Discussion On
A White Rabbit based
CERN Control and Timing Network
version 1.1

Jean-Claude Bau, Maciej Lipinski

October 2011
<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Authors</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>20/06/2011</td>
<td>J-C.B, M.L.</td>
<td>Created as a summary of week-long brainstorming.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2. Re-wrote Data Masters chapter based on the latest discussion:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- described possible redundant Data Master solutions</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- proposed desired Data Master’s features and redundancy solution</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- changed Figure 5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4. Added Conclusions.</td>
</tr>
</tbody>
</table>
Contents

1 Introduction 3
2 Network Topology 4
3 Accelerator Networks (VLANs) 6
4 Multicast groups 9
5 Time distribution 9
6 Data Master (DM) 11
  6.0.1 Normal operation in redundant DM’s configuration 11
  6.0.2 Issues regarding switching between DMs 12
  6.0.3 Consideration of different topologies/solutions for redundant DMs 13
  6.0.4 The final proposition of redundant DM layout and DM’s features 15
7 Timing Master (very preliminary draft) 17
8 I’m alive 18
9 Node 18
10 WR CERN Control and Timing Network Layout 19
11 Issues 20
12 Conclusions 20
1 Introduction

This document was created to summarize a week-long brainstorming and discussion on how a new CERN Control and Timing Network based on the White Rabbit technology could look like.

Such a discussion, thus this document, had the following aims:

- Evaluate how the WR could be actually applicable in practice in the "real CERN world".
- Investigate which features provided by WR Network (e.g. VLANs) could be useful in the Control and Timing Network and what kind of special features (not standard in Ethernet Network) could be required (mainly from the WR Switch).
- Evaluate how the topology restrictions introduced by WR translate into "real" network layout.
- Evaluate if and what kind of customization of WR node is needed to create Data Master.
- In short: verify how the theory translates into practice...

Disclaimer:
This document might be used in the future as basis and source of ideas to design New Control and Timing Network for CERN’s accelerators. However, it might be also abandoned and forgotten. This document should not by taken, by all means, as an official design description.
2 Network Topology

- We have 4 Data Masters:
  - LHC - controls LHC accelerator.
  - LIC - controls LHC Injection Chain (BPS,PS,SPS,...), but also LHC, AD and REX.
  - AD - controls Antimatter Decelerator.
  - REX - controls some other devices.

- Network Backbone:

Figure 1: Network Topology overview.
Switches connected directly to Data Masters create so-called network Backbone which is located in Prevessin (mainly in CCR, for REX in ISOLDE).

Switches in the Backbone (with exception of REX) are connected to a common Source of Time (SoT), we assume that all of the switches connected to the SoC are perfectly synchronized.

Switches in the Backbone (except in REX) are Timing Masters.

Connections in the Backbone are redundant.

- **Accelerator networks:**
  - Each accelerator has a dedicated network.
  - Each Data Master (DM) is connected by two links (we assume the data on both link is mirrored and synchronized) to two separate switches in the Backbone.
  - There is always interconnection between Backbone switches (requirement of RSTP).
  - The Backbone switches are connected to the actual accelerator network (i.e. for LIC to the accelerator network in Meyrin) by two separate fibers, each fiber is connected to a separate switch (these two switches are interconnected as well), thus the redundancy of the Backbone is distributed to the "site" of the accelerator. This redundancy can be used in connecting critical devices (it’s available but optional).
  - An accelerator network is mapped into VLAN.

- **Optimization and interconnection between accelerator networks:**
  - For optimization and because there needs to be communication between LIC and other Data Masters (LIC\(\Rightarrow\)LHC, LIC\(\Rightarrow\)AD, LIC\(\Rightarrow\)REX) as well as communication between LIC Data Master and nodes on other networks then LIC, all the accelerator networks are interconnected by the switches in the Backbone.
  - The principles of the interconnection (which are not really clear):
    * Each non-LIC Data Master has a single switch common with LIC Data Master.
    * The other Backbone switch of non-LIC Network is connected to the other LIC switch.

