The White Rabbit project

an Ethernet-based solution for sub-ns synchronization and deterministic delivery

Greg Daniluk

CERN BE-CO Hardware and Timing section

2 June 2014

Outline

Outline

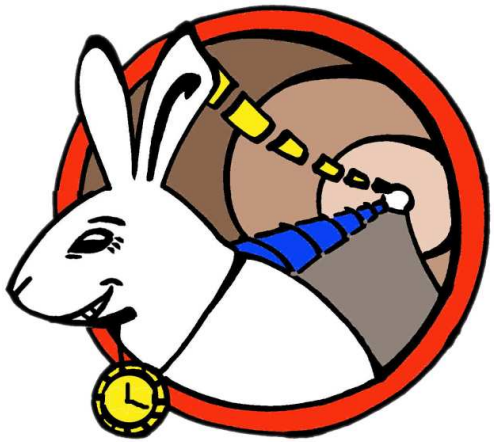
What's in a name ?

Oh dear! Oh dear! I shall be too late!

The White Rabbit in charge of real time

What is White Rabbit?

- Renovation of accelerator's control and timing
- Based on well-known technologies
- Open Hardware and Open Software with commercial support
- International collaboration
- Many users: CERN, GSI, KM3NET, cosmic ray detectors, metrology labs...



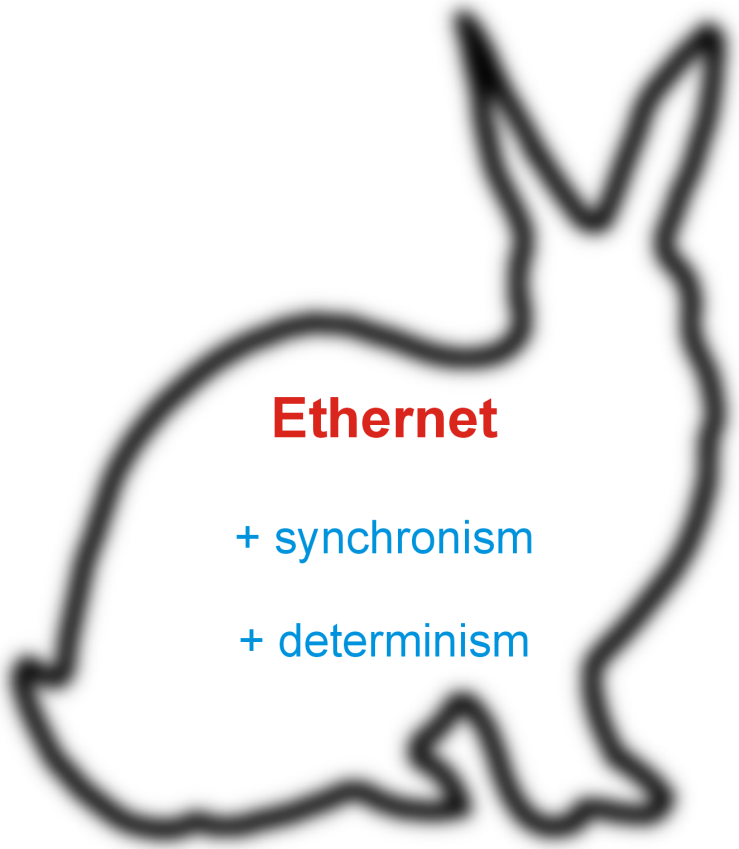
Why we use Open Hardware ?

	Commercial	Non-commercial
Open	**Winning combination. Best of both worlds.**	Whole support burden falls on developers. Not scalable.
Proprietary	Vendor lock-in.	Dedicated non-reusable projects.

- Get a design just the way we want it
- Peer review
- Healthier relationship with companies

White Rabbit features

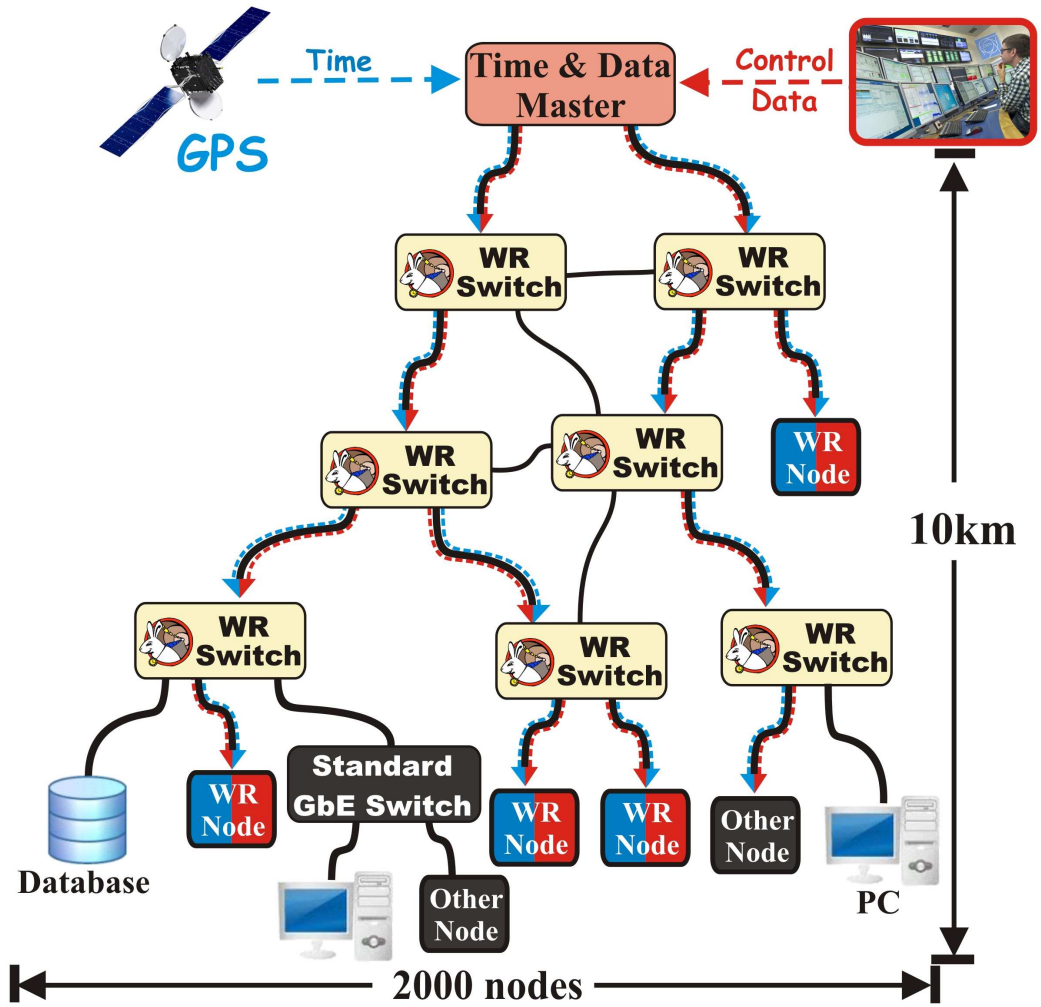
- Ethernet-based
  - thousands-nodes system
  - tens-km span
- Synchronism
  - sub-ns accuracy
  - tens-ps precision
- Determinism
  - upper-bound low-latency
  - high reliability



Outline

White Rabbit Network

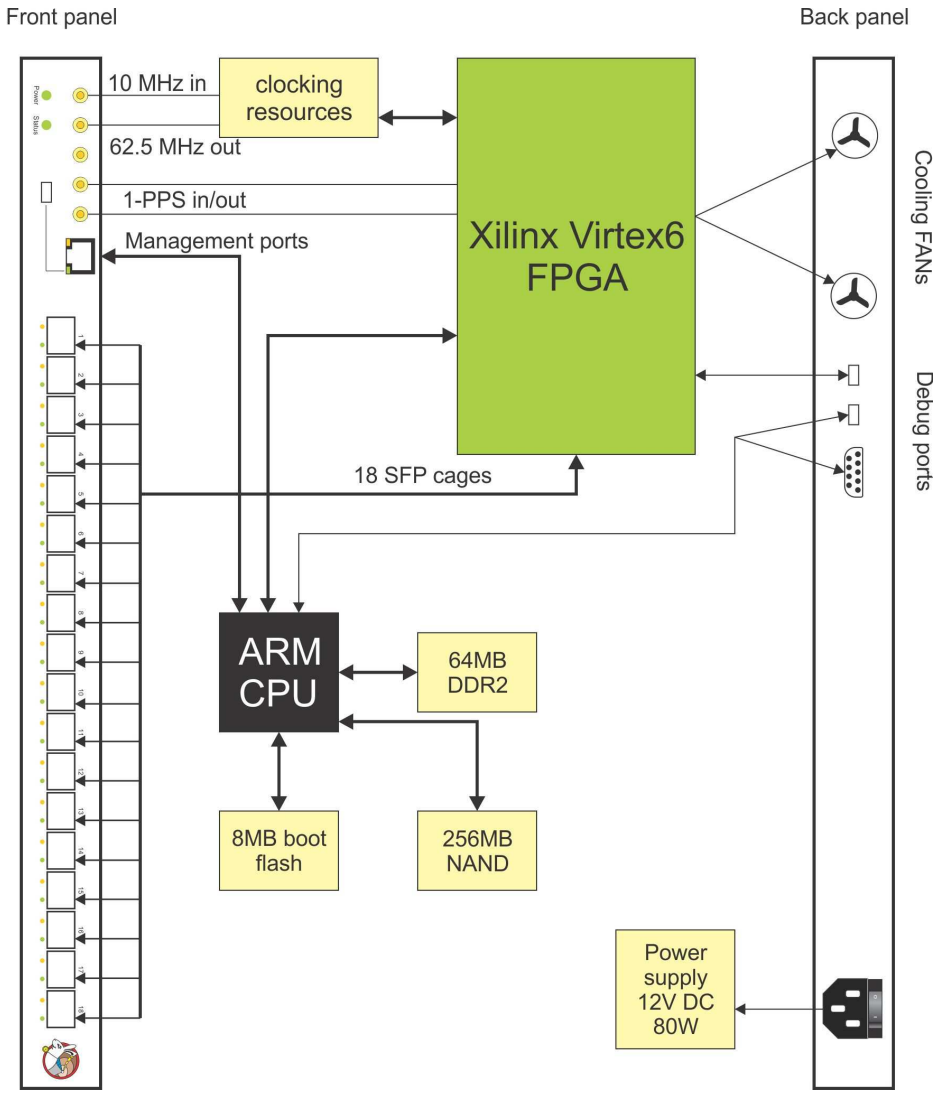
- Standard Ethernet network
- Ethernet features (VLAN) & protocols (SNMP)
- High accuracy synchronization
- Reliable and low-latency Control Data



White Rabbit Switch

- Central element of WR network
- Designed from scratch
- 18 ports
- 1000BASE-BX10 SFPs: up to 10 km, single-mode fiber
- Open design (H/W and S/W), commercially available

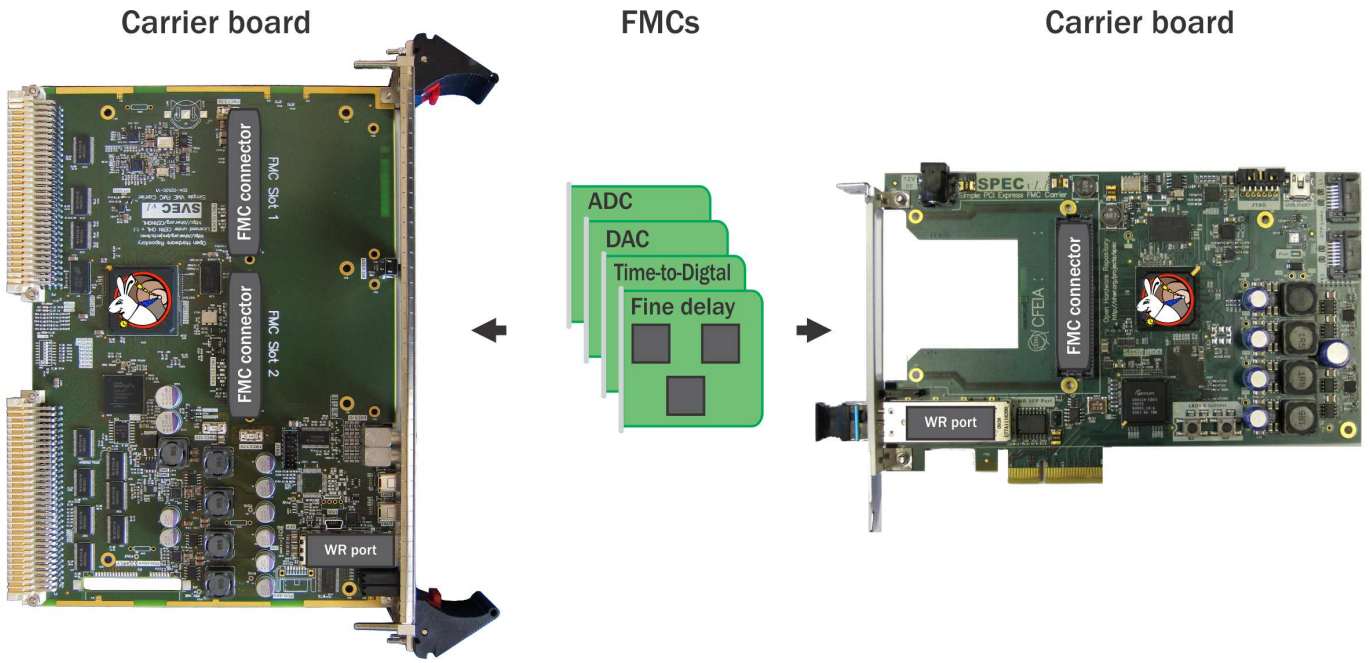
Simplified block diagram of WR switch



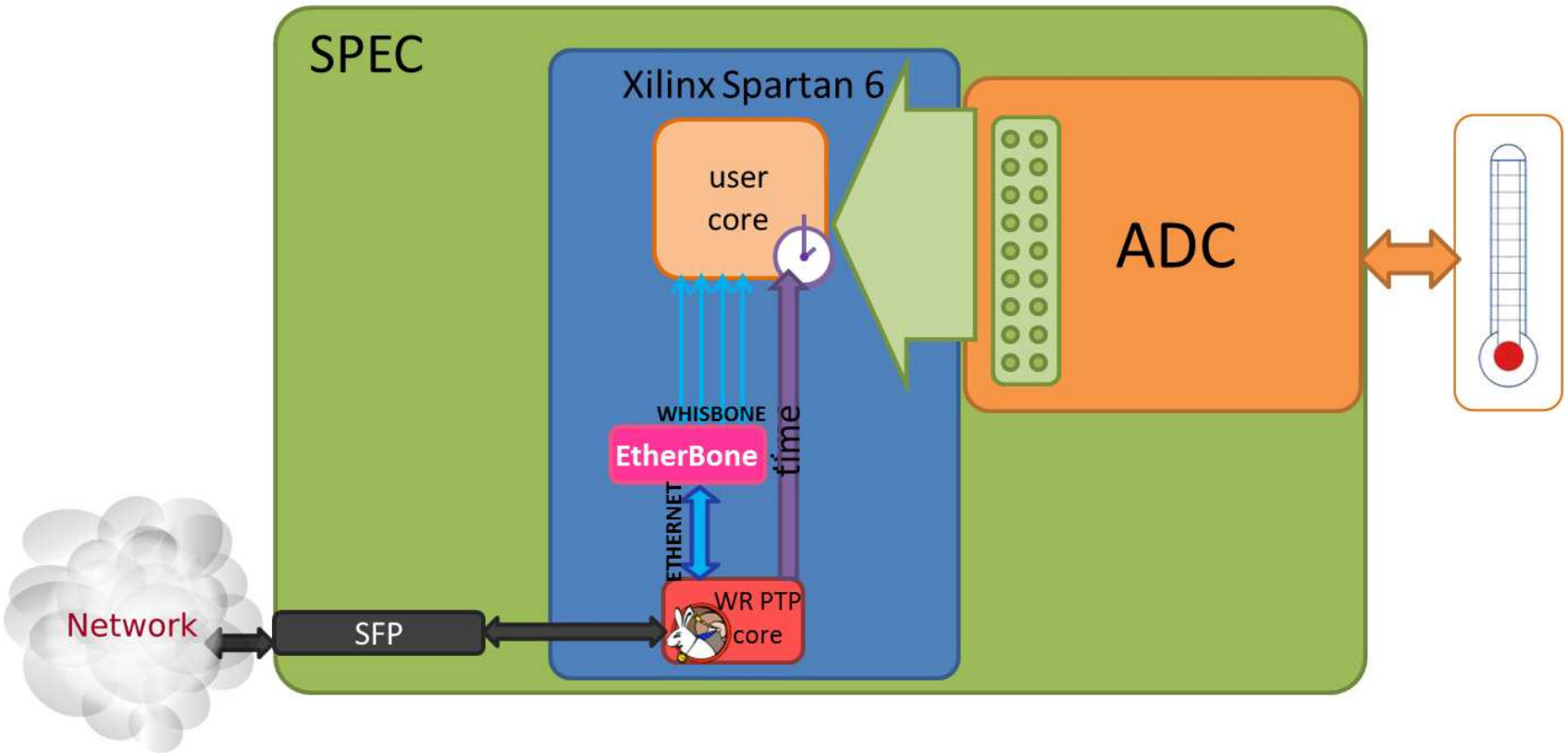
White Rabbit Node

Modular hardware kit:

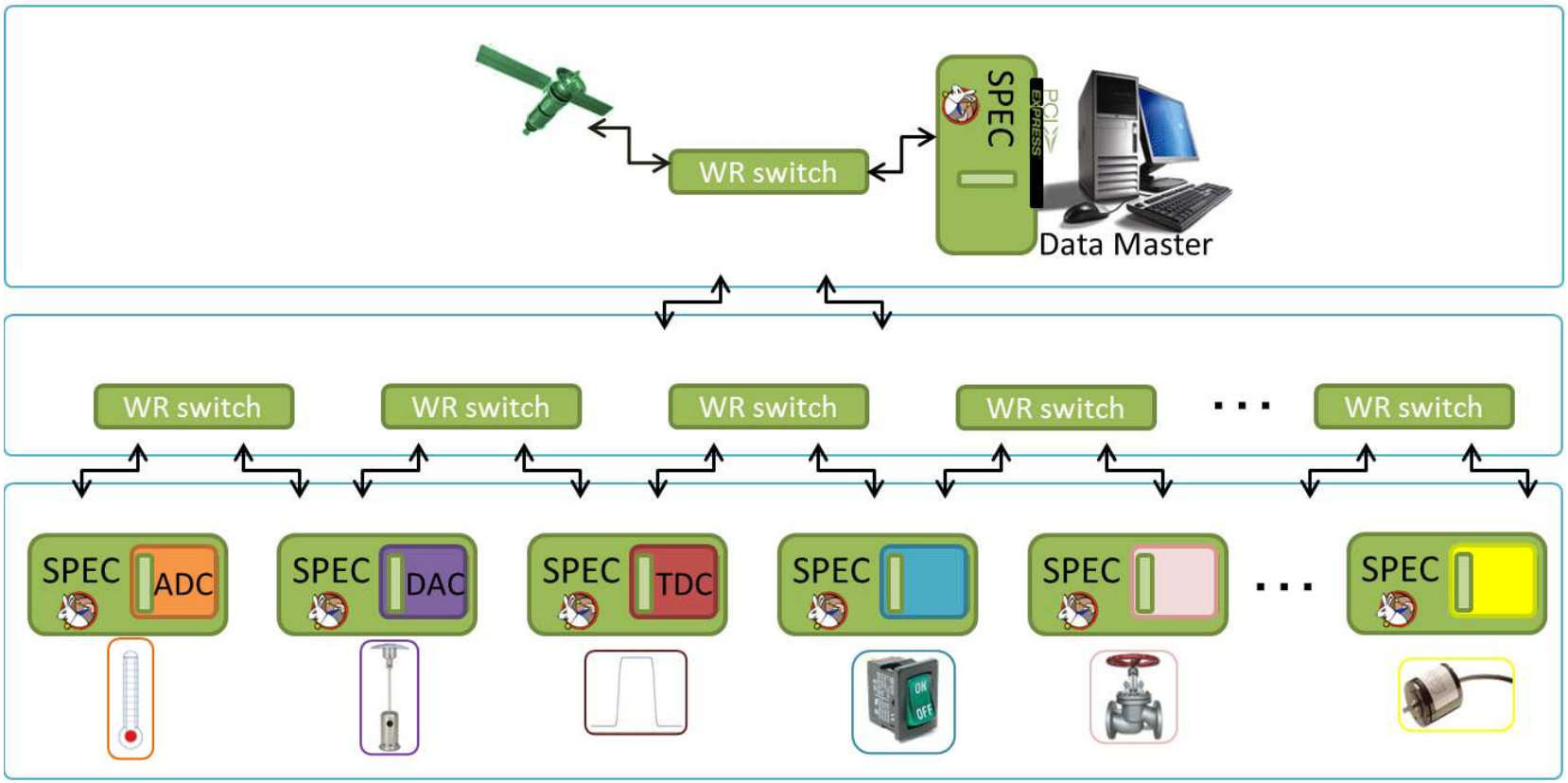
- set of Mezzanine boards: ADC, DAC, TDC, Fine delay...
- set of carriers for various needs: PCIe, VME64x, PXIe...
- all carriers equipped with a White Rabbit port



