

## Scientific Monograph

# Methods to Increase Reliability and Ensure Determinism in a White Rabbit Network

Lipinski, Maciej Marek (Warsaw U. of Tech.)



The EuCARD-2 Enhanced European Coordination for Accelerator Research & Development project is co-funded by the partners and the European Commission under Capacities 7th Framework Programme, Grant Agreement 312453.

This work is part of EuCARD-2 Work Package 1: **Management and Communication (MANCOM)**.

The electronic version of this EuCARD-2 Publication is available via the EuCARD-2 web site <http://eucard2.web.cern.ch/> or on the CERN Document Server at the following URL:  
<<http://cds.cern.ch/search?p=CERN-THESIS-2016-283>>

WARSAW UNIVERSITY OF TECHNOLOGY

**Faculty of Electronics  
and Information Technology**



**Ph.D THESIS**

Maciej Lipiński, M.Sc.

**Methods to Increase Reliability  
and Ensure Determinism  
in a White Rabbit Network**

Supervisor

Professor Ryszard Romaniuk, Ph.D, D.Sc.

Warsaw, 2016







**POLITECHNIKA WARSZAWSKA**

**Wydział Elektroniki  
i Technik Informatycznych**



**ROZPRAWA DOKTORSKA**

**mgr inż. Maciej Lipiński**

**Metody zwiększenia niezawodności  
i zapewnienia determinizmu  
w sieci White Rabbit**

Promotor

Prof. dr hab. inż. Ryszard Romaniuk

Warszawa, 2016



*Thanks to Z.  
The thesis was developed within the White Rabbit Project,  
the author would like to thank the WR Team at CERN for all the support.*



---

# Abstract

---

The current control and timing system at the European Organization for Nuclear Research (CERN) has been serving its accelerators for several decades and is reaching its design limits. In preparation is the next generation system, called White Rabbit. White Rabbit is intentionally based on commonly used networking technologies to ensure the flexibility, maintainability and wide commercial support that were missing in the old system. The new system is meant to coordinate the actions of thousands of individual devices constituting the CERN accelerator complex in a timely manner for several decades.

At the time of White Rabbit's conception in 2008, none of the existing networking standards could provide the unprecedented characteristics required by a future-proof accelerator control and timing system. Therefore, the most suitable solutions needed to be enhanced with new specialized services. Two non-existent enhancements are proposed and developed in the context of this thesis. The first ensures that critical information to coordinate accelerator actions is delivered to all the devices within a specified time – it ensures the network's determinism. The second increases the probability that the system works without interruption for at least a year – it increases the network's reliability.

The methods to provide the two specialized services, along with the new network design guidelines, are proposed in this thesis. The following standards were chosen for enhancements: Ethernet-based Virtual Local Area Network, Shortest Path Bridging and Precision Time Protocol. These standard technologies are used in a network arranged into a topology that provides many alternative paths. As a result of this work, specialized hardware to switch between the faulty and the backup paths fast enough so the accelerators' devices experience no disruption has been implemented. Similarly, the specialized hardware ensures the timely delivery of information through the network. The proposed strategy describes the network topologies in which the devised methods are allowed. This strategy and methods permit the White Rabbit network to control all CERN accelerators.

The performance of the developed enhancements exceeds that of the standard solutions while being interoperable with them. When switching between alternative paths, both the synchronisation performance and the reduction in data loss are improved 1000-fold compared to the best-known implementations of the original standards. The worst-case time of information

delivery is improved fourfold. The White Rabbit devices implementing the developed enhancements can be interconnected and work with standard implementations, but in such cases the benefits of the enhancements are lost.

The fact that White Rabbit is based on and interoperable with well-known technologies makes it a preferred generic solution to many control, acquisition and synchronisation problems. At CERN, with certain enhancements presented in this thesis, White Rabbit is currently used for diagnostics of the Large Hadron Collider accelerator and the distribution of the magnetic field in the Proton Synchrotron accelerator. Its integration into the CERN control and timing system will take place over the coming years. Outside CERN, the proposed network strategy is already being applied in the design of the Large High Altitude Air Shower Observatory in Tibet where White Rabbit is used for synchronisation and data acquisition. The list of other applications is long and growing. In fact, White Rabbit has the potential to become a commonly used technology not only for controlling accelerators, but also in other applications. In many cases, the specialized services presented in this thesis will play an important role.

---

# Streszczenie

---

Europejska Organizacja Badań Jądrowych, CERN, stosuje w swoich akceleratorach od dziesięcioleci ten sam system sterowania i synchronizowania urządzeń. Granice możliwości konstrukcyjnych tego systemu zostały osiągnięte. Dlatego, rozpoczęte zostały prace nad systemem nowej generacji nazwanym White Rabbit (tłumaczenie: Biały Królik). White Rabbit oparty jest na powszechnie znanych i stosowanych technologiach sieciowych. Głównym celem takiego podejścia jest utworzenie systemu, który byłby elastyczny, uniwersalny i zapewniał wieloletnie wsparcie przez firmy komercyjne. Tego wszystkiego nie oferuje obecny system sterowania. Przez następnych kilka dekad, nowy system będzie miał za zadanie koordynowanie działań tysięcy urządzeń składających się na kompleks akceleratorów w CERN.

Kiedy w 2008 roku narodził się pomysł stworzenia White Rabbit'a, żaden z istniejących standardów sieciowych nie był w stanie spełnić bezprecedensowych wymagań stawianych przyszłej sieci sterowania i synchronizowania akceleratorów. Postanowiono więc, że do stworzenia White Rabbit'a zaadoptowane zostaną standardy, które pozwalają na wymagane udoskonalenia. Udoskonalenia te powstają w formie specjalnych usług (specialized services) pozwalających na sterowanie i synchronizowanie akceleratorów. W ramach niniejszej pracy doktorskiej opracowane i zaimplementowane zostały dwie specjalne usługi, które do tej pory nie istniały w wymaganej formie. Pierwsza z nich zapewnia dostarczenie w ściśle określonym czasie krytycznych informacji, które koordynują działania akceleratorów – innymi słowy zapewnia determinizm sieci. Druga z usług zwiększa prawdopodobieństwo bezawaryjnego działania sieci przez co najmniej rok – innymi słowy zwiększa niezawodność sieci.

W ramach doktoratu zrealizowane zostały dwie wspomniane usługi oraz opracowano wytyczne projektowania sieci, która te usługi wykorzystuje. Realizacja usług wymagała rozszerzenia następujących standardów: Virtual Local Area Network, Shortest Path Bridging oraz Precision Time Protocol. W sieci White Rabbit standardy te zastosowano w nadmiarowej topologii, która posiada wiele alternatywnych ścieżek przesyłu czasu i danych. Opracowane rozwiązania pozwalają na przełączenie się między ścieżką, która uległa awarii a jej zapasową ścieżką tak szybko, że akceleratory działając nie zauważają zmiany. Co więcej, opracowane rozwiązania zapewniają, iż czas potrzebny na dostarczenie krytycznych informacji nigdy nie przekroczy założonej wartości, nawet w przypadku przełączania ścieżek. Zawarta w doktoracie strategia za-



stosowania opracowanych rozwiązań opisuje topologie sieci, dla których rozwiązania te mogą być zastosowane. Stosując się do tej strategii można stworzyć sieć White Rabbit, która pozwala na sterowanie i synchronizowanie wszystkich akceleratorów w CERN.

Wyniki, osiągnięte przez opracowane w ramach doktoratu specjalne usługi, przewyższają te osiągnięte przez oryginalne standardy, na których bazie usługi te powstały. Zarówno jakość synchronizacji jak i ilość utraconych danych podczas przełączania między alternatywnymi ścieżkami zostały poprawione 1000-krotnie w porównaniu do najlepszych implementacji oryginalnych standardów. W najgorszym wypadku, czas dostarczenia krytycznych informacji przez sieć White Rabbit jest czterokrotnie lepszy od innych rozwiązań. Urządzenia White Rabbit, które implementują rozwiązania opracowane w ramach doktoratu, współdziałają z urządzeniami, które implementują oryginalne standardy. W przypadku takiego połączenia osiągnięte są wyniki oryginalnych standardów.

Fakt, iż urządzenia White Rabbit współdziałają z urządzeniami implementującymi popularne standardy bazowe sprawił, że White Rabbit stał się dość powszechnie używanym systemem do sterowania, zbierania danych i synchronizowania. W CERN niektóre z zaprezentowanych w tym doktoracie rozwiązań, jeszcze przed jego ukończeniem, zastosowane zostały do diagnostyki Wielkiego Zderzacza Hadronów (Large Hadron Collider) i do przesyłu pola magnetycznego w Synchrotronie Protonów (Proton Synchrotron). Docelowe zastosowanie White Rabbit'a jako sieci do sterowania i synchronizowania akceleratorów, będzie realizowane w przeciągu kolejnych kilku lat. Poza CERN, zaproponowana strategia projektowania sieci White Rabbit jest stosowana w Obserwatorium Promieniowania Kosmicznego na Dużej Wysokości (Large High Altitude Air Shower Observatory) w Tybecie. W obserwatorium LHAASO, sieć White Rabbit jest stosowana do synchronizacji i zbierania danych. Lista innych zastosowań jest długa i ciągle się powiększa. Jest wielce prawdopodobne, iż White Rabbit stanie się technologią powszechnie stosowaną w wielu dziedzinach, nie tylko do sterowania i synchronizowania akceleratorów. W wielu z tych zastosowań, usługi opracowane w ramach tego doktoratu będą miały niemałe znaczenie.

---

# Contents

---

<b>Abstract</b>	<b>i</b>
<b>Streszczenie</b>	<b>iii</b>
<b>Contents</b>	<b>v</b>
<b>1 Introduction</b>	<b>1</b>
1.1 CERN Control and Timing . . . . .	3
1.1.1 LHC Injection Chain (LIC) . . . . .	3
1.1.2 The General Machine Timing (GMT) System . . . . .	4
1.1.3 Current System – Limitations . . . . .	5
1.1.4 New System – Requirements . . . . .	6
1.2 White Rabbit . . . . .	8
1.2.1 Meaning of Reliability and Determinism in a White Rabbit Network . . . . .	10
1.2.2 White Rabbit Technologies, Standards and Switch Architecture . . . . .	11
1.2.3 White Rabbit Applications . . . . .	13
<b>2 Aim and Scope of the Thesis</b>	<b>15</b>
2.1 The Goals and Achievements of This Thesis . . . . .	16
2.2 Limitations of Methods Proposed in This Thesis . . . . .	16
2.3 Thesis . . . . .	17
<b>3 Reliability and Determinism in Ethernet Networks - State of the Art</b>	<b>19</b>
3.1 Redundant Network Topologies . . . . .	20
3.2 Data Transfer in Redundant Networks . . . . .	22
3.2.1 Distributed Protocols . . . . .	22
3.2.2 Software-Defined Networking . . . . .	24
3.3 Reliable Data Transmission . . . . .	25
3.4 Determinism and Latency . . . . .	26
3.5 Reliable Time and Frequency Transfer . . . . .	28
3.5.1 L1 Syntonisation in Redundant Networks . . . . .	28
3.5.2 Precision Time Protocol in Redundant Networks . . . . .	28
3.6 Summary . . . . .	30

<b>4</b>	<b>Strategy to Increase Reliability and Ensure Determinism of a WR Network</b>	<b>31</b>
4.1	Basic Assumptions for WR Network Operation . . . . .	32
4.2	Basis Assumptions for Reliability Calculations . . . . .	34
4.3	Target Reliability of the White Rabbit Network . . . . .	36
4.4	Factors Contributing to the Reliability of a WR Network . . . . .	38
4.5	Reliability of Network Connectivity . . . . .	39
4.5.1	Reliability in Non-Redundant WR Network . . . . .	39
4.5.2	Reliability in Redundant WR Network . . . . .	41
4.6	Reliability of Message Transmission . . . . .	44
4.6.1	Expected Message Loss . . . . .	44
4.6.2	Mitigation of Control Message Loss . . . . .	45
4.6.3	Forward Error Correction Schema and its Reliability Analysis . . . . .	46
4.7	Congestion-Less Transmission . . . . .	48
4.8	Latency and Timing Performance . . . . .	49
4.8.1	Synchronisation Performance . . . . .	49
4.8.2	WR Network Latency Performance . . . . .	51
4.8.3	WR Switch Latency Performance . . . . .	54
4.9	Proposed Strategy . . . . .	55
<b>5</b>	<b>Methods and Algorithms for Synchronisation Resilience</b>	<b>57</b>
5.1	Background . . . . .	58
5.1.1	Synchronisation and Syntonisation in White Rabbit . . . . .	58
5.1.2	White Rabbit Phase-Locked Loop (WR PLL) . . . . .	60
5.1.3	WR Requirements for Frequency and Time Transfer . . . . .	61
5.2	Support for Network Redundancy and Seamless Reconfiguration . . . . .	62
5.2.1	Problem Statement . . . . .	62
5.2.2	Architecture . . . . .	63
5.2.3	Switchover Theoretical Model and Simulation . . . . .	67
5.2.4	Failure Detection . . . . .	71
5.2.5	Backup Port Control Loop . . . . .	72
5.2.6	Reconfiguration in Cascaded Partially Redundant Topologies . . . . .	72
5.2.7	Holdover . . . . .	74
5.2.8	Propagation of clockClass . . . . .	76
5.2.9	PTP Support and Configuration for Seamless Switchover . . . . .	76
5.3	Implementation . . . . .	80
5.3.1	Multi-Channel WR PLL . . . . .	81
5.3.2	PTP Daemon (PPSi) . . . . .	82
5.3.3	PTP Support Unit . . . . .	82
5.4	Limitations of the Used Methods, Alternative Solutions . . . . .	83
5.5	Usefulness and Applications of Proposed Methods . . . . .	83

5.6	Measurements and Tests . . . . .	84
5.6.1	Direct Redundant Connection (scenario a) . . . . .	86
5.6.2	Direct Redundant Connection in Cascade of Switches (scenario b) . . .	87
5.6.3	Indirect Redundant Connection (scenario c) . . . . .	88
5.6.4	Indirect Redundant Connection in Cascade of Switches (scenario d) . .	89
5.6.5	Test and Measurement Summary . . . . .	90
<b>6</b>	<b>Methods to Support Seamless Redundancy and Determinism for Data</b>	<b>91</b>
6.1	Background . . . . .	93
6.1.1	Network Redundancy . . . . .	93
6.1.2	Determinism . . . . .	95
6.2	Support for Seamless Redundancy and Determinism . . . . .	97
6.2.1	Problem Statement . . . . .	97
6.2.2	Architecture . . . . .	98
6.2.3	Fast Switchover Between Pre-Configured Active and Backup Ports . . .	99
6.2.4	Applicable Network Topologies and Pseudo-Multipath Spanning Tree .	104
6.2.5	Lossless Reconfiguration when Adding an Element to the Network . . .	110
6.2.6	Deterministic Data Forwarding . . . . .	112
6.3	Implementation . . . . .	116
6.3.1	Topology Resolution Unit . . . . .	117
6.3.2	Deterministic Data Forwarding . . . . .	121
6.4	Limitations of the Used Methods, Alternative Solutions . . . . .	126
6.5	Usefulness and Applications of Proposed Methods . . . . .	127
6.6	Measurements and Tests . . . . .	128
6.6.1	Determinism . . . . .	128
6.6.2	Fast Switchover Between Redundant Paths . . . . .	134
6.7	Summary . . . . .	136
<b>7</b>	<b>WR-Based Control and Timing Network</b>	<b>137</b>
7.1	Network Design . . . . .	138
7.1.1	Physical Network Design . . . . .	139
7.1.2	Time and Frequency Distribution . . . . .	141
7.1.3	Data Distribution . . . . .	142
7.2	Characteristics of the Proposed Network . . . . .	146
7.2.1	Synchronisation . . . . .	146
7.2.2	Determinism . . . . .	147
7.2.3	Seamless Redundancy . . . . .	149
7.2.4	Reliability . . . . .	150
7.3	Summary . . . . .	152
<b>8</b>	<b>Conclusions</b>	<b>153</b>

<b>A</b>	<b>White Rabbit</b>	<b>157</b>
A.1	Basic Technologies and Standards . . . . .	158
A.2	White Rabbit Switch . . . . .	161
<b>B</b>	<b>Explanation of Requirements for the New CERN Control and Timing System</b>	<b>163</b>
<b>C</b>	<b>Network Reliability and Availability Values for all Considered Topologies</b>	<b>165</b>
<b>D</b>	<b>Network Reliability Calculations for Doubly-Redundant WR Network</b>	<b>167</b>
D.1	Probability Laws Used in the Calculations . . . . .	168
D.2	Symbols Used in the Calculations . . . . .	169
D.3	Calculations . . . . .	170
D.3.1	Layer 1 . . . . .	170
D.3.2	Layer 2 . . . . .	171
D.3.3	Layer 3 . . . . .	172
D.3.4	Layer 4 . . . . .	173
D.3.5	Layer 5 - WR Network Reliability . . . . .	174
D.4	Probability Calculations Results and their Interpretation . . . . .	175
<b>E</b>	<b>Forward Error Correction Header</b>	<b>177</b>
<b>F</b>	<b>Latency of Control Messages in a WR-Based Control and Timing Network</b>	<b>179</b>
<b>G</b>	<b>PTP Support Unit</b>	<b>181</b>
<b>H</b>	<b>Time Switchover Test Results</b>	<b>185</b>
H.1	Direct Redundant Connection (scenario a) . . . . .	186
H.2	Direct Redundant Connection in Cascade of Switches (scenario b) . . . . .	188
H.3	Indirect Redundant Connection (scenario c) . . . . .	189
H.4	Indirect Redundant Connection in Cascade of Switches (scenario d) . . . . .	190
<b>I</b>	<b>Basic Configuration of the Reference WR-Based Control and Timing Network</b>	<b>191</b>
<b>J</b>	<b>Reflections and Recommendations</b>	<b>195</b>
	<b>Bibliography</b>	<b>197</b>
	<b>List of Figures</b>	<b>207</b>
	<b>List of Tables</b>	<b>210</b>
	<b>List of Listings</b>	<b>213</b>
	<b>List of Abbreviations</b>	<b>215</b>

# Chapter 1

## Introduction

The European Organization for Nuclear Research (CERN) [1] operates a complex of accelerators that provides beams of particles to a number of physics experiments. Detectors installed at the experiment sites enable research on the fundamental laws of nature and the structure of the universe. A key system controlling and coordinating CERN's accelerators is undergoing renovation, which includes a major contribution that is the outcome of this thesis.

Figure 1.1 shows that each CERN accelerator is a part of a chain that provides different types of beams to many destinations. The chain starts with a linear accelerator, LINAC2 or

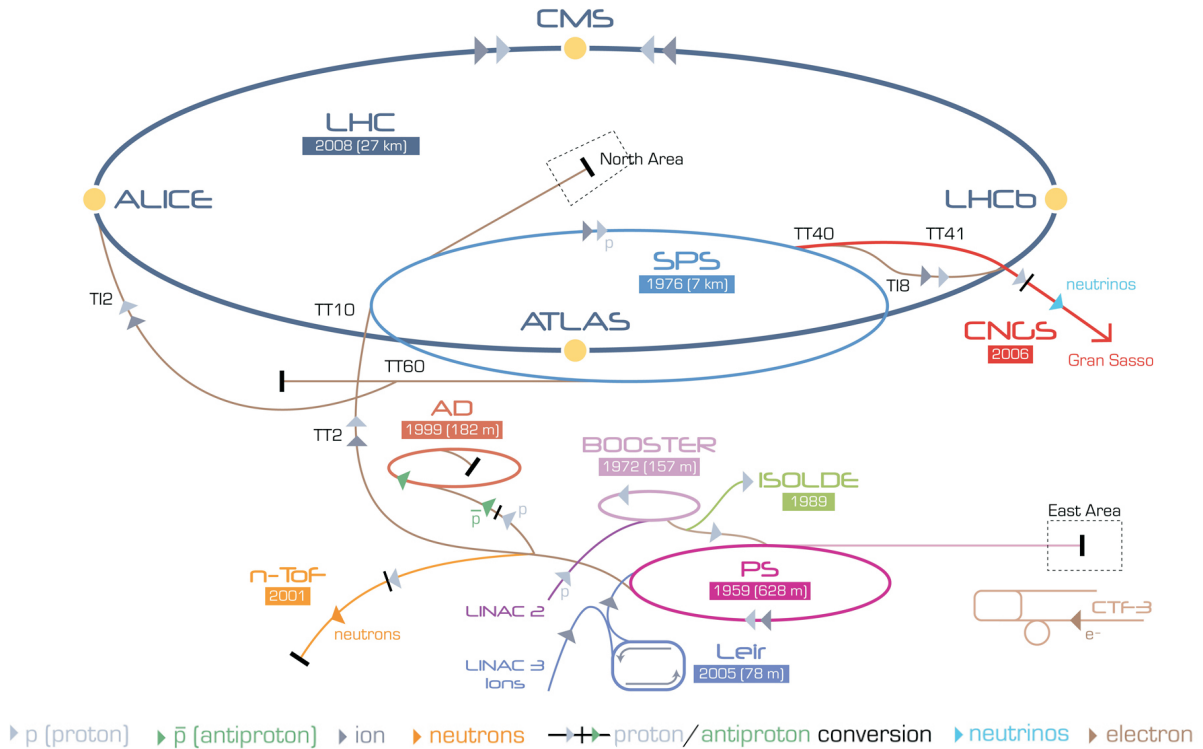


Figure 1.1: CERN accelerator complex.

LINAC3, which provides an initial boost of energy to a beam of protons or ions, respectively. The particles are then injected into a circular accelerator, which further increases their energy. The process is repeated and the particles are injected into subsequent accelerators until they reach their final destination, which can be the Large Hadron Collider (LHC), the Antiproton Decelerator (AD), the Isotope Separator On Line DEtector (ISOLDE), or the CERN Neutrinos to Gran Sasso (CNGS) extraction line.

The individual devices constituting each of the accelerators in the CERN complex must operate in a precise and timely way to provide destination-specific beams with certain required characteristics: energy, luminosity and spatial structure. This is possible thanks to a common control and timing system that synchronises and coordinates devices in all the accelerators. The system provides a common notion of Coordinated Universal Time (UTC) throughout the complex and allows the triggering of simultaneous actions in remote locations, effectively turning the accelerator complex into a large hard real-time<sup>1</sup> distributed system. Any failure in this system interrupts scientific experiments and endangers expensive accelerator equipment. Therefore high reliability is required.

General Machine Timing (GMT) [2, 3, 4] is the current control and timing system in charge of the synchronisation and coordination of the accelerator devices. This system is reaching its limits in terms of performance and applicability. A new system called White Rabbit (WR) [5, 6, 7], intended to gradually replace GMT, is being developed. The probability of WR's undisturbed operation and the timely delivery of control information using WR, in other words its reliability and determinism, are the main themes of this thesis.

---

<sup>1</sup>A real-time system guarantees response or execution within specified time constraints, often referred to as deadlines. While a missed deadline in a soft real-time system deteriorates the usefulness of the system, a missed deadline in a hard real-time system causes total system failure.

## 1.1 CERN Control and Timing

The CERN control and timing system is a critical component in a toolkit [8] of generic solutions that allow controlling all types of devices installed in the CERN accelerators. Upgrading a single component of this toolkit requires a global understanding of all its pieces, their interactions and applications. This section explains the role of the current control and timing system, GMT, in the delivery of a beam to the LHC by the accelerators of the LHC Injection Chain (LIC), which is one of GMT's major applications. The section then explains GMT's limitations, and the requirements for its successor, WR.

### 1.1.1 LHC Injection Chain (LIC)

The LIC is like a factory whose end products are beams; manufacturing each beam requires a series of sequential actions in the chain of accelerators. The LIC accelerators include the proton and the ion linear accelerators, the Low Energy Ion Ring (LEIR), the 1 GeV Booster, the 26 GeV Proton Synchrotron (PS) and the 450 GeV Super Proton Synchrotron (SPS), all depicted in Figure 1.1. A series of sequential actions taken for a particular beam in a single accelerator is called a cycle. As depicted in Figure 1.2, each cycle consists of:

- **injection** – a beam is injected from the preceding accelerator
- **energy ramp-up** – the energy of the beam is increased
- **extraction** – the beam is extracted to the subsequent accelerator or the final destination.

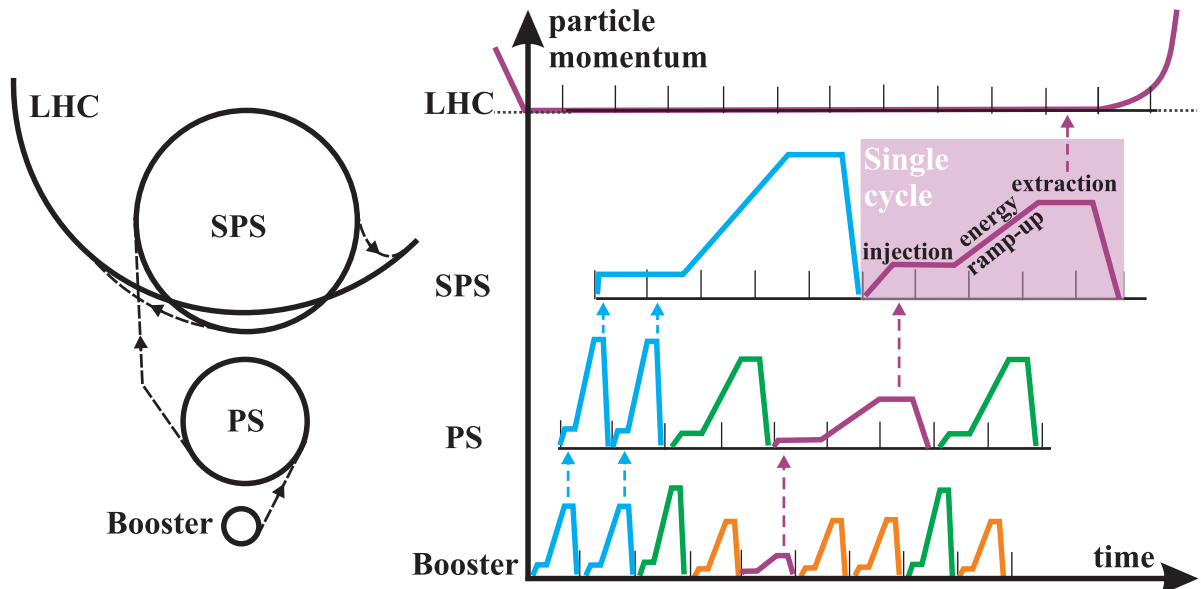


Figure 1.2: LHC Injection Chain sequential cycles.

A series of sequential cycles in the cascaded accelerators provides a beam customised to its end user, namely the LHC, AD, ISOLDE and CNGS. For example, the series of violet cycles in



Figure 1.2 shows how a beam traverses subsequent accelerators in order to fill the LHC; in each cycle, an accelerator is first filled with particles (injection), then the energy is increased (energy ramp-up), and finally the beam is ejected to the next accelerator (extraction), the last being the LHC.

The cycles are managed, or "played" by a central controller<sup>2</sup> that sends out messages in a timely manner over the GMT network to hundreds of devices installed in the CERN accelerator complex.

### 1.1.2 The General Machine Timing (GMT) System

The GMT system enables the coordination and synchronisation of actions in all CERN accelerators by transmitting time, frequency and messages to the devices constituting the accelerators.

All the devices connected to the GMT network are synchronised with UTC over Global Positioning System (GPS). The GMT generator is fed with the signal from a GPS-disciplined oscillator and uses its stable clock to encode the messages sent out to the GMT receivers. The GMT receivers recover the clock from the data stream and use it to ensure that their local notion of time is within 25 ns of the generator's notion of time, with a jitter<sup>3</sup> of less than 1 ns [3]. Any transmission delays introduced by the network are manually compensated for at each receiver using a portable caesium atomic clock.

The messages sent over the GMT network carry "events" that are used to "play" the accelerator cycles. The messages make it to the receivers within a precisely predictable time. Each receiver is programmed to identify events of interest by their identification (ID) number and react to the received events by performing specialised actions. Such actions include arming counters, producing pulses on the receiver's front panel connectors or triggering interrupts<sup>4</sup> in the microprocessors to synchronise the software applications with processes happening in the accelerators. Depending on the settings of the receiver, these actions are produced immediately upon the arrival of an event or fired precisely when the next millisecond elapses.

As an example, Figure 1.3 depicts a simplified injection of the beam from the PS to the SPS accelerator. Events are depicted by red dots on the time axis. The events trigger such actions as starting and stopping an energy ramp. A single event, identified by ID number, can trigger different actions in different locations of different accelerators. Event number 3 in Figure 1.3, for example, marks the end of the energy ramp in the PS and the start of the ramp in the SPS accelerator.

This simplified example shows how the GMT system is used to control accelerator devices (e.g. kickers, magnets) and synchronise their actions by distributing time, frequency and events in the entire CERN complex. These actions and their synchronisation requirements are constantly growing and are approaching the limits of GMT.

---

<sup>2</sup>The Central Beam and Cycle Manager (CBCM) [9].

<sup>3</sup>Deviation from reference periodicity; in this context, it is quantified with peak-to-peak value.

<sup>4</sup> A signal emitted by hardware or software indicating an event that needs immediate attention.

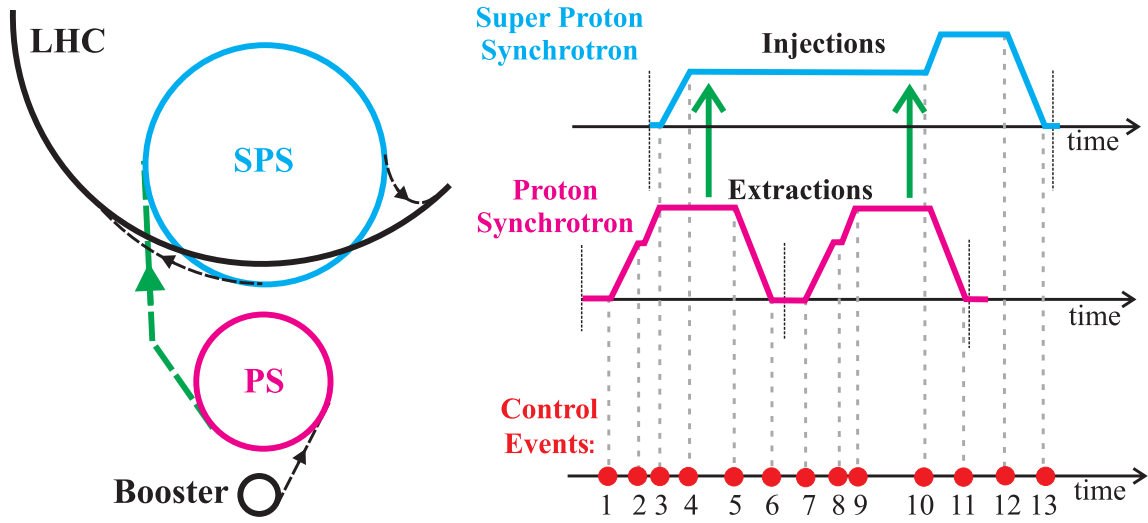


Figure 1.3: Concept of CERN accelerator cycles and events

### 1.1.3 Current System – Limitations

Created over 20 years ago for the PS complex, then adapted for the SPS and finally for the LHC, GMT is a reliable and flexible system that serves its purpose very well. However, it has limitations resulting from its design and technological choices.

The main drawbacks of GMT include limited bandwidth and lack of bi-directionality. The messages are sent by the timing generator to the timing receivers over a network composed of cascaded unidirectional fan-outs connected with fibre or copper cables. Consequently, any feedback information, such as interlocks or diagnostics, requires a parallel Ethernet [10] network. This additional network is also needed to send the configuration of the timing receivers, as the bandwidth of GMT is limited and reserved only for events. In fact, the bandwidth limitation is compensated through separate GMT networks for different accelerators, which adds complexity to the system pertaining to cabling and software.

The unidirectional nature of the network makes it impossible to automatically compensate the time of the receivers for the cable delay, the time it takes for the signal to travel through the cable. Therefore, an expensive, manual and error-prone campaign is required to accurately synchronise all the timing receivers. This method of one-time calibration does not take into account time-error caused by temperature variations, which amounts to nanoseconds on long distances, such as in the LHC.

Another important drawback is the fact that the system is custom-made and based on old RS-422 technology [11], which requires costly in-house support and tailor-made spare parts, resulting in maintenance challenges

Although the drawbacks of GMT are not critical and the system is working successfully, it was decided that the growing demands of its end users in the ever-expanding accelerator complex justifies its renovation. The new system's requirements take into account the legacy constraints and the experience of using GMT, and anticipated future needs.

### 1.1.4 New System – Requirements

The new system must provide the functionality of the current system and address its drawbacks, while enabling a smooth upgrade and fulfilling CERN’s anticipated needs. In particular, the new system is required to connect 2000 devices spatially distributed within the area of LHC, providing their receivers with control and timing, as defined for the GMT. This control and timing includes delivery of messages that can trigger simultaneous processes in spatially distributed accelerator devices and synchronisation of these devices.

The control requires delivery of dedicated messages, called control messages, in a deterministic and reliable manner. A control message is sent periodically and contains an assembly of events that trigger actions of accelerator devices. Deterministic delivery of control messages translates into a guaranteed and calculable maximum time of delivery from a central controller (timing generator) to all the receiving accelerator devices through a predictable path. This maximum time is called *upper-bound latency*. It is required to be less than 1 millisecond from the scheduling of an event in the controller to the occurrence of the event’s triggered action in the receiver. The deterministic delivery of control messages, which are critical traffic, is required while allowing bi-directional flows of other traffic. The control messages are usually sent from a controller to a large number of receivers at predictable times, so special attention must be paid to one-to-many periodic traffic. However, unpredictable and sporadic critical traffic must also be taken into account; such traffic might carry information about, for example, interlocks. The reliability of the new system is quantified by the probability of not losing more than one control message throughout a year, provided synchronisation is maintained during that time.

The timing requires synchronisation with sub-ns accuracy and better than 50 ps precision [12] in a reliable manner. Accuracy is interpreted in this thesis as the absolute value of the average time error (TE) between the reference and the device synchronised to this reference. Precision is interpreted in this thesis as the standard deviation of the TE. The required accuracy and precision exceeded CERN’s needs at the time the new system was specified, and are currently required in many applications [13, 14]. The reliability of synchronisation is quantified by the probability of maintaining the specified synchronisation throughout a year of operation.

The new control and timing system to fulfil the above requirements, which are summarised in Table 1.1 and detailed in Appendix B, is being developed under the name of White Rabbit. White Rabbit was also chosen to upgrade the system that provides real-time distribution of

Requirement	Value(s)
Network size: maximum distance and number of receivers	10km & 2000
Accuracy [12]: $ avg(TE) $	sub-ns
Precision [12]: $sdev(TE)$	sub-50ps
Control message size	1200-6000 bytes
Maximum number of control messages lost per year	1
Upper-bound latency through a network & a single switch	1ms & 10 $\mu$ s

Table 1.1: CERN requirements for the new WR-based systems (see also Appendix B).

the value of the main bending magnetic field in a synchrotron, called Btrain [15]. This system requires an upper-bound latency through a switch of less than  $10\ \mu\text{s}$  with minimum latency variation, a requirement that was included in the scope of this thesis.

## 1.2 White Rabbit

The WR project is a multilaboratory, multicompany and multinational collaboration to develop new technology providing a versatile solution for control and data acquisition systems. The project was started within an effort to renovate the current CERN control and timing system. Since then, it has expanded beyond this initial application. One of the reasons for the expansion is the open source paradigm used in the project for the development of hardware, gateway<sup>5</sup> and software. Openness facilitates collaboration and encourages new contributions. This section gives an overview of the WR project. It explains the terms *determinism* and *reliability* used in the context of WR, introduces the basic technologies employed in the project, and gives examples of current WR applications. The main component of a WR network, the WR switch, has been enhanced in the context of this thesis, so its architecture before the enhancements is described.

The architecture and implementation of the WR switch are the keys to meeting the requirements listed in Table 1.1. One of the main aims of the WR project is to build a system while using – and extending where needed – existing standards. Ethernet [10] and Bridged Local Area Network (LAN) [16] were chosen as the basic family of networking technologies and standards. Based on these standards, WR should provide a set of standard-compatible specialised services that can be used in any combination according to a user’s needs. Using these WR add-on services together should allow the creation of a control and timing network that fulfils the requirements outlined in subsection 1.1.4 and summarised in Table 1.1. These specialised services include:

- distribution of time and frequency with sub-ns synchronisation accuracy
- deterministic transmission of control messages (i.e. assembly of events)
- means to ensure the high reliability of the WR network.

Time and frequency are delivered from a reference to all the WR nodes. As depicted in Figure 1.4, the timing master of a WR network (a switch or a node) receives its notion of time and frequency from a reference clock, e.g. a GPS-disciplined oscillator. Time and frequency are distributed through the WR network in a hierarchical manner over a spanning tree of switches to all the WR nodes. Consequently, the clocks of all the WR devices (nodes and switches) in the network work with the same frequency and their time counters are incremented at the same instant to within a nanosecond. Thus the time of all the WR devices is traceable to the reference. The specialised service that distributes time and frequency with sub-ns accuracy was developed by Tomasz Wlostowski in the context of his thesis [7].

---

<sup>5</sup>Gateway describes the configuration of logic gates in a Field Programmable Gate Array (FPGA) chip. This configuration is specified using Hardware Description Language (HDL) such as Very High Speed Integrated Circuits Hardware Description Language (VHDL) or Verilog.

The specialised services to provide determinism and reliability have been developed in the context of this thesis. A detailed explanation of the meaning of these two terms in the WR network is provided in the next subsection.

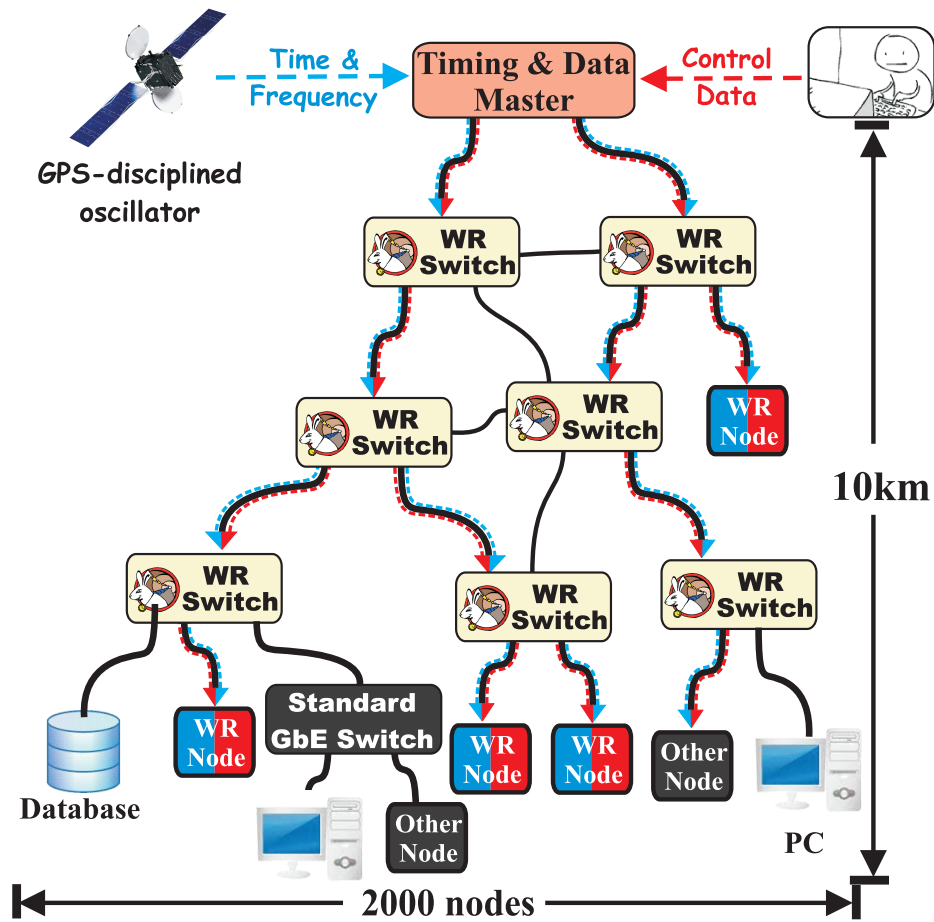


Figure 1.4: White Rabbit network.

### 1.2.1 Meaning of Reliability and Determinism in a White Rabbit Network

The definitions of network *reliability* and *determinism* depend on their application. Ethernet-based LANs, such as WR, are used to connect devices in a variety of applications: from home and office networks, through large-scale installations in data centres and industrial automation, to networks controlling vehicles and airplanes. While the malfunction of an enterprise network for a few seconds is unnoticeable, a disruption of a few milliseconds in a factory can have disastrous consequences, and even short congestion in a car's control network can be fatal for drivers. In the two latter examples, the network is used to control devices and must be deterministic and reliable. A deterministic system is predictable: it provides calculable and consistent characteristics of operation that are required by the application, e.g. latency of data transmission and its rate of delivery, throughput. A reliable system performs its required functions under stated conditions for a specified period of time [17].

The components of reliability are identified as performance and fault tolerance [18]. Performance in this context depends on the ability of the network to meet peak application requirements. It is especially important in deterministic networks. While many applications might accept occasional and temporary drops in performance, the ones that require determinism cannot accept any anomalies. Fault tolerance means that the elements of the network can break down without affecting the services provided to the application. It is achieved through redundancy, e.g. spare cables and switches. The time it takes for the network to detect a faulty element and activate a spare is called *failover*. Network redundancy increases reliability only if the application can accept the failover time and during that time unacceptable performance anomaly does not occur. Applications with zero-failover require seamless redundancy.

The determinism in a WR network concerns the transmission of control messages that are used to coordinate devices in the accelerators. The control messages are transmitted periodically from the data master (WR node) to all the WR nodes. Occasionally, a WR node can send a control message to the data master. The WR network is deterministic if the control messages are always delivered within the specified upper-bound latency regardless of any other traffic. If a control message is delivered too late, it is considered lost.

The reliability of a WR network concerns undisturbed deterministic transmission of the control messages and continuous synchronisation over a period of one year. The WR network is considered reliable if the control messages are delivered with the specified upper-bound latency to all the required WR nodes, and all these nodes are synchronised with the required accuracy and precision. During one year of operation, not more than one single control message can be lost. This means that the reliability in the WR network requires seamless redundancy.

None of the technologies and standards that constitute the basis of WR provides the required determinism or seamless redundancy. These technologies are described in the next subsection

## 1.2.2 White Rabbit Technologies, Standards and Switch Architecture

This subsection provides very basic information<sup>6</sup> about the technologies and standards that are implemented in the WR devices. The focus is on the WR switch, and its architecture is explained. The described technologies and standards as well as the WR switch architecture have been extended in the context of this thesis. Therefore, their comprehension is essential to understand the methods described later.

A WR network, depicted in Figure 1.4, is a Bridged Local Area Network (IEEE 802.1D [16]) that uses Ethernet (IEEE 802.3 [10]) to interconnect WR switches and nodes, and Precision Time Protocol (PTP, IEEE 1588-2008 [19]) to synchronise them. The components and architecture of the networking devices, such as WR switches and nodes, are commonly characterised using the Open Systems Interconnection (OSI) model [20]. Therefore, Figure 1.5 depicts the WR switch architecture and the implemented IEEE standards in the context of the OSI model. Regarding implementation, the WR switch architecture is divided into:

- Time-critical tasks implemented in gateway (G/W) in an FPGA chip<sup>7</sup>.
- Non-time-critical tasks implemented in software (S/W) and running in a microprocessor (CPU).

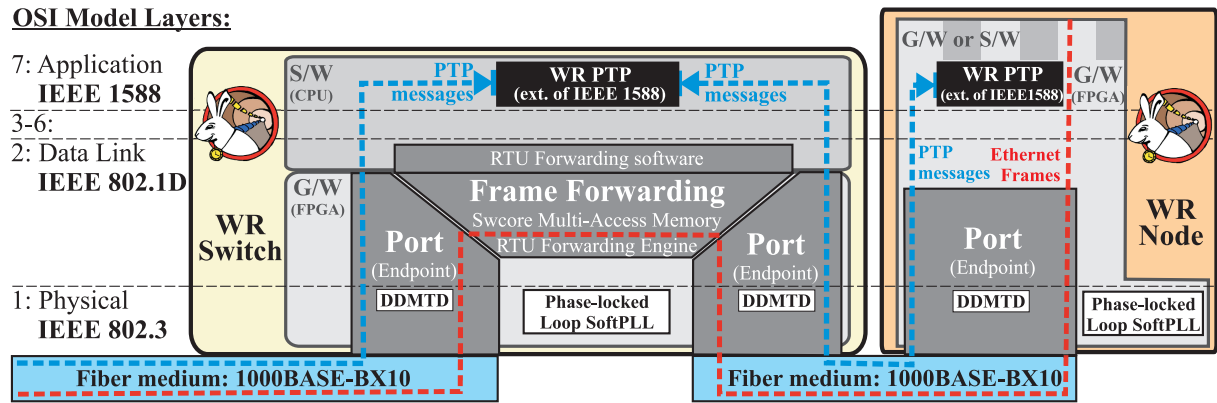


Figure 1.5: Simplified architecture of a two-port WR switch.

Concerning the distribution of data, WR switches forward Ethernet frames (red dashed line in Figure 1.5) between any WR nodes. Every node can communicate with any other; there is no hierarchy. As depicted in Figure 1.5, for data forwarding, the switch implements the first two layers of the OSI model, and so is called a Layer 2 (L2) switch. When an Ethernet frame is sent by a WR node, it is received on one of the WR switch's ports. The functionalities of a port described in IEEE 802.1D and 802.3 are implemented in gateway by a module called Endpoint. The Ethernet frame that has been received on a port and that does not carry a Precision Time Protocol (PTP) message is forwarded to one or more ports by two cooperating

<sup>6</sup>A detailed description is provided in Appendix A.

<sup>7</sup>Field Programmable Gate Array.



gateway modules that implement IEEE 802.1D: the RTU Forwarding Engine (RTU) and the Swcore Multi-Access Memory (Swcore). The former decides to which port(s) the frame should be forwarded, the latter implements specialised memory to temporarily store the frame. All three gateway modules perform tasks that are time-critical and influence switch latency. The non-time-critical task of "learning" to which ports an Ethernet frame should be forwarded is implemented in the RTU software.

Concerning time, the WR switches allow hierarchical distribution of time through the WR network from the timing master to all the WR nodes, as depicted in Figure 1.4. This synchronisation is performed using the WR extension to PTP (WR PTP). Each WR switch participates in the distribution of time by propagating the time from its slave port to its master ports. Any port of the WR switch can be either a master or a slave, depending on the configuration or an exchange of PTP messages. The WR PTP uses PTP messages (blue dashed lines in Figure 1.5) to communicate between WR devices; these messages are not forwarded between ports by the switches. Instead, PTP messages are forwarded to the CPU that implements the WR PTP protocol. This software implementation is aided by two gateway models: the Digital Dual Mixer Time-Difference (DDMTD) [21] phase detector and a dedicated WR phase-locked loop (WR PLL), called also SoftPLL. The former allows precise measurement of the transmission and reception times, the timestamps, of PTP messages at each port. The former and the latter together allow precise syntonisation to the clock signal carried in the physical Layer 1 (L1); this process is called L1 syntonisation.

The architecture and implementation of a WR node, depicted in Figure 1.5, is application-specific, and it reuses the following components of the WR switch: the Endpoint module with the DDMTD phase detector, the WR PLL, and the implementation of a single-port WR PTP protocol that runs in a soft core processor inside the FPGA. The WR nodes determine the application of a WR network. So, the WR-based control and timing network uses the same switches but different nodes than the WR-Brain network. The application-specific part of the WR node can be implemented in software or gateway; it uses the time and frequency provided by WR PTP and/or communicates using Ethernet frames. As the WR node is not a subject of this thesis, it is not described in detail.

While WR nodes are developed for particular applications, all applications use the very same WR switch that is at the heart of all WR networks. A number of such applications is described in the next subsection.

### 1.2.3 White Rabbit Applications

Since the start of the WR project, its applications have grown far beyond that initially intended for CERN's control and timing system, and even outside of accelerator systems [3, 22].

At CERN, in addition to its application in the new control and timing system, WR was chosen to upgrade a system called Btrain [15]. The Btrain is used to transport the value of a magnetic field in real-time from a reference magnet to other sub-systems in the accelerator. This allows the changes in the particles' energy to be followed coherently by the changes in the magnetic field. Such real-time distribution requires deterministic latency with low variation, provided by WR. The new control and timing system and the new Btrain system are two examples of WR applications at CERN.

An example of a WR application outside of accelerators is the observation of showers of high-energy particles to study the universe. This is done by cosmic ray observatories that consist of detectors distributed over tens or hundreds of metres. By comparing the time of arrival of the particles observed by the spatially distributed detectors, it is possible to reconstruct the direction of their flight, provided the detectors are synchronised. The synchronisation required to achieve good angular resolution and precisely determine the trajectory of cosmic rays in the observatories is at the nanosecond or sub-nanosecond level. Observatories in Siberia [23] and China [14] have chosen WR, and it is being evaluated for other locations.

Similarly, neutrino detectors, observing elementary particles that interact only via the weak subatomic force, require accurate synchronisation of spatially distributed measurement units. A neutrino detector is being installed on the Mediterranean Sea bed [24], 100 km off the coast of Italy. The detector will consist of 4140 measurement units spanning  $1 \text{ km}^3$  to detect Cherenkov light that is emitted by neutrinos passing through the seawater. All the units will be synchronised with the onshore station using WR.

Another very interesting WR application is the long-distance time-transfer between national time laboratories, which provide official UTC for their countries. These laboratories provide the reference for time and frequency to institutions and companies, and compare their UTC with that of other national laboratories to calculate the global UTC. In most of these cases, long-distance time-transfer is required. It is currently performed mostly using common-view time-transfer [25] based on GPS. GPS, however, has its drawbacks and performance limitations, so an alternative ground-based method is needed. WR can provide better synchronisation performance than GPS and is being evaluated by laboratories in Finland [26], the Netherlands [27], and France [28].

Commercial applications of WR are likely in the future, such as in test and measurement equipment [29].

In a number of applications, White Rabbit is used in places where access is difficult or where system failure can possibly result in the damage of expensive equipment and in equally expensive downtime. Such applications require high reliability.



## Chapter 2

---

# Aim and Scope of the Thesis

---

The current control and timing system that is used at CERN to synchronise and control accelerator devices is ageing and needs a worthy successor. No suitable solution existed to meet CERN's requirement for the next generation control and timing system. It was therefore decided to create a new and fully open technology that is based on well-established standards and extends them where needed. This technology is developed within the WR project. The longevity and success of WR depends greatly on keeping the solution as standard as possible. Ethernet and Bridged Local Area Network (LAN) were chosen as the basic family of networking technologies and standards that are adapted to meet the requirements listed in Table 1.1.

WR is a long-term project involving many contributors, with applications to be deployed and used over several decades. The project started in 2008 with the work of Tomasz Wlostowski [7] which allowed meeting the synchronisation requirements of providing network-wide sub-nanosecond accuracy and synchronisation precision of below 50 ps. His work provides the basis for the WR switch implementation. The work of Grzegorz Daniluk [30] provided a compact and universal HDL IP core<sup>1</sup> that is the basis for the WR node implementation.

The following section outlines the goals and achievements of the current thesis in its investigation of methods to increase reliability and ensure determinism in a WR network.

---

<sup>1</sup>An HDL IP core is a module that is implemented using a Hardware Description Language (HDL) that is shared in the form of an Intellectual Property (IP) core.

## 2.1 The Goals and Achievements of This Thesis

The work for this thesis aims at providing standard-compatible specialised services that:

- ensure deterministic transmission of selected traffic through the WR network, and
- increase the reliability of the WR network

to meet the requirements in Table 1.1 and allow the application of WR as the next-generation control and timing system at CERN. The goal has been achieved.

In the context of this thesis, the following methods, considered to be the author's own work, have been developed to provide the specialised services:

**The strategy to increase the reliability of the WR network**, described in Chapter 4, proposes a topology type and redundancy mechanisms for enhancement suitable to fulfil the requirements in Table 1.1. These requirements are translated into reliability values to evaluate the proposed solutions;

**The support of seamless redundancy for synchronisation**, described in Chapter 5, proposes and implements methods to preserve sub-nanosecond synchronisation accuracy during network reconfiguration;

**The support of seamless redundancy and determinism for data transfer**, described in Chapter 6, proposes and implements methods to assist network redundancy and to minimise data loss due to network reconfiguration while ensuring the determinism of the network;

**The CERN reference network design**, described in Chapter 7, proposes a reference design for a WR-based control and timing network for the CERN accelerator complex using the proposed methods and strategy.

