
6. The slave sends a Delay_Req message (timestamped on transmission, t_3).
7. The master receives the Delay_Req message sent by the slave (timestamped on reception, t_4).
8. The master sends the Delay_Resp message which carries t_4.
9. The slave receives the Delay_Resp.
10. The slave adjusts its clock using the offset and the delay calculated with timestamps (t_1, t_2, t_3, t_4). It results in the Slave's synchronization with the Master clock.
11. Repeat 1-10.

3 Link Delay Model

The delay of a message travelling from master to slave (see Figure 4) can be expressed as the sum

$$\text{delay}_{ms} = \Delta_{tx_m} + \delta_{ms} + \Delta_{rx_s} \tag{1}$$

where Δ_{tx_m} is the fixed delay due to the master's transmission circuitry, δ_{ms} is the variable delay incurred in the transmission medium and Δ_{rx_s} is the fixed delay due to the slave's reception circuitry. In a similar fashion, the delay of a message travelling from slave to master can be decomposed as

$$\text{delay}_{sm} = \Delta_{tx_s} + \delta_{sm} + \Delta_{rx_m} \tag{2}$$

The characterization of the link is completed with an equation to relate the two variable delays δ_{ms} and δ_{sm}. Describing a procedure to obtain this equation is out of the scope of this document. However, section 3.1 provides such an equation obtained empirically for one scenario.

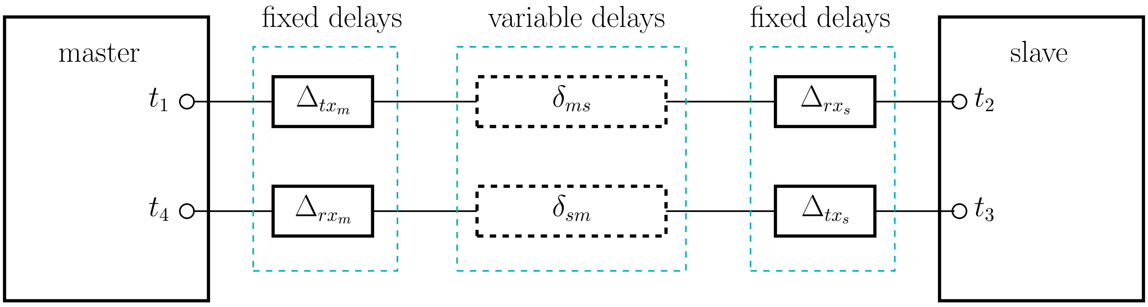


Figure 4: Delay model of a WR link. The timestamps are accurately corrected for link asymmetries by the usage of the four fixed delays $\Delta_{\{tx_m,rx_s,tx_s,rx_m\}}$ and the relationship between both variable delays $\delta_{\{ms,sm\}}$.

3.1 Relative Delay Coefficient

An accurate relation between both variable delays on the transmission line is essential for obtaining an acceptable estimate of the delay asymmetry on a WR link. The relation between δ_{ms} and δ_{sm} is represented in this document by the *relative delay coefficient* (α). Its origin is highly implementation-dependent. Thus this document just assumes that it exists and is known.

3.1.1 1000base-X over a Single-mode Optical Fibre

When a single-mode fibre is used as bi-directional communication medium, it can be shown that both variable delays are related by an equation of the form [3]:

$$\delta_{ms} = (1+\alpha)\,\delta_{sm} \tag{3}$$

4 Delay Asymmetry Calculation

Let us start from the PTP sync timestamps, represented by the familiar set t_1, t_2, t_3 and t_4. The mean path delay is then defined as

$$\mu = \frac{(t_2 - t_1) + (t_4 - t_3)}{2} \tag{4}$$

Note that the transmission delays $t_2 - t_1$ and $t_4 - t_3$ can be expressed in terms of WR's Delay Model:

$$t_2 - t_1 = \Delta_{tx_m} + \delta_{ms} + \Delta_{rx_s} + \text{offset}_{ms} \tag{5}$$

$$t_4 - t_3 = \Delta_{tx_s} + \delta_{sm} + \Delta_{rx_m} - \text{offset}_{ms} \tag{6}$$

where offset_{ms} is the time offset between the slave's clock and the master's. Combining the three equations above we obtain

$$2\mu = \Delta + \delta_{sm} + \delta_{ms} \tag{7}$$

where Δ accounts for all fixed delays in the path, i.e.

$$\Delta = \Delta_{tx_m} + \Delta_{rx_s} + \Delta_{tx_s} + \Delta_{rx_m} \tag{8}$$

The delay asymmetry as specified in section 7.4.2 of the PTP standard is expressed in our own notation by using equations (1), (2) and (7) as follows:

$$\text{delay}_{ms} = \mu + \text{asymmetry} \tag{9}$$

$$\text{delay}_{sm} = \mu - \text{asymmetry} \tag{10}$$

4.1 Solution for Ethernet over a Single-mode Optical Fiber

Combining equations (3) and (7) we obtain:

$$\delta_{ms} = \frac{1+\alpha}{2+\alpha}(2\mu - \Delta) \tag{11}$$

$$\delta_{sm} = \frac{2\mu - \Delta}{2+\alpha} \tag{12}$$

The delay asymmetry can then be derived from equations (1), (9), (11) and (12):

$$\text{asymmetry} = \Delta_{tx_m} + \Delta_{rx_s} - \frac{\Delta - \alpha\mu + \alpha\Delta}{2+\alpha} \tag{13}$$

5 Fixed delays

The knowledge of fixed delays $\Delta_{\{tx_m, rx_s, tx_s, rx_m\}}$ is necessary to calculate delay asymmetry (13). Such delays may be constant for the lifetime of the hardware, its up-time or the duration of the link connection. Therefore, the method for obtaining fixed delays is medium-specific and implementation-dependent. The delays are measured (if necessary) and their values are distributed across the link during the process of establishing the WR link, which is called *WR Link Setup* in this document. A WR node participates in the measurement of another WR node's reception fixed delay ($\Delta_{\{rx_m, rx_s\}}$) upon request, e.g. by sending a calibration pattern in Gigabit Ethernet. Measurement of fixed delays during WR Link Setup is optional and in principle needed only once, while the WR link is being set up. It is only required if non-deterministic reception/transmission elements are used

The calibration pattern sent by WR nodes to measure fixed delays for non-deterministic Gigabit Ethernet PHY shall be generated by repeated transmission of RD+ K28.7 code-group as described in Appendix B.

6 White Rabbit PTP

White Rabbit extends the IEEE1588-2008 (PTP) standard to achieve sub-ns accuracy while still benefiting from PTP's synchronization, management and messaging mechanisms. From now on in this document, the White Rabbit extension to the PTP standard will be referred to as *WRPTP*.

WRPTP takes advantage of PTP's customization facilities, i.e. *PTP profiles* and *Type-Length-Value* (TLV), and makes an extension to the PTP standard which is out of the scope of these facilities. It also defines *implementation specific* functionalities (e.g. WR-specific fields of Data Sets, hardware support) which are compatible with the PTP standard. For the sake of simplicity and clarity, this section explains all the mechanisms of WRPTP without classifying them into PTP-profile, PTP-extension and implementation-specific. The distinction is made in the last subsections, where the *White Rabbit PTP profile* is defined and the extensions to PTP are listed.

6.1 Overview

WRPTP introduces *the White Rabbit Link Setup* (WR Link Setup), which is a process for establishing the WR link (Figure 6). It includes syntonization of the local clock over the physical layer, measurement of fixed delays (calibration) and distribution of the information about fixed delays over the link. WRPTP uses the Link Delay Model to obtain an accurate delay estimation, e.g. it uses the delay asymmetry equation (13) for Gigabit Ethernet over optical fiber. The additional communication between a master and a slave needed to exchange parameters and state information is done through extended PTP messaging facilities.

WRPTP is compliant with standard PTP. Figure 5 depicts the tree topology of a *hybrid* WR/IEEE1588 network with a grandmaster as a root.

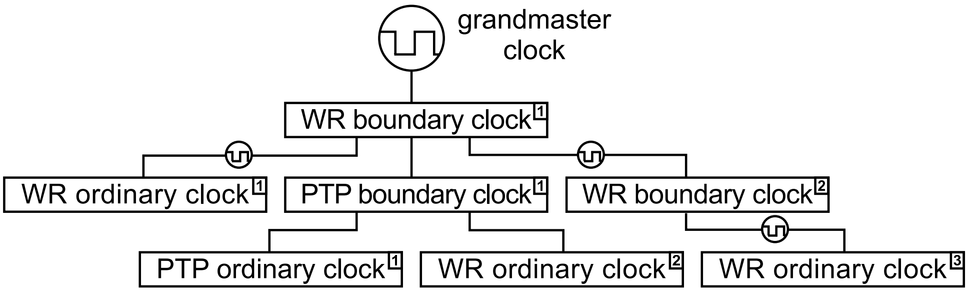


Figure 5: Hybrid WR/IEEE1588 network. White Rabbit nodes work transparently with PTP nodes. WR ordinary clock 3 is more accurately synchronised to the grandmaster than WR ordinary clock 2, which is below a PTP boundary clock.

