White Rabbit: a next generation synchronization and control network for large distributed systems

Maciej Lipiński

Hardware and Timing Section / CERN
Institute of Electronic Systems / Warsaw University of Technology

presented by Rodney Cummings
May 2012

2012-05-11

White Rabbit: a next generation synchronization and control network for large distributed systems

Maciej Lipiński

Hardware and Timing Section / CERN
Institute of Electronic Systems / Warsaw University of Technology

presented by Rodney Cummings
May 2012

Many thanks to Rodney Cummings for agreeing to present White Rabbit on my behalf !

Maciej

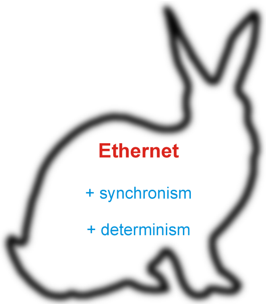
Outline

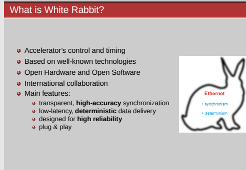
Outline
Introduction
CERN Control System
White Rabbit
Time Distribution
Data Distribution
WR-based Control System
AVB gen2 vs. WR
Summary

1. First, very short introduction to the White Rabbit project
2. Then, some basics on how the accelerators are controlled in order to better understand the requirements. At the end of this section, the requirements of accelerator control and timing, which abstract from transportation layer, are presented.
3. Then, explanation of how WR has solved (in case of timing) or is intending to solve (in case of data) some problems/challenges to meet the above requirements.
4. A description of how WR is intended to be used for controlling accelerators.
5. Comparison of the ideas introduced in WR and in AVB gen 2
6. Summary and conclusions

What is White Rabbit?

- Accelerator's control and timing
- Based on well-known technologies
- Open Hardware and Open Software
- International collaboration
- Main features:
  - transparent, **high-accuracy** synchronization
  - low-latency, **deterministic** data delivery
  - designed for **high reliability**
  - plug & play





White Rabbit is a project which was started to develop next generation control and timing network for the CERN accelerator complex. Later, another accelerator facility (GSI in Germany) joined the project. Currently, the project is a collaboration of many institutes and companies around the world. The project is both, open hardware and opens software. Any company can download sources and produce WR gear.

The projects aims at creating an Ethernet-based network with low-latency, deterministic data delivery and network-wide, transparent, high-accuracy timing distribution. The White Rabbit Network (WRN) is based on existing standards, namely Ethernet, Synchronous Ethernet and PTP.

Why White Rabbit?

- White Rabbit started within renovation effort of General Machine Timing (GMT) – the current control and timing system at CERN
- GMT works great but has some disadvantages:
  - Based on RS-422, low speed (500kbps)
  - Unidirectional communication (controller$-$ $>$end stations)
  - Separate network required for end station$-$ $>$controller communication
  - Custom design, complicated maintenance
- White Rabbit is meant to solve these problems

Why White Rabbit?
- White Rabbit started within renovation effort of General Machine Timing (GMT) – the current control and timing system at CERN
- GMT works great but has some disadvantages:
  - Based on RS-422, low speed (500kbps)
  - Unidirectional communication (controller$-$ $>$end stations)
  - Separate network required for end station$-$ $>$controller communication
  - Custom design, complicated maintenance
- White Rabbit is meant to solve these problems

CERN started thinking about a suitable successor for the timing system of the LHC injectors in 2006. The main drawbacks of the current system (General Machine Timing) are its limited bandwidth (500 kb/s on a multi-drop RS-422 line) and its lack of bi- directionality. Limited bandwidth results in an otherwise unnecessary proliferation of different timing networks for each accelerator, and this extra complication propagates through low-level software making maintenance harder. Lack of bi-directionality has two main disadvantages:

- *Stand-alone timing receivers, though requested by clients, cannot be designed because there is no way to read status information back from the cards remotely.*
- *Cabling delay compensation cannot be automated. Instead, manual calibration using traveling clocks is used, resulting in manpower-intensive error-prone campaigns.*

At the same time, GSI began brainstorming about the timing system for the FAIR facility, and since other collaborations with CERN were already underway it seemed natural to try to come up with a single timing system which served both sets of requirements. The similarities of the two complexes in terms of timing precision and sequencing needs helped in this regard. The requirement for a high-bandwidth full-duplex link quickly resulted in the choice of Ethernet for the physical layer. Indeed, Ethernet is not only a very high-performance and well known solution but also one where long-term support is beyond doubt, and this was an important requirement for both CERN and GSI.

Reference [8]

When/where White Rabbit?

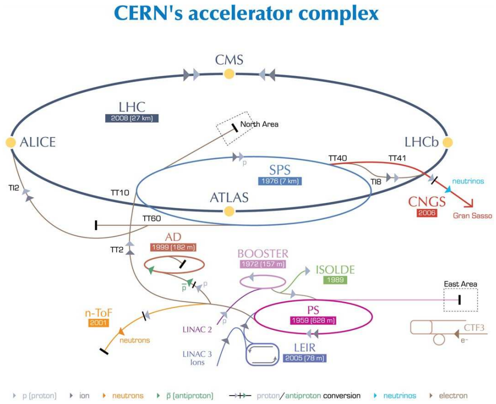
- Current status of the project:
  - **NOW:** WR-based timing installation deployed for CERN Neutrino to Gran Sasso (CNGS) project
  - **End of 2012:** (commercial) release of (basic) WR Switch
- Foreseen applications: CERN, GSI (Darmstadt, Germany) and DESY (Gamma-Ray and Cosmic-Ray experiment)
- Potential applications: Cherenkov Telescope Array, The Large High Altitude Air Shower Observatory, The Cubic Kilometre Neutrino Telescope