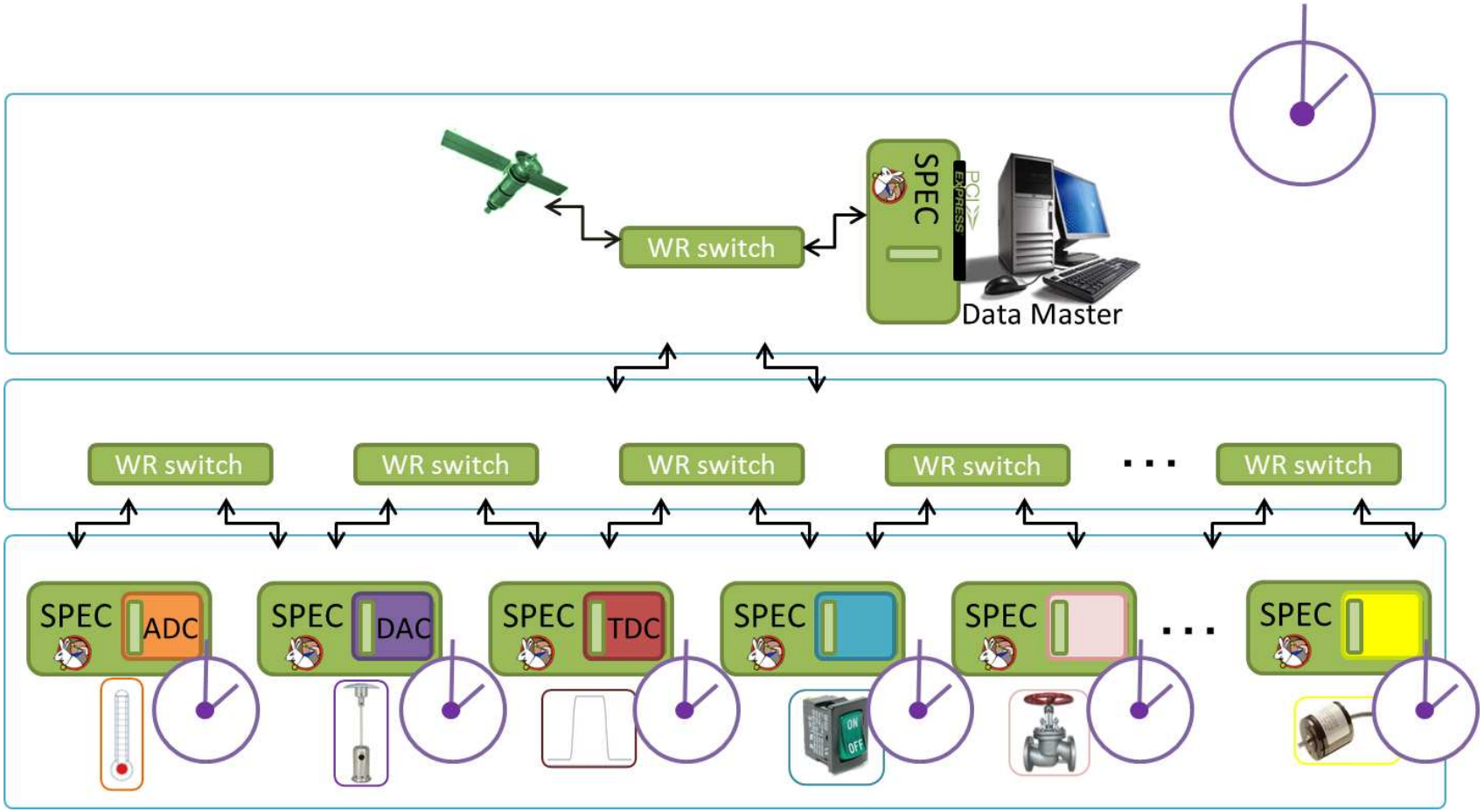
White Rabbit Node - example



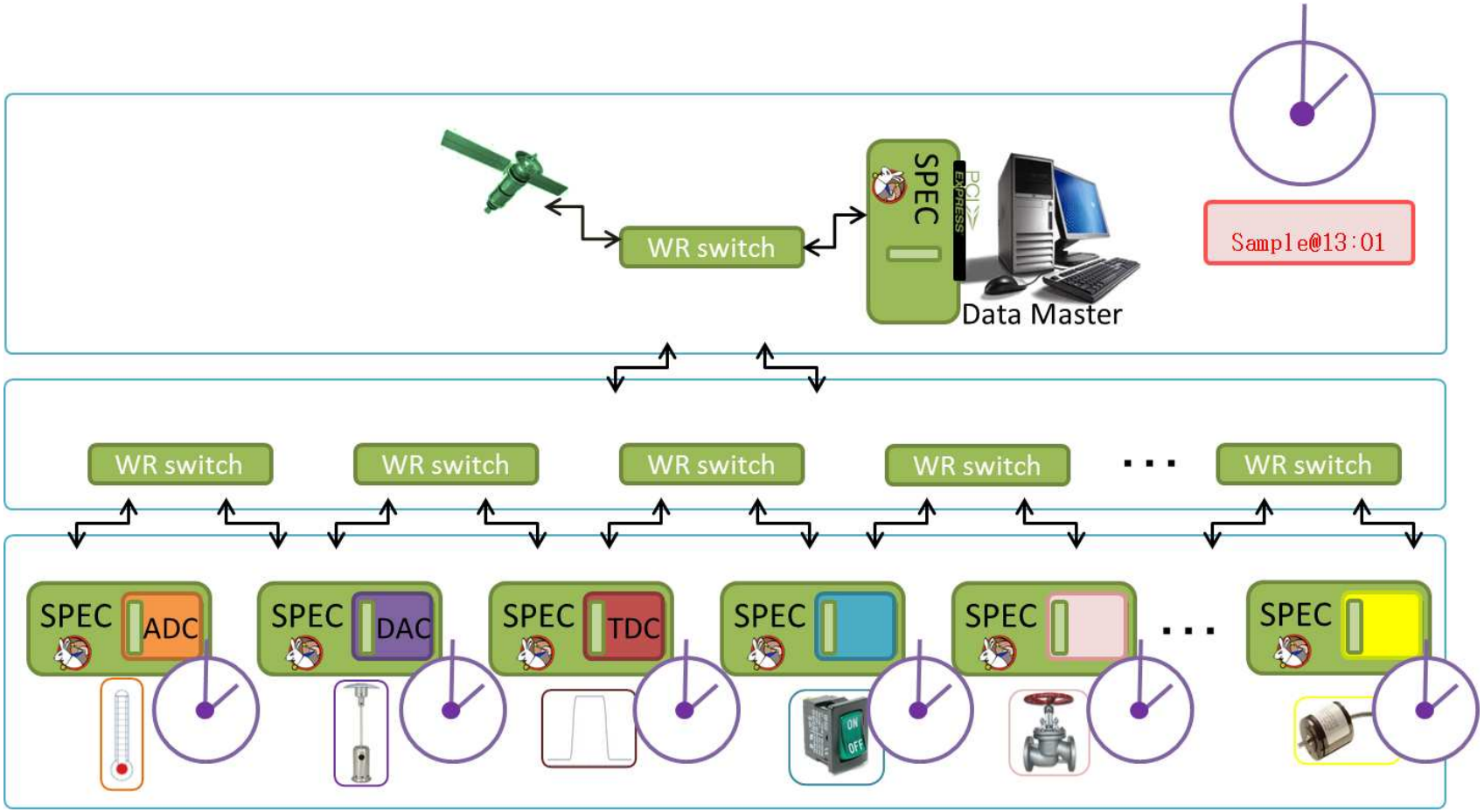
White Rabbit Node - example



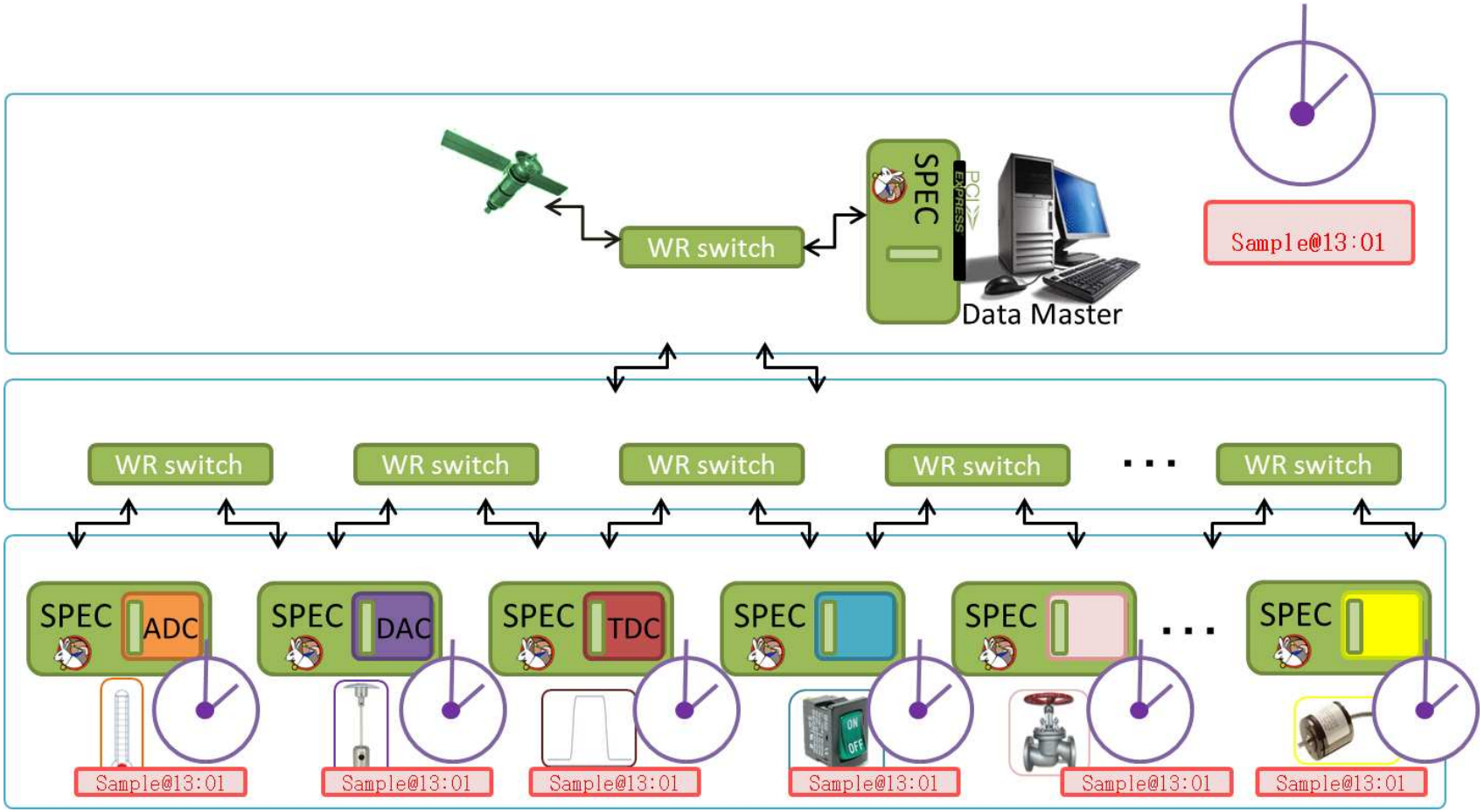
White Rabbit Node - example



White Rabbit Node - example

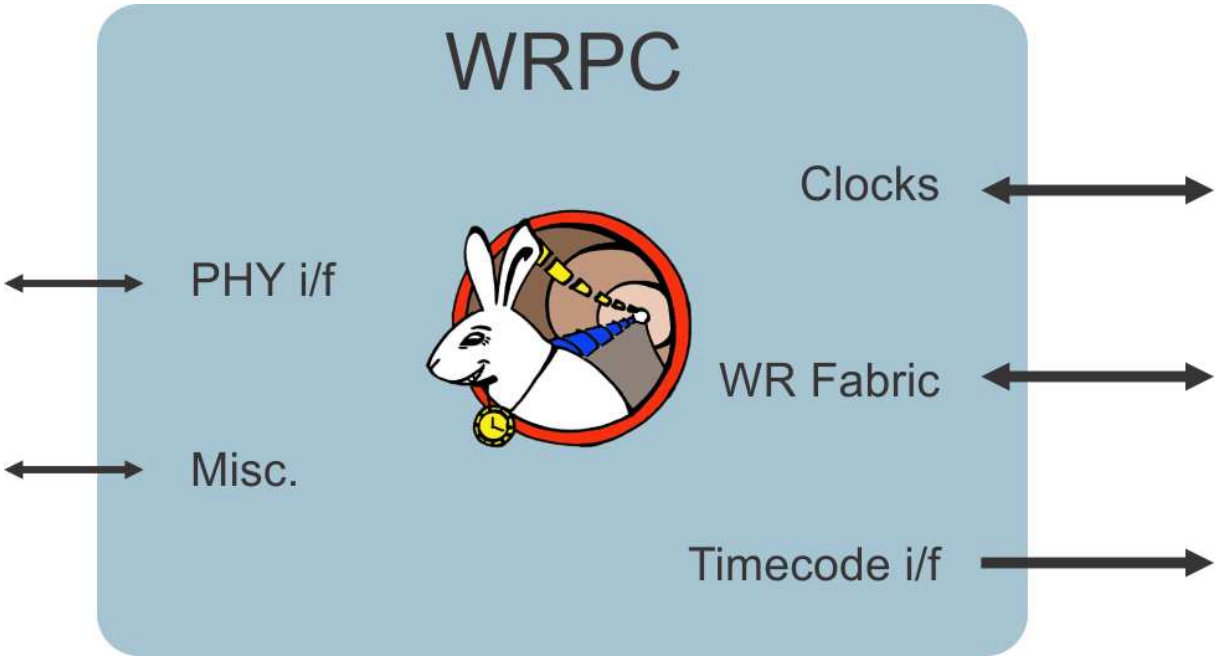


White Rabbit Node - example



White Rabbit PTP Core

- Fancy Ethernet MAC with White Rabbit support
- Open IP Core
- Easily integrated into custom FPGA-based designs



Open Hardware Repository (OHWR)

- All schematics, HDL designs and software sources available in OHWR
- Over 100 projects currently hosted
- 11 scientific institutes and 16 companies involved

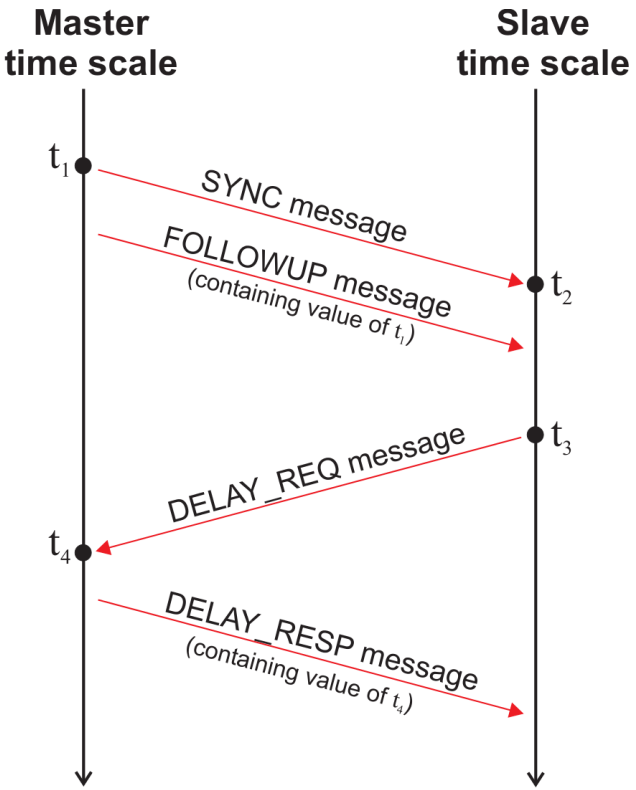
http://www.ohwr.org

Outline

Time Distribution in White Rabbit Network

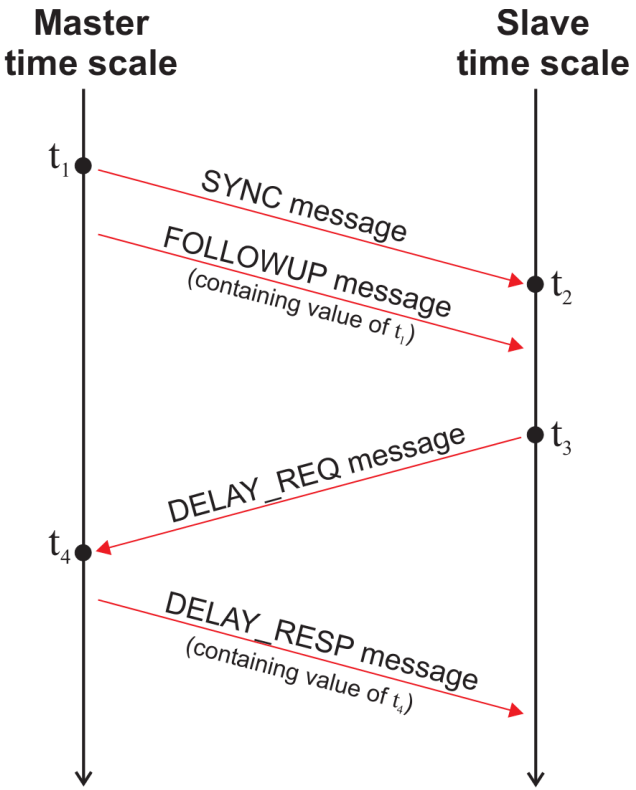
- Synchronization with **sub-ns** accuracy **tens-ps** precision
- Combination of
  - Precision Time Protocol (**IEEE1588**) synchronization
  - Layer 1 syntonization
  - Phase measurements

Precision Time Protocol (IEEE1588)



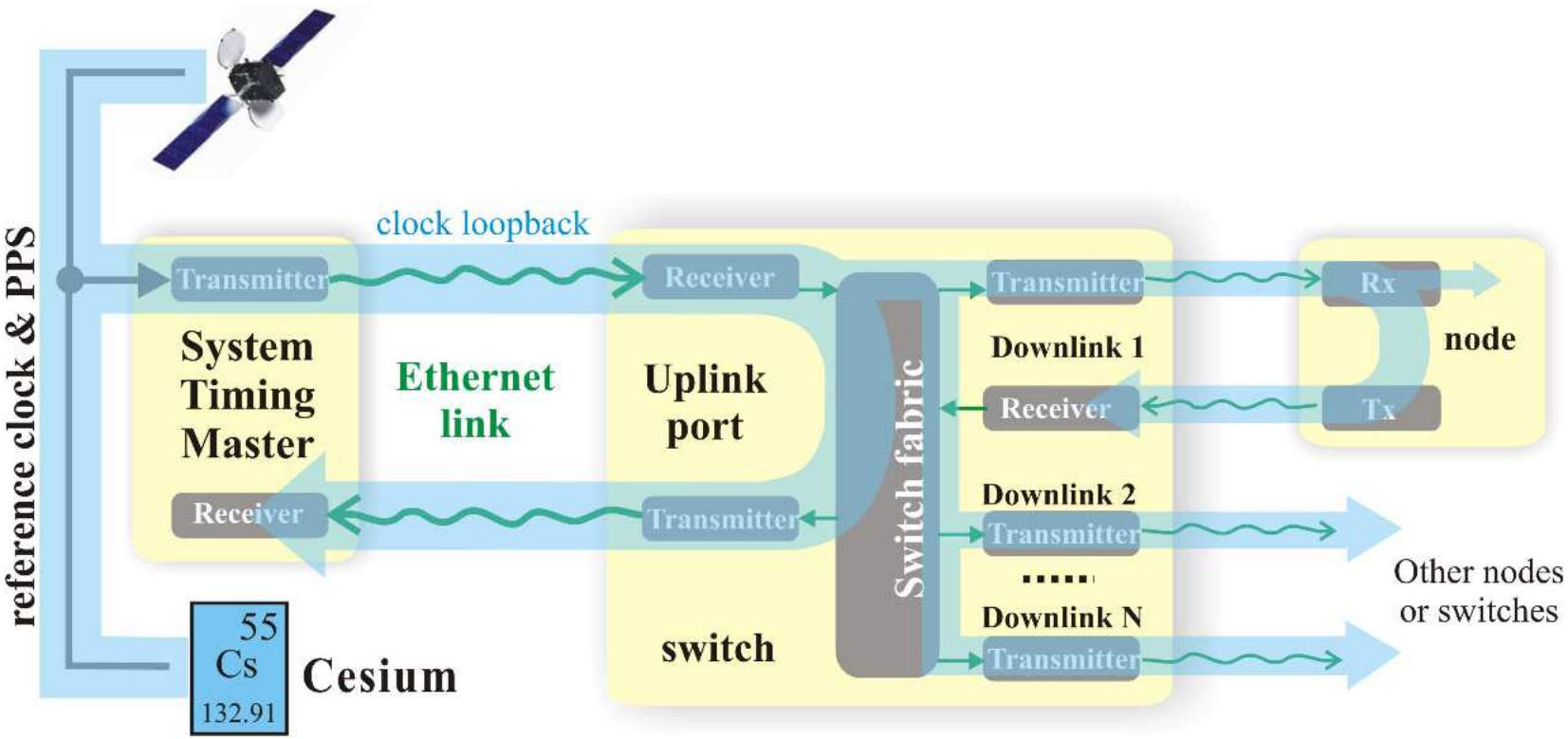
- Simple calculations:
  - link $delay_{ms}$: $\delta_{ms} = \frac{(t_4 - t_1) - (t_3 - t_2)}{2}$
  - clock $offset_{ms} = t_2 - t_1 + \delta_{ms}$

Precision Time Protocol (IEEE1588)



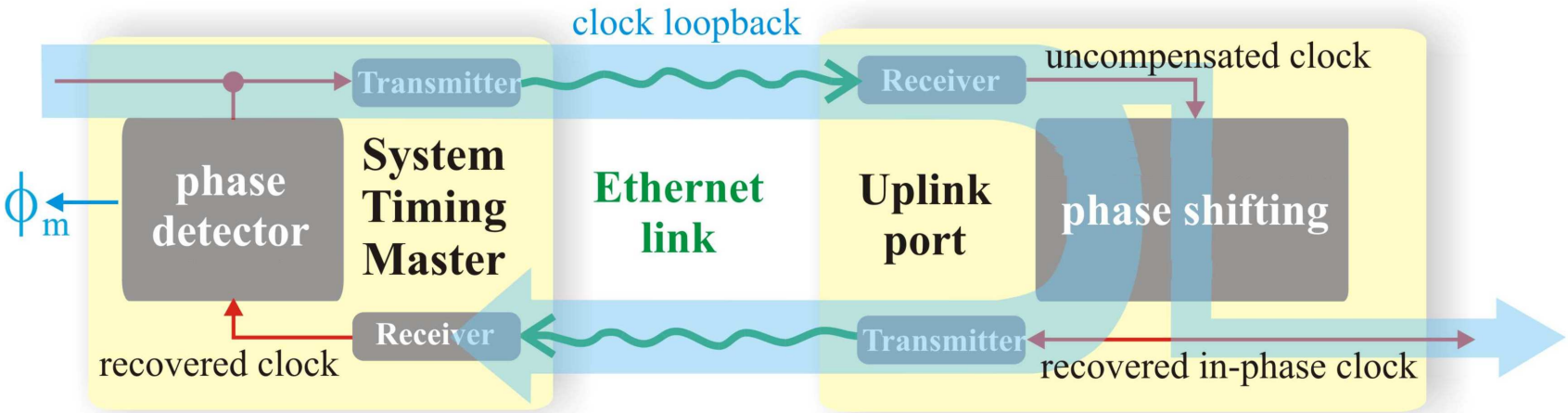
- Simple calculations:
  - link $delay_{ms}$: $\delta_{ms} = \frac{(t_4 - t_1) - (t_3 - t_2)}{2}$
  - clock $offset_{ms} = t_2 - t_1 + \delta_{ms}$
- Disadvantages
  - assumes symmetry of medium
  - all nodes have free-running oscillators
  - frequency drift compensation vs. message exchange traffic

