White Rabbit Specification: Draft for Comments

September 2010

1 What is White Rabbit?

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) with two further requirements: precise knowledge of the link delay and clock syntonization over the physical layer.

A WR link is formed by a pair of nodes, master and slave. The master node distributes a traceable clock over the physical layer, while the slave recovers this clock and bases its timekeeping on it. Absolute time synchronisation between master and slave is achieved by adjusting the slave's clock's phase and offset to that of the master's. This adjustment is done through the two-way exchange of PTP sync messages, which are corrected to achieve sub-ns accuracy due to the precise knowledge of the link delay.

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 grandparent must be distributed over the physical layer, resulting in a *cascade* of master and slave nodes. As a result of this topology, all WR nodes are PTP boundary clocks. It should be noted that the problem of non-linear error accumulation of chained boundary clocks does not apply, or is at least greatly diminished by the clock recovery mechanism.

Some applications need WR and IEEE1588-2008 nodes to coexist. Examples of this are existing IEEE1588 installations which are to be migrated progressively to White Rabbit, and 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. Figure 1 depicts the topology of a *hybrid* WR/IEEE1588 network.

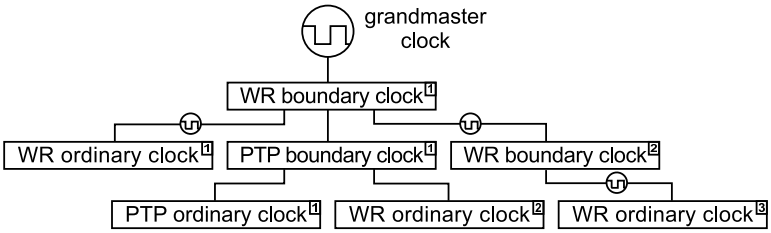

Figure 1: Hybrid WR/IEEE1588 network. White Rabbit nodes work transparently with PTP nodes. WR ordinary clock 3 is more accurately synchronised to the grandmaster than WR ordinary clock 2, which is below a PTP boundary clock.

2 Precision Time Protocol

The IEEE1588-2008 standard [1], known as Precise 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. Full comprehension of the White Rabbit protocol requires fluency in PTP and a copy of the standard at hand.

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*. The time of transmission and reception of event messages is *timestamped* (measured). General messages are used by PTP nodes to identify other PTP nodes, establish clock hierarchy and exchange data, i.e. timestamps, settings, parameters. PTP defines several methods for node' synchronization. Figure 2 presents the messages used when the *delay request-response mechanism* (with *two-step clock*) is used, which is the case in White Rabbit. For simplicity reasons, a PTP node is considered as an *ordinary clock* in the remainder of this section; such clocks have only one port.

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. The algorithm defines the role of each PTP node in PTP network hierarchy; the outcome of the algorithm is the recommended next state of the PTP node and the node's synchronization source (grandparent). In other words, a PTP node decides to which other PTP node it should synchronize based on the information provided in announce messages and using 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 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

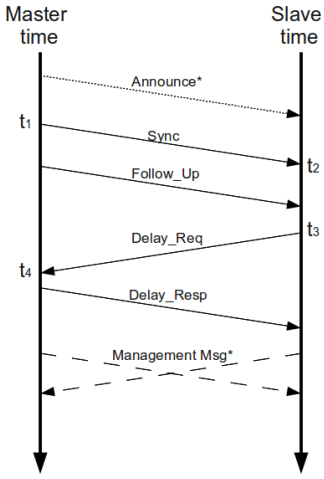

Figure 2: PTP messages used by WRPTP (messages annotated with * are extended by WR Protocol).

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. They are not essential for PTP synchronization.

Adjustment of the Slave's clock using the offset and the delay calculated with timestamps (t_1, t_2, t_3, t_4) results in the Slave's synchronization with the Master clock.

3 Link Delay Model

The delay of a message travelling from master to slave (see Figure 3) 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}. From now on in this document we refer to this missing equation as the *physical medium correlation*. Describing a procedure to obtain this equation is out of the scope of this document. However, section 3.1 provides correlations obtained empirically for one scenario.