When/where White Rabbit?
- Current status of the project:
  - **NOW:** WR-based timing installation deployed for CERN Neutrino to Gran Sasso (CNGS) project
  - **End of 2012:** (commercial) release of (basic) WR Switch
- Foreseen applications: CERN, GSI (Darmstadt, Germany) and DESY (Gamma-Ray and Cosmic-Ray experiment)
- Potential applications: Cherenkov Telescope Array, The Large High Altitude Air Shower Observatory, The Cubic Kilometre Neutrino Telescope

- See project website for news
  www.ohwr.org/projects/white-rabbit
- See presentations from the latest workshop for the latest developments
  www.ohwr.org/projects/white-rabbit/wiki/Mar2012Meeting
- See a list of potential users
  www.ohwr.org/projects/white-rabbit/wiki/WRUsers

CERN

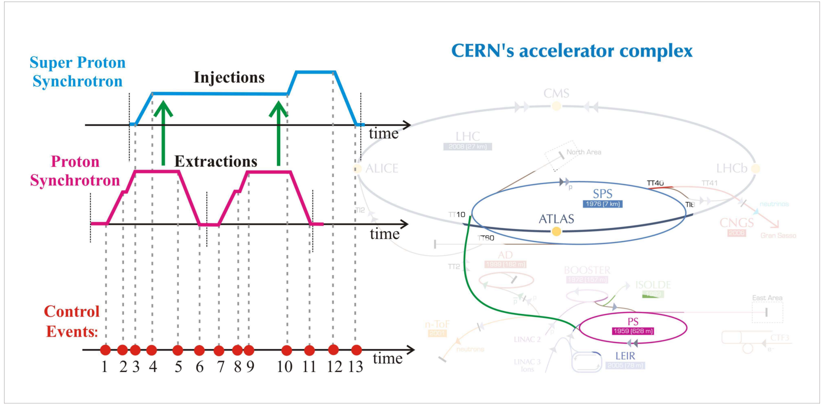
- 6 accelerators
- LHC: 27km perimeter
- Thousands of devices to be controlled and synchronized
- A huge real-time distributed system



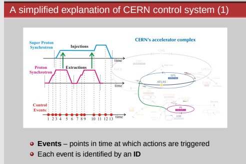
CERN
6 accelerators
LHC: 27km perimeter
Thousands of devices to be controlled and synchronized
A huge real-time distributed system

CERN is a complex of 6 circular and some linear accelerators which are interconnected. The biggest accelerator is the Large Hadron Collider (LHC) which is 27 km long. All the devices which serve the accelerators (magnets, kickers, etc) need to be precisely synchronized and controlled by a central control system.

A simplified explanation of CERN control system (1)

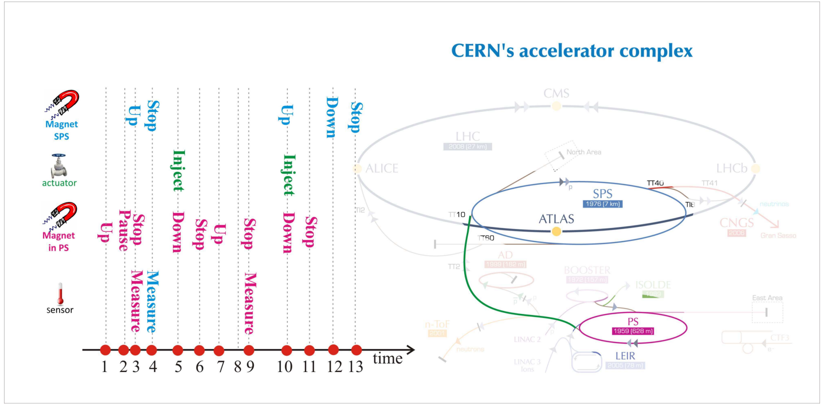


- **Events** – points in time at which actions are triggered
- Each event is identified by an **ID**

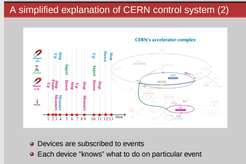


- Each accelerator ramps up the energy/speed of particles (a beam)
- Once a desired energy is achieved, the beam is injected to a more powerful accelerator and the process is repeated
- A process of ramping up energy is done in cycles
- In each cycle, different devices at different points in time need to perform various activities
- This activities are triggered by events
- Events are identified by ID
- Figure:
  - Pink color - Proton Synchrotron, PS (left and right)
  - Blue color - Super Proton Synchrotron,SPS (left and right)
  - Green color - injection line between PS and SPS (left and right)

A simplified explanation of CERN control system (2)

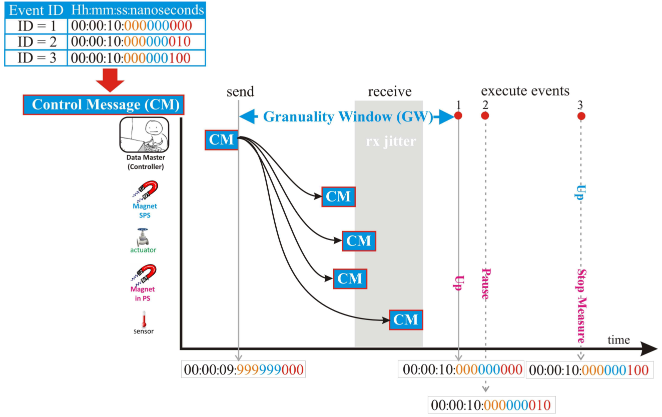


- Devices are subscribed to events
- Each device "knows" what to do on particular event