The flow of events for standard PTP which is presented in section 2 is extended as depicted in Figure 6 and described below:

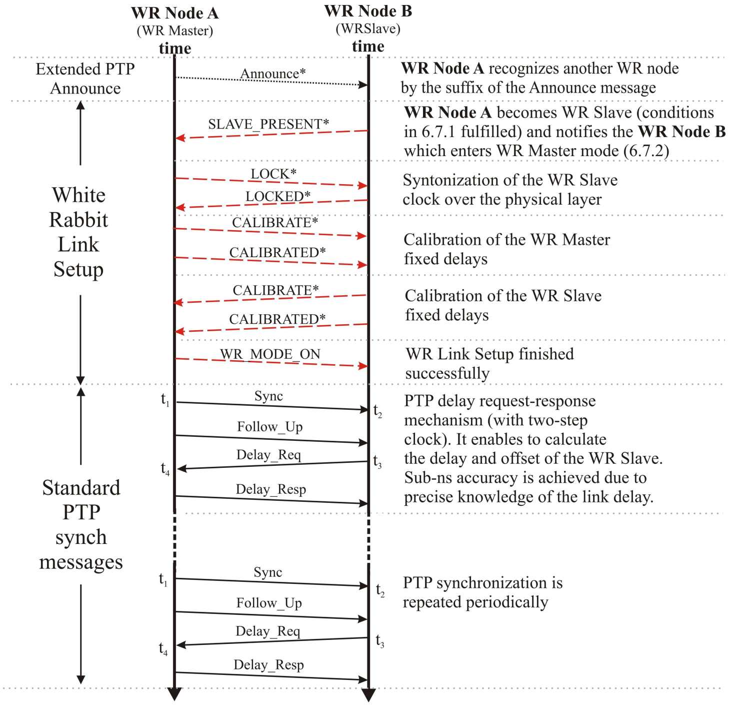

WR Node A
(WR Master)
time
WR Node B
(WRSlave)
time
Extended PTP Announce
Announce*
WR Node A recognizes another WR node by the suffix of the Announce message
SLAVE_PRESENT*
WR Node A becomes WR Slave (conditions in 6.7.1 fulfilled) and notifies the WR Node B which enters WR Master mode (6.7.2)
White Rabbit Link Setup
LOCK*
LOCKED*
Syntonization of the WR Slave clock over the physical layer
CALIBRATE*
CALIBRATED*
Calibration of the WR Master fixed delays
CALIBRATE*
CALIBRATED*
Calibration of the WR Slave fixed delays
WR_MODE_ON
WR Link Setup finished successfully
t1
Sync
t2
Follow_Up
Delay_Req
t4
t3
Delay_Resp
PTP delay request-response mechanism (with two-step clock). It enables to calculate the delay and offset of the WR Slave. Sub-ns accuracy is achieved due to precise knowledge of the link delay.
Standard PTP synch messages
t1
Sync
t2
Follow_Up
Delay_Req
t4
t3
Delay_Resp
PTP synchronization is repeated periodically

Figure 6: Simplified overview of the message flow in WRPTP.

1. The **WR Node A** which is in PTP_MASTER state periodically sends WR Announce messages with a custom suffix.

2. The **WR Node B** receives Announce message(s), recognizes the WR Announce message and uses the modified BMC algorithm (section 6.4) to establish its place in the WR network hierarchy.

3. The **WR Node B** enters WR Slave mode (based on the conditions in 6.7.1) and starts the WR Link Setup by sending the SLAVE_PRESENT WR Signaling message.

4. The **WR Node A** enters WR Master mode (based on the conditions in 6.7.2) and sends the LOCK WR Signaling message to request the WR Slave to start syntonization.

5. The WR Slave sends the LOCKED WR Signaling message as soon as the syntonization process is finished (notification from the hardware).

6. The WR Master sends the CALIBRATE WR Signaling message to request calibration of its reception fixed delay.

7. The WR Master sends the CALIBRATED WR Signaling message as soon as the calibration is finished (notification from hardware).

8. The WR Slave sends the CALIBRATE WR Signaling message to request calibration of its reception fixed delay.

9. The WR Slave sends the CALIBRATED WR Signaling message as soon as the calibration is finished (notification from hardware).

10. The WR Master sends the WR_MODE_ON WR Signaling message to indicate completion of the WR Link Setup process.

11. The WR Master periodically sends a Sync message (timestamped on transmission, t_1) followed by a Follow_UP message which carries t_1.

12. The WR Slave receives the Sync message sent by the master (timestamped on reception, t_2).

13. The WR Slave receives the Follow_Up message sent by the master.

14. The WR Slave sends a Delay_Req message (timestamped on transmission, t_3).

15. The WR Master receives the Delay_Req message sent by the master (timestamped on reception, t_4).

16. The WR Master sends a Delay_Resp message which carries t_4.

17. The WR Slave receives the Delay_Resp.

18. The WR Slave adjusts its clock using the offset and the delay calculated with the timestamps (t_1, t_2, t_3, t_4) corrected due to the precise knowledge of the link delay. It results in the Slave's synchronization with the Master clock with sub-ns accuracy.

19. Repeat 1, 11-18.

6.2 Definitions

The following definitions used in this document are introduced in Clause 3.1 of PTP:
node: A device that can issue or receive Precision Time Protocol (PTP) communications on a network.
boundary clock: A clock that has multiple Precision Time Protocol (PTP) ports in a domain and maintains the timescale used in the domain. It may serve as the source of time, i.e., be a master clock, and may synchronize to another clock, i.e., be a slave clock.
ordinary clock: A clock that has a single Precision Time Protocol (PTP) port in a domain and maintains the timescale used in the domain. It may serve as a source of time, i.e., be a master clock, or may synchronize to another clock, i.e., be a slave clock.

This document introduces the following definitions:
White Rabbit Port (WR Port): a WRPTP-enabled port of a boundary clock or an ordinary clock.
White Rabbit Master (WR Master): a WRPTP-enabled port of a boundary clock or an ordinary clock which acts as a source of timing information for another White Rabbit enabled port of a boundary or an ordinary clock.
White Rabbit Slave (WR Slave): a WRPTP-enabled port of a boundary clock or an ordinary clock which retrieves timing information sent over a link by a WR Master.
White Rabbit Switch (WR Switch): a boundary clock implementing WRPTP, compatible with standard PTP.
White Rabbit Node (WR Node): an ordinary clock implementing WRPTP, compatible with

standard PTP.

6.3 WRPTP Data Set Fields

The PTP standard defines data sets (DS) to store the static and dynamic variables needed for the operation of the protocol (section 8, PTP). WRPTP:

- adds fields to the DSs defined in the PTP standard to store the WR-specific parameters,
- defines a new DS: backupParentDS.

6.3.1 New fields of PTP Data Sets

Table 1 defines the additional DS fields required by WRPTP that are not part of the PTP standard, described below.

6.3.1.1 wrConfig

Determines the predefined function of a WR port.

6.3.1.2 wrMode

Contains the current WR-role of a WR port which is determined using the modified BMC algorithm (section 6.4) by the conditions described in the sections 6.7.1 and 6.7.2.

6.3.1.3 wrModeON

If true, it indicates that the WR Link Setup has been performed successfully and the WR port is synchronized using WRPTP, and that the role defined in wrMode is active.

6.3.1.4 wrPortState

Stores the current state of the WRPTP state machine (see chapter 6.7).

6.3.1.5 calibrated

If true, it indicates that the fixed delays of the given port are known.

6.3.1.6 deltaTx

Port’s Δ_{tx} measured in picoseconds and multiplied by 2^{16}.

6.3.1.7 deltaRx

Port’s Δ_{rx} measured in picoseconds and multiplied by 2^{16}.

6.3.1.8 calPeriod

Calibration period in microseconds.

6.3.1.9 primarySlavePortNumber

If 0, no Primary Slave is selected. 1, 2 ,...N –value of the portNumber (clause 7.5.2.3 PTP) selected as the Primary Slave, see section 6.4.

Table 1: WRPTP Data Sets fields

DS member	**DS name**	**Values**	**D**ynamic or **S**tatic
wrConfig	portDS	NON_WR, WR_S_ONLY, WR_M_ONLY WR_M_AND_S	S
wrMode	portDS	NON_WR, WR_SLAVE, WR_MASTER	D
wrModeON	portDS	TRUE, FALSE	D
wrPortState	portDS	Table 13	D
calibrated	portDS	TRUE, FALSE	D
deltaTx	portDS	64 bit value	D
deltaRx	portDS	64 bit value	D
calPeriod	portDS	32 bit value	S
parentWrConfig	portDS	NON_WR, WR_S_ONLY, WR_M_ONLY WR_M_AND_S	D
parentWrMode	portDS	NON_WR, WR_SLAVE, WR_MASTER	D
parentWrModeON	portDS	TRUE, FALSE	D
parentDeltaTx	portDS	64 bit value	D
parentDeltaRx	portDS	64 bit value	D
parentCalPeriod	portDS	32 bit value	S
primarySlavePortNumber	currentDS	16 bits value	D

6.3.1.10 parentWrConfig

Stores the value of wrConfig of the parent port.

6.3.1.11 parentWrMode

Stores the value of wrMode of the parent port.

6.3.1.12 parentWrModeON

Stores the value of wrModeON of the parent port.

6.3.1.13 parentCalibrated

Stores the value of calibrated of the parent port.

6.3.1.14 parentDeltaTx

Stores the value of deltaTx of the parent port.