- It is important to remember that, if redundant connections are used, it introduces extra layer in the hierarchy - this needs to be taken into account.

- **Auxiliary Networks (see Figure 2):**
  - Servers - Servers in CERN Control Room (CCR) dedicated to an accelerator will be connected through a switch to the Backbone switches belonging to the accelerator’s network.
  - Labs - a single switch will be connected to all Backbone switches. It will concentrate all the VLANs, distribute it to the "Lab buildings" and then separate to mirror the real "Accelerator Network".
3 Accelerator Networks (VLANs)

- Why VLANs:
  - They allow to separate accelerator networks.
  - They are safer than Multicast groups because: If a node starts sending data to an unknown multicast group (or broadcast), the default behaviour of a switch is to flood all ports with such data. This means that a misbehaving node in LHC network can break LIC network. In VLANs, the networks are logically separated, and traffic limited.
  - VLANs operation subsums multicast pruning, so VLAN = multicast ++
  - Multicast is more end-user oriented: this is normally end-user (node) to decide to which Multicast group it belongs. While, VLANs are usually administrator-oriented: it is administrator to decide the topology of VLANs. Of course, we can make static Multicast Groups and dynamic VLANs...

- Each accelerator network is mapped into VLAN. However, this is not enough since:
– LIC control messages need to reach some of LHC/AD/REX nodes.
– There need to be communication between LIC and LHC/AD/REX Data Masters.

• Therefore, we have 3 kinds (or more if needed) of VLANs:
  – per-accelerator VLANs (VL00x, x is accelerator number): cover all the nodes belonging to a given accelerator.
  – Shared VLANs (VL1xy, x and y are accelerator numbers between which VLANs are shared): since some of the LIC’s control messages need to reach other accelerators’ nodes, we need shared VLANs. If a message is designated for LIC and LHC, it shall be sent to VL121.
  – DM-to-DM VLANs (VL2xy, x and y are accelerator numbers): a direct communication between Data Masters is needed. It shall be performed through dedicated VLANs. This ensures that access to Data Masters is limited (a node belonging to VL001 cannot send message to LIC unless it belongs to shared VLAN VL121).

• When we connect a node to the network:
– it is good to verify that it is connected to the right accelerator network (compare its configuration with the reality)
– it is sometimes needed to make auto-configuration, so it needs to auto-detect to which accelerator network it belongs.

Therefore, the Data Master shall send every some time interval a message to VL00x VLAN which enable the nodes to tell to which VLAN they belong.
4 Multicast groups

- Shall be used to address Data Master and configuration/management node:
- It enables to change the hardware (MAC) without reconfiguration of the network.
- We set single Multicast Group for all Data Master’s ports of all Data Masters of the same accelerator.
- Multicast groups are used within VLANs

5 Time distribution

- There is single source of timing, which is connected to Backbone switches.
- Each accelerator network has two Timing Masters (one of which is shared).
- It is still to be discussed whether REX network is to be treated differently and it’s Data Master located out of CCR.
- Some problems with this layout, single for the time being (timing point of few).
- **important:** A Data Master synchronizes to the Backbone switches (Timing Masters). Data Masters are assumed to have two (or more) links, so the multiple sources of timing need to be managed. One of possibilities is for a Data Master to be a WRPTP boundary clock (in terms of timing distribution: like a WR Switch). WRPTP boundary clock can manage multiple sources of timing (upstream links).
- The switches directly connected to the Backbone switches (layer 1 in the figure) should not be connected to nodes, there must be another layer of switches - layer 2. It is due to the timing-distribution redundancy constraints – it is impossible to guaranty redundant connection to the Timing Master for each switch in the layer 1 with the proposed topology layout. For this reason, each of the switches in layer 2 shall be connected to both layer 1 switches (this is depicted in the figure).

- For timing, the number of switch layers is important, with the current layout the number of layers required seems to be 4. Layer 0 is the layer of Timing Masters. Assuming that WR switch has 18 ports, layer 4 can accommodate 2048 (with full redundancy) to 4096 (no redundancy) nodes. This number does not take into account external topology restrictions (later on this).