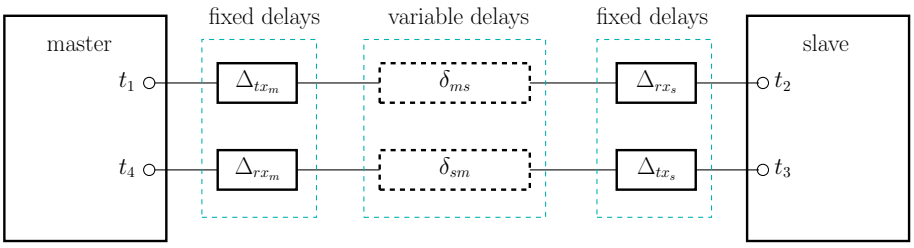

Figure 3: Delay model of a WR link. The timestamps are accurately corrected for link asymmetries by the usage of the four fixed delays $\Delta_{\{tx_m,rx_s,tx_s,rx_m\}}$ and the relationship between both variable delays $\delta_{\{ms,sm\}}$.

3.1 Physical Medium Correlation

An accurate correlation between both variable delays on the transmission line is essential for obtaining an acceptable estimate of the delay asymmetry on a WR link. The origin of this correlation is highly implementation-dependent. Thus this document just assumes that such correlation exists and is known.

The rest of this section presents a well-known physical medium correlation.

3.1.1 Ethernet 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 [2]:

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

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 IEEE1588-2008 is expressed in our own notation by using equations (1), (2) and (7) as follows:

$$\text{delay}_{ms} = \mu + \text{asymmetry} \tag{9}$$
$$\text{delay}_{sm} = \mu - \text{asymmetry} \tag{10}$$

The delay asymmetry cannot be calculated unless we use the physical medium correlation.

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):

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

It can be noticed that if $\Delta << \mu$, the above equation can be simplified:

$$\text{asymmetry} = \Delta_{tx_m} + \Delta_{rx_s} - \frac{\Delta - \alpha\mu}{2+\alpha} \tag{14}$$

.

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 uptime 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 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. The method for obtaining fixed delays for Gigabit Ethernet is described in the rest of this section.

5.1 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 can be done 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 4). The repeated pattern of five "0" and five "1" is an example of *calibration pattern* which is defined by the node requesting calibration.

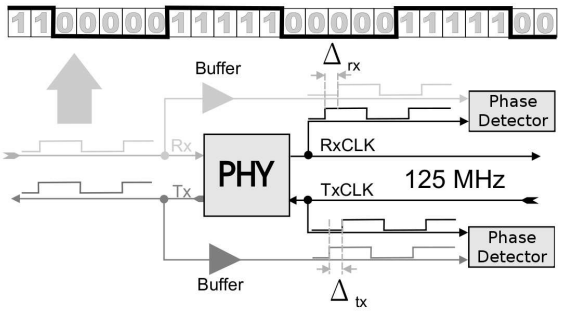

Figure 4: Measurement of fixed delays $\Delta_{\{tx,rx\}}$ in Gigabit Ethernet-based WR node with not full-deterministic PHY.

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 ([2]) are used. In such case, know fixed delays are distributed across the link without preceding measurement.

6 White Rabbit PTP Extension

6.1 Overview

White Rabbit extends the IEEE1588-2008 (PTP) standard to achieve sub-ns accuracy while still benefiting from PTP's synchronization and management mechanisms. From now on in this document, the White Rabbit extension to the PTP standard will be referred to as *WRPTP*. WRPTP introduces *the White Rabbit Link Setup* (WR Link Setup), which is a process for establishing the WR link. It includes syntonization of the local clock over the physical layer, measurement of fixed delays and their distribution over the link. The WR extension takes advantage of the Link Delay Model to obtain an accurate delay estimation, i.e. it uses the delay asymmetry equation (13) for Gigabit Ethernet over Fiber Optic. WRPTP extends the PTP messages and Data Sets (DS).

Since the WR Link Setup is performed in the PTP UNCALIBRATED state, it is essential for a WR node to implement this state as specified in the PTP state machine (Appendix A): a transition state between the LISTENING or PRE_MASTER or MASTER or PASSIVE state and the SLAVE state.

A *White Rabbit Switch* (WRSW) is not a PTP-compliant boundary clock. It is considered as a set of ordinary clocks with predefined functionality (WR Master or WR Slave) rather than a clock with multiple PTP ports. As a consequence, WRPTP messages are never forwarded. In this document, the term *node* is used interchangeably with *port*. A WR Master node/port is an ordinary clock with predefined Master functionality, working as a Master on the link. A WR Slave node/port is an ordinary clock with predefined Slave functionality, working as a Slave on the link. Proper performance of a WR network is ensured by connecting a WR Slave port to a WR Master port. Additionally, optimal performance of a hybrid WR/IEEE1588 network can only be achieved by connecting a non-WR port to WR Master port.