Layer 1 Syntonization

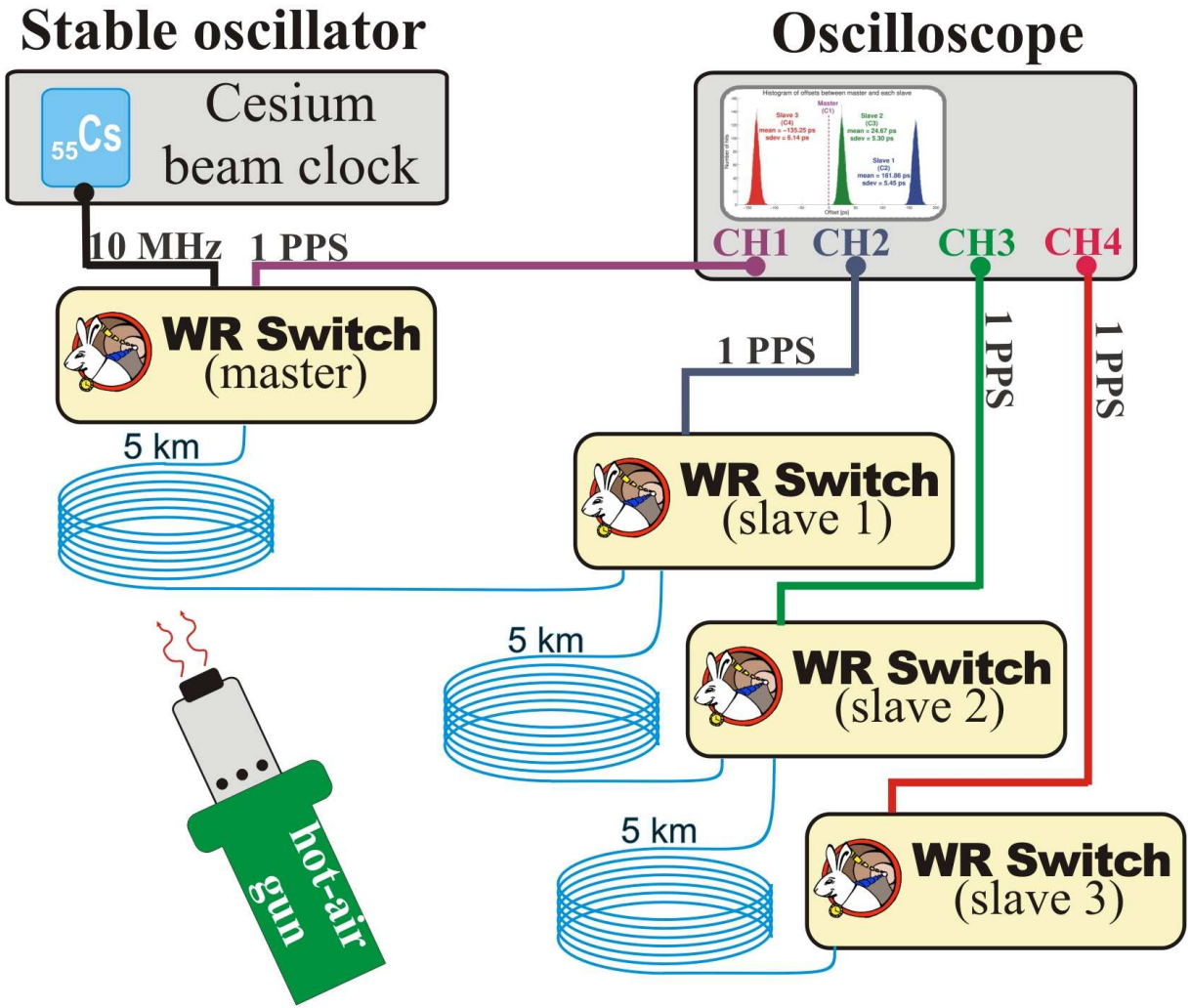


Phase measurements

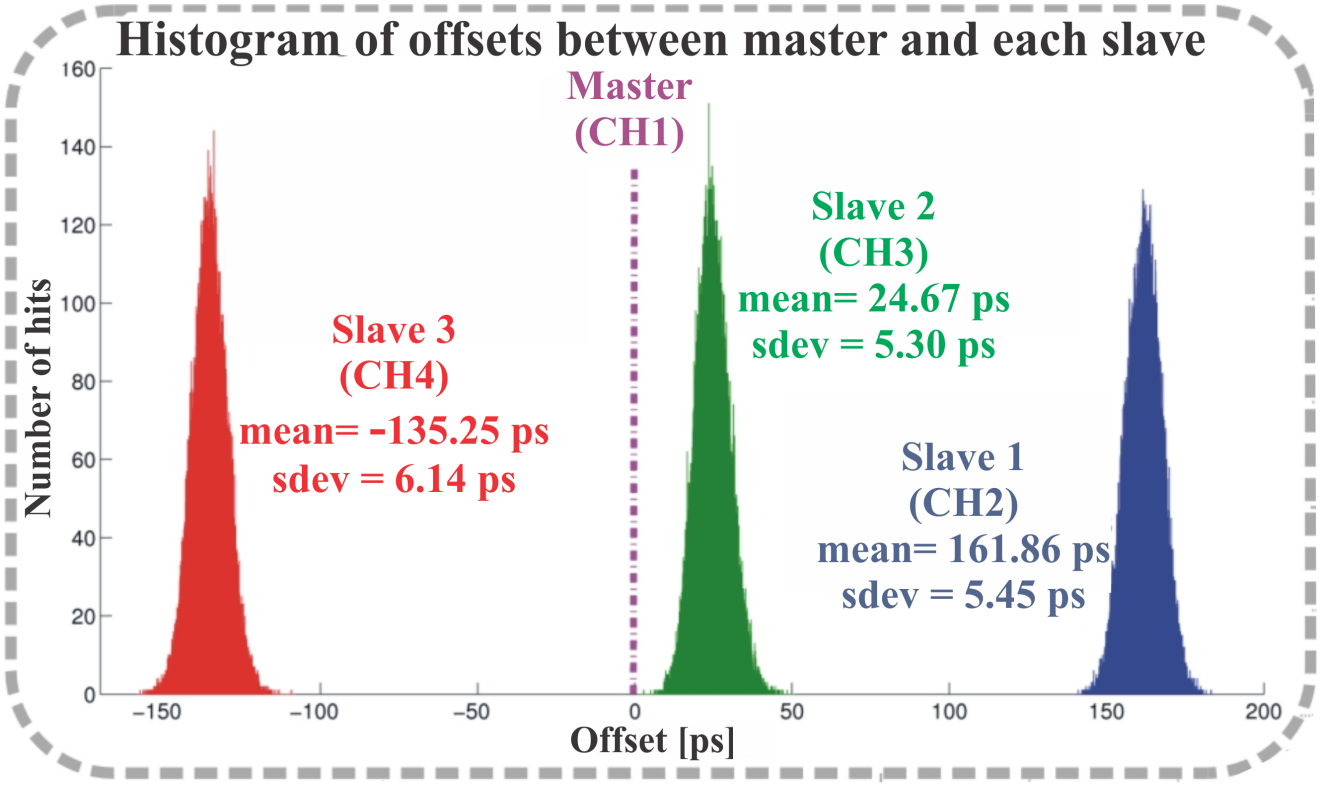
- Monitor phase of bounced-back clock
- Enhance PTP timestamps with phase measurement
- Phase-locked loop in the slave follows the phase changes



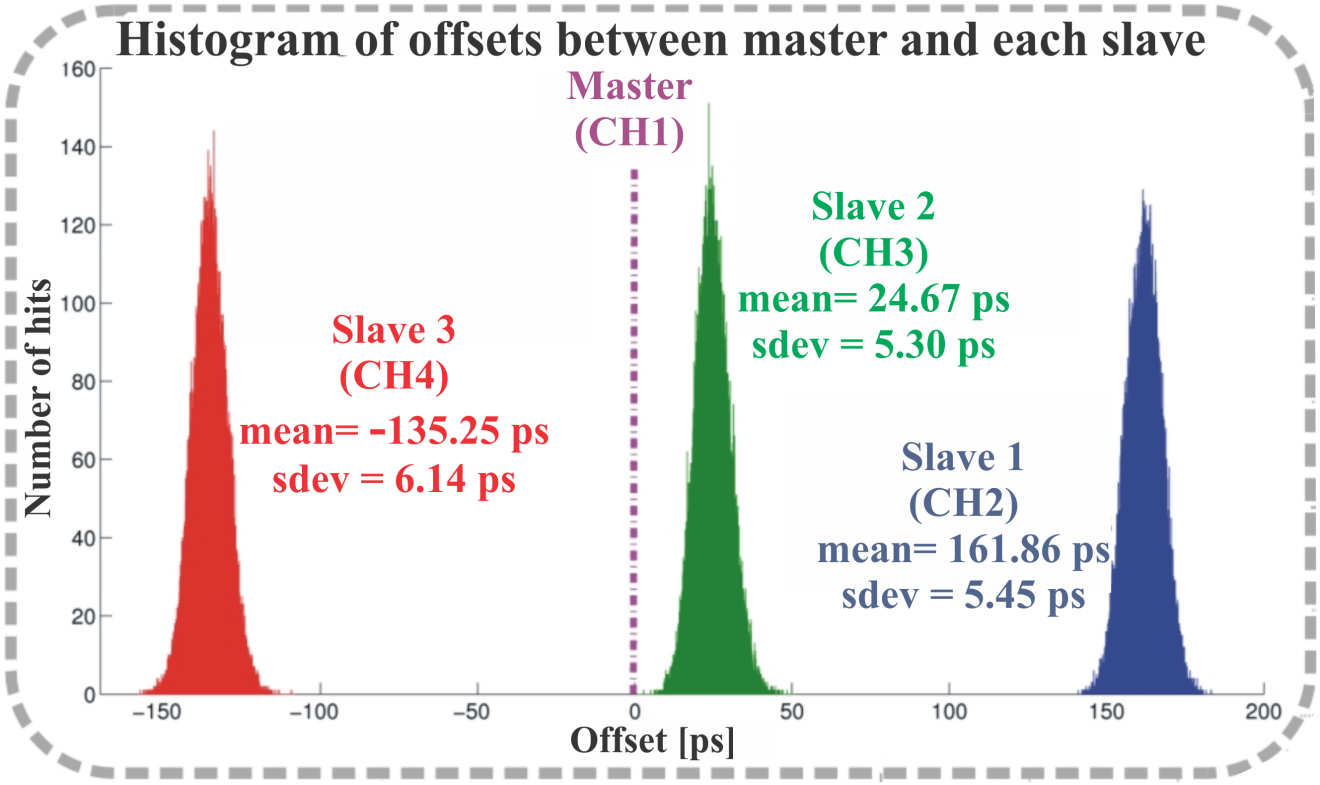
WR synchronization performance



WR synchronization performance



WR synchronization performance



ISPCS Plug Fest

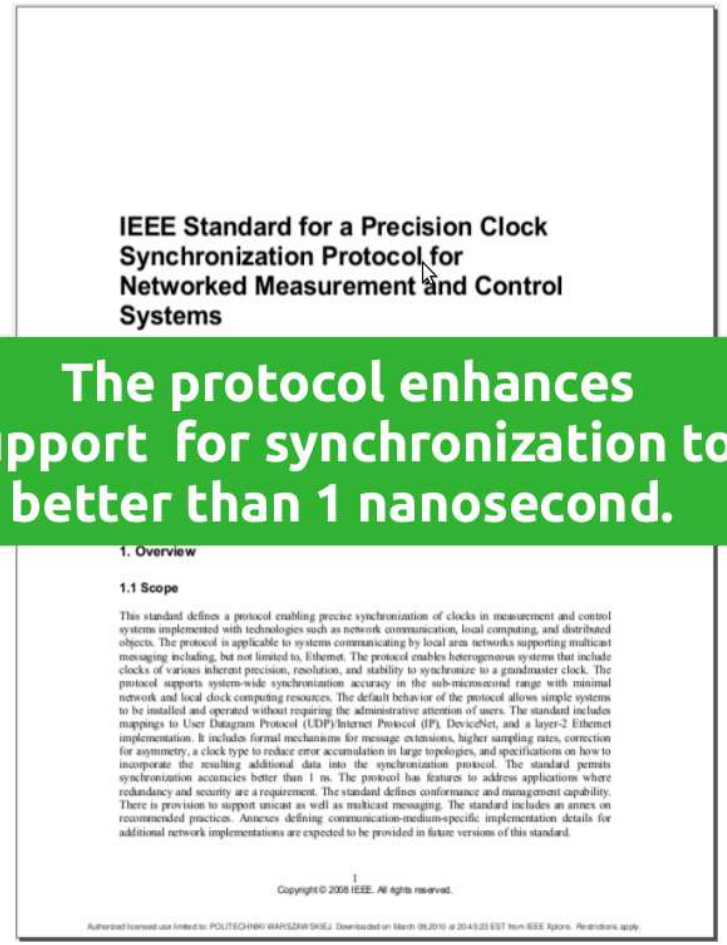
WR: most accurate PTP implementation in the world!

WR Standardization under IEEE1588

- We want to standardize!

WR Standardization under IEEE1588

- We want to standardize!
- Intention by 1588 Standardization Group expressed in Project Authorization Request



IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems

The protocol enhances support for synchronization to better than 1 nanosecond.

1. Overview

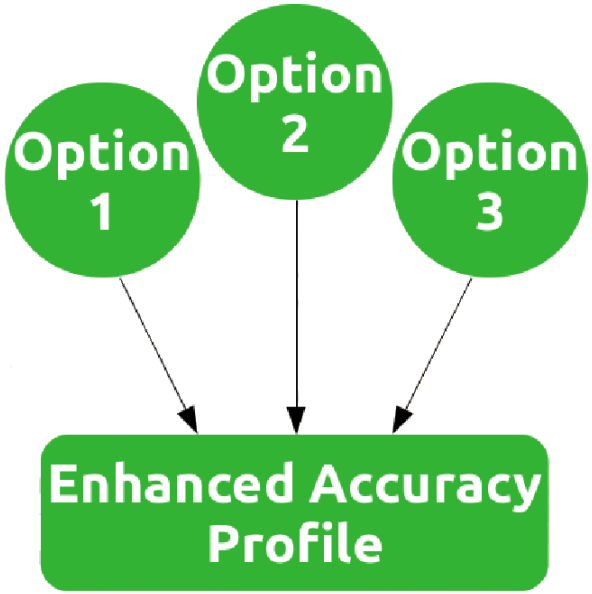
1.1 Scope

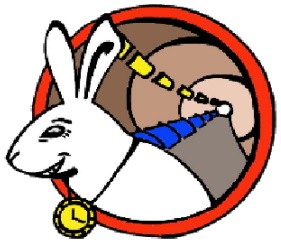
This standard defines a protocol enabling precise synchronization of clocks in measurement and control systems implemented with technologies such as network communication, local computing, and distributed objects. The protocol is applicable to systems communicating by local area networks supporting multicast messaging including, but not limited to, Ethernet. The protocol enables heterogeneous systems that include clocks of various inherent precision, resolution, and stability to synchronize to a grandmaster clock. The protocol supports system-wide synchronization accuracy in the sub-microsecond range with minimal network and local clock computing resources. The default behavior of the protocol allows simple systems to be installed and operated without requiring the administrative attention of users. The standard includes mappings to User Datagram Protocol (UDP)/Internet Protocol (IP), DeviceNet, and a layer-2 Ethernet implementation. It includes formal mechanisms for message extensions, higher sampling rates, correction for asymmetry, a clock type to reduce error accumulation in large topologies, and specifications on how to incorporate the resulting additional data into the synchronization protocol. The standard permits synchronization accuracies better than 1 ns. The protocol has features to address applications where redundancy and security are a requirement. The standard defines conformance and management capability. There is provision to support unicast as well as multicast messaging. The standard includes an annex on recommended practices. Annexes defining communication-medium-specific implementation details for additional network implementations are expected to be provided in future versions of this standard.