6.3.1.15 parentDeltaRx

Stores the value of deltaRx of the parent port.

6.3.1.16 parentCalPeriod

Stores the value of calPeriod parameter of the parent port.

6.3.2 backupParentDS data set specifications

A boundary clock shall maintain an implementation-specific backupParentDS data set for the purpose of qualifying redundant sources of synchronisation, i.e.:

- backup paths to the current grandmaster clock, or
- paths to alternate grandmaster clock(s).

Each entry of the data set contains:

- backupParentDS.secondarySlavePortNumber - the number of the port on which the Announce message from backup parent has been received (Secondary Slave port),
- backupParentDS.backupParentSourcePortIdentity - the portIdentity of the port on the backup master.

The implementation-specific backupParentDS set shall have a minimum capacity of three backup parent records. The order of the records is established using Data Set Comparison Algorithm as described in the section 6.4.2.

6.4 Modified Best Master Clock Algorithm

6.4.1 Overview

The Best Master Clock (BMC) algorithm is used in PTP to compare clocks (determine which clock is the *best*) and to recommend the next state of the PTP state machine (Clause 9.3 of PTP). The BMC is used to prune the physical network topology to obtain logical tree(s) with the *best* clock, the grandmaster (preferably the one synchronized to a primary reference time source), as a root. In the case when more then one clock in a network is synchronized to a primary reference time source, the BMC produces disjoint logic trees. Each tree has a single grandmaster.

As a result of BMC, no more than one port of a Boundary Clock can be in the PTP_SLAVE state. Such a solution allows for redundancy of the time source and topology but is not optimal for the continuity of the synchronization. For example, in case of a failure of one of the *best* clocks (reference-synchronized grandmasters), all the slave nodes in its logic tree need to switch synchronization to the alternate grandmaster. This requires the restart of the estimation of the clock drift, mean path delay and offset which might cause fluctuations in the notion of time.

The modified BMC allows for more than one *best* clock in a single domain, enabling the creation of a logic topology with multiple roots. A Boundary Clock running the modified BMC can have more than one port in the PTP_SLAVE state. This means that timing information is exchanged between the Boundary Clock and more than one source of time (Ordinary Clock or Boundary Clock). At any time any of these links can be used to perform synchronization, including a weighted average from all SLAVE ports as mentioned in [4].

The BMC algorithm includes the State Decision Algorithm (SDA) and the Data Set Comparison Algorithm (DCA) and defines which fields of Data Sets should be updated depending on the outcome of the SDA. The modifications to the BMC are detailed below. the DCA is not modified, therefore it is not mentioned.

6.4.2 State Decision Algorithm (SDA)

The original SDA is depicted in Figure 26 of the PTP standard. The modified SDA, depicted in Figure 7, enforces *BMC_SLAVE* state instead of *BMC_PASSIVE* state. A port being in a *PTP_SLAVE* state as a result of SDA modification (block-8 and block-13 of Figure 7) is considered a Secondary SLAVE port. The port which enters the *PTP_SLAVE* state based on the decision at block-11, is considered the Primary SLAVE port. It means that:

- the Primary SLAVE port and the Secondary SLAVE port(s) are synchronized to the same grandmaster clock but the connection of the Primary SLAVE port is possibly better by path, or

- the Primary SLAVE port and the Secondary SLAVE port(s) are synchronized to different grandmaster clocks of the same clockClass range (1-127 or 128-255,see Table 5 of PTP) but the connection of the Primary SLAVE port is possibly better by path.

The best qualified Announce messages (E_{rbest}, see Clause 9.3.2.3 of PTP) from all Secondary SLAVE ports shall be compared with the DCA to determine the ”second best master” and the lower order masters. The sequence of events is as follows (the idea originated from [4]):

- The best master is computed with the BMC – the Primary SLAVE port established (block-11 of Figure 7) and parentDS data set updated (S1), the port's number is stored in currentDS.primarySlavePortNumber.

- If the recommended state is BMC_SLAVE at block-8 or block-13, the master is considered a second best master and its data is stored in E_{srbest}.

- The BMC algorithm is executed on all the E_{srbest}.

- The best master selected is considered second best master.

- The BMC is repeated excluding E_{srbest} of the best master until the set of E_{srbest} is empty.

The order of second and lower level best masters determined in this way is used to update the backupParentDS data set.

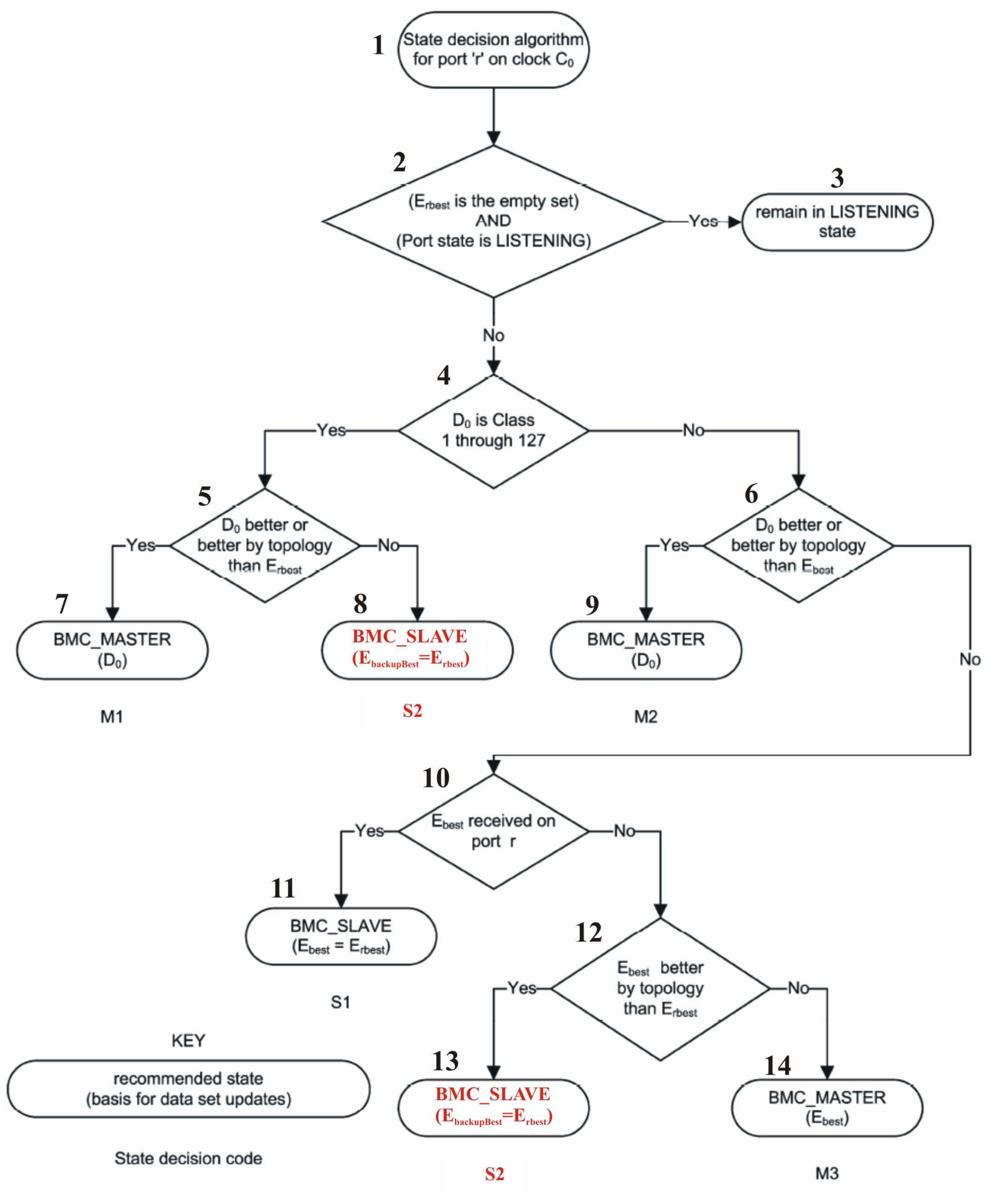

1
State decision algorithm for port 'r' on clock C_0
2
(E_{rbest} is the empty set) AND (Port state is LISTENING)
Yes
3
remain in LISTENING state
No
4
D_0 is Class 1 through 127
Yes
No
5
D_0 better or better by topology than E_{rbest}
Yes
No
6
D_0 better or better by topology than E_{best}
Yes
No
7
BMC_MASTER (D_0)
M1
8
BMC_SLAVE ($E_{backupBest}$=E_{rbest})
S2
9
BMC_MASTER (D_0)
M2
10
E_{best} received on port r
Yes
No
11
BMC_SLAVE (E_{best} = E_{rbest})
S1
12
E_{best} better by topology than E_{rbest}
Yes
No
13
BMC_SLAVE ($E_{backupBest}$=E_{rbest})
S2
14
BMC_MASTER (E_{best})
M3
KEY
recommended state (basis for data set updates)
State decision code

Figure 7: Modified State Decision Algorithm (modifications in red).

6.4.3 Update of Data Sets

The modifications to the rules governing the update of the data sets are twofold:

- Modifications to already existing ”update tables” (Table 13 and Table 16, clause 9.3 of PTP) to accommodate the new DS fields.