6.2 WRPTP Data Sets Fields

The PTP standard defines data sets (DS) to store the static and dynamic variables needed for the operation of the protocol (section 8, IEEE1588 [1]). WRPTP requires additional DS fields to store the WR-specific parameters. Table 1 defines and describes required DS fields.

Table 1: WRPTP Data Sets fields

DS member	**DS name**	**Values**	**Description**
wrPortMode	portDS	NON_WR, WR_SLAVE, WR_MASTER	Determines predefined function of WR port (static).
calibrated	portDS	TRUE, FALSE	Indicates whether fixed delays of the given port are known.
deltaTx	portDS	64 bit value	Port's Δ_{tx} measured in picoseconds and multiplied by 2^{16}.
deltaRx	portDS	64 bit value	Port's Δ_{rx} measured in picoseconds and multiplied by 2^{16}.
calPeriod	portDS	32 bit value	Calibration period in microseconds.
calPattern	portDS	32 bit value	Medium specific calibration pattern.
calPatternLen	portDS	16 bit value	Number of bits of calPattern to be repeated.
wrMode	portDS	TRUE, FALSE	If TRUE, the port is working in WR mode.
wrAlpha	portDS	32 bit value	α parameter as described in section 3.1.1.
grandmasterWrPortMode	parentDS	NON_WR, WR_SLAVE, WR_MASTER	Determines predefined function of the PTP grandmaster.
grandmasterDeltaTx	parentDS	64 bit value	Grandmaster's Δ_{tx} measured in picoseconds and multiplied by 2^{16}.
grandmasterDeltaRx	parentDS	64 bit value	Grandmaster's Δ_{rx} measured in picoseconds and multiplied by 2^{16}.
grandmasterWrMode	parentDS	TRUE, FALSE	If TRUE, the grandmaster is working in WR mode.

6.3 Modified Best Master Clock Algorithm

The Best Master Clock algorithm is used in PTP to compare local clocks, determine which clock is the *"best"* and recommend the next state of the PTP state machine (section 9.3, IEEE1588 [1]).

It is required from modified BMC that the comparison of a WR Master with a WR Slave or non-WR clock results in the WR Master being the *"best"* clock. To ensure a proper BMC outcome, the WR Master *clockClass* value shall be in the range of 1 through 127 (recommended 6) and the WR Slave *clockClass* shall be in the range of 128 through 255 (recommended 255).

6.4 WRPTP Messages

6.4.1 Overview

White Rabbit benefits from PTP's messaging facilities. It uses two-step clock delay request-response mechanism, and customizes Announce and Management messages (Figure 2). In particular, it adds suffix to Announce message, defines *WR Type-Length-Valu*e (WR TLV) type, WR management action and management IDs.

A White Rabbit Master node announces its presence by adding a WR TLV suffix to the Announce message. The suffix is ignored by standard PTP nodes, but read and interpreted by White Rabbit nodes. The information provided in the WRPTP Announce message is sufficient for a WR Slave to decide whether the WR link can be established and maintained. The WR link is established through the WR Link Setup process in the PTP UNCALIBRATED state. If a WR Link Setup is required, the WR Slave starts the process and requests the WR Master to do the same. During the WR Link Setup, communication between the WR Master and the WR Slave is performed using the PTP Management mechanism extended for White Rabbit requirements. Once the WR link has been established, the WR nodes use a PTP delay request-response mechanism (section 11.3, IEEE1588 [1]).

6.4.2 WR Type-Length-Value Type

All PTP messages can be extended by means of a standard type, length, value (TLV) extension mechanism. White Rabbit defines the value of TLV type out of the range reserved for Experimental TLVs (Table 34, IEEE1588 [1]) as depicted in Table 2. This value is used to recognize the WR TLV entity in all WR custom messages.

Table 2: White Rabbit Type-Length-Value (WR TLV) type

tlvType values	Value (hex)	Defined in clause
White Rabbit TLV (WR TLV type)	0x2004	—

6.4.3 WRPTP Announce Message

The standard PTP Announce Message is suffixed by one entity of the data type TLV with the tlvType of WR TLV. The WRPTP Announce message has the structure defined in Table 3. The *dataField* of the suffix TLV stores the *wrFlags* which are defined in Table 4.

Table 3: White Rabbit Announce Message

Bits								Octets	TLV Offset	Content
7	6	5	4	3	2	1	0			
header								34	0	section 13.3, IEEE1588 [1].
body								30	34	section 13.5, IEEE1588 [1].
tlvType								2	64	0x2004, see 6.4.2.
lengthField								2	66	0x2, section 14.1.2, IEEE1588 [1].
wrFlags								2	68	see Table 4.