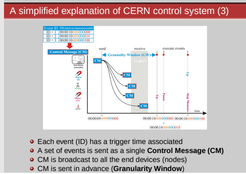


- Many devices (in different accelerators) can subscribe to the same event (ID)
- The action taken by the device on the reception of a particular event is defined in each device
- E.g.: one can see in the figure that on event ID=3:
  - a sensor starts takes measurement
  - ramping of magnet energy is stopped in PS
  - ramping of magnet energy is started in SPS

A simplified explanation of CERN control system (3)



- Each event (ID) has a trigger time associated
- A set of events is sent as a single **Control Message (CM)**
- CM is broadcast to all the end devices (nodes)

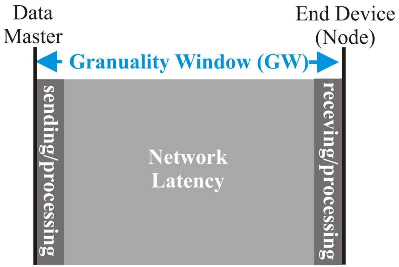


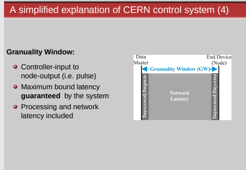
- Each event has a trigger time(stamp) associated with it
- Many events are accumulated into Control Messages which are sent every Granularity Window (e.g. 1ms)
- Control Messages are sent by a controller device (so called **Data Master**).
- Control Messages are be sent at least Granularity Window in advance with regards to the trigger/execution time of the events they contain
- The arrival time of Control Messages is different in different devices (jitter) but the underlying network must guarantee it to be within the Granularity Window

A simplified explanation of CERN control system (4)

Granuality Window:

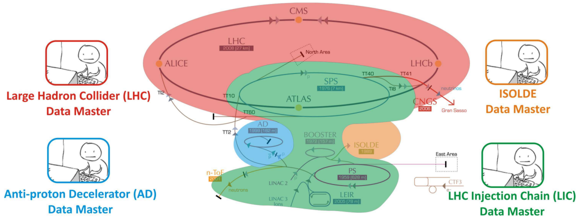
- Controller-input to node-output (i.e. pulse)
- Maximum bound latency **guaranteed** by the system
- Processing and network latency included



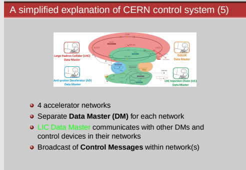


- The Granularity Window includes not only the network latency, but also the processing and transmission/reception delays
- In other words: Granularity Window is the time between the moment a controller ”says” send Control Message, and the time an impulse (physical signal) is triggered by an end device (node) on an event.

A simplified explanation of CERN control system (5)



- 4 accelerator networks
- Separate **Data Master (DM)** for each network
- LIC Data Master communicates with other DMs and control devices in their networks
- Broadcast of **Control Messages** within network(s)



- Accelerators at CERN are grouped into 4 accelerator networks which should be as independent as possible but interaction between them is inherent
- Devices connected to each accelerator network are controlled (mostly) by it's own Data Master
- Since LIC provides beam to LHC/AD/ISOLDE, the LIC Data Master sometimes needs to control devices connected to other networks
- Communication between Data Masters is necessary
- All the devices in a given accelerator network need to receive all the Control Messages sent to this network (the device decides how/whether to use events from the Control Message)
- Some Control Messages need to be send on more then one accelerator networks (the case when LIC Data Master controls devices in more then one network during injection)

Control network requirement

Requirement	Case 1	Case 2
Synchronization accuracy	**sub-ns**	
Synchronization precision	**picoseconds**	
Granuality Window (GW)	**1000us**	**200us**
Network span	**10km**	**2km**
End device number	**2000**	
Control Message size	**1500-5000 bytes**	$<$ **1500 bytes**
Control Message frequency	**every GW**	
Data Master number	**4**	**1**
Traffic characteristics	**one-to-alot**	
Number of CM lost per year	**1**	

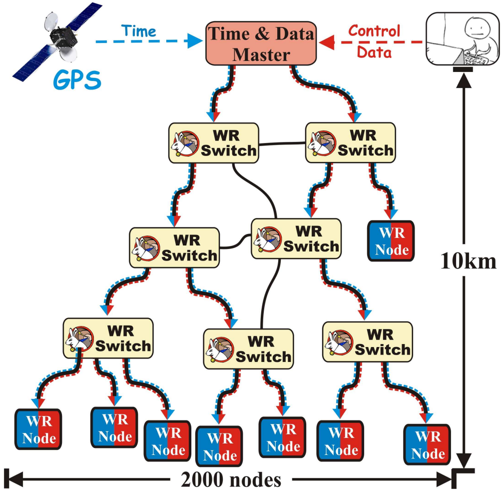
Control network requirement

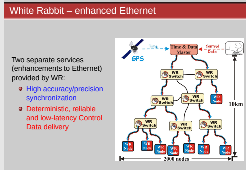
Two use cases of control and timing system requirements (quite abstracted from the underlying medium). This is because White Rabbit will be used in two accelerator facilities (GSI and CERN). The control principals in both facilities are similar but the parameters (i.e. size) are different.