1

WR Standardization under IEEE1588

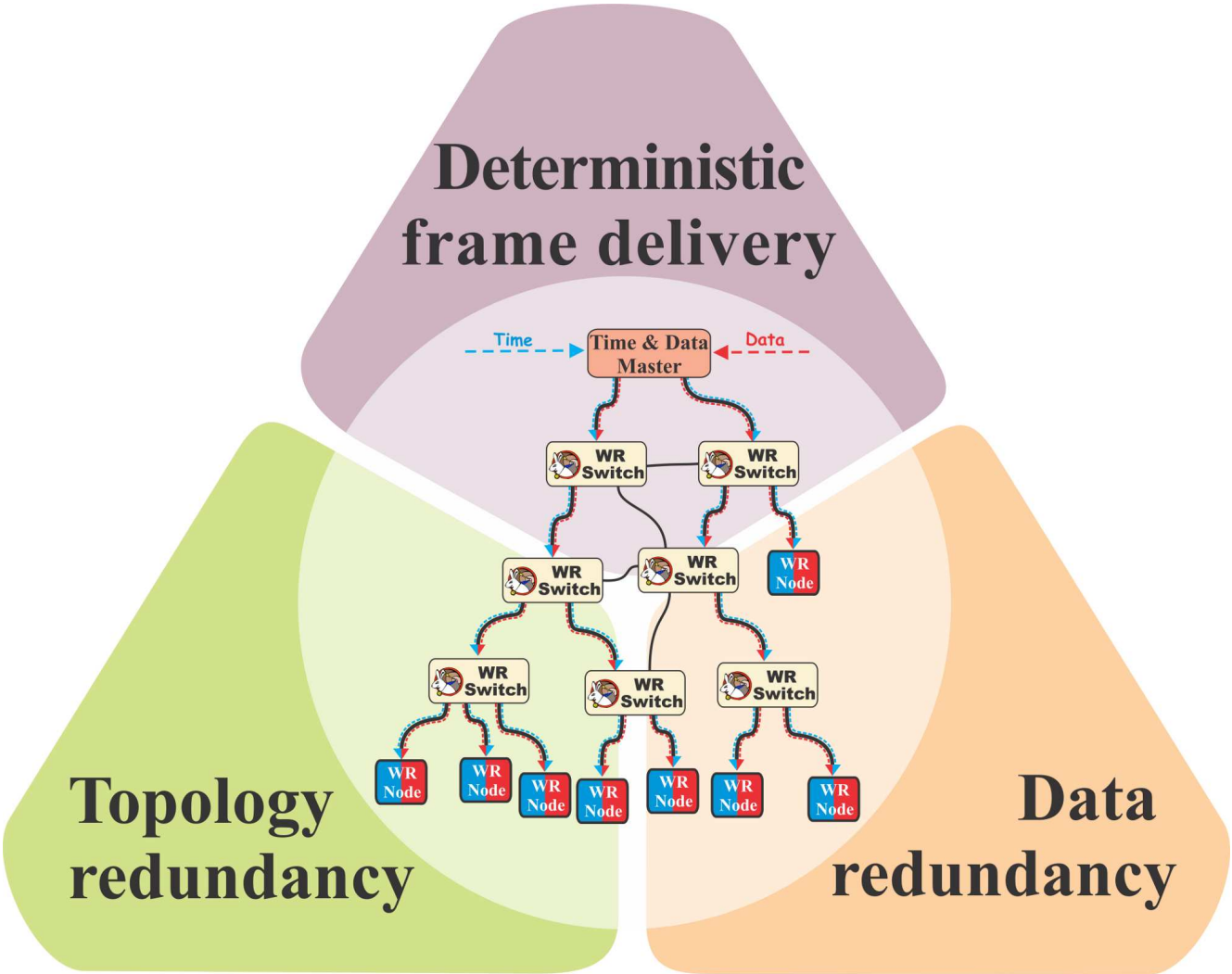
- We want to standardize!
- Intention by 1588 Standardization Group expressed in Project Authorization Request
- Enhanced Accuracy Options / Profile





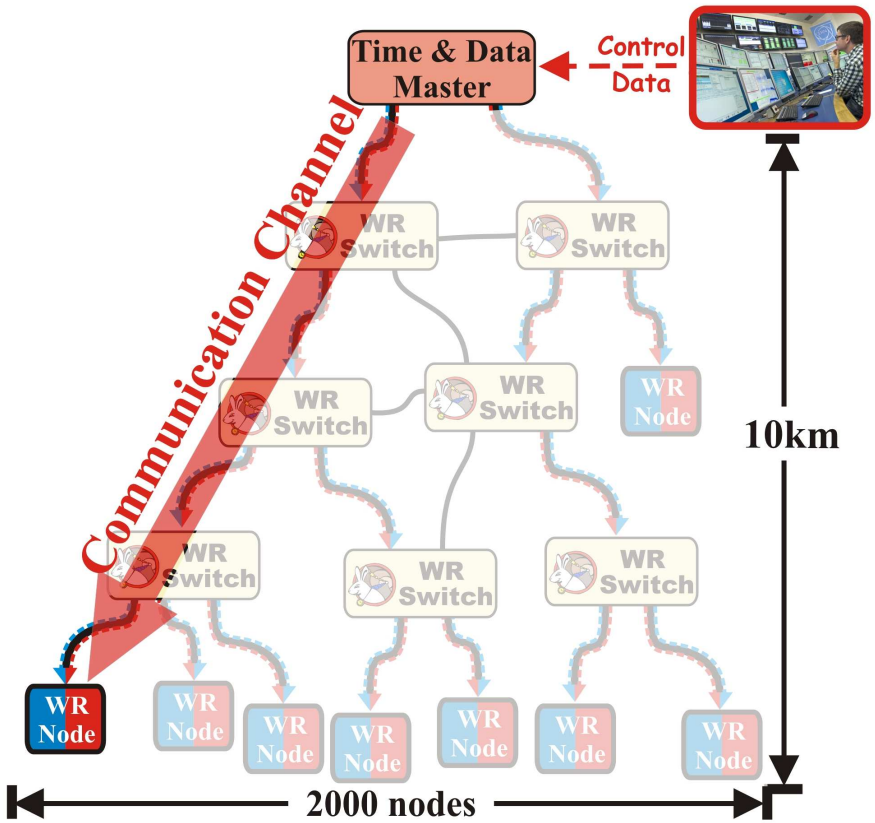
Outline

Data Distribution in a White Rabbit Network



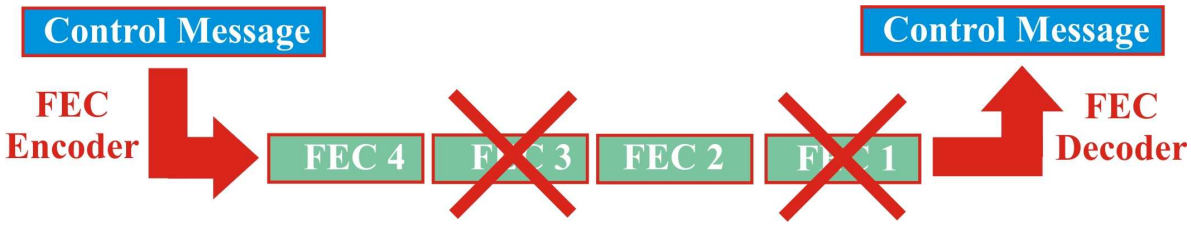
Deterministic data delivery

- Types of data distinguished by 802.1Q tag:
  - **Control Data** (strict priority)
  - Standard Data (Best Effort)
- **Control Data** characteristics:
  - Sent by Data Master(s)
  - Broadcast (one-to-many)
  - Deterministic and low-latency
  - Reliable delivery
- Low-latency WR Switch by design ($< 10us$)



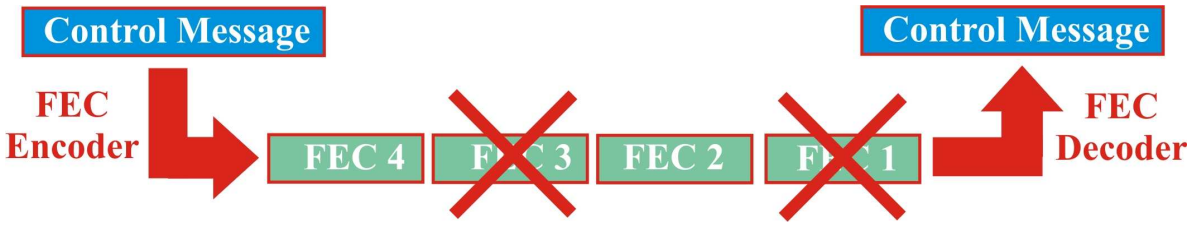
Data Redundancy (Node)

- **Forward Error Correction (FEC)** – transparent layer:
  - One message encoded into 4 Ethernet frames
  - Recovery of message from any 2 frames



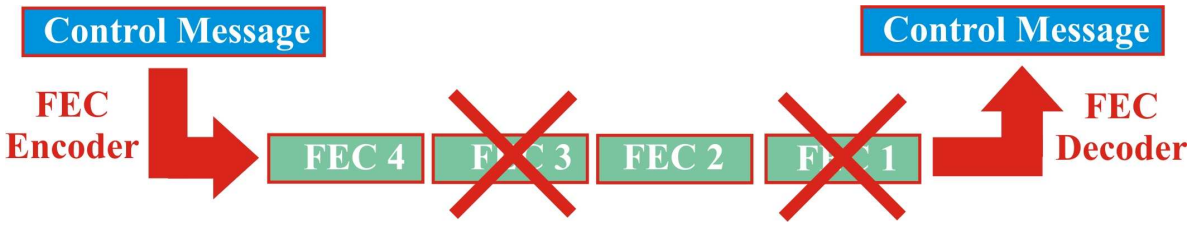
Data Redundancy (Node)

- **Forward Error Correction (FEC)** – transparent layer:
  - One message encoded into 4 Ethernet frames
  - Recovery of message from any 2 frames
- FEC can prevent data loss due to:



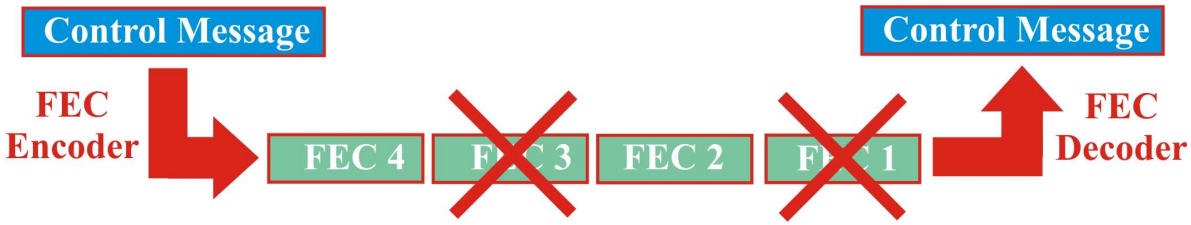
Data Redundancy (Node)

- **Forward Error Correction (FEC)** – transparent layer:
  - One message encoded into 4 Ethernet frames
  - Recovery of message from any 2 frames
- FEC can prevent data loss due to:
  - **bit errors**



Data Redundancy (Node)

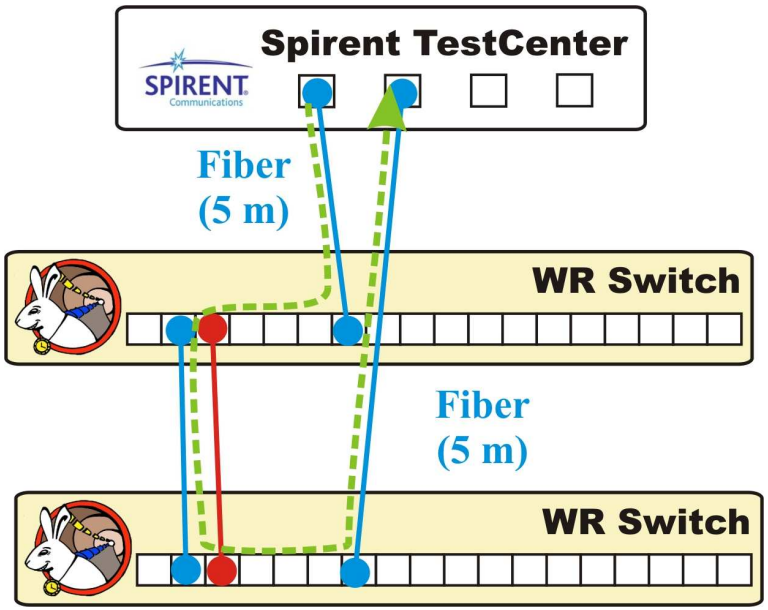
- **Forward Error Correction (FEC)** – transparent layer:
  - One message encoded into 4 Ethernet frames
  - Recovery of message from any 2 frames
- FEC can prevent data loss due to:
  - **bit errors**
  - **network reconfiguration**



Topology Redundancy (Switch)

- Ideas:
  - Using VLANs
  - H/W switch-over to the backup link
  - WR Rapid Spanning Tree Protocol
  - WR Shortest Path Bridging
- Seamless redundancy requires Forward Error Correction

Topology reconfiguration performance

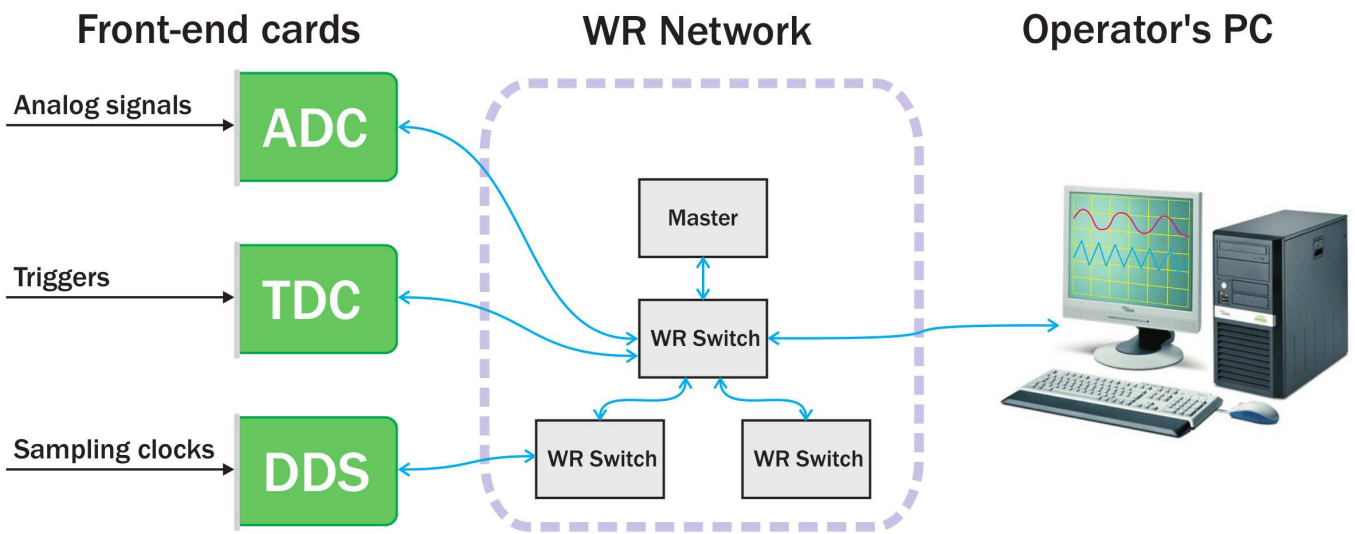


Frame Loss and Latencies

Frame Size (bytes)	Load (%)	Tx Frames	Rx Frames	Frame Loss	Max Latency (uSec)
288	10	1,217,533	1,217,533	0	5.84
288	30	3,652,598	3,652,597	1	5.84
288	50	6,087,663	6,087,663	0	5.84
288	70	8,522,728	8,522,727	1	5.84
288	90	10,957,793	10,957,792	1	6.12