## 2.2 Limitations of Methods Proposed in This Thesis

Although this thesis attempts to provide a generic solution, its main goal is to meet the requirements of the next-generation CERN control and timing system and ensure a smooth transition from the current system. The work is thus limited to providing characteristics needed by the control and timing system in which the WR network is used in a very particular way and configuration. Moreover, it is assumed that the administrator of such a network has absolute control over the traffic and network topology. Therefore, the proposed methods can be applied for a limited number of topologies in carefully engineered and well-controlled networks.

The considerations and proposed methods are limited to modifications of software and gateware but not hardware. Hardware modifications are only suggested for further consideration.

The detailed limitations of each of the methods, and the network design, are explained in appropriate chapters.

## 2.3 Thesis

WR requirements are unprecedented in the technologies on which it is based. Concepts and standard solutions that increase the reliability of Ethernet-based Bridged LANs and provide deterministic transmission exist. However, none of them meets all the stringent characteristics required by CERN, and some of the solutions are mutually exclusive. The methods proposed in this thesis provide means to increase the reliability of synchronisation and data transmission in the WR network in a seamless, deterministic and standard-compatible manner. The reliability achievable using these methods cannot be provided by any other existing system based on similar technologies and standards.

This thesis first provides background information on the state of the art regarding synchronisation, reliability and determinism in Ethernet-based networks (Chapter 3). Based on the review of existing technologies and CERN's requirements, it then proposes a suitable network topology and a set of existing standards that, after being enhanced, theoretically allow to meet CERN's requirements (Chapter 4). These enhancements are proposed and concern synchronisation and data transmission, both described in separate chapters (Chapter 5 and Chapter 6). These chapters each provide specialised background information, describe the methods and their implementation, outline the limitations and usefulness of the proposed solutions, and provide results of their tests. The strategy and the methods are then used to design a WR-based control and timing network for all the accelerators at CERN as a reference (Chapter 7).



## Chapter 3

---

# Reliability and Determinism in Ethernet Networks - State of the Art

---

This chapter reviews currently available methods that ensure determinism and increase reliability of synchronisation and data transfer in Ethernet networks. The reliability and determinism are analysed in the context of the WR network and its requirements, which are explained in sections 1.2.1 and 1.1.4.

The chapter is organised as follows. Section 3.1 presents different types of redundant topologies for which dedicated protocols are introduced in the subsequent sections. In particular, section 3.2 describes protocols that manage transfer of data through the described redundant networks. These protocols are analysed focusing on the reconfiguration time and the allowed size of networks. Section 3.3 reviews different methods that introduce redundancy of data rather than redundancy of network to prevent data loss. Regardless of the type of redundancy used, a message is considered lost if it makes it too late through a deterministic network. Therefore, section 3.4 reviews methods to ensure network determinism in terms of latency. Finally, section 3.5 reviews existing methods that support the redundant topologies introduced in the first section. These methods are analysed focusing on the synchronisation performance during reconfiguration.

The characteristics and performance of all the described existing methods are summarized and compared with the CERN requirements in section 3.6. It clearly shows that none of these solutions can meet the CERN requirements.



## 3.1 Redundant Network Topologies

Different types of redundant topologies used in Ethernet Local Area Networks (LANs) to increase their reliability are depicted in Figure 3.1. Each topology is briefly described.

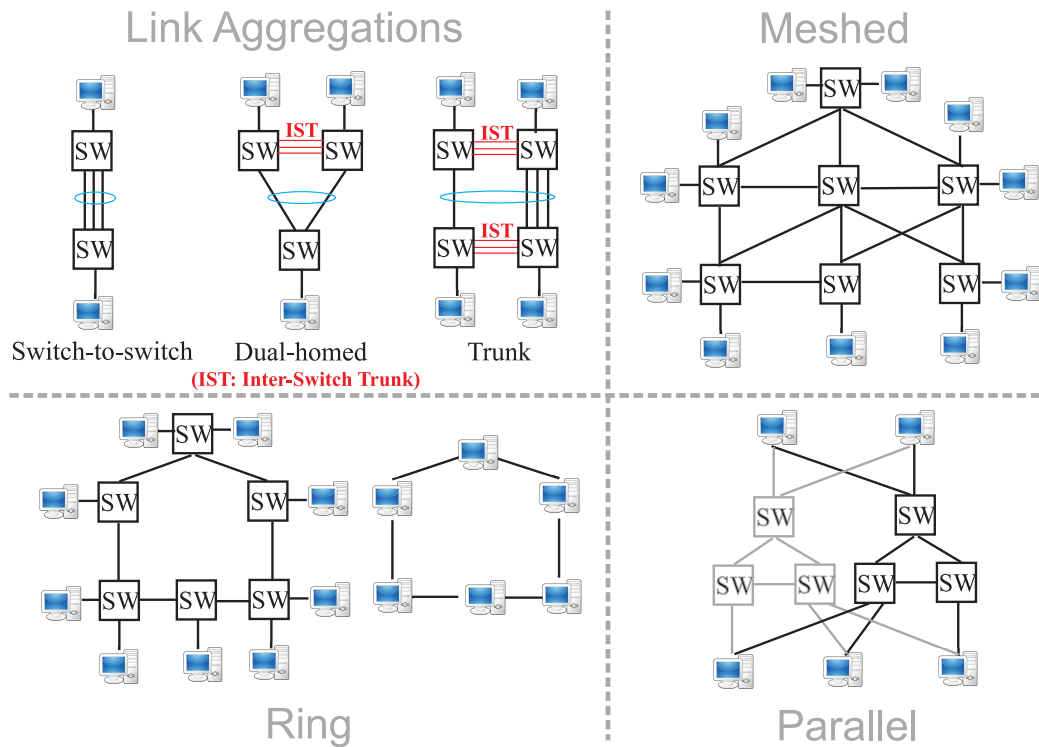


Figure 3.1: Different types of redundant network topologies.

### Link aggregations

Link aggregation is applied to critical parts of networks and provides load-balancing, increased throughput and redundancy. Switches are connected using two or more parallel links which are called *link aggregation* and function as one logic channel. The initial idea of *switch-to-switch* aggregation was extended to complement redundancy of cables with redundancy of switches in *multi-switch* aggregations (*dual-homed*, *trunks*), as depicted in Figure 3.1. Aggregated switches require synchronisation of configuration and a backup data-link which is provided by the Inter-Switch Trunk (IST). In any type of aggregation, the frames are alternatively sent down each of the links and recombined at the other end. If one of the links breaks, the data is sent over the remaining connection(s).

### Mesh topology

In principle, any network topology that provides some level of redundancy can be called a mesh. Ring topology and multi-switch link aggregations can also be considered mesh topologies but they are treated separately in this thesis due to their special properties. Mesh networks provide a flexible and scalable solution at the cost of complexity in the protocols which are required to manage their configuration.

**Ring topology**

Ring topology is a simple and cost-effective redundancy arrangement where any network device has only two neighbours to relate to. No distributed algorithm is needed to calculate the active and backup paths, which are known by the nature of the ring. Switches can be easily interconnected in a ring, and nodes singly-attached to them, as depicted in Figure 3.1. However, higher reliability and better performance are achieved by integrating switching capabilities in the end nodes. Such devices have two Ethernet interfaces that connect them directly to their neighbours, eliminating the single point of failure in the node-to-switch link.

**Parallel redundancy**

Connecting nodes through disjoint parallel LANs provides seamless redundancy with higher reliability and flexibility than ring topology, and at a higher cost. A doubly-attached node sends replicated data through both LANs. In normal conditions, the receiving node discards duplicates; there is no distinction between working and backup paths. In case of failure in one of the LANs, undisturbed communication between nodes is maintained through the second LAN. Both networks are usually similar, in topology and equipment, to guarantee comparable characteristics (latency, synchronisation) between alternate paths – a feature which cannot be accomplished using ring topology. Notably, the parallel LANs are "not aware" of the redundancy. Therefore no special network equipment (i.e. switches) is required. All the workload related to handling redundancy is done in the doubly-attached nodes. The parallel LANs are required to be disjoint, recommended to be powered independently, and not co-located for the highest reliability. Each of the LANs can implement any of the redundancy schemes already described to further increase reliability.

## 3.2 Data Transfer in Redundant Networks

This section reviews methods to manage network redundancy for different types of topologies. Special attention is paid to the speed of reconfiguration (failover), capabilities to provide seamless redundancy and scalability restrictions. There are currently two approaches to manage redundant networks: (1) protocols running distributed algorithms, and (2) software-defined networking. Both are described in the following subsections.

### 3.2.1 Distributed Protocols

Distributed protocols are classified according to the network arrangement they are dedicated or optimised to work with.

#### Link aggregations

The Link Aggregation Control Protocol (LACP, IEEE 802.1AX), initially supporting only *switch-to-switch* aggregation, is now extended (IEEE 802.1AXbq) to allow other arrangements. Functionalities similar to LACP are provided by a number of proprietary protocols, such as Split Multi-Link Trunking (SMLT), Multi-Chassis Link Aggregation (MLAG), or EtherChannel. The protocols enable detection of *switch-to-switch* aggregations, require hand-configuration of multi-switch aggregations, and provide automatic failover between redundant links in any of these arrangements. The failover time is in the order of a second for LACP [31] and sub-second for SMLT [32] (typically  $< 100ms$ ). Importantly, *multi-link* aggregations can be configured in a group of a limited number of switches which is seen as a single logic channel by the rest of the network.

#### Mesh topology

Two approaches to manage mesh topologies, depicted in Figure 3.2, can be distinguished: (1) configuration of a single logic spanning tree that connects all the nodes, and (2) configuration of shortest paths between nodes connected to different switches.

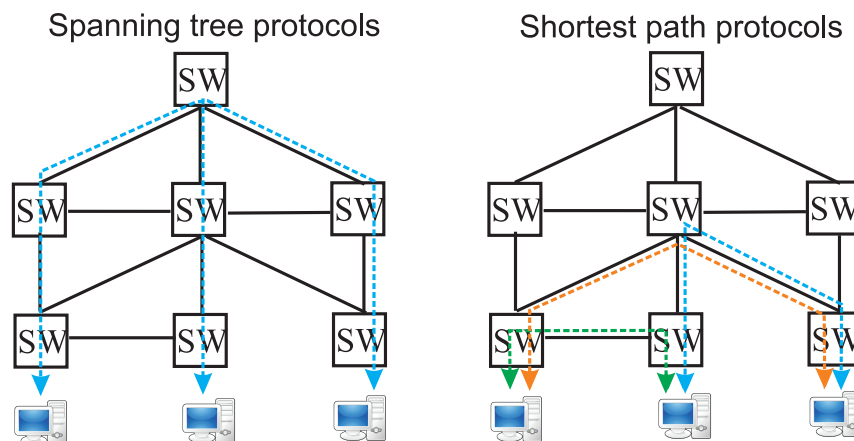


Figure 3.2: Path configuration in spanning tree protocols and shortest path protocols.

The former approach is taken by distance vector protocols, namely the Spanning Tree Protocol (STP) [16] and its successors: the Rapid Spanning Tree Protocol (RSTP) and the Multiple Spanning Tree Protocol (MSTP). These protocols use a distributed computation of a logic tree based on path costs and priorities. Each switch runs the algorithm independently based on the information about adjacent switches. Redundant ports are blocked to prevent loops and activated only in case of failure in the active tree. Such an approach limits the amount of the data exchanged by the protocol between switches but special care is required to avoid loops during topology change and spanning tree recalculation. This is done by temporarily blocking ports when recalculating the topology. When the topology changes, the time to calculate a new tree scales with the size of the longest loop-free path. The reconfiguration of the STP typically takes tens of seconds to minutes [33], while the RSTP and MSTP are up to 50 times faster [34] and take a few seconds [35]. The spanning tree approach forces the traffic to go through a single switch: the root of the logic tree. This results in high loads (bottle-necks) at the root and high latencies between end-nodes located at the leaves of the tree, even if shorter physical paths exist. These characteristics are particularly bad for networks with high horizontal traffic (e.g. Data Centres), and better suited for vertical communication between many nodes and a central server.

An alternative approach is taken by the link-state protocols that calculate shortest paths individually for each pair of switches that connect to nodes: Shortest Path Bridging (SPB) [36] defined by the Institute of Electrical and Electronics Engineers (IEEE) and Transparent Interconnection of Lots of Links (TRILL) [37] standardized by the Internet Engineering Task Force (IETF). These protocols were created to address shortcomings of the spanning tree protocols and remain compatible with them. SPB & TRILL are similar in operation to Layer 3 routing and introduce additional encapsulation of frames routed through the network. SPB comes in two flavours: Shortest Path Bridging VID Mode (SPB-VID) which uses an additional 802.1Q-tag for routing based on the VLAN ID (VID), and Shortest Path Bridging MAC Mode (SPB-MAC) which adds a new Ethernet header for routing based on the Media Access Control (MAC) address. TRILL adds a TRILL-specific header and a new Ethernet header to the original Ethernet frame. Both, SPB and TRILL, are controlled using the Intermediate System to Intermediate System (IS-IS) protocol which propagates the link-state information of all the switches across the entire network. Having a unified view of the entire network, each switch can calculate the shortest path and its alternatives. The failover time depends on the IS-IS-based update of link states triggered by a link or switch failure. SPB augments IS-IS to speed up its convergence and achieve sub-50ms failover in large networks [38].

### **Ring topology**

Depending on the application requirements, one of the two approaches is used to manage ring topologies.

In the first approach a dedicated switch (ring manager) breaks the chain by blocking one of its ports. The connection is de-blocked when failure in the active chain is detected. The protocol

to support this arrangement is called Media Redundancy Protocol (MRP) [39]. Depending on the settings, MRP is specified [39] to provide a worst-case failover time of 500, 200, or 30 ms in rings of up to 50 switches, and 10 ms in rings of up to 14 switches.

Seamless redundancy is provided by an alternative approach standardized in IEC 62439-3 as High-availability Seamless Redundancy (HSR) [40, 41]. The data in this arrangement is sent in both directions all the time and forwarded by all nodes except the destination device. The price to pay for a zero-failover time is the dedicated equipment (integrated switch and node) and half the bandwidth.

### **Parallel redundancy**

An Ethernet-based industrial standard that supports parallel topologies is called Parallel Redundancy Protocol (PRP) and is defined in IEC 62439-3 [41]. It provides an intermediate layer in the node's communication stack, called Link Redundancy Entity (LRE). LRE hides the redundancy of the Ethernet interfaces (ports) from the upper layers. Therefore an application communicates as if through a single LAN. But, in fact, data is sent redundantly through parallel networks and zero-failover time is ensured in case of failure in one of these networks.

## **3.2.2 Software-Defined Networking**

Software-defined networking (SDN) separates control plane from data plane in switches and routers. Data paths between nodes are controlled by a central management node (controller) which has a global view of the entire network. This controller configures the forwarding tables of all the switches. OpenFlow [42] is the most popular protocol to exchange information between switches and the controller. Alternative protocols include FlexForward [43], as well as the work by the Interface to the Routing System (I2RS) working group [44], and the Forwarding and Control Element Separation (ForCES) working group [45] in the IETF.

There are a number of publicly available controllers [46, 47, 48] which can be used with OpenFlow to configure networks. These controllers implement different algorithms to define network topology and provide its failover reconfiguration.

Controller-initiated reconfiguration of a redundant SDN networks requires approximately 100ms [49, 50]. However, publications [49, 51] show that convergence can be optimised to few milliseconds by pre-configuring backup paths and first handling the failure locally on the switches, and later reconfiguring the network to obtain the most optimal topology.

### 3.3 Reliable Data Transmission

This section reviews methods to ensure reliable data transmission through an Ethernet network. The path taken by the data is considered a communication channel as depicted in Figure 3.3. It is



Figure 3.3: The path over an Ethernet network is considered an erasure communication channel.

modelled as a packet erasure channel in which a packet is either received or lost. The reliability of data transmission over such a channel can be increased in three ways: 1) retransmission of erased data, 2) transmission of redundant data over a single communication channel, or 3) transmission of data over multiple communication channels. Each of the methods is briefly described.

**Retransmission** of lost or corrupted data is used in point-to-point communication and it is provided by such protocol as the Transmission Control Protocol (TCP) [52]. It is however not suitable for broadcast traffic (one-to-many communication) which can trigger multiple retransmission requests to a single sender. In such case, the sender or the network can be overwhelmed resulting in further communication problems. Since retransmission protocols rely on timeouts, the maximum latency of data transmission using this method is difficult to predict, thus non-deterministic.

**Redundancy of data** is typically used on communication channels where broadcast transmission and/or predictable latency are required. The data is either replicated or encoded using Forward Error Correction (FEC) scheme. FEC encoding/decoding algorithms include Reed-Solomon, Hamming and Fountain codes. FEC is used in the Real-Time Transport Protocol (RTP) [53] and Reliable Multicast [54].

**Redundancy of channels** requires disjoint paths through the network. Such redundant communication channels are provided by HSR rings and PRP parallel topologies. An ongoing standardisation within the IEEE 802.1CB Task Group [55] aims at providing *seamless redundancy* (zero-failover) in mesh networks by replicating frames for redundant transmission and eliminating the duplicates at reception and in the intermediate switches. This solution is meant to be used in Time-Sensitive Networking (TSN) [56] for applications in Industrial and Energy Automation.

Redundancy of channels and data (FEC) can be combined to further increase the reliability of communication.

### 3.4 Determinism and Latency

Best-effort by design, Ethernet-based LANs are being adapted to applications that require deterministic and low-latency data transfers. A number of such adaptations have already been standardized and a variety of new ideas are being developed.

The basic Ethernet frame header can be extended with a **VLAN tag (IEEE 802.1Q [57])** which allows to organise traffic into Virtual LANs (VLANs) and assign priority to each Ethernet frame. This mechanism is called Class of Service (CoS) and allows prioritisation of Ethernet traffic. Critical traffic is sent with higher priority than other frames, thus increasing the probability of its fast delivery. Along with a careful network design, CoS can provide a certain level of determinism in Ethernet networks.

Quality of Service (QoS) enhancements to Ethernet-based networks guarantee networking performance (e.g. latency, bandwidth) for time-, mission- and safety-critical systems as well as audio-video applications. Solutions such as TTEthernet (SAE AS68902), Ethernet POWERLINK or PROFINET use time-division multiplexing of the traffic and statically allocate to each node a dedicated timeslot for transmission within a so-called cycle. Ethernet POWERLINK allows a shortest cycle of  $200\mu\text{s}$  for a maximum number of 240 nodes. TTEthernet and PROFINET (Isochronous Real-Time version) provide  $500\mu\text{s}$  and  $1\text{ms}$  cycles respectively. Since the traffic in such networks is ordered and predictable, the latency is deterministic and depends only on the network size, which can be a limitation if a large network is required.

The Ethernet-based solution for audio-video applications is required to guarantee upper-bound latency and bandwidth for dynamic streams of entertainment data. The Audio-Video-Bridging (AVB, IEEE 802.1AS [58]) standard implements the Stream Reservation Protocol (SRP) that queries all the switches between the communicating nodes in order to allocate the required bandwidth. Both nodes and switches cooperate to optimise the latency: nodes send traffic evenly spaced, and switches use credit-based shaping of their output queues to evenly distribute latency among streams. This is meant to enable transmitting and playing audio within a maximum latency of  $2\text{ms}$  over 7 hops [59] (at best, a latency of  $125\mu\text{s}$  per switch can be guaranteed [60]).

The successor of AVB networks is being developed by the TSN Task Group within IEEE [56]. The group started its work recently and prepares a number of standards to provide synchronized low latency streaming services through Ethernet networks for industrial and automotive applications. The requirements in these industries are very stringent and call for latencies per switch in the order of  $10\mu\text{s}$  with cycles shorter than  $100\mu\text{s}$  [61][62]. One of the developed mechanisms proposes Ethernet frame pre-emption [63] of best-effort traffic to minimise the latency of critical data and save bandwidth. A complementary solution provides Time-Aware Shaping [63] of Ethernet traffic; it enables nodes to send critical data at predefined timeslots while switches block non-critical traffic during this time to minimise the latency of the critical transmission. Such time-aware switches need accurate synchronisation, which is provided by

the IEEE 802.1AS profile of the Precision Time Protocol (PTP) standard.

The importance of low and deterministic latency in Data Centres is also growing. One of the attempted approaches in such networks is based on Software-Defined Networking, described in subsection 3.2.2. For example, Fastpass [64] developed by Massachusetts Institute of Technology (MIT) and Facebook includes algorithms to centrally administrate the scheduled transmission and paths of each data flow. This enables to minimise queuing and avoid congestion. The result is a substantially (15.5x) reduced median latency and its distribution tail, compared to traditional networks, at the price of slightly worse throughput and minimal latency.



## 3.5 Reliable Time and Frequency Transfer

This section reviews the application of PTP and Layer 1 (L1) syntonisation for time and frequency transfer in redundant networks. A number of PTP profile extensions to support network redundancy are introduced.

### 3.5.1 L1 Syntonisation in Redundant Networks

Synchronous Ethernet (SyncE) [65] is used by telecommunication operators to transfer frequency over the physical L1 in Ethernet networks. SyncE supports redundancy of topology but requires careful hand-configuration; no automatic loop prevention is provided. The short-term phase transient allowed by SyncE during network reconfiguration due to failure is specified to be 120 ns over 16 ms and 1  $\mu$ s over 15 s [66]. SyncE does not restrict the type of network topology but limits the number of hops from the frequency reference source. The performance specifications of SyncE are burdened with legacy requirements and do not reflect the state of the current technology.

### 3.5.2 Precision Time Protocol in Redundant Networks

The Precision Time Protocol (PTP) is designed to support redundant networks. The default profiles of PTP support mesh and ring topologies. Notably, synchronisation performance during network reconfiguration can be substantially different for these topologies and profiles.

In the case of default PTP profiles, deterioration of synchronisation during reconfiguration is expected. The *Delay Request-Response Default PTP profile* uses the Best Master Clock Algorithm (BMCA) to create a synchronisation spanning tree which is independent from the logic topology of the data transfer. The BMCA is similar in operation and failover time to STP (section 3.2), i.e. it can take many seconds to reconfigure. The *Peer-to-Peer Default PTP profile* relies on the data redundancy protocol (section 3.2) to determine the spanning tree for PTP messages. Therefore, its convergence time depends directly on the failover of the used redundancy protocol (such as RSTP or SPB).

A number of application-specific PTP profiles provide dedicated support for network redundancy. The *PTP telecom profile for frequency synchronisation* (ITU-T G.8265.1 [67]) supports only syntonisation. It extends the default behaviour to enable end-nodes (Ordinary Clocks) to observe many Grandmasters simultaneously. Consequently, the switchover between redundant sources takes only a few seconds. The phase jump during network re-arrangements is guaranteed to stay below 1  $\mu$ s. There are no restrictions on network topology type. The *PTP telecom profile for phase/time synchronisation with full timing support from the network* (ITU-T G.8275.1 [68]) requires SyncE syntonisation along with PTP synchronisation. The accuracy of phase synchronisation with respect to a reference source is guaranteed to be below 1.5  $\mu$ s, also

during network reconfiguration. There is no restriction on network topology type but the number of Boundary Clocks (BCs), which are switches or routers, between the node and the source is limited to 10.

A PTP Profile to enable seamless synchronisation over HSR and PRP networks is currently being standardized in IEC 62439-3 [69]. It is based on the *Peer-to-Peer Transparent Clocks (TCs)* and should enable to maintain  $1\mu s$  accuracy during network reconfiguration due to failure.

## 3.6 Summary

The review of the relevant networking solutions presented in this chapter is summarized in Table 3.1. The table shows that there is no existing solution, known to the author, that can fulfil all the CERN requirements (see Table 1.1.4). The solution that is the closest to match these requirements is being prepared by the TSN group. It is an ongoing effort that started during the work on this thesis. Once completed, it might potentially meet CERN's requirements regarding reliability and determinism of data distribution but not the synchronisation performance.

The information about the existing solutions, their limitations and performance, is the input to the next chapter which develops a strategy to meet CERN requirements by enhancing the most suitable existing solution.

Abrev.	Name	Network topology	Network size	Time of reconfig	Latency	Accuracy in transient
LACP	Link Aggregation Protocol	aggregations, dual-homed		1s		
SMLP	Split Multi-Link Trunking	aggregations, dual-homed		< 100ms		
STP	Spanning Tree Protocol	mesh		tens seconds		
RSTP, MSTP	Rapid/Multiple Spanning Tree Protocol	mesh		few seconds		
SPB	Shortest Path Bridging	mesh		sub-50ms		
TRILL	Transparent Interconnection of Lots of Links	mesh		> 50ms		
MRP	Media Redundancy Protocol	ring	50 switches 14 switches	30ms 10ms		
HSR	High-availability Seamless Redundancy	ring	16 nodes <sup>2</sup>	zero-failover	depends on pos. in ring	1 $\mu$
PRP	Parallel Redundancy Protocol	parallel, mesh, ring	16 switches <sup>2</sup>	zero-failover		1 $\mu$
SDN	Software-Defined Networking: controller-initiated	mesh, ring		100ms		
	Software-Defined Networking: pre-configured	mesh, ring		a few ms		
TTEthernet	Time-Triggered Ethernet	mesh, ring			500 $\mu$ s	
POWRLINK	Ethernet POWERLINK	ring	240 nodes		200 $\mu$ s	sub- $\mu$ s
PROFINET	Isochronous Real-Time version of PROFINET	ring	240 nodes		1ms	
AVB	Audio-Video-Bridging	mesh	7 hops		2ms (125 $\mu$ s per hop)	< 1 $\mu$ s
TSN	Time Sensitive Networking (ongoing work)	mesh	7 hops	zero-failover	100 $\mu$ s (10 $\mu$ s per hop)	< 1 $\mu$ s
SyncE	Synchronous Ethernet (synchronisation only)	mesh, ring	20 hops <sup>1</sup>	milliseconds to seconds		160ns (reconf < 16 $\mu$ s) 1 $\mu$ s (reconf < 15s)
PTP	Default profile of Precision Time Protocol	mesh, ring		tens seconds		implement. dependent
G.8265.1	PTP telecom profile for freq (synchronisation only)	mesh, ring	20 hops	a few seconds		< 1 $\mu$
G.8275.1	PTP telecom profile for phase/time	mesh, ring	10 hops	a few seconds		< 1.5 $\mu$
WR	CERN requirements for WR	not defined	2000 nodes	zero-failover	1ms (10 $\mu$ per hop)	sub-ns

Table 3.1: Comparison of existing networking solutions and the CERN requirements for WR.

<sup>1</sup> A hop being a synchronous Ethernet equipment slave clock (EEC).

<sup>2</sup> This restriction is made for the intended < 1 $\mu$  accuracy of synchronisation.

## Chapter 4

---

# Strategy to Increase Reliability and Ensure Determinism of a WR Network

---

This chapter analyses the CERN requirements (1.1.4) to conclude with a strategy for increasing reliability and ensuring determinism in a WR network. The chapter translates the requirements into precise networking terms, explains mathematical tools, and uses these tools to analyse different approaches for increasing reliability and ensuring determinism. The strategy proposed in this chapter is the basis for enhancements developed further in this thesis.

The strategy is developed in a number of steps described in separate sections. First, in section 4.1, **basic assumptions** are described. They define precise boundaries of the WR network considered in this thesis. They also specify the meaning of successful operation and failure of the WR network in the context of the CERN control and timing system. Then, in section 4.2, the **basis for reliability calculations** are specified. This includes basic mathematical tools and numerical values to represent the reliability of network components. Consequently, the reliability of the entire network can be evaluated unambiguously. In order to know whether the achieved reliability is sufficient, the **target reliability of the WR network** is needed. This value is derived from the initial CERN requirements in section 4.3. There is a number of **factors contributing to the overall reliability of a WR network**. These factors and their contribution are specified in mathematical form in section 4.4. Each factor and its mitigation techniques are then analysed:

- **Network failure** and its mitigation through redundancy are evaluated in 4.5.
- **Message loss** and its mitigation through Forward Error Correction (FEC) are analysed in 4.6
- **Traffic congestion** and its mitigation through proper configuration are described in 4.7.
- **Insufficient performance** and its mitigation through known and deterministic characteristics of the WR switch are analysed in 4.8.

Once the factors contributing to the overall reliability are analysed, quantified and their interactions understood, the final strategy to increase reliability and ensure determinism of the WR network is proposed in section 4.9. The proposed strategy of enhancing networking solutions and designing the WR network constitutes a guideline for the methods presented in Chapter 5 and Chapter 6, as well as the reference WR-based control and timing network in Chapter 7.

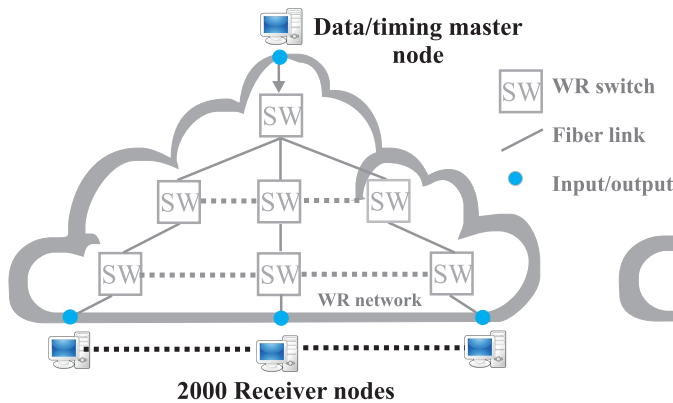
## 4.1 Basic Assumptions for WR Network Operation

This section outlines the boundaries of the WR network analysed in this thesis and specifies the meaning of its successful operation in the context of CERN control and timing system. It also lists the considered reasons for the WR network to fail. The basic assumptions presented in this section are used throughout the thesis.

The WR network is considered a multi-input-multi-output system that consists of WR switches and fibre links. The links use Gigabit Ethernet and include both, the interconnections among the switches and the connections between the switches and the nodes. The nodes are not a part of the network. Thus, the fibre links that connect switches with nodes are considered the inputs/outputs of the WR network, as depicted in Figure 4.1. The WR network is connected to two types of nodes:

- **data/timing master** - a node that is the source of control messages and/or timing
- **receiver** - a node that receives control messages and timing. It is embedded into or connected to the devices in the accelerators.

a) Non-redundant WR network



b) N-redundant WR network

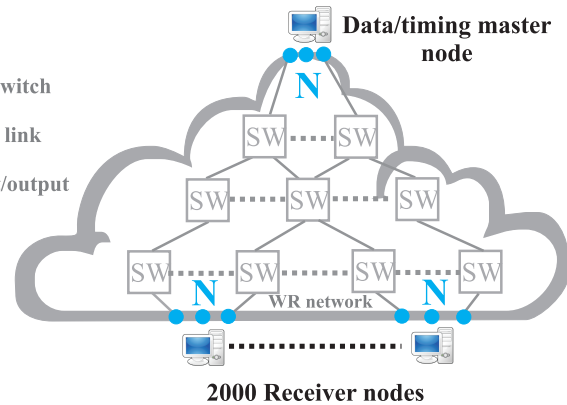


Figure 4.1: Boundary of a White Rabbit network considered in reliability calculations.

In order to increase reliability of the non-redundant WR network presented in Figure 4.1-a, network redundancy is introduced. Figure 4.1-b presents N-redundant WR network. For the sake of simplicity and effectiveness of redundancy, in further consideration, N-redundant WR network obeys the following rules:

- each node is connected through N links, each of these links is connected to a separate switch
- each of the N links that is connected to a node, receives from this node exactly the same information at exactly the same instant.

Having set the boundaries of WR network consideration, the meaning of WR network successful operation in the context of CERN control and timing network can be defined. In such

a WR network, in accordance with the requirements (1.1.4) and principles of operation (1.1), the data master sends control messages every 1ms. These messages must be delivered to all the receiver nodes with upper-bound latency that allows execution of events in the next millisecond. Out of all the messages sent to all the nodes over 1 year of operation, only a single message can fail to be delivered to one or more nodes. During this year, the synchronisation between the timing master and all the 2000 receiver nodes is required to be always within the specified accuracy and precision (i.e. sub-ns & sub-50 ps). Thus, **a WR network is considered functional if and only if it successfully delivers control messages and timing (synchronisation) from the data/timing master to all of the 2000 nodes.** If the WR network is N-redundant, a node must receive control messages and timing from at least one of its N links.

Network redundancy mitigates one of the reasons that causes unreliability: a single point of failures. Other factors that can render WR network non-functional exist, need to be identified, understood and mitigated. In this thesis, the WR network is considered non-functional (fails) if any of the following occurs during one year of its operation:

- Two or more distinct control messages fail to reach any of the nodes through all  $N$  links connected to this node. A control message fails to reach a node when:
  - it arrives at the node too late (latency exceeds upper-bound), or
  - it arrives at the node corrupted, or
  - it does not arrive at the node.
- Synchronisation between the timing master and any of the nodes exceeds specification on all  $N$  link connected to this node.

The potential reasons for the above failures are listed in Table 4.1. The table indicates which of the two services, delivery of control messages or timing, is affected by this failure. Each of these failure factors is considered in the following sections and the strategy, proposed in the last section, addresses all of them.

$N^o$	Failure factors	Affects	
		Control message	Timing (synchronisation)
1	Temporary traffic congestion	yes	no
2	Performance degradation due to number of hops (switches from data/timing master)	yes	yes
3	Network element failure (switch, fibre)	yes	yes
4	Bit Error Rate (BER)	yes	no
5	Network re-arrangement (failure or adding new element)	yes	yes

Table 4.1: Factors that cause failure and their impact on the delivery of control messages and timing.

## 4.2 Basis Assumptions for Reliability Calculations

Having established the boundary of the WR network and the definition of its successful operation, the terminology and formalisms used in the reliability calculations are explained below:

- a) The reliability is quantified exchangeably using:
  - Mean Time Between Failures (MTBF) - the length of time after which, statistically, half of all equipment of a given type is non-functional.
  - $R(t)$  - the probability that the equipment performs its intended functions continuously for a given period of time,  $t$ .
- b) The MTBF is considered equal to the Mean Time To Failure (MTTF) in this thesis.
- c) The values of MTBF used in the calculations are presented in Table 4.2 and their sources explained below:
  - **MTBF of WR switch** has not been obtained yet. Representative values for different classes of switches available in the market are used in the calculations [70][71][72][73], i.e. MTBF of 40 000, 100 000, and 650 000 hours.
  - **MTBF of fibre** is based on the cable cut failure rate reported by [74]. It is translated into MTBF of 3 000 000 hours per fibre <sup>1</sup>.
- d) Network elements are maintained in the "useful life" period of the "bathtub" model [75] which means that their failure rate ( $\lambda$ ) is considered nearly constant and expressed as:

$$\lambda = \frac{1}{MTBF} \quad (4.1)$$

- e) The exponential law [75] is used to calculate reliability:

$$R_{\{fibre|switch|network\}}(t) = e^{-\lambda t} = e^{-\frac{t}{MTBF}} \quad (4.2)$$

- f) For WR network and its elements two values of Mean Time To Repair (MTTR) are assumed:
  - $MTTR_n=4h$  if a failure causes unavailability of the entire WR network, or
  - $MTTR_e=12h$  if a redundant element fails and it does not cause network unavailability.
- g) The probability laws are used to calculate reliability of the entire network,  $R_{network}(t)$ , based on the reliability of individual elements,  $R_{\{fibre|switch\}}(t)$ , and their interconnections:
  - for non-redundant network, the reliability of elements is calculated over  $t = 1 \text{ year}$
  - for redundant networks, no single point of failure is assumed and the reliability is calculated over  $t = MTTR_e$ , assuming that any broken redundant element is replaced within 12h, which essentially resets the calculations.

---

<sup>1</sup>The failure rate reported in [74] is 4.39 per year per 1000 sheath miles for long-haul fibres. It translates into approximately 1 cut every 3 200 000 hours per kilometer of fibre. In this thesis, an MTBF of 3 000 000 hours per fibre, regardless of its length, is assumed. It gives approximately 6 cuts per year in a non-redundant WR network connecting 2000 nodes and consisting of 2136 fibres. MTBF of 3 000 000 hours per fibre assumes that, on an average, a fibre link in WR network has a length of 1km. In reality, the average fibre length will be much smaller. However, the short fibres in CERN installations are more exposed to the human error, increasing probability of failure. The appropriateness of the MTBF value used in this thesis for fibre is confirmed by similar value used in [18].

- h) A year has 8766 hours on average, including leap years (i.e.  $365.25 \cdot 24$  [days x hours]).
- i) Availability,  $A$ , is the ability of a system to be in a state to perform a required function at a given instant of time [76]. It is calculated using the  $MTTR_n = 4h$  as follows:

$$A = \frac{MTTF}{MTTF + MTTR_n} = \frac{MTBF}{MTBF + MTTR_n} \quad (4.3)$$

Network element		MTBF [hours]	MTBF [years]	R(t=1 year) [probability]	R(t=12 hour) [probability]
Fibre link		3 000 000	342.2	0.997082265	0.999996000
switch	low reliability	40 000	4.6	0.803201229	0.999700045
	medium reliability	100 000	11.4	0.916072288	0.999880007
	high reliability	650 000	74.2	0.986604377	0.999981539

Table 4.2: Representative values of switch and fibre Mean Time Between Failures used in the analysis.



### 4.3 Target Reliability of the White Rabbit Network

The target reliability value of the WR network, as defined mathematically in the previous section, is not specified in the CERN initial requirements (1.1.4). Such a value is typically established when designing systems which require high reliability. It allows to use the mathematical tools presented in the previous section to evaluate the appropriateness and effectiveness of the applied reliability mechanisms. Thus, based on the existing requirements and commonly-used practices, the target value for the WR network is set in this section.

Commonly, the target reliability value of a system depends on the consequences of its failure. The most stringent and well-documented reliability requirements are provided for safety-critical systems. Failure of such a system has catastrophic consequences which usually means casualties. For example, airplanes, Global Positioning System (GPS) or International Space Station (ISS) are safety-critical systems and require extremely high reliability to operate without failure for many years. On the other hand, systems that can safely fail but their prolonged downtime is expensive or catastrophic require high availability. Availability depends not only on the probability of the system's failure, i.e. its reliability, but also on the time it takes for the system to be repaired, MTTR. During one year of operation, a highly reliable system that takes few days to repair can have similar availability to a less reliable system that is always repaired in one hour.

Although WR network is essential to accelerator's operation, it cannot be considered a safety-critical system. The accelerators have independent fail-safe protection that is used in the case of WR network's failure, e.g. the Beam Interlock System [77]. Therefore, WR network is considered in this thesis a high-availability rather than safety-critical system. Since it is essential to accelerator's operation, the WR network is expected to be serviced within hours in case of its failure. A widely-held but difficult-to-achieve standard of availability for such a system is known as "five 9s" and require availability of  $A = 99.999\%$ . This is indeed a reasonable target for the WR network.

Knowing the target availability and assuming MTTR of the WR network to be  $MTTR_n = 4h$ , the target reliability value can be calculated. First, Equation 4.3 is used to obtain the MTBF of the WR network:  $MTBF = 399\,996\,h \approx 45.6\,years$ . This MTBF is translated into reliability over 1 year of operation using Equation 4.2:  $R(1\,year) = 0.978948$ . A WR network with such parameters will work without interruptions for 10 years with 80% chances.

In order to verify that the obtained reliability value is a reasonable target, it is compared with reliability values defined for safety-critical systems. The reliability values defined by the US Department of Defense in MIL-STD-882E [78] standard, by the International Electrotechnical Commission (IEC) in the IEC 61508 standard [79] standard, as well as the National Aeronautics and Space Administration (NASA) and the Federal Aviation Administration (FAA) are provided in Table 4.3. For comparison, the table includes the reliability values assumed for switches and fibres (grey font).

Reliability R(1 year)	MTBF [years]	Reference
0.999999	992 471	System whose failure is improbable according to MIL-STD-822E
0.999990	100 000	System whose failure is improbable according to IEC 61508
0.999900	9 993	System fulfilling NASA's safety goal of a less than 1 in 10,000 chance of casualty
0.999001	1000	System whose failure is remotely probable according IEC 61508
0.999000	992	System whose failure is remotely probable according MIL-STD-822E
0.997264	365	Fibre optic reliability assumed in this thesis
0.990005	100	System whose failure is occasional according to IEC 61508
0.990000	$\approx 100$	System with "five 9s" availability and MTTR=8h – ideal reliability target for WR
0.989950	99	System whose failure is occasional according to MIL-STD-822E
0.986604	74	Ethernet switch that is considered very reliable (MTBF=650 000h)
0.980000	$\approx 50$	System with "five 9s" availability and MTTR=4h – min. reliability target for WR
0.916072	11	Ethernet switch that is considered medium reliable (MTBF=100 000h)
0.904837	10	System whose failure is probable according to IEC 61508
0.900000	9	System whose failure is probable according to MIL-STD-822E
0.803201	5	Ethernet switch that provides low reliability (MTBF=40 000h)

Table 4.3: Reliability values of safety-critical systems (black) and network elements (grey).

Based on the reliability calculated from "five 9s" availability and the reliability values for safety-critical systems, two target reliability values are proposed:

1. **Minimum target reliability** which is represented by the following values:

- $R(1 \text{ year}) = 0.98$
- $A(1 \text{ year}) = 99.999\%$
- $MTBF \approx 433\,900 \text{ h} = 50 \text{ years}$
- $MTTR = 4h$ .

2. **Ideal target reliability** which is represented by the following values:

- $R(1 \text{ year}) = 0.99$
- $A(1 \text{ year}) = 99.999\%$
- $MTBF \approx 876\,600 \text{ h} = 100 \text{ years}$
- $MTTR = 8h$ .

These values represent the intended WR network reliability over 1 year of operation in which the network delivers control messages and timing from a data/timing master node to 2000 receiver nodes with specified characteristics. This reliability is affected by a number of factors. The mathematical dependency of the WR network reliability on these factors is described in the next section.

## 4.4 Factors Contributing to the Reliability of a WR Network

This section explains how, mathematically, different factors that disturb operation of WR network contribute to the overall reliability of the WR network. These factors have been identified in section 4.1 as: a) traffic congestion, b) performance, c) network element failure, d) BER, e) network re-arrangement.

Each of the considered factors can be represented by a reliability value which contributes to the overall WR network reliability. The reliability,  $R(t)$ , of a WR network represents the probability that the network performs its intended functions continuously for a given period of time,  $t$ . During this period, each of the factors might cause network failure with probability  $uR_{factor}(t)$ . The value of  $uR_{factor}(t)$  can be considered unreliability of a functionality that ceases due to a particular factor. Thus, reliability of this functionality is expressed as  $R_{factor}(y) = 1 - uR_{factor}(t)$  and can be defined for the mentioned factor as follows:

$R_c(t)$  : probability of congestion-less communication

$R_l(t)$  : probability of meeting performance specifications (latency and synchronisation)

$R_n(t)$  : probability of network connectivity between data/timing master and all the nodes

$R_t(t)$  : probability of loss-less transmission of control messages (both, BER or rearrangement).

Any of these factors can cause unreliability of the WR network and thus they are considered serially in the reliability calculations:

$$R(t) = R_c(t) \cdot R_l(t) \cdot R_n(t) \cdot R_t(t) \quad (4.4)$$

While in the above high-level analysis the factors are considered independent, identified dependencies will be included in more detailed calculations in the following sections.

The first two factors,  $R_c(t)$  and  $R_l(t)$ , can be fully eliminated. They depend on the design of a switch, the topology of the network, the configured path of control messages, and the traffic engineering. It is possible to devise a network in which the control messages do not experience congestion and their latency is always within an upper bound. Thus the proposed strategy should recommend mechanisms and configuration where:  $\forall t : R_c(t) = R_l(t) = 1$ .

The latter two factors,  $R_n(t)$  and  $R_t(t)$ , cannot be fully eliminated. They are caused by the imperfections of the network elements. It is however possible to mitigate these imperfections introducing redundancy. Such a redundancy can be applied to both, network elements and data. Thus, the proposed strategy should recommend mechanisms to optimally increase  $R_n(t)$  and  $R_t(t)$  while not affecting the other two factors,  $R_c(t)$  and  $R_l(t)$ . Assuming that  $R_c(t) = R_l(t)$ , their target values are calculated for the two WR network target reliability values:

$$\begin{aligned} \text{Minimum: } R_n(1 \text{ year}) = R_c(1 \text{ year}) &= \sqrt{0.98} \approx 0.9899 \text{ and } MTBF_n = 99 \text{ years} \\ \text{Ideal: } R_n(1 \text{ year}) = R_c(1 \text{ year}) &= \sqrt{0.99} \approx 0.9950 \text{ and } MTBF_n = 199 \text{ years} \end{aligned} \quad (4.5)$$

The following sections discuss each factor affecting WR network reliability, starting with the reliability of network connectivity.

## 4.5 Reliability of Network Connectivity

This section analyses the reliability of network connectivity,  $R_n(t)$ , to recommend the most optimal topology for the WR network. The reliability of network connectivity translates into the probability that there exist paths from the data/timing master to all the receiver nodes. Or, in other words, the probability that all the nodes are interconnected by the WR network. This probability can be increased by introducing redundant elements. Different types of redundant topologies are evaluated using the tools introduced in section 4.2 to recommend the most optimal topology. This topology must provide the target reliability value,  $R_n(1 \text{ year})$ , and have minimal impact on the other failure factors presented in 4.4. Chapters 5 and 6 develop mechanism to support the redundant topology recommended in this section.

### 4.5.1 Reliability in Non-Redundant WR Network

Firstly, non-redundant network topologies are considered illustrating a simple case of reliability analysis. Two types of non-redundant topologies are analysed: tree and line. WR networks arranged in these topologies are depicted in Figure 4.2. Any of the networks in the figure is

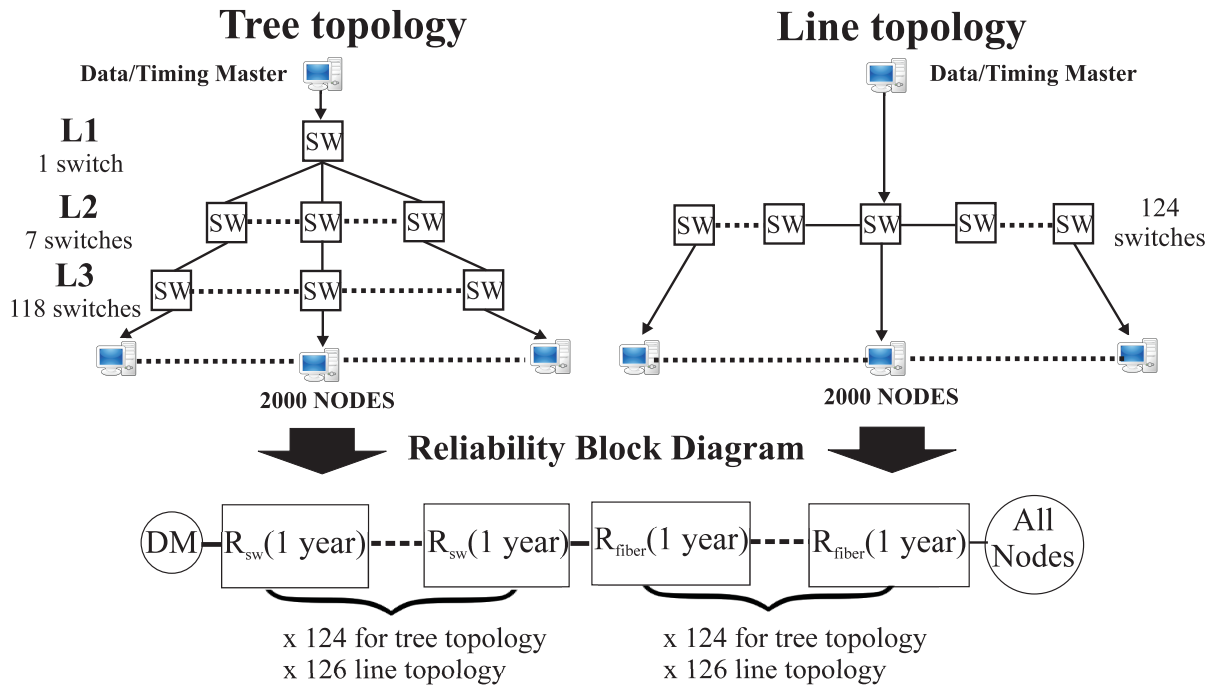


Figure 4.2: Possible non-redundant topologies of a WR network and Reliability Block Diagram.

considered functional if and only if it delivers control messages and timing from the data/timing master to all the 2000 receiver nodes. This requirement translates into one-to-all reliability analysis which are similar for both network topologies. For both topologies, Figure 4.2 depicts a common Reliability Block Diagram (RBD) [80] that is used to analyse their network reliability. The parameters of the RBD are different for each topology. In the RBD, all the network

elements are considered in a serial arrangement since a failure of any of the elements results in a failure of the entire network. The reliability of network connectivity during a year of operation,  $R_n(1 \text{ year})$ , for each network in Figure 4.2 is calculated as follows:

$$R_{n\_tree}(1 \text{ year}) = (R_{switch})^{126} \cdot (R_{fibre})^{125} \cdot (R_{fibre})^{2001} = 0.00000038 \quad (4.6)$$

$$R_{n\_line}(1 \text{ year}) = (R_{switch})^{124} \cdot (R_{fibre})^{123} \cdot (R_{fibre})^{2001} = 0.00000046 \quad (4.7)$$

where  $R_{switch}$  and  $R_{fibre}$  are the reliability of the switch and the fibre during one year of operation:  $R_{switch} = R_{switch}(1 \text{ year}) = 0.916072288$  and  $R_{fibre} = R_{fibre}(1 \text{ year}) = 0.997082265$ , see Table 4.2. For the switch, the medium MTBF value is used, it will be used in all the further calculations. Relevant calculations for all considered switch MTBF values are provided in Appendix C. The values of MTBF and availability for the two networks are provided in Table 4.4. These values indicate that a non-redundant WR network connecting 2000 receiver nodes would surely break throughout a year, on average more than once a month. Therefore redundancy is required.

Network	Topology type	Reliability $R_n(1 \text{ year})$	Availability [%]	MTBF		MTTR	Number of	
				[hours]	[years]	[hours]	switches	fibres
Non-redundant	Tree	0.000000032	99.2187	508	0.058	4	126	2126
	Line	0.000000038	99.2268	513	0.059	4	124	2124

Table 4.4: Results of the reliability calculations for non-redundant topologies.

## 4.5.2 Reliability in Redundant WR Network

Reliability values for different types of redundant networks in Figure 4.3 are calculated and compared to find the most optimal topology. Reliability calculation for a redundant network is distinct from such a calculation for a non-redundant network in two ways. First, multiple alternative paths need to be taken into account, ideally as few elements as possible is connected serially in the RBD analysis. Second, the redundant elements that fail can be replaced without affecting operation of the network. Thus, the reliability value of a single network element, a switch or a fibre, used in the calculation of the entire network is not the reliability calculated over the entire year of operation. Instead, the reliability of a single element is calculated over the time needed for its replacement which is assumed to be MTTR=12h, see 4.2.