White Rabbit – enhanced Ethernet

Two separate services (enhancements to Ethernet) provided by WR:

- High accuracy/precision synchronization
- Deterministic, reliable and low-latency Control Data delivery



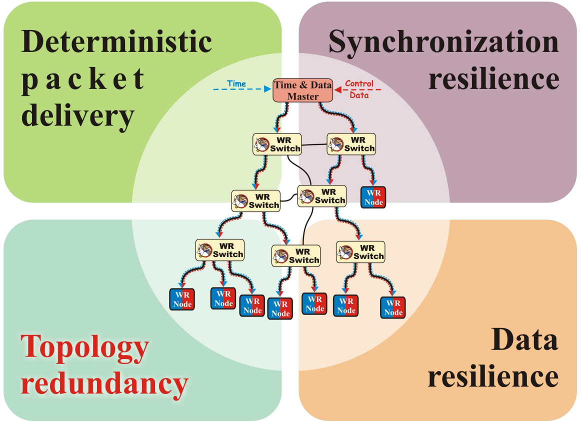


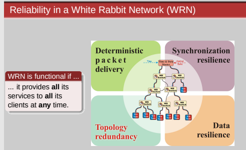
WR should be treated as a standard Ethernet with additional (optional) features (characteristics) which should be used if one needs them. These two features should be treated separately/independently.

Reliability in a White Rabbit Network (WRN)

WRN is functional if ...

... it provides **all** its services to **all** its clients at **any** time.





An accelerator in order to work needs all it's devices to perform appropriate actions (react to events at appropriate time) and any device can perform action at any event. Therefore, WR Network is considered functional only if **all** the devices are synchronized with required accuracy/precision and **all** the devices receive **each** Control Message **within** the Granularity Window. The problem of reliability in WR has been divided into sub-domains which can be handled independently but need to interact.

Time Distribution in White Rabbit Network

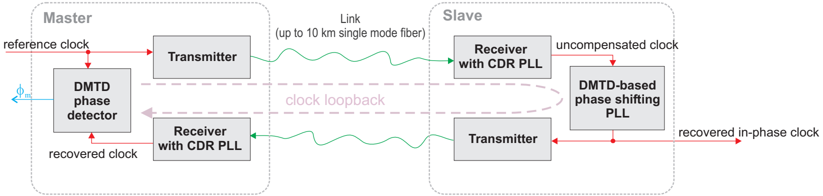
Time Distribution in White Rabbit Network
(Working solution, first deployment underway)

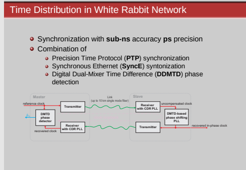
Time Distribution in White Rabbit Network
Time Distribution in White Rabbit Network
(Working solution, first deployment underway)

Timing distribution in WR is a working solution which is being currently deployed in Cern Neutrino to Gran Sasso (CNGS) experiment.

Time Distribution in White Rabbit Network

- Synchronization with **sub-ns** accuracy **ps** precision
- Combination of
  - Precision Time Protocol (**PTP**) synchronization
  - Synchronous Ethernet (**SyncE**) syntonization
  - Digital Dual-Mixer Time Difference (**DDMTD**) phase detection





- Highlights
  - Auto-calibration of link asymmetry
  - Enhanced precision of HW timestamps (picoseconds level)
  - High accuracy only with WRPTP-capable devices
  - Interoperability with (virtually) default PTP Profile - hybrid networks possible
  - Synchronization topology independent from data topology
- Extension to PTP – defined as PTP Profile: **WRPTP**
  - delay-request two-step mechanism
  - modified BMC (reliability-oriented)
  - Mapping onto Ethernet
- Figure
  - The slave recovers frequency (SyncE) and performs (if necessary phase-shift (based on the data from PTP) to align phase with that of master. The recovered and phase-shifted frequency is looped back to the master

PTP is OK but ...

What are the issues...	and	... how we address them
PTP-base syntonization	⇒	SyncE
limited precision and resolution	⇒	SyncE DDTMD phase detection
unknown link asymmetry	⇒	SyncE DDTMD phase detection WR Link Delay Model

WR extension to PTP (**WRPTP**) for
extra data exchange and logic

PTP is OK but ...
What are the issues... and ... how we address them
PTP-base syntonization ⇒ SyncE
limited precision and resolution ⇒ SyncE DDTMD phase detection
unknown link asymmetry ⇒ SyncE DDTMD phase detection WR Link Delay Model
WR extension to PTP (WRPTP) for extra data exchange and logic

- We free PTP from governing the frequency fluctuations by using Synchronous Ethernet (a known solution)
- The common notion of frequency in the entire network (thanks to SyncE) is used to increase the precision of timestamping. This is possible by casting the problem of time measurement into phase detection. Phase detection is possible with picoseconds precision.
- White Rabbit provides means to evaluate link asymmetry due to transmission/reception hardware and physical link asymmetry (wavelengths, when used over single fiber for two-way communication)
- These improvements, in order to work, require additional logic and data exchange. And, this is precisely why the extension to PTP is proposed - to accommodate these implementation improvements.

Time Distribution in White Rabbit Network

Data Distribution in White Rabbit
(Solutions in a design or implementation phase)

Time Distribution in White Rabbit Network
Data Distribution in White Rabbit
(Solutions in a design or implementation phase)

The solutions for deterministic and reliable data distribution are scheduled to be ready for preliminary tests through 2013

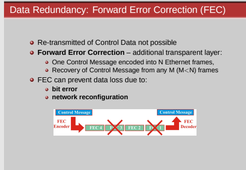
Control Data

- Two types of data transported over White Rabbit Network:
  - **Control Data** (High Priority, HP) - priority 7 and broadcast
  - Standard Data (Best Effort) - all the rest
- Characteristics of Control Data
  - Control Messages sent by Data Master(s)
  - Deterministic and low latency delivery (within GW)
  - Broadcast (one-to-alot)
  - Reliable delivery (one Control Message lost per year)