Figure 4: Network Topology - layers (in calculations we assume a WR Switch to have 18 ports).
6 Data Master (DM)

- The DM’s features and specification depend greatly on:
  - the way it is connected with the network,
  - whether there is DM’s redundancy (yes we want and need it!) and how we handle it,
  - requirements of the network/redundancy/etc.

- This section first considers how DM’s redundancy can be handled, we present different solutions which have been taken into account and explain our choices. Finally, we present DM’s desired features resulting from this choices.

- Some names used:
  - **Accelerator Controlling Process (ACP)** - an imaginary application which runs on the Data Master. Its purpose is to control accelerator(s) by sending events aggregated into Control Messages. These messages are DM’s output to the Network. Control Messages are encoded into several Ethernet Frames.
  - **Control Data** - Control Messages.
  - **DM Monitor** - a WR node which monitors a proper functioning of DM(s).
  - **Master DM** - The Data Master which is active at a given moment – its Control Data is delivered through WR Network to accelerator devices (WR Nodes).
  - **Slave DM** - The Data Master which is backup at a given moment – its Control Data is not delivered through WR Network to accelerator devices (WR Nodes). However, the Accelerator Controlling Process running on the Slave DM ”thinks” that it is actually controlling the accelerators and send the Control Data – the data is blocked (later on it) from getting to the network.
  - **DM-switching window (no-message-sent window)** a time during which DMs are not sending Control Data. It is foreseen as a time during which the switch between redundant DMs takes place (detailed description in the following subsection).
  - **Layer 0 Switches** these are WR Switches which are directly connected to DM(s) and create network backbone.

- The redundant DMs’ topologies considered and basic ideas:
  - See Figure 5.
  - Control Data from only one of the DMs shall get to the Accelerator Network(s) at a time.
  - All DM(s) shall receive data from the Network at all time.
  - Switching between redundant DMs shall introduce no disruption to normal functioning of the accelerators and it shall be possible to perform it on-the-fly.

6.0.1 Normal operation in redundant DM’s configuration

- **One DM (ex. DM-A)** is the active (Master DM): it sends through its $P_1,...,P_N$ ports the Control Data (mirrored data on all ports). It also accepts data on all ports (redundancy of data shall be handled somehow).

- **The second DM (ex. DM-B)** is inactive (Slave DM). Its Control Data does not get to the accelerator network(s). The data is actually sent to the ports by the Accelerator Controlling Process but it’s blocked independently of this process. The process ”thinks” that it is controlling accelerators.

- **Both DMs (DM-A and DM-B)**:
  - accept input data, this data is needed by accelerator control process.
- synchronize with the Timing Master (WRPTP communication is allowed on the Slave DM).
- have port $P_{N+1}$ which is used for control/configuration.

- A DM Monitor can be used to verify a proper functioning of the Slave DM, if necessary, before switching the DMs

### 6.0.2 Issues regarding switching between DMs

- When considering the architecture and operation of redundant DMs, the following issues need to be taken into consideration:
  - **Where** to perform the re-configuration which results in switching between the DMs (on the DMs, on the switches connected to DMs)?
  - **When and how** to switch between DMs to make sure that during the process we don’t break/loose/duplicate Control Data (preferably we avoid breaking/loosing/duplicating Ethernet Frames either).
  - **How to verify** that the (backup) Slave DM is working properly.

- It is important to remember that Master and Slave DMs are identical. Both can take up any role (Master/Slave).