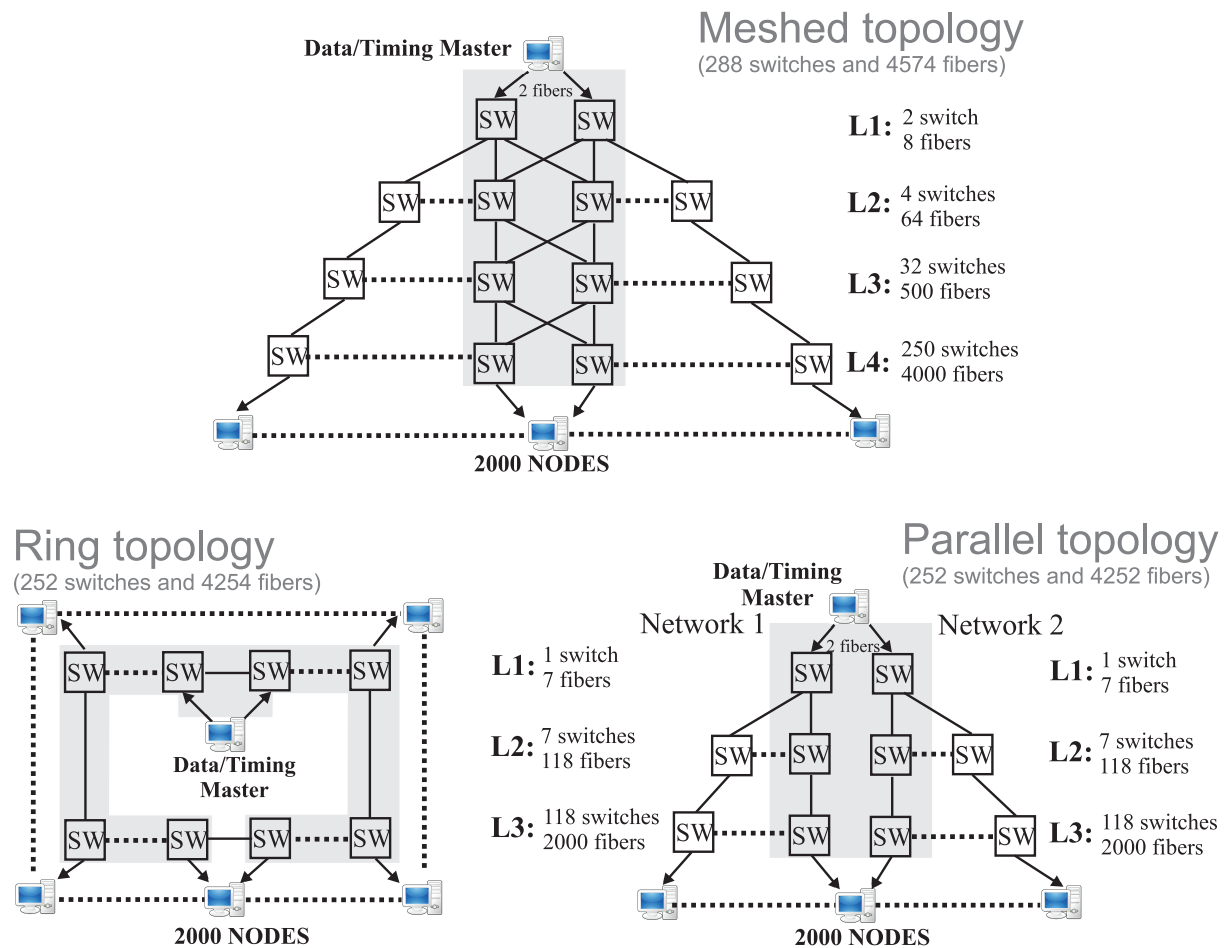


Figure 4.3: Different network topologies that can accommodate 2000 nodes.

The reliability calculation of the entire network is therefore performed in the following steps:

1. The reliability of each element (fibre or switch),  $R_{\{switch|fibre\}}(t)$ , is calculated over 12 hours:

$$R_{\{switch|fibre\}}(12h) = e^{-\frac{12h}{MTBF_{\{switch|fibre\}}}} \quad (4.8)$$

2. The reliability of the entire network,  $R_n(t)$ , is calculated as the probability that any of the redundant paths is available within 12 hours to all the nodes in the network. This calculations depend on the network topology. As an example, detailed calculations for the mesh topology are described in Appendix D.

3. The MTBF of the entire network is calculated:

$$MTBF_{network} = -\frac{12h}{\ln(R_n(12h))} \quad (4.9)$$

4. The network reliability over a year,  $R_n(8766h)$ , is calculated:

$$R_n(1 \text{ year}) = e^{-\frac{8766h}{MTBF_{network}}} \quad (4.10)$$

5. The network availability,  $A_n(4h)$  is calculated assuming  $MTTR_n = 4h$ , see 4.2:

$$A_n = \frac{MTBF_{network}}{MTBF_{network} + MTTR_n} \quad (4.11)$$

The presented steps are used to calculate the values of reliability, MTBF and availability of the redundant networks with three topologies: ring, parallel and mesh, depicted in Figure 4.3. The results are provided in Table 4.5 for the  $MTBF_{switch}$  of 100 000h (the results for all switch MTBF values are provided in Appendix C). They indicate that the most efficient, in terms of reliability, is the mesh topology. It is the only topology that meets the reliability target values set in the previous section. It meets both, the minimum and the ideal targets, i.e.  $R_n(1 \text{ year}) = 0.9899$  and  $R_n(1 \text{ year}) = 0.9950$ .

Network	Topology type	Reliability $R_n(1 \text{ year})$	Availability [%]	MTBF [hours]   [years]		MTTR [hours]	Number of switches   fibres	
Non-redundant	Tree	0.000000032	99.2187	508	0.058	4	126	2126
	Line	0.000000038	99.2268	513	0.059	4	124	2124
Redundant	Ring	0.70507457	99.9841	25085	2.9	4	252	4254
	Parallel	0.93347464	99.9969	127336	14.5	4	252	4254
	Meshed	0.99699778	99.9999	2915452	332.6	4	288	4574
Target values sec. 4.4, Eq. 4.5	Minimum	0.98990000	99.9995	867803	99.0	4	N/A	N/A
	Ideal	0.99500000	99.9995	1744419	199.0	8	N/A	N/A

Table 4.5: Results of the reliability calculations for all the considered network topologies (data in grey is provided from Table 4.4).

The mesh topology is not only better-suited regarding reliability of network connectivity but it has also a number of advantages in the context of other failure factors and its application

as the CERN control and timing network. In particular, the ring topology could not provide the required latency and synchronisation performance for the daisy-chained 252 switches needed to connect 2000 nodes, see section 4.8. On the other hand, the parallel topology pushes all the complexity of redundancy implementation to the nodes. Consequently, unlike in the mesh and ring topologies, when parallel topology is used, a singly-attached node takes no advantage of the expensive network reliability. In practice, such singly-attached nodes will constitute a substantial number in the WR-based CERN control and timing system. Finally, original principles and standards supporting the parallel and ring networks would need to be substantially extended to meet the target reliability. On the other hand, the idea behind the mesh topology has proven to provide sufficient reliability and it is only the supporting standards that need extension. All this arguments further support the recommendation of the mesh topology for the WR network.

Knowing the number of elements in the chosen mesh network and their reliability, it is easy to estimate how many elements are expected to fail causing network reconfiguration in a year. For the considered switch and fibre MTBF values (100 000h and 3 000 000h), statistically, 25 switches and 2 fibre connecting switches<sup>2</sup> fail yearly. This translates into 27 network rearrangements which can cause the loss of control messages and need to be taken into account in the next section.

---

<sup>2</sup>This takes into account only the fibres between switches, it excludes the fibres connecting switches and nodes since the redundancy mechanism implemented by the node is not considered in this thesis. In principle, as long as the control messages and timing is available on one of the N links provided to the node, the network is considered functional.



## 4.6 Reliability of Message Transmission

This section analysis the probability of successfully transmitting the control messages and the Precision Time Protocol (PTP) messages that are used in synchronisation. Since the PTP protocol is essentially immune to occasional loss of messages, the focus is on the probability that a control message is successfully delivered from the data master to all the receiver nodes through a redundant network that provides reliable connectivity without traffic congestion. In such a network, there are two possible reasons for losing control messages: (1) network reconfiguration, and (2) data corruption due to medium imperfection. The number of network reconfigurations is quantified in the previous section to be 27 a year. The data corruption is quantified by the Bit Error Rate (BER) specified in IEEE 802.3 [10] to be  $10^{-12}$  for fibre optic (1000BASE-BX10). The methods to mitigate both reasons of losing control messages are discussed in this section.

### 4.6.1 Expected Message Loss

The expected effect of BER and network reconfiguration on the operation of a WR-based control and timing network without any mitigation techniques is calculated in this subsection.

First, the statistically expected effect of BER is quantified. Assuming that the redundant mesh topology recommended in the previous section is deployed, the expected number of bit errors in this network is calculated and presented in Table 4.6. The calculations take into account

Transmitted data	Data size sent each period [bytes]	Transmission period [seconds]	Data on wire per year (see Note 5) [bits]	Bit errors per year per fibre	Bit errors per year per network (see Note 6)	Network frame loss rate [1 every x time]
Control message	1200	0.001	$3.86 \cdot 10^{13}$	386	1 764 661	18 seconds
	6000	0.001	$1.90 \cdot 10^{15}$	1900	8 696 224	4 seconds
PTP Announce with WR_TLV	78	2	$1.58 \cdot 10^{10}$	0.02	72	121 hours
PTP messages in req-resp mech. (see Note 1)	186 (see Note 3)	1	$8.65 \cdot 10^{10}$	0.09	396	22 hours
PTP messages peer-delay mech. (see Note 2)	368 (see Note 4)	1	$1.65 \cdot 10^{11}$	0.16	754	12 hours
Note 1: 2-step request-response mechanism: Sync, Delay_Req, Delay_Resp, Follow_Up. Note 2: 2-step peer delay mechanism: Sync, two way: Pdelay_Req, Pdelay_Resp, Pdelay_Resp_Follow_Up. Note 3: A sum of payload sizes needed to transport all the PTP messages required, i.e. $78 + 44 + 44 + 54 + 44$ bytes. Note 4: A sum of payload sizes needed to transport all the PTP messages required, i.e. $78 + 44 + 2 \cdot (54 + 54 + 54)$ bytes. Note 5: BER is defined at the PHY service interface (59.1.1 in [10]), each data byte is translated into 10 bits on the wire. Note 6: The mesh redundant network in Figure 4.3 with 4574 fibres.						

Table 4.6: Loss of messages due to Bit Error Rate in a WR-based control and timing network.

the rate at which the messages are sent and assume that control messages are the payloads of Ethernet frames. Losses for two payload sizes are calculated: 1200 and 6000<sup>3</sup> bytes. In the worst but probable case [81], each bit error translates into a lost control or PTP message. On a single link, it is expected that a single PTP message is lost due to BER every few years which confirms that BER is not likely to provoke failures of synchronisation. On the other hand, the number of control messages lost due to BER throughout a year of operation on a single fibre link is in the order of few hundreds. This amounts to a single control messages lost every several seconds in the entire WR network. Clearly, the effect of BER on the control messages needs to be mitigated.

Additionally to BER, network reconfiguration results in the loss of control messages and needs to be mitigated. The statistically expected number of 27 network reconfigurations throughout a year of operation is calculated in the previous section. The number of control messages lost due to each reconfiguration depends on the failover duration. The fastest available reconfiguration mechanism for mesh networks (see section 3.2) provides sub-50ms reconfiguration. This translates into 50 control messages lost during one reconfiguration, assuming they are sent every 1ms. Clearly, the effect of network reconfiguration on the control messages needs to be mitigated.

#### 4.6.2 Mitigation of Control Message Loss

Forward Error Correction (FEC) is proposed as a method to mitigate both reasons for control message loss, network reconfiguration and BER. Using an alternative method, retransmission, in WR network is impossible due to the broadcast character of the control message transmission and the requirement of low latency. Unlike in retransmission where additional data is sent only when data loss occurs, in FEC the redundant data is always sent so that it can be used in the case of loss. The amount of transmitted redundant data needs to be well-tuned to the expected loss so that, statistically, losses can be recovered.

While FEC can be easily tuned to mitigate the BER, FEC alone is not sufficient to recover loss of over 50 consecutive control messages during network reconfiguration. It is not only due to the huge number of lost messages but also because each control message is needed by the accelerators within the subsequent millisecond. Therefore, the following scheme is proposed:

- Transmission of a single control message using a number of Ethernet frames.
- A FEC that can recover a control message from a sub-set of sent Ethernet frames (lost due to BER or network reconfiguration).
- Network reconfiguration (switchover) fast enough to never lose more frames than the FEC can recover.

The next subsection analyses different FEC configurations that allow to compensate BER in a WR-based control and timing network. This configuration is further tuned when discussing network reconfiguration.

---

<sup>3</sup>Jamboo frames are assumed.

### 4.6.3 Forward Error Correction Schema and its Reliability Analysis

This subsection proposes a FEC schema in which a single control message is transmitted in a number of Ethernet frames, some of which carry redundant data. It is analysed whether and with which configuration such schema can mitigate control message loss due to BER.

The proposed FEC schema divides a control message into  $N$  blocks of equal size ( $block_{size}$ ) and produces  $M$  parity blocks of the same size. All these blocks ( $Z = N + M$ ) are prepended with a special FEC header of 8 bytes (see Appendix E) and sent in payloads of separate Ethernet frames. Receiving any  $N$  Ethernet frames out of all the transmitted  $Z$  frames allows to recover the original control message. Reed–Solomon [82] encoding is proposed in [83] to produce parity blocks. A variety of other methods exist. This thesis analyses FEC's configuration, i.e.  $N$ ,  $M$ , while the specification of the FEC algorithm is outside of its scope<sup>4</sup>.

As an example of the proposed schema, Figure 4.4 depicts a simple configuration with  $N = 2$  and  $M = 2$ . If a bit error due to the fibre's BER occurs during transmission, it is likely that the Ethernet frame is corrupted [81] and considered lost. With this example configuration, any 2 out of the 4 FEC frames can be corrupted and the original message still can be recovered.

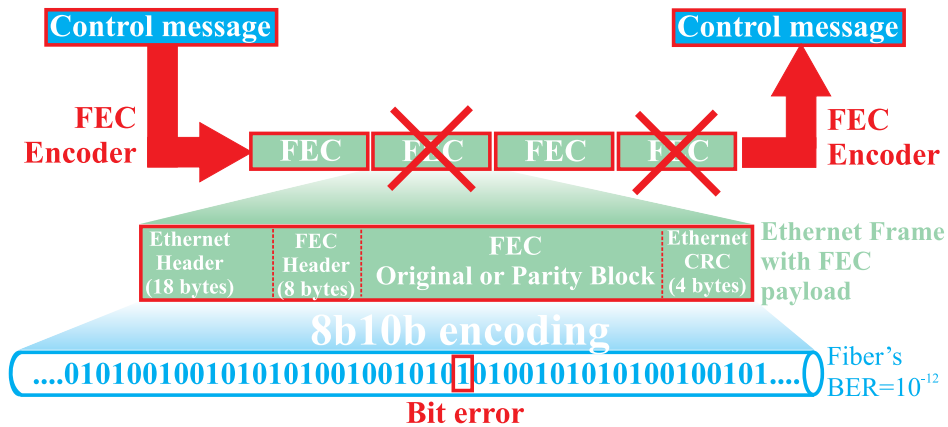


Figure 4.4: Principles of Forward Error Correction operation with an example configuration where  $N = 2$  and  $M = 2$ .

The probability of losing zero or one control message throughout a year of operation due to BER is calculated for different configurations (i.e.  $N$ ,  $M$ ) and control message sizes to verify that the schema allows to achieve the target reliability. The total number of bits transmitted on-the-wire (after 8B10B encoding) of each Ethernet frame that carries a part of original control message or generated parity data is

$$frame\_bits\_on\_wire [bits] = (Frame\_Overhead + FEC\_header + FEC\_block) \cdot 10 \quad (4.12)$$

where  $Frame\_Overhead = 22 [bytes]$ ,  $FEC\_header = 8 [bytes]$  and  $FEC\_block$  depends on the size of the control message and  $N$ . The probability of successful frame transfer over a single

<sup>4</sup>The thesis by Cesar Prados is to cover specific algorithms and optimisation of the FEC in WR.

fibre is calculated by accounting for the BER:

$$P_{frame\_OK} = (1 - BER)^{frame\_bits\_on\_wire} \quad (4.13)$$

The probability that a control message is successfully transferred over a single fibre, when it is sent using  $Z$  Ethernet frames of which  $M$  can be lost, is calculated as follows:

$$P_{CM\_OK} = \sum_{i=0}^M \left\{ \binom{Z}{i} P_{frame\_OK}^{Z-i} \cdot (1 - P_{frame\_OK})^i \right\} \quad (4.14)$$

Finally, the reliability,  $R_t$ , of a broadcast transmission of control messages from the data master to all the nodes throughout 1 year of operation is calculated. It is the probability that zero or one control message is lost in a network consisting of  $n_{fibres} = 4574$  fibres while being sent every 1 ms ( $ms\_in\_year = 3.15576 \cdot 10^{10}$ ):

$$R_t = P_{CM\_OK}^{(n_{fibres} \cdot ms\_in\_year)} + (n_{fibres} \cdot ms\_in\_year) \cdot P_{CM\_OK}^{(n_{fibres} \cdot ms\_in\_year - 1)} \cdot (1 - P_{CM\_OK}) \quad (4.15)$$

Table 4.7 presents reliabilities of transmission ( $R_t$ ) calculated for different configuration parameters ( $N$ ,  $M$ ) and different control message sizes. The results show that using a single parity frame allows to meet the target transmission reliability chosen in section 4.4 only for reasonably small control messages (green in column  $R_t(year, 1)$  in the table). However, when two parity frames are sent, the proposed FEC schema allows to meet the target reliability regardless of the control message size (green in column  $R_t(year, 2)$  in the table).

Control message size [bytes]	FEC_block: Block size [bytes]	N: original blocks [number]	R <sub>t</sub> for M:	
			R <sub>t</sub> (year, M) - probability of successful transmission with M parity blocks in a year	
			R <sub>t</sub> (year, 1)	R <sub>t</sub> (year, 2)
600	308	2	0.9999887	0.99999(9)
	608	1	0.9999836	0.99999(9)
1200	608	2	0.9998531	0.99999(9)
	1208	1	0.9997642	0.99999(9)
1492	1500	1	0.9994518	0.99999(9)
4800	608	8	0.9813315	0.99999(9)
	1208	4	0.9792960	0.99999(9)
	4808	1	0.9545252	0.99999(9)
6000	758	8	0.9592208	0.99999(9)
	1208	5	0.9566182	0.99999(9)
	6008	1	0.9021146	0.99999(9)

Table 4.7: Transmission reliability calculated for different Forward Error Correction parameters.

While FEC can prevent data loss, it also increases the data traffic. This must be taken into account when considering the methods to prevent congestion and guarantee an upper bound latency that are discussed in the next sections.

## 4.7 Congestion-Less Transmission

This section recommends strategy to mitigate the third failure factor considered in this thesis which is traffic congestion. Traffic congestion can affect mainly data transmission in terms of latency and loss of control messages. However, a prolonged congestion can also result in synchronisation problems. It is feasible and critical to completely eliminate the possibility that traffic congestion affects control and PTP messages.

However, using the commonly-available Class of Service (CoS), IEEE 802.1Q priorities, to give precedence to the control and PTP messages over other traffic is not sufficient. When the Ethernet frames transporting these messages are given higher priority than other traffic, these frames can still be affected by congestion in the case when:

- too many control messages are sent at the same time by too many nodes
- congestion of the frames with lower priority exhausts resources common to all the priorities.

The former reason can only be eliminated by proper design and configuration of the WR network. This includes strict control of the sources that transmit control messages. It is assumed in this thesis that a WR network designer and administrator has full and absolute control in this regards eliminating the possibility that too many control messages is sent simultaneously. A configuration that disallows receiver nodes from sending control messages is specified in this thesis.

The latter reason can be mitigated by providing dedicated resources to the control and PTP messages, similarly to Quality of Service (QoS) in IEEE 802.1AS Stream Reservation mechanism. The separation of resources in the WR switch needs to be such that the control and PTP message are not affected by congestion of any other traffic. This means that the switch should have memory and output queues reserved exclusively for control and PTP messages.

Proper configuration of the network, strict control of data masters, and dedicated switch resources for the control and PTP messages are the recommended means to provide congestion-less transmission in the WR-based control and timing network.

## 4.8 Latency and Timing Performance

This section analyses the last failure factor considered in this thesis which is the network performance. A WR network, in which all the network elements operate normally and all the control messages are delivered successfully without congestion, is considered non-operational (fails) if its performance does not meet the requirements. These requirements are defined and measured in terms of

- accuracy and precision of synchronisation and
- latency of control messages.

During normal operation of the network, its latency and synchronisation performance depend mainly on the number of switches (hops) from the data/timing master to the receiver nodes. During reconfiguration of the network, its latency and synchronisation performance depend on the efficiency of the used mechanisms and their implementation. The required determinism of the WR network translates into a known and guaranteed upper-bound latency and synchronisation performance both, in normal operation and during reconfiguration. At any time, the performance must meet CERN requirements.

In the following subsections the synchronisation and latency provided by the WR network are analysed and discussed separately. Both are characterised during normal operation of the WR network. Based on the performance in normal operation and the CERN requirements, the acceptable performance deterioration during reconfiguration is evaluated and the required switchover characteristics are proposed. These are the input requirements when developing mechanisms to support seamless redundancy in Chapter 5 and Chapter 6.

### 4.8.1 Synchronisation Performance

The synchronisation performance in normal operation of the WR network is characterised versus the number of switches (hops) from the timing master which is a WR switch acting as the PTP Grandmaster. The synchronisation is expected to deteriorate with the number of hops, primarily due to the cascaded phase-locked loops (PLLs) and uncalibrated hardware asymmetries.

Figure 4.5 depicts the test setup and the measured synchronisation performance between the Grandmaster and nine cascaded switches. The values of accuracy and precision are obtained by measuring the skew, the time error (TE), between the Pulse Per Second (PPS) or 10 MHz output of the Grandmaster and that of each of the switches. The PPS is used in the measurement for which results are indicated with red plots. This measurement is performed without paying attention to careful calibration of the switches prior to the measurement. The 10 MHz clock is used in the measurement for which results are indicated with the blue plots. This measurement is performed with careful prior calibration of the switches and it has been reported in [84]. For both measurements, dashed lines indicate the worst case accuracy scenario in which all

the offsets between the 9 switches are positive and thus the inaccuracy accumulates. In reality, the offsets between switches usually cancel out and better end-to-end accuracy is achieved. The standard deviation of the skew indicates that the precision is deteriorating exponentially with the distance from the Grandmaster, regardless of the calibration. The sub-50 ps precision

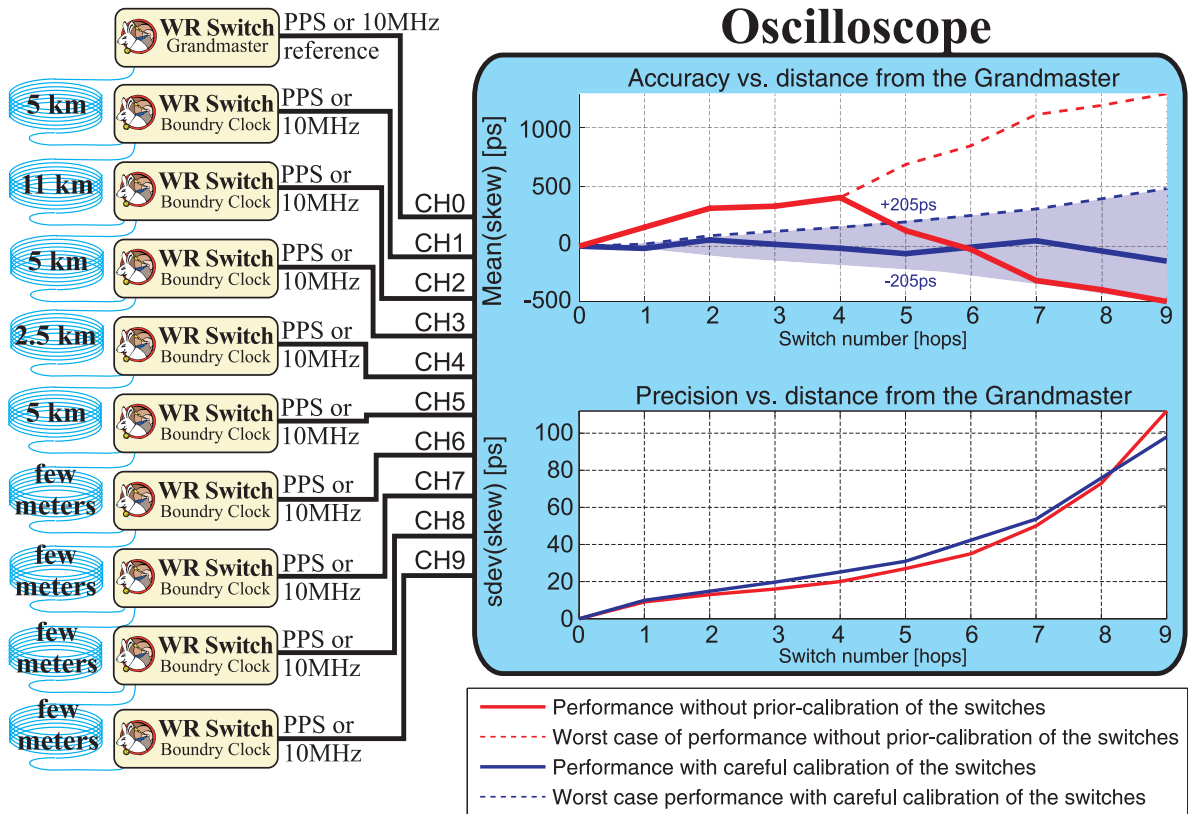


Figure 4.5: Measurement of synchronisation performance in a cascade of switches

requirement is not met at the 7th switch. Similarly, the worst-case accuracy exceeds 1 ns at the 7th switch in the case of the not-carefully-calibrated switches.

The results set the maximum number of hops in the network to 6 to meet CERN requirements. The network topology suggested in section 4.4 has 4 layers of switches and the receiving nodes are 5th in the cascade. Such a network meets the requirements of sub-ns accuracy and sub-50 ps precision in normal operation. When the receiver nodes are 5th in the cascade and all the switches/nodes are carefully calibrated, the margin for accuracy deterioration during reconfiguration is around 800 ps. For not-carefully-calibrated switches/nodes, this margin is around 300 ps. It is therefore recommended that the deterioration of synchronisation accuracy during network reconfiguration (switchover) is at most 500 ps and preferably below 300 ps.

## 4.8.2 WR Network Latency Performance

This subsection translates the CERN requirement from Table 1.1 into the latency performance of the WR network. The contributors to this latency are identified and optimised to meet the requirements. These optimisations affect the FEC schema and allow to estimate the switchover time. It is the time allowed for reconfiguration during which network connectivity is disrupted and FEC frames can be lost. As long as this time is not exceeded, the original control message can be recover using FEC and thus seamless redundancy is provided. The switchover time (also referred to as failover) is an important parameter for the methods developed in Chapter 6.

The requirement of the CERN control and timing network to schedule events for the subsequent millisecond means that the latency introduced by the WR network must be much smaller than 1ms. This is because in 1ms a control message needs to be properly generated and encoded into FEC frames by the data master, transmitted over the network, decoded from the FEC frames, finally interpreted and executed by the node, as depicted in Figure 4.6. This thesis is

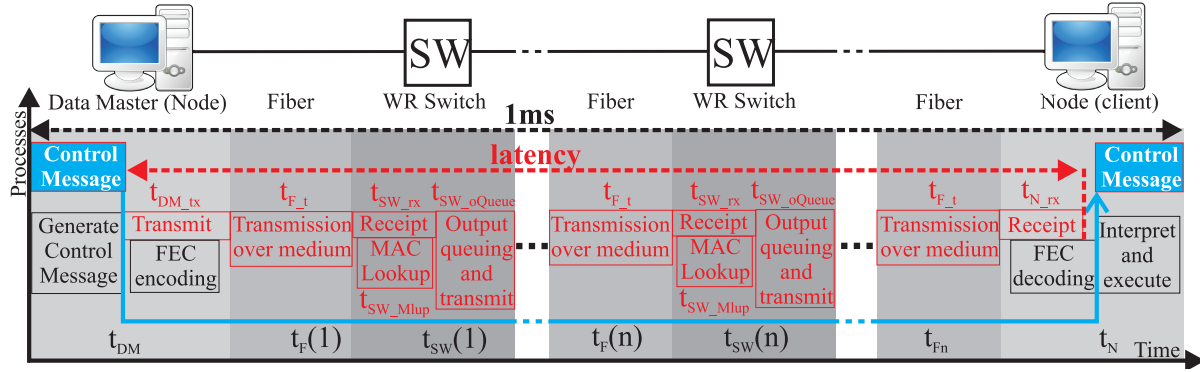


Figure 4.6: Contributors to the latency between generating control message by data master and triggering event at a node.

concerned only with the latency introduced by the WR network which is measured between the transmit of the first bit and the receipt of the last bit of the frames carrying a single control message. Out of the 1ms, allocating a maximum of  $500\mu s$  for the latency introduced by the WR network seems a reasonable target. Therefore, the 1ms requirement of the entire control and timing system in Table 1.1 is actually  $500\mu s$  for the underlying WR network.

Figure 4.6 depicts in red colour contributors to the WR network latency which are described in detail in Appendix F. The total WR network latency can be described as follows:

$$latency [\mu s] = t_{DM\_tx} + t_{F\_total} + n \cdot (t_{SW\_rx} + t_{SW\_oQueue}) \quad (4.16)$$

where

$t_{DM\_tx}$  is the transmission time of all FEC frames carrying a control message

$t_{F\_total}$  is the transmission time over 10km of fibre, i.e.  $t_{F\_total} = \sum_{i=0}^n t_{F\_t}(i) = 50\mu s$

$t_{SW\_rx}$  is the reception time of a single FEC frame by the WR switch, or the time it takes the switch to provide a forwarding decision, depending on which is dominant

$t_{SW\_oQueue}$  is the time introduced by the output queues of the WR switch.



Analysis of the WR network latency revealed that the size of the Inter-Frame Gap (IFG) between the FEC frames has a significant impact on the total WR network latency. It affects not only the transmission time of all the FEC frames,  $t_{DM\_tx}$ , but most importantly the latency introduced by the output queues of the WR switch,  $t_{SW\_oQueue}$ . If FEC frames are sent in a burst with minimum IFG, only the first FEC frame is affected by any other traffic already in the output queues. If the IFG between FEC frames is larger than minimal, transmission of other traffic can be scheduled between each FEC frame, thus each FEC frame is equally affected by such traffic. The difference in the total WR network latency is significant. Figure 4.7 shows the

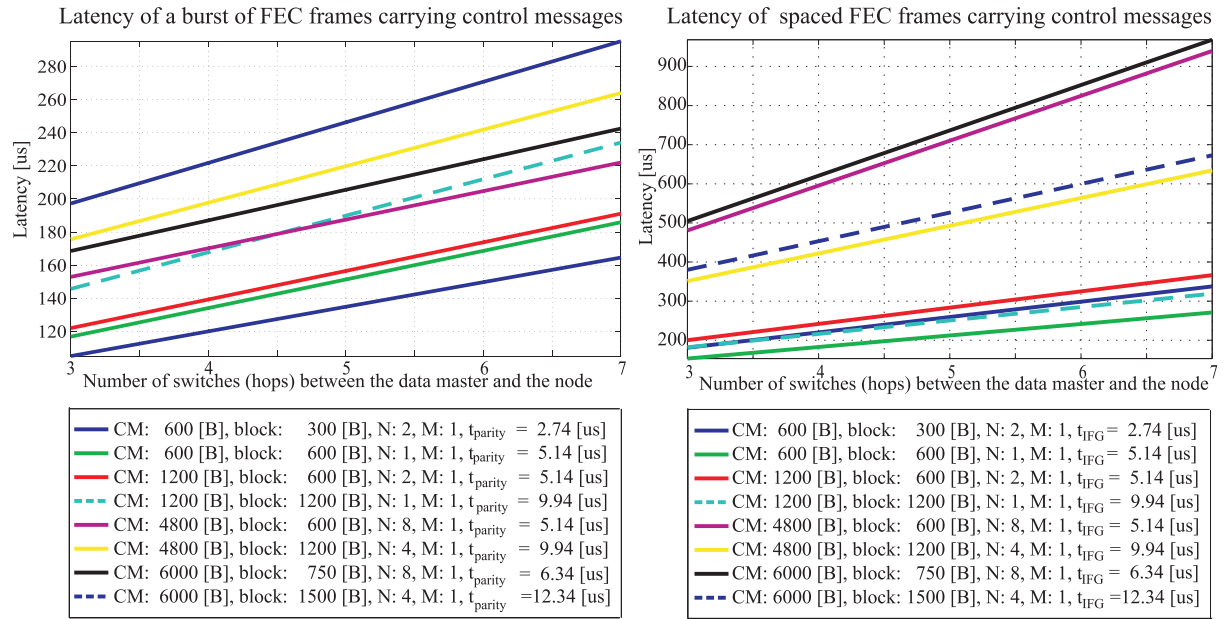


Figure 4.7: Network latency for a control message

latencies for different control message sizes, FEC schemas and network sizes (hops) when the IFG is minimal (left) and when it is of a size comparable to a single FEC frame (right). The IFG is made comparable with the FEC frame size so that the two simulations are comparable in terms of the value of  $t_{DM\_tx}$  and the allowed time for switchover. The later is important in further considerations. When the IFG is not minimal, the WR network latency might exceeded  $500 \mu s$  for some FEC schemas and message sizes. On the other hand, when FEC frames are sent in a burst, the latency is always much below  $500 \mu s$ . Thus, it is recommended that control messages are always sent in burst with minimal IFG. This has an impact on the FEC schema to be used.

The FEC schema specifies the number of the original and parity FEC frames,  $M$  and  $N$ . An effective FEC schema provides sufficiently high probability of recovering lost frames not only due to the BER (see subsection 4.6.3) but also due to the network reconfiguration. While BER is likely to result in a loss of a single frame, the network reconfiguration is likely to affect a number of subsequent frames, especially if they are sent in a burst with minimum IFG. As depicted in Figure 4.8, the loss of subsequent FEC frames can be prevented by increasing the

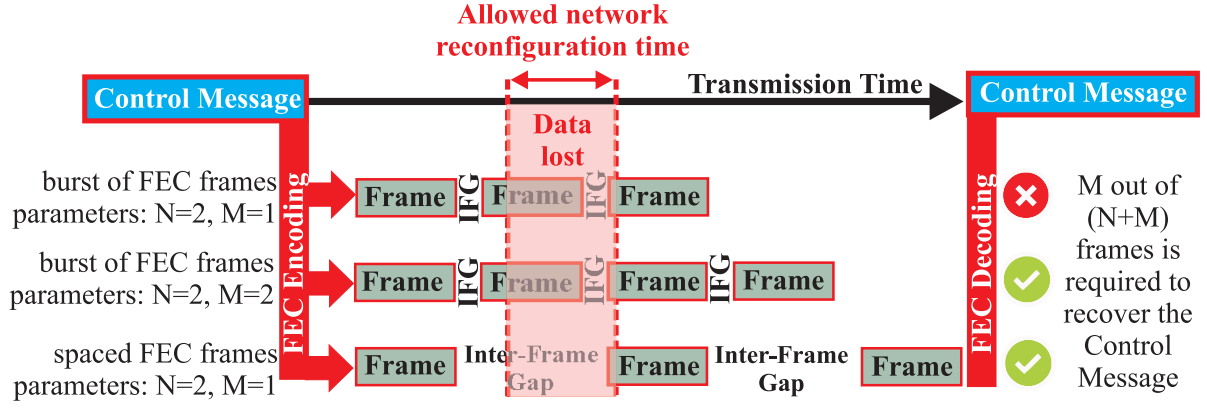


Figure 4.8: Relation between the allowed network reconfiguration time, Forward Error Correction configuration, and Inter-Frame Gap size.

IFG between the frames. However, this results in a substantial increase in latency that has been explained in the previous paragraph and shown in Figure 4.7. Alternatively, loss of subsequent FEC frame can be accounted for by sending sufficient number of parity frame. In such case, when FEC frames are sent in a burst with the minimum IFG, the relation between the FEC schema and the time allowed for reconfiguration can be precisely specified as follows:

$$switchver [\mu s] = (M - 1) \cdot (t_{IFG} + t_{FEC}) \quad (4.17)$$

where

$M$  is the number of parity frames

$t_{IFG}$  is the transmission time of minimum IFG, which is  $0.096\mu s$

$t_{FEC}$  is the transmission time of a FEC frame.

Table 4.8 provides calculations of WR network latency and the allowed switchover time for different sizes of control message and different FEC schemas assuming that FEC frame are sent in a burst with the minimum IFG. Depending on the FEC schema and control message size, the time allowed for switchover is between  $2.7$  and  $24.7 \mu s$ . In order to cover the control message size range required in Table 1.1 (1200-6000 bytes) without excessive number of the FEC parity frames ( $M$ ), the switchover time needs to be in the range between  $5$  and  $10 \mu s$ .

The network latency provided in Table 4.8 is calculated for a chain of 5 switches between the data master and the receiver nodes to verify the performance for the maximum number of hops to meet the synchronisation performance. The calculations are done for two types of switch architecture: store-and-forward (S-and-F) and cut-through (C-T). The latency meets the  $500\mu s$  requirement for both switch types and all FEC schemas. However, a C-T switch architecture is recommended to meet the latency requirement of the Btrain application, as described in the following subsection.

Control message size	FEC frame: Payload size	N: original blocks	M: parity blocks	Network latency		Switchover time allowed
				5 switches S-and-F	C-T	
[bytes]	[bytes]	[number]	[number]	[ $\mu s$ ]	[ $\mu s$ ]	[ $\mu s$ ]
600	308	2	2	135	127	2.7
			3	138	130	5.5
	608	1	2	151	131	5.1
			3	157	136	10.3
1200	608	2	2	157	136	5.1
			3	162	142	10.3
	1208	1	2	190	146	9.9
			3	200	156	19.9
4800	608	8	2	188	167	5.1
			3	193	172	10.3
	1208	4	2	220	175	9.9
			3	230	185	19.9
6000	758	8	2	206	179	6.3
			3	212	186	12.7
	1508	4	2	246	190	12.3
			3	258	202	24.7

Table 4.8: Network latency and allowed switchover times calculated for different Forward Error Correction configurations and number of switches between the data master and node.

### 4.8.3 WR Switch Latency Performance

The application of WR for the Btrain system requires latency of a control message through the WR switch below  $10\mu s$ . This latency is measured between the reception and the transmission of the first bit in the frames carrying the control message.

As described in the previous subsection, the latency of a switch has two components:  $t_{SW\_rx}$  and  $t_{SW\_oQueue}$ . A typical switch has store-and-forward (S-and-F) architecture. It receive an entire frame before forwarding it to destination ports, thus the reception time  $t_{SW\_rx}$  depends on the size of this frame. A common optimisation of switch latency is the cut-through (C-T) architecture. A C-T switch forwards each frame as soon as the forwarding decision is ready even before the frame is fully received. Thus, the reception time  $t_{SW\_rx}$  depends solely on the time it takes to make the forwarding decision. Despite this optimisation, the worst-case latency of the C-T switch is greater than  $12\mu s$ , which is the transmission time of the maximum-size Ethernet frame. In the worst-case scenario, the transmission of such a frame needs to be completed before the FEC frame carrying control message can be transmitted.

In order to meet Btrain latency requirement, both of the two contributors to the worst-case latency of a C-T switch must be optimised: the time it takes to prepare the forwarding decision and an ongoing transmission of any frame not carrying control message. The later can be done by either restricting the size of frames not carrying control messages in the entire WR network, or devising a method to make the latency independent from such frames. Restricting the size of the traffic is not standard-compatible. Thus, it is recommended to devise a method to make the latency independent from the size of frames that do not carry control messages.

## 4.9 Proposed Strategy

The discussions, simulations and calculation in the previous sections result in the following strategy to increase the reliability and ensure the determinism of a WR network:

1. Network topology: redundant mesh with maximum 5 switches (hops) between the data/timing master and any receiver node.
2. FEC used to increase the reliability of control message transmission:
  - a) Minimum two parity frames.
  - b) Frames sent in a burst with a minimum IFG.
3. Requirements of the network redundancy support for data:
  - a) Network reconfiguration time of  $5 - 10 \mu s$ , determined by FEC configuration.
  - b) Network reconfiguration due to topology update without losing a control message.
4. Requirements of the network redundancy support for synchronisation:
  - a) Multi-path synchronisation in meshed network.
  - b) Switchover such that sub-ns accuracy is maintained, the accuracy deterioration during reconfiguration at most 500 ps and preferably below 300 ps.
5. Elimination of congestion for control and PTP messages:
  - a) The highest priority reserved exclusively for control and PTP messages.
  - b) Dedicated resources for the highest priority, thus control and PTP messages.
  - c) Configuration of the switches to prevent receiver nodes from sending the highest priority traffic.
6. Requirement of switch latency performance for Btrain:
  - a) Guaranteed upper-bound latency through the switch for control messages of  $5 - 10 \mu s$ .
  - b) Latency of frames carrying control messages independent from their size and the size of other frames.

The following chapters describe mechanisms that increase the reliability and ensure determinism of the WR network following the outlined strategy.



## Chapter 5

---

# Methods and Algorithms for Synchronisation Resilience

---

WR has consistently shown sub-ns synchronisation accuracy [85][86][87], provided no failure occurs. The application of WR in critical and high-availability systems requires redundancy to increase the overall reliability of the system. Only if the required synchronisation accuracy is preserved during network re-arrangement, can the redundancy of elements seamlessly increase the reliability of the entire WR network.

This chapter describes methods and algorithms that have been developed by the author to support high accuracy of synchronisation during network reconfiguration. It first briefly explains the essentials of synchronisation and syntonisation in WR to allow proper interpretation of the initial CERN requirements (subsection 1.1.4). Based on this interpretation, the exact problem to be solved is stated and the high-level architecture of the solution proposed. This solution is then analysed and outlined in detail. Finally, its implementation and tests are described. The tests confirm that the proposed methods can ensure sub-ns synchronisation in redundant networks which is a 1000-fold improvement compared to the other existing solutions.

The developed mechanisms support the topology recommended in Chapter 4 for the WR network and can be used with the mechanisms developed in Chapter 6. The mechanisms developed in this chapter are used in the reference design of a WR-based control and timing network that is described in Chapter 7.

## 5.1 Background

### 5.1.1 Synchronisation and Syntonisation in White Rabbit

The particular relation between synchronisation and syntonisation in WR is critical in achieving its unprecedented performance and thus in maintaining it during network reconfiguration. Any acceptable deterioration of performance during network reconfiguration depends on how the synchronisation and syntonisation are used by WR application. This section analysis the relation between synchronisation and syntonisation and their usage by WR applications. In WR, the synchronisation is essentially performed by the Precision Time Protocol (PTP) protocol, therefore it is called *PTP synchronisation* in this section. The syntonisation is performed physically at Layer 1 (L1), therefore it is called *L1 syntonisation*. While the details of WR operation are provided in [7], this section focuses on essentials required to understand the implication of CERN requirements and the mechanisms developed in this thesis.

In PTP synchronisation, depicted in Figure 5.1, the *Local PTP Clock* of a slave node provides the *Local PTP Time* synchronised to that of the master node and consequently to the reference of time in the network, the PTP Grandmaster. The *Local PTP Clock* is a *time counter* incremented at a rate of the *Local PTP Clock signal*. A slave device is syntonised to a master device, if their *time counters* advance at the same pace, i.e. if both devices share the same definition of 1 second. A slave device is synchronized to a master device if their *time counters* show the same value at the same instant of time. In many PTP implementations, synchronisation is achieved by adjusting the value of the *time counter* at the slave but not the frequency of the *clock signal* driving this counter.

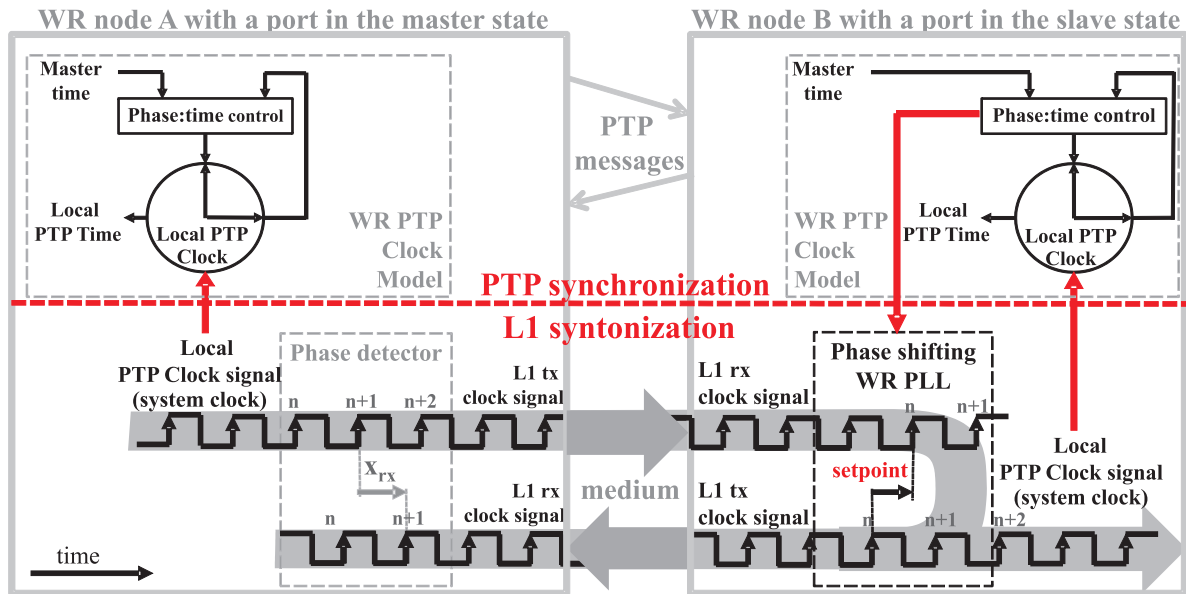


Figure 5.1: Synchronisation and syntonisation in WR.

In WR, not only is *Local PTP Clock signal* on the slave syntonised physically (i.e. L1 syntonisation) to that of the master but also the phase of the *Local PTP Clock signal* at the slave is aligned to that of the master by phase-shifting, as depicted in Figure 5.1. The process of synchronizing a WR slave is performed in the following steps:

1. the *Local PTP Clock signal* at the slave is syntonised with the *L1 rx clock signal* recovered from the data received from the master
2. the *time counter* at the slave is synchronised with that at the master
3. the *Local PTP Clock signal* at the slave is phase-aligned with that at the master with sub-ns accuracy<sup>1</sup>
4. the sub-ns synchronisation is maintained by adjusting the phase of the *Local PTP Clock signal*, so-called *phase tracking*<sup>2</sup>.

Notably, the synchronisation in WR is maintained by adjusting the phase (phase-steering) rather than manipulating the *time counter* value. During *phase tracking*, the phase-locked loop (PLL) not only syntonises the *Local PTP Clock signal* with the *L1 rx clock signal* recovered from the carrier but also maintains the desired phase offset between these two clock signals, a so-called setpoint. The setpoint is derived from the PTP measurement of master-to-slave offset ( $offset_{ms}$ ):

$$setpoint = offset_{ms} \bmod(T_{REF}) \quad (5.1)$$

where  $T_{REF}$  is the period of the *Local PTP Clock signal*.

Furthermore, the *Local PTP Clock signal* is used as the *system clock* that drives the advancement of processes in WR switches and WR nodes. As a result, all WR devices in the entire WR network work with the very same frequency and their *time counters* are incremented at the rising edge which occurs at the same instant of time to within 1ns. This accurate synchronisation and syntonisation of the WR devices is used in a variety of applications. Such applications timestamp incoming signals with sub-ns accuracy and tens of ps precision in spatially distributed locations or generate in these locations phase-aligned clocks. Notably, for such applications, maintaining performance of synchronisation during network reconfiguration concerns the *time counter*, the *clock signal* (frequency) and its phase alignment.

The key element in achieving the sub-ns synchronisation performance in WR, and thus maintaining it during switchover, is the dedicated WR phase-locked loop (WR PLL). Its operation is described in the next subsection.

---

<sup>1</sup>Phase adjustment is performed by temporarily manipulating the frequency.

<sup>2</sup>The necessary adjustments are mainly caused by the changes of the medium temperature which are quite small. The variation of the link delay due to temperature is estimated at 40 ps per degree Celsius per 1 km, which means a phase-shift of 8ns for a 10km link after a temperature change of 20 degrees Celsius.



### 5.1.2 White Rabbit Phase-Locked Loop (WR PLL)

This section provides an outline of the original architecture of the WR PLL, detailed in [7]. This WR component is enhanced in the context of this thesis to provide support for network redundancy.

The WR PLL uses an FPGA-based Digital Dual Mixer Time-Difference (DDMTD) [21] phase detector which produces timestamps at the edges of the compared *clock signals*. These timestamps are fed into a software implementation of a Proportional-Integral (PI) controller which runs in in a embedded CPU [88]. The controller steers a Voltage-Controlled Crystal Oscillator (VCXO). Figure 5.2 depicts a simplified block diagram of the WR PLL, which

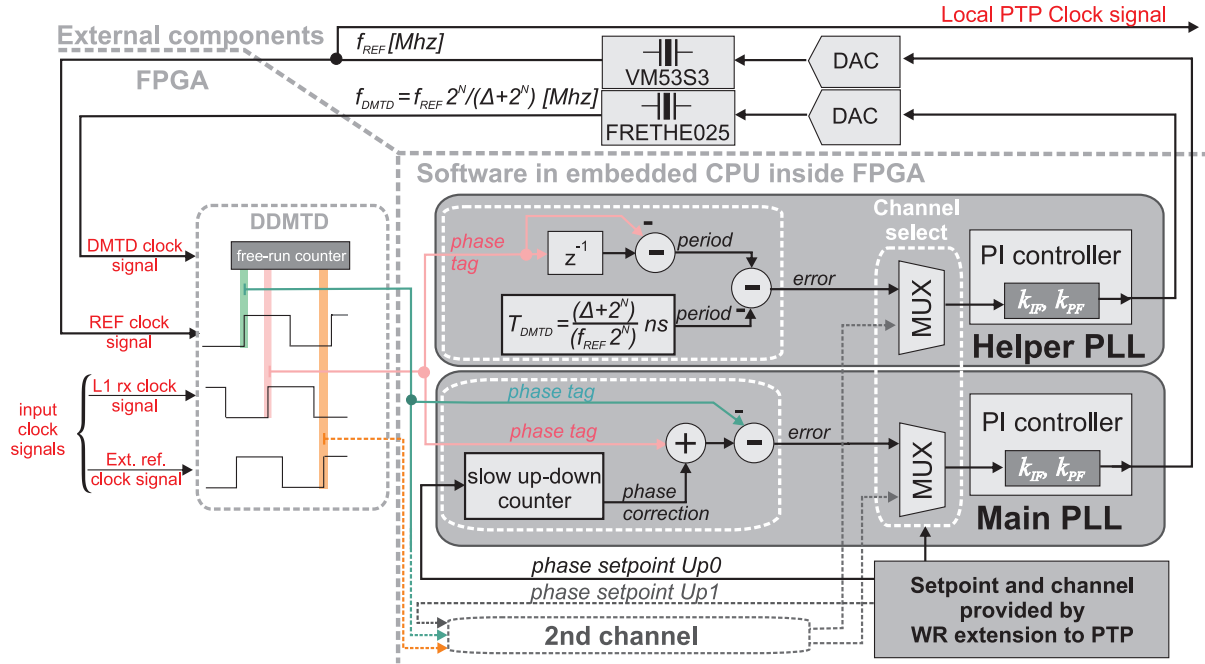


Figure 5.2: Overview of the WR phase-locked loop design.

actually consists of two PLLs: Helper and Main.

The *Helper PLL* (*hPLL*) produces the *DMTD clock signal*, which is necessary for the operation of the DDMTD. Its frequency is close but not equal to that of the *REF clock signal*:

$$f_{DMTD} = f_{REF} * \frac{2^N}{\Delta + 2^N} \quad (5.2)$$

It works by comparing the difference between subsequent timestamps to the "ideal" period of the *DMTD clock signal*.

The *Main PLL* (*mPLL*) produces the *REF clock signal* which is a copy of one of the *input clock signals*, phase shifted by a programmable *setpoint* provided by the WR extension to PTP (WR PTP). It works by comparing the timestamps of the chosen *input clock signal* (*L1 rx clock signal* in Figure 5.2) with the timestamps of the *REF clock signal*, corrected for the *setpoint*.

The output of the WR PLL, the *Local PTP Clock signal*, is used in WR applications. Their requirements are discussed in the next subsection.

### 5.1.3 WR Requirements for Frequency and Time Transfer

Due to the nature of WR applications and the way WR operates, the synchronisation accuracy requirement in the WR network has much more stringent and wider meaning than in most of the PTP networks. It concerns both, frequency and time transfer.

As already explained, in WR applications, the phase-aligned exact copy of the WR Grandmaster's *Local PTP Clock signal*, along with the value of its *time counter*, are used directly by the WR nodes. These applications include timestamping of asynchronous input signals with sub-ns accuracy and tens of ps precision, as well generation of precisely synchronized output signals.