Table 4: White Rabbit flags (unused flags are reserved)

Octet	Bit	Message type	Name	Description
0	0	Announce	wrMaster	TRUE if the port of the originator is predefined WR Master.
0	1	Announce	wrSlave	TRUE if the port of the originator is predefined WR Slave.
0	2	Announce	calibrated	TRUE if the port of the originator is calibrated.
0	3	Announce	wrModeOn	TRUE if the port of the originator is in WR mode.

6.4.4 WRPTP Management Messages

White Rabbit extends the default PTP management mechanism described in section 15.2 of IEEE1588. The extension conforms to the PTP management message format presented in Table 5. It uses the reserved range of *actionField* values (Table 38, IEEE1588 [1]) to define a *White Rabbit Command* (WRC) as described in Table 6. WRC management messages trigger transitions in WR state machines. They are recognized by the *managementId* field of *managementTLV* (Table 8, 9 & 10). WR *management IDs* are defined in Table 11. WRPTP management messages are exchanged only within one link connection (no forwarding), therefore the *startingBoundaryHops* and *boundaryHops* fields are unused and set to 0x0. The rest of this subsection describes the WR management messages in detail.

Table 5: PTP Management Message (Table 37, IEEE1588 [1])

Bits								**Octets**	**TLV Offset**
7	6	5	4	3	2	1	0		
header								34	0
targetPortIdentity								10	34
startingBoundaryHops								1	44
boundaryHops								1	45
reserved			actionField					1	46
reserved								1	47
managementTLV								M	48

Table 6: White Rabbit value of the actionField

Action	**Action taken**	**Value (hex)**
WR_CMD	The management message shall carry a single TLV. The *managementId* field of the TLV indicates the specific event which triggers transition in WR FSMs.	0x5

Table 7: White Rabbit managementId values

managementId name	**managementId value (hex)**	**Allowed actions**	**Applies to**
SLAVE_PRESENT	0x6000	WR_CMD	port
LOCK	0x6001	WR_CMD	port
LOCKED	0x6002	WR_CMD	port
MASTER_CALIBRATE	0x6003	WR_CMD	port
MASTER_CALIBRATED	0x6004	WR_CMD	port
SLAVE_CALIBRATE	0x6005	WR_CMD	port
SLAVE_CALIBRATED	0x6006	WR_CMD	port
WR_MODE_ON	0x6007	WR_CMD	port

6.4.4.1 SLAVE_PRESENT

Message sent by WR Slave to WR Master. It initiates the WR Link Setup process in the master. The message shall have the form specified in Table 8.

6.4.4.2 LOCK

Message sent by WR Master to WR Slave to request the start of frequency locking. The message shall have the form specified in Table 8.

6.4.4.3 LOCKED

Message sent by WR Slave to WR Master. It indicates successful completion of frequency locking. The message shall have the format specified in Table 8.

Table 8: WR Management TLV

Bits								**Octets**	**TLV Offset**	**Content**
7	6	5	4	3	2	1	0			
tlvType								2	0	0x2004, see 6.4.2.
lengthField								2	2	0x2, section 15.5.2.3 IEEE1588 [1].
managementId								2	4	Defined in Table 11.

6.4.4.4 MASTER_CALIBRATE and SLAVE_CALIBRATE

Messages sent to request calibration pattern (see 5.1). *SLAVE_CALIBRATE* is sent by WR Slave to WR Master. *MASTER_CALIBRATE* is sent by WR Master to

WR Slave. Both messages carry a set of parameters defining the calibration pattern to be sent. The message format and parameters are described in Table 9.

Table 9: CALIBRATE WR Management TLV

Bits								**Octets**	**TLV Offset**	**Content**
7	6	5	4	3	2	1	0			
tlvType								2	0	0x2004, see 6.4.2.
lengthField								2	2	0xC, section 15.5.2.3 IEEE1588 [1].
managementId								2	4	MASTER_CALIBRATE or SLAVE_CALIBRATE.
calibrationPeriod								4	6	The value defines the time (in microseconds) for which the calibration pattern should be sent by receiving node.
calibrationPattern								4	10	The value defines the calibration pattern which should be sent by the receiving node.
calibrationPatternLen								2	12	The value defines the number of bits of *calibrationPattern* field which should be used as repeated pattern (starting with the LSB).