Control Data
- Two types of data transported over White Rabbit Network:
  - Control Data (High Priority, HP) - priority 7 and broadcast
  - Standard Data (Best Effort) - all the rest
- Characteristics of Control Data
  - Control Messages sent by Data Master(s)
  - Deterministic and low latency delivery (within GW)
  - Broadcast (one-to-alot)
  - Reliable delivery (one Control Message lost per year)

Data Redundancy: Forward Error Correction (FEC)

- Re-transmitted of Control Data not possible
- **Forward Error Correction** – additional transparent layer:
  - One Control Message encoded into N Ethernet frames,
  - Recovery of Control Message from any M (M<N) frames
- FEC can prevent data loss due to:
  - **bit error**
  - **network reconfiguration**



- Broadcast and low latency characteristics of Control Data prevents from using re-transmission for reliable delivery
- Example FEC encoding: **500 bytes** frame encoded into **4 of 288 bytes** frames, **any 2 frames** enable recovery of the original message

Topology Redundancy: eSTP or eLACP

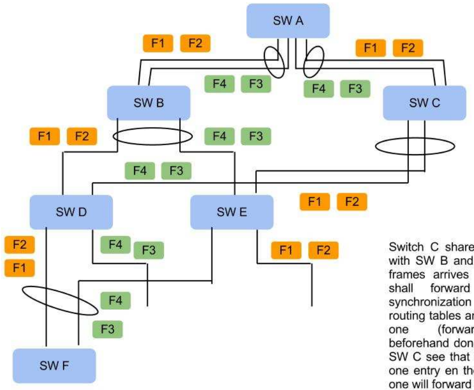
- Two ideas:
  - enhanced (Multiple/Rapid) Spanning Tree Protocol (eRSTP)
  - enhanced Link Aggregation Control Protocol (eLACP)
- FEC + eRSTP/eLACP = seamless redundancy (FEC used in both ideas)
- Redundant data received in end stations
- **Solutions take advantage of broadcast characteristic of Control Data traffic (within VLAN)**

Topology Redundancy: eSTP or eLACP

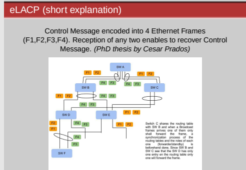
- Two ideas:
  - enhanced (Multiple/Rapid) Spanning Tree Protocol (eRSTP)
  - enhanced Link Aggregation Control Protocol (eLACP)
- FEC + eRSTP/eLACP = seamless redundancy (FEC used in both ideas)
- Redundant data received in end stations
- **Solutions take advantage of broadcast characteristic of Control Data traffic (within VLAN)**

eLACP (short explanation)

Control Message encoded into 4 Ethernet Frames (F1,F2,F3,F4). Reception of any two enables to recover Control Message. *(PhD thesis by Cesar Prados)*



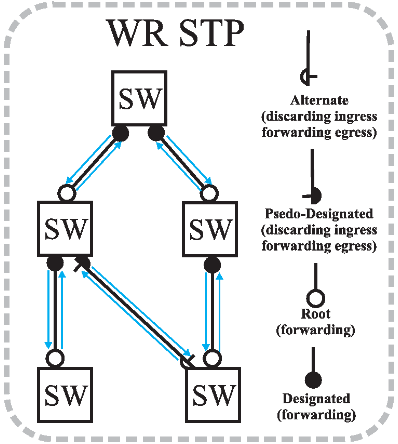
Switch C shares the routing table with SW B and when a Broadcast frames arrives one of them only shall forward the frame, a synchronization process of the routing tables and the roles of each one (forwarder/standby) is beforehand done. Since SW B and SW C see that the SW D has only one entry en the routing table only one will forward the frame.

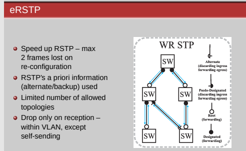


- So far it's a paper-base idea – implementation effort within a year
- LACP provides the means to show two networks interface as a single interface
- When the network interfaces are not in the same physical switch a multipath is created for a frame
- Normally the multipath is used for balancing the traffic
- On the other hand, the FEC distributes a single Control Message in several Ethernet frames
- What if we send $X\%$ of this FEC frames in one path and $Y\%$ in the alternative path
- Even if one of the paths breaks, we can still recover Control Message based on the frames delivered through an alternative path.

eRSTP

- Speed up RSTP – max 2 frames lost on re-configuration
- RSTP's a priori information (alternate/backup) used
- Limited number of allowed topologies
- Drop only on reception – within VLAN, except self-sending

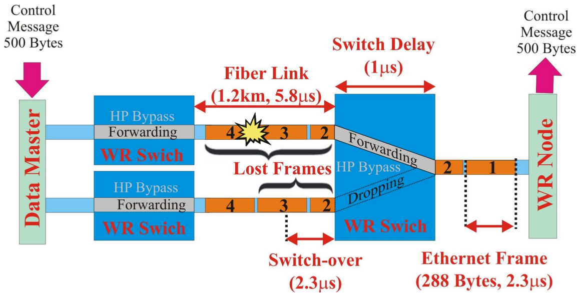


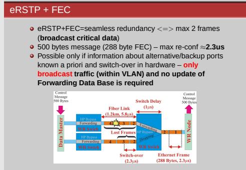


- The switch-over (re-configuration) needs to be fast enough to loss not more then 2 Ethernet Frames, so we can recover the original Control Message encoded with FEC
- In simple tree-like topologies, the information provided by "standard" RSTP is sufficient to perform instant HW-supported switch-over between redundant ports in case of link/switch failure.
- Unlike in a standard RSTP, we forward frames to ports in both alternate and designated states (within VLAN)
- Frames are dropped only on reception (so they are always available to be forwarded)

eRSTP + FEC

- eRSTP+FEC=seamless redundancy <=> max 2 frames (**broadcast critical data**)
- 500 bytes message (288 byte FEC) – max re-conf ≈**2.3us**
- Possible only if information about alternative/backup ports known a priori and switch-over in hardware – **only broadcast traffic (within VLAN) and no update of Forwarding Data Base is required**