Consequently, seamless redundancy of synchronisation in White Rabbit requires continuous phase-alignment, syntonisation, and synchronisation of all the WR nodes with the WR Grandmaster. The maximum time error (MTE) between the time of the WR Grandmaster and the time of any WR node, measured as the phase error between their *clock signals*, should not exceed the required accuracy over the considered period of operation, including the network rearrangement. In this thesis, such seamless switchover is considered for the networks with redundant meshed topologies, according to the strategy in Chapter 4. The considered network arrangements are presented in Figure 5.3 and include:

- (a) redundancy of links only
- (b) redundancy of links and switches
- (c) redundancy of Grandmasters, assuming they are synchronised at sub-ns level.

For each type of arrangement, or a mixture of them, failure of a redundant element (link or switch) should not deteriorate the synchronisation performance beyond the required accuracy. The mechanisms to support such a seamless redundancy are developed in the next section.

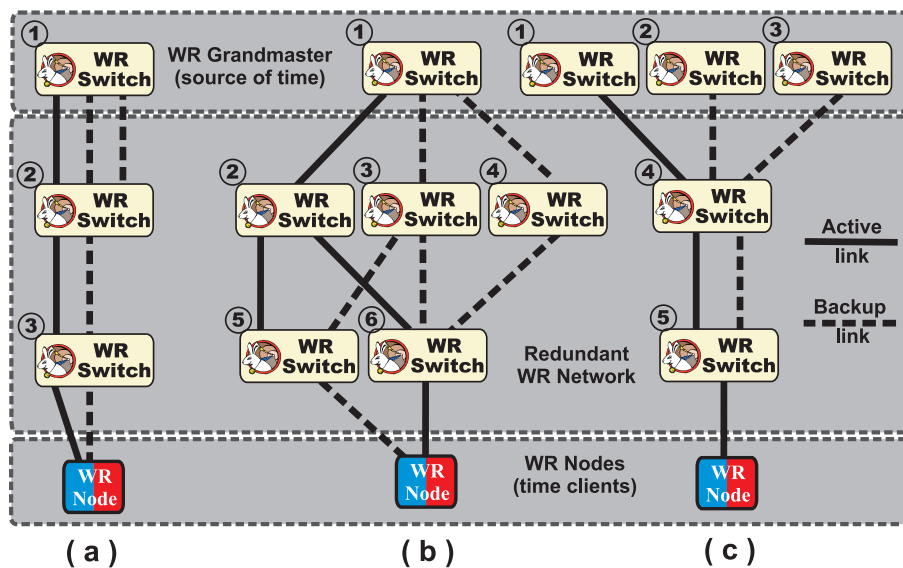


Figure 5.3: Considered network arrangements to support redundancy for synchronisation.



## 5.2.2 Architecture

This subsection outlines the support for seamless redundancy developed in the context of this thesis. The subsequent subsections provide details of the introduced ideas.

Figure 5.5 summarizes the principles of seamless switchover. The Grandmaster's Local PTP Clock signal (*GM clk*, red) is the reference of time and frequency for the entire WR network. Its rising edge marks its PTP time. A slave WR switch (bottom right) is synchronized to the Grandmaster over a WR network through two redundant paths: one connected to the active Port A, another connected to the backup Port B. The performance of the synchronisation between the Grandmaster and the slave switch is evaluated by measuring the time error (TE) between the Grandmaster's Local PTP Clock signal (*GM clk*, red) and the switch's Local PTP Clock signal (*ref clk*, orange). The Local PTP Clock signal of the switch (*ref clk*) is syntonised to the L1 rx Clock signal (*rx clk A*, green) recovered at Port A until switchover takes place. It then becomes syntonised to the L1 rx Clock signal (*rx clk B*, blue) recovered at the Port B.

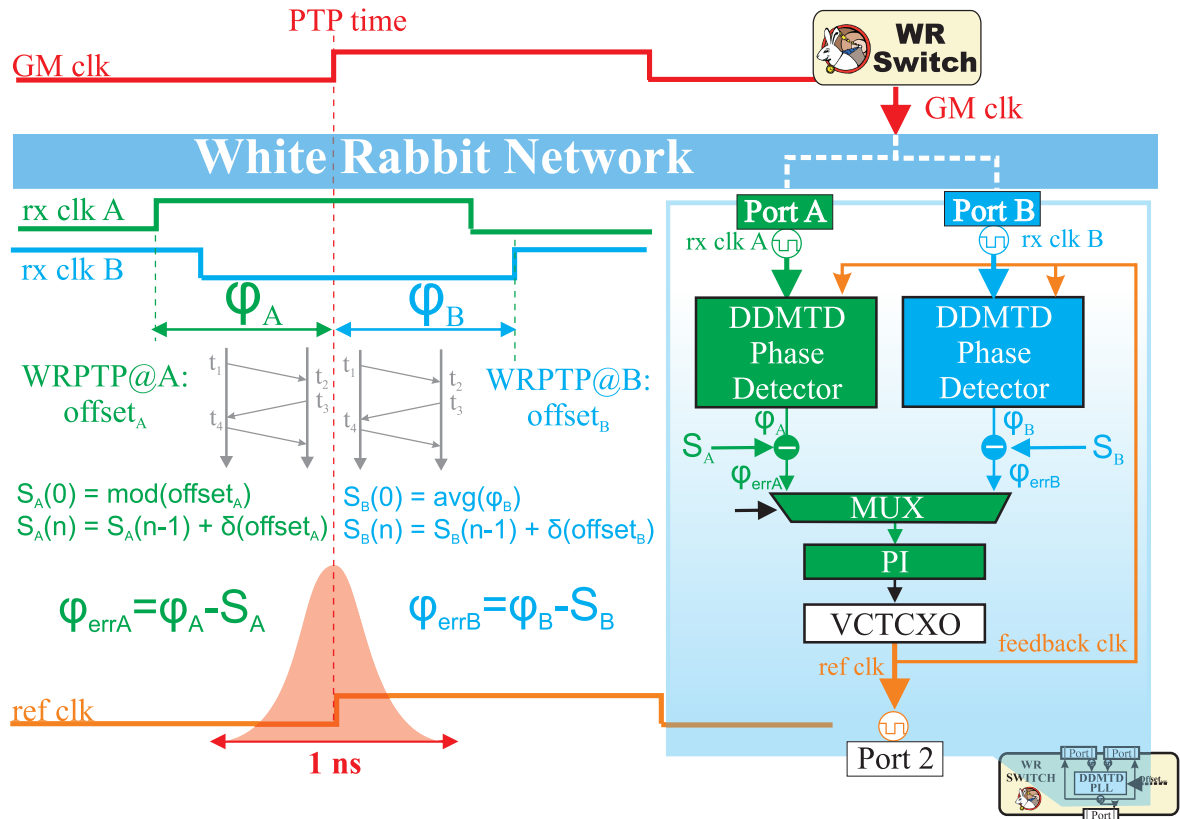


Figure 5.5: Principles of functioning of support for synchronisation redundancy.

Initially, the slave switch establishes synchronisation via active Port A following the description in subsection 5.1.1: the *clock signal (ref clk)* of the slave switch is syntonised to the recovered clock (*rx clk A*), its *time counter* is synchronized, then the setpoint is calculated from the measured offset from master:

$$S_A(0) = offset_A \bmod(T_{REF}) \quad (5.3)$$

where  $T_{REF}$  is the period of the *clock signal*.

At this point, the phase error provided to the PI controller results from comparing the intended setpoint value and the currently measured phase offset  $\varphi_A$ :

$$\varphi_{errA} = \varphi_A - S_A \quad (5.4)$$

The synchronisation is maintained by updating the setpoint differentially:

$$\begin{aligned} \delta offset(n) &= offset_A(n) - offset_A(n-1) \\ S_A(n) &= S_A(n-1) + \delta offset_A \end{aligned} \quad (5.5)$$

The developed support for seamless switchover supplements the above WR synchronisation with a number of stages in the operation of active and backup ports. Each of these stages is briefly described below.

**Backup slave setup:** A port selected by protocol or configuration to be a backup slave is connected. The WR synchronisation is started following the same steps as it would happen on the active slave port. However, unlike on the active slave, the initial value of the setpoint ( $S_B$ ) on the backup slave is set to the average of the detected phase difference between the recovered L1 rx clock signal (*rx clk B*, blue) and the switch's Local PTP Clock signal (*ref clk*, orange), as depicted in Figure 5.5:

$$S_B(0) = avg(\varphi_B) \quad (5.6)$$

**Multi-phase tracking:** The active and backup connections work properly, the slave switch is syntonised and synchronized via active port. The offset measured by WR PTP at the backup port is used for differential update of the setpoint:

$$\begin{aligned} \delta offset_B(n) &= offset_B(n) - offset_B(n-1) \\ S_B(n) &= S_B(n-1) + \delta offset_B \end{aligned} \quad (5.7)$$

Correcting only for the changes of the master-to-slave offset allows to ignore any potential differences between offset values measured by WR PTP on the active and backup ports. These differences might result, for example, from uncompensated asymmetries and cannot be corrected on backup ports. The phase error provided as a backup input to the PI controller results from comparing the "intended" setpoint value and the measured phase offset  $\varphi_B$ :

$$\varphi_{errB} = \varphi_B - S_B \quad (5.8)$$

At this stage, the phase errors at the active and backup ports ( $\phi_{errA}$ ,  $\phi_{errB}$ ) are close to zero whereas the master-to-slave offset measured by the WR PTP at the backup port might not be close to zero. It should however, stay below 1 ns. The "behaviour" of the most important WR PLL parameters during this and the following stages are presented in Figure 5.6. These parameters include: phase errors measured at ports A and B in Figure 5.5 ( $\phi_{errA}$  and  $\phi_{errB}$ ) that are initially active and backup respectively, the setpoint at the backup port B ( $S_B$ ), and the phase error ( $\phi_{err}$ ) provided to the PI controller (it is equal to  $\phi_{errA}$  before switchover and to  $\phi_{errB}$  after switchover).

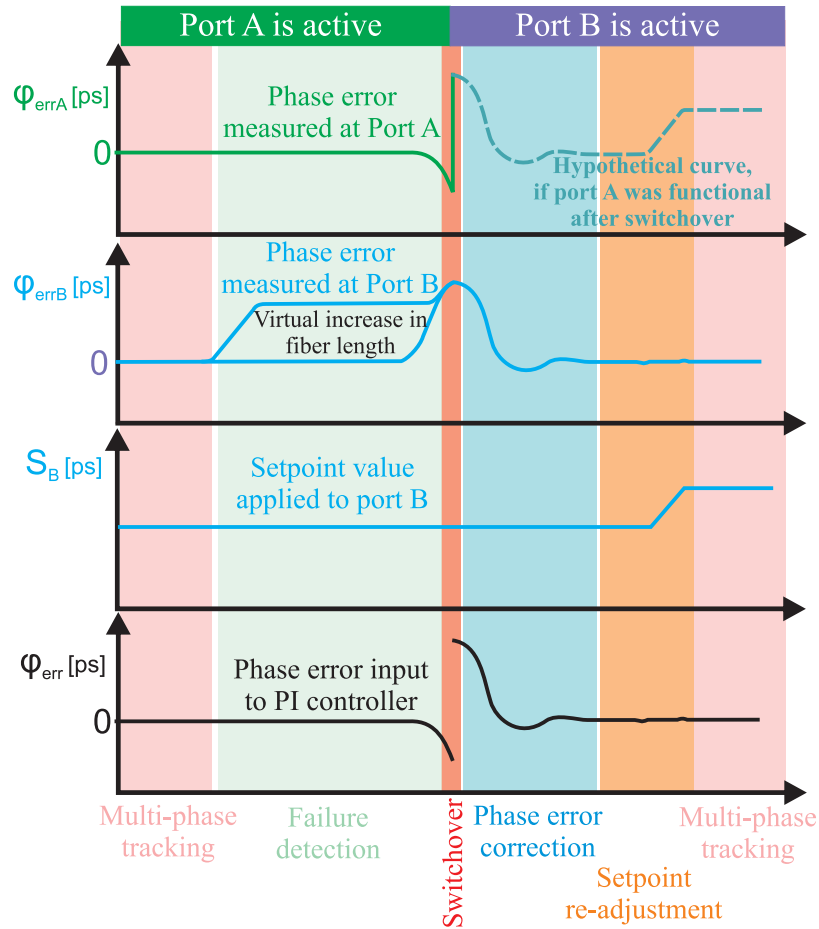


Figure 5.6: Changes of the WR phase-locked loop parameters for active and backup port before, during, and after switchover.

**Failure detection:** A malfunction of the active connection is detected while the backup connection works properly. Ultra-fast detection of the failure is required to prevent the phase error from exceeding the requirements. Some failures might start with a slow drift of phase at the active port ( $\phi_A$ ). Such a phase change is followed by the WR PLL before being detected as a problem. The resultant change of the timebase can be detected looking at the phase error on the backup ports. This is depicted in Figure 5.6, with the phase error at port B increasing at some point of the *failure detection* stage. However, it can be unambiguously recognised as failure of

the active port only if two or more backup ports are available. Studies presented in the next subsections show that sufficiently fast and reliable failure detection is a very challenging task.

**Switchover:** The source of phase-error input to the PI controller is changed from the currently active port A ( $\phi_{errA}$ ) to the backup port B ( $\phi_{errB}$ ). The phase of the *ref clk* has likely followed the phase of the failing port. Thus, when switching to the backup reference a phase correction is needed.

**Phase error correction:** Any phase error that is built up during *failure detection* appears as a unit step to the PI controller. This error is corrected by the controller and the phase is moved back to the position it was in the *multi-phase tracking* stage.

**Setpoint re-adjustment:** Until now, the value of the  $offset_B$  at Port B has been ignored. Only the changes of this value have been taken into account. At this stage, the setpoint is re-adjusted taking into account the current average phase error and the master-to-slave offset measured ( $offset_B$ ) by the protocol:

$$S_B(0) = avg(\phi_B) + offset_B \quad (5.9)$$

**(Multi-)phase tracking:** The synchronisation is maintained. If there are backup ports available, the multi-phase tracking is applied. Otherwise, phase-tracking without backups is used.

The above approach to a seamless switchover presents a number of challenges. First, the backup port channel is an open-loop system which must mimic a closed-loop behaviour in order to be always ready to take the role of the active port. Second, the most critical stage, *failure detection*, must be performed in a reliable and fast manner. Tests show that hardware detection of syntonisation loss is not always fast enough to prevent large phase-drift. Last, the phase-jump after switchover must not destabilise the WR PLL at the switch on which switchover occurs, and all the WR PLLs in the downstream switches and nodes. These challenges are addressed in the following subsections.

### 5.2.3 Switchover Theoretical Model and Simulation

A multiple-source model of the WR PLL is presented in Figure 5.7. This model is used in simulation to:

- verify that switchover does not destabilise the PI controller
- minimise overshoot during switchover
- propose a solution that works for a wide range of parameters.

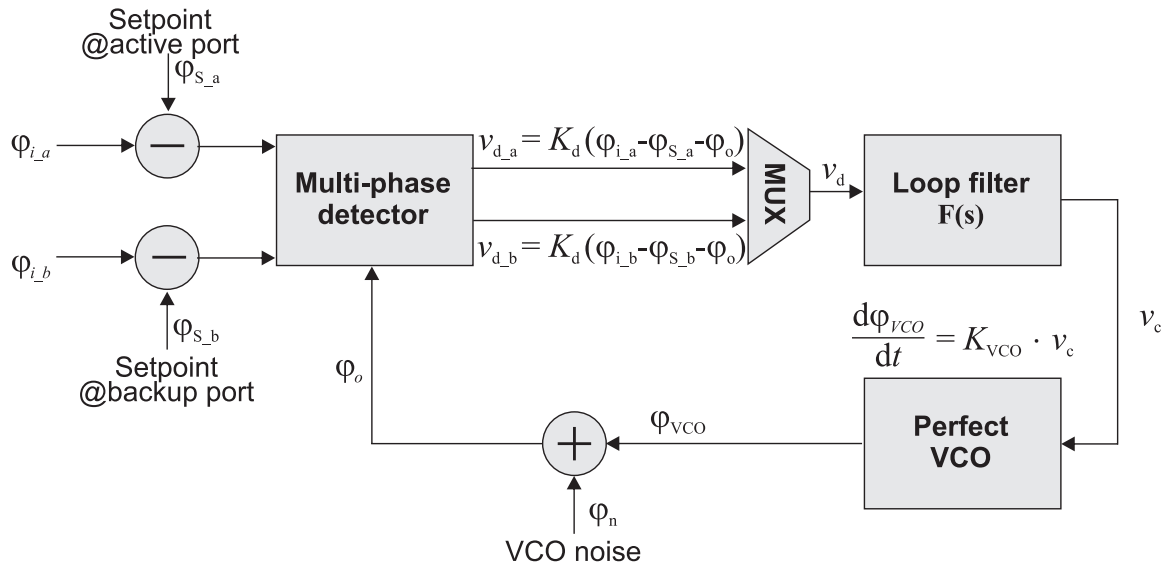


Figure 5.7: Model of WR phase-locked loop with support for synchronisation redundancy.

When switching over from active to backup port, there are two main contributors to phase-jumps which need to be considered in the simulations:

1. Phase error which develops during the *failure detection* stage.
2. Phase error which results from difference in the measured value of the master-to-slave offset at the active port and the backup port.

These two contributors are treated separately as they have their source in two different feedback systems: fast WR PLL phase measurement and slow PTP master-to-slave offset measurement. The phase error which develops during *failure detection* is potentially the most substantial and is corrected for in the first place.



### First stage: phase error correction

Correction of the phase error that is built-up during *failure detection* is critical in seamless switchover. A number of experiments showed that the phase of the failing active port shifts before the failure is detected. This shift is followed by the PI controller and results in a phase error between the Local PTP Clock signal of the slave and that of the Grandmaster (orange *ref clk* and red *GM clk* in Figure 5.5). While the active failing port follows a virtual phase-shift, the backup port measures increased phase error, as depicted in Figure 5.6, blue plot. When the failure of an active port is finally detected, the new input value to the PI controller is substantially different than the one measured so far. Such an input step change to the controller, depicted in Figure 5.6, is undesired. First, its compensation can result in deviation of frequency from the "ideal" (Grandmaster's) that is much bigger than the frequency deviation before the switchover. Such a frequency deviation might be unacceptable by the WR applications that measure precisely, for example, short time intervals. Second, it can be dangerous for the stable operation of the WR PLLs in the downstream WR devices.

In order to avoid significant input step change to the PI controller, the phase error registered at the switchover instant is applied gradually to the controller with a minimal, least significant bit (LSB), increment, i.e. 0.977 ps every 262.16  $\mu$ s. Such a solution limits the deviation from the "ideal" frequency during phase correction by a factor of 10-100, depending on the PI parameters. The model presented in Figure 5.7 is used to simulate behaviour of the WR PLL with such a solution for different use cases of phase error profiles and for a range of PI parameters. Figure 5.8 shows one of the tested use cases simulated for a number of PI parameters. The solid lines show the WR PLL behaviour when the proposed correction is not used, it is called "normal". The dashed lines show the same cases when correction is applied, it is called "correc". It can be clearly seen from the figure that the correction of the time error that results from the failure, Figure 5.8 a), is much smoother for the dashed lines that simulate the gradual correction of the phase.

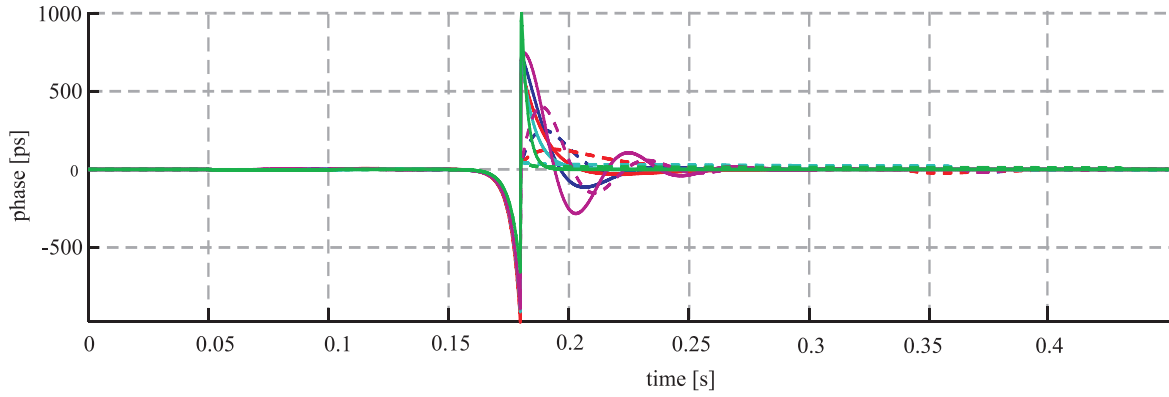
After completing the *first stage: phase error correction*, any synchronisation deterioration built up during *failure detection* and *switchover* are corrected for and stable syntonisation is regained – the WR PLL feedback system is back in business and the accuracy of synchronisation is comparable to that before the failure. The next step is to recover the slow PTP feedback system.

### Second stage: setpoint re-adjustment

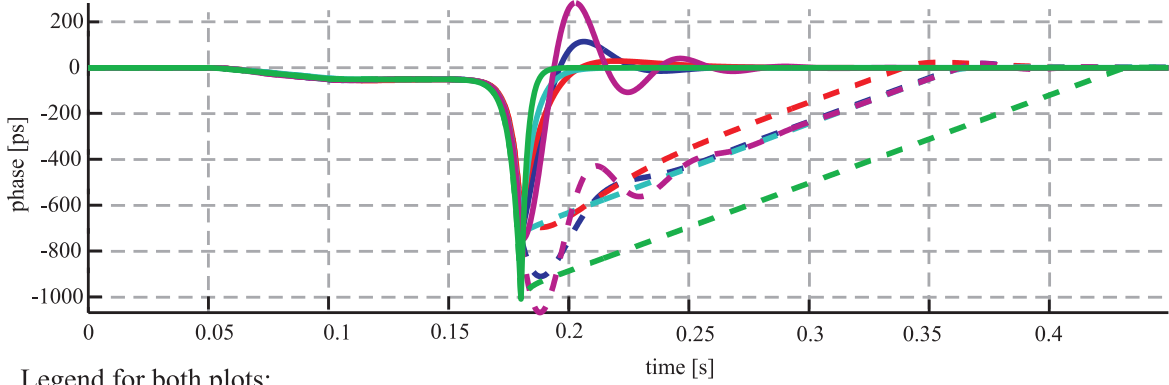
The offset from master that is measured by WR PTP at the active port is likely to be different from the offsets measured at the backup ports. This results from uncompensated asymmetries. During stable operation, the offset on the active port is considered correct while the offsets on backup ports are ignored, only their changes are evaluated. When the active port is changed, the new value of the offset from master must be used to re-adjust the setpoint.

The correction of the setpoint is applied with minimum increment which results in a phase

a) Phase error, which is the input to the PI controller, plotter for different PI parameters ( $k_i$ ,  $k_p$ )



b) Time Error (TE) between ideal reference and local clock plotted for different PI parameters ( $k_i$ ,  $k_p$ )



Legend for both plots:

- $k_i = 30$ ,  $k_p = 1100$  (current settings): phase jump: 849.8 [ps] (normal)
- -  $k_i = 30$ ,  $k_p = 1100$  (current settings): phase jump: 928.5 [ps] (correc)
- $k_i = 10$ ,  $k_p = 1100$ : phase jump: 674.5 [ps] (normal)
- -  $k_i = 10$ ,  $k_p = 1100$ : phase jump: 718.9 [ps] (correc)
- $k_i = 50$ ,  $k_p = 800$ : phase jump: 1029.6 [ps] (normal)
- -  $k_i = 50$ ,  $k_p = 800$ : phase jump: 1085.4 [ps] (correc)
- $k_i = 1$ ,  $k_p = 1500$ : phase jump: 718.9 [ps] (normal)
- -  $k_i = 1$ ,  $k_p = 1500$ : phase jump: 721.8 [ps] (correc)
- $k_i = 1$ ,  $k_p = 3000$ : phase jump: 1007.9 [ps] (normal)
- -  $k_i = 1$ ,  $k_p = 3000$ : phase jump: 1009.3 [ps] (correc)

Figure 5.8: Different parameters of the Proportional-Integral controller applied to the model of WR phase-locked loop with gradual correction of phase error.

shift slope of 3.726 ns/s (3.726ppb) and a constant overshoot regardless of the correction magnitude. The left plot in Figure 5.9 depict the simulation of phase correction for offset differences of 200 ps, 400 ps, 600 ps, 1000 ps using the described method. Each correction introduces phase error which is at the level of the standard deviation of the synchronisation in steady state, thus negligible. The right plot in Figure 5.9 provide a simulation of the 600ps phase correction for different PI parameters. Similarly, for all simulated PI parameters, the time error introduced by the phase correction is negligible.

The simulations clearly show that the main contributor to the time error during switchover is developed during the *failure detection* stage. Reliable and fast failure detection is investigated in the next subsection.

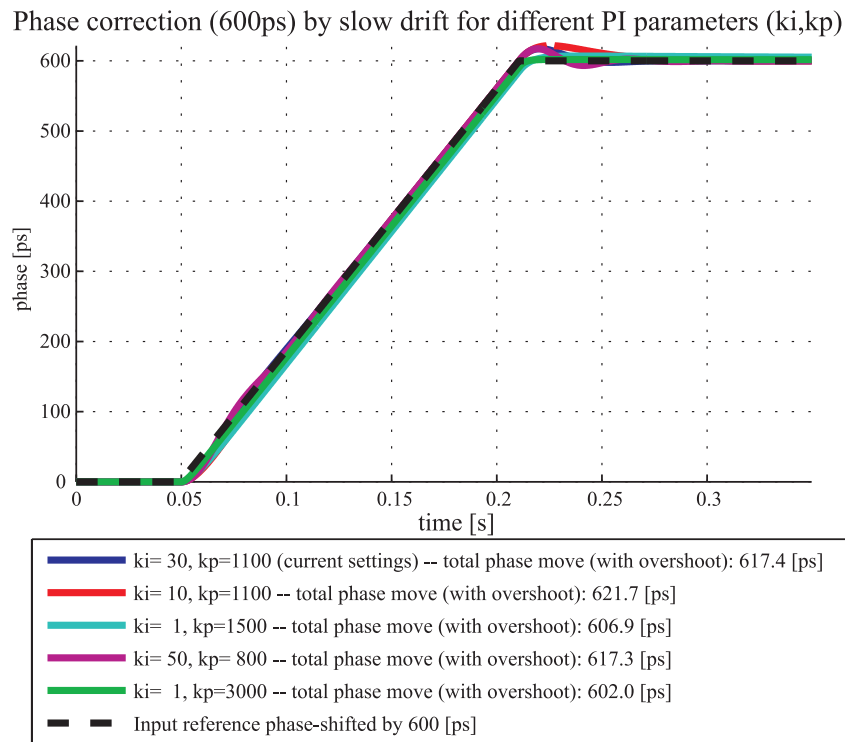
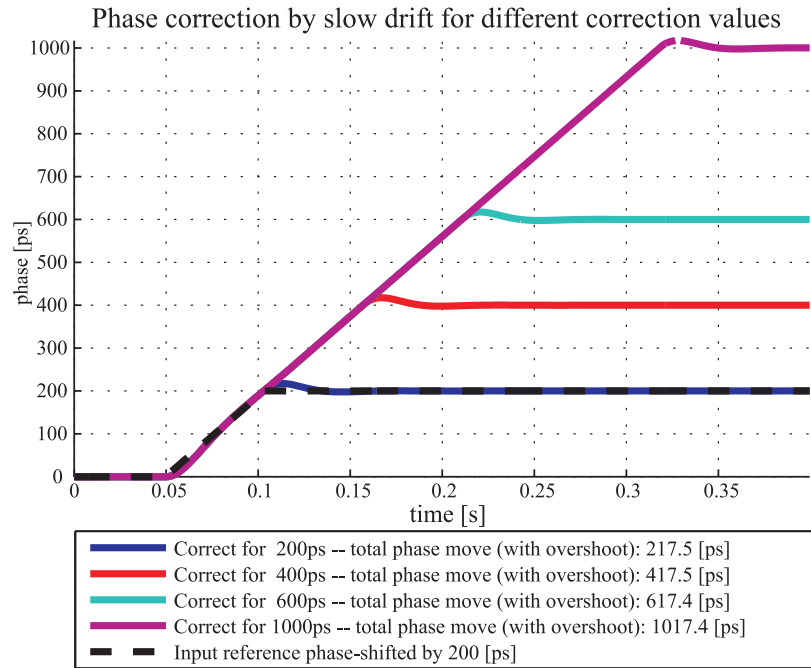


Figure 5.9: Simulation of setpoint re-adjustment and its contribution to time error.

### 5.2.4 Failure Detection

Reliable and fast detection of failure is the key and most critical factor in seamless switchover. Too long failure detection can result in time error exceeding the required accuracy before the switchover even takes place. Unreliable failure detection can either cause a complete failure of the system or result in a mistrust of the operator. When seamless switchover requires maintaining sub-ns accuracy of synchronisation, phase error during *failure detection*, ideally, should not accumulate to more than 100 ps. If the accumulated error exceeds 500 ps during *failure detection*, maintaining sub-ns accuracy during switchover in a large network might be hard.

Effective detection of failures needs to be done at low level and requires understanding the reasons and the symptoms of failures. In this thesis, failure detection is considered from the point of view of the WR PLL which can see the following symptoms of failure: signal loss, fast phase drift and, slow phase wander.

The first symptom, **signal loss**, can result from medium disconnection, mechanical stress, medium or hardware (e.g. laser diode) wear-off, or a hardware failure of a peer device. Its reliable detection is ensured by monitoring characters received by the Clock and Data Recovery (CDR) unit. A number of subsequent erroneous 8B10B characters is recognised as a link-down and triggers switchover. Tests proved that such a detection is a reliable indicator of failure. After the detection, synchronisation is not performed on the link until it is re-enabled by an operator.

The second symptom, **fast phase drift**, can result from a loss of synchronisation by an upstream devices with a poor holdover and its consequent fast drift, or a sudden (apparent) change in the length of medium. The latter appears to be the case while a cable is being unplugged and virtually extends its length. This results from integration of the signal which travels through the air from the partially unplugged fibre<sup>3</sup>. Consequently, the phase at the slave is shifted resulting in a temporary deterioration of accuracy (maximum theoretical error of 800 ps, maximum observed of 500ps). This means that at the instant when the link-down is detected, the inaccuracy is already in the order of few hundreds picoseconds. Consequently, the hardware detection of signal loss in the simple case of unplugging the cable is often not sufficiently fast to maintain the required accuracy. Reliable detection of fast phase drifts can be accomplished by majority voting when more than a single backup port is available. Such a detection allows to keep the phase error during the *failure detection* at the desired 100 ps level.

The third symptom, **slow phase wander**, can result from a byzantine fault, malfunction of PTP synchronisation while the physical layer is operational, or from a malfunction of the time/frequency source. Detection of such faults usually requires time-consuming exchange of information between devices. During this process, the phase error is likely to exceed the required accuracy. Detection of such failures is not attempted in this thesis. On the contrary, *slow phase wander* is corrected by the backup port control loop described in the next subsection.

---

<sup>3</sup>A good explanation of a similar effect that caused neutrinos to apparently travel faster than the speed of light can be found in [89].

### 5.2.5 Backup Port Control Loop

The measurement of the phase by the WR PLL and the offset by the WR PTP on the backup port is used to control neither time nor phase of the slave switch. This open-loop system must pretend to be a closed-loop system so that it can take over this role at any time. Additionally, the phase error measured by this pseudo-closed-loop system is essential in detecting failures (*fast drift*) of the active port. This section explains how the pseudo-closed-loop of the backup port works.

It is essentially an open-loop that tracks changes of the setpoint value based on the WR PTP measurement. This open-loop is re-initialized if the phase error measured by the WR PLL wanders off too much. When backup port is set-up, it updates the setpoint value as detailed in subsection 5.2.2: the initial value of setpoint,  $S_B(0)$ , is set to the measured average phase error (Equation 5.6). Then, the setpoint value is updated with the changes of the offset measured using the WR PTP (Equation 5.7). This open-loop is stable enough to be used for detection of fast-drifts in majority voting. However, it will eventually cause the setpoint to wander away without a bound due to thermal effects on the link, quantisation error, etc. In such case, when slow phase accumulation is detected at the backup port, the setpoint value is re-initialized to the measured (average) phase error. This way the pseudo-control loop is closed.

In summary, depending on their characteristics, the values of phase error measured at the backup port are interpreted two-fold:

1. Fast drifts, detected by comparing short-term and long-term moving averages, indicate possible link failure when more than one backup port exists and majority voting is enabled.
2. Slow phase accumulation, detected by comparing a number of phase error samples in and outside an acceptable range, indicates a need to re-set the value of the setpoint.

### 5.2.6 Reconfiguration in Cascaded Partially Redundant Topologies

Switchover based exclusively on fault detection of directly connected links, as described in the previous sections, makes the network design strict and rather unrealistic. Figure 5.10 (a) depicts such network in which each switch has a backup link. Notably, it requires multiple Grandmasters.

It is common in redundant networks to introduce non-redundant parts. The topology recommended by the strategy in Chapter 4 allows a single Grandmaster which requires support for reconfiguration in cascaded partially redundant topologies. This is depicted in Figure 5.10 (b) in which switches 2, 3 and 4 have a single link to the upstream Grandmaster switch. These switches are then redundantly connected to switches 5 and 6 at the lower layer. Switch 5 has a single backup port and switch 6 has two backup ports.

Having two backup ports, switch 6 can use majority voting to reliably detect problems of the links directly connected to it, as well as the links between upper-layer switches. This is because

any failure of a link between the Grandmaster and the below switches will likely result in a fast phase drift that is easily detectable through majority voting. However, switch 5 has only one backup port and cannot reliably trigger switchover. Instead, switch 5 must be informed by switch 2 that it lost its source of timing.

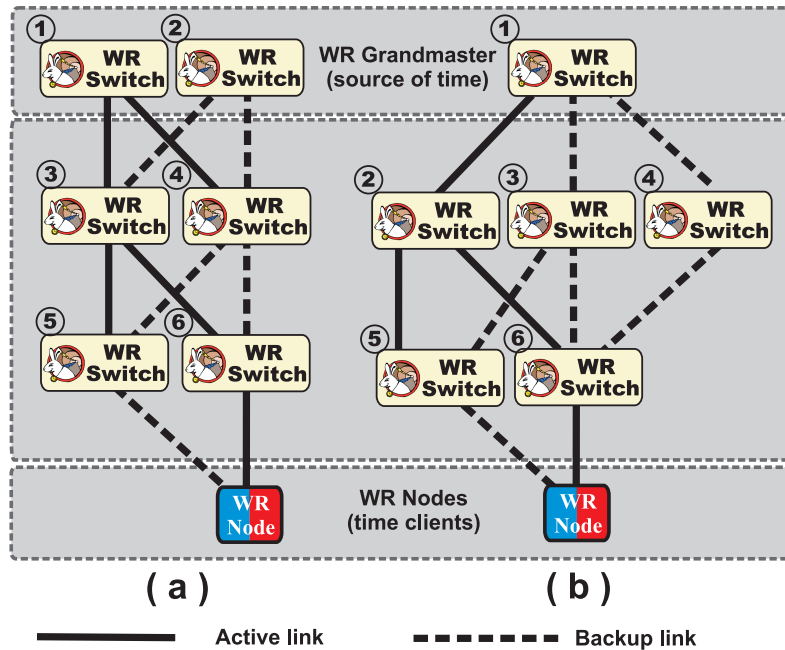


Figure 5.10: Fully redundant (a), and partially redundant (b) White Rabbit networks.

The PTP protocol allows propagation of this information. All the switches, including switch 2, send periodically PTP Announce messages on their downstream ports which are in the PTP master state. These messages include a clockClass field whose value is changed when the connection of the switch to the master is lost and the switch enters holdover. It is therefore used by switch 2 in Figure 5.10 to inform switch 5 about failure of the connection to the WR Grandmaster. The challenge is, however, to make sure that during this transfer of information, the synchronisation of switch 5 maintains sub-ns accuracy with respect to the WR Grandmaster.

In order to meet this requirement, two elements are needed: **(1) good enough holdover** at switch 2 and **(2) fast enough propagation of information** to the downstream switches. The holdover must provide sub-ns stability for the time needed to (a) send the PTP Announce message between switches and (b) trigger switchover at the receiving switch.

In the following sections, the holdover capabilities of the oscillator currently used in the switch are measured and compared with the minimum possible time needed for (a) & (b). Then necessary changes are proposed.

### 5.2.7 Holdover

After a successful detection of failure by a switch without backup port, the switch enters holdover and tries to maintain synchronisation. For a given oscillator, it is possible to predict the performance of such holdover and thus the time for which sub-ns synchronisation accuracy with the Grandmaster can be maintained. During this time, the downstream switches need to be informed about the problem and switchover to their backup ports, if available. This section measures the holdover capabilities of the oscillator currently used in the WR switch and studies if and how this holdover can be improved.

Holdover studies [90][91][92][93] and commercial [94][95] implementations focus on mechanisms to compensate effects of temperature and ageing on frequency stability. Sophisticated techniques to learn the characteristics of the oscillator are employed when the device is locked to a frequency standard, such as a caesium or rubidium clock. Once in holdover, based on the historical data and temperature detection, the created model can predict and compensate frequency drifts. This can provide even a 10-fold [92] reduction in OCXO time error variation during holdover with respect to the uncorrected oscillator.

The holdover during reconfiguration of WR network requires short-term stability at high level. In particular, sub-ns phase stability for up to a few seconds is needed. Slow processes, such as temperature changes and ageing, have little significance in such case and the mentioned holdover techniques are not applicable. Instead, random processes which cannot be modelled dominate. Thus, the performance is determined only by the quality and characteristics of the oscillator and requires finding the most optimal average control word to the VCXO.

The VCXO used in the WR switch [96] is characterized against a caesium standard to verify its stability and holdover performance. First, the frequency is locked to the standard and the average of the control word over 60 s of stable synchronisation is captured. Then, the average control word with different types of dithering (i.e. random, big-leaking [91], no dither) is applied and the phase error between the clock signal produced by VCXO and the standard is measured using a DDMTD phase detector.

The results of the measurements are used to plot the Allan Deviation, depicted in Figure 5.11. The plot indicates that a maximum of 100 ms can be spent in holdover to obtain reasonable switchover performance (i.e. sub-100 ps contribution of holdover to the switchover inaccuracy). This means that with the currently used oscillator the information about clockClass degradation needs to be delivered within 100 ms, or the oscillator needs to be changed a more expensive in order to gain more time. The next subsection investigates the speed of propagation of the clockClass information to the downstream switches as well as how and if it can be improved.

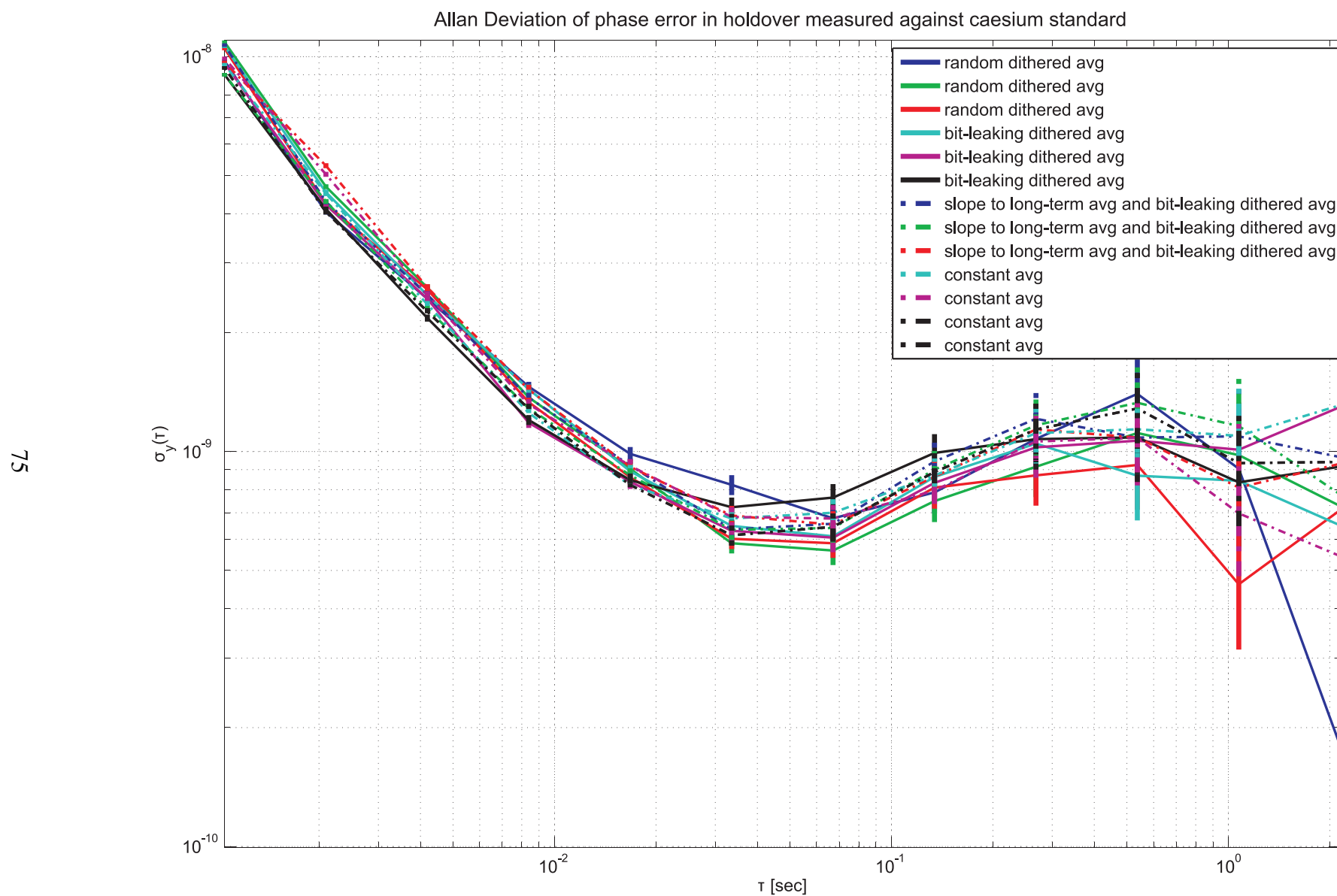


Figure 5.11: Allan Deviation of phase error between the WR switch clock signal in holdover (after learning) and a caesium frequency standard.



### 5.2.8 Propagation of clockClass

The value of clockClass is exchanged between PTP devices and informs about the connectivity of the master to the source of time. The previous subsection suggests that in the WR network this information might need to be exchanged within 100 ms in case such connectivity is lost.

In the Default PTP Profiles, the value of clockClass is sent in PTP Announce messages every 2 s. The definition of periodical transmission of PTP Announce messages in the PTP standard is based on statistics (9.5.8 of [19]) and indicates that the mean value of the interval is required to be correct while 10% of intervals can be outside of  $\pm 30\%$  of the specified value. This means that occasional transmission of PTP Announce messages immediately after entering holdover does not break the compatibility with the standard. Furthermore, the PTP Announce messages are not used in synchronisation but only in establishing spanning tree, so their unexpected transmission does not affect the accuracy of synchronisation. Thus, in case of WR network reconfiguration, clockClass degradation can be transmitted in PTP Announce message as soon as possible without breaking the compatibility with the PTP standard or jeopardizing the PTP synchronisation.

The required delivery time of the PTP Announce message can be considered a real-time task. As described in subsection 1.2.2, real-time tasks are implemented in WR switch in the Field Programmable Gate Array (FPGA). Exchange of the PTP Announce messages with degraded clockClass is not an exception and should be handled in the FPGA as well. By doing so, the notification about holdover can be delivered to a downstream switch or node in an estimated time of 1 ms. During 1 ms the phase error is expected to deviate around 10 ps ensuring sub-ns synchronisation, even in a cascade of non-redundantly connected switches.

### 5.2.9 PTP Support and Configuration for Seamless Switchover

PTP supports neither multi-path nor hot-spare backup nor seamless redundancy. The distribution of time from the source (Grandmaster) to the nodes over a physical network of switches is organised into a single logic spanning tree, even if the network is redundant. This section describes how to augment this behaviour to the needs of WR and provide multi-path time distribution using PTP.

Firstly, it is important to note that PTP aids backup links with its peer-delay mechanism. In switches that implement this mechanism, link delays on all the ports are measured at any time, thus the offset from the master can be calculated intermediately after switchover between redundant paths occurs. Although peer-delay mechanism is mostly used with Transparent Clocks that are not applicable in WR, it is permissible to use this mechanism on switches that are Boundary Clocks, such as the WR switch. The peer-delay mechanism in such case measures link delays on redundant ports and allows to calculate the offset from the master on these ports, at any time. The standard PTP devices put their redundant ports in the PTP\_PASSIVE state and do not per-

form such a redundant measurement of the offset. In WR, these ports can be considered backup ports.

It is the Best Master Clock Algorithm (BMCA) that creates a synchronisation spanning tree in PTP networks. The BMCA prevents redundant links from synchronisation by recommending their state to be PTP\_PASSIVE. The PTP\_PASSIVE state is essentially recommended on the ports which would be in PTP\_SLAVE state if the port that is currently in the PTP\_SLAVE state were not functional. This, in other words, means that the BMCA marks potential backup slave ports as PTP\_PASSIVE. Such a way of recognising backup ports in a redundant PTP network works for a limited number of topologies. In particular, it covers the simple topology examples demonstrated in Figure 5.12 where the ports in the PTP\_PASSIVE state indeed mark the backup paths. This means that, in theory, BMCA can be used in WR networks to recognise backup ports.

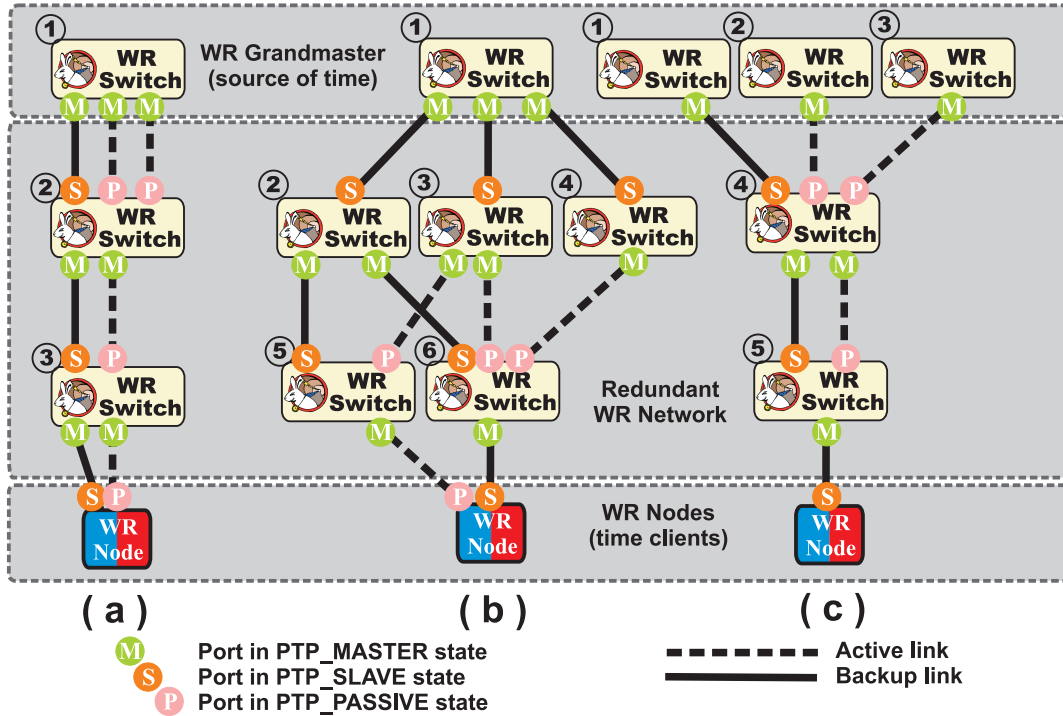


Figure 5.12: Port PTP states in basic redundant topologies considered in this thesis.

In practice, automatic configuration of critical control networks can be dangerous, unpredictable in transients, and it is better avoided. Therefore, WR supports hand-configuration<sup>4</sup> of the synchronisation spanning tree by assigning a static PTP state to each port (i.e. PTP\_MASTER and PTP\_SLAVE). This hand-configuration is extended in the context of this thesis by adding per-port priorities to support flexible backup ports configuration.

<sup>4</sup>Although hand-configuration of PTP port states is not allowed by the current version of PTP [19], it is very likely to be supported in its next revision and this solution will become standard-compatible.

The extension allows more than one port in a WR switch to be assigned the role of a PTP\_SLAVE. The slave port with the best port priority is the active slave. If two ports are assigned the same priority, the first port to be connected is the active slave. The calculation of link delay and offset from the master is performed on all the slave ports. However, the switch is synchronized and syntonised using the active slave port. All the other connected ports in the PTP\_SLAVE state act as hot-spare backups. When the active port fails, the slave port which is next in the priority queue takes over the role of an active slave.

If a port is connected and it has a priority that is higher than the priority of the currently active slave port, the following steps are taken:

1. The port with the best priority is added as a backup port.
2. Stable synchronisation on backup port is established.
3. The port with the higher priority becomes the active slave port.

A generic algorithm for adding a new slave port is depicted in Figure 5.13. When a port not in PTP\_SLAVE state is recommended or configured to be PTP\_SLAVE, it is verified whether

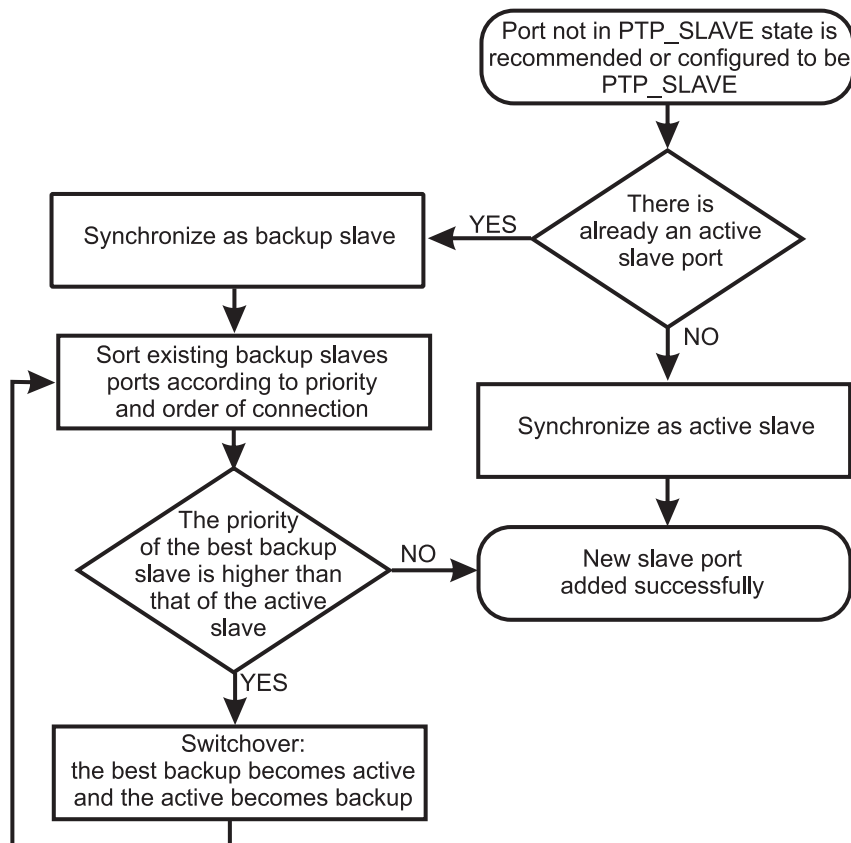


Figure 5.13: A generic algorithm for adding a new PTP slave port.

there already exists synchronisation through another port in the PTP\_SLAVE state (the active slave port). If there is no existing active slave, synchronisation on the new port in the PTP\_SLAVE state is established and the port becomes an active slave. If there already exists

an active slave, the port is added as a backup slave regardless of its priority (backup slave setup described in subsection 5.2.2). Once the backup slave is operational (*Multi-phase tracking* in Figure 5.6), the new list of backup ports is sorted according to the priorities and order of connection. The priority of the backup port which is first on the list is compared to the currently active port. If the active slave has worse priority than the best priority backup port, switchover occurs. The backup becomes active and vice versa. The list of backup ports is sorted to allocate a proper place for the new backup port according to its priority, considering it as a newly connected port.