- **Where** to switch between redundant DMs (make the reconfiguration):
  - basically, the idea is that the Accelerator Controlling Process which is running on the (backup) Slave DM can send out the Control Data at any time as though it was really controlling accelerators. This output data is blocked somewhere and the process is unaware of that. The input data is not blocked. This is to allow:
    - running the Accelerator Controlling Processes simultaneously on both Master and Slave DMs at all time.
    - verifying Accelerator Controlling Process running on the Slave Master before switching to it.
  - The Control Data from the Slave DM can be blocked:
    - **on the outputs of the backup DM.**
      - Pros: no extra features needed in the switches (i.e. time-triggered configuration).
      - Cons: we want to switch to backup DM if the active DM is misbehaving: e.g.: there is no communication with it, in such case a flawless switch-over would be impossible because we would not be able to disable outputs of the active DM, in the effect we would have mess.
    - **on the inputs of the Layer 0 switches.**
      - Pros: independent of DMs, so we can cut-off a malfunctioning DM even if we have no communication with it. We avoid adding yet another functionality to the DM.
      - Cons: we need special functionality in the switches, i.e.: time triggered reconfiguration.
  - Based on the above: delegating the process of switching (configuration change) between DMs onto the Layer 0 switches, seems to be better.

- **When and how** to switch between redundant DMs (Master and Slave): We propose to introduce **time-triggered reconfiguration** which is performed during a time window called **DM-switch window** (or no-message-sent window).
  - This solution can work for both: DM’s and switch’s reconfiguration, however we’ve chosen Layer 0 Switch-based reconfiguration, therefore...
  - Switching between redundant DMs means time-triggered reconfiguration of all the Layer 0 Switches at the same moment being:
    - disabling ports connected to the Master DM(active),
enabling ports connected to the Slave DM (backup)

- **Time-triggered reconfiguration:**
  - It shall be possible to request WR Switch to reconfigure on a particular time.
  - This is additional WR-specific feature which shall be implemented on the WR Switch.

- **DM-switching window** (or no-message-sent window):
  - It is the time during which the DMs are not sending Control Data so that a safe switch between redundant DMs can take place.
  - The window is important only for the Control Data sent by DM. No usage of the window for other types of communication, i.e. Node(s)-to-DM, was found.
  - The window is needed to avoid different forwarding behavior of Layer 0 Switches on the moment of their reconfiguration. In other words: even if we change configuration in a perfectly synchronous way on different switches (to disable old DM and enable new DM), if the change happens during forwarding process of the Control Data, the effect on the forwarding traffic behavior can be different for different switches.
  - The window should be of appropriate length which is to be calculated/estimated (my work).
  - The window should be always present, it can be rare, e.g.: every 1s or even more. The idea of opening it only before the reconfiguration has been rejected because:
    - One of the possible reasons of changing the DMs is the problem with communication. In such case, we would not be able to open the window.
    - It would complicate the process of DM switching - synchronization between DMs and Layer 0 Switches would be required.
    - It is easier to design the Accelerator Controlling Process if the behavior is predictable – the window is always there.

- **How to verify** that the Slave (backup) DM is working properly: we suggest introducing a DM Monitor:
  - It receives Control Data from the Slave DM and verifies that everything works fine.
  - It has not been investigated yet how the verification process would look like, e.g. the DM Monitor can compare data from active and backup DMs.
  - We considered 3 possible solutions of connecting the DM Monitor, DMs and Layer 0 Switches together.
    These solutions are described in detail in the next sub-section.

6.0.3 **Consideration of different topologies/solutions for redundant DMs**

- The topologies and solutions vary depending on:
  - where and how the DM Monitor is connected.
  - where the Control Data from the Slave DM is blocked.

- We present three most seriously considered solutions.

- **Solution 1**: connecting DM Monitor directly to the Data Master(s):
  
  - Figure 5: Solution 1.
  - How switching of DMs works (can be done either on the DMs or Layer 0 Switches):
    1. Starting status:
      - Accelerator Controlling Process sends Control Data on Master and Slave DMs in the same manner (to appropriate accelerator VLAN(s)).
      - Control Data send by the Slave DM is blocked (either on the DM or the switches)
2. Verification of Slave DM - can be done at any time without any extra configuration.
3. Switch DMs: set time-triggered re-configuration to block Control Data sent by Master DM and un-block the one sent by Slave DM.
   - Pros: simple solution, DM can be completely unaware of the re-configuration, no extra layer of switches (like in solution 3).
   - Cons: it requires additional port on the DM to connect DM Monitor, so the DM becomes more a more multi-port device which is closer to the switch then a node.