- Addition of a new ”update table” to accommodate the new backupParentDS data set and the modifications in the SDA.

Table 2: Modification to Table 13 of IEEE1588-2008: Update for state decision code M1 and M2.

Update this field	From the indicated field of the defaultDS data set of the clock unless otherwise stated
currentDS	
currentDS.primarySlavePortNumber	set to 0

Table 3: Modification to Table 16 of IEEE1588-2008: Update for state decision code S1.

Update this field	From the indicated source
currentDS	
currentDS.primarySlavePortNumber	portDS.portIdentity.portNumber

Table 4: Update for state decision code S2

Update this field	From the indicated source
backupParentDS	
backupParentDS.secondarySlavePortNumber	portDS.portIdentity.portNumber
backupParentDS.backupParentSourcePortIdentity	sourcePortIdentity of E_{rbest}

6.5 WRPTP Messages

White Rabbit benefits from PTP's messaging facilities. It defines a *WR Type-Length-Value* (WR TLV) extension to exchange WR-specific information and uses the two-step clock delay request-response mechanism for synchronization. In particular, it adds a suffix to the Announce message to enable recognition of WR nodes and uses Signaling Messages to exchange WR-specific information (Figure 6).

A WR port in the PTP_MASTER state announces its presence by adding a suffix to the Announce message. The suffix is defined by the PTP standard as a set of TLV entities (section 13.4, PTP). Unrecognised TLVs are ignored by standard PTP nodes (section 14.1, PTP), but read and interpreted by White Rabbit nodes. The information provided in the WRPTP Announce message is sufficient for another WR port to decide whether the WR link can be established and maintained. A WR port receiving a WRPTP Announce message enters (if conditions in 6.7.1 are fulfilled) the WR Slave mode and starts the Link Setup process (performed during $PTP\_UNCALIBRATED$ state. see Appendix D) It requests the sender of the WRPTP announce message to enter the WR Master mode (see section 6.7.2) and start the Link Setup process. During the WR Link Setup, communication between the WR Master and the WR Slave is performed using PTP Signaling Messages carrying WR TLVs. Once the WR link has been established, the WR nodes use a PTP delay request-response mechanism (section 11.3, PTP).

6.5.1 WR Type-Length-Value

All PTP messages can be extended by means of a standard *type*, *length*, *value* (TLV) extension mechanism. White Rabbit uses the *ORGANIZATION_EXTENSION* TLV type (tlvType=0x0003) which is defined in clause 14.3 of PTP. TLVs represented by this type are designated for vendors and standard organizations to extend the protocol for specific needs. The organization-specific TLV fields for WR extension are defined and described in Table 5.

Table 5: Organization specific TLV fields for WR (see Table 35, PTP).

Bits								**Octets**	**TLV**	**WR TLV**	
7	6	5	4	3	2	1	0		**Offset**	**Content**	**Description**
tlvType								2	0	0x0003	Organization extension, see Table 34 PTP.
lengthField								2	2	8+N	N is an even number of wrDataField octets.
OrganizatinId								3	4	0x080030	OUI owned by CERN.
organizationSubType								2	7	magicNumber	e.g.: 0xABCD - identifies WRPTP within protocols identified by CERN's OUI.
								1	9	versionNumber	0x01 - WRPTP version.
DataField								2	10	wrMessageID	WR-specific messages identifier, see Table 6.
								N	12	wrDataField	Content of WR-specific message.

The WR-specific TLVs are identified by wrMessageID which is defined in Table 6. The different types of WR messages are described in subsequent sections.

Table 6: White Rabbit Message ID values

managementId name	**wrMessageId value (hex)**	**Sent in message type**
SLAVE_PRESENT	0x1000	Signaling
LOCK	0x1001	Signaling
LOCKED	0x1002	Signaling
CALIBRATE	0x1003	Signaling
CALIBRATED	0x1004	Signaling
WR_MODE_ON	0x1005	Signaling
ANN_SUFIX	0x2000	Announce

6.5.2 WRPTP Announce Message

The standard PTP Announce Message is suffixed by one WR TLV entity. The WRPTP Announce message has the structure defined in Table 7. The *dataField* of the suffix WR TLV stores the $wrFlags$ defined in Table 8.

Table 7: White Rabbit Announce Message.

Bits								**Octets**	**TLV Offset**	**Content**
7	6	5	4	3	2	1	0			
header								34	0	section 13.3, PTP.
body								30	34	section 13.5, PTP.
tlvType								2	64	0x0003, see 6.5.1.
lengthField								2	66	0xA.
OrganizationId								3	68	0x080030.
magicNumber								2	71	0xABCD.
versionNumber								1	73	0x01.
wrMessageId								2	74	0x0010.
wrFlags								2	76	see Table 8.

Table 8: White Rabbit flags.

Octet	**Bit**	**Message type**	**Name**	**Description**
0	0	Announce	wrConfig	The value of wrConfig parameter from portDS.
0	1			
0	2	Announce	calibrated	TRUE if the source port is calibrated.
0	3	Announce	wrModeON	The value of wrModeON parameter of portDS.

6.5.3 WRPTP Signaling Messages

White Rabbit uses Signaling Messages (defined in clause 13.12 of PTP) to exchange WR-specific information, except for the flags transported in the suffix of the Announce message.

The Signaling Messages conform to the format presented in Table 9. Each WR Signaling Message transports a single WR TLV structure as defined in Section 6.5.1. The *targetPortIdentity* shall be always set to the *clockIdentity* of the port on the other side of the link (conveyed in the Announce Message).

Signaling Messages are used in WRPTP to trigger transitions in the WR state machine and exchange WR-specific parameters. They are distinguished by the *wrMessageId* field of *WR TLV*, defined in Table 6. Signaling Messages are exchanged only within a single link connection (no forwarding) which is ensured by setting the targetPortId appropriately. The rest of this subsection describes the WRPTP Signaling Messages in details.

Table 9: PTP Signaling message fields (Table 33, PTP)

Bits								**Octets**	**TLV Offset**
7	6	5	4	3	2	1	0		
header								34	0
targetPortIdentity								10	34
WR TLV								M	44

6.5.3.1 SLAVE_PRESENT

Message sent by the WR Node which entered WR Slave mode to the WR Node to become the WR Master. It initiates the WR Master mode and starts WR Link Setup process in the WR Master. The message shall have the form specified in Table 10.

6.5.3.2 LOCK

Message sent by the WR Master to the WR Slave to request the start of frequency locking. The message shall have the form specified in Table 10.

6.5.3.3 LOCKED

Message sent by the WR Slave to the WR Master. It indicates successful completion of frequency locking. The message shall have the format specified in Table 10.

6.5.3.4 CALIBRATE

Message sent by the WR port entering the REQ_CALIBRATION state (see section 6.7). It informs the other port whether sending a calibration pattern (see section 5) is required (defined by the value of *calibrationSendPattern* flag). If calibration is required, it carries the calibration period. The message format and parameters are described in Table 11.

Table 10: WR TLV for SLAVE_PRESENT, LOCK, LOCKED AND WR_MODE_ON Signaling Messages.

Bits								Octets	TLV	Content
7	6	5	4	3	2	1	0		Offset	
tlvType								2	0	0x0003, see 6.5.1.
lengthField								2	2	0x8.
OrganizationId								3	4	0x080030.
magicNumber								2	7	0xABCD.
versionNumber								1	9	0x01.
wrMessageId								2	10	Defined in Table 6.

Table 11: WR TLV CALIBRATE Signaling Message

Bits								Octets	TLV	Content
7	6	5	4	3	2	1	0		Offset	
tlvType								2	0	0x0003, see 6.5.1.
lengthField								2	2	0x14.
OrganizationId								3	4	0x080030.
magicNumber								2	7	0xABCD.
versionNumber								1	9	0x01.
wrMessageId								2	10	CALIBRATE.
calibrationSendPattern								2	12	The value determines whether calibration pattern should be sent. If the value is 0x1, the calibration pattern is sent. If the value is 0x0, the calibration pattern is not sent.
calibrationPeriod								4	14	The value defines the time (in microseconds) for which the calibration pattern should be sent by the receiving node.

6.5.3.5 CALIBRATED

Message sent by a WR port entering the *CALIBRATED* state. If preceded by the *CALIBRATE* message with *calibrationSendPattern* set to TRUE, it indicates successful completion of the calibration. The message provides the other node with the values of its fixed delays (Δ_{tx} and Δ_{rx}). The message shall have the format specified in Table 12.

6.5.3.6 WR_MODE_ON

Message sent by WR Master to WR Slave. It indicates successful completion of the WR Link Setup process and requests the WR Slave to validate its WR mode (by setting wrModeON to true). The message shall have the format specified in Table 10.

Table 12: WR TLV for CALIBRATED Signaling Message.

Bits								**Octets**	**TLV**	
7	6	5	4	3	2	1	0		**Offset**	**Content**
tlvType								2	0	0x0003, see 6.5.1.
lengthField								2	2	0x28.
OrganizationId								3	4	0x080030.
magicNumber								2	7	0xABCD.
versionNumber								1	9	0x01.
wrMessageId								2	10	CALIBRATED.
deltaTx								16	12	The value of Δ_{tx_m} measured in picoseconds and multiplied by 2^{16}.
deltaRx								16	28	The value of Δ_{rx_m} measured in picoseconds and multiplied by 2^{16}.