The implementation of this mechanism, along with the extension of the WR PLL and FPGA-based support of PTP are described in the following subsection.

## 5.3 Implementation

The mechanisms implemented in the context of this thesis involve extensions of the WR PLL implementation (SoftPLL), extensions of the WR PTP daemon (PPSi) and development of a PTP Support Unit (PSU) VHDL module. All three components are depicted in Figure 5.14 and described in the following subsections.

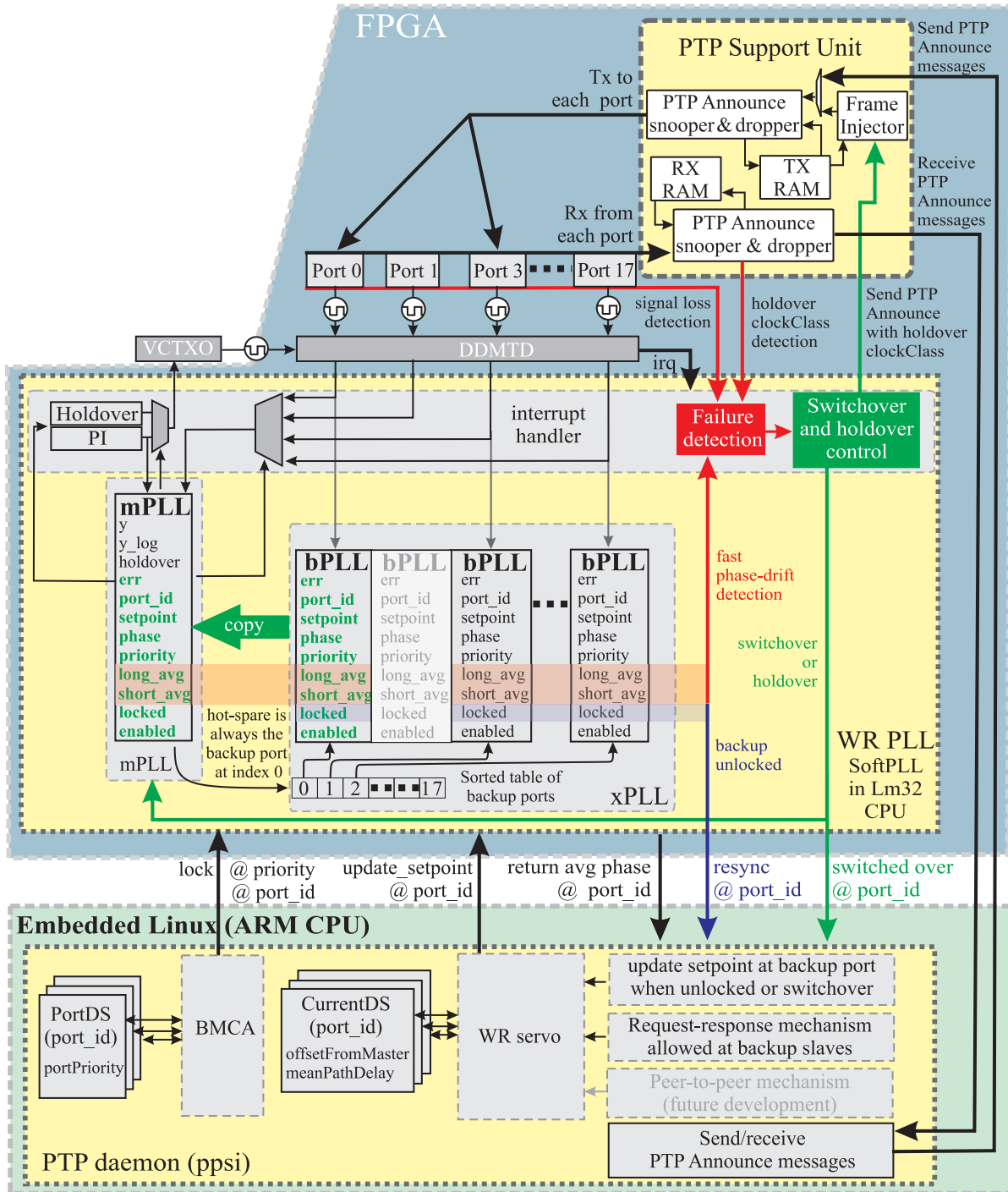


Figure 5.14: Implementation of the seamless switchover.

### 5.3.1 Multi-Channel WR PLL

The WR PLL described in section 5.1 has been extended to support multi-channel operation, fault detection based on phase error and majority voting, as well as seamless switchover.

The new multi-channel WR PLL is connected to the clock signals recovered from the received data at each port, as depicted in Figure 5.14. Each of these input channels is represented by a Backup PLL (bPLL) structure. This structure is required for the operation of a backup port and stores the information necessary to take over the role of the mPLL, i.e. phase error (err), setpoint, source port number (port\_id) and boolean values indicating whether the port is locked and enabled. The structure also keeps track of short- and long-term moving averages of phase errors. Such averages are also kept for the mPLL. They are used to detect link failure in switches connected with multiple backup ports. The priority stored in bPLL is used to decide on the role (backup, active) of the ports and their order in the table of backup ports. The backup ports are sorted by priority and order of connection. Index zero always indicates the backup port to be used when the active port fails.

The multi-channel WR PLL supports three sources that can trigger immediate switchover between the active port and the backup port with index zero:

1. Signal loss – indicates that a sequence of invalid characters was received on a port. The detection takes place in each port.
2. Fast phase-drift – indicates that the difference between short-term moving average (short\_avg) and long-term moving average (long\_avg) of the phase error (err) has exceeded a threshold on majority of the backup ports, provided there is more than one backup port.
3. Holdover clockClass – indicates that a PTP Announce message with degraded clockClass at the active slave port has been received from the current PTP master. The detection takes place in the PSU.

If one of the above sources is TRUE and a backup port exists, switchover is executed in the following steps:

1. Interrupts in embedded CPU are disabled.
2. An intermediate holdover is applied to the mPLL as well as to the hPLL by supplying the previously calculated control word (y).
3. The reference of the hPLL is switched to the new one and any history of phase error is cleared (this is sufficient in the case of the hPLL).
4. The relevant data from the bPLL is copied to the mPLL structure and the bPLL is disabled.
5. A flag is set to notify PPSi that switchover occurred.
6. The intermediate holdover is exited and interrupts are enabled.

If failure is detected but there is no backup port, holdover is entered by applying the average of the control word (y) calculated over the past 60 seconds. The applied value is exposed to dithering [91] to reflect its fractional part.

### 5.3.2 PTP Daemon (PPSi)

The changes in the PTP daemon, depicted in Figure 5.14 with dash-framed rectangles, concern the operation of the WR servo, BMCA, port state configuration and expanding the Data Set appropriately.

The extended WR servo supports operation of the backup port control loop and the proper update of the setpoint after the switchover, described in subsections 5.2.5 and 5.2.3. The servo allows updates of the setpoint in the same way on all the slave ports, regardless of whether they are active or backup ports. Additionally, a need for resynchronisation of the backup port can be indicated by the WR PLL (blue arrow in Figure 5.14). This results in resetting the setpoint to the value provided by the WR PLL. Similarly, after switchover, the WR servo is provided by the WR PLL with the measured phase to update the setpoint.

The BMCA is extended to allow the PTP\_SLAVE state on multiple ports and to distinguish between the active and the backup ports during their setup.

The port Data Sets are modified to support synchronisation on each slave port (both active and backup) in exactly the same way. Additionally, the slave priority is added for each port to allow configuration of the active and backup ports.

### 5.3.3 PTP Support Unit

The PTP Support Unit (PSU) is developed to enable ultra-fast ( $\sim 1\text{ ms}$ ) and standard-compatible notification between WR devices about holdover. It is a VHDL module that is placed on the transmission and reception paths of all the frames that are sent and received by the WR switch's CPU, including the PTP messages.

The developed PSU module detects and remembers all the received and transmitted PTP Announce messages, as described in details in Appendix G. It uses this information in two cases:

1. **Request to send PTP Announce with holdover clockClass:** when WR PLL enters holdover, it notifies PSU. PSU sends the PTP Announce message stored in the *TX RAM* with the degraded clockClass. When a PTP Announce message with the same sequence is later sent by the PTP daemon, this message is dropped by PSU to preserve the sequence ID order of the messages and minimise impact on the average interval time between message transmissions.
2. **Reception of PTP Announce with holdover clockClass:** each received PTP Announce message is stored and compared with the previously received message. When clockClass deterioration is detected, the PSU module notifies the WR PLL. If backup port is available, the WR PLL performs switchover. Otherwise, it enters holdover and requests PSU to send a PTP Announce message with holdover clockClass.

## 5.4 Limitations of the Used Methods, Alternative Solutions

The developed methods provide flexibility in terms of the level of redundancy but restrict the number of applicable topologies. The allowed time distribution topology is a tree-like meshed arrangement where the alternate synchronisation paths can be unambiguously configured a priori, the number of hops must be kept to minimal, and ring topology is not allowed.

Full redundancy requires multiple Grandmasters. It is an arrangement which is supported by the proposed methods provided the Grandmasters are synchronized with sub-ns accuracy. The method of synchronizing the redundant Grandmasters is outside of the scope of this thesis.

The presented methods do not provide byzantine fault tolerance or detection of malicious masters. Detection of such faults usually requires time in which the accuracy deterioration would greatly exceed 1ns.

There is no existing alternative solution to provide the required level of synchronisation and reliability. The most accurate highly reliable solution, High-availability Seamless Redundancy (HSR) [69], provides 1  $\mu$ s synchronisation. A non-PTP-based solution that provides sufficient accuracy of synchronisation, developed by AGH University of Science and Technology [97], is designed for point-to-point synchronisation and provides no support for redundancy; it is therefore not applicable.

The proposed methods can be extended to waver some of the limitations. For example, a more stable oscillator can be used as a second (internal) reference when detecting faults through majority voting. There is also an ongoing work to extend the presented methods to ring topology and support for HSR.

## 5.5 Usefulness and Applications of Proposed Methods

Applications of WR include synchronisation in accelerator control systems [98][22], distributed acquisition and measurement systems [13][29], cosmic ray and neutrinos detectors [14][23][86][24], and time transfer between national time laboratories [99][26][27]. WR is also considered for such applications as Smart Grid.

In many of these applications, high availability of synchronisation is either critical or highly desired. For example, the cosmic ray detectors in China (LHAASO [14]), Siberia (HiSCORE [23]), and the neutrinos detector at bed of Mediterranean Sea (KM3Net [24]) require a large WR network with limited access. The scale of the network increases the probability of failure, the limited access increases the time and cost of repair. Support for redundancy of synchronisation can greatly increase the availability of these detectors.



## 5.6 Measurements and Tests

This section describes tests of the methods developed in the context of this thesis to provide synchronisation resilience through seamless redundancy. These methods are tested in a number of network scenarios depicted in Figure 5.15.

The tests are performed similarly for all the scenarios depicted in Figure 5.15. In each scenario, a downstream switch is connected redundantly to the Grandmaster switch. After establishing a stable redundant connection, the currently active link is made to fail (red cross) and the backup link takes over. Two types of failures are tested:

- **Physical disconnection of fibre link** which might be a result of human error, mechanical destruction of the link, or fatal failure of a WR switch. The test is performed by manually disconnecting the fibre optic from the upstream or downstream port.
- **Slow signal degradation of a fibre link** which might be a result of mechanical stress, laser ageing, as well as loose or contaminated connector. The test is performed using an Optical Attenuator [100] to introduce signal degradation on the link.

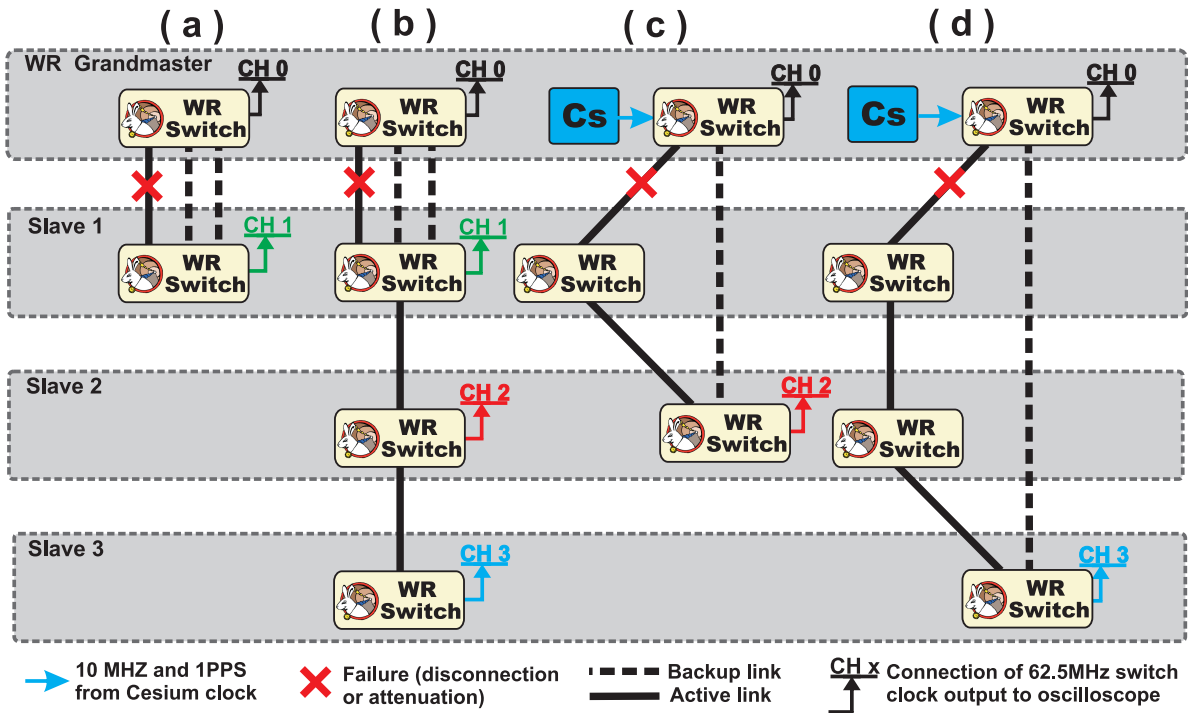


Figure 5.15: Test scenarios for which the developed mechanisms were tested and their performance measured.

The synchronisation performance is evaluated by measuring the time error between the Grandmaster switch and the downstream switch(es). The increase of time error during switchover, compared to the time error during stable operation, represents the performance deterioration. It is referred to as a *phase jump during switchover*.

Two methods are used to estimate the *phase jump during switchover*. One is based on an oscilloscope measurement, the other is based on the phase error information provided by the WR PLL. Both methods are detailed below.

**Oscilloscope measurement:** A LeCroy Wavepro 7300A 3GHz oscilloscope is used to measure the skew between the clock signal of the Grandmaster switch and that of the downstream switch. A reference maximum time error ( $MTE_{ref}$ ) measurement is performed over a period of 5 minutes during stable operation of the network. The same measurement is performed during switchover ( $MTE_{swover}$ ). The increase in the maximum time error is interpreted as a deterioration of the performance introduced by switchover:

$$\phi_{jump\_osc} = abs(MTE_{swover} - MTE_{ref}) \quad (5.10)$$

**SoftPLL measurement:** The values of phase errors measured by the SoftPLL are analysed to evaluate the phase jump. In particular, the maximum phase error on the backup port before the switchover ( $\phi_{errB}$ ) and the maximum phase error on the newly active port after the switchover ( $\phi_{errA}$ ) are interpreted as the range of the phase jump value:

$$\phi_{jump\_pll} = [max(abs(\phi_{errB})) , max(abs(\phi_{errA}))] \quad (5.11)$$

Both values are shown in the test results. The one with a smaller value is called *SoftPLL estimate min*, and the one with a greater value is called *SoftPLL estimate max*.

The tests in each of the scenarios in Figure 5.15 and their results are described in the following subsection.

### 5.6.1 Direct Redundant Connection (scenario a)

A direct redundant connection between two WR switches is tested using a single and double backup link. It is the scenario (a) in Figure 5.15. The failure of an active link is caused through fibre disconnection and attenuation. The following tests are performed:

- **Tests 1 to 3:** switchover with single backup, fibre is physically disconnected.
- **Tests 4 to 6:** switchover with double backup, fibre is physically disconnected.
- **Tests 7 & 8:** switchover with single backup, fibre is gradually attenuated until it fails.
- **Tests 9 & 10:** switchover with double backup, fibre is gradually attenuated until it fails.

Figure 5.16 shows *phase jump during switchover* for each test in this scenario while detailed data collected during tests is included in Appendix H. In all the tests, the *phase jump during switchover* is below 500 ps. It actually is below 100 ps for all but the tests with a single and physically disconnected backup link.

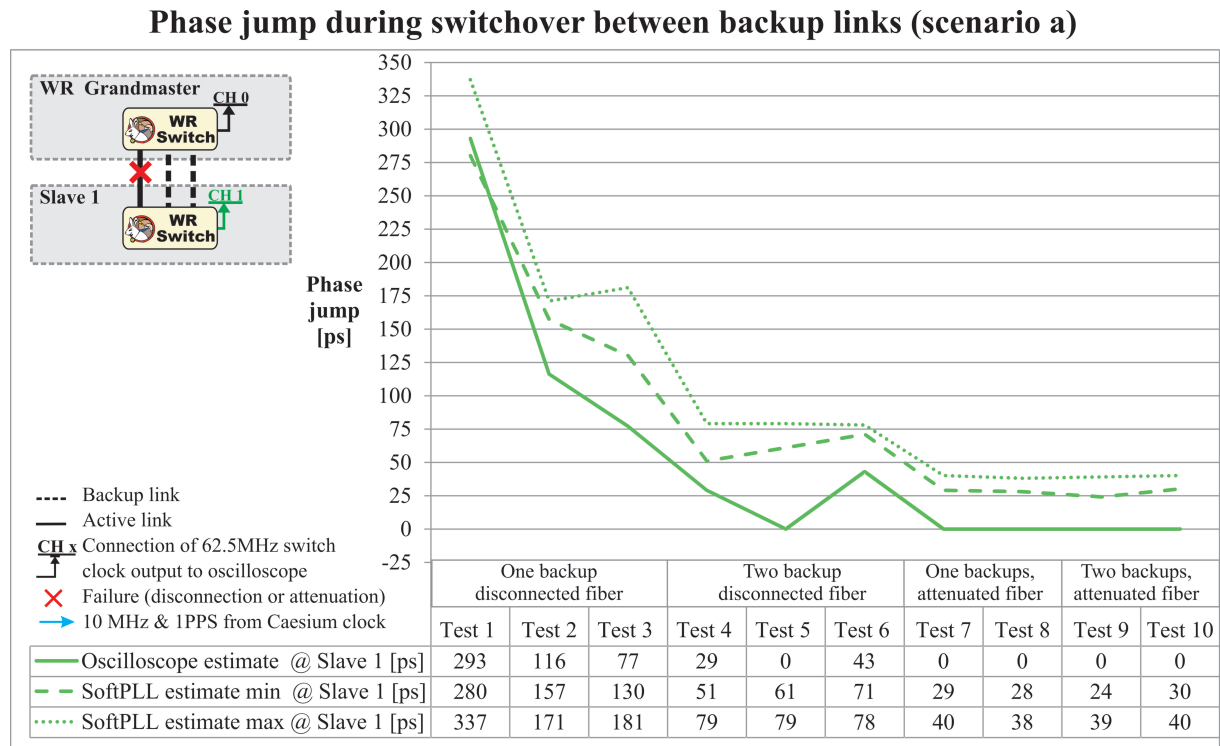


Figure 5.16: Phase jump measured during switchover between direct redundant connections.

## 5.6.2 Direct Redundant Connection in Cascade of Switches (scenario b)

A direct redundant connection between two WR switches is tested in a cascade of switches. It is the scenario (b) in Figure 5.15. The purpose of this test is to observe how phase jump during switchover affects the downstream synchronisation. The following tests are performed:

- **Tests 1 to 3:** switchover with a single backup, fibre is physically disconnected.
- **Tests 4 to 6:** switchover with double backup, fibre is physically disconnected.
- **Tests 7 & 8:** switchover with double backup, fibre is gradually attenuated until it fails.

Figure 5.17 shows the *phase jump during switchover* for all three cascaded switches. Its estimation using an oscilloscope-based measurement is performed for all the switches in the cascade. A WR PLL-based measurement is performed for switch 1 only. Appendix H includes detailed test results.

For most of the tests, the *phase jump during switchover* is below 500 ps. Similarly to scenario (a), only the tests with a single and physically disconnected backup link exceed 100 ps. It is worth noting that the phase jump does not escalate down the cascade, except for the case where it is substantial (Test 2) and exceeds 500 ps at the third switch in the cascade.

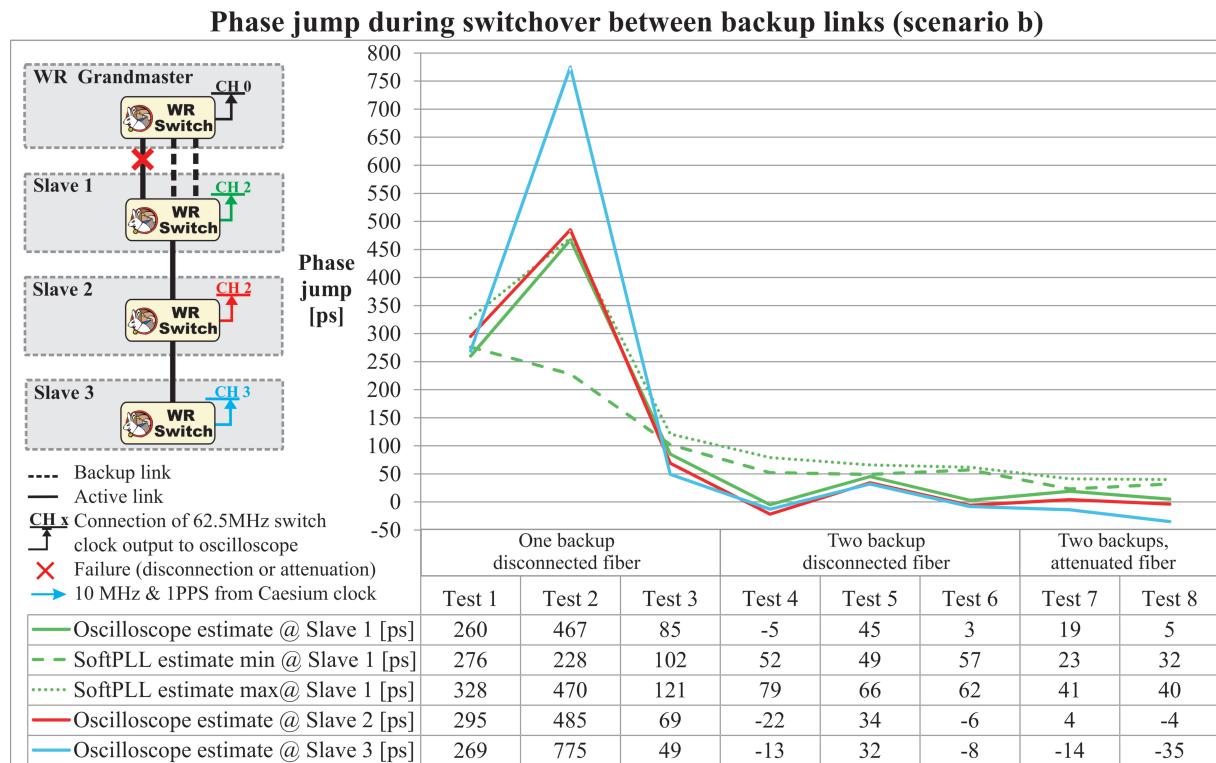


Figure 5.17: Phase jump measured during switchover between direct redundant connections in cascade of switches.

### 5.6.3 Indirect Redundant Connection (scenario c)

An indirect redundant connection between WR switches is tested where WR switch 2 is synchronized to the Grandmaster through WR switch 1. It is the scenario (c) in Figure 5.15. The link between the Grandmaster and switch 1 fails causing switch 1 to enter holdover and notify switch 2 about it. The notification is done by sending PTP Announce messages with degraded clockClass. As soon as switch 2 receives this information, switchover to the backup port occurs. The following tests are performed:

- **Tests 1 to 7:** switchover with a single backup, fibre is physically disconnected.
- **Tests 8 to 10:** switchover with a single backup, fibre is gradually attenuated until it fails.

Figure 5.18 shows *phase jump during switchover* for each test in this scenario while detailed data collected during tests is included in Appendix H. In all the tests, the *phase jump during switchover* is below 500 ps. It is below 100 ps for the tests where fibre is attenuated. Analysis of the data presented in Appendix H show that the initial phase jump due to disconnection of switch 1 is not magnified but followed in the downstream switch 2 (the one on which the measurement is made). The *phase jump during switchover* on switch 2 is a sum on two factors: (1) the phase jump of the upstream switch 1; and (2) the short-term performance of holdover in the upstream switch 1 while the information about holdover is delivered to switch 2. Although the performance in the case of disconnection is worse than in scenario (a), it stays within a 500 ps range. The results from tests with an attenuator are comparable with the results from scenario (a) which can be attributed to efficient (no phase jump) detection of link failure in switch 1.

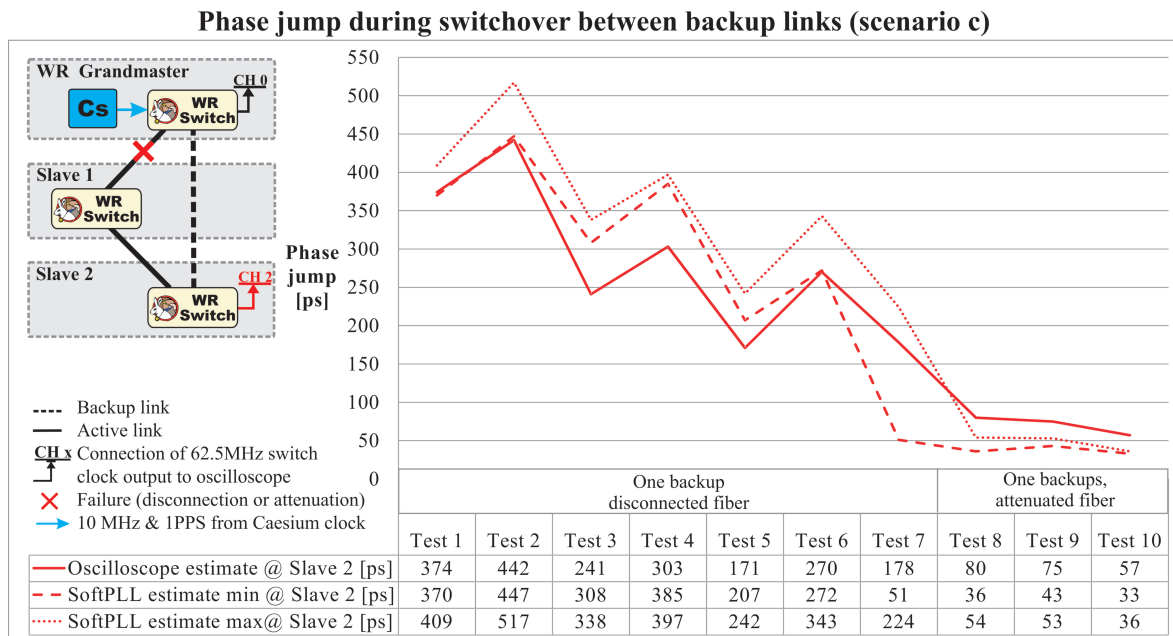


Figure 5.18: Phase jump measured during switchover between indirect redundant connection and a backup direct connection.

## 5.6.4 Indirect Redundant Connection in Cascade of Switches (scenario d)

An indirect redundant connection in a cascade of switches is tested. It is the scenario (d) in Figure 5.15. In this scenario WR switch 3 is connected to the Grandmaster through switch 1 and switch 2. The link between the Grandmaster and switch 1 fails causing switch 1 to enter holdover and notify downstream switches 2 and 3 about it. The notification is done by sending a PTP Announce message with degraded clockClass. As soon as switch 3 receives this information, switchover to the backup port occurs. The following tests are performed:

- **Tests 1 to 3:** switchover with a single backup, fibre is physically disconnected. Switch 1 and switch 2 enter holdover.
- **Tests 4 to 8:** switchover with a single backup, fibre is physically disconnected. Switch 1 enters holdover, switch 2 stays synchronized to switch 1.
- **Tests 9 to 11:** switchover with a single backup, fibre is gradually attenuated until it fails. Switch 1 and switch 2 enter holdover.

Figure 5.19 shows *phase jump during switchover* for each test in this scenario while detailed data collected during tests is included in Appendix H. For most of the tests, the *phase jump during switchover* is below or around 500 ps. It is around 100 ps for the tests where the fibre is attenuated. The presented results show that the worst-case phase jump may exceed 500 ps and reach similar level to the worst-case in the scenario (b).

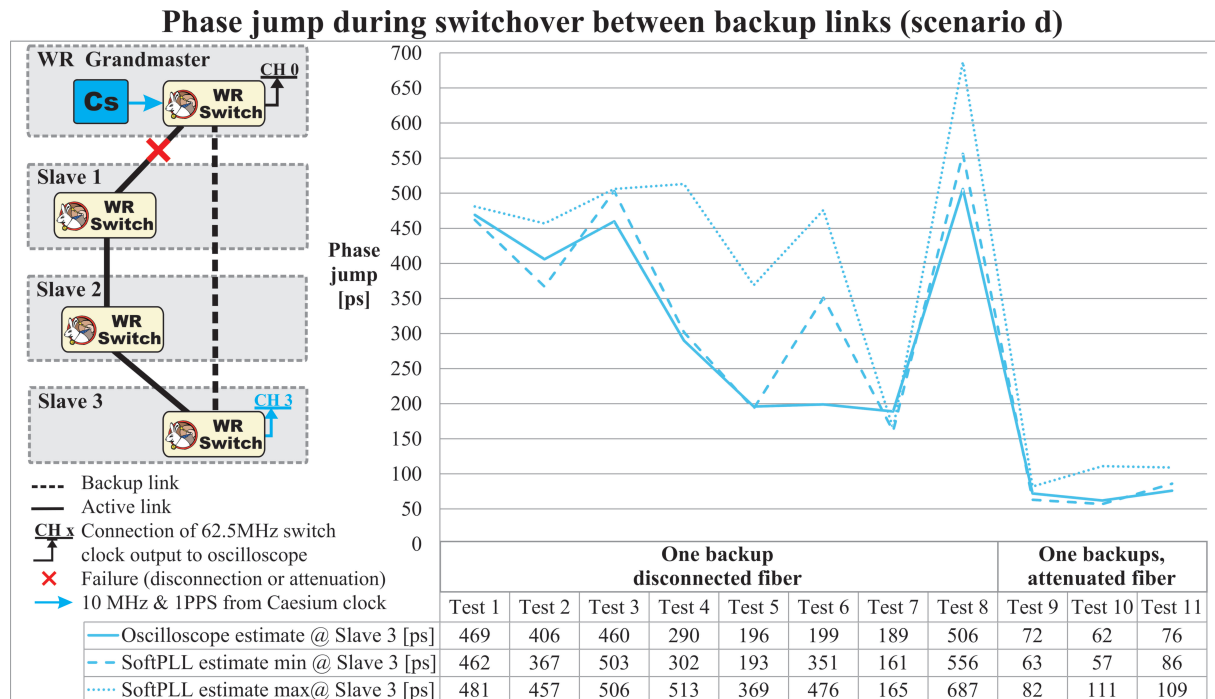


Figure 5.19: Phase jump measured during switchover between an indirect redundant connection and a backup direct connection in a cascade of switches.

## 5.6.5 Test and Measurement Summary

The test results show that the developed mechanisms ensure sub-ns *phase jump during switchover* for all the considered network scenarios, even a reasonably advanced scenario (d). The maximum values of the phase jump from each of the test scenarios are compared in Figure 5.20.

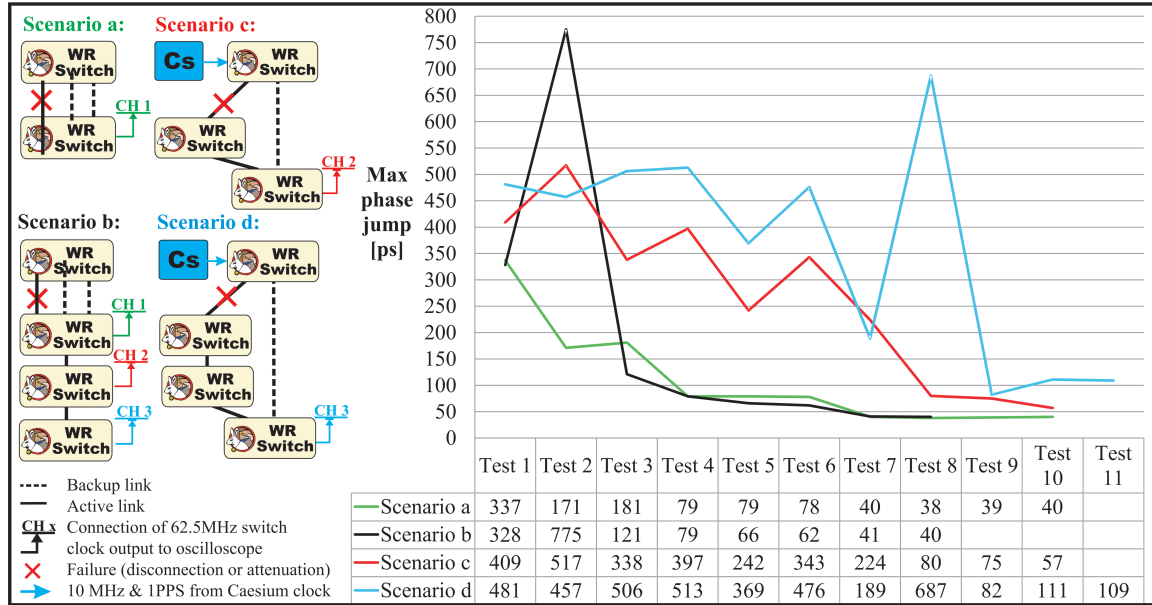


Figure 5.20: The worst-case phase jumps during switchover for each test in each scenario.

Most of the *phase jumps during switchover* in the performed tests do not exceed 500ps, and the worst case measured is below 800 ps. 800 ps is the worst-case phase error that can occur during *failure detection*. It is confirmed by the tests that the greatest contributor to the performance during switchover is the phase error accumulated during *failure detection*. This phase error can be dramatically reduced by using more than one backup port. When two backup ports are used in the test, the *phase jumps during switchover* is consistently at the 100 ps level and easily meets the budget recommended in the Proposed Strategy (section 4.9).

The budget that can be allocated to *phase jump during switchover* depends on the size of the network and the level of network calibration (see section 4.8). In small and carefully calibrated WR networks, sub-ns accuracy of synchronisation can be maintained with a worst-case *phase jump during switchover* of 800 ps. Large WR networks will likely require more phase error budget for inaccuracies that occur during normal operation and thus require *phase jump during switchover* much below 800 ps. Ensuring sub-ns accuracy of synchronisation in such networks calls for double backup links.

The tests have shown that the developed methods support seamless redundancy when sub-ns accuracy of synchronisation is required. This represents a 1000-fold improvement when compared to any other existing synchronisation method that supports redundancy (see section 3.6). Notably, such performance is achieved without replacing a rather cheap VCXO on the switch.



## Chapter 6

# Methods to Support Seamless Redundancy and Determinism for Data

This chapter describes the enhancements of data transmission through a WR network which have been developed to provide the two specialized services: determinism and reliability of data. These enhancements follow the strategy proposed in Chapter 4 with the goal of fulfilling the CERN requirements provided in Table 1.1. In particular, the important requirements for this chapter include: the network size of 2000 nodes and maximum 10 km between the nodes, the frame size of 1200-6000 bytes, the upper-bound latency through the WR network and a single switch of less than  $500\ \mu s$ <sup>1</sup> and  $10\ \mu s$ , respectively. A network topology that follows the strategy proposed in Chapter 4 is depicted in Figure 6.1 and consists of 288 switches that

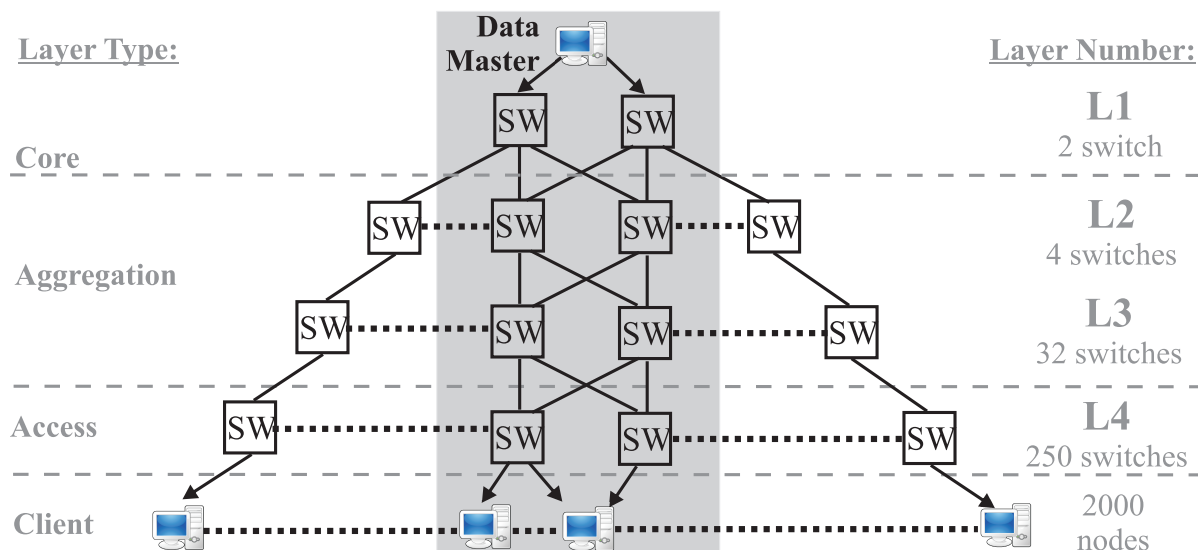


Figure 6.1: Redundant mesh topology of the White Rabbit network.

<sup>1</sup>The latency requirement of 1 ms in Table 1.1 is translated into a maximum network latency of  $500\ \mu s$ , see section 4.8.2.



connect 2000 nodes in a redundant mesh network topology. Such topology is suitable for the methods that provide synchronisation resilience, described in Chapter 5, and the methods developed in this chapter. The enhancements described in this chapter aim at providing seamless redundancy and guaranteed upper-bound latency, per the requirements in Table 1.1. None of the standards and technologies available for Bridged Local Area Networks (LANs) can fulfil CERN's requirements.

This chapter is organised as follows. It first introduces the solutions available in the IEEE 802.1Q standard to support network redundancy and some level of determinism. Their performance is analysed to quantify the required improvement. Then, the chapter describes the methods that have been developed in the context of this thesis to speed up the reconfiguration of a redundant network from milliseconds to microseconds and decrease the latency through a switch few-folds, from  $\sim 12 \mu s$  to  $\sim 3 \mu s$ . Finally, a detailed description of their implementation and tests are provided.

## 6.1 Background

### 6.1.1 Network Redundancy

Bridged LANs are best effort by design and the mechanisms provided by their defining standards do not support seamless redundancy. The two alternative standard protocols that can be used to configure redundant LANs are the Rapid Spanning Tree Protocol (RSTP)<sup>2</sup> and Shortest Path Bridging (SPB). When a network element fails or a new one is added to the network, the protocol that is used, RSTP or SPB, reconfigures the network to ensure connectivity and prevent logic loops. During such reconfiguration, by design, frame loss occurs.

The operation of RSTP and SPB is explained for a redundant network topology suggested in the strategy and depicted in Figure 6.1. This network is well-suited for the traffic expected in the CERN control and timing network: vertical transmission of data from a single node at the top of the network, the data master, to many nodes at the bottom of the network, the clients. The network consists of three layers: core, aggregation and access. The data master is redundantly connected to the core switches that are at the top of the hierarchy. All the client nodes are connected to the access switches that are at the bottom of the hierarchy. These connections are redundant or single. No node is attached to the aggregation switches that connect the core to the access switches. In further discussions two types of traffic are considered:

- downstream broadcast from the data master to all the nodes within a Virtual LAN (VLAN)
- upstream multicast<sup>3</sup> from a client node to the data master.

The two protocols provided by the IEEE 802.1Q standard are described, focusing on the network topology and the traffic of interest.

#### Rapid Spanning Tree Protocol (RSTP)

The RSTP establishes a single bi-directional spanning tree among all the switches. The distance-vector algorithm underlying the RSTP creates a logic spanning tree, such as depicted with blue dash lines in Figure 6.2 a. This figure shows the shaded sub-set of the considered redundant network in Figure 6.1. The RSTP logic spanning tree is rooted at the switch that is elected by the algorithm, marked in blue in Figure 6.2 a. The protocol blocks forwarding on downstream ports of redundant connections to prevent loops. The resulting single tree is used to communicate all types of traffic (unicast, multicast, broadcast) between all the nodes. Each switch has information only about its neighbours and the root switch. When the network changes, a new configuration of the network logic tree needs to be calculated and updated by all the switches in the network. During such update, ports downstream from the root are likely to be temporarily blocked in order to prevent loops. This may take seconds, during which data is lost.

---

<sup>2</sup>In particular, Spanning Tree Protocol (STP), Rapid Spanning Tree Protocol (RSTP) and Multiple Spanning Tree Protocol (MSTP)

<sup>3</sup>Redundant ports of the data master are configured into a Multicast group.

### a) Rapid Spanning Tree Protocol (RSTP)

### b) Shortest Path Bridging (SPB)

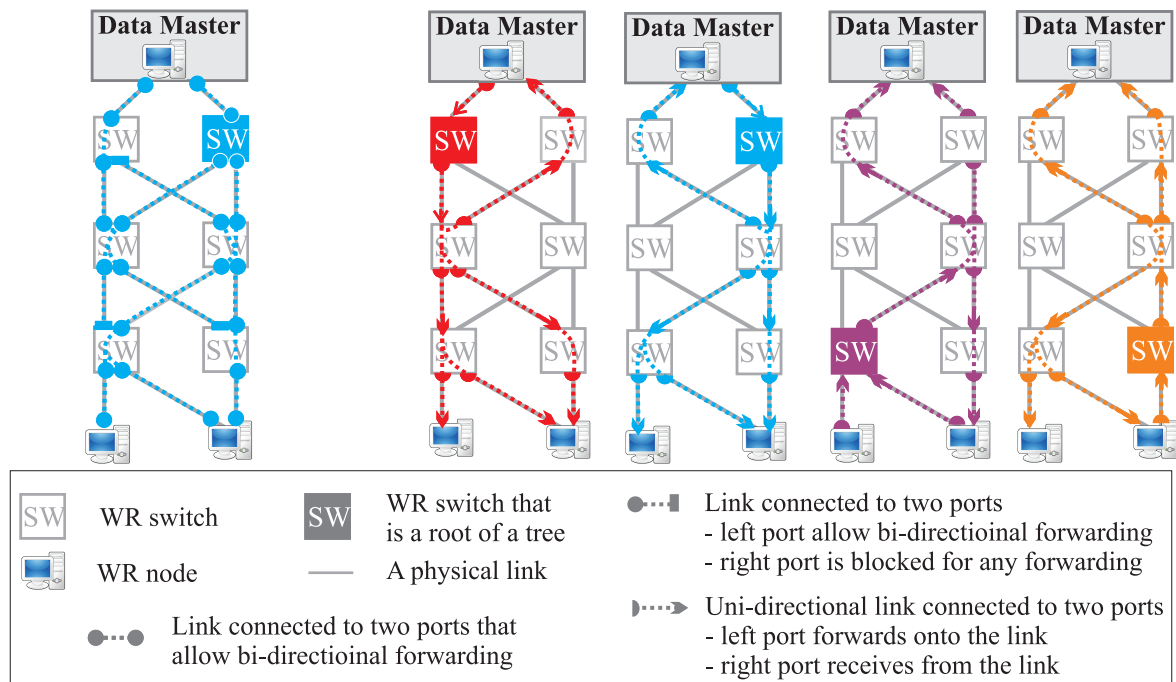


Figure 6.2: Logic trees in the Rapid Spanning Tree Protocol and the Shortest Path Bridging.

### Shortest Path Bridging (SPB)

The SPB establishes a dedicated unidirectional spanning tree for each switch to which a node is connected. In the case of the considered network topology, these are either access or core switches. For the shaded sub-set of the network depicted in Figure 6.1, the SPB establishes four trees, as depicted in Figure 6.2 b. Each tree is identified by a dedicated VLAN ID (VID)<sup>4</sup> or a Media Access Control (MAC) address, depending on which mode of protocol is used, SPB-VID or SPB-MAC. Regardless of the SPB mode, each tree provides shortest paths from the root to all the other switches. If more than one shortest path tree can be established, the root switch can send frames through different trees, but one frame can be sent only through a single tree. Unlike switches running RSTP, all the switches running SPB have information about all the possible paths in the network. Each switch can calculate independently the result of a physical topology change. Therefore, the network reconfiguration is much faster, in the sub-50 ms range. However, 50 ms accounts for many frames lost during the reconfiguration.

Neither RSTP nor SPB allow fast enough reconfiguration or multipath transmission of frames through a redundant network that would sufficiently maximise the probability of data delivery.

<sup>4</sup>A VID that identifies shortest path tree is called shortest path VID and it is referred to as SPVID in the standard and literature.

### 6.1.2 Determinism

The IEEE 802.1Q standard provides a tool to achieve a certain level of determinism of the data transmitted over a Bridged LAN; it is explained and quantified in this subsection for the Gigabit Ethernet that is used in White Rabbit.

IEEE 802.1Q adds to the Ethernet header a tag with priority that can be assigned per-frame. The frame with the highest priority is given forwarding precedence over these with lower priorities. The smaller the number of nodes that can send the highest priority traffic, the more deterministic is the latency of this traffic through the network. A network that shows the most deterministic behaviour for the highest priority frames:

- It has a single source of the highest priority frames.
- It consists of switches that implement strict priority policy and process frames in negligible time.

The worst-case latency through a switch implementing the IEEE 802.1Q standard depends on the size of both, the highest priority frames and all the other frames. Such a switch is called store-and-forward. It stores each frame entirely, checks its validity using Cyclic Redundancy Check (CRC), and only if it is uncorrupted, forwards the frame. The maximum latency that results from such switch behaviour, neglecting processing time, depends on:

- the maximum size of frames that have the highest priority, and
- the maximum size of frames that have lower priorities – if such a frame is being sent when the frame with the highest priority is scheduled for transmission, the transmission of the lower priority frame needs to be completed before the highest priority frame is sent.

The maximum and minimum latency through a store-and-forward switch, assuming a full range of frame sizes for all priorities and not considering jumbo frames, can be easily calculated. See the column *store-and-forward* of Table 6.1.

	<b>Store-and-forward</b> [ $\mu s$ ]	<b>Cut-through</b> [ $\mu s$ ]
Maximum latency	24.672 (see Note 1)	12.544 (see Note 2)
Minimum latency	0.512 (see Note 3)	0.208 (see Note 4)
Variation of latency (max-min)	24.160	12.336
Note 1: Reception plus transmission time of maximum size Ethernet frame. Note 2: Reception time of the Ethernet header with VLAN tag and transmission time of maximum size Ethernet frame. Note 3: Reception time of minimum size Ethernet frame. Note 4: Transmission time of the Ethernet header with VLAN tag.		

Table 6.1: Latency through different types of switches for Gigabit Ethernet.

A common-practice in decreasing latency through a switch is a silently-accepted violation of the standard: transmission of the frame as soon as its destination is known without checking its CRC, a so-called "cut-through". With such an implementation, the latency through a switch does not depend on the size of the frame that is being forwarded, thus the latency is substantially improved, compared to a store-and-forward switch. However, the worst-case latency still depends on the size of frames which are not the highest priority. See the column *cut-through* of Table 6.1.

Thus, regardless of the type of the switch, a frame with the highest priority that is forwarded to a port currently transmitting another frame, has to wait until this frame's transmission is completed. In the worst case, this is an entire maximum-size frame and an Inter-Frame Gap (IFG) which is a total of 1542 bytes or  $12.336 \mu s$ . Unless the highest priority traffic is the only one in the network, even in cutting-edge cut-through switches, optimised for extremely-low and constant processing latency, the maximum latency depends on the maximum size of the non-critical traffic and can be larger than  $10 \mu s$ .

## 6.2 Support for Seamless Redundancy and Determinism

### 6.2.1 Problem Statement

The seamless redundancy and determinism, needed by CERN, require the following functionalities that cannot be provided either by IEEE 802.1Q or other Ethernet-based network:

- Transfer of frames through all optimal paths in order to increase the probability of their delivery.
- Failover time of network reconfiguration between alternative paths that is shorter than the transmission of (N-1) Forward Error Correction (FEC) parity frames; it is recommended 5 – 10  $\mu s$  by the strategy in Chapter 4.
- Deterministic latency that can be predicted with sufficient precision so that the lowest-latency path can be set as active.
- Upper-bound latency for critical traffic through a single switch below 10  $\mu s$ .
- Support for multi-redundant topology in which more than two alternative paths are available.

In the applications of White Rabbit, the network is shared between critical traffic (control messages) and non-critical best-effort traffic, such as management and diagnostic frames. The above functionalities must work for the selected critical traffic in the presence of a best-effort traffic.

The White Rabbit switch has been enhanced by the author to provide the above functionalities, as described in the following subsections.

## 6.2.2 Architecture

The developed architecture minimises loss of frames during reconfiguration and maximises the determinism of a selected traffic by introducing the following four key ideas: unidirectional pseudo-multipath logic trees, pre-configured alternative paths, low-level failure handling and absolute priority for the selected traffic.

The unidirectional pseudo-multipath logic trees are defined by VIDs, as depicted with a simple network in Figure 6.3. These trees are rooted at the nodes or at the edge switches and

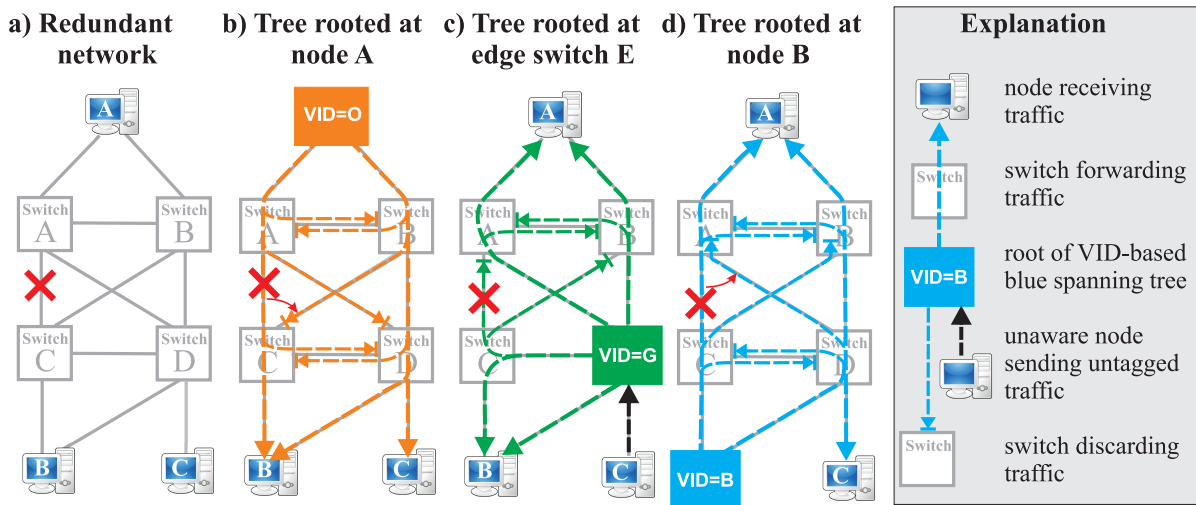


Figure 6.3: Pseudo-multipath broadcast tree rooted at a switch or a node.

span optimal paths between the root and all the edge switches in the network. Each switch in the network:

- forwards frames away from the root at ports belonging to optimal paths defined by VID
- accepts frames from the root at one of the ports belonging to the optimal paths defined by VID.

The port on which the frames from the root are accepted is referred to as an *active port*. The port on which the frames from the root are discarded is referred to as a *backup port*.