6.4.4.5 MASTER_CALIBRATED

Message sent by WR Master to WR Slave. If preceeded by *MASTER_CALIBRATE*, it indicates successful calibration. Otherwise, it indicates that master does not need calibration. In both cases, the message provides the WR Slave with the values of the Master's fixed delays (Δ_{tx_m} and Δ_{rx_m}). The messages shall have the format specified in Table 10.

Table 10: CALIBRATED WR Management TLV

Bits								**Octets**	**TLV Offset**	**Content**
7	6	5	4	3	2	1	0			
tlvType								2	0	0x2004, see 6.4.2.
lengthField								2	2	0x24, section 15.5.2.3 IEEE1588 [1].
managementId								2	4	MASTER_CALIBRATED.
deltaTx								16	6	The value of Δ_{tx_m} measured in picoseconds and multiplied by 2^{16}.
deltaRx								16	22	The value of Δ_{rx_m} measured in picoseconds and multiplied by 2^{16}.

6.4.4.6 WR_MODE_ON

Message sent by WR Slave to WR Master. It indicates successful completion of the WR Link Setup process and requests the WR Master to enter WR mode. The message shall have the format specified in Table 8.

6.5 White Rabbit State Machines

The White Rabbit finite state machines (WR FSMs) control the process of establishing a White Rabbit link between a WR Master and a WR Slave (WR Link Setup). It involves 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 two separate FSM are defined.

The fields of the WR Data Set (Table 4) determine whether the WR Master FSM, the WR Slave FSM, or no WR FSM shall be executed in the PTP UNCALIBRATED state.

6.5.1 WR Slave FSM

The WR Slave FSM exits the IDLE state (is executed) only when the PTP state machine (Appendix A) is in the PTP UNCALIBRATED state and the following conditions are met:

- the node is WR Slave:
  ($portDS.wrPortMode = WR\_SLAVE$) **AND**
- the parent node is WR Master:
  ($parentDS.grandmasterWrPortMode = WR\_MASTER$) **AND**
- the node or parent node or both nodes are not in WR Mode:
  (*portDS.wrMode = FALSE OR parentDS.grandmasterWrMode = FALSE*).

The state machine is presented in Figure 5 and described in the rest of this subsection.

6.5.1.1 WR Slave FSM Transition Events and Conditions

Start Power up.

WR LINK SETUP REQUIRED (abrv. WR_SETUP_REQ) Event indicating that WR Link Setup is required and WR Slave FSM should be executed, explanation above.

LOCK WR *LOCK* Management message which triggers frequency locking over the physical layer.

LOCKED Notification from the hardware that the frequency locking has been completed successfully.

MASTER_CALIBRATE (abrv. M_CALIBRATE) WR Management message. Upon reception, WR Slave starts sending calibration pattern for the period of time indicated by $calibrationPeriod$.

MASTER_CALIBRATED (abrv. M_CALIBRATED) WR Management message. It indicates that the WR Master is calibrated. If the *MASTER_CALIBRATED* message is received in the $LOCKED$ state, it means that the WR Master knows its fixed delays, it does not need to be calibrated, therefore the *MASTER_CALIBRATE* state can be skipped. If the message is received in *MASTER_CALIBRATE* state, it indicates successful completion of the WR Master's calibration.

PORT IS CALIBRATED (abrv. IS_CAL) Information stored in portDS (*calibrated*) indicating that the fixed delays for the port are known.

PORT IS NOT CALIBRATED (abrv. NOT_CAL) Information stored in the portDS (*calibrated*) indicating that the fixed delays for port are not known.

ANY_TIMEOUT The time spent by the WR FSM in the following states: *present*, *lock*, *locked*, *calib_m*, *calib_s* is limited by timeouts. Exceeding any of these timeouts is represented by $ANY\_TIMEOUT$.

6.5.1.2 State Description

Idle WR FSM shall be in the Init state if the PTP FSM is in a state other than UNCALIBRATED.

Present WR Slave sends SLAVE_PRESENT message to WR Master and waits for the *LOCK* message.

Lock Locking of WR Slave's logic to the frequency distributed over physical layer by the WR Master. When finished successfully, a LOCKED message is sent to the WR Master.

Locked WR Slave waits for MASTER_CALIBRATE or MASTER_CALIBRATED message.

Calibrate Master (abrv. *calib_m*) Reception fixed delay is measured for the WR Master. The WR Slave sends the calibration pattern for the period indicated by *calibratePeriod* in the message received from the WR Master. When the MASTER_CALIBRATED message is received from the WR Master, the state is exited.