- The size of a FEC-encoded frame translates directly into required max re-configuration time
- The order of u-seconds achievable only re-configuration done in hardware based on a priori information of alternative/backup ports
- Figure shows the simplest possible case

eRSTP: semi-automatic re-configuration

- In critical networks alternate/backup paths and root switches should be known in advance (i.e. in RSTP: through proper priority configuration and network topology)
- Two types of re-configuration:
  1. Foreseen: know what to do immediately – safe
  2. Unforeseen: most probably causes "long" data delivery disruption and network-wide re-configuration– unsafe
- Perform the re-configuration of type (1) immediately but simulate results and ask for management acknowledgement for reconfiguration type (2)

eRSTP: semi-automatic re-configuration

- In critical networks alternate/backup paths and root switches should be known in advance (i.e. in RSTP: through proper priority configuration and network topology)
- Two types of re-configuration:
  1. Foreseen: know what to do immediately – safe
  2. Unforeseen: most probably causes "long" data delivery disruption and network-wide re-configuration– unsafe
- Perform the re-configuration of type (1) immediately but simulate results and ask for management acknowledgement for reconfiguration type (2)

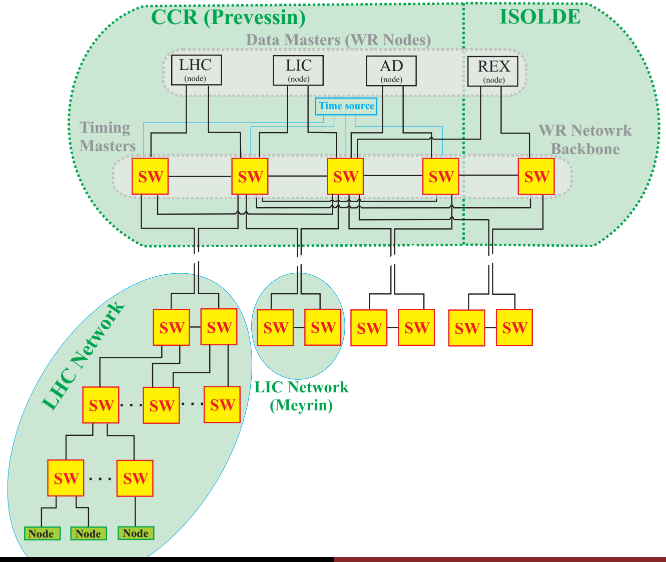
WR-based Control System

WR-based Control System
(a concept)

WR-based Control System
WR-based Control System
(a concept)

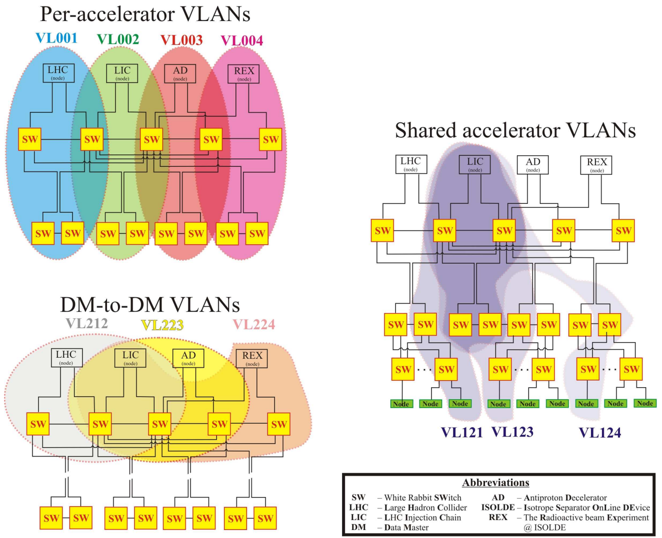
This is an outcome of a brainstorming of possible White Rabbit based control and timing network arrangement at CERN (see reference [4] for details)

Accelerator Networks



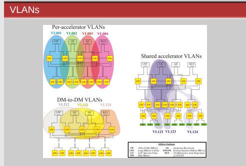
- 4 Data Master connected to a fully redundant (local) backbone
- Except the backbone, accelerator networks are separated (physically), but it's not required
- Redundancy is distributed to different points in accelerator (connection between L-1 (backbone) and L-2 switches – this is measured in kilometers
- In different points of accelerators, the network redundancy is used if required
- Timing-wise: each (except one) of the backbone switches is a grandmaster (timing redundancy)

VLANs



Abbreviations

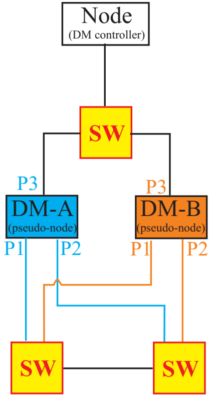
SW	– White Rabbit SWitch	AD	– Antiproton Decelerator
LHC	– Large Hadron Collider	ISOLDE	– Isotrope Separator OnLine DEvice
LIC	– LHC Injection Chain	REX	– The Radioactive beam Experiment @ ISOLDE
DM	– Data Master		