The *backup ports* constitute the pre-configured alternative paths from the root. As soon as the active port fails, a backup port becomes active and starts accepting the frames. The knowledge of the alternative path in advance by the switch's software is not sufficient. The time it takes to switchover between the active and the backup ports is critical and requires low level support for failure detection and the activation of a new configuration.

Furthermore, the active and backup ports are chosen according to their estimated latency from the root of the tree. This is to avoid discarding frames in the backup port before the same frames are successfully received on the active port. For this solution to work, the latency from the root over a number of hops (switches) must be calculable with an uncertainty at the

microseconds-level. To achieve that, absolute priority is given to the forwarding of the dedicated critical frames.

The complete solution that has been developed includes the following components detailed in separate subsections of this section:

1. Fast switchover between pre-configured active and backup ports.
2. Network topology applicable for pseudo-multipath spanning trees.
3. Loss-less network configuration update.
4. Deterministic frame transmission through the network.

### **6.2.3 Fast Switchover Between Pre-Configured Active and Backup Ports**

The underlying idea is fairly simple: the active and backup ports are determined at the same time. As soon as failure of the active link is detected, switchover occurs and the backup port is activated. The switchover is implemented in gateway (see switch architecture in Figure 1.5) to make the operation as fast as possible. However, the required switchover time of a few microseconds reaches physical limits, such as speed of light or memory access, which cannot be overcome by low-level implementation alone. This subsection details the challenges faced and the developed solutions: hot-spare backup port and consistent failure detection, activation time of backup configuration, unicast learning on redundant paths and variation of transmission latency.

#### **Hot-spare backup port and consistent failure detection**

The first challenge concerns the need to use hot-spare link configuration in order to achieve the required microsecond-level switchover time and minimise the number of frames lost during the switchover. A Gigabit link with the maximum length required, 10 km, introduces around 50  $\mu s$  of transmission time and can "buffer" 4 full-size 1522 bytes Ethernet frames or 9 FEC frames of 600 bytes. These frames are lost when the link fails. Therefore, hot-spare backup links are required. On such links, the frames are forwarded on transmission but discarded on reception. Therefore a fast enough switchover can receive from the backup link the frames that are lost on the one that was active but has failed. However, such a configuration can cause temporary loops or frame loss in case when the information about link state is inconsistent between the interconnected devices. These risks need to be mitigated.

The described risks result from an inconsistent failure detection that manifests in two ways: thrashing and unidirectional failure. Unidirectional failure is a situation, either transient or static, when only one of the ports of the link detects the failure while the other considers the same link functional. This results in topology inconsistency that might cause data loss, loops and congestion. On the other hand, thrashing manifests in short periods of failure detection and proper operation. The failure is detected and, while being handled by the protocol, it ceases.



Therefore recovery of the original configuration is attempted. Repeated occurrence of such a behaviour can cause great disruption in network operation and thus significant data loss.

The proposed and implemented method to ensure a consistent view of link state between adjacent switches and to prevent thrashing is as follows. The port that detects link-down physically and instantly powers off its transmitter. Consequently, the other port consistently detects failure. This is implemented in gateway, which makes it fast and results in a reliable detection algorithm. The information is propagated between ports with the speed of light in fibre. In this way, a consistent view of the link state by two interconnected switches is ensured and thrashing prevented. Before traffic is allowed on any port, this port is first tested for stable operation; the port's transmitter is powered on but the traffic is blocked. Only after stable operation without thrashing is confirmed, the switch is reconfigured to allow traffic on the port.

### **Backup configuration activation time**

The second challenge concerns the time of activation of the backup configuration which needs to take place in a few microseconds. Even if the information about backup configuration is pre-computed locally on each switch, its application (download of the new forwarding rules) takes time, especially if backup configuration is defined for each MAC address. As an example, [36] mentions that in reconfiguration of MAC forwarding tables for SPB-MAC, the table download actually dominates the computation time. Moreover, providing a backup for each MAC entry is resource-inefficient. Therefore, in this thesis a VID-based backup configuration is proposed. In this approach, for each mask that defines a port associated with a given VID, a number of backup masks is provided. This is a trade-off between flexibility and resource-efficiency. The reasonably small number of Virtual Local Area Networks (VLANs) makes it possible to pre-load (store in gateway) many alternate configurations. Moreover, an atomic update of the entire VLAN table through bank-swapping is possible. This allows to activate backup configuration in a single clock cycle, i.e. 16ns, which is sufficiently fast.

### **Unicast learning**

The third challenge concerns the perturbation of unicast traffic after switchover. This disruption needs to be minimised to avoid unnecessary congestion and prevent a temporary increase of latency during the switchover. In such solutions as RSTP or SPB-VID, the unicast MAC addresses of nodes attached to the network are learnt from the source MAC addresses of the forwarded frames. Since a single frame is forwarded from a single port, these addresses are associated, through learning, only with this single port. After switchover occurs, the unicast addresses must be updated by removing the entries associated with the failed port and learning the new port association. This takes time. During that time, the traffic is temporarily perturbed and increased. Some frames are sent to the failed port and lost. Some frames are broadcast unnecessarily. In order to provide correct unicast forwarding instantly after switchover, this thesis proposes to allow multi-port unicast entries in the MAC Table<sup>5</sup>. The multi-port unicast entries

---

<sup>5</sup>See Appendix A and Figure A.4 for the switch architecture.

are learnt and automatically installed from the accepted frames on the active port as well as from the discarded frames on the backup ports. This guarantees zero-time for backup installation of unicast traffic as the backup ports are already installed when the switchover occurs. For this to work, congruency of the unidirectional spanning trees between nodes is important. This means that a frame sent from node A to node B over an unidirectional pseudo-multipath spanning tree rooted at node A needs to traverse the same path when sent from node B to node A over an unidirectional pseudo-multipath spanning tree rooted at B.

### Variation of transmission latency

The last challenge concerns the difference in arrival-time of the same frame received on the active port and the backup port. This difference needs to be known and guaranteed at microseconds-level. Otherwise the frame loss during switchover<sup>6</sup> cannot be recovered by the FEC, even if the switchover happens instantly. Figure 6.4 illustrates this problem in the case when a control message is sent in 4 FEC frames. Any 2 of these 4 frames can be lost and the schema still allows recovery of the original control message.

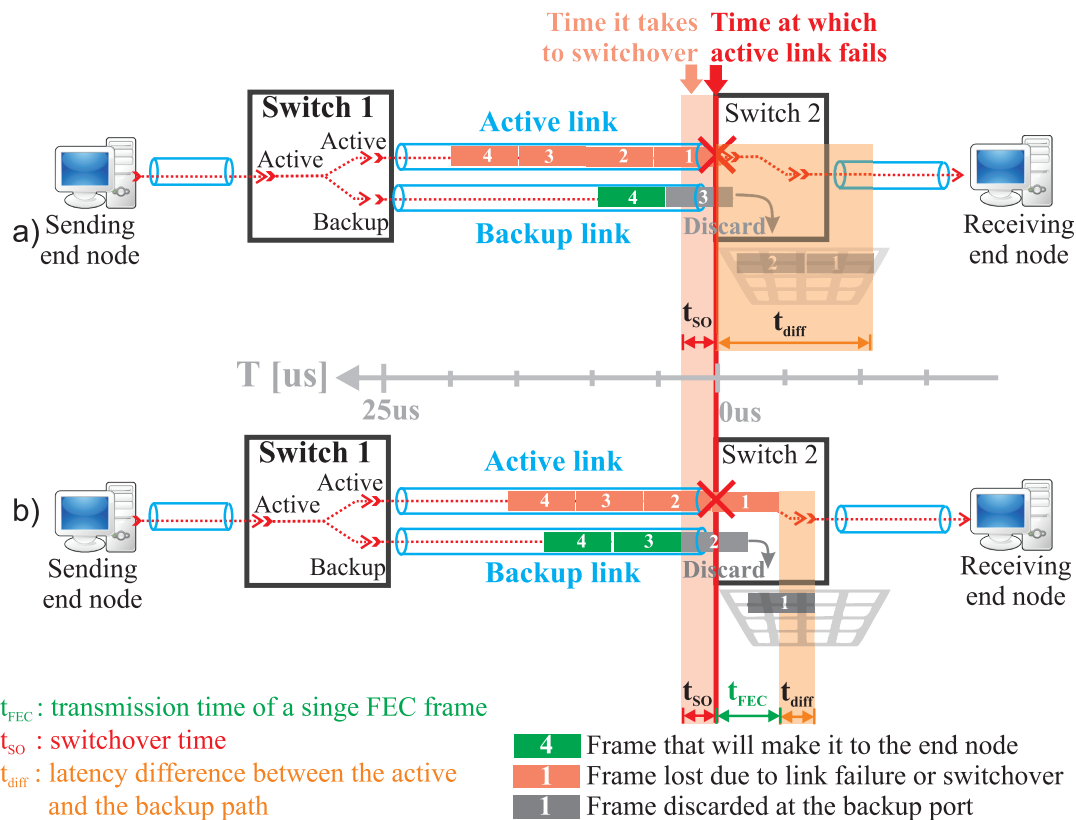


Figure 6.4: Frame loss during switchover due to variation of transmission latency.

In Figure 6.4 a, the variation of latency is greater than the time of transmission of 2 FEC frames in a burst. If such a large latency difference occurs at the time of failure, unrecoverable frame loss is unavoidable. The figure illustrates a situation where the latency of the frames

<sup>6</sup>The original message cannot be recovered from FEC frames when the number of lost frames is greater than the number of FEC parity frames (see section 4.6).

through the active path from the sender is significantly larger than the latency through the backup path. This results in the frames arriving at the backup port of *Switch 2* much sooner than at the active port. Such a situation might happen when a maximum-size frame is transmitted at the active port of *Switch 1*, towards *Switch 2*, when the burst of critical frames arrived at *Switch 1*. In particular, the latency difference introduced by such a maximum-size frame is around 12  $\mu s$ . Thus, when the active link fails, the third FEC frame is already being discarded by the receiving *Switch 2*. Only the forth frame will be forwarded by this switch and received by the end node. This is too few FEC frames to recover the original control message.

In Figure 6.4 b, the total time of the switchover and the latency variation between the active and the backup port is small enough so that the frame loss during switchover is recoverable. The figure shows the corner-case scenario in which the original control message can be still recovered. The failure happens when the first frame is nearly fully received at the active port. The switchover is completed just before the third frame is to be received at the backup port. Thus, the third and the forth frames are forwarded from the backup port to the receiving node, while the first and the second frames are discarded. The two forwarded frames are enough to recover the original control message.

The corner-case scenario from Figure 6.4 b can be used to evaluate the precision with which the latency over the active and backup paths must be known to mitigate the problem presented in Figure 6.4 a. The maximum difference between the latencies,  $latency_{MAX\_diff}$ , that allows recovery of the original control message can be calculated as follows:

$$latency_{MAX\_diff} = (N - 1) \cdot (s_F + s_{IFG}) \cdot t_{byte} - t_{SO} \quad (6.1)$$

where

$s_F$  [bytes] is the size of a single FEC frame

$s_{IFG}$  [bytes] is the size of Inter-Frame Gap

$t_{SO}$  [s] is the time it takes to switchover

$N$  is the number of FEC parity frames

$t_{byte}$  [s/bytes] is the transmission time of 1 byte

The actual value of the latency difference between two ports of a switch,  $latency_{port\_diff}$ , can be estimated for a given pair of paths. Such an estimation is based on the knowledge of the latency through a switch and the Precision Time Protocol (PTP) measurement of link delay. The worst-case difference of the frame latency between its transmission by a node and its reception at the active and the backup port is:

$$latency_{port\_diff} = \max(latency_{port\_active}) - \min(latency_{port\_backup}) \quad (6.2)$$

where

$latency_{port\_active}$  is the transmission time between the sending node and the active port

$latency_{port\_backup}$  is the transmission time between the sending node and the backup port

The estimated latency difference ( $latency_{port\_diff}$ ) must be smaller than the maximum allowed difference ( $latency_{MAX\_diff}$ ) to prevent loss of too many frames during switchover:

$$latency_{port\_diff} < latency_{MAX\_diff} \quad (6.3)$$

The active port should be the port with a smaller latency. If the estimated latency difference between the active port and backup port is greater than the estimation error, it is easy to fulfil the above condition. However, the different paths from the transmitter to the active port and the backup port are likely to have similar latencies. The corner-case is when the latencies estimated on the active and the backup port are approximately equal. In such case, the precision of the latency estimation is essential. If the estimated latency difference ( $latency_{port\_diff}$ ) has an error that exceeds the maximum allowed latency difference ( $latency_{MAX\_diff}$ ), an unacceptable frame loss might occur.

Knowing the maximum allowed error of the latency difference estimation, the required determinism of the network can be calculated. Let's assume that the latency of the active and backup port is estimated to be equal, within the error:

$$\begin{aligned} latency_{port\_active} &= latency_{estimate} \pm \delta \\ latency_{port\_backup} &= latency_{estimate} \pm \delta \end{aligned} \quad (6.4)$$

The maximum estimated latency difference between the backup and active port is:

$$\begin{aligned} latency_{port\_diff} &= \max(latency_{estimate} \pm \delta) - \min(latency_{estimate} \pm \delta) \\ latency_{port\_diff} &= 2 \cdot \delta \end{aligned} \quad (6.5)$$

This gives a peak-to-peak jitter of latency,  $2 \cdot \delta$ , that should be smaller than the maximum allowed latency difference,  $latency_{MAX\_diff}$ . Therefore using Equation 6.1:

$$2 \cdot \delta < (N - 1) \cdot (s_F + s_{IFG}) \cdot t_{byte} - t_{SO} \quad (6.6)$$

The above equation gives the relation between the parameters of FEC and the required determinism of the WR network. The value  $2 \cdot \delta$  is the peak-to-peak uncertainty of latency estimation through disjoint paths.

As an example, the peak-to-peak jitter of latency is calculated for the scenario depicted in Figure 6.4. In this example, 4 FEC frames of 600 bytes include 2 parity frames, the IFG is of the minimum size, and the switchover time is  $t_{SO} = 2.5 \mu s$ . The required determinism in of the network is quantified:

$$2 \cdot \delta < 2.5 \mu s \quad (6.7)$$

In the particular simple network of Figure 6.4, it means that the latency through *Switch 1* must be known with a precision of  $\pm 1.25 \mu s$ . Subsection 6.2.6 discusses deterministic data forwarding and explains how to meet this requirement. The following subsection explains which network topologies are optimal for the mechanisms developed in this thesis, including minimisation of latency jitter.

## 6.2.4 Applicable Network Topologies and Pseudo-Multipath Spanning Tree

The commonly available protocols that configure redundant mesh networks are designed to handle sophisticated physical topologies. This is unnecessary in carefully designed networks, such as the WR control and timing network. The possibility of handling sophisticated network topologies requires special precautions and slows down the reconfiguration process. However, in the case of the CERN control and timing network, its topology is carefully engineered and does not change dynamically. Moreover, if a newly connected switch introduces an undesired change of topology, reconnection can be requested. Therefore, the mechanisms presented in this thesis are designed to work with selected mesh topologies. This subsection details the redundant topology type that is required for the optimal operation of the mechanisms described in this thesis. It then explains the algorithm to create a pseudo-multipath logic spanning tree in the redundant physical network. Such a spanning tree provides configuration of active and backup ports.

Figure 6.5 presents mesh network that provides double redundancy ( $R = 2$ ) applicable for the mechanisms in this thesis. In this topology, the nodes are connected only to the access and

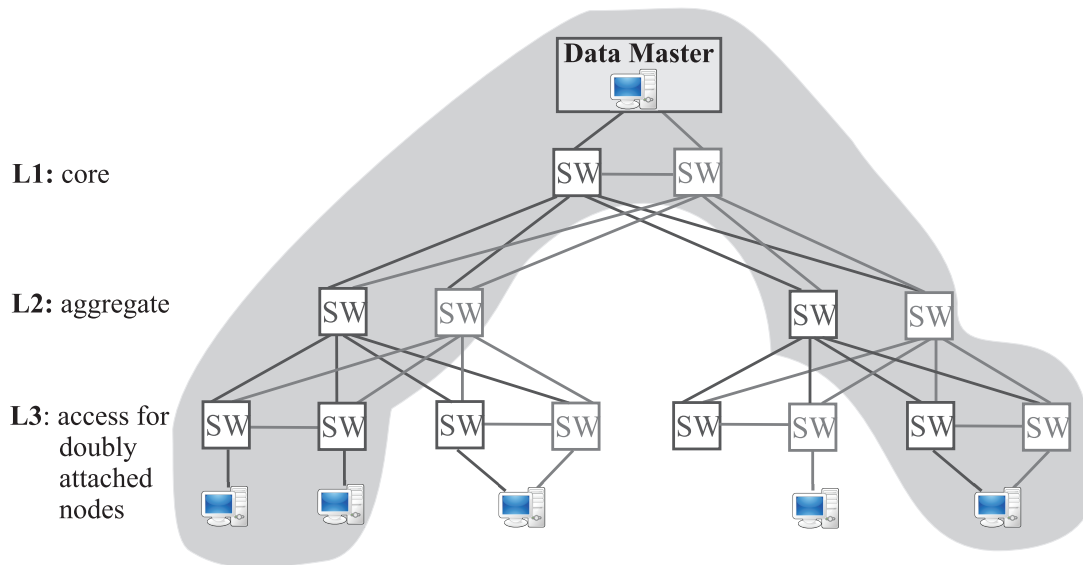


Figure 6.5: Topology applicable for the developed methods.

core switches. Each switch has at least two paths to any node and these paths include  $N$  or  $N + 1$  other switches, referred to as hops. In order to ensure two alternative paths from each core switch to the doubly attached data master, and similarly from each access switch to the doubly attached nodes, a link is required between switches to which nodes are doubly-attached, as depicted in the figure.

In the physical redundant network presented in Figure 6.5, pseudo-multipath logic spanning trees can be created as presented in Figure 6.6. Figure 6.6-a presents a subset of the network in Figure 6.5 that has grey background. Since this form is more readable, it is used for further

analysis. However, it is important to remember that this is a simplified view of a subset of a tree-like topology. The pseudo-multipath logic trees are defined using the same mechanism that is

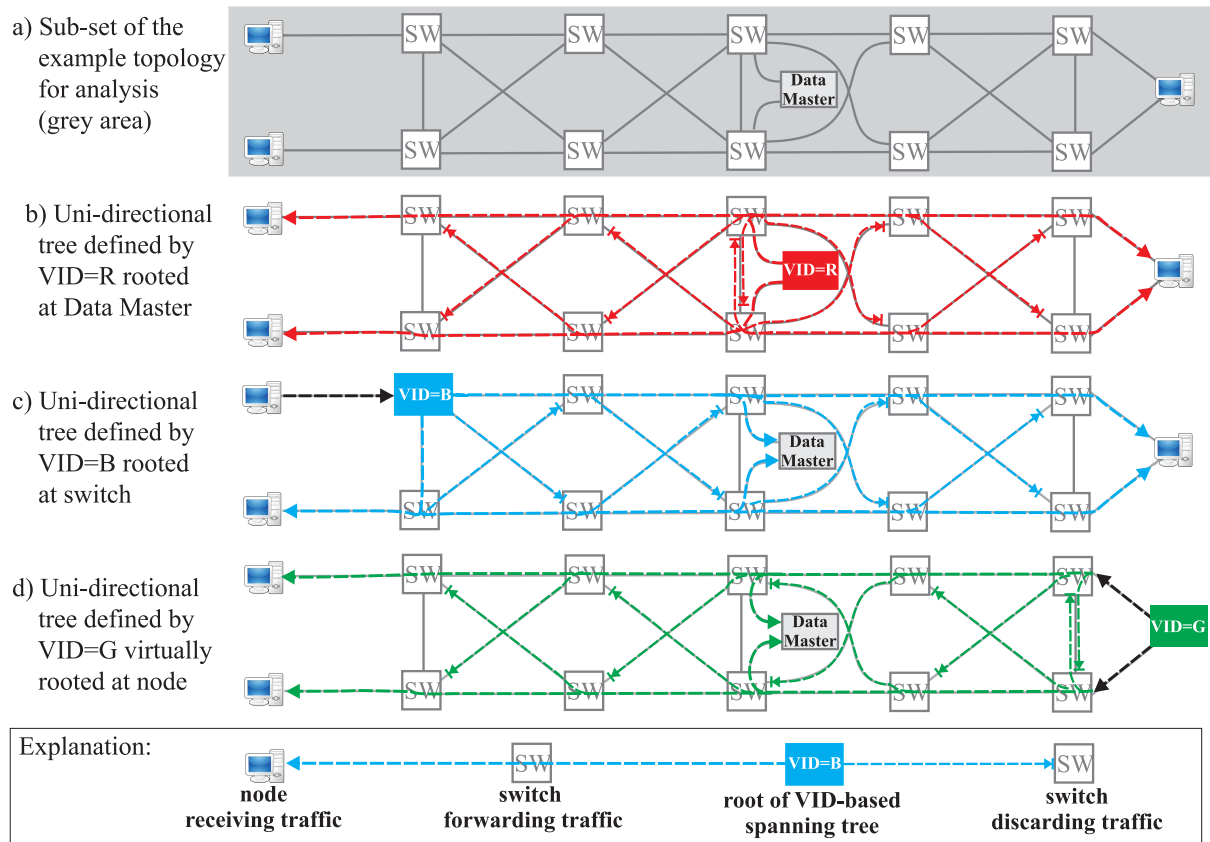


Figure 6.6: Unidirectional pseudo-redundant spanning trees applicable for the developed methods.

used to create VLANs. This is the IEEE 802.1Q tag that includes VID and priority. Unlike in the case of VLANs which define virtual broadcast LAN, here VIDs are used to create virtual unidirectional broadcast spanning trees. Such trees can be either rooted at a node or a switch that is connected to a node:

1. An *aware node* tags frames with a VID that defines a spanning tree rooted at that node. This is depicted in Figure 6.6 b.
2. An *unaware node* sends untagged frames. The switch is responsible for tagging these frames at ingress and untagging them at egress. The spanning tree can be either:
  - a) rooted at a switch, as depicted in Figure 6.6 c; this should be the case only if all the nodes connected to the switch are singly-attached, or
  - b) *virtually* rooted at a node, as depicted in Figure 6.6 d; this should be the case if nodes are multi-attached to many switches.

The guidelines on how to create an applicable redundant network and the algorithm to establish pseudo-multipath spanning trees in that network are provided below.

### Guidelines to create an applicable network topology

The following rules should be followed when creating a WR network with  $R$  redundancy level:

- Nodes are connected only to core and access switches.
- The number of core switches is correlated with the level of redundancy, e.g. triple redundancy ( $R = 3$ ) requires 3 core switches.
- Each aggregate switch is connected to  $R$  separate switches which are higher or lower in the hierarchy.
- Access and core switches to which the same node is multi-attached should be inter-connected.

As an example, these rules are applied to create the topology with triple redundancy ( $R = 3$ ) depicted in Figure 6.7. In the topology created according to these rules, the following algorithm can be used to create pseudo-multipath spanning trees.

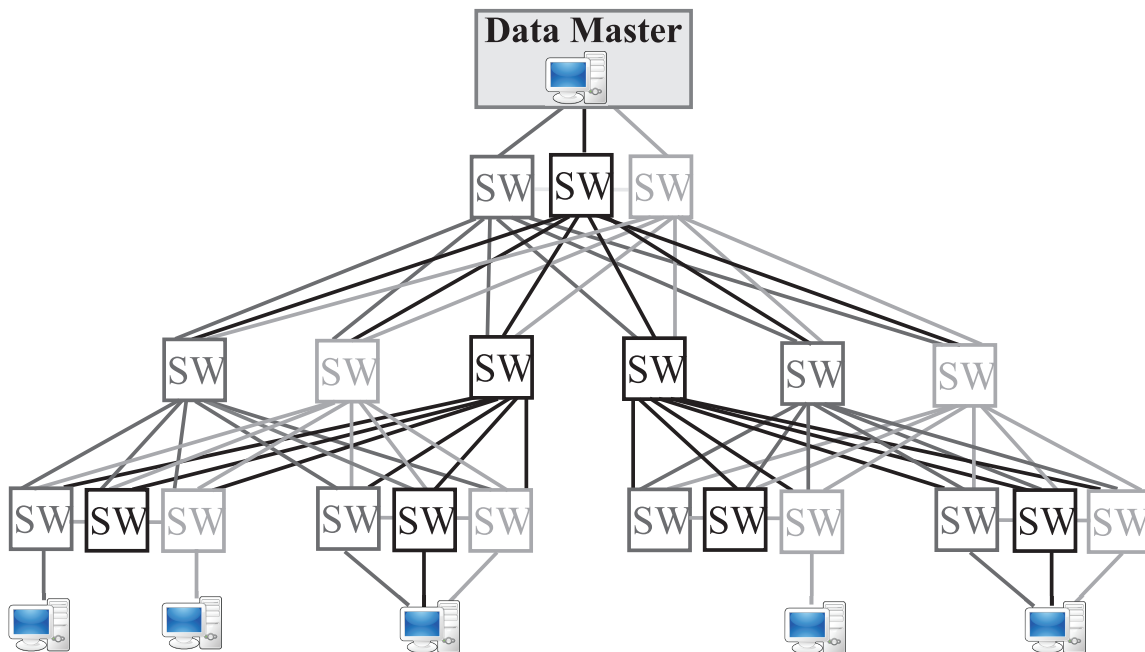


Figure 6.7: Topology with triple redundancy applicable for the developed methods.

### Pseudo-multipath spanning trees algorithm

The algorithm described in this subsection creates logic pseudo-multipath unidirectional spanning trees rooted at switches, or nodes, as depicted in Figure 6.6. Each tree is identified using a VID and determined by assigning to each port of a switch one of four roles: active, backup, forwarding, or OFF. The mechanisms developed in this thesis for data reliability and determinism require a network with a topology that follows the guidelines outlined above and the configuration provided by the algorithm presented in this subsection.

The prerequisite to the algorithm is the knowledge of network topology and a shortest path (Dijkstra) computation. Such a computation is performed by switches implementing SPB, as detailed in [36]. The SPB switches use the Intermediate System to Intermediate System (IS-IS) protocol to discover the network topology and distribute this information in the network. Each switch that implements SPB and runs the Dijkstra computation knows the shortest paths it belongs to. These paths are identified by unique path IDs.

The algorithm presented in Figure 6.8 and described below is performed for each port of each switch in the network, assuming the switch has already performed the Dijkstra computation of the shortest paths. The algorithm requires the input parameters listed below:

**VID:** the VLAN ID assigned to the spanning tree with a particular root,

**N:** the number of hops in the path

– with that particular root (VID)

– that includes the switch at which the algorithm is performed (not necessarily the port).

$N_{best}$ : the number of hops in the path that is shortest-by-hops

**n:** number of hops from the root to the switch at which the algorithm is performed,

$n_{best}$ : number of hops in the path from the root that is the shortest-by-hops,

**l:** estimated latency from the root to the port.

Each spanning tree is identified by a VID and created by proper configuration of the ports belonging to this VID on each switch. The configuration of a port is determined by the role that is assigned by the algorithm. At each switch in the network, the algorithm assigns roles of its ports, per root that is identified by the VID. The roles are assigned to ports as follows:

1. **Participating:** ports that are on the path to the root that is:
  - shortest-by-hops ( $N=N_{best}$ )
  - one-but-shortest-by-hops ( $N=N_{best} + 1$ ).
2. **Candidate:** subset of *participating* ports that have the shortest path to the root ( $n=n_{best}$ ); if  $n_{best}=1$  for the switch, the ports that have the path to the root of  $n=2$  are also the *candidate* ports.
3. **Active ingress:** the candidate port that has the lowest latency to the root.
4. **Backup ingress:** the port selected from the reminding *candidate* ports that has the lowest latency to the root (and an ordered list of backup ingress ports is created).
5. **Forwarding egress:** the *participating* ports that are not selected *candidate*.
6. **OFF:** all the reminding ports.



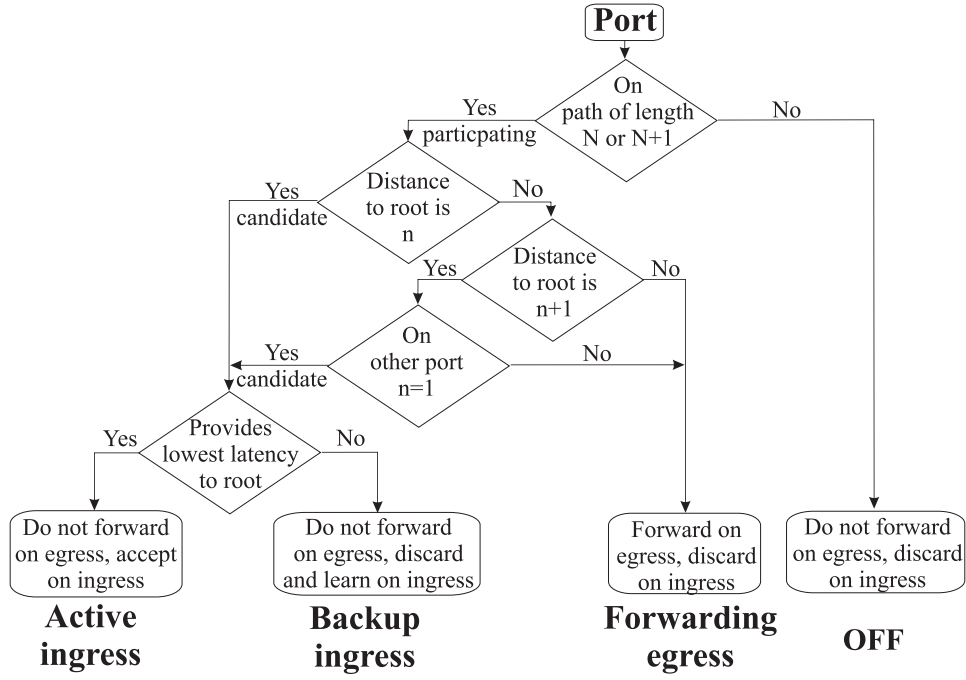


Figure 6.8: Algorithm to establish pseudo-multipath unidirectional spanning trees.

The per-VID port configuration is done according to Table 6.2, following the classification algorithm depicted in Figure 6.8 and described in this subsection.

Classification	Forward on egress	Accept on ingress	Learn on ingress
Active ingress	NO	YES	YES
Backup egress	NO	NO	YES
Forwarding egress	YES	NO	NO
OFF	NO	NO	NO

Table 6.2: Classification and configuration of ports on the switch.

### Pseudo-multipath E-TREE

The characteristics of traffic in the WR-based control and timing network allow application of the Ethernet Tree (E-TREE) concept [101] to optimise the usage of VIDs which is important as the number of VIDs is limited to 4096. In the E-TREE model, two types of end nodes are connected to the network: roots and leaves. A root can communicate with leaves and other roots. A leaf can communicate only with the root(s), communication between leaves is prevented. The concept of E-TREE is supported by the SPB as detailed in [36]. When using the Shortest Path Bridging VID Mode (SPB-VID), the configuration of an E-TREE requires two VIDs only. One VID, called Trunk VID, defines the set of root-to-leaf and root-to-root paths. Another VID, called Branch VID, defines the set of root-to-leaf paths.

The traffic in the control and timing network is either data-master-to-nodes, or node-to-data-master. Thus, the concept of E-TREE can be used to save VIDs. The pseudo-multipath version

of E-TREE proposed in this thesis is created using the algorithm presented in the previous subsection.

The pseudo-multipath E-TREE modifies the original E-TREE concept by disallowing communication between roots. The Trunk VID is used by the root to communicate only with the nodes. The Branch VID is used by the nodes to communicate only with the root, as depicted in Figure 6.9. The root-to-nodes tree is a pseudo-multipath spanning tree calculated as described

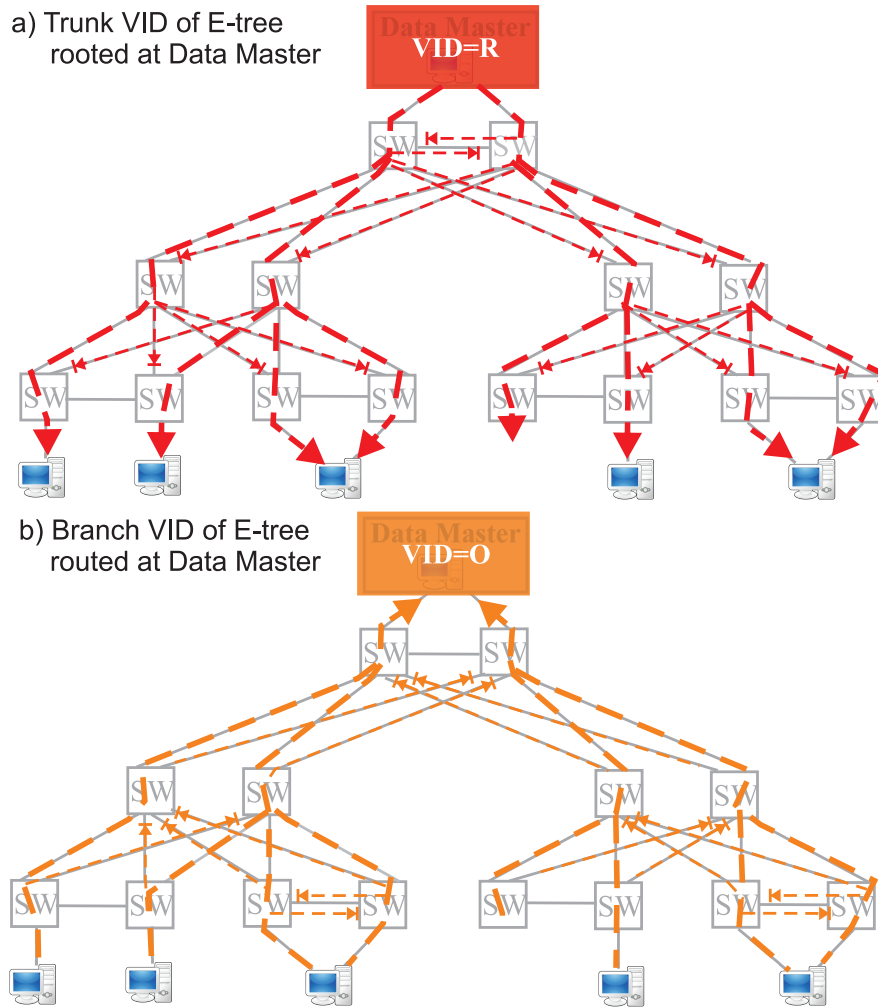


Figure 6.9: Pseudo-multipath Ethernet Tree.

in the previous subsection. The nodes-to-root tree is established as described below:

1. Pseudo-multipath-spanning branch trees rooted at access switches or nodes are calculated.
2. The intersection of all the branch trees that connect the root of each branch with the root of the trunk tree is found and configured.

The resulting E-TREE provides pseudo-multipath communication between data master and nodes that saves VID resources but restricts the communication to exclusively horizontal.

### 6.2.5 Lossless Reconfiguration when Adding an Element to the Network

Reconfiguration of a network after adding a new element, a switch or a fibre, can cause a substantial data loss. In the existing protocols, e.g. SPB or RSTP, such a frame loss after adding a new element is caused by temporarily blocking ports to prevent loops. The solutions presented so far in this thesis allow to create a redundant network with pre-configured backup ports and minimise frame loss when a network element fails. However, after such failure, the element needs to be replaced and a new configuration applied. Similarly, a new configuration is required after expanding an existing network with new elements. In both cases, the upgrade of the network configuration should be performed without disruption to the traffic, which is not the case for the existing solutions.

The method proposed below guarantees that no data is lost and no loops created during the reconfiguration process. It uses the port configuration names defined in Table 6.2 and proposes to upgrade port configuration in the following steps:

1. When a new network element is added, the ports affected are OFF by default.
2. New spanning trees are calculated and installed using new VIDs (e.g. Branch and Trunk VID). Optional tests can take place by sending test traffic to the new VIDs.
3. The new VIDs start being used:
  - if the node tags the outgoing frames, its configuration is changed
  - if the switch tags ingress frames, its configuration is changed.
4. The old VIDs remain effective until the reception of the last frames transmitted using these VIDs is detected, or a sufficiently long timeout expires.
5. The old VIDs are removed.

This process should be repeated for each spanning tree affected by the network update. The change can be done for each spanning tree at a time to minimise the number of additional VIDs needed.

As an example, Figure 6.10 depicts the proposed 5 steps of the lossless network reconfiguration in a simple network. The network consists of four switches (A, B, C, D) and two nodes (sending and receiving). Before the upgrade, the switches and nodes were connected using fibres a, b, c, d, e, f. The pseudo-multipath spanning tree was configured using VID=G and had two paths from the root: active (a, d, e, f) and backup (a, b, c, e, f). This network is upgraded with a new link g between switch A and C. The network configuration is updated as follows:

1. When the new link g is connected, the ports of the two affected switches, A and C, are configured with OFF role.
2. A new spanning tree is calculated and installed using VID=R.
3. The node starts sending frames using the new VID=R. Not all the frames that were sent using the old VID=G have been received yet.

4. Last frames sent using VID=G reach the receiving node.
5. The old VID=G is removed.

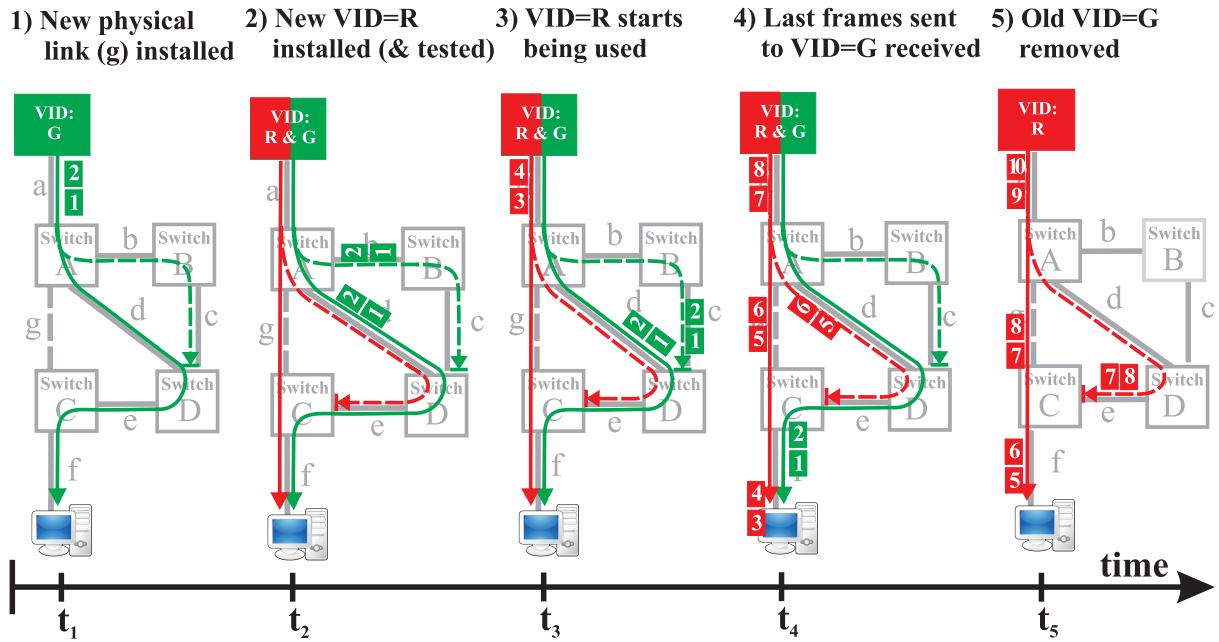


Figure 6.10: VID-based loss-less reconfiguration of network.

Such a method provides deterministic reconfiguration process that is precisely controlled and should not disrupt determinism of data forwarding, described in the next subsection.

### 6.2.6 Deterministic Data Forwarding

This subsection proposes a standard-compatible way of reducing the worst-case theoretical latency through a switch for a selected traffic. The reduction is required to ensure sub-10  $\mu s$  latency for the new Btrain system (subsection 1.1.4) and to ensure proper operation of the mechanisms introduced in the previous subsections. The theoretical latency of a switch results from the principles of the IEEE 802.1Q standard. This latency is improved by the methods proposed in this subsection. Once the worst-case theoretical latency is reduced, the latency introduced by the implementation becomes dominant. Thus, the implementation of the WR switch has been optimised for latency in the context of this thesis, as described in subsection 6.3.2.

The proposed method is referred to as *deterministic cut-through* and provides to the selected traffic an absolute precedence over any other traffic. The two types of traffic, selected and best-effort, are distinguished using priorities provided by the IEEE 802.1Q VLAN-tag. Similarly to any *cut-through* switch, a frame received at a port of the WR switch is evaluated and scheduled for transmission as soon as its header is received, even before the full frame is received and its CRC verified. However, unlike in other *cut-through* switches, the proposed solution instantly classifies each frame, based on its priority, as belonging or not belonging to the selected traffic. If the received frame belongs to the selected traffic i.e. it is critical, an ongoing transmission of the best effort frame is terminated, and the critical frame is transmitted instantly. The CRC of the interrupted frame is intentionally made incorrect so that:

- another receiving switch or node recognises the frame as being incomplete and discards it
- WR switches implementing this method are compatible with devices implementing the IEEE 802.1Q standard<sup>7</sup>.

The interrupted best effort frame can be either discarded completely or re-transmitted after the critical frame is sent. The WR switch implements the former, it is discarded.

As a result of the proposed solution, the latency through a single switch is independent from the the best effort traffic. If there is only one source that sends the selected traffic, the theoretical latency through a single switch is reduced to sub-microsecond (i.e. from 12.544  $\mu s$  to 0.208  $\mu s$ ) and the total switch latency depends mainly on the latency introduced by the implementation. Table 6.3 compares the total latency of the *store-and-forward*, *cut-through* and *deterministic cut-through* switches. The latency introduced by the implementation is denoted a sum of a constant and a variable element, i.e.  $\Delta \pm \delta$ . It is quantified in microseconds ( $\mu s$ ). The comparison of the three types of switches in Table 6.3 assumes a single source of the selected traffic. If more sources exist, the latency depends on the maximum frame size of the selected traffic and the number of sources sending this traffic. Even if more sources exist, the proposed solution can still provide a substantial control over the expected worst-case latency.

---

<sup>7</sup>With such solution, a WR switch is compatible but not compliant, i.e. it correctly interoperates with the devices implementing the IEEE802.1Q standard but it does not strictly follow the standard.

<b>Total latency</b> (Note 1)	<b>Store-and-forward</b> [ $\mu s$ ]	<b>Cut-through</b> [ $\mu s$ ]	<b>Deterministic cut-through</b> [ $\mu s$ ]	<b>Deterministic cut-through with PTP</b> [ $\mu s$ ]
Minimum	$(\Delta - \delta) + 0.512$ (Note 2)	$(\Delta - \delta) + 0.208$ (Note 3)	$(\Delta - \delta) + 0.208$ (Note 3)	$(\Delta - \delta) + 0.208$ (Note 3)
Maximum	$(\Delta + \delta) + 24.672$ (Note 4)	$(\Delta + \delta) + 12.544$ (Note 5)	$(\Delta + \delta) + 0.208$ (Note 3)	$(\Delta + \delta) + 0.208 + 0.96$ (Note 6)
Variation	$24.160 + 2\delta$	$12.336 + 2\delta$	$2\delta$	$0.96 + 2\delta$
Latency depends on	implementation and maximum frame-size of the selected traffic	implementation and frame-size of best effort traffic	implementation only	implementation and size of the PTP messages
Note 1: Total latency through the switch = theoretical latency + latency introduced by implementation Note 2: Reception time of minimum size Ethernet frame. Note 3: Reception time of the Ethernet header with VLAN tag. Note 4: Reception plus transmission time of maximum size Ethernet frame. Note 5: Reception time of the Ethernet header with VLAN tag and transmission time of maximum size Ethernet frame. Note 6: Reception time of the Ethernet header with VLAN tag and transmission time of the frame with PTP Announce				

Table 6.3: Latency through different types of switches for Gigabit Ethernet.

When the proposed solution is applied, the variation and maximum value of the total latency through a switch, in other words its determinism, greatly depends on the implementation. In particular, special attention must be paid to:

- the time needed for the lookup (search) of the MAC address and the forwarding processes
- the availability of resources to receive and forward the selected traffic even if congestion of the best effort traffic occurs.

In order to ensure availability of the memory resources, the memory that stores the incoming frames has been divided in the WR switch. Dedicated memory resources are reserved for the incoming critical frames. The reserved memory is sufficient to ensure undisturbed availability for critical frames, provided the number of the critical traffic sources is limited. Moreover, the MAC lookup and forwarding processes have been latency-optimised by the author for the selected traffic. These implementation improvements are described in subsection 6.3.2.

In practice, more than a single source of the selected traffic exists in any WR network. Still, the proposed solution can guarantee better performance than commonly available switches. Let's analyse the latency through a WR switch in a network that has a single *data master*, a *critical device* and a *non-critical device*. The single *data master* sends control messages that must be received by the *critical device* with an upper-bound in latency. The *non-critical device* can send any best effort traffic, such as diagnostics. Figure 6.11 shows how the different types of traffic are forwarded internally in the switch. The data master is one out of two sources of the selected critical traffic. The second source is the PTP protocol stack that runs on the switch. The PTP messages are configured to be sent as the selected (critical) traffic to ensures PTP operation even if the traffic sent by the data master is very high. A frame sent by the data master, in the

worst-case scenario, will experience a latency through the switch that is caused by the transmission of a maximum size PTP message, as depicted in the figure. The largest PTP message

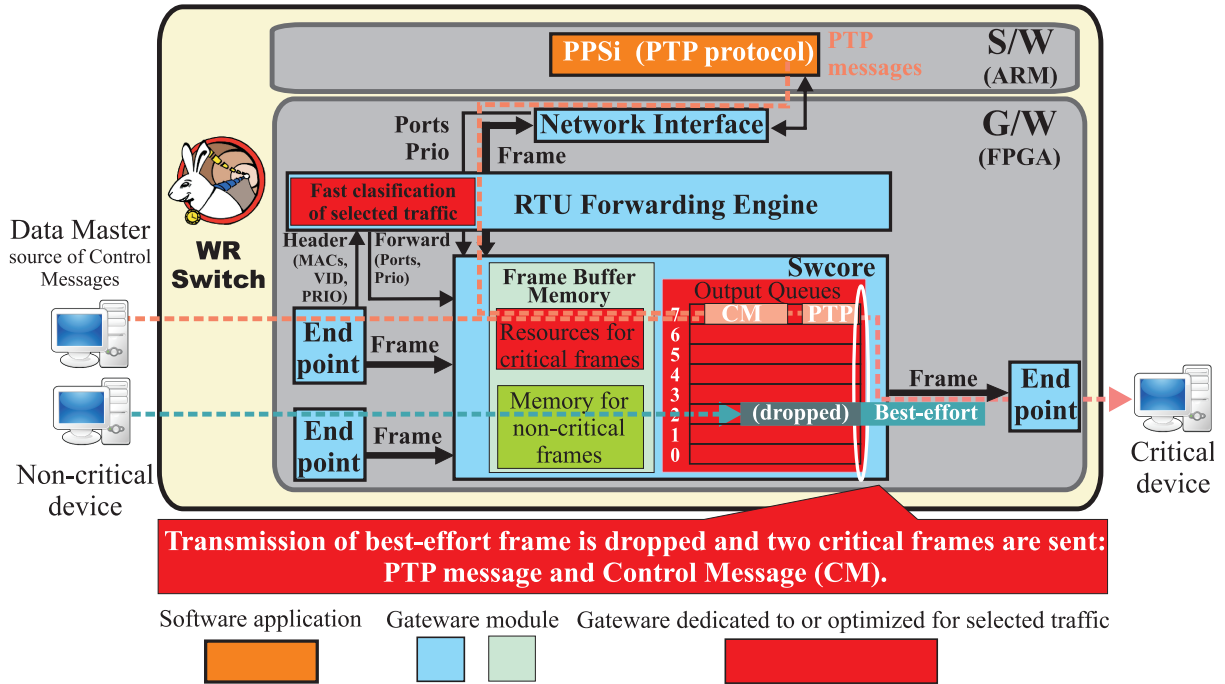


Figure 6.11: Deterministic forwarding of the selected and best effort traffic in the WR switch.

is the Announce message with WR Type-Length-Value (TLV)<sup>8</sup>. An Ethernet frame containing the PTP Announce message with WR TLV has a size of 100 bytes. The maximum theoretical latency introduced by this message is  $0.96 \mu s$ . It increases to  $(2\delta + 0.96) \mu s$  the latency variation of the frames sent by the data master as presented in the column of Table 6.3 called *Deterministic cut-through with PTP*. Compared to the *cut-through* switches, the improvement of the theoretical worst-case latency through the WR switch is 10-fold, which is significant. It must be noted that in this particular case, the variation of the theoretical latency of the PTP traffic will depend on the size of the frames sent by the data master. However, this is irrelevant for the operation of the WR protocol which requires high precision of timestamps which are generated whenever the frame is transmitted. Therefore, PTP in the WR network does not require deterministic latency.

The proposed method proves substantial theoretical improvement of determinism for a selected critical traffic that shares the network with a non-critical best-effort traffic. The method is supported by a latency-optimised implementation described in section 6.3.2. The implementation minimises the maximum latency and its variation for the forwarding and lookup processes. The theoretical variation calculated in this subsection is verified through tests in subsection 6.6.1. The unavoidable consequence of this solution is the deterioration in reliability of the traffic that is not selected as critical. However, this traffic is best-effort by design. Its increased

<sup>8</sup>Type-Length-Value extension with WR-specific information suffixed to the PTP Announce message.

latency and loss rate should not have critical impact. The higher-layer protocols are responsible to ensure successful delivery of information carried using best-effort traffic.



## 6.3 Implementation

This section describes the implementation of the methods outlined in the previous sections to support seamless redundancy and ensure determinism. Their implementation has been realised mainly in gateway while simple software has been developed to control the gateway. In the context of this work, the architecture and implementation of the WR switch has been modified by the author as depicted in Figure 6.12. The changes include:

- Creation of a Topology Resolution Unit (TRU) VHDL module that supports fast switchover.
- Modification of the RTU Forwarding Engine (RTU) and Swcore Multi-Access Memory (Swcore) modules to ensure deterministic forwarding of critical frames.

The design of the TRU and the most important modifications to the RTU & Swcore are described in the following two sections.

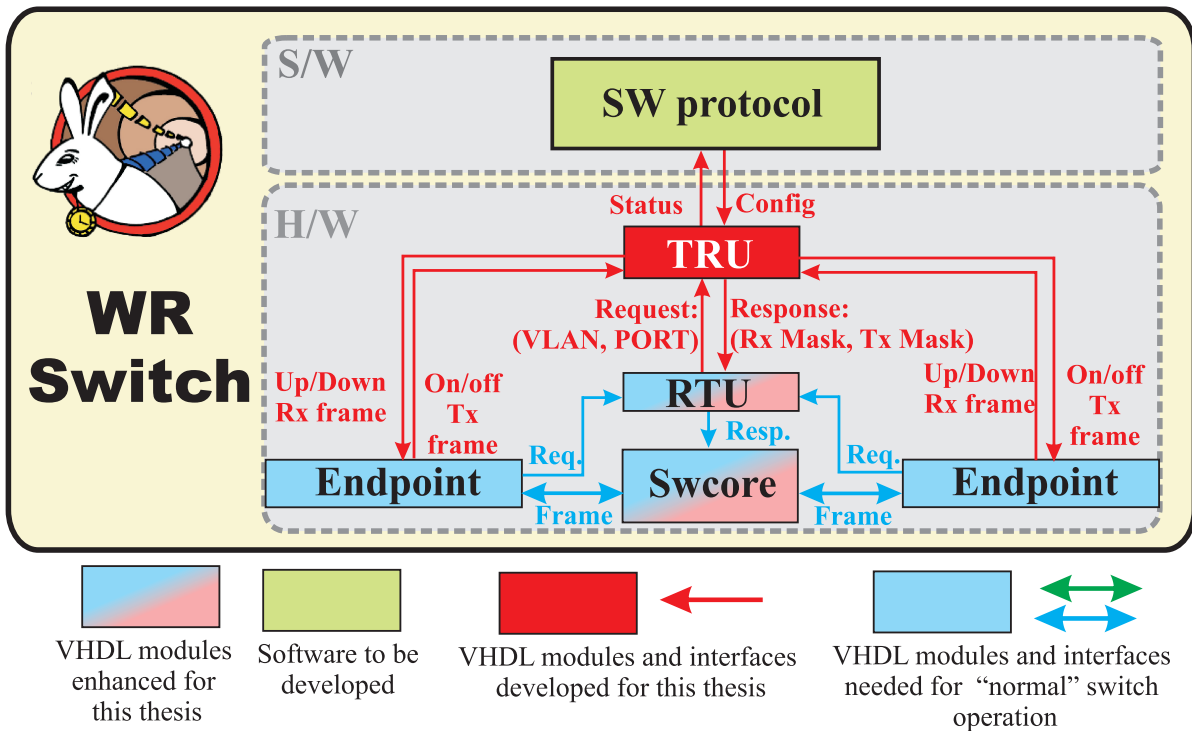


Figure 6.12: Updates and modifications to the WR switch developed for this thesis.