- **Solution 2:** connecting the DM Monitor to the Layer 0 Switch(s) and defining a special testing VLAN (t-VLAN) to be used for the Slave DM verification (Control Data from the Slave DM is sent to t-VLAN).
  - Figure 5: Solution 2.
  - How switching of DMs works (involves both DM and Layer 0 Switches):
    1. Starting status:
       - The Master DM sends Control Data to Accelerator VLAN(s)
       - Port(s) connected to Master DM are enabled by the Switch(es) Layer 0
       - Port(s) connected to Slave DM are disabled by the Switch(es) Layer 0
    2. Start verifying Slave DM
       - Set the port of the Switch(es) Layer 0 which is connected to the DM Monitor to allow t-VLAN (this can be constantly in place).
       - Set Slave DM to send data to t-VLAN.
       - Enable port(s) connected to Slave DM on the Switch(es) Layer 0.
    3. Stop verifying backup DM: disable port(s) connected to the Slave DM
    4. Switching DMs
· Set the Slave DM to send Control Data to Accelerator VLAN(s)
· Optional: set the port of the Layer 0 Switch(es) connected to the DM Monitor to Accelerator VLAN (to enable further control of the DM)
· Set the time-triggered reconfiguration of Layer 0 Switch(es) to disable the port(s) connected to Master DM and enable the port(s) connected to Slave DM.

– pros: no additional DM ports are needed (like in the previous solution), neither extra layer switches (like in the next solution).
– cons: the process of switching is complicated and needs to be conducted on both: switches and DMs.

- Solution 3: introducing additional layer of switches (call it Layer -1) between the Layer 0 Switch(es) and DMs in order to connect the DM Monitor:

  – Figure 5: Solution 3.
  – How switching of DMs works (can be done only by reconfiguration of the Layer 0 Switches):

    1. Starting status:
       · Accelerator Controlling Process sends Control Data on Master and Slave DMs in the same manner (to appropriate accelerator VLAN(s)).
       · Control Data sent by the Slave DM is blocked by Layer 0 Switches – it means that it is forwarded to the DM Monitor which is connected to Layer -1 Switches

    2. Verification of Slave DM - can be done at any time without any extra configuration.

    3. Switch DMs: set time-triggered configuration change to block Control Data sent by Master DM and unblock the one sent by Slave DM.

– Pros: no extra port needed on the DM, no special requirements/functionalities/procedures,
– Cons: any addition layer of switches is bad!

6.0.4 The final proposition of redundant DM layout and DM’s features

- Data Master:

  – It has a number (N) of mirrored WR-compatible ports(N ≥ 3: P₁, P₂,..., Pₙ):
    * These ports send perfectly synchronized control data to the network.
    * Reception of the redundant packages shall be handled (somehow :).
    * Time shall be retrieved on such ports: one of the ports shall be a primary WRPTP Slave port, the other(s) shall be secondary WRPTP Slave port(s) or WRPTP Master(s).

  – WRPTP-wise Data Master is a boundary clock.

  – There should be an additional (not necessarily WR-compatible) independent port for control/configuration/synchronization: Pₙ₊₁.

  – The Accelerator Controlling Process running on the DM is completely “unaware” whether it is controlling accelerators or not. The re-configuration (switching between redundant DMs is delegated to Layer 0 Switches).

- DM Monitor is connected directly to the DM (to port Pₙ, n ≤ N).

- The Control Data outputted by the Slave DM is blocked by the Layer 0 Switches.

- Switching between DMs is done by performing a time-triggered re-configuration of the Layer 0 Switches, i.e.:

  – Blocking input of the ports connected to the Master DM.
  – Unblocking inputs of the ports connected to the Slave DM.
• Blocking/unblocking of the ports on the switches shall affect only input data traffic (i.e. WRPTP, BPDU and other configuration Ethernet Frames shall be allowed).