6.6 PTP State Machine

The WR Link Setup (WR state machine) is performed in the PTP_UNCALIBRATED state of the PTP state machine (see Appendix D). Therefore it is essential for a WR port to implement this state as specified in the PTP state machine (2 in Figure 8): a transition state between the LISTENING or PRE_MASTER or MASTER or PASSIVE state and the SLAVE state.

WRPTP specifies an implementation-specific SYNCHRONIZATION_FAULT PTP state transition event (3 in Figure 8, defined in Clause 9.2.6.12 of PTP). The port being in WR_Slave mode and PTP_SLAVE state shall evaluate values of two WR-specific parameters: *wrModeON* and *parentWrModeON*. If the value of at least one of the parameters is FALSE, the SYNCHRONIZATION_FAULT event shall occur and PTP_UNCALIBRATED shall be entered. Consequently the WR Link Setup is performed. Such implementation enables forced re-calibration of WR ports which can be triggered by both, WR Master and WR Slave.

WRPTP introduces a new state transition event to the original PTP state machine: *CALIBRATION_FORCED_BY_SLAVE*. It is depicted in red (number 2) in Figure 8. The event shall be instantiated by the reception of the *SLAVE_PRESENT* Signaling Message on a WR-Master-enabled port (the value of portDS.wrConfig is WR_M_AND_S or WR_M_ONLY) in *PTP_MASTER* state. This transition allows a WR Slave to start WR Master mode and trigger a WR Link Setup process in another WR port.

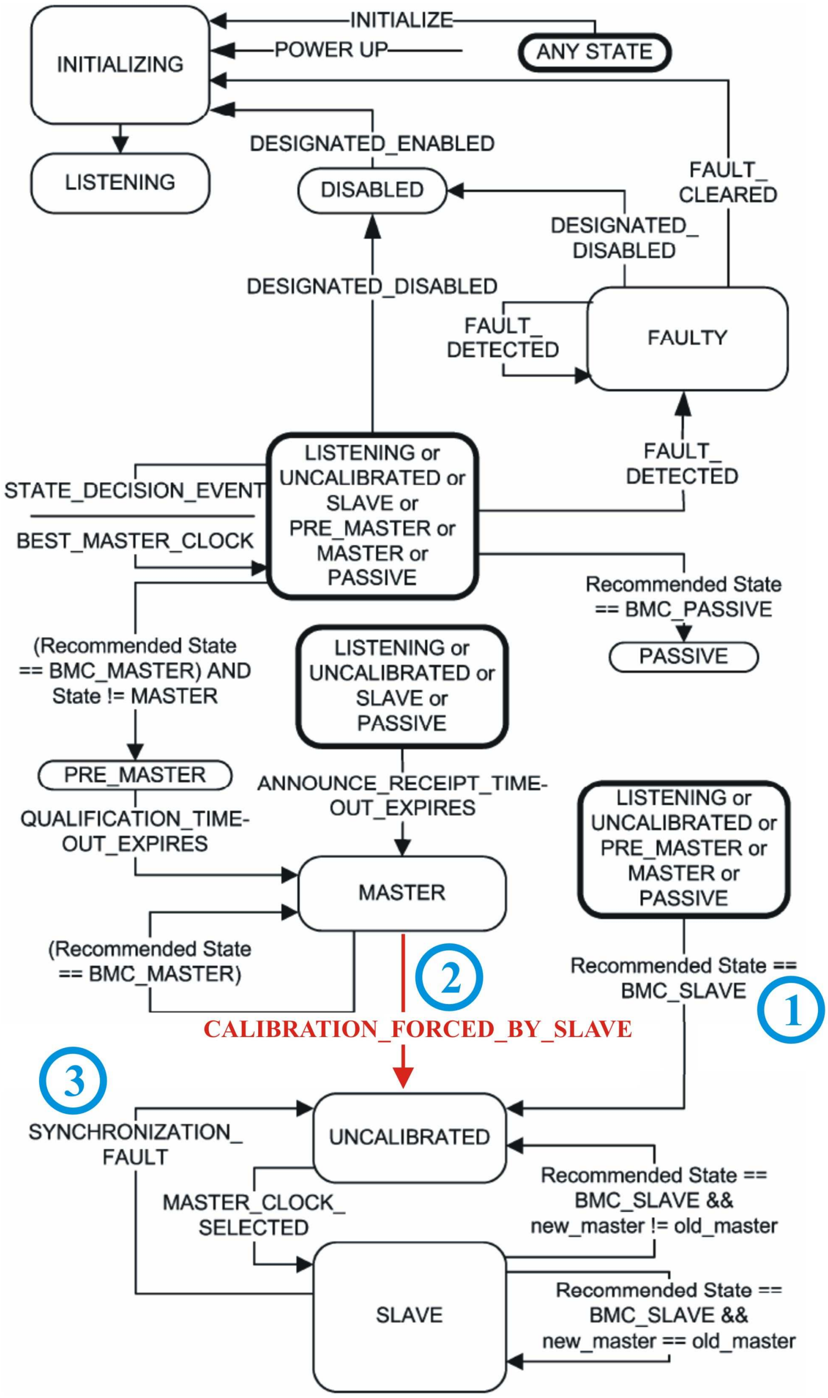

INITIALIZE
POWER UP
ANY STATE
INITIALIZING
LISTENING
DESIGNATED_ENABLED
DISABLED
FAULT_ CLEARED
DESIGNATED_ DISABLED
DESIGNATED_DISABLED
FAULT_ DETECTED
FAULTY
STATE_DECISION_EVENT
BEST_MASTER_CLOCK
LISTENING or UNCALIBRATED or SLAVE or PRE_MASTER or MASTER or PASSIVE
FAULT_ DETECTED
Recommended State == BMC_PASSIVE
PASSIVE
(Recommended State == BMC_MASTER) AND State != MASTER
LISTENING or UNCALIBRATED or SLAVE or PASSIVE
PRE_MASTER
ANNOUNCE_RECEIPT_TIME-OUT_EXPIRES
LISTENING or UNCALIBRATED or PRE_MASTER or MASTER or PASSIVE
QUALIFICATION_TIME-OUT_EXPIRES
MASTER
(Recommended State == BMC_MASTER)
2
Recommended State == BMC_SLAVE
1
CALIBRATION_FORCED_BY_SLAVE
3
SYNCHRONIZATION_ FAULT
UNCALIBRATED
MASTER_CLOCK_ SELECTED
Recommended State == BMC_SLAVE && new_master != old_master
SLAVE
Recommended State == BMC_SLAVE && new_master == old_master

Figure 8: Modified PTP Finite State Machine.

6.7 White Rabbit State Machine

The White Rabbit finite state machine (WR FSM) controls the process of establishing a White Rabbit link between a WR Master and a WR Slave (WR Link Setup). It involves the recognition of two compatible WR nodes, syntonization over the physical layer, measurement of fixed delays and exchange of their values across the link. The procedure differs between WR Master and WR Slave, therefore three states are Slave-only (entered only if the node is in WR Slave mode) and one state is Master-only (entered only if the node is in WR Master mode). The WR FSM shall be executed in the PTP UNCALIBRATED state, it is depicted in Figure 9 and described in the rest of this section.

A typical flow of a complete WR Signaling Message exchange between a WR Slave and a WR Master during WR Link Setup is presented in Appendix C. For clarity, the cooperation of PTP and WR state machines from power up is presented in Appendix D.

6.7.1 Conditions to enter WR Slave mode and start the WR FSM

A WRPTP-enabled port becomes a WR Slave (portDS.wrMode = WR_SLAVE) and starts execution of WR FSM by entering the *PRESENT* state, only when the following conditions are met:

- the port is WR Slave-enabled:
  *portDS.portWrConfig = (WR_S_ONLY **OR** WR_M_AND_S)* **AND**

- the PTP state machine enters the PTP UNCALIBRATED state as a result of STATE_DECISION_EVENT: Recommended State == BMC_SLAVE (1 in Figure 8) **AND**

- the parent port is WR Master-enabled:
  *portDS.parentPortWrConfig = (WR_M_ONLY **OR** WR_M_AND_S)* **AND**

- at least one of the ports on the link is not in WR Mode:
  *portDS.portWrMode = FALSE **OR** portDS.ParentPortWrMode = FALSE.*

6.7.2 Conditions to enter WR Master mode and start the WR FSM

A WRPTP-enabled port becomes a WR Master (portDS.wrMode = WR_Master) and starts execution of WR FSM by entering the *M_LOCK* state only when the following conditions are met:

- the port is WR Master-enabled:
  *portDS.portWrConfig = WR_M_ONLY **OR** WR_M_AND_S* **AND**

- the PTP state machine enters the a PTP_UNCALIBRATED state as a result of CALIBRATION_FORCED_BY_SLAVE transition event (2 in Figure 8)

6.7.2.1 State Description

Table 13 specifies the WR states notation used in Figure 9.