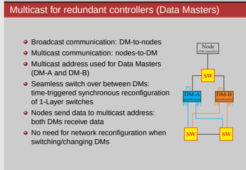


- VLANs are used extensively to logically separate accelerator networks and different streams (i.e. communication between Data Masters)
- We define:
  1. Per-accelerator VLANs : defines accelerator network (LIC, LHC, AD, ISOLDE)
  2. Shared accelerator VLANs: to enable LIC Data Master to send events to many accelerator networks
  3. DM-to-DM VLAN: to enable reliable communication between DMs (point-to-point streams)

Multicast for redundant controllers (Data Masters)

- Broadcast communication: DM-to-nodes
- Multicast communication: nodes-to-DM
- Multicast address used for Data Masters (DM-A and DM-B)
- Seamless switch over between DMs: time-triggered synchronous reconfiguration of 1-Layer switches
- Nodes send data to multicast address: both DMs receive data
- No need for network reconfiguration when switching/changing DMs





- Redundancy of Data Master is considered, mostly for maintenance/upgrade purpose
- In such arrangement, two data masters work simultaneously, both send out Control Messages and receive data as well (interlocks, etc). However, only messages from the currently active Data Master (i.e. DM-A) are forwarded by the L-1 switches , the messages sent out by the backup Data Master (i.e. DM-B) are dropped by the L-1 switches.
- Such arrangement enables the DMs to work simultaneously as if the were both in control, during that time, proper (basically, identical) functioning of both of them can be verified before the switch-over is done (maintainable case)
- A simple time-triggered WR Switch re-configuration to be implemented to enable synchronous reconfiguration of both L-1 switches in order to provide seamless switch-over between redundant switches (a window in the sent Control Data might be required)
- Usage of multi-cast addresses for DMs is considered, so that data sent upstream always reaches both DMs and no network re-configuration is needed after exchanging DMs

ABV gen2 vs. WR

AVB gen2 vs. WR

ABV gen2 vs. WR

AVB gen2 vs. WR

Synchronization

- Differences
  - WR: physical syntonization is obligatory (using SyncE)
  - AVB: logical syntonization is optional
  - WR: request-response mechanism (no technical problem to align with AVB)
  - AVB: peer-delay mechanism
  - WR: phase alignment
- Similarities
  - Support of network redundancy and fast re-configuration
  - (Semi-)automatic link asymmetry calibration
- WR is inter-operable with (virtually) default PTP profile

Synchronization
- Differences
  - WR: physical syntonization is obligatory (using SyncE)
  - AVB: logical syntonization is optional
  - WR: request-response mechanism (no technical problem to align with AVB)
  - AVB: peer-delay mechanism
  - WR: phase alignment
- Similarities
  - Support of network redundancy and fast re-configuration
  - (Semi-)automatic link asymmetry calibration
- WR is inter-operable with (virtually) default PTP profile

Streams vs. Control Data

- Control Data sent by Data Master can be viewed as a AVB Stream sent by a talker
- WR statically reserves (single) resource for all (altogether) Control Data (streams)
- Static stream (Control Data) reservation in WR
- Stream in WR is broadcast within a VLAN - stream in WR defined by VLAN
- WR Control Data distribution – a corner case of AVB Stream

Streams vs. Control Data

- Control Data sent by Data Master can be viewed as a AVB Stream sent by a talker
- WR statically reserves (single) resource for all (altogether) Control Data (streams)
- Static stream (Control Data) reservation in WR
- Stream in WR is broadcast within a VLAN - stream in WR defined by VLAN
- WR Control Data distribution – a corner case of AVB Stream

Reliability (network/data redundancy)

- WR requires seamless redundancy – achieved by adding FEC to (very fast) dynamic redundancy (eRSTP) or multipath solution (eLACP)
- FEC introduces additional latency
- Advantage is taken from broadcast traffic characteristics
- In the current concept of WR-based control network - VLAN and mutlicast addresses are used - any redundancy solutions needs to support both

Reliability (network/data redundancy)

- WR requires seamless redundancy – achieved by adding FEC to (very fast) dynamic redundancy (eRSTP) or multipath solution (eLACP)
- FEC introduces additional latency
- Advantage is taken from broadcast traffic characteristics
- In the current concept of WR-based control network - VLAN and mutlicast addresses are used - any redundancy solutions needs to support both

Latency

- Strict priority output queue scheduling for Control Data is considered
- Time Aware Shaper (both at end stations and WR Switches) is worth considering for WR
- Preemption was considered in WR, it is currently stated an obsolete idea
- Cut-through forwarding in WR Switches

White Rabbit 34/39
AVB gen2 vs. WR
Latency
2012-05-11

Latency

- Strict priority output queue scheduling for Control Data is considered
- Time Aware Shaper (both at end stations and WR Switches) is worth considering for WR
- Preemption was considered in WR, it is currently stated an obsolete idea
- Cut-through forwarding in WR Switches

Fault tolerance / isolation

- VLANs are used to separate logically accelerator networks to limit propagation of fault due to mis-behaving node/switch
- FEC header has ID and sequence number - can provide help in duplication issues
- Control of throughput from nodes which are not supposed to send too much data

White Rabbit 35/39
AVB gen2 vs. WR
Fault tolerance / isolation
2012-05-11
Fault tolerance / isolation
VLANs are used to separate logically accelerator networks to limit propagation of fault due to mis-behaving node/switch
FEC header has ID and sequence number - can provide help in duplication issues
Control of throughput from nodes which are not supposed to send too much data

WR requirements in AVB terms (Case 1)

