White Rabbit Specification:
First Draft

Emilio G. Cota
<cota@braap.org>
July 2010

1 What is White Rabbit?

White Rabbit (WR) is a protocol developed to synchronise with sub-ns accuracy two contiguous nodes in a packet-based network. The protocol results from the combination of IEEE1588-2008 (PTP) with two further requirements: precise knowledge of the link delay and clock syntonisation 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 can be regarded as boundary clocks, keeping all sync messages private to each of the links.

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 forces 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. [on the figure I’ll mark links that make use of the physical layer in a different colour to emphasize why WR ordinary clock 3 is more accurately synchronised to the grandmaster than WR ordinary clock 2, which is below a PTP boundary clock.]

2 Link Delay Model

The delay of a message travelling from master to slave can be expressed as the sum

\[ delay_{ms} = \Delta_{tx_m} + \delta_{ms} + \Delta_{rx_s} \] (1)

where \( \Delta_{tx_m} \) is the fixed delay due to the master’s transmission circuitry, \( \delta_{ms} \) is the variable delay incurred in the transmission medium, and \( \Delta_{rx_s} \) 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

\[ delay_{sm} = \Delta_{tx_s} + \delta_{sm} + \Delta_{rx_m} \] (2)

The characterisation of the link is completed with an equation to relate the two variable delays, \( \delta_{ms} \) and \( \delta_{sm} \). 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 2.1 provides correlations obtained empirically for some scenarios.

Figure 2: 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 \( \delta\{ms,sm\} \).

(The drawing is incomplete)
2.1 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 some well-known physical medium correlations.

2.1.1 Ethernet over a Single-mode Optical Fibre

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

\[ \delta_{ms} = (1 + \alpha)\delta_{sm} \]  

(This probably should go as in an appendix devoted to Ethernet over fibre, citing the paper from the Dutch, and possibly some other remarks about this (e.g. the wavelengths with less dispersion). The point is that I want to keep separate implementation (Ethernet + single fibre is an implementation!) from the protocol as much as possible)

3 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

\[ \mu = \frac{(t_2 - t_1) + (t_4 - t_3)}{2} \]  

Note that the transmission delays \( t_2 - t_1 \) and \( t_4 - t_3 \) can be expressed in terms of WR’s Delay Model:

\[ t_2 - t_1 = \Delta_{tx_m} + \delta_{ms} + \Delta_{rx_s} + offset_{ms} \]  
\[ t_4 - t_3 = \Delta_{tx_s} + \delta_{sm} + \Delta_{rx_m} - offset_{ms} \]  

, where \( offset_{ms} \) is the time offset between the slave’s clock and the master’s. Combining the three equations above we obtain

\[ 2\mu = \Delta + \delta_{sm} + \delta_{ms} \]  

, where \( \Delta \) accounts for all fixed delays in the path, i.e.

\[ \Delta = \Delta_{tx_m} + \Delta_{rx_s} + \Delta_{tx_s} + \Delta_{rx_m} \]
The delay asymmetry $a$, defined 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:

\[
\begin{align*}
\text{delay}_{ms} &= \mu + a \\
\text{delay}_{sm} &= \mu - a
\end{align*}
\]

(9) \hspace{2cm} (10)

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

3.1 Solution for Ethernet over a Single-mode Optical Fibre

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

\[
\begin{align*}
\delta_{ms} &= \frac{1 + \alpha}{2 + \alpha} (2\mu - \Delta) \\
\delta_{sm} &= \frac{2\mu - \Delta}{2 + \alpha}
\end{align*}
\]

(11) \hspace{2cm} (12)

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

\[
a = \Delta_{txm} + \Delta_{rxs} - \frac{\Delta + \alpha\mu}{2 + \alpha}
\]

(13)

4 White Rabbit State Machines

4.1 White Rabbit Master

State descriptions ☑️

Init Initial state. Hardware start-up.

Announce Send a master_announce message. Includes the master’s fixed delays.

Any Any state.

4.2 White Rabbit Slave

State descriptions ☑️

Init Initial state. Hardware start-up.

Announce Send a master_announce message. Includes the master’s fixed delays.

Any Any state.
5 PTP exchanges

[AFAICT a smart slave that knows the master’s delays could just do with a normal Sync-event exchange. This means the peer delay mechanism isn’t needed. We do NOT subtract $\Delta_{rx,m}$ & Co. and leave it all up to the smart slave.]

TODO: everything is a boundary clock

6 Phase compensation

[how the slave interprets the timestamps]

7 TODO

- Ethernet. Ethertype? $\$2500. Use slow-protocols temporarily?
- Once a full implementation has been successfully tested, we may consider specifying a certain PTP protocol for White Rabbit networks.
Figure 4: White Rabbit slave FSM