Table 13: WR state definition

PTP portState	Description
IDLE	The WR FSM shall be in the IDLE state if the PTP FSM is in a state other than UNCALIBRATED.
PRESENT	Slave-only state. The WR Slave sends a SLAVE_PRESENT message to the WR Master and waits for the *LOCK* message.
M_LOCK	Master-only state. The WR master waits for the WR Slave to finish successfully the locking process.
S_LOCK	Slave-only state. The WR Slave locks its logic to the frequency distributed over physical layer by the WR Master.
LOCKED	Slave-only state. The WR Slave is syntonized, it sends a *LOCKED* message to the WR Master and waits for the *CALIBRATE* message.
REQ_CALIBRATION	In this state, optional calibration of the node's reception fixed delay can be performed. The node sends a *CALIBRATE* message to the other node. If the calibration is needed, (*calibrated* is set to false), the *calibrationSendPattern* flag in the *CALIBRATE* message is sent to TRUE (0x1). If the calibration is not needed, the *calibrationSendPattern* flag is set to FALSE (0x0). If calibration is not needed, the next state is entered directly, otherwise an indication from the hardware that the calibration has been finished successfully is awaited.
CALIBRATED	The node sends a CALIBRATED message with information about its fixed delays.
RESP_CALIB_REQ	The node's action in this state depends on the value of the *calibrationSendPattern* flag received in the *CALIBRATE* message. A TRUE value of the flag indicates that the calibration pattern shall be enabled. The pattern shall be disabled after *calibrationPeriod* or on reception of the *CALIBRATED* message. If the value of the *calibrationSendPattern* flag is FALSE, the *CALIBRATED* message is awaited for a default timeout. On reception of the *CALIBRATED* message, the next state is entered.
WR_LINK_ON	The value of *wrMode* is set to TRUE and the *IDLE* state is entered.

6.7.2.2 WR FSM Transition Events and Conditions

POWERUP Turning on power to the device or reseting.

WR LINK SETUP REQUIRED DECISION (abrv. D_WR_SETUP_REQ) Event indicating that a WR Link Setup is required and that the WR FSM should be executed starting at the *PRESENT* state, see section 6.7.1.

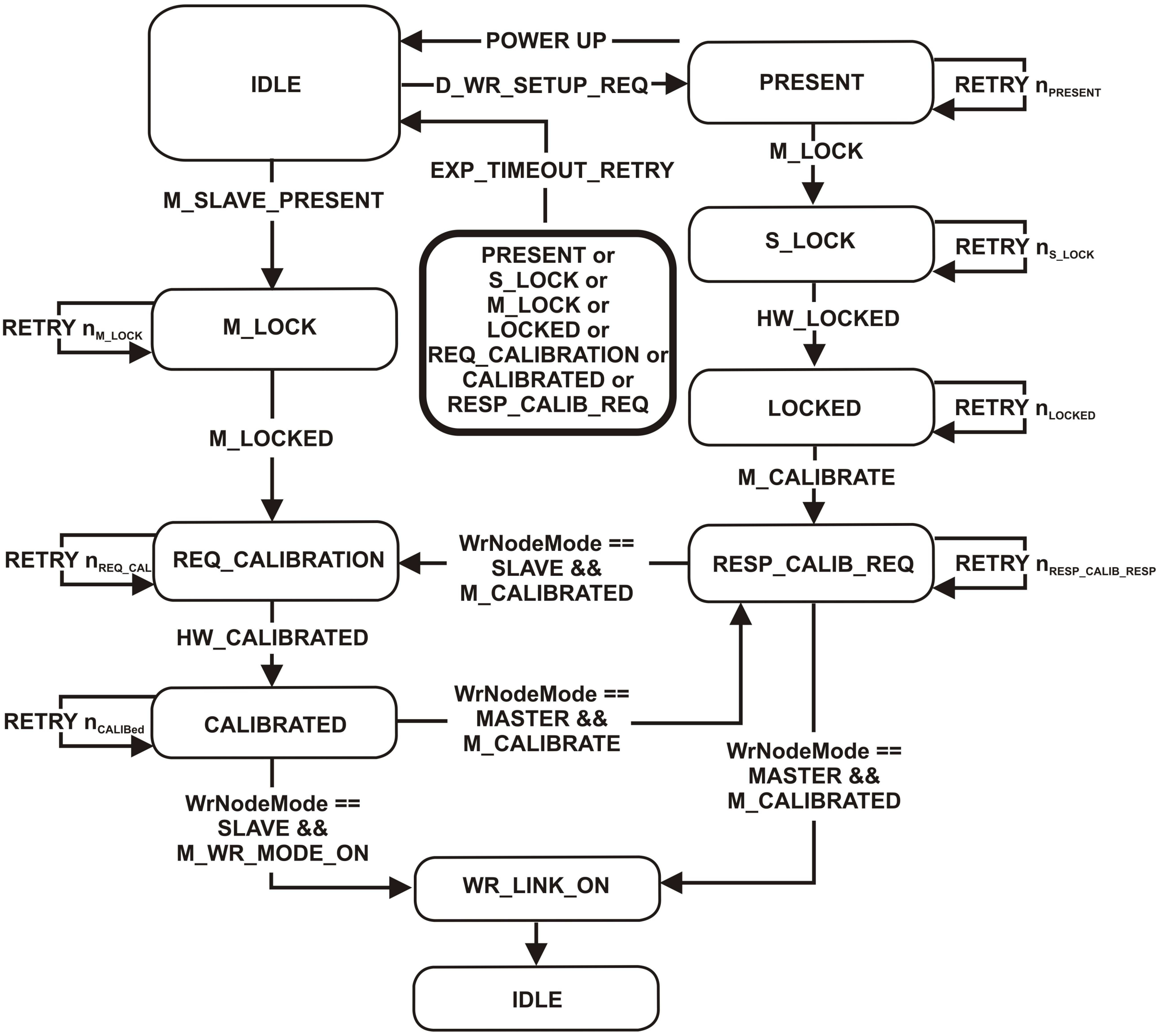


Figure 9: White Rabbit state machine.

LOCK MESSAGE (abrv. M_LOCK) WR *LOCK* Signaling message which triggers frequency locking over the physical layer.

LOCKED HARDWARE EVENT (abrv. HW_LOCKED) Indication from the hardware that frequency locking has been completed successfully.

LOCKED MESSAGE (abrv. M_LOCKED) WR *LOCKED* Signaling message which notifies the WR Master that frequency locking has been completed successfully by the WR Slave.

CALIBRATE MESSAGE (abrv. M_CALIBRATE) WR *CALIBRATE* Signaling message. It requests the recipient to enter the *RESP_CALIB_REQ* state, indicates whether sending of the calibration pattern is required and provides calibration parameters.

CALIBRATED HARDWARE EVENT (abrv. HW_CALIBRATE) Notification from the hardware indicating that calibration has been completed successfully.

CALIBRATED MESSAGE (abrv. M_CALIBRATED) WR CALIBRATED Signaling message. It indicates that the node is calibrated. If the *CALIBRATED* message is received when the calibration pattern is being sent by the recipient, sending of the pattern shall be disabled. The message carries information about fixed delays of the sending node.

WR MODE ON (abrv. M_WR_MODE_ON) WR *WR_MODE_ON* Signaling message. It indicates that the WR Master finished successfully the WR Link Setup and set the wrMode flag to TRUE. It requests the WR Slave to set the wrMode flag to TRUE.

RETRY n_{name} The White Rabbit state machine waits in a given state for a transition event only for a limited time (*TIMEOUT*) The states to which this rule applies are the following: *M_LOCK*, *REQ_CALIBRATEION*, *CALIBRATED*, *PRESENT*, *S_LOCK*, *LOCKED*,*RESP_CALIB_REQ*. After the *TIMEOUT* expires, the state is re-entered for $n_{\{M\_LOCK,REQ\_CAL,CALIBed,PRESENT,S\_LOCK,LOCKED,RESP\_CALIB\_RESP\}}$ number of times.

EXCEED TIMEOUT RETRIES (abrv. EXC_TIMEOUT_RETRY) Indicates that the state has been re-entered for a set number of times ($n_{\{M\_LOCK,REQ\_CAL,CALIBed,PRESENT,S\_LOCK,LOCKED,RESP\_CALIB\_RESP\}}$) and the *TIMEOUT* has expired for the $n+1$ time.

6.8 White Rabbit PTP Profile Summary

6.8.1 Identification

Table 14: Profile print form (clause 19.3.3 of PTP)

PTP Profile	
profleName	White Rabbit
profileVersion	1.0
profileIdentifier	08-00-03-00-01-00
organizationName	European Organization for Nuclear Research (CERN)
sourceIdentification	http://www.ohwr.org/projects/white-rabbit

6.8.2 PTP attribute values

All nodes shall support the ranges and shall have the default initialization values for the attributes as follows:

- portDS.logSyncInterval: The default initialization value shall be 0. The configuration range shall be -1 to 6.
- defaultDS.priority1: The default initialization value shall be 64.

6.8.3 PTP Options

All options of 15.5.4.1.7 and clause 17 of PTP are permitted. By default, these options shall be inactive unless specifically activated by a management procedure.
The node management shall implement the management message mechanism of the IEEE1588-2008 standard.
The best master algorithm shall be the algorithm specified in Chapter 6.4 of this document (Modified BMC).
The delay request-response mechanism shall be the only path delay measurement mechanism.
The TLV mechanism described in Chapter 6.5 shall be supported.

6.9 WRPTP Extension to PTP

WRPTP extends PTP by adding the state transition event defined in section 6.6.