- max latency: $<$ **1000us** over $\approx$ **5 hops**[1] @ **1Gbps**
- guaranteed latency over **tree-like (meshed) topology**
- main network characteristics: **2000 end stations**, links of max 10km, max controller-to-node distance: **10km**
- traffic characteristics: "control data" size (payload): **500-5000 bytes** (in FEC scheme: encoded into 4 or 6 frames of size 300-1500 bytes and sent in burst), $\approx$ 8 "control streams" (defined within separate VLANs) sent every **1000us**, critical stream is one-to-many, "normal data" size (payload): 1500 bytes
- Support for VLAN and multicast
- Seamless redundancy or ultra fast dynamic reconfiguration for critical data ($<$ 2.3 us)
- Sub-nanosecond accuracy and picoseconds precision of synchronization

[1] hop=switch

WR requirements in AVB terms (Case 1)

- max latency: $<$ **1000us** over $\approx$ **5 hops**[1] @ **1Gbps**
- guaranteed latency over **tree-like (meshed) topology**
- main network characteristics: **2000 end stations**, links of max 10km, max controller-to-node distance: **10km**
- traffic characteristics: "control data" size (payload): **500-5000 bytes** (in FEC scheme: encoded into 4 or 6 frames of size 300-1500 bytes and sent in burst), $\approx$ 8 "control streams" (defined within separate VLANs) sent every **1000us**, critical stream is one-to-many, "normal data" size (payload): 1500 bytes
- Support for VLAN and multicast
- Seamless redundancy or ultra fast dynamic reconfiguration for critical data ($<$ 2.3 us)
- Sub-nanosecond accuracy and picoseconds precision of synchronization

[1] hop=switch

WR requirements in AVB terms (Case 2)

- max latency: **200us** over $\approx$ **5 hops**[2] @ **1Gbps**
- guaranteed latency over **tree-like (meshed) topology**
- main network characteristics: **2000-4000 end stations**, links of max 2km, max controller-to-node distance: **2km**
- traffic characteristics: ”control data” size (payload): $<$ **1500 bytes** (in FEC scheme: encoded into 4 frames of size 300-1000 bytes and sent in burst), sent every **200us**, critical stream is one-to-many, ”normal data” size (payload): 1500 bytes
- Support for VLAN and multicast
- Seamless redundancy or ultra fast dynamic reconfiguration for critical data ($<$ 2.3 us)
- Sub-nanosecond accuracy and picoseconds precision of synchronization

[2]hop=switch

WR requirements in AVB terms (Case 2)
- max latency: 200us over ≈ 5 hops[2] @ 1Gbps
- guaranteed latency over tree-like (meshed) topology
- main network characteristics: 2000-4000 end stations, links of max 2km, max controller-to-node distance: 2km
- traffic characteristics: ”control data” size (payload): < 1500 bytes (in FEC scheme: encoded into 4 frames of size 300-1000 bytes and sent in burst), sent every 200us, critical stream is one-to-many, ”normal data” size (payload): 1500 bytes
- Support for VLAN and multicast
- Seamless redundancy or ultra fast dynamic reconfiguration for critical data (< 2.3 us)
- Sub-nanosecond accuracy and picoseconds precision of synchronization
[2]hop=switch

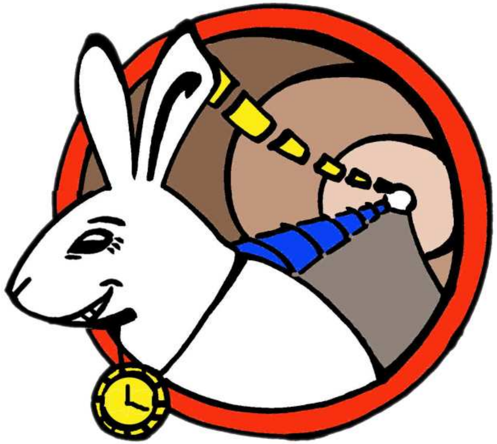
Conclusions

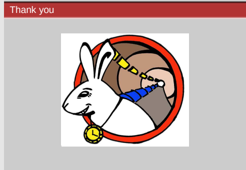
- WR is in many regards a corner case of AVB Gen2 (optimized for broadcast and static streams)
- WR adds to AVB Gen2 such requirements as:
  - Physical (SyncE-based) syntonization
  - VLAN-wide streams and broadcast
  - Static stream/resource configuration
- WR could benefit from AVB gen2 from:
  - Time Aware Shaper
  - Preemption
  - Multipath solution
- FEC is a separate and transparent layer – can be decoupled

Conclusions

- WR is in many regards a corner case of AVB Gen2 (optimized for broadcast and static streams)
- WR adds to AVB Gen2 such requirements as:
  - Physical (SyncE-based) syntonization
  - VLAN-wide streams and broadcast
  - Static stream/resource configuration
- WR could benefit from AVB gen2 from:
  - Time Aware Shaper
  - Preemption
  - Multipath solution
- FEC is a separate and transparent layer – can be decoupled

Thank you





1. White Rabbit Project homepage
   http://www.ohwr.org/projects/white-rabbit
2. White Rabbit Specification
   http://www.ohwr.org/documents/21
3. White Rabbit Standardization wiki
   ttp://www.ohwr.org/projects/wr-std/wiki/WRinAVBgen2
4. White Rabbit CERN Control and Timing Network
   http://www.ohwr.org/documents/85
5. White Rabbit – a PTP application for sub-ns synchronization
   http://www.ohwr.org/documents/155
6. White Rabbit and Robustness
   http://www.ohwr.org/documents/103
7. Topology Resolution, Redundant Links Handling and Fast Convergence in White Rabbit Network
   http://www.ohwr.org/documents/103
8. THE WHITE RABBIT PROJECT (ICALEPCS2009)
   http://www.ohwr.org/attachments/312/