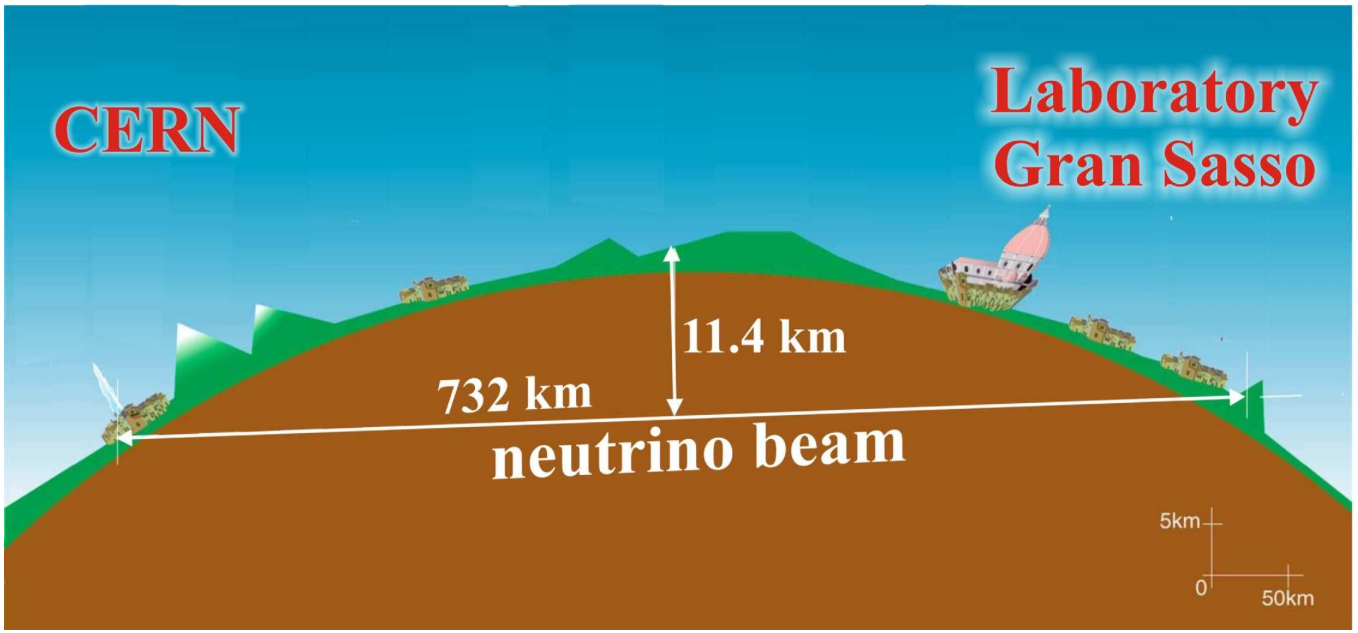
Outline

Distributed oscilloscope



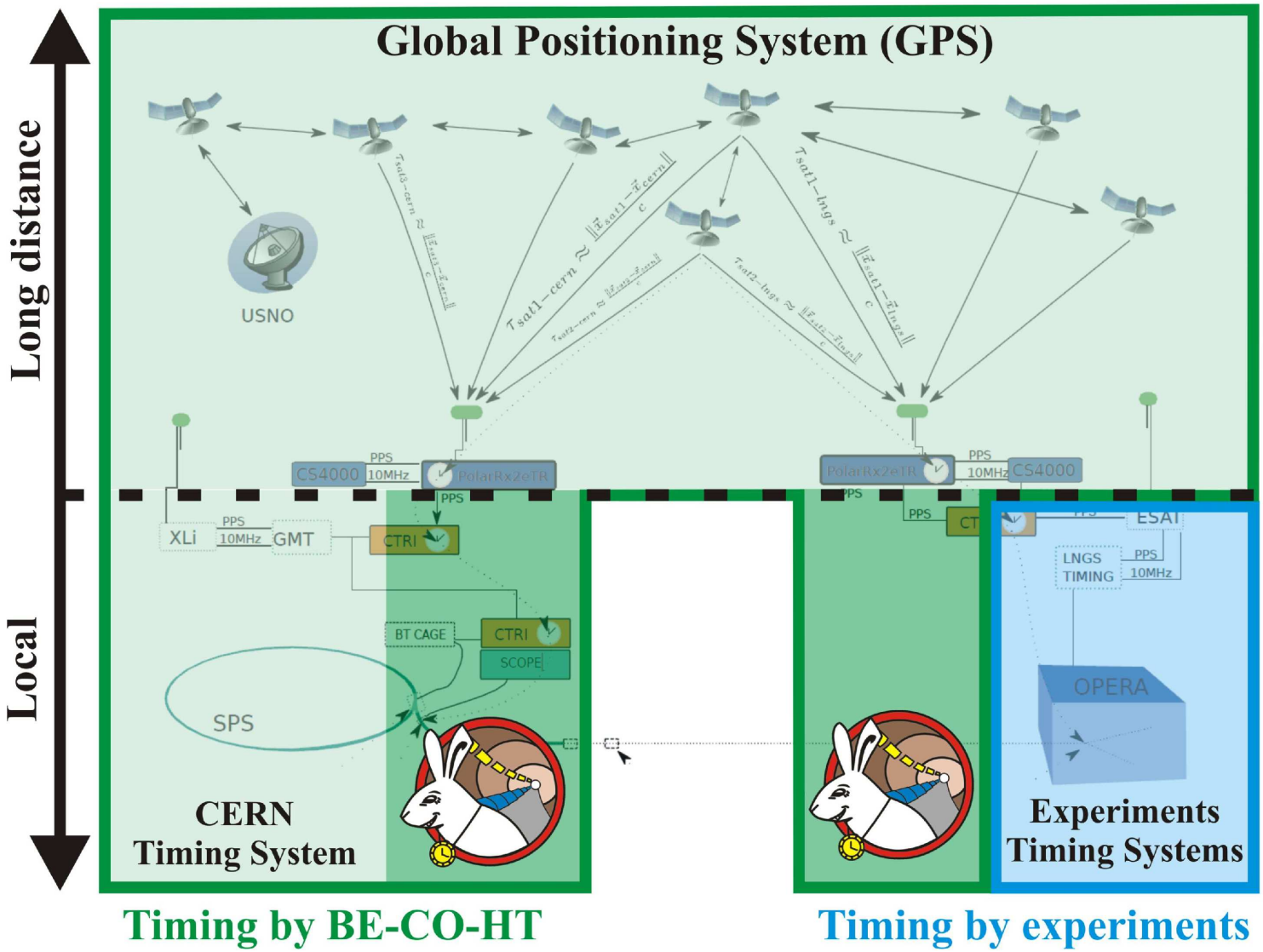
- Common clock in the entire network: no skew between ADCs.
- Ability to sample with different clocks
- Internal time triggers or external asynchronous triggers time tagged with a TDC

CERN Neutrinos to Gran Sasso project



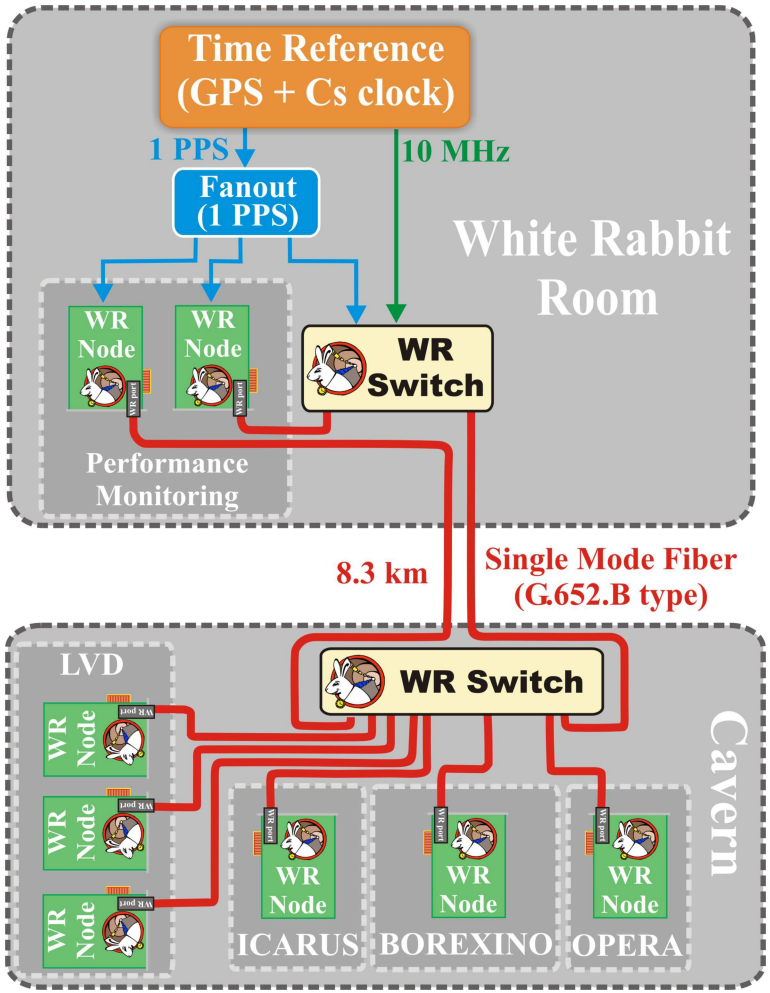
- Investigation of neutrino oscillation
- Time of Flight measurement

CERN Neutrinos to Gran Sasso project



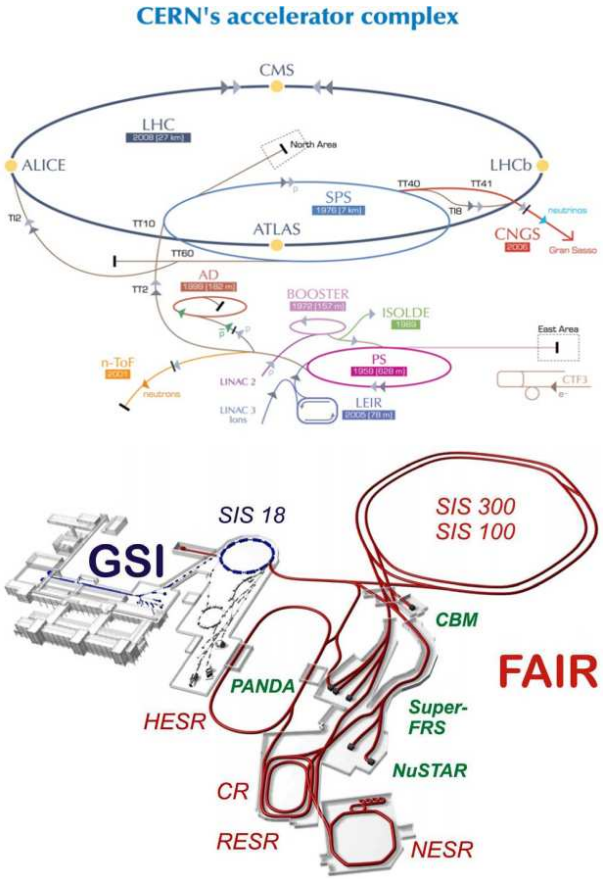
CERN Neutrinos to Gran Sasso project

- WR transferring UTC from GPS receiver to the measurement point
- 8km of fiber between WR Switches
- WR Switch in the cavern serves various experiments
- Performance monitoring
- Results from ∼31 days:
  - Accuracy: 0.517 ns
  - Precision: 0.119ns (std. dev)



Other WR Applications

- CERN and GSI



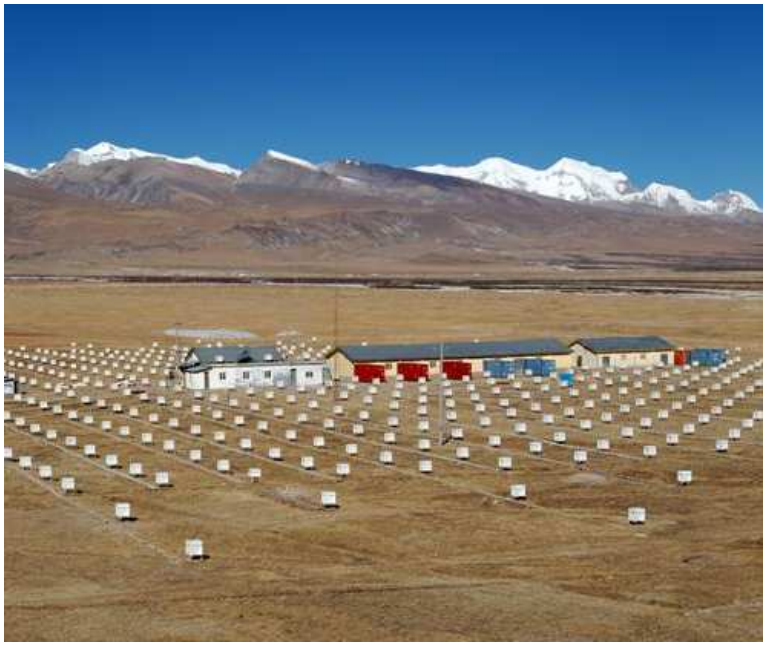
Other WR Applications

- CERN and GSI
- HiSCORE: Gamma&Cosmic-Ray experiment

- Institute for Nuclear Research of the Russian Academy of Sciences
- Moscow State University
- Irkutsk State University

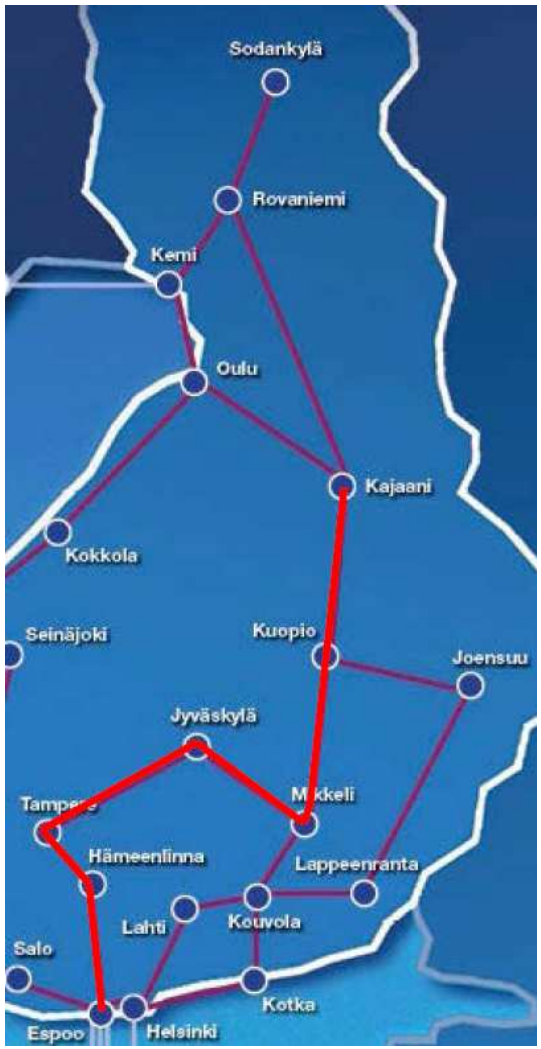
Other WR Applications

- CERN and GSI
- HiSCORE: Gamma&Cosmic-Ray experiment
- The Large High Altitude Air Shower Observatory



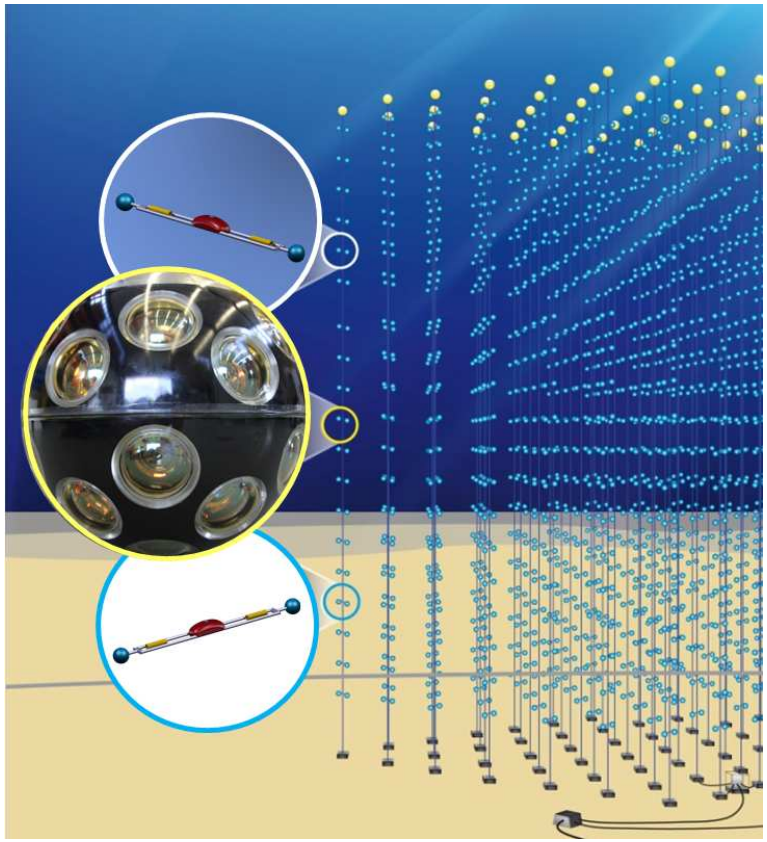
Other WR Applications

- CERN and GSI
- HiSCORE: Gamma&Cosmic-Ray experiment
- The Large High Altitude Air Shower Observatory
- MIKES: Centre for metrology and accreditation