### 6.3.1 Topology Resolution Unit

The Topology Resolution Unit (TRU) is a VHDL module that speeds up failure-triggered reconfiguration of VID-based forwarding rules to a single clock cycle. For a given VID that defines a unidirectional spanning tree, the TRU allows to specify a number of alternative configurations through different types of logic transformation. The activation of a transformation is based on the information about the port state or detection of a specific frame. The initial configuration and its transformations are provided by the software running on the ARM microprocessor. The output of the TRU module supplements the forwarding decisions provided by the RTU.

The architecture of the TRU module is depicted in Figure 6.13. To allow flexibility and adapt to the internal operation of the RTU, the TRU works with Filtering IDs (FIDs) rather than with VIDs. As defined in IEEE 802.1Q, a single FID can be associated with one or more

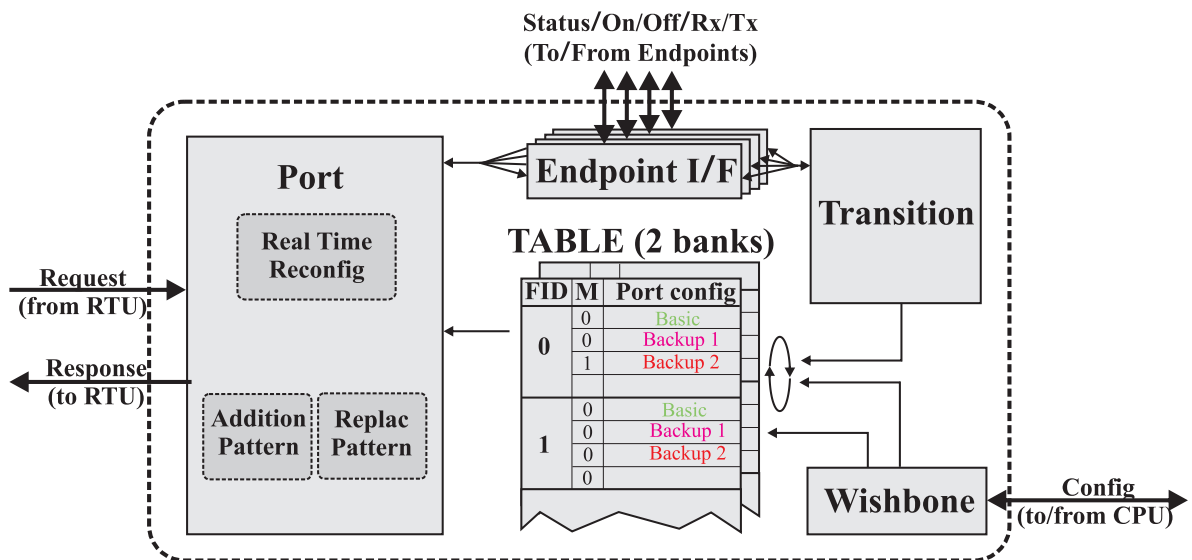


Figure 6.13: Architecture of the Topology Resolution Unit.

VIDs which allows Shared VLAN Learning (SVL). The TRU module provides the following functionalities:

- Pipelined processing of RTU requests in 2 cycles. The TRU can accept a new request from the RTU in every cycle and it provides a response with a forwarding mask within 2 clock cycles. This forwarding mask further restricts the RTU decision ensuring proper behaviour on redundant ports and preventing loops.
- Its configuration is stored in two-bank memory that allows atomic updates.
- It can activate a backup configuration based on exchange of Ethernet frames between peer WR switches. The generation and detection of these frames takes place entirely in gateway. The Transition sub-module of the TRU controls this process to ensure that no critical frames are lost during the reconfiguration.

- It controls the ports of the switch, the Endpoints, to prevent thrashing and unidirectional detection of failure. The TRU uses its *Endpoint I/F* to turn off the port as soon as its failure is detected. When a port goes up, the TRU verifies its stable operation before accepting and forwarding traffic on that port.

The TRU module provides flexible gateway support for different redundancy solutions (protocols) that are based on the principle of pre-configuration of backup paths. The module speeds up the switchover time by having the alternative configuration available in gateway and by accessing directly port state information. A good balance between flexibility and the required FPGA resources is achieved by per-FID reconfiguration through transformation using logic operations. A number of such transformations can be defined for each FID. For each transformation the following masks are defined:

1. Ingress - defines which ports accept and which discard incoming frames.
2. Egress - defines which ports transmit and which do not transmit frames.
3. Port Mask - defines which bits of ingress and egress masks are evaluated and modified by this transformation.
4. Pattern - defines the bit-pattern that is compared with an input vector from a configurable source. If the pattern and the input vector match at the bits defined by Pattern Mask, this transition is used.
5. Pattern Mask - defines which bits of the pattern are matched with the input vector.

The input-vector can be configured to indicate one of the following events:

1. Link failure – each bit indicates state of a single port, in particular whether the link is faulty/disconnected or operational.
2. Fast-block – each bit indicates whether a special *block frame* has been received on the corresponding port. This frame requests that the port is blocked instantly.
3. Fast-forward – each bit indicates whether a special *unblock frame* has been received on the corresponding port. This frame requests that the port is unblocked instantly.

A given event is expected only on a subset of ports defined by the Pattern Mask and it can affect only a subset of ports defined by the Port Mask. Each set of masks is configured for a specific input vector and one of three possible transformations:

1. Replace a subset of bits in the ingress and egress masks with their new values.
2. Add (i.e. logic OR) a subset of bits to the ingress and egress masks.
3. Subtract (i.e. AND NOT) a subset of bits from the ingress and egress mask.

Once the Pattern matches the input vector for the bits defined by the Pattern Mask, a particular type of transformation is performed.

For each FID, a number of entries with backup transformations are defined. Each entry has the set of parameters detailed in Table 6.4. Transformations of all entries that match their

Parameter name	Explanation
valid	Bit indicating whether the entry is valid
patMode	Integer indicating the pattern mode (replace, add, subtract)
patMask	Mask indicating bits of the patMatch to be matched with the input vector: readPatAdd or readPatRepl depending on the patMode .
patMatch	Pattern matching the entry – only if patMatch, masked by the patMask, matches the input vector, the transition in this entry is applied.
portMask	Mask defining which bits are to be modified by the portIng and portEgr
portIng	If replacement is used, it defines the new values of the bits of the ingress mask
portEgr	If replacement is used, it defines the new values of the bits of the egress mask

Table 6.4: Parameters in entries stored for each FID in the TRU\_TAB.

Patterns, per FID, are combined, as described in Listing 1. Among the entries defined for a FID, the first entry always defines an active topology and unconditionally matches. The next entries

**Data:** TRU\_TAB[MAX\_VLAN][MAX\_SUB\_VLAN], fid, readPatAdd, readPatRepl, readPatSub

**Result:** ingMask, egrMask

ingMask = egrMask = 0x0;

ent = TRU\_TAB[fid];

**while**  $i++ < SUB\_VID$  **do**

**if** ent[i].valid == 0 **then**

        continue;

**end**

**switch** ent[i].patMode **do**

**case REPLACE: do**

**if** (readPatRepl & ent[i].patMatch) == (ent[i].patMatch & ent[i].patMask) **then**

                ingMask = (ingMask & !ent[i].portMask) | (ent[i].portIng & ent[i].portMask);

                egrMask = (egrMask & !ent[i].portMask) | (ent[i].portEgr & ent[i].portMask);

**end**

**case ADD\_MASK: do**

**if** (readPatAdd & ent[i].patMatch) == (ent[i].patMatch & ent[i].patMask) **then**

                ingMask = (ingMask | (ent[i].portIngr & ent[i].portMask));

                egrMask = (egrMask | (ent[i].portEgr & ent[i].portMask));

**end**

**case ADD\_PATTERN: do**

**if** (readPatAdd & ent[i].patMatch) == (ent[i].patMatch & ent[i].patMask) **then**

                ingMask = (ingMask | (ent[i].portIngr & ent[i].portMask & readPatAdd));

                egrMask = (egrMask | (ent[i].portEgr & ent[i].portMask & readPatAdd));

**end**

**case SUBTRACT: do**

**if** (readPatSub & ent[i].patMatch) == (ent[i].patMatch & ent[i].patMask) **then**

                ingMask = (ingMask !& (ent[i].portIngr & ent[i].portMask & readPatSub));

                egrMask = (egrMask !& (ent[i].portEgr & ent[i].portMask & readPatSub));

**end**

**end**

**end**

**Listing 1:** Pseudo-code showing how the ingress and egress mask in the Topology Resolution Unit are generated.

transform the initial configuration into the backup one. The configuration and operation of a network with TRU support is summarized as follows:

1. On network (re-)start, no traffic is allowed except for the protocols that discover and configure the network.
2. Pseudo-multipath spanning trees are configured on each switch by software. This includes initial configuration and its transformation into a backup configuration.
3. The critical and best-effort traffic are allowed in the network.
4. Failure occurs, it is detected by the TRU that switches off the port at which the failure has been detected and transforms its initial configuration into the backup one.
5. The software is notified, a new topology is calculated and applied by the software as new TRU configuration. The reconfiguration is atomic through bank-swapping.

### 6.3.2 Deterministic Data Forwarding

The determinism of the WR switch is ensured through proper adaptation of the two gateway modules that are responsible for forwarding of frames between ports: the RTU Forwarding Engine (RTU) and the Swcore Multi-Access Memory (Swcore). The RTU decides to which ports a received frame should be forwarded, based on the *VLAN Table* with configured VLANs and the *HASH Table* with known MAC addresses. The Swcore implements multi-port multi-access memory that temporarily stores each received frame and allows all the transmitting ports to read that frame simultaneously.

The mechanisms described in subsection 6.2.3 have been implemented in both of these modules. Furthermore, the two modules have been optimised for low latency and deterministic forwarding of selected critical traffic. These determinism-enabling enhancements are detailed in the following subsections for each of the two modules.

#### RTU Forwarding Engine (RTU)

The time it takes the RTU to make the forwarding decision is a critical contributor to the latency of the switch, especially when the mechanism proposed in subsection 6.2.3 is applied. The RTU receives a forwarding request with the Ethernet header information from each port (Endpoint module). Based on this information, as well as the *VLAN Table* and *HASH Table*, the RTU provides a forwarding decision to the Swcore. This forwarding decision specifies to which port(s) the frame should be transmitted.

Before the modifications described in this subsection, the latency introduced by the RTU alone was in the range between  $0.3\ \mu s$  and  $4.8\ \mu s$ , depending on the traffic load. Actually, the RTU could not process all the requests at full bandwidth. As a consequence, the switch was not deterministic. The bottleneck was the common *Full Lookup Engine* used to prepare all the forwarding decisions by reading the VLAN Table and looking up the MAC addresses in the HASH Table. Each request was handled by the engine in  $0.24\ \mu s$  (15 cycles) and the requests were not pipelined. Thus the 18th request would wait  $4.8\ \mu s$  for a response. The grey area in Figure 6.14 shows the RTU original architecture.

In order to ensure determinism of the switch, the RTU must provide forwarding decision within the time it takes to receive a minimum-size frame when such frames are received on all the ports at full bandwidth. This means that the common lookup engine must handle forwarding requests every  $0.08\ \mu s$  (5 cycles) and complete them within  $0.68\ \mu s$  (42 cycles). It is a substantial improvement to be made.

However, not all the frames handled by the WR switch must be forwarded in a deterministic manner. In the intended application at CERN, the control and timing system, there are two possible types of the critical traffic that must be deterministic:

1. Broadcast or multicast frames from the data master to all nodes within a VLAN.
2. Multicast frames from a node to the data master.

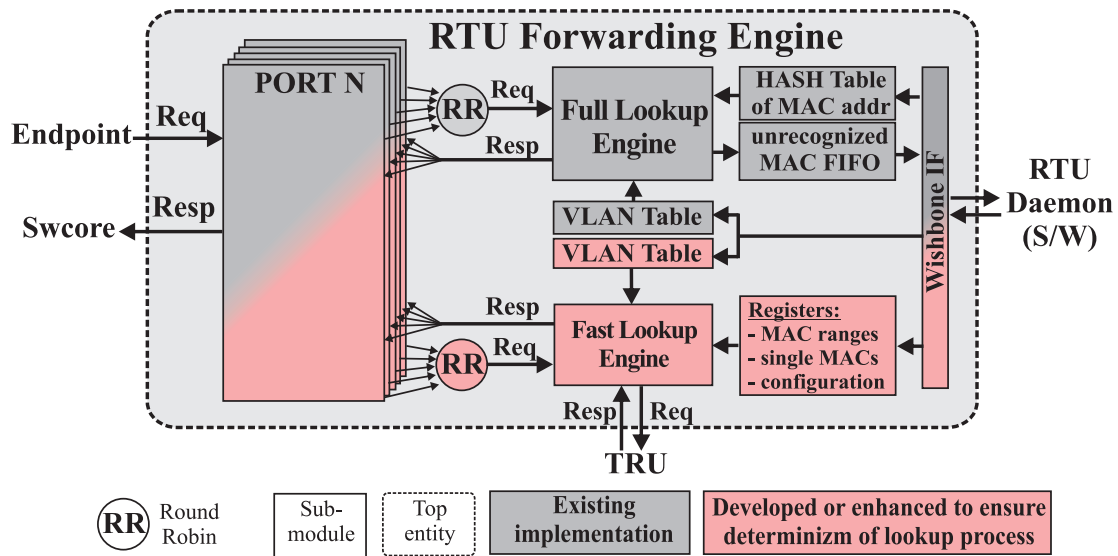


Figure 6.14: Architecture of the RTU Forwarding Engine enhanced for determinism.

The required behaviour of the multicast traffic is broadcast within a VLAN. All the critical traffic is sent within a pseudo-multipath spanning tree defined by a VID, as described in subsection 6.2.4 and the lookup of MAC addresses in the *HASH Table* seems superfluous.

It was therefore decided to ensure determinism of the RTU only for the intended critical traffic which is broadcast or multicast within a VLAN. Such traffic does not require the time-consuming lookup of MAC addresses in the *HASH Table*. The existing architecture of the RTU, the grey part of Figure 6.14, was extended with a dedicated *Fast Lookup Engine* and *VLAN Table*, as depicted with salmon colour in the figure. The two *VLAN Tables* have the same content. The new engine can accept requests every cycle, it processes them in a pipeline and provides a forwarding decision within  $0.4 \mu s$  (23 cycles). The engine makes forwarding decisions based on the *VLAN Table* and the input from the TRU module. Its response provides the following information:

- broadcast mask within VLAN – indicates to which port(s) the frame should be forwarded
- fast-forwarding flag – indicates whether the frame belongs to traffic for which fast-forwarding is configured
- critical-traffic flag – indicates whether the frame belongs to the selected critical traffic.

If fast-forwarding is configured for a given type of traffic, only the *Fast Lookup Engine* is used to prepare a forwarding decision for frames belonging to this type of traffic. Therefore, the latency of this traffic is deterministic. The following traffic can be configured as fast-forwarding:

- broadcast
- configurable ranges of MAC addresses (unicast or multicast)
- a few single MAC addresses (unicast or multicast)
- link-local protocols (including PTP).

In addition, fast-forwarded frames with priority specified by configuration are considered critical. For these frames, the mechanisms described in subsection 6.2.3 are implemented in the Swcore, as described in the next subsection.

For all the traffic that is not fast-forwarding, the two lookup engines, the *Fast Lookup Engine* and the *Full Lookup Engine*, work in parallel. If the next request arrives before the *Full Lookup Engine* provides a response, only the decision from the *Fast Lookup Engine* is used. Otherwise, the outputs from both engines are combined to create a final forwarding decision.

### Swcore Multi-Access Memory (Swcore)

The Swcore must be always available to accept new frames received from the Endpoints and have enough resources to handle RTU forwarding decisions of critical traffic. Only then, the mechanisms described in subsection 6.2.6 can be effective. The design and implementation of the RTU ensure that the critical frames are always recognised in a short and predictable time. There are two enhancements of the Swcore, depicted in Figure 6.15, that complement the ones in the RTU:

1. Split of memory resources to ensure that critical traffic can be stored regardless of any congestion of the non-critical best-effort traffic.
2. Interruption in transmission of a non-critical frame in order to send a critical frame.

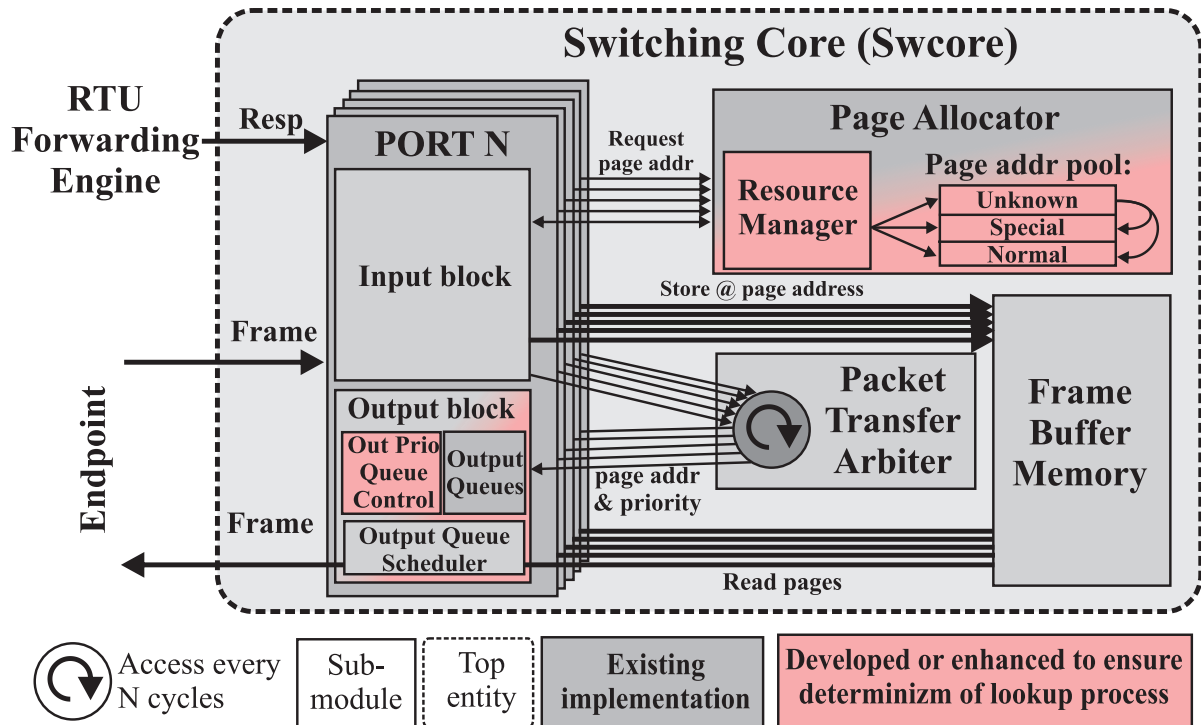


Figure 6.15: Architecture of the Swcore Multi-Access Memory enhanced for determinism.

The Swcore stores the frames received from the Endpoint in small chunks, called memory pages, in the *Frame Buffer Memory*. The addresses of the pages are dynamically allocated from



a pool of available pages by the *Page Allocator*. The allocation mechanism has been modified to divide the pool of all pages into three parts:

1. Unknown – dedicated to frames which are being classified.
2. Special – dedicated to frames classified by the RTU as critical.
3. Normal – dedicated to frames not classified by the RTU as critical, thus best-effort.

Each part is represented by a counter that tracks the number of pages allocated in that part of the pool and indicates when the number of pages dedicated to that part has been exhausted. Based on the decision from the RTU, pages from the appropriate pool are allocated to store the received frames. Before the RTU decision is available, a pool of unknown pages is used to store the frame. As soon as the RTU decision is provided, the unknown pages are classified as special or normal and the unknown count is released. The number of unknown pages is sufficient to accept a new frame at any port so that additional latency is avoided and the potentially critical frames are always served.

The received critical frames are classified appropriately by the RTU, stored using dedicated resources in Swcore, and finally forwarded to the output queues. There is one queue per priority. The *Output Queue Scheduler* that implements strict-priority indicates which frame is to be sent next based on its priority. This scheduling mechanism has been modified to stop current transmission of a non-critical frame if the next frame to be sent is critical, according to the idea described in subsection 6.2.6. This mechanism makes the latency of critical frames independent of the size of non-critical frames in output queues. However, the process of discarding the non-critical frame and starting to send the critical frame takes between  $0.4 \mu s$  and  $0.7 \mu s$  and this adds to the implementation-specific variation ( $2\delta$ ) of the latency through the switch.

As a consequence of the presented enhancements to the RTU and the Swcore, the entire forwarding chain (RTU lookup, Swcore storage and transmission) is almost non-blocking for the critical traffic. The only significant latency in each of the two modules comes from the arbitration of the access to the common resources. Since the access is pipelined, each port is served after a maximum number of cycles that equals the number of ports. Such an arbitration is cascaded as the Swcore must wait for the RTU. For a single stream, the maximum variation of latency through the switch results from serial arbitration and amounts to approximately:  $2\delta \approx 0.6\mu s^9$ . This variation is increased when critical and non-critical traffic is sent to the same port. In such case, the maximum latency variation amounts to approximately:  $2\delta \approx 1.3 \mu s$ . This is the major contributor to the variation of the latency introduced by the switch implementation modified in the context of this thesis.

Based on the design evaluation and simulations, the latency values introduced by the switch are predicted :  $\Delta = 2.432 \mu s$  and  $\delta = 0.296 \mu s$ . Table 6.5 applies these values in the theoret-

---

<sup>9</sup>The RTU introduces max 18 cycles and the SWcore 19 cycles (additional internal port to CPU). Each cycle is 16 ns.

ical considerations from subsection 6.2.6. These values are verified in tests presented in the following section 6.6.1.

Use case	Latency	
	Maximum [ $\mu s$ ]	Variation [ $\mu s$ ]
Deterministic cut-through without intervening traffic	$(\Delta + \delta) + 0.208 = 2.936$	$2\delta = 0.596$
Deterministic cut-through with intervening best-effort	$(\Delta + \delta) + 0.208 + 0.7 = 3.636$ (Note 1)	$2\delta + 0.7 = 1.292$ (Note 1)
Deterministic cut-through with intervening PTP	$(\Delta + \delta) + 0.208 + 0.96 = 3.896$	$2\delta + 0.96 = 1.556$
Note 1: The additional $0.7 \mu s$ is needed for discarding the best-effort frame. .		

Table 6.5: Latency values through WR switches estimated based on design evaluation and simulation.

## 6.4 Limitations of the Used Methods, Alternative Solutions

The methods developed in the context of this thesis provide a solution for the type of topology and traffic that are expected in the CERN control and timing network, the initially intended application of WR. During the work on this thesis, a number of new applications appeared that might not take full advantage of all these methods. In parallel with the proposed methods, standards that might provide alternative solutions are being created by the Institute of Electrical and Electronics Engineers (IEEE).

The developed methods to support seamless redundancy provide flexibility in the supported level of redundancy but restrict the number of supported topologies. This means that as long as the topology resembles the multi-path spanning tree detailed in subsection 6.2.4, the methods will work and the number of redundant paths can be greater than 2, if needed. However, other types of topologies, in particular topologies that create rings, will not work with the proposed methods.

The methods to ensure determinism and their implementation are optimised for specific characteristics of the critical traffic. In particular, broadcast traffic that is sent by a limited number of nodes, data masters, to all other nodes in the VLAN. The nodes are expected to send only sporadic critical traffic to the multicast address of the data master. A strict control of sources of critical traffic is required for the methods to work.

In 2013, during the work on this thesis, the IEEE 802.1CB [55] project was started. It prepares a standard for "native support" of Seamless Redundancy in Bridged LANs. This standard, when finished, will provide alternative solutions to the methods presented in the following sections:

- 6.6.2: Fast switchover between pre-configured active and backup ports.
- 6.2.5: Lossless reconfiguration when adding an element to the network.

Another IEEE project, a joint effort by IEEE P802.3br [102] and P802.1Qbu [103] groups, will offer an alternative to the Deterministic Data Forwarding mechanism presented in section 6.2.6.

The developed standards and the proposed methods are aligned in the direction of their solutions. However, the IEEE solutions are more generic. It is therefore likely that some of the IEEE standard enhancements will be integrated with the methods presented in this thesis in the future.

## 6.5 Usefulness and Applications of Proposed Methods

The described methods and implemented enhancements are prepared for the next generation CERN control and timing network, yet to be deployed. The described latency optimisations of the WR switch are used in two systems already deployed at CERN:

- WR-Btrain [104] that is currently tested in operation in the Proton Synchrotron (PS), it will be deployed in ELENA, the PS Booster and possibly other accelerators in the future.
- WR Trigger Distribution (WRTD) [105] that is currently used in the Large Hadron Collider (LHC) to diagnose beam instabilities.

The latency optimisations will be useful in other WR-based systems that are currently being developed, such as the Radio-frequency (RF) distribution over WR [105].

The presented solutions should be considered a palette of features that can be used in various configurations. Moreover, the TRU module is intended to support a different approach to seamless redundancy that is developed at GSI Helmholtz Centre for Heavy Ion Research (GSI) [22]. This flexibility of TRU configuration can be useful in testing different software implementations of protocols that use the idea of pre-configured backup paths.

## 6.6 Measurements and Tests

### 6.6.1 Determinism

The latency of the WR switch is tested to confirm its determinism for the selected critical traffic. For WR applications, the determinism of the WR switch forwarding is quantified by a guaranteed maximum latency and its peak-to-peak variation. The selected critical traffic is a broadcast within a VLAN. The tests are performed with Spirent testers<sup>10</sup> borrowed from the CERN IT department where these devices are used for exhaustive tests of switches and routers before purchases.

#### Latency of critical traffic without other intervening traffic

The first test evaluates the maximum latency and the latency variation over one and two WR switches without other intervening traffic. This latency is introduced only by the gateway modules optimised by the author for latency. The test setup, depicted in Figure 6.16, consists of two

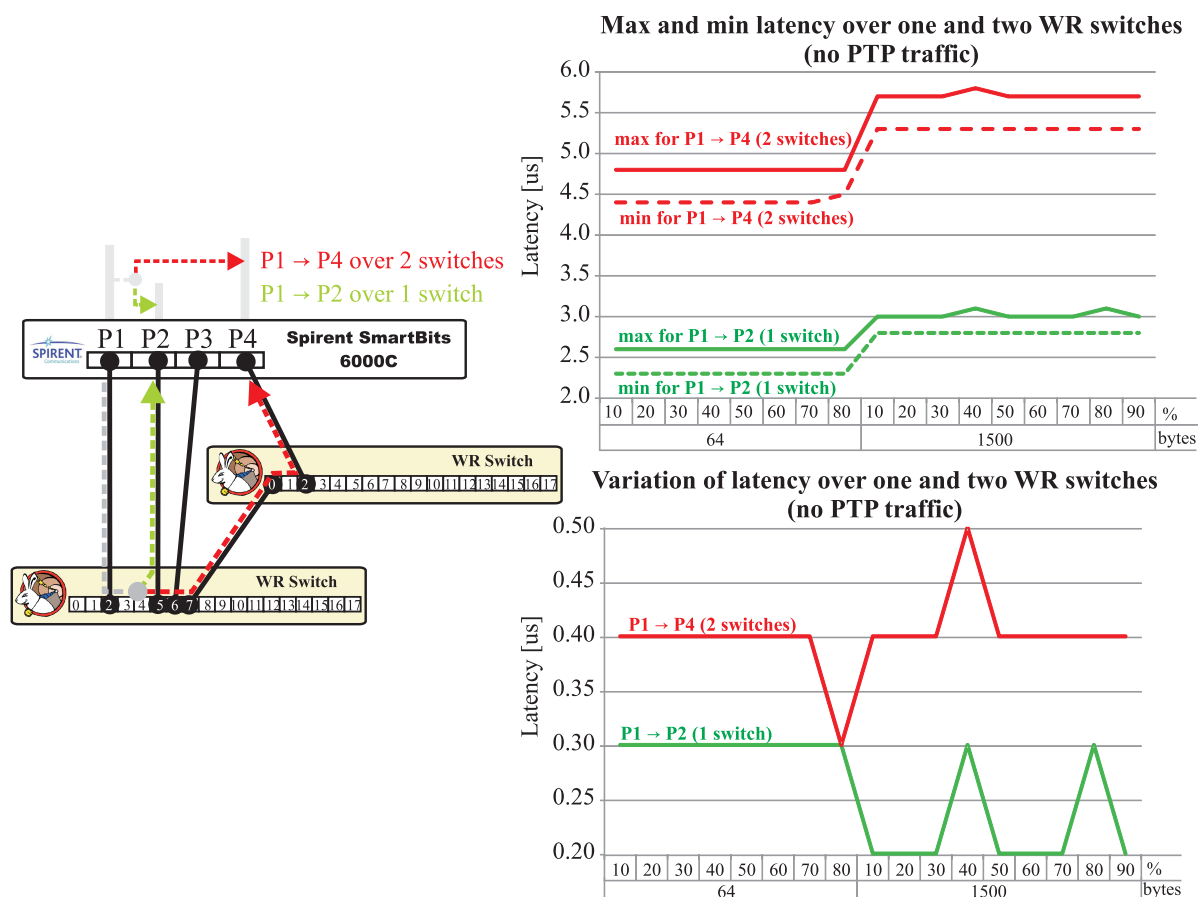


Figure 6.16: Latency over one and two WR switches without intervening traffic.

<sup>10</sup>SmartBits 6000C Performance Analysis System and Spirent TestCenter

cascaded WR switches and a Spirent SmartBits 6000C Performance Analysis System connected using very short fibres that introduce negligible latency. The Spirent sends a broadcast stream of critical traffic on port P1. This stream is received by two other ports, P2 and P4. The traffic between port P1 and P2 traverses one WR switch while the traffic between P1 and P4 traverses two WR switches. This allows to evaluate how cascading of switches influences latency.

During this test, a burst of data is sent for 100 s for each configuration of traffic load and two extreme frame sizes, 64 and 1500 bytes. The measurement results in Figure 6.16 show that:

- the maximum latency over a single switch is  $3.1 \mu s$  and its peak-to-peak variation is  $0.3 \mu s$
- the maximum latency over two switches is  $5.8 \mu s$  and its peak-to-peak variation is  $0.5 \mu s$ .

### Latency of critical traffic with intervening PTP traffic

The second test evaluates the maximum latency and the latency variation over one and two WR switches with intervening PTP traffic. This use case is more realistic and thus more interesting. Therefore more exhaustive test is performed. While the same setup as in the first tests are used, additionally to the stream sent from port P1 to ports P2 & P4, another stream is sent from port P4 to ports P1 & P3.

During this test, a burst of data is sent for 100 s for different configurations of traffic load and four different frame sizes: 64, 128, 512 and 1500 bytes. The measurement results in Figure 6.17 show that with the intervening PTP traffic:

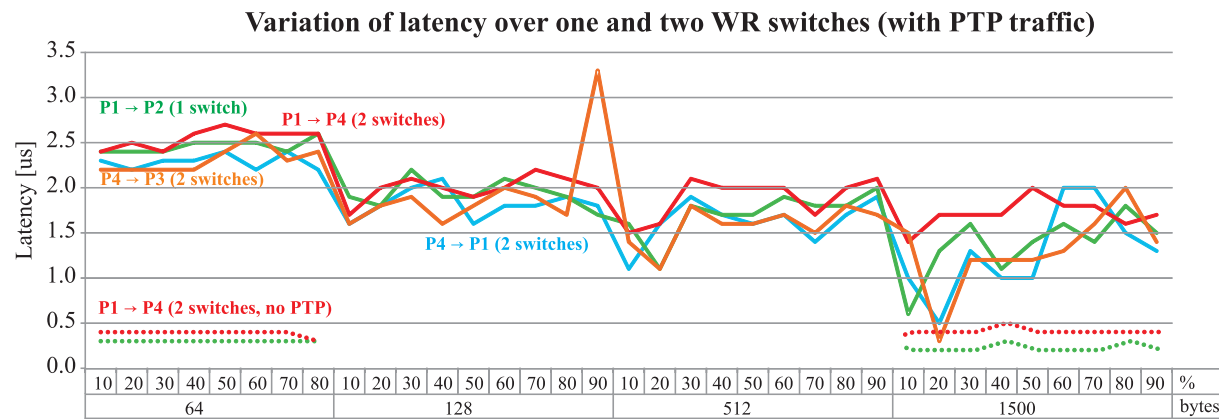
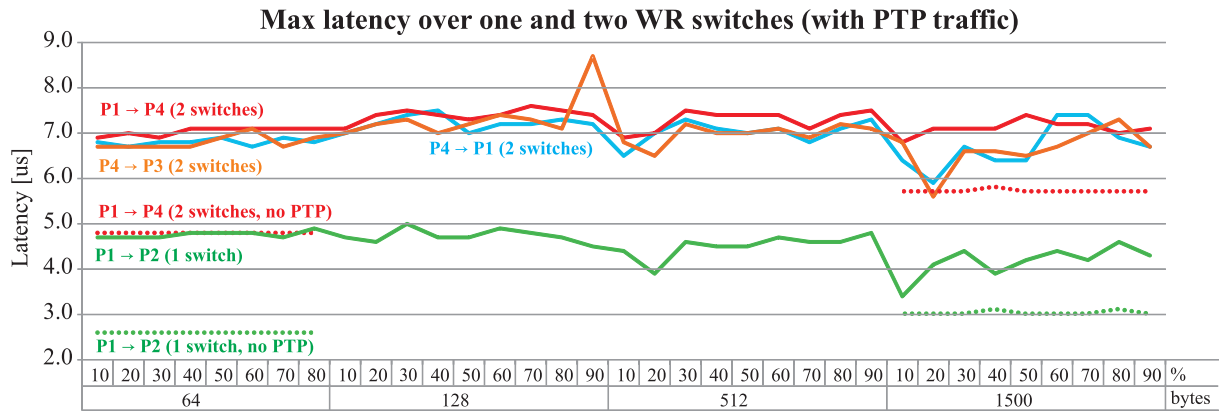
- the maximum latency over single switch increases to  $5.0 \mu s$  and its variation to  $2.6 \mu s$  (increase by  $2.9 \mu s$  and  $2.3 \mu s$  respectively, compared to latency without PTP traffic)
- the maximum latency over two switches increases to  $8.7 \mu s$  and its variation to  $3.3 \mu s$  (increase by  $2.9 \mu s$  and  $2.8 \mu s$  respectively, compared to latency without PTP traffic).

For selected loads and frame sizes, long-term latency measurements with PTP traffic were performed to verify whether the latency values increase. The results in Table 6.6 show that during 10 days and 90.56 Gigabytes of data sent, the maximum latency experienced by some of the frames is  $5.6 \mu s$  and the maximum latency variation is  $2.7 \mu s$ . The latency variation further increases to  $2.8 \mu s$  when a different traffic pattern is used.

	Test parameters	Test Duration	Latency			
			over one switch Max	pk-pk	over two switches Max	pk-pk
1	64 bytes, 12.61% and 0.05% load (WR-Btrain scenario)	35.7h (1.5d)	5	2.7	7.7	3.2
2		233.8h (9.7d)	5.6	2.7	-	-
3	64 bytes, 50% load	38.5h (1.5d)	5.1	2.8	8.4	3.9

Table 6.6: Long-term latency test with intervening PTP traffic.

Based on the short- and long-term tests, the worst-case maximum latency of a single stream of critical data over a single WR switch, with intervening PTP traffic, can be assumed to be  $4.2 \pm 1.4 \mu s$ , regardless of the destination port, as verified in the next test.



**Test setup with Spirent SmartBits 6000C**

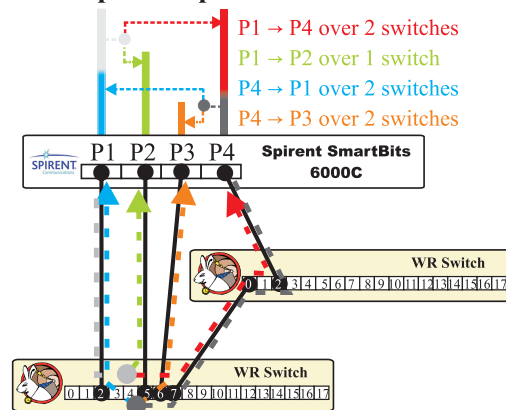


Figure 6.17: Latency over one and two WR switches with intervening PTP traffic.

### Latency of critical traffic for all ports

The third test verifies that the latency through the WR switch does not depend on the destination port to which the frames are forwarded. The Spirent SmartBits 6000C sends a burst of data for 100 s for each configuration of two traffic loads and frame sizes, i.e. 10% and 50%, and 64 and 1500 bytes. As depicted in Figure 6.18, the data is broadcast by the switch to all the other ports. Latency through the switch is measured on each port. Since the PTP traffic introduces a substantial variation of latency, most of the tests are performed without PTP traffic. Only one

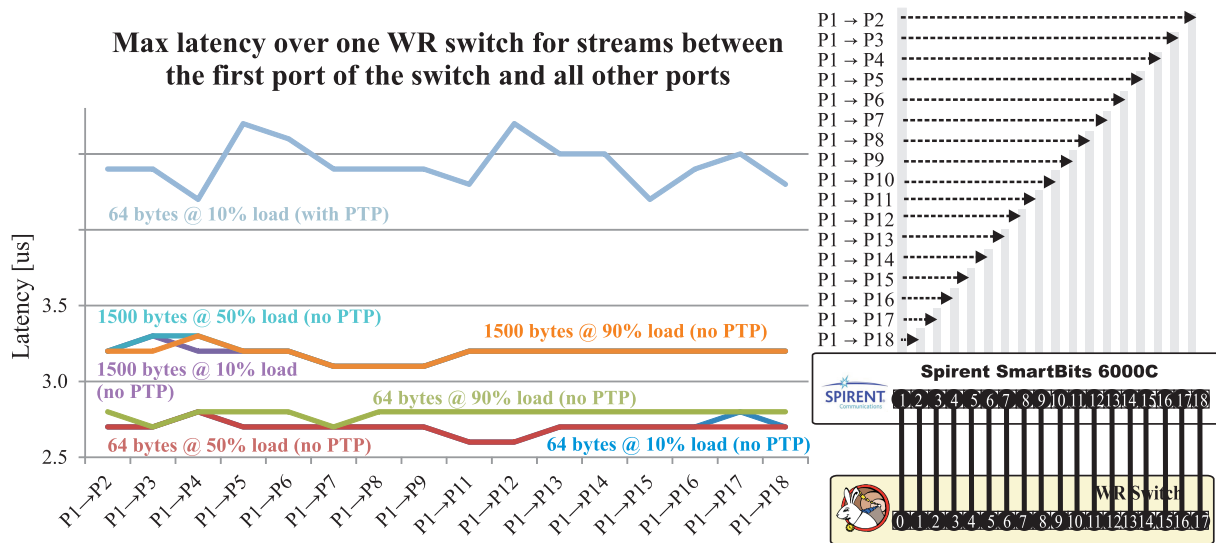


Figure 6.18: Latency of critical traffic for all ports with and without intervening PTP traffic.

test is performed with PTP traffic to verify that the PTP traffic does not introduce port-dependent latency variation. The measurement results in Figure 6.18 show that:

- The port-to-port difference in latency is negligible.
- The latency through the switch depends on the number of destination ports when the frames are forwarded to all ports, the maximum latency through a single WR switch increases to  $3.3 \mu s$  and the latency variation to  $0.6 \mu s$ ; this is an increase of  $0.2 \mu s$  and  $0.3 \mu s$ , respectively, compared to the latency of forwarding to only two ports in the previous tests.
- The PTP traffic introduces much greater latency variation than the factors mentioned above. This variation is not correlated with the port numbers between the frames are forwarded.



## Latency of critical traffic with intervening best-effort traffic

The last test verifies the latency of a single stream of selected critical traffic in the presence of best-effort traffic. The Spirent TestCenter generates 10 streams that are sent between its 4 ports through a single WR switch, as depicted in Figure 6.19. The blue stream is sent to multicast

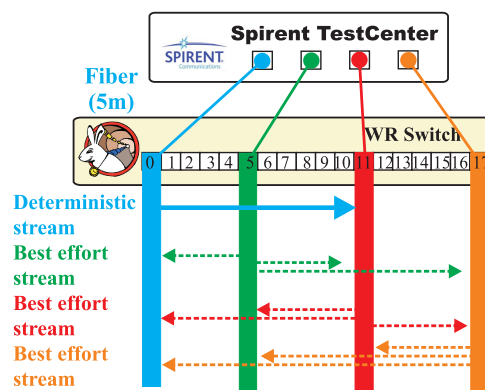
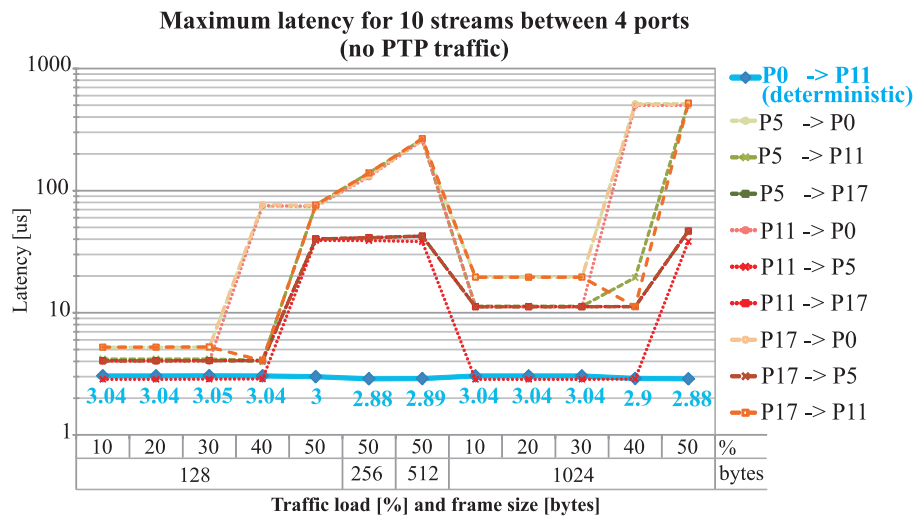


Figure 6.19: Latency of critical traffic with intervening best-effort traffic.

address with priority 7 and the switch is configured to consider such traffic as critical. The other 9 streams have address and priorities that are treated by the switch as best-effort. PTP traffic is disabled to allow better precision of latency measurement. The latency of fibre links is negligible.

During this test, a burst of the 10 streams is sent for 10 s for each configuration of different frame sizes and traffic loads<sup>11</sup>. The measurement results in Figure 6.19 show that the maximum latency of the blue critical stream remains below  $3.1 \mu s$  and its variation remains below  $0.3 \mu s$ . On the other hand, the best-effort streams experience large latencies to give room for the critical data. The maximum latency of the best effort streams is around  $20 \mu s$  without congestion (below 40% load) and up to  $500 \mu s$  when the congestion happens. Regardless of the congestion and losses of the best effort frames, no frames of the critical stream are lost.

<sup>11</sup>The provided load value is an accumulated load of all streams sent by a single port.

## Conclusions

The tests confirm that WR switch guarantees latency below  $10 \mu s$  for critical traffic. Table 6.7 compares the theoretical latency values of store-and-forward and cut-through switches with the expected and measured latency values of the WR switch. In general, the latency values expected through simulation and design evaluation match closely the measured values. The latency values measured for the case with the intervening PTP traffic are greater than expected which indicates that improvement is possible and further work is needed.

Use case	Latency	
	Maximum [ $\mu s$ ]	Variation [ $\mu s$ ]
Standard store-and-forward switch	$(\Delta + \delta) + 24.672$	$2\delta + 24.160$
Commonly-available cut-through switch	$(\Delta + \delta) + 12.544$	$2\delta + 12.336$
Values predicated based on design evaluation and simulation: $\Delta = 2.432 \mu s$ and $\delta = 0.296 \mu s$		
Expected WR deterministic cut-through without intervening traffic	2.936	0.592
Expected WR deterministic cut-through with intervening best-effort	3.636	1.290
Expected WR deterministic cut-through with intervening PTP	3.896	1.556
Measured values: $\Delta = 2.792 \mu s$ and $\delta = 0.3 \mu s$		
Measured WR switch without intervening traffic	3.3	0.6
Measured WR switch with intervening best-effort traffic	3.1	0.3
Measured WR switch with intervening PTP traffic	5.6	2.8
CERN requirement	< 10	minimal
Requirement for seamless latency when using 4 FEC frames of 600 bytes	-	2.5

Table 6.7: Latency through different types of switches for Gigabit Ethernet.

The measurements performed by the author are in agreement with tests conducted independently to evaluate the new WR-Btrain system. These results are summarized in Table 6.8. The measured latency fulfils the requirements presented in subsection 1.1.4 and provide a worst-case maximum latency that is better than other currently available standard solutions. This worst-case latency is not degraded during fast switchover, as tested in the next subsection.

The document that reports the latency measurement	Latency through a WR switch measured for min-size frames sent by					
	one data master			two data masters		
	Min	Max	pk-pk	Min	Max	pk-pk
Tests of WR-Btrain for the PS Frequency Program [106]	$4.0 \mu s$	$4.4 \mu s$	$0.4 \mu s$	$4.0 \mu s$	$5.2 \mu s$	$1.2 \mu s$
Tests of WR-Btrain for the POver for the PS (POPS) [107]	$3.7 \mu s$	$4.1 \mu s$	$0.4 \mu s$	$3.7 \mu s$	$4.5 \mu s$	$0.8 \mu s$

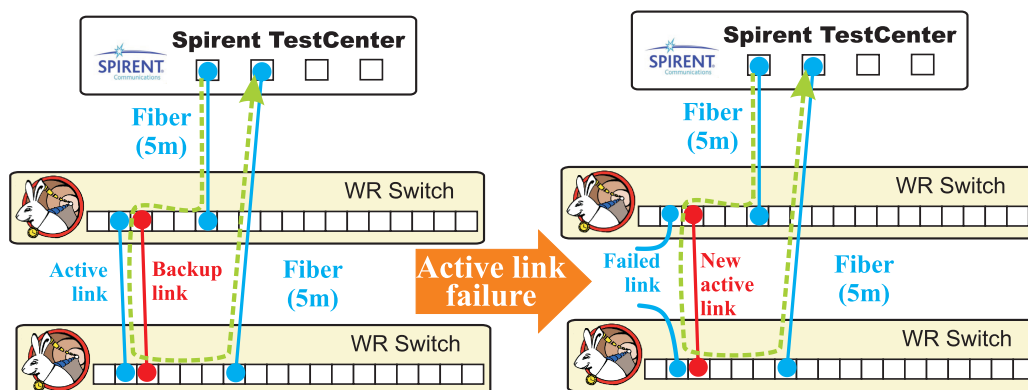
Table 6.8: Latency measurement performed within the context of the WR-Btrain project.

## 6.6.2 Fast Switchover Between Redundant Paths

The WR switch is required to switch over between the path that fails and its preconfigured backup fast enough to lose not more frames than can be recovered by FEC. As described in section 4.6, seamless redundancy in a WR network can be achieved by encoding a single control message into a number of FEC frames. Not all of these frames need to be received to recover the original control message. The sizes of the FEC frames analysed in section 4.6 are 300, 600 and 1200 bytes. The number of frames that can be lost, parity frames, is recommended in section 4.9 to be two or more.

The fast switchover in the WR switch is implemented by the TRU described in subsection 6.3.1. In order to verify that the implementation works as expected, the test presented in Figure 6.20 is performed. In the test, a professional Spirent traffic generator and analyser sends

- a) Active and backup link are functional      b) Active link has failed, backup link has become active



- c) Frame loss and latency measured by Spirent for each stream during failure

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

~3GB of data      Lost not more than 1 frame during switchover

Figure 6.20: Test of fast switchover between redundant links.

streams of frames through two WR switches connected redundantly. The streams have constant frame size of 288 bytes and different loads, i.e. 10, 30, 50, 70 and 90%. One of the Spirent ports sends streams to the upper WR Switch. This switch is connected to the lower WR switch with

two links. Both switches are configured to recognise one of the redundant links as active and the other as backup. The lower WR switch is connected back to the Spirent analyser that counts the number of lost or corrupted frames and measures the transmission latency. All the fibres connecting the switches and the tester are of the same length and introduce negligible latency. While the stream of each load flows, failure of the active link is simulated by disconnecting the link, see Figure 6.20-a and Figure 6.20-b.

The results in Figure 6.20-c show that no more than a single frame is lost in each test while the maximum latency through the two switches is barely affected by the failure. The loss of not more than a single 288 byte frame at 90% load demonstrates that the switchover time is below  $2.7 \mu s$ , as explained in Figure 6.21. It falls into the range of  $2 - 20 \mu s$ . It is less than the range

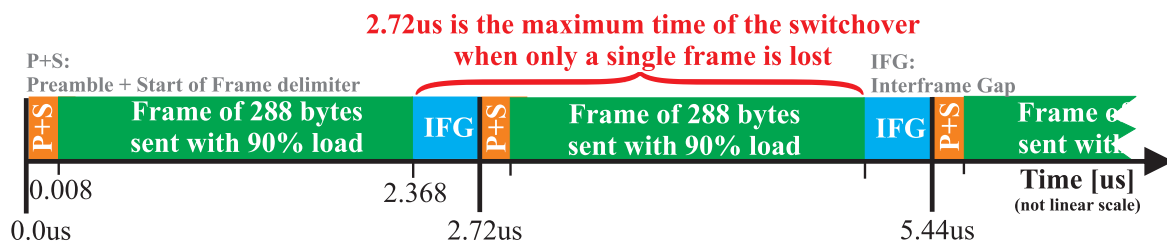


Figure 6.21: Estimation of switchover time based on the test results.

of  $5 - 10 \mu s$  recommended in the proposed strategy (section 4.9) and is 3 orders of magnitude better than any existing solution.

## 6.7 Summary

The tests show that the switchover time and determinism provided by the WR switch fulfil the initial requirements, though improvement is still possible.

The theoretical variation of latency ( $2\delta = 0.592 \mu s$ ) matches closely test measurements ( $0.6 \mu s$ ). The tests indicate that the maximum latency of critical traffic through a single WR switch without any other intervening traffic is  $3.3 \mu s$ . Since WR is commonly used with PTP traffic, the more useful value is  $5.6 \mu s$ , which is maximum latency with intervening PTP traffic. This latency does not depend on the destination port. The test-estimated maximum time of switchover is  $t_{SO} = 2.7 \mu s$ .

These values are much better than the initially required parameters and present a 1000-fold and a four-fold improvement in comparison to the current state of art for the switchover time and the latency through the switch respectively.

The methods developed in this chapter and the previous chapter are used in the next chapter to design a reference WR-based control and timing network for the CERN accelerator complex.

## Chapter 7

---

# WR-Based Control and Timing Network

---

This chapter describes a reference design of a WR control and timing network for the entire CERN accelerator complex. The strategy and mechanisms proposed in the context of this thesis are used to increase the reliability and ensure the determinism of this network. The presented design can be considered a reference in a long-term process of upgrading the current control and timing network at CERN.

The physical design of the network takes into account the spatial distribution of the accelerators and overall locations of the switches and nodes. The other factors considered in the design include the required number of end-nodes, the limit of switch layers and the number of switch ports. This physical design meets the guidelines proposed in subsection 6.2.4 and allows redundant distribution of time and data using the methods proposed in this thesis. While for the time distribution the entire network is logically uniform, for data distribution the network is logically separated using Virtual Local Area Networks (VLANs).

Considering the physical and logic design of the network, its configuration and expected traffic, the performance of the network in terms of reliability, determinism and synchronisation is analysed.

## 7.1 Network Design

The WR control and timing network presented in this section has been designed to serve all of the accelerators in the CERN complex. The design takes into account the spatial distribution of accelerator devices, their interactions and the principles of operation of the current control and timing network.