6.10 Implementation-specific additions to PTP required by WRPTP

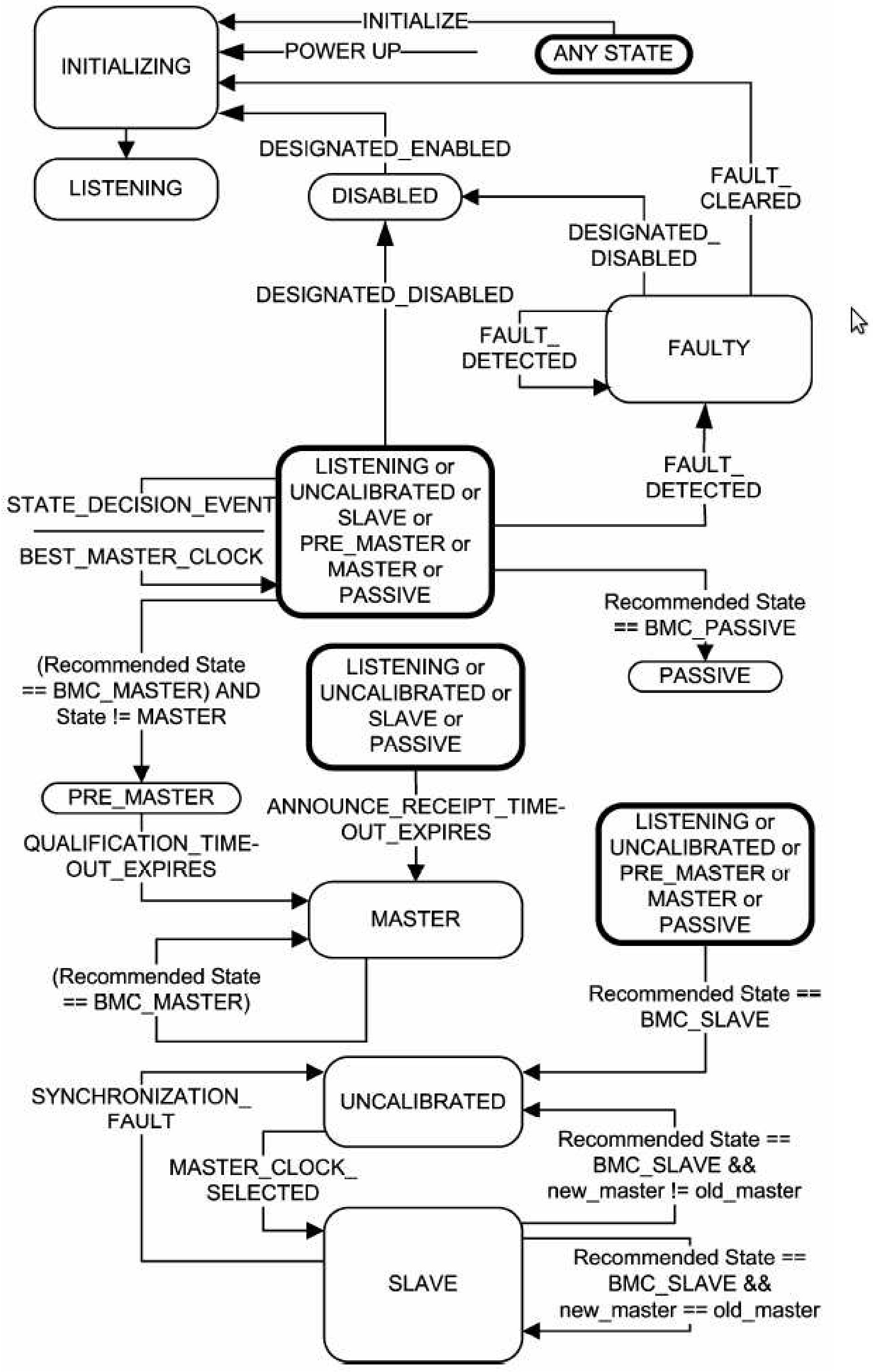
WRPTP shall implement the following implementation-specific features:

- issuing PTP messages with a WR TLV (i.e. suffixed Announce and Signaling messages), as described in section 6.5.2,
- proper handling of these messages as described in 6.5.2,
- WR state machine as defined in 6.7,

- WR data set and additional WR-specific fields to PTP data sets as defined in 6.3,
- SYNCHRONIZATION_FAULT state transition as defined in 6.6.

Appendix

A PTP State Machine



INITIALIZE
POWER UP
ANY STATE
INITIALIZING
LISTENING
DESIGNATED_ENABLED
DISABLED
FAULT_ CLEARED
DESIGNATED_ DISABLED
DESIGNATED_DISABLED
FAULT_ DETECTED
FAULTY
STATE_DECISION_EVENT
BEST_MASTER_CLOCK
LISTENING or UNCALIBRATED or SLAVE or PRE_MASTER or MASTER or PASSIVE
FAULT_ DETECTED
Recommended State == BMC_PASSIVE
PASSIVE
(Recommended State == BMC_MASTER) AND State != MASTER
LISTENING or UNCALIBRATED or SLAVE or PASSIVE
PRE_MASTER
ANNOUNCE_RECEIPT_TIME-OUT_EXPIRES
QUALIFICATION_TIME-OUT_EXPIRES
LISTENING or UNCALIBRATED or PRE_MASTER or MASTER or PASSIVE
MASTER
(Recommended State == BMC_MASTER)
Recommended State == BMC_SLAVE
SYNCHRONIZATION_ FAULT
UNCALIBRATED
MASTER_CLOCK_ SELECTED
Recommended State == BMC_SLAVE && new_master != old_master
SLAVE
Recommended State == BMC_SLAVE && new_master == old_master

Figure 10: State machine for a full implementation of PTP (Figure 23, IEEE1588).

Table 15: PTP portState definition (Table 10, IEEE1588).

PTP portState	**Description**
INITIALIZING	While a port is in the INITIALIZING state, the port initializes its data sets, hardware, and communication facilities. No port of the clock shall place any PTP messages on its communication path. If one port of a boundary clock is in the INITIALIZING state, then all ports shall be in the INITIALIZING state.
FAULTY	The fault state of the protocol. A port in this state shall not place any PTP messages except for management messages that are a required response to another management message on its communication path. In a boundary clock, no activity on a faulty port shall affect the other ports of the device. If fault activity on a port in this state cannot be confined to the faulty port, then all ports shall be in the FAULTY state.
DISABLED	The port shall not place any messages on its communication path. In a boundary clock, no activity at the port shall be allowed to affect the activity at any other port of the boundary clock. A port in this state shall discard all PTP received messages except for management messages.
LISTENING	The port is waiting for the announceReceiptTimeout to expire or to receive an Announce message from a master. The purpose of this state is to allow orderly addition of clocks to a domain. A port in this state shall not place any PTP messages on its communication path except for Pdelay_Req, Pdelay_Resp, Pdelay_Resp_Follow_Up, or signaling messages, or management messages that are a required response to another management message.
PRE_MASTER	The port shall behave in all respects as though it were in the MASTER state except that it shall not place any messages on its communication path except for Pdelay_Req, Pdelay_Resp, Pdelay_Resp_Follow_Up, signaling, or management messages.
MASTER	The port is behaving as a master port.
PASSIVE	The port shall not place any messages on its communication path except for Pdelay_Req, Pdelay_Resp, Pdelay_Resp_Follow_Up, or signaling messages, or management messages that are a required response to another management message.
UNCALIBRATED	One or more master ports have been detected in the domain. The appropriate master port has been selected, and the local port is preparing to synchronize to the selected master port. This is a transient state to allow initialization of synchronization servos, updating of data sets when a new master port has been selected, and other implementation-specific activity.
SLAVE	The port is synchronizing to the selected master port.

B Measurement of fixed delays for Gigabit Ethernet over Optic Fiber

The variation of $\Delta_{\{tx_m, rx_s, tx_s, rx_m\}}$ delays is often caused by the PHY's serializer / deserializer (SerDes), phase locked loop (PLL) or clock and data recovery circuitry (CDR). The delay on the PHY can be measured by detecting the phase shift between SerDes I/O and Tx/Rx clock. This be done in WR by sending a repeated pattern of five ”0” and five ”1” (0000011111) over Gigabit Ethernet. Such signal creates a 125 MHz clock on the SerDes I/O. Since the Tx/Rx clock frequency is 125 MHz, the phase shift between the SerDes I/O and the Tx/Rx clocks is equal to the fixed delay of the PHY (see Figure 11).

The repeated pattern of five ”0” and five ”1” is defined by the IEEE 802.3 standard [2] as *Low-frequency test pattern* (Appendix 36A.2). It shall be generated in WR nodes by a repeated transmission of RD+ K28.7 code-group.

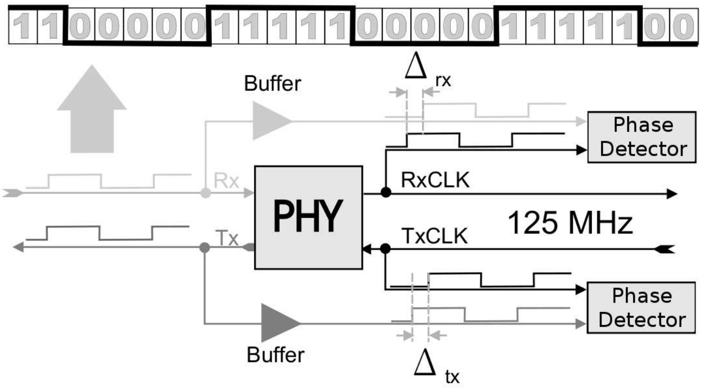


Figure 11: Measurement of fixed delays $\Delta_{\{tx,rx\}}$ in Gigabit Ethernet-based WR node with not fully-deterministic PHY.