Other WR Applications

- CERN and GSI
- HiSCORE: Gamma&Cosmic-Ray experiment
- The Large High Altitude Air Shower Observatory
- MIKES: Centre for metrology and accreditation
- KM3NET: European deep-sea research infrastructure



Full list of WR users:

`http://www.ohwr.org/projects/white-rabbit/wiki/WRUsers`

Outline

White Rabbit Family

Successful international collaboration of institutes, universities and companies

WR Users:

`http://www.ohwr.org/projects/white-rabbit/wiki/WRUsers`

White Rabbit Family

Successful international collaboration of institutes, universities and companies

WR Users:

http://www.ohwr.org/projects/white-rabbit/wiki/WRUsers

Pushing frontiers

- Scientific, open (H/W & S/W), with commercial support

Pushing frontiers

- Scientific, open (H/W & S/W), with commercial support
- More applications than ever expected

Pushing frontiers

- Scientific, open (H/W & S/W), with commercial support
- More applications than ever expected
- A versatile solution for general control and data acquisition

Pushing frontiers

- Scientific, open (H/W & S/W), with commercial support
- More applications than ever expected
- A versatile solution for general control and data acquisition
- Fulfilling all our needs in synchronization and determinism

Pushing frontiers

- Scientific, open (H/W & S/W), with commercial support
- More applications than ever expected
- A versatile solution for general control and data acquisition
- Fulfilling all our needs in synchronization and determinism
- Standard-compatible and standard-extending

Pushing frontiers

- Scientific, open (H/W & S/W), with commercial support
- More applications than ever expected
- A versatile solution for general control and data acquisition
- Fulfilling all our needs in synchronization and determinism
- Standard-compatible and standard-extending
- Active participation in IEEE1588 revision process

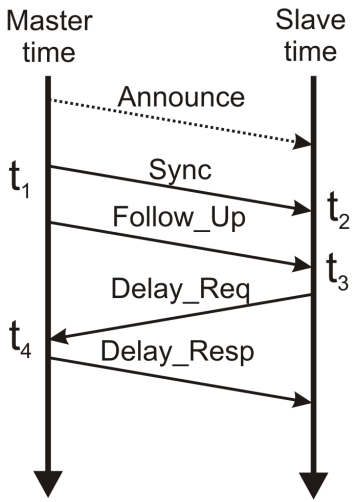
Pushing frontiers

- Scientific, open (H/W & S/W), with commercial support
- More applications than ever expected
- A versatile solution for general control and data acquisition
- Fulfilling all our needs in synchronization and determinism
- Standard-compatible and standard-extending
- Active participation in IEEE1588 revision process

Thank you

More information:
http://www.ohwr.org/projects/white-rabbit/wiki

Link Delay Model

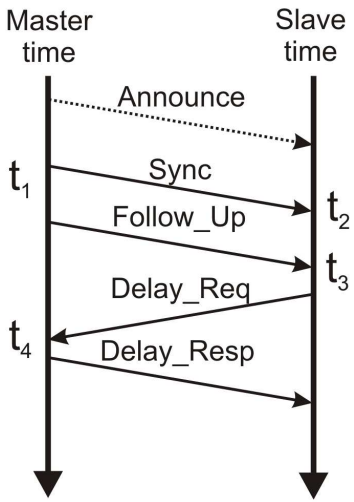


$$\mu = \frac{(t_4 - t_1) - (t_3 - t_2)}{2}$$

$$delay_{ms} = \mu + asymmetry$$

$$offset_{ms} = t_2 - (t_1 + delay_{ms})$$

Link Delay Model



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

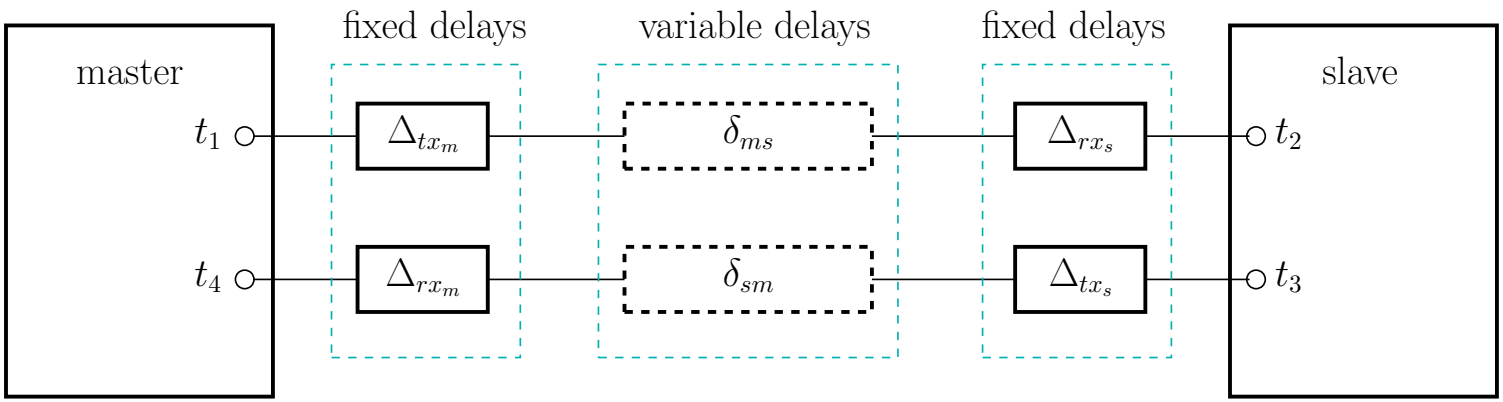
?

$$delay_{ms} = \mu + asymmetry$$

$$offset_{ms} = t_2 - (t_1 + delay_{ms})$$

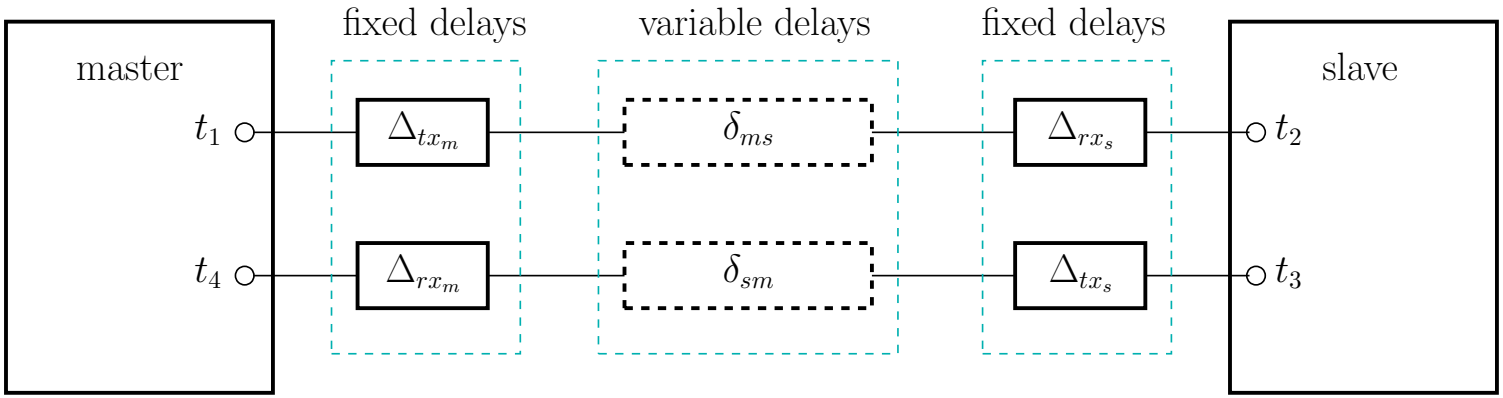
Link Delay Model

$$delay_{ms} = \Delta_{tx_m} + \delta_{ms} + \Delta_{rx_s}$$
$$delay_{sm} = \Delta_{tx_s} + \delta_{sm} + \Delta_{rx_m}$$



Link Delay Model

$$delay_{ms} = \Delta_{tx_m} + \delta_{ms} + \Delta_{rx_s}$$
$$delay_{sm} = \Delta_{tx_s} + \delta_{sm} + \Delta_{rx_m}$$

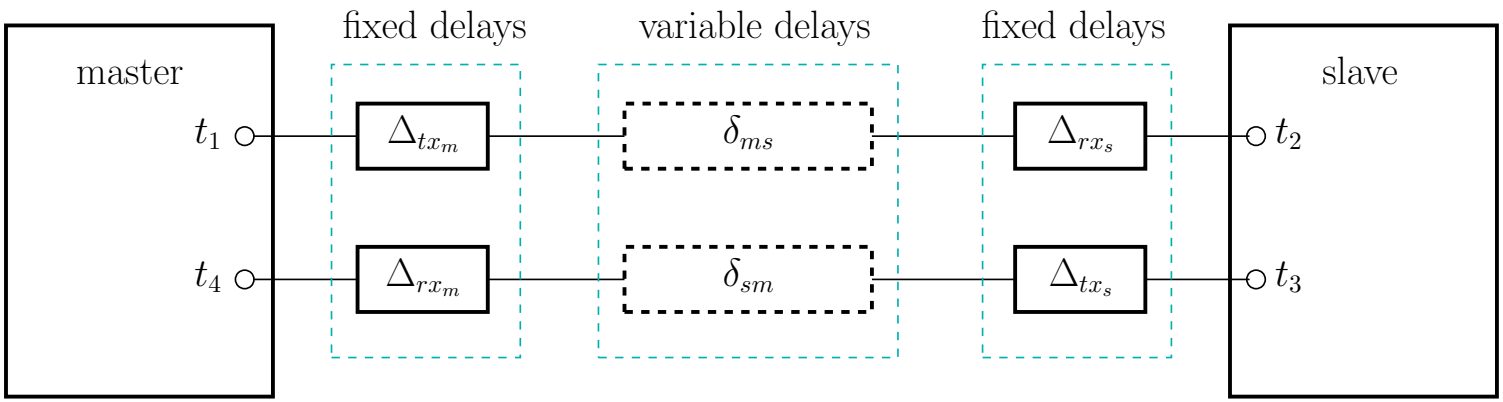


Relative Delay Coefficient (α)
for 1000base-X over a Single-mode Optical Fibre

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

Link Delay Model

$$delay_{ms} = \Delta_{tx_m} + \delta_{ms} + \Delta_{rx_s}$$
$$delay_{sm} = \Delta_{tx_s} + \delta_{sm} + \Delta_{rx_m}$$

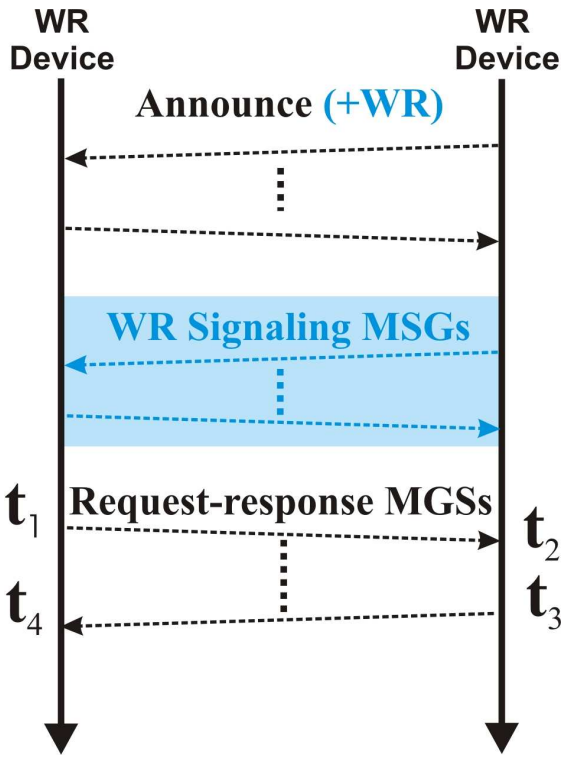


Measuring fixed delays is hard

but we use mathematical tricks for that - *WR Calibration procedure* (`http://www.ohwr.org/documents/213`)

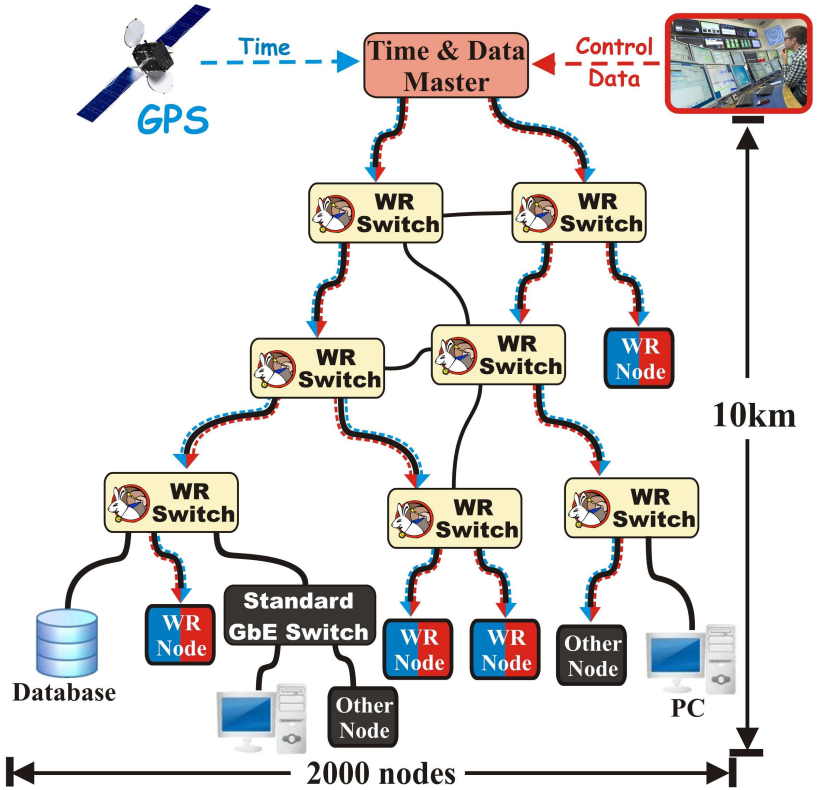
White Rabbit extension to PTP

- White Rabbit requires:
  - WR-specific states
  - Exchange of WR-specific information
  - asymmetry estimation based on Link Delay Model
- WR PTP
  - PTP extensions mechanisms
  - Enhanced precision t_1, t_2, t_3, t_4
  - Correction for asymmetry
  - Interoperability with PTP gear



White Rabbit Network

- White Rabbit Switch
- White Rabbit Node (White Rabbit PTP Core)