Calibrate Slave (abrv. *calib_s*) Reception fixed delay is measured for the WR Slave. It sends a SLAVE_CALIBRATE message to the WR Master (indicating the period of calibration pattern). As soon as the measurement is finished successfully, a SLAVE_CALIBRATED message is issued and the state is exited.

Calibration_completed (abrv. cal_cmp) WR Slave sends WR_MODE_ON message to WR Master, enters WR Idle state; PTP UNCALIBRATED state is exited.

Any Any state.

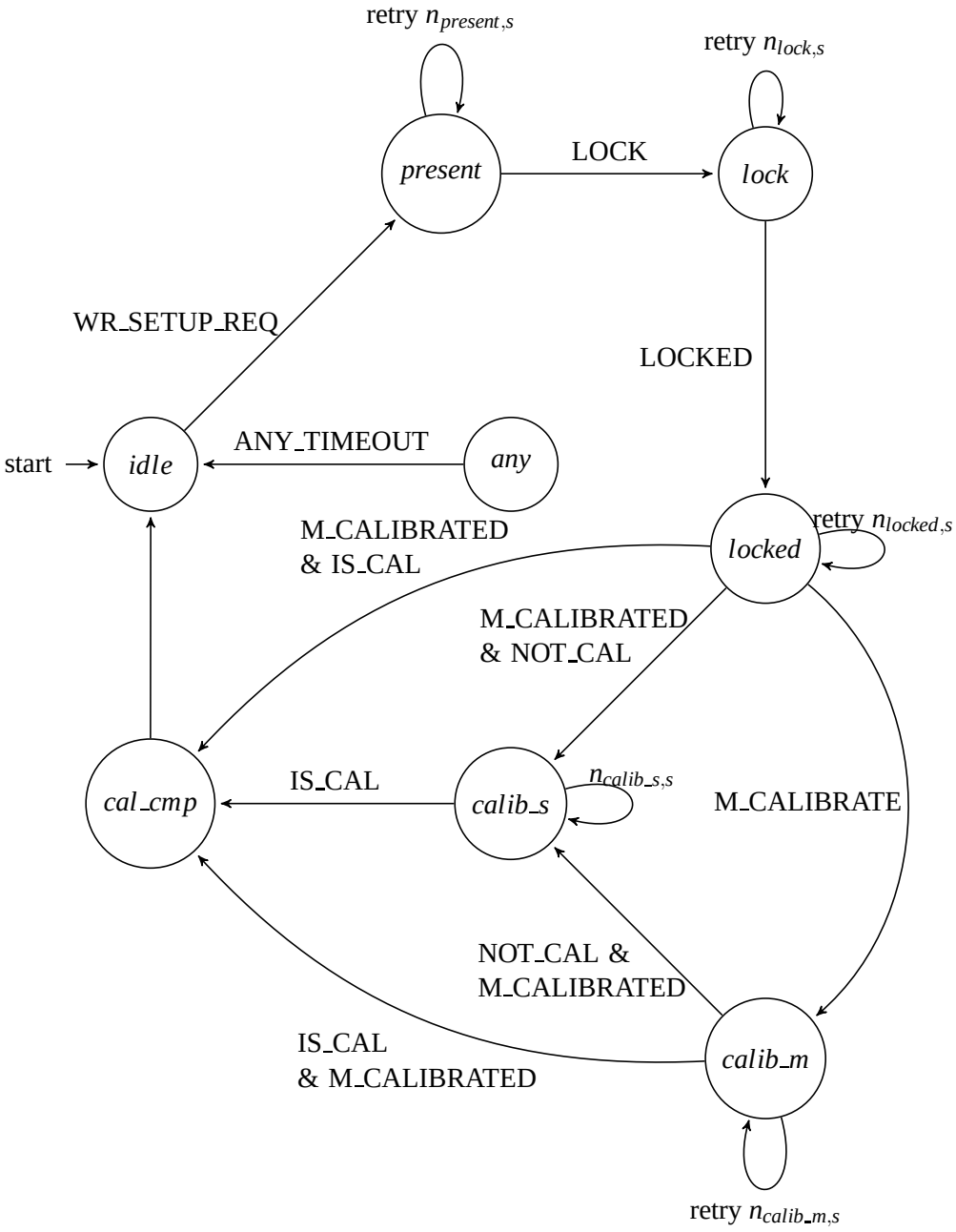

Figure 5: White Rabbit Slave FSM.

6.5.2 Master FSM

The WR Master node shall enter the PTP UNCALBRATED state, and start execution of the WR Master FSM, when it receives a SLAVE_PRESENT WR Management message (Table 11). The state machine is presented in Figure 6 and described in the rest of this subsection.