Measurement of fixed delays for Gigabit Ethernet over optic fiber, and any other medium, is optional. It is not needed if deterministic PHYs or internal FPGA transceivers which can be internally characterized [3] are used. In such case, the information about the fixed delays is distributed across the link without preceding measurement.

C Typical flow of WR Signaling message exchange

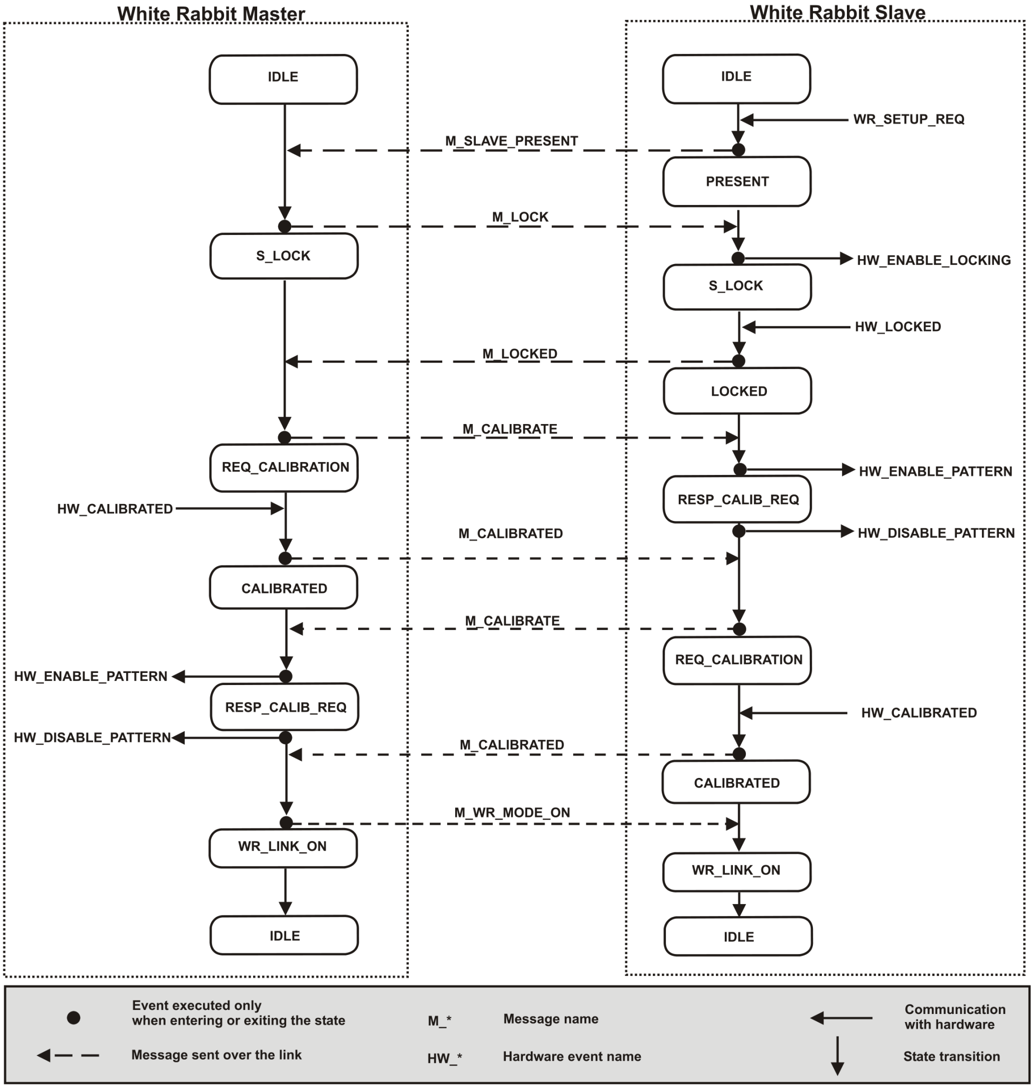


Figure 12: Typical flow of events (no exceptions) during WR Link Setup.

D PTP and WR FSMs from POWER ON use case

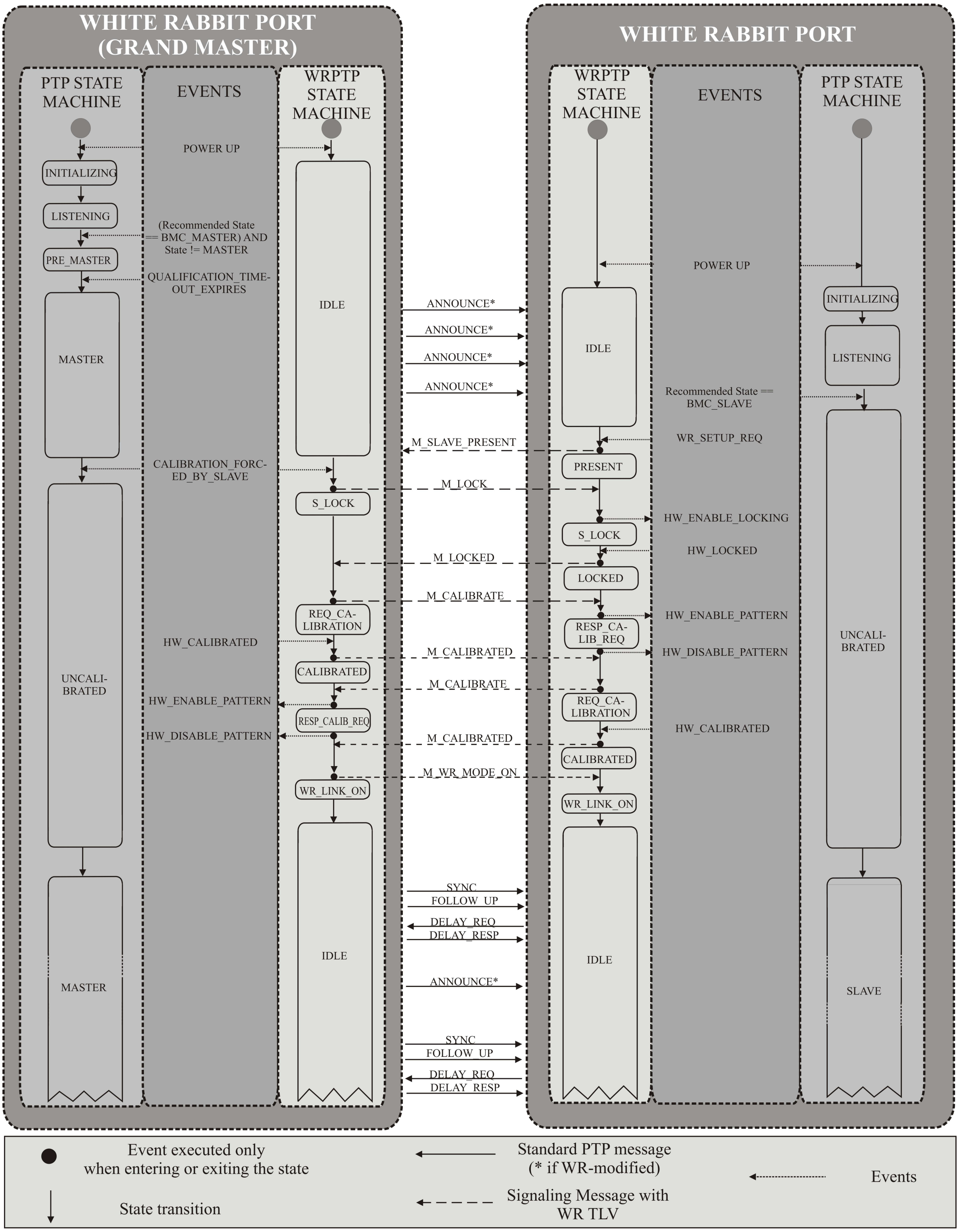


Figure 13: PTP and WR FSMs from POWER ON use case

References

[1] IEEE Std 1588-2008 *IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems*. IEEE Instrumentation and Measurement Society, New York, 2008, http://ieee1588.nist.gov/.

[2] IEEE Std 802.3-2008 *IEEE Standard for Information technology – Telecommunications and information exchange between systems – Local and Metropolitan area networks – Specific requirements*. IEEE Computer Society, New York, 2008.

[3] P.P.M. Jansweijer, H.Z. Peek, *Measuring propagation delay over a 1.25 Gbps bidirectional data link*. National Institute for Subatomic Physics, Amsterdam, 2010, http://www.nikhef.nl/pub/services/biblio/technicalreports/ETR2010-01.pdf.

[4] Takahide Murakami and Yukio Horiuchi, *A Master Redundancy Technique in IEEE 1588 Synchronization with a Link Congestion Estimation*. ISPCS Proccedings, 2010,