• Time-triggered re-configuration:
  – The switching takes place in HW of switches.
  – It is a WR-specific functionality of WR Switches which will be handled by SNMP\(^1\) protocol using WR-MIPs\(^2\).
  – It shall enable for atomic reconfiguration of WR Switch which takes place within a single clock cycle on a precise time (precision of 1\(\mu\)s).

• Such solution is scalable: more then two DMs can be connected.

• Such solution is flexible: it allows for automatic switching between redundant DMs online (we don’t really want it but it’s an option), but it also allows for human-controlled switch-over.

• The Data Master is more then a node, it’s more a switch:
  – It’s good if it’s running WRPTP.
  – It’s good if it’s running RSTP so it can be a tree root.

\(^1\)Simple Network Management Protocol
\(^2\)Management Information Base
7 Timing Master (very preliminary draft)

- There must be a single source of time (SoT), connecting two or more sources of perfectly synchronized SoTs is not acceptable:
  - If one of the SoT breaks in the way it starts drifting (ex. problems with GPS) a part of the system starts drifting, even if we detect its failure (drifting) and we want to re-synchronize to the part of the system connected to it to the working source, we need to jump in time - this is even worse.
  - If we have single source of time, even if it drifts (because there is ex. problem with GPS), the entire system drifts which is acceptable.

- The source of time comprises of GPS, Cesium Clock and Time Monitor.
  - GPS.
  - Cesium Clock.
  - Time Monitor:
    * It regenerates PPS/10MHz signals based on GPS and Cesium Clock (??).
    * Controls signal from GPS with the Cesium Clock signal (checks it against drifting).
    * If GPS fails, it provides the time with Cesium/internal clock
    * Allows to synchronize with the backup Timing Master (measure the offset between the two TMs and apply it to the regenerated clock by the backup TM).
    * Provides means of alarming administration about the timing problems.
    * Performs UTC sanity check (ex. with NTP).
8 I’m alive

- To enable fast detection of lost connection (with DM), each Data Masters shall send to it’s shared VLANs a message "I’m alive" (IA) every fixed time interval.
- The time interval reflects the system’s responsiveness - the more often IA message is send the faster the connection loss is detected.
- "I’m alive" messages does not solve the problem of detecting a loss of timing (connection to the Timing Master or Source of Timing).

9 Node

- A node needs to be able to detect a loss of connection to the DM(s):
  - A node configures a counter to be re-setted by reception of "I’m alive"(IA) message.
  - A node can follow IA messages from more then one Data Master.
  - A node should be able to produce a pulse when a connection with a chosen DM is detected.
Figure 7: WR CERN Control and Timing Network Layout (in calculations we assume a WR Switch to have 18 ports).
11 Issues

- What to do when a switch loses connection with Timing Master ??
- Connection of WR Data Master by SMP to Beam Interlock System (converter GMT-to-WR or WR BIS node).
- We need to detect two types of system malfunctions:
  - No connection to Data Master, so no control messages - this is detected by not reception of "I’m alive" messages.
  - No connection to Timing Master - this problem overlaps with the previous but is not always equivalent: we can have Control Data but no Timing (I need to find use case here because I’m not 100% sure).
- Migration.

- The fact that each WR Card (node) is to be connected directly to the Switch will add extra layers to the topology. If it was possible to have single WR receiver (node) per crate, and to distribute the timing internally in a crate, it would be much better network-layer-wise.

- The initial estimations of the required number of inputs to the Backbone switches show that two backbone switches per-accelerator network (one of which is shared) are not enough for the needs of LHC network. Two possible solutions exist:
  - Vertical - Increase the number of layers (instead of connecting 8 points directly to Backbone switches, connect single switch and then 8 points to that switch.
  - Horizontal - Increase the number of outputs of the LHC Data Master.

12 Conclusions

- Network redundancy is useful and should be optional so that it can be make available for the most critical parts/elements of the network - the suggested network layout enables optional application of redundant elements. To make redundancy available for the accelerator network(s), the network’s backbone needs to be fully redundant.
  
  
  