6.5.2.1 WR Slave FSM Transition Events and Conditions

Start Power up.

SLAVE_PRESENT (abrv. S_PRESENT) WR Management message received from WR Slave. It indicates that a WR Link Setup is required. Reception of this message forces the PTP FSM to enter the UNCALIBRATED state and starts execution of the WR FSM.

LOCKED WR Management message. It indicates that the syntonization has been finished by the WR Slave.

PORT IS CALIBRATED (abrv. IS_CAL) Information stored in the portDS (*calibrated*) indicating that the fixed delays for the port are known.

PORT IS NOT CALIBRATED (abrv. NOT_CAL) Information stored in portDS (*calibrated*) indicating that the fixed delays for the port are not known.

SLAVE_CALIBRATE (abrv. S_CALIBRATE) WR Management message. Upon reception, the WR Master starts sending a calibration pattern for the period of time indicated by $calibrationPeriod$.

SLAVE_CALIBRATED (abrv. S_CALIBRATED) WR Management message. It indicates that the WR Slave is calibrated. If $SLAVE\_CALIBRATED$ is received without a preceding *SLAVE_CALIBRATE* message, it means that the WR Slave knows its fixed delays, it does not need to be calibrated. If the message is received after a $SLAVE\_CALIBRATE$ message, it indicates successful completion of the WR Slave's calibration.

WR_MODE_ON WR Management message. It indicates that the WR Slave has finished calibration and requests WR Master to turn on White Rabbit mode.

ANY_TIMEOUT The time spent by the WR FSM in the following states: *lock*, *locked*, *calib_m*, *calib_s*, *cal_cmp* is limited by timeouts. Exceeding any of these timeouts is represented by *ANY_TIMEOUT*.

6.5.2.2 State Description

Idle WR FSM shall be in the Init state if the PTP FSM is in a state other than UNCALIBRATED.

Lock Locking of WR Slave's logic to the frequency distributed over physical layer by the WR Master. When finished successfully, a LOCKED message is received from the WR Slave.

Locked If the WR Master needs to measure its reception fixed delay (*calibrated* is FALSE), it enters the *calib_m* state. Otherwise, it sends a *MASTER_CALIBRATED* message and enters *calib_s*.

Calibrate Master (abrv. *calib_m*) WR Master's reception fixed delay is measured. When this state is entered, a *MASTER_CALIBRATE* message is sent to the WR Slave. On measurement successful completion, a *MASTER_CALIBRATED* message is sent and the calib_s state is entered.

Calibrate Slave (abrv. *calib_s*) WR Slave's reception fixed delay is measured. WR Master waits for a SLAVE_CALIBRATE or a SLAVE_CALIBRATED message. On reception of a SLAVE_CALIBRATE message, the WR Master starts sending a calibration pattern for the period indicated by *calibratePeriod*. When a SLAVE_CALIBRATED message is received, the state is exited.

Calibration_completed (abrv. cal_cmp). WR Master waits for a WR_MODE_ON message. On reception of the message, *wrMode* is set to TRUE, the WR Idle state is entered and the PTP UNCALIBRATED state is exited.

Any Any state.