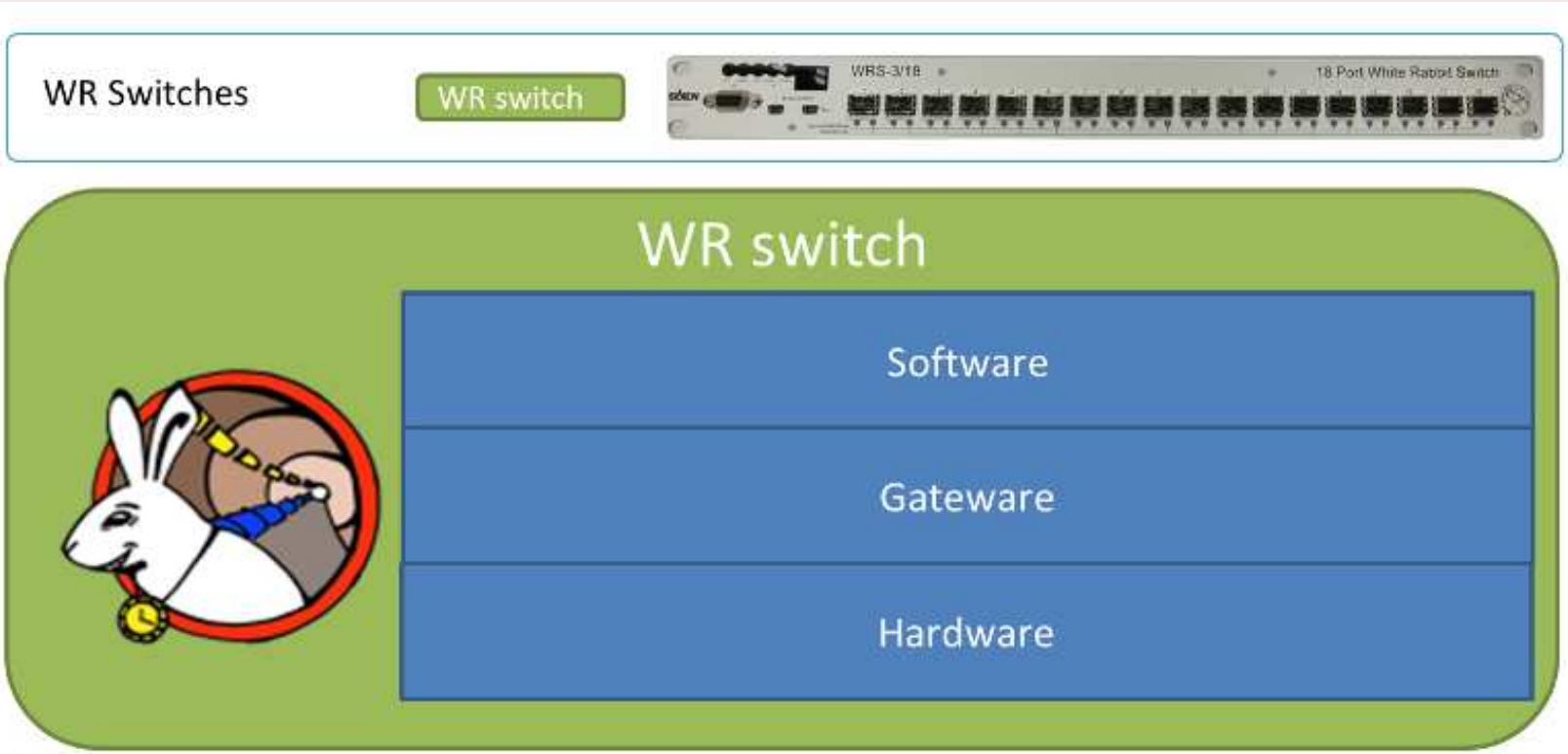


White Rabbit Switch

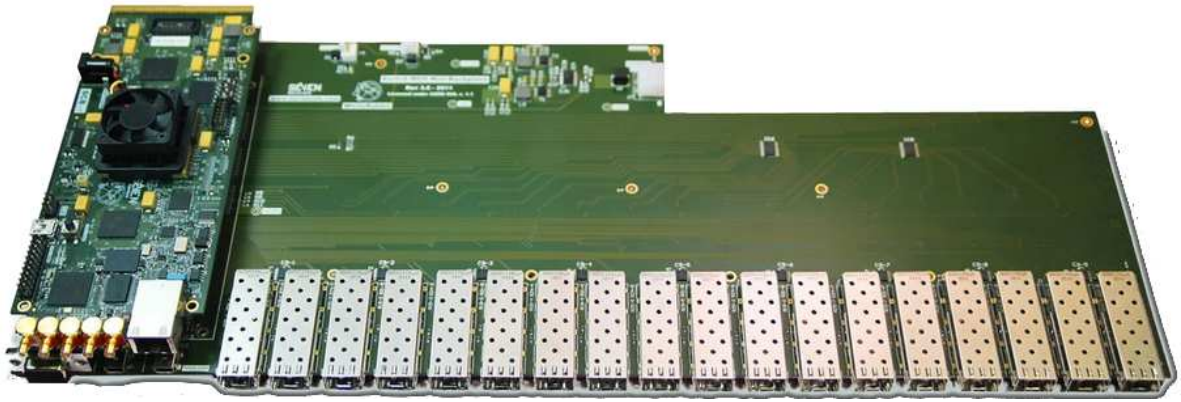
Functionality of a professional Gigabit Ethernet Switch

with White Rabbit extensions

3 layers of design:



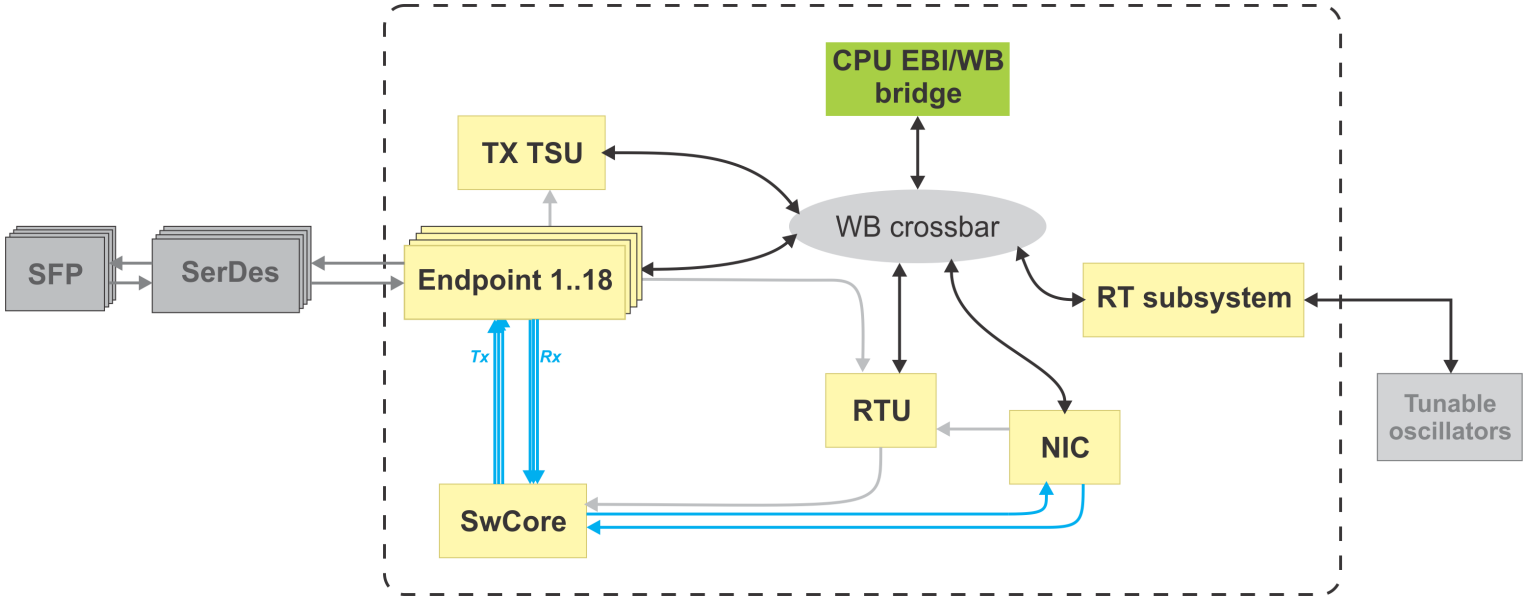
WR Switch: hardware



- Xilinx Virtex 6, Atmel AT91SAM9G45
- 18 cages for Gigabit SFPs, 10/100 Ethernet management port
- 5 SMC connectors (1-PPS in/out, CLK in/out)
- designed and produced by *Seven Solutions* in cooperation with CERN

WR Switch: gateware

Implemented in Xilinx Virtex6 FPGA:



WR Switch: software

Running on ARM processor:

- Embedded Linux
- kernel 2.6.39 with patches and modules for HDL components
- Hardware Abstraction Layer
- RTU daemon
- PTP daemon with WR extension
- CLI and SNMP support coming

White Rabbit Node - WR PTP Core

HDL IP-Core

developed on Xilinx Spartan 6 but not tied to Xilinx

it is a fancy Ethernet MAC

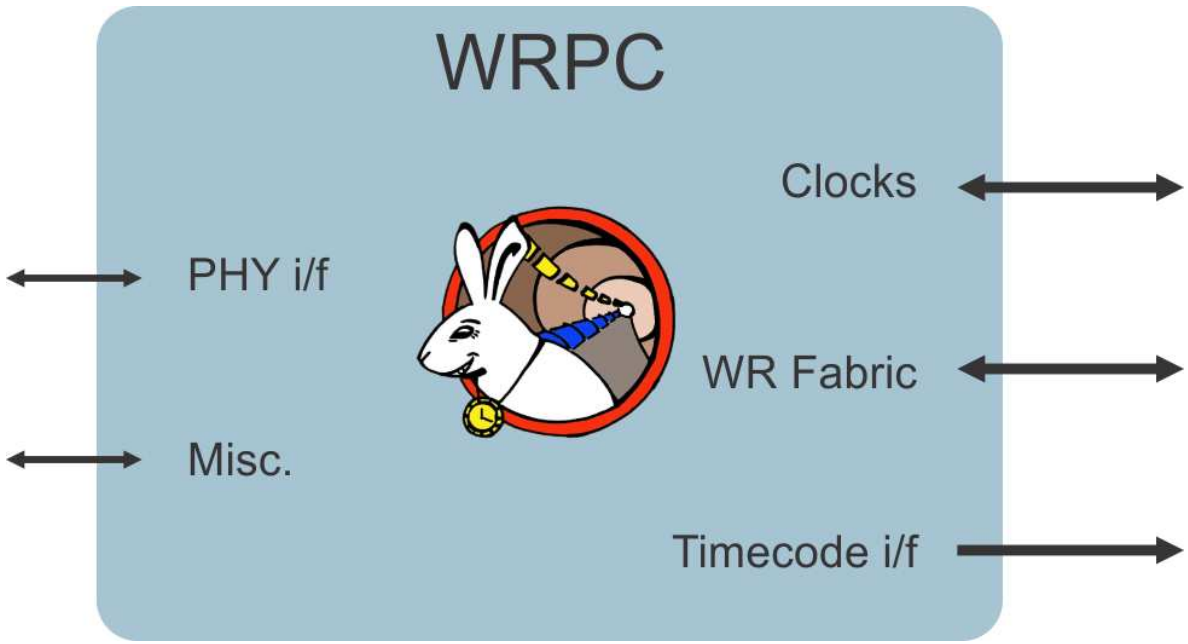
- interfaces user-defined module sending/receiving Ethernet frames with PHY layer
- provides precise timing by implementing WR protocol

ready to be integrated in user's devices

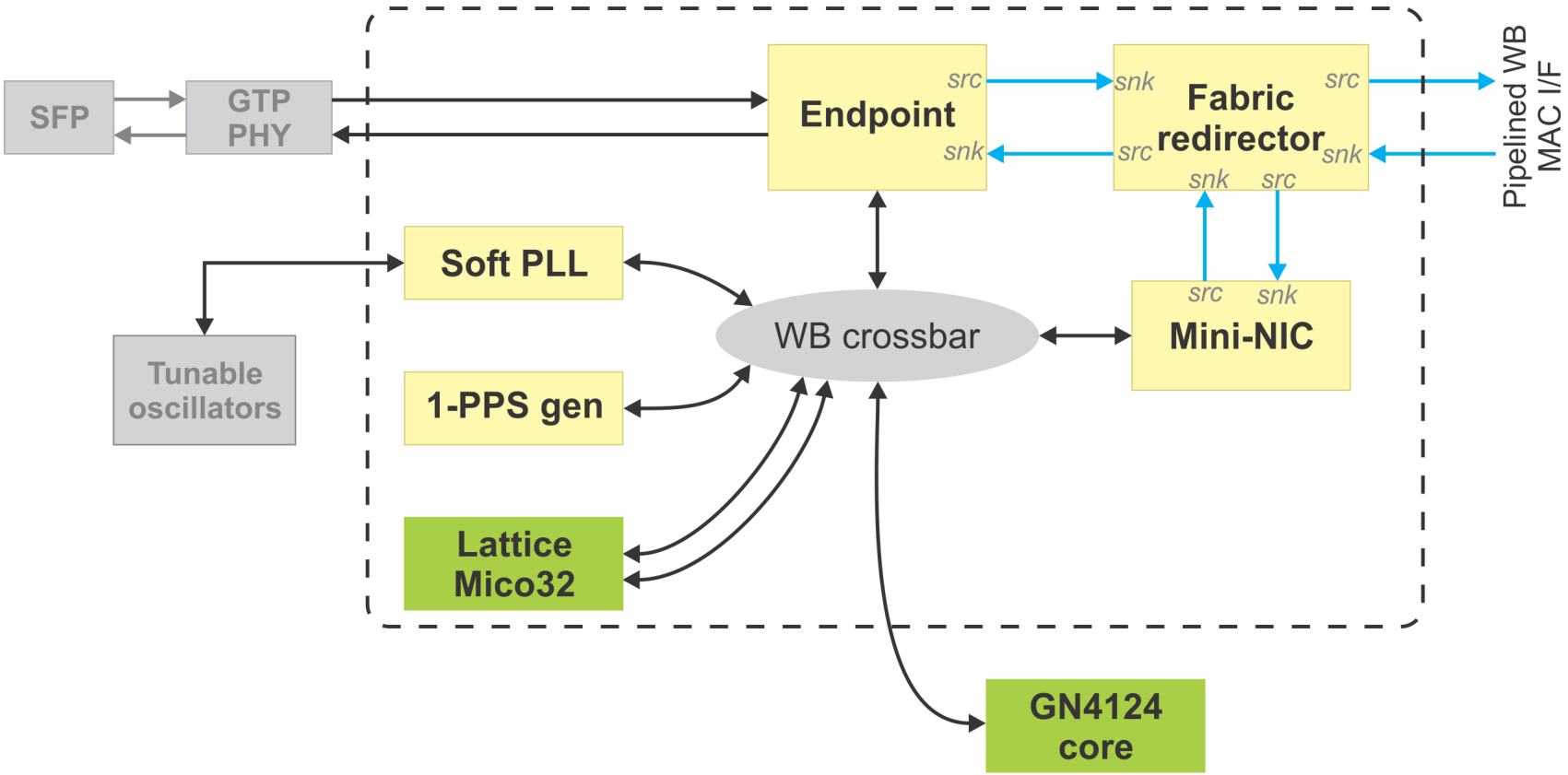
requires only two tunable oscillators and EEPROM to store the configuration

WR PTP Core: interfaces

- clocks and reset
- frame interface (WR Fabric)
- timecode and 1-PPS output
- PHY interface (*GTP*/*GTX* tested and supported)
- I^2C, 1-Wire, UART, GPIO



WR PTP Core: HDL design



WR PTP Core: HDL design