In terms of control and timing, the CERN complex can be divided into four groups depicted in Figure 7.1:

1. LHC Injection Chain (LIC) – a chain of accelerators that produces beam for the other groups.
2. Large Hadron Collider (LHC) – a single accelerator that receives beam from LIC.
3. Antiproton Decelerator (AD) – a single decelerator that receives beam from LIC.
4. Radiation Beam Experiment (REX) – an experimental facility at Isotope Separator On Line DEtector (ISOLDE) that receives beam from LIC.

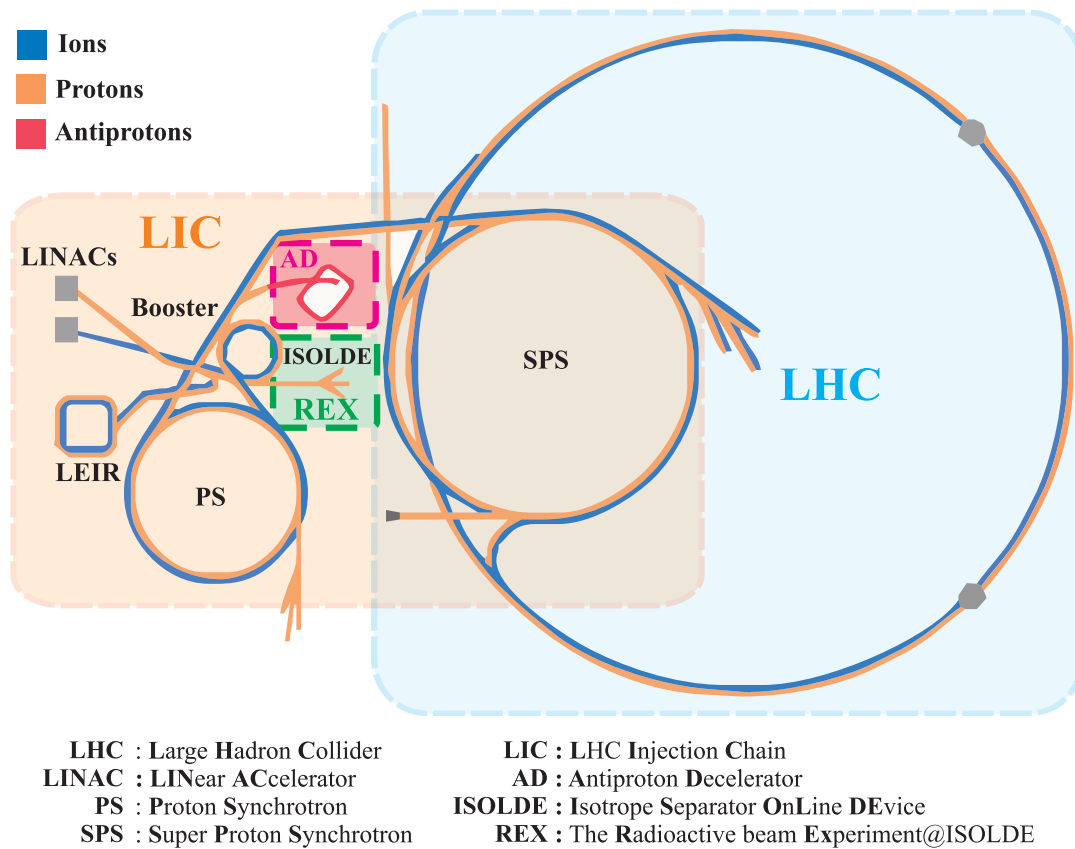


Figure 7.1: CERN accelerator complex.

The LIC accelerators are tightly coupled in sequential actions to produce beams customized for the LIC's clients. On the other hand, the recipients of this beam are loosely coupled with the LIC accelerators. The three clients, LHC, AD, REX, interact with LIC only when receiving the beam but otherwise their operation is independent.

### 7.1.1 Physical Network Design

The described grouping and interactions are the reason to propose a separate controller, a data master, for each group and a single physical WR network for all of the groups. This network is depicted in Figure 7.2. It connects switches and accelerator devices (nodes) dedicated to different groups with the data masters located in the CERN Control Room (CCR). Additionally, connection of auxiliary networks, such as server and LAB networks, has been foreseen in the design. The data masters, core switches and some of the aggregate switches are physically located in the CCR. These switches are connected with different accelerator sites spatially distributed around the complex: the LIC accelerators, the 8 Points around LHC, the AD hall, and the REX experimental facility. The network design allows connection of more than 2000 accelerator devices, as required in the specification. There exist two shortest paths between any of the accelerator devices and any of the data masters. Detailed numbers of switches and nodes are provided in Figure 7.2.

This single physical network is divided logically using VLANs to distribute group-specific data to appropriate accelerators and to allow communication across the groups when needed, as described in section 7.1.3. A common notion of time and frequency is distributed to all the accelerators, as described in section 7.1.2.



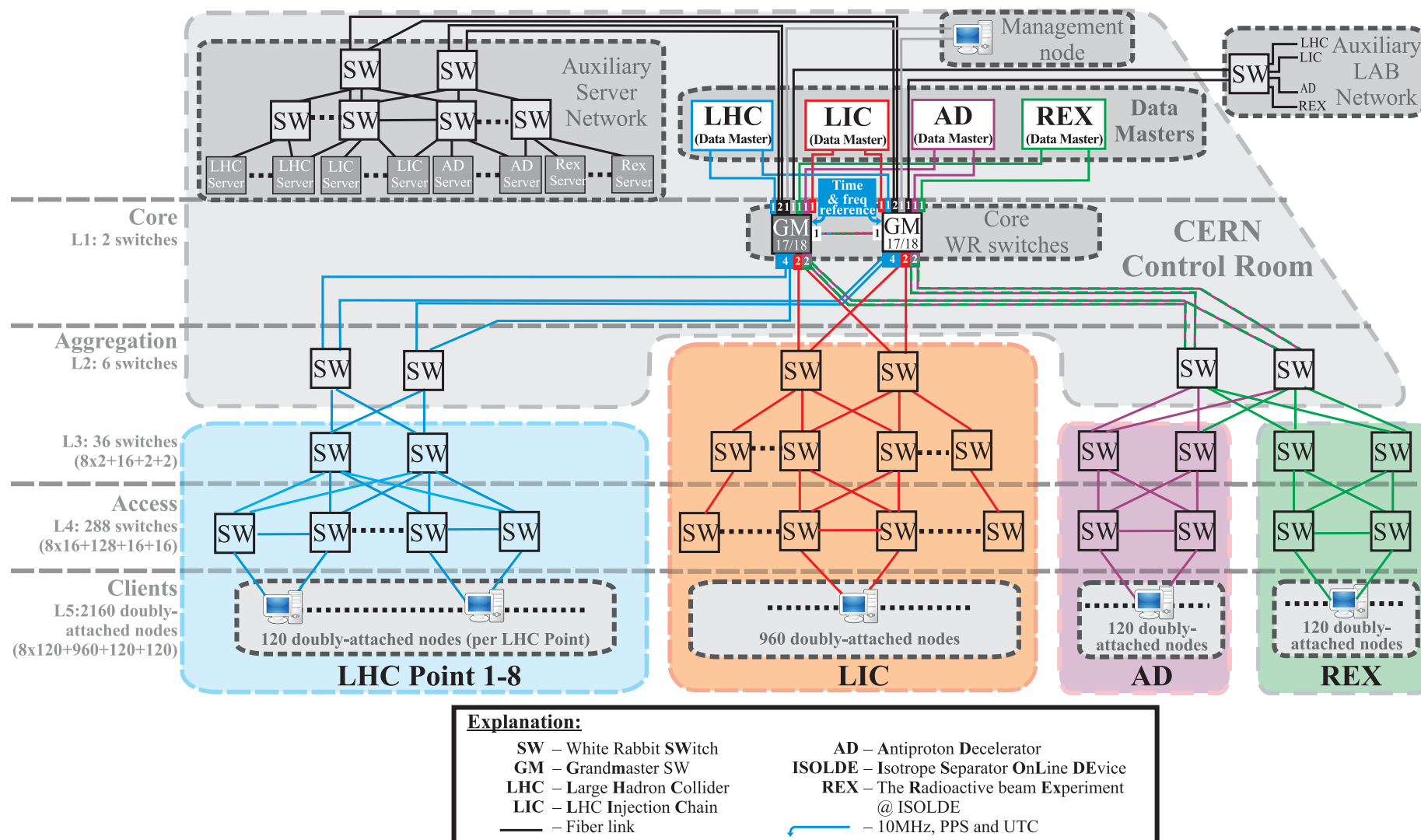


Figure 7.2: Detailed layered design of WR-based control and timing network.

### 7.1.2 Time and Frequency Distribution

The network design presented in Figure 7.2 provides fully redundant distribution of time and frequency to all the accelerator devices. This distribution is explained in Figure 7.3 using a simplified network view.

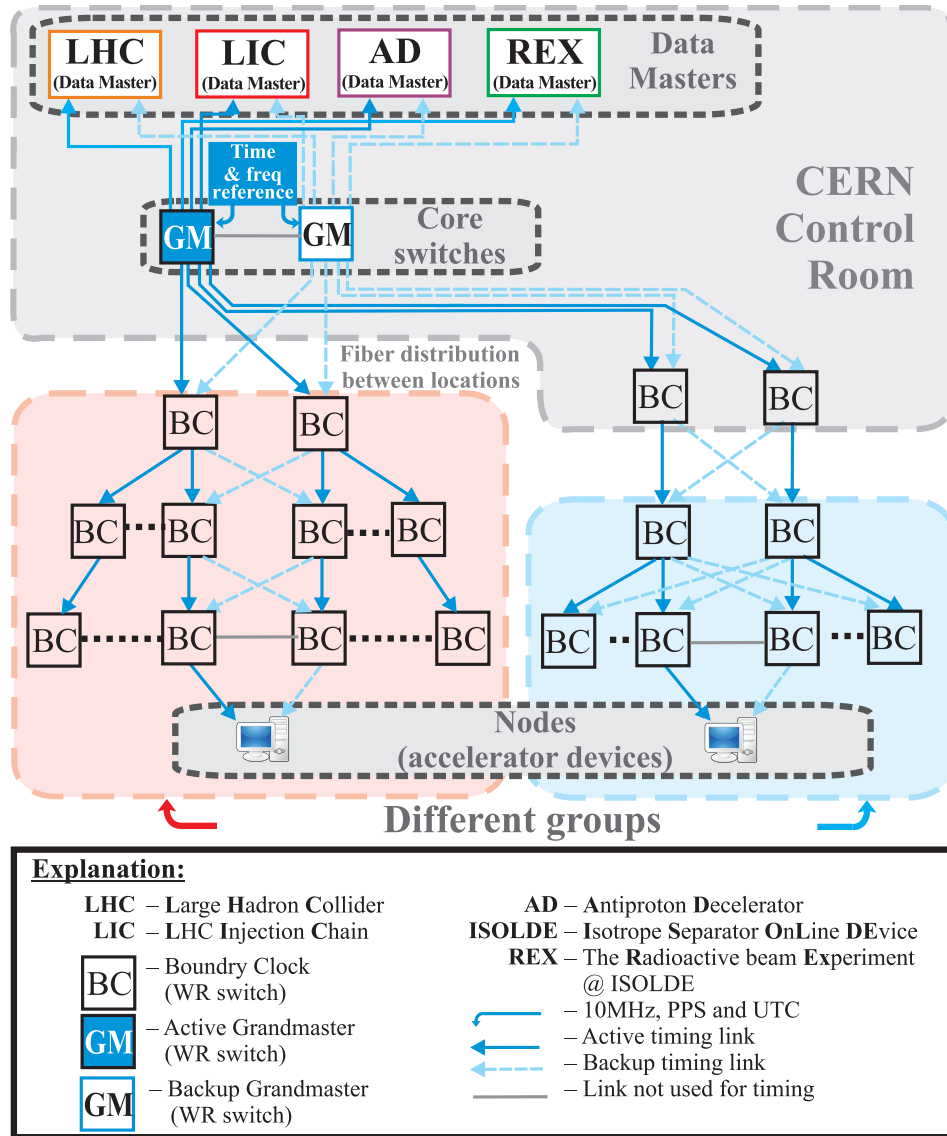


Figure 7.3: Distribution of time and frequency in the WR-based control and timing network.

The two core switches in the network are configured as Grandmaster (GM) connected to a reference of time and frequency. The left GM is the active source of time and frequency in the network. The right GM is a backup. The GMs are not aware of their roles, the choice of active and backup GM is made through proper configuration of the downstream switches and the four data masters. All the downstream switches are configured as PTP Boundary Clocks (BCs). Each BC, each data master and each node in the network is provided with two paths to the GMs: one active and one backup, as depicted in Figure 7.3.

The maximum number of hops from any of the GM to any node in the network is 3 BCs. Without failures, such a chain provides sub-ns synchronisation accuracy. This accuracy might temporarily deteriorate to above nanosecond when switchover occurs. The expected synchronisation performance of the WR control and timing network is discussed in subsection 7.2.1.

The next subsection presents the same physical network from the point of view of data distribution.

### 7.1.3 Data Distribution

Unlike for the time distribution, for the data distribution the WR network is logically separated. This separation reflects the 4 groups, LIC, LHC, AD, REX, interconnected by the network and different types of interactions between them. Logic division of the WR network using VLANs, and data distribution within VLANs are described in this section.

The data master of each group usually transmits control messages that are useful only for the accelerator devices belonging to this group. However, the injection of a beam from LIC to its client requires communication between the group data masters in order to negotiate the parameters of this process. Moreover, during the injection, the LIC data master might need to control accelerator devices of its clients in a different group. These 3 use cases are reflected in the logic separation of the WR network using VLANs. Figure 7.4 depicts the three types of VLANs that are proposed:

- Per-group VLANs to allow communication within the group.
- DM-to-DM VLANs to allow communication between the data masters.
- Shared-group VLANs to allow communication of the LIC data master with accelerator devices belonging to other groups.

The concept of VLANs in the WR network is used in accordance with the Shortest Path Bridging VID Mode (SPB-VID) [36]. This means that a VLAN, a Virtual Local Area Network (LAN), is associated with more than a single VLAN ID (VID). While Base-VID represents a VLAN in the process of configuration by the network administrator, it is translated into a number of shortest path VIDs that are actually used to forward frames through the network. In the case of the concept adapted from SPB-VID in subsection 6.2.4, the pseudo-multipath Ethernet Tree (E-TREE), the Base-VID is translated into two VIDs:

- Trunk-VID that is used to transmit frames from the data master to all the nodes within the VLAN.
- Branch-VID that is used to transmit frames from any of the nodes in the VLAN to the data master.

This is depicted in Figure 7.5 for two per-group VLANs: LIC and REX. The Base-VID for LIC is translated into LIC-T and LIC-B, which are Trunk-VID and Branch-VID respectively.

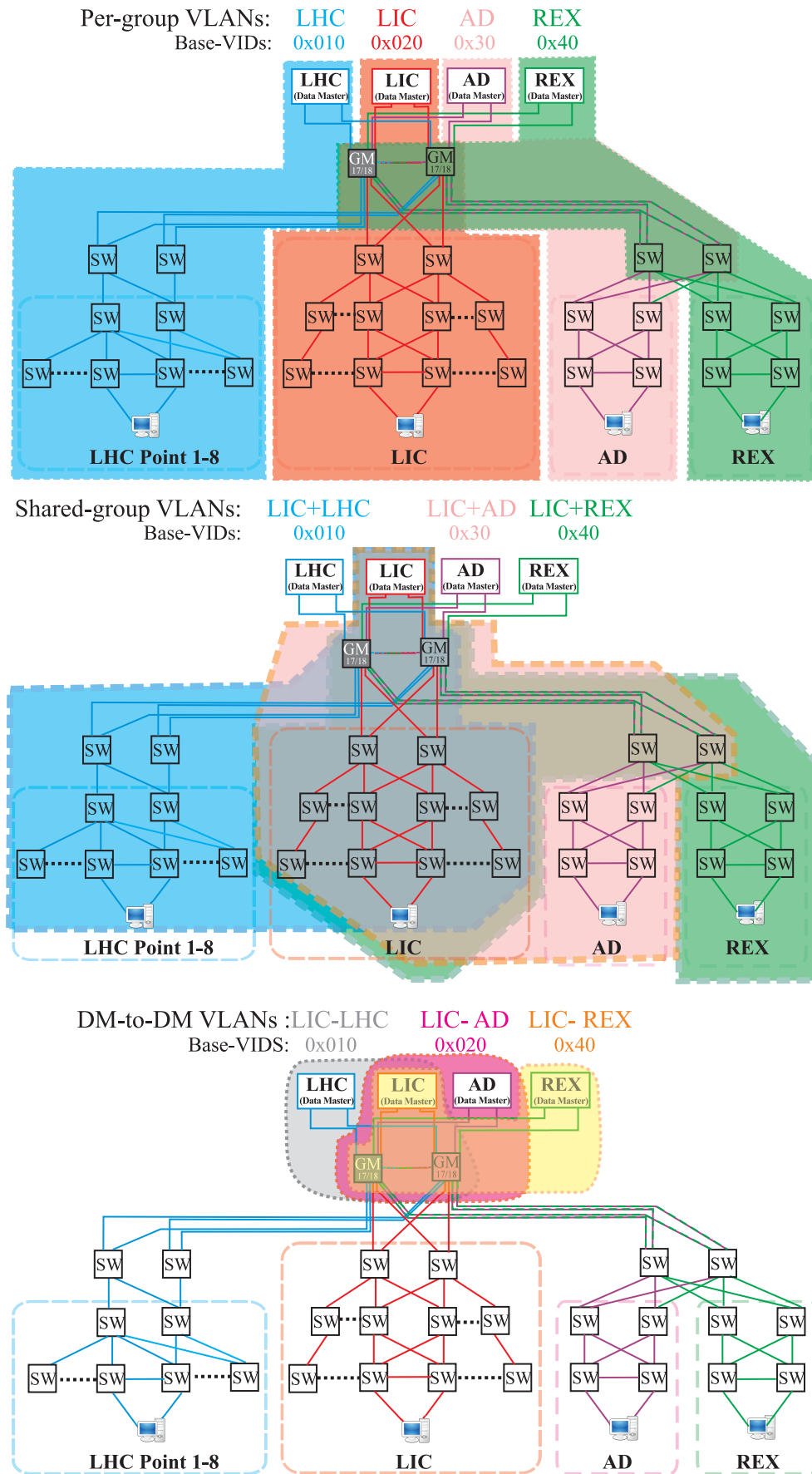


Figure 7.4: VLANs in the WR-based control and timing network.

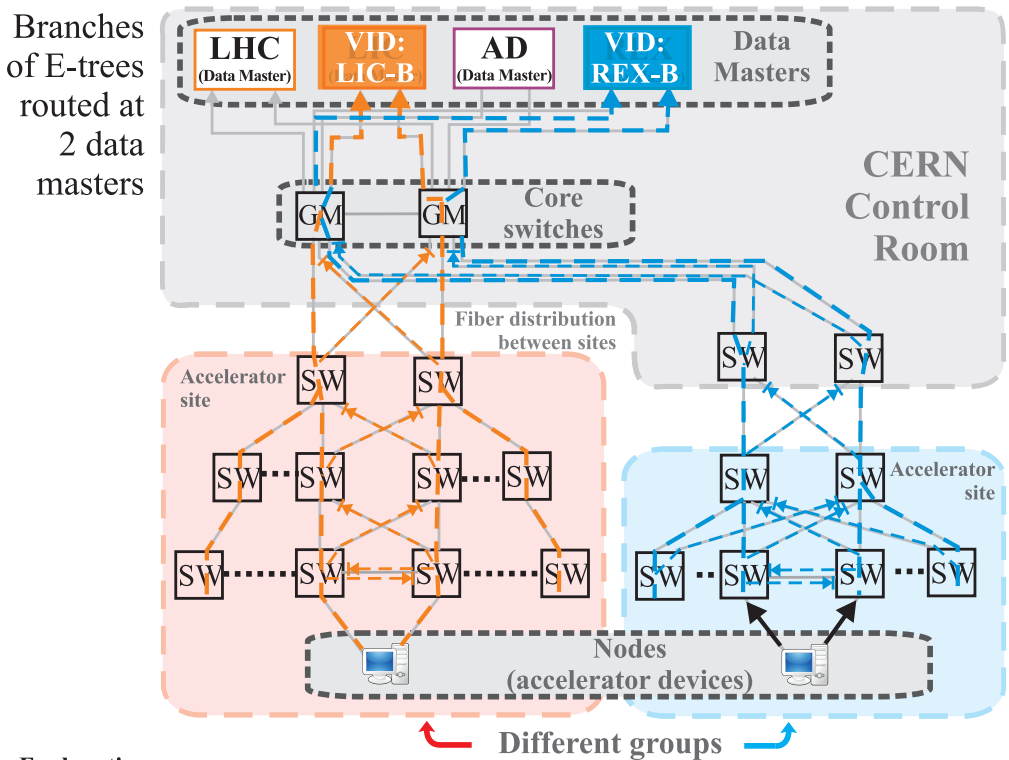
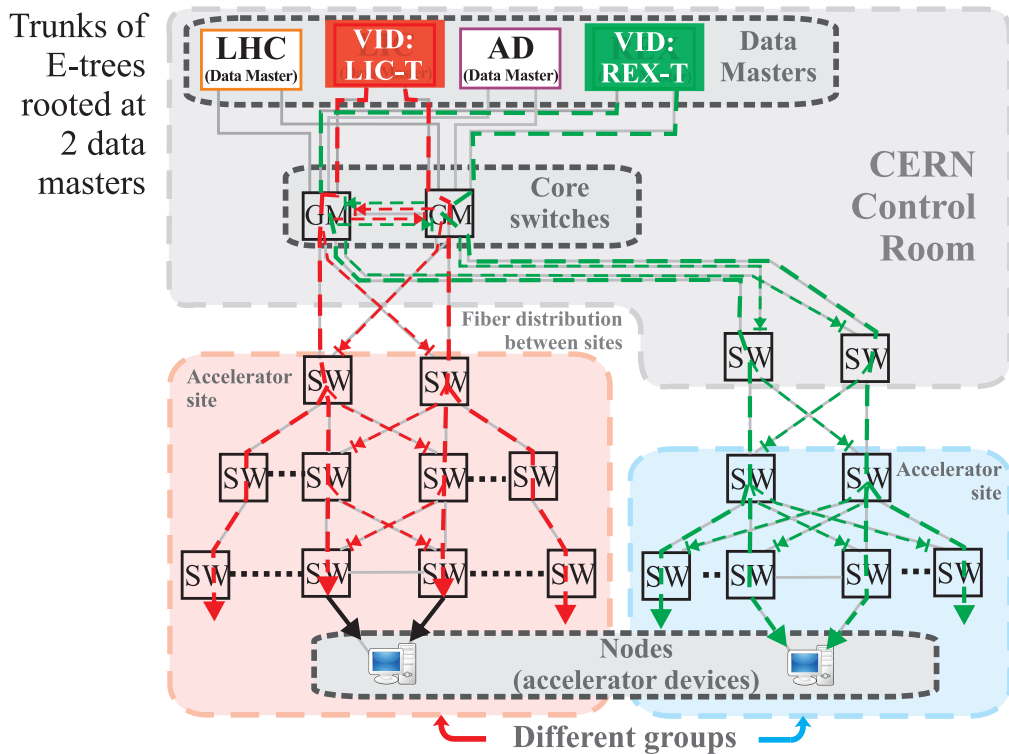
Similarly, the Base-VID for REX is translated into REX-T and REX-B. Each E-TREE is rooted at the data master that sends either:

- tagged traffic with the Trunk-VID, or
- untagged traffic which is tagged with the Trunk-VID by the WR switches.

Similarly, the nodes can either send frames tagged with the Branch-VID or untagged frames.

In the case when communication between nodes is required, instead of the pseudo-multipath E-TREES, pseudo-multicast spanning trees can be calculated, as described in subsection 6.2.4. Each such tree is recognized by a dedicated shortest path VID (SPVID).

Details regarding configuration and management of the WR network in terms of data distribution can be found in Appendix I.



**Explanation:**

SW – White Rabbit Switch  
LHC – Large Hadron Collider  
LIC – LHC Injection Chain

— Fiber link

← - - - Unidirectional Q-tagged discarded stream  
← — Unidirectional untaged stream

GM – Grandmaster SW

AD – Antiproton Decelerator

ISOLDE – Isotope Separator OnLine DEvice

REX – The Radioactive beam Experiment @ ISOLDE

← - - - Unidirectional Q-tagged forwarded stream

VID=R – Root of VID-based spanning tree

Figure 7.5: E-TREE rooted at the data masters and spanning the entire WR network.

## 7.2 Characteristics of the Proposed Network

The proposed network design is evaluated using the results of tests performed in the context of this thesis to estimate the expected characteristics and reliability of the network. This section provides estimated worst-case performance of the network in terms of synchronisation and latency determinism. The estimations are compared with 1-day measurement in a cascade of switches depicted in Figure 7.6. Based on the latency parameters, a suitable schema for Forward Error Correction (FEC) is proposed. Finally, the expected reliability of the entire network is provided.

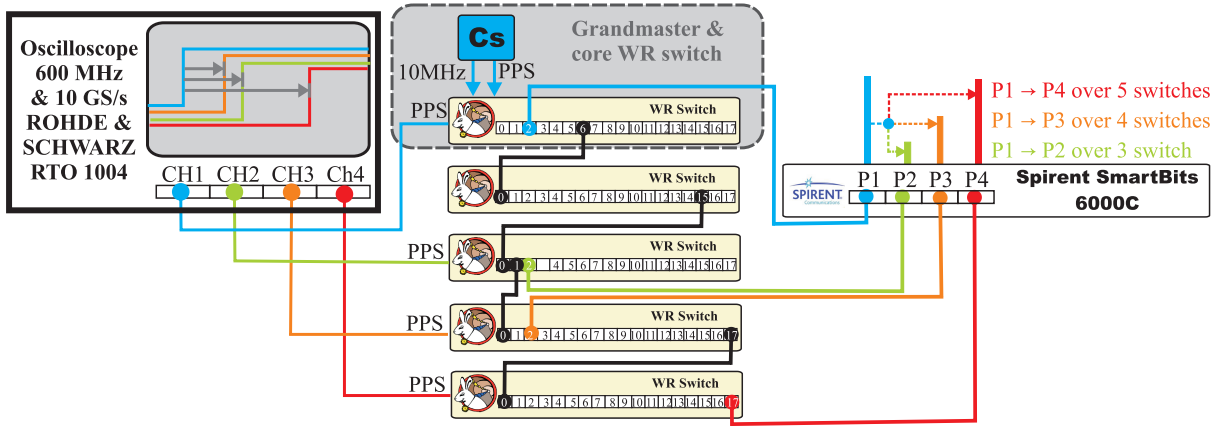


Figure 7.6: Test setup of the proposed WR-based control and timing network design.

### 7.2.1 Synchronisation

The synchronisation tests that have been performed in the context of this thesis are used to estimate the expected synchronisation performance in the proposed WR network design.

In this design, the maximum number of switches between the GM and a node is always 3, regardless of any failures. Tests in steady-state (no failures) presented in section 4.8.1 show that the 4th WR switch in the cascade can expect a precision of  $25ps$  and an accuracy of  $\pm 0.41ns$ . This accuracy can be improved to  $\pm 0.16ns$  if the network is carefully calibrated. The tests performed using the setup in Figure 7.6 showed an accuracy of  $\pm 0.23ns$  and precision of  $31ps$ , without careful calibration. In transient state (during failure), the relevant tests in section 5.6.2 indicate a temporary phase/time deviation during switchover of  $775ps$  when single backup is used. Based on these results, the worst-case accuracy is estimated  $\pm(0.41 + 0.78)ns = \pm 1.19ns$ . With careful calibration, a most likely performance is  $\pm(0.2 + 0.5)ns = \pm 0.7ns$ .

The expected characteristics of synchronisation between GMs and any of the 2000 nodes in the designed WR network over 1 year of operation are presented in Table 7.1. These expectations are compared with the measurement of the setup in Figure 7.6.

It must be noted that the Maximum Time Interval Error (MTIE) can be improved to  $MTIE(\tau = 1 \text{ year}) = 1ns$  if multiple redundancy is used.

Characteristic	Worst-case estimation $\tau = 1 \text{ year}$	Measured $\tau = 1 \text{ day}$
Accuracy calculated as average of time error: $ \text{avg}(TE) $	0.41 ns	0.226 ns
Precision calculated as standard deviation of time error: $\text{sdev}(TE)$	0.03 ns	0.031 ns
Maximum Time Interval Error: $MTIE(\tau)$	2.38 ns	0.408 ns

Table 7.1: Expected synchronisation performance in the proposed WR-based control and timing network.

### 7.2.2 Determinism

The results of the latency tests that have been performed in the context of this thesis are used to estimate the expected transmission latency and its determinism in the proposed WR network design.

The shortest path between a data master and any node in the proposed network consists of 4 switches. This number increases to 5 when one of the core switches fails, as depicted in Figure 7.7. The latencies between the ingress of the first bit to the network ( $L_0$ ) and the egress of the first bit from the third, fourth and fifth switch are presented in Table 7.2. The value of latency variation at the egress of the third switch in normal operation needs to be compensated together with the switchover time, by the FEC schema. The latency presented in Table 7.2 does not include the time it takes to transmit/receive and encode/decode the control message. The estimations assume that the fibres introduce a latency of 50  $\mu s$ . The table compares latency values obtained through:

- calculations using simulation-based estimation of the worst-case switch latency, section 6.3.2
- calculations using test-based estimation of the worst-case switch latency, section 6.6.1
- measurement in a setup presented in Figure 7.6.

Considered at egress of	Latency								
	Simulation-based worst-case estimation			Tests-base worst-case estimation for 1 year			Measurement for 1 day		
	value [ $\mu s$ ]	max [ $\mu s$ ]	pk-pk [ $\mu s$ ]	value [ $\mu s$ ]	max [ $\mu s$ ]	pk-pk [ $\mu s$ ]	value [ $\mu s$ ]	max [ $\mu s$ ]	pk-pk [ $\mu s$ ]
3rd switch ( $L_3$ )	$59.4 \pm 2.22$	61.7	4.7	$62.6 \pm 4.2$	66.8	8.4	$62.8 \pm 2.00$	62.8	4.0
4th switch ( $L_4$ )	$62.6 \pm 3.11$	65.6	6.2	$66.8 \pm 5.6$	72.4	11.2	$63.6 \pm 2.15$	65.7	4.3
5th switch ( $L_5$ )	$65.6 \pm 3.89$	69.5	7.8	$71.0 \pm 7.0$	78.0	14.0	$66.4 \pm 2.30$	68.7	4.6

Table 7.2: Latency values estimated and measured for the WR-based control and timing network.

The latency at the egress of the 4th switch is the network latency during normal operation. When one of the core switches fails, the latency through the network is the latency at the egress of the 5th switch, thus the worst-case latency throughout a year of operation. It is safe to say that the worst possible latency through the network will always stay below 80  $\mu s$  while usually



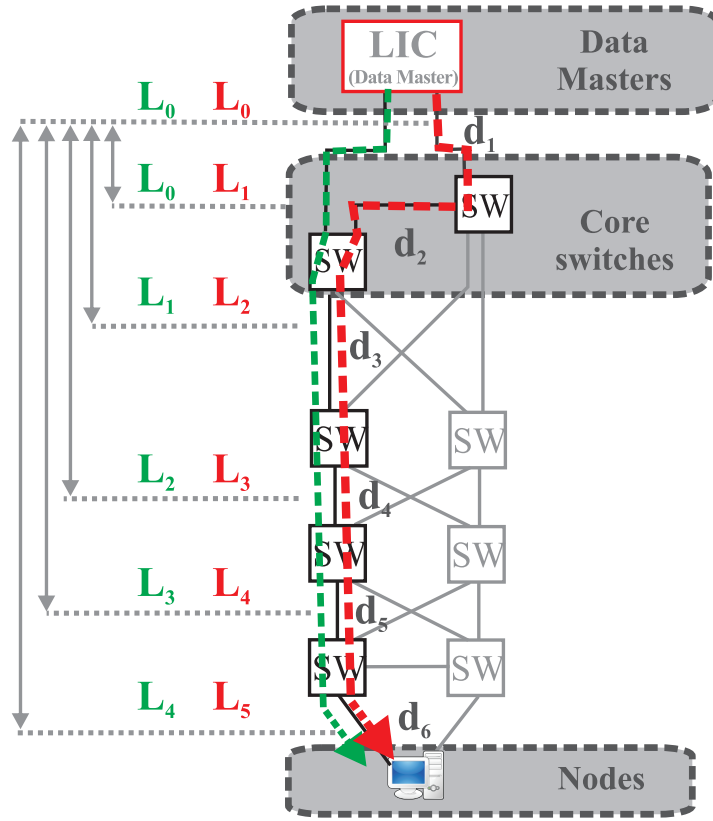


Figure 7.7: Latency through the WR-based control and timing network.

it should be around  $63 \mu s$ , as in the 1 day measurement. Interestingly, the measured variation of the latency is much smaller than the worst-case estimate.

The above analysis, however, are performed assuming transmission of critical traffic by a single data master. In the proposed design, there are 4 data masters that communicate over common parts of the network, the core switches. Clashes in critical traffic from different data masters would significantly increase the latency and its variation and thus must be avoided. It is possible to do so by making an assumption that the new control and timing system will inherit from the old one a millisecond scheduling. It is translated to distribution of control messages in the WR network by assuming that a control message is sent every  $1ms$  by each data master. It is therefore proposed to divide the transmission time into  $200 \mu s$  timeslots, one for each data master and an empty one.

### 7.2.3 Seamless Redundancy

Seamless redundancy is achieved by compensation of frame losses due to switchover and latency variation by using Forward Error Correction (FEC). FEC encodes a control message into  $N$  original and  $M$  parity FEC frames. The control message can be recovered even if any  $M$  of the  $(N + M)$  FEC frames are lost. To ensure that no control message is lost during switchover, the time it takes to transmit  $(M - 1)$  parity frames must be sufficient to cover the time of switchover and the time of latency variation. The former has been measured in subsection 6.6.2 to be  $2.7 \mu s$ . The latter, the latency variation, is estimated in the previous section.

The biggest variation of latency in a normally operating network is experienced by frames at the egress of the 3rd switch. This variation is estimated to be  $8.4 \mu s$ , see Table 7.2. Knowing the expected latency variation and the switchover time, a sufficient FEC configuration can be found to cover switchover and jitter time of  $\leq 10.1 \mu s$ . Such configurations are provided in Table 7.3.

Control message size	Forward Error Correction or simple replication			Time for switchover and jitter	Control message worst-case network latency over 5 switches	Probability that none or 1 message is lost in 1 year	Expected number of messages lost per year
	Block size	Orig frames	Parity frames				
[bytes]	[bytes]	[number]	[number]	[ $\mu s$ ]	[ $\mu s$ ]	[probability]	[number]
600	600	1	-	-	74.5	0.00000(0)	$4.08e + 06$
			1	-	79.7	0.9996743	$2.57e - 02$
			3	10.4	90.0	0.99999(9)	$1.02e - 18$
			4	15.5	95.1	0.99999(9)	$6.47e - 27$
1200	1200	1	-	-	79.3	0.00000(0)	$7.96e + 06$
			1	-	89.3	0.9955026	$9.80e - 02$
			2	10.0	99.2	0.99999(9)	$1.21e - 09$
	600	2	3	10.4	95.1	0.99999(9)	$5.12e - 18$
1492	1492	1	-	-	81.6	0.00000(0)	$9.90e + 06$
			1	-	93.9	0.9898181	$1.50e - 01$
			2	12.4	106.2	0.99999(9)	$2.32e - 09$
			3	24.8	118.5	0.99999(9)	$3.55e - 17$
	746	2	3	12.8	100.9	0.99999(9)	$1.20e - 17$
			4	19.1	107.3	0.99999(9)	$1.13e - 25$
3000	3000 (see Note 1)	1	-	-	93.7	0.00000(0)	$1.96e + 07$
			1	-	118.1	0.8800753	$5.94e - 01$
			2	24.4	142.4	0.99999(9)	$1.80e - 08$
	1500	2	2	12.4	118.8	0.99999(9)	$9.28e - 09$
6000	6000 (see Note 1)	1	-	-	117.7	0.00000(0)	$3.90e + 07$
			1	-	166.1	0.3191056	$2.35e + 00$
			2	48.4	214.4	0.99999(9)	$1.42e - 07$
	1500	4	2	12.4	143.4	0.99999(9)	$4.64e - 08$
Note 1: Jamboo frame used.							

Table 7.3: Parameters and performance of different Forward Error Correction schemas and for a simple replication of frames.

The table shows that the usage of an advanced FEC algorithm is justified only if the control messages are of considerable size. Otherwise, resending the same control message multiple times seems more optimal. For example, if the control message has 1492 bytes, it can be either resend 3 times in the maximum size Ethernet frames or sent using 5 frames of 746 bytes, 3

"parity" and 2 "normal", using sophisticated FEC encoding/decoding. The former seems more optimal.

The table provides also the worst-case latency of control message transmission. This is the time between sending the first bit of the first frame and the last bit of the last frame carrying the control message, assuming a minimum Inter-Frame Gap (IFG). This time in most FEC configurations is below  $120\mu s$ . It is above  $200\mu s$  for one configuration when the biggest possible message size is considered.

Any of the configurations presented in Table 7.3 with black font provides sufficiently high probability that not more than a single control message is lost during the year, assuming 4 data masters sending control messages every 1ms. The configurations in grey font are provided for comparison. They mainly show the dramatic change of transmission reliability using no frame-loss mitigation technique and using one or more parity frames.

The redundancy of data is combined with redundant network topology to provide seamless redundancy. The reliability of the proposed network topology is discussed in the next subsection.

## 7.2.4 Reliability

The reliability of the reference WR network is evaluated analytically using the assumptions and mathematical tools explained in Chapter 4.

The reliability is calculated exclusively for the WR-based control and timing network presented in Figure 7.8. The Auxiliary Server and LAB networks, as well as the data masters and

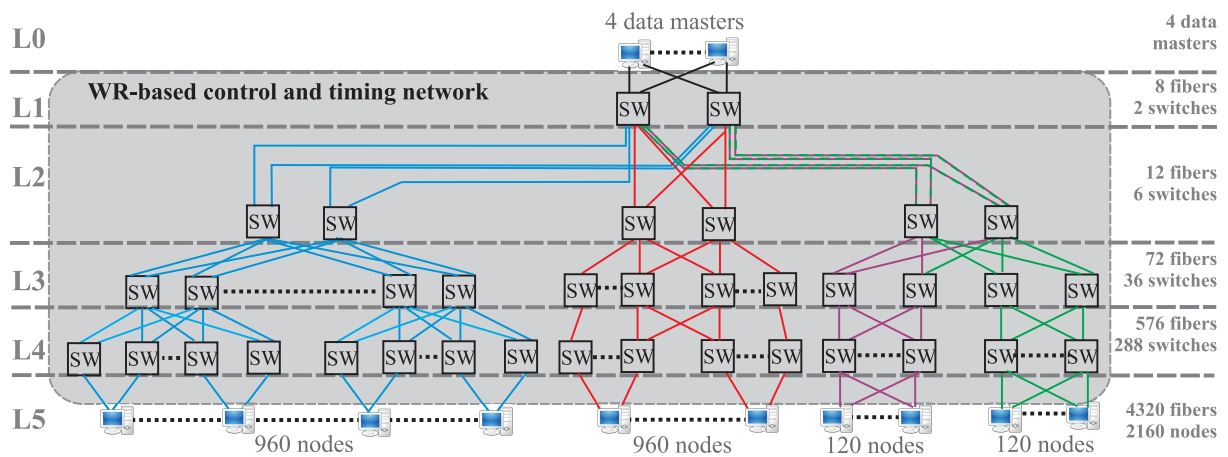


Figure 7.8: Reliability of the WR-based control and timing network.

receiver nodes that are depicted in Figure 7.2 are not included. The calculations are made analogously to those explained in detail in Appendix D. using the values of Mean Time Between Failures (MTBF) assumed in section 4.2: 40 000, 100 000 and 650 000 hours for the WR switch and 3 000 000 hours for the fibre. The Mean Time To Repair (MTTR) for a redundant element and for the network are assumed to be 12h and 4h, respectively.

Table 7.4 shows the calculated reliability and availability values for the reference WR network, as well as the target values established in section 4.3. For the assumed MTTR, both, the

Network consists		MTBF of		Reliability characteristics of the network				MTTR	
Switches [number]	Fibres [number]	Fibre [hours]	Switch [hours]	Reliability [probability]	MTBF [hours]   [years]		Availability [%]	elem. [hours]	net. [hours]
332	4988	3 000 000	40 000	0.9853797	595183	67.9	99.9993279	12	4
			100 000	0.9967167	2665490	304.1	99.9998499	12	4
			650 000	0.9996998	29198283	3330.9	99.9999726	12	8
Minimum target reliability value:				0.9899000	867803	99.0	99.9995	12	4
Ideal target reliability value:				0.9950000	1744419	199.0	99.9995	12	8
MTTR of element optimised									
322	4988	3 000 000	40 000	0.9902282	892680	101.8	99.9995	8	4
			100 000	0.9950794	1777125	202.7	99.9995	18	4
			650 000	0.9950120	1753027	200.0	99.9995	200	8

Table 7.4: Reliability of the reference WR-based control and timing network.

minimum and ideal targets are achieved when the MTBF of the switch is 100 000 or 650 000 hours.

However, the reliability of the network does not only depend on the reliability of individual elements but also on the time a network needs to be operational when such a redundant element fails. During this time, failure of another element is much more likely to cause failure of the entire network. The calculations show that a low MTBF of elements can be, to some extent, compensated by fast replacement of redundant elements that fail. On the other hand, higher reliability of elements gives more time for their replacement. The table provides MTTR values optimised so that the corresponding reliabilities match closely the target values.

## 7.3 Summary

The proposed reference WR-based control and timing network meets all but one initial CERN requirement. It provides the required determinism and reliability of data transmission and network connectivity for over 2000 receiver nodes. The synchronisation performance of this network might, in the worst-case, exceed temporarily the required sub-ns synchronisation accuracy, which for most of the receivers is acceptable.

Table 7.5 compares the initial CERN requirements (section 1.1) with the performance of the reference WR network that uses the mechanisms developed in this thesis. The developed mechanisms allow to meet all the initial requirements, including sub-ns synchronisation at all times. However, to do so a triple redundancy would be needed. Since most of the receiver nodes can easily accept the temporary possible small deterioration of accuracy performance, the network was designed with double redundancy. If needed, a third backup can be added for the nodes that need sub-ns at all times.

Name	Requirement	In the reference WR network
Network size: - max distance - number of receivers	10 km 2000	10 km 2160
Synchronisation (Note 1): - accuracy over a year: $ avg(TE) $ - accuracy in transient: $max( TE )$ - precision: $sdev(TE)$	sub-ns  sub-ns  sub-50 ps	0.41 ns  1.19 ns (see Note 2)  31 ps
Control message - allowed size - max lost per year	1200–6000 bytes 1	1200–6000 bytes 1 with probability R(t)
Upper-bound network latency	$< 500 \mu s$ (derived from 1 ms)	$\leq 78 \mu s$ for network (see Note 3) $\leq 150 \mu s$ for control message (see Note 4 and 5)
Total reliability $R(t)$ (see Note 6)	$\geq 0.98$	0.9854 for $MTBF_{switch} = 40\,000\,h$ and $M = 2$ parity frames 0.9967 for $MTBF_{switch} = 100\,000\,h$ and $M = 2$ parity frames 0.9997 for $MTBF_{switch} = 650\,000\,h$ and $M = 2$ parity frames
Availability A	99.999	$\geq 99.999$ for all $MTBF_s$ - element's $MTTR_e = 12\,h$ - network's $MTTR_e = 4\,h$
Note 1: TE stands for time error Note 2: The accuracy might temporarily deteriorate by $\pm 0.78\,ns$ which will not affect the average over a year. Note 3: Latency between transmitting the first bit of the control message by the data master and receiving the first bit of this message by a node. Note 4: Latency between transmitting the first bit of the control message by the data master and receiving the last bit of this message a node. Note 5: The FEC configuration where 6000 byte control message is resent is not taken into consideration. Note 6: $R(t) = R_c(t) \cdot R_l(t) \cdot R_n(t) \cdot R_i(t)$ .		

Table 7.5: Comparison of initial CERN requirements and characteristics of the proposed reference WR-based control and timing network.

The proposed reference WR network and its characteristics presented in Table 7.5 show clearly that the mechanisms developed and implemented in the context of this thesis are applicable in creating a next generation control and timing network for the CERN accelerator complex.

## Chapter 8

---

# Conclusions

---

The White Rabbit (WR) project started in 2008 with the goal of creating for the European Organization for Nuclear Research (CERN) a next-generation accelerator control and timing system that exceeds both CERN's needs and the capability of commonly available standard technologies. Acknowledging the benefits of using standards, the White Rabbit project leaders decided to put extra effort into enhancing the most suitable technologies with specialised services to meet CERN's anticipated needs. Standard compatibility is a means to achieve longevity, maintainability and support for the enhancements, as already proven through a service developed to provide sub-nanosecond synchronisation accuracy<sup>1</sup>. For the purposes of this thesis, two specialised services along with the network design guidelines were developed, implemented and tested. The first service ensures that the critical information to coordinate accelerator actions is delivered to all the accelerator devices within a specified time. The second service increases the probability that the control and timing system works without interruption for at least a year. The third component of this thesis outlines how to design a White Rabbit control and timing network for all CERN accelerators using these two services. Moreover, the experience from this thesis resulted in reflections and recommendations for further work that are included in Appendix J.

The requirements of the two specialised services developed within this thesis are determined by three factors: anticipated future CERN needs, compatibility with a current control and timing system to ensure a smooth transition, and compatibility with existing standards. A study of the current system, as well as of available networking solutions, allowed the author to translate the input requirements into networking terms, propose a suitable network topology, and choose standards for enhancements. The study results show that the required networking performance is unprecedented. They demonstrate that the timely delivery of critical information requires sub-10 microsecond latency through a single switch with peak-to-peak variation at the microsecond level; this is a few-fold improvement compared to the best previous implementation. The results

---

<sup>1</sup>The service to provide sub-nanosecond synchronisation accuracy enhances the Precision Time Protocol (PTP) defined as IEEE1588. It is now commercially available and in the process of standardisation.

also indicate that the needed increase in the probability of uninterrupted operation requires specialised support for network redundancy that improves 1000-fold the performance of the best previous standard implementation. In particular, the results show that the switchover between alternative paths:

- for data transmission needs to take place in a few microseconds, compared to 1 millisecond in the previous implementation
- for synchronisation needs to be performed such that sub-nanosecond accuracy of synchronisation is maintained, compared to 1 microsecond in the previous implementation.

The methods proposed in this thesis enhance existing standards with specialised software and gateway<sup>2</sup> for redundant networks arranged in a multi-path tree topology. The timely delivery of control information, referred to as determinism, is achieved by developing specialised gateway to recognise the critical traffic by using IEEE 802.1Q standard priorities. This gateway gives to the critical traffic strict precedence and forwards it in a timely manner. The measured latency of the critical traffic through the switch is  $(3 \pm 0.3)$  microsecond without and  $(4.2 \pm 1.4)$  microsecond with PTP traffic; in both cases it is better than initially required.

The probability of the White Rabbit network's undisturbed operation, in other words its reliability, has been increased by using redundant paths for synchronisation and transmission of the critical information. The Shortest Path Bridging standard (IEEE 802.1aq) has been enhanced with specialised gateway to switch between the alternative paths in a few microseconds, fast enough so that no more than a few frames with critical information are lost. These frames can be recovered by using Forward Error Correction (FEC) developed in the context of a separate thesis. The specialised support and the FEC therefore provide seamless redundancy of data transmission – the accelerator devices experience no disruption in the transmission of critical data even if a network element fails.

Similarly, the existing White Rabbit extension to the Precision Time Protocol (IEEE 1588) standard has been enhanced with specialised gateway and software to detect failure extremely quickly and switch between synchronisation paths. The maximum temporary deterioration of synchronisation accuracy during such a switchover is  $\pm 0.8$  nanoseconds when using a single backup path. It is slightly worse than required, but this performance can be improved to  $\pm 0.1$  nanoseconds by using multiple backup paths. The reference design of the WR-based control and timing network, described in this thesis, uses a single backup path. Such a single redundancy is used since most of the receiver nodes can accept temporary deterioration at the nanosecond level and the additional cost of a network-wide multiple redundancy is unjustified. Multiple backup paths can be delivered to the receiver nodes that absolutely need the sub-ns accuracy of synchronisation at all times. All the methods developed allow such different levels

---

<sup>2</sup>Gateway describes the configuration of logic gates in a Field Programmable Gate Array (FPGA) chip. This configuration is specified using a Hardware Description Language (HDL) such as the Very High Speed Integrated Circuits Hardware Description Language (VHDL) or Verilog.

of network redundancy; if a single backup path is not sufficient, two or more backup paths can be used to further increase the reliability. The reliability of the network proposed for CERN has been shown mathematically to meet its requirements. The solutions developed for this thesis were tested individually and have yet to be integrated into a single working control and timing network.

The methods proposed in this thesis are based on and enhance the IEEE 802.1 and IEEE 1588 standards, which are constantly evolving. The author accommodated a number of updates to the IEEE802.1-related standards at the early stage of the work and presented his ideas to the newly established Time-Sensitive Networking (TSN) Task Group. As the TSN's methods and those proposed in this thesis follow a similar direction, the author's ambition is that once all the standards are finalised, the White Rabbit network can be upgraded to implement TSN. At the same time, the author is taking part in the evolution of the IEEE 1588 standard, which he has enhanced to provide the reliable synchronisation. His involvement ensures that the proposed methods and the standard are aligned.

As work on this thesis progressed, the number of White Rabbit applications grew beyond all expectations, both inside and outside CERN. The fact that White Rabbit is based on and is interoperable with well-known technologies makes it a preferred generic solution for many control, acquisition and synchronisation problems, while open access to its sources makes it a good platform to try out new ideas. This applies also to the specialised services presented in this thesis. The author originally designed these services for CERN's next generation control and timing system, which will gradually be deployed at the Organization over many years. During the work on this thesis, the latency requirements were tightened to accommodate a second White Rabbit application at CERN, the new Btrain network that distributes the value of the magnetic field. The new Btrain over White Rabbit is currently being deployed and tested in the Proton Synchrotron accelerator; the other accelerators will be upgraded over several years. A third application of the work developed for this thesis is the diagnostics of beam instabilities in the Large Hadron Collider (LHC). A number of other projects at CERN will use White Rabbit and its specialised features.

Outside CERN, the author's proposed network strategy is already being applied in the design of the Large High Altitude Air Shower Observatory (LHAASO) in Tibet, where White Rabbit is to be used for synchronisation and data acquisition. Moreover, the specialised gateway developed in the context of this thesis was purposely designed in a generic manner so it can be used to test new protocol and software ideas and potentially follow the evolution of standards. As a consequence of this design and the open nature of the White Rabbit project, the specialised support developed and described within the context of this thesis is already being used by a number of doctoral candidates to test alternative ideas regarding both the seamless



transfer of critical data<sup>3</sup> and seamless synchronisation<sup>4,5</sup>.

The methods to increase reliability and ensure determinism in a White Rabbit network that has been presented in this thesis are to be deployed at CERN in the coming years, while White Rabbit is being tested, evaluated and installed in many other institutes around the world. In many such places, the methods developed in this thesis will also be used.

---

<sup>3</sup>Cesar Prados, draft title "Deterministic Control Systems", *Technische Universität Darmstadt*, Germany

<sup>4</sup>José Luis Gutiérrez, "Dependable Systems Over Synchronous Network", University of Granada, Spain

<sup>5</sup>Mattia Rizzi, "Flexible Wired and Wireless Systems for Measurement Applications", Univ. of Brescia, Italy

## **Appendix A**

---

### **White Rabbit**

---

This Appedix provides basic information WR [5, 6, 7]. It includes a short explanation of the technologies and standards that are used in WR as well as the architecture of the WR's main component, the WR switch. These technologies, standards and the WR switch's architecture are extended in the context of this theses. Their understanding is essential for the reader.

## A.1 Basic Technologies and Standards

This section provides basic information about the technologies and standards that are used in WR and that are enhanced in the context of this thesis.

A WR network is a **Bridged Local Area Network (LAN) defined in IEEE 802.1D-2004 [16]**. LANs are composed of Layer 2 (L2) switches that interconnect end stations, such as WR nodes, as depicted in Figure A.1. Each end station is unequivocally identified by its Media