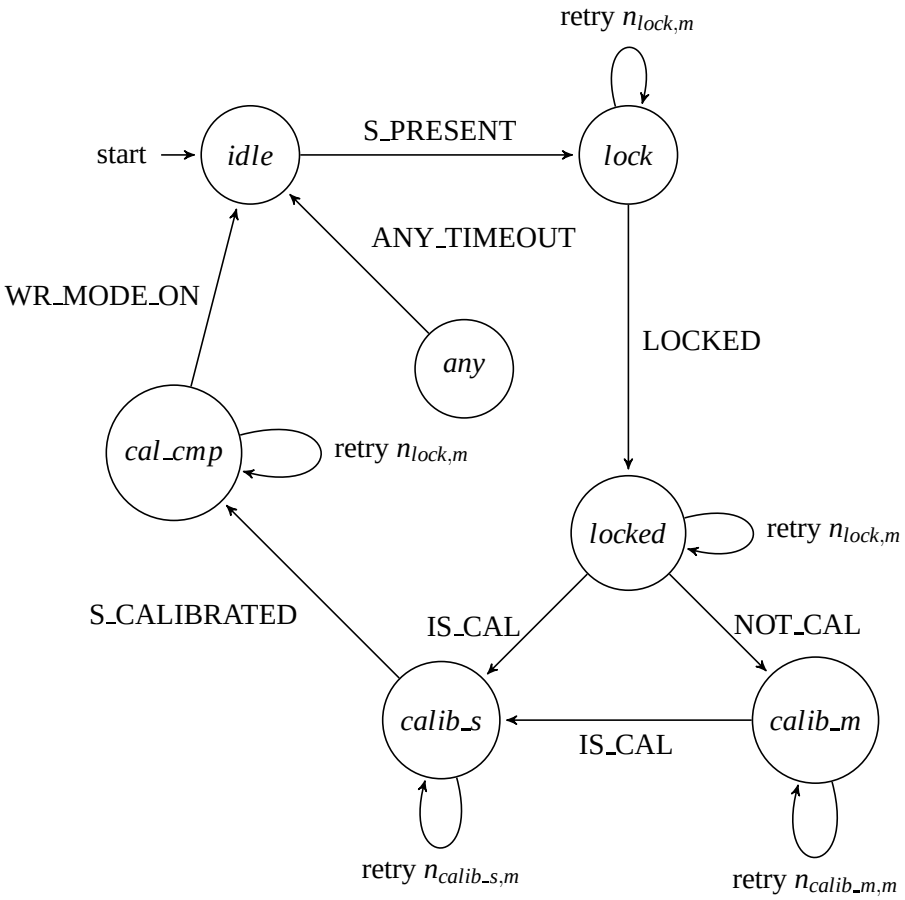

Figure 6: White Rabbit Master FSM

A PTP State Machine

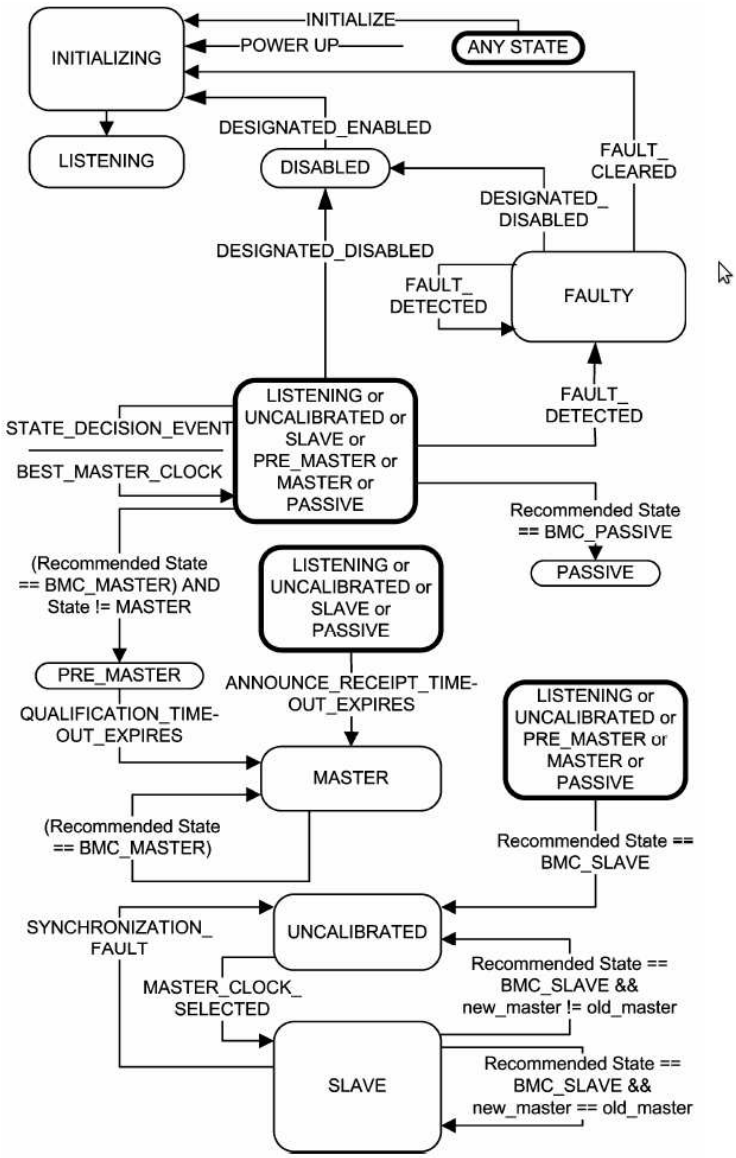

Figure 7: State machine for a full implementation of PTP (Figure 23, IEEE1588).

Table 11: 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.

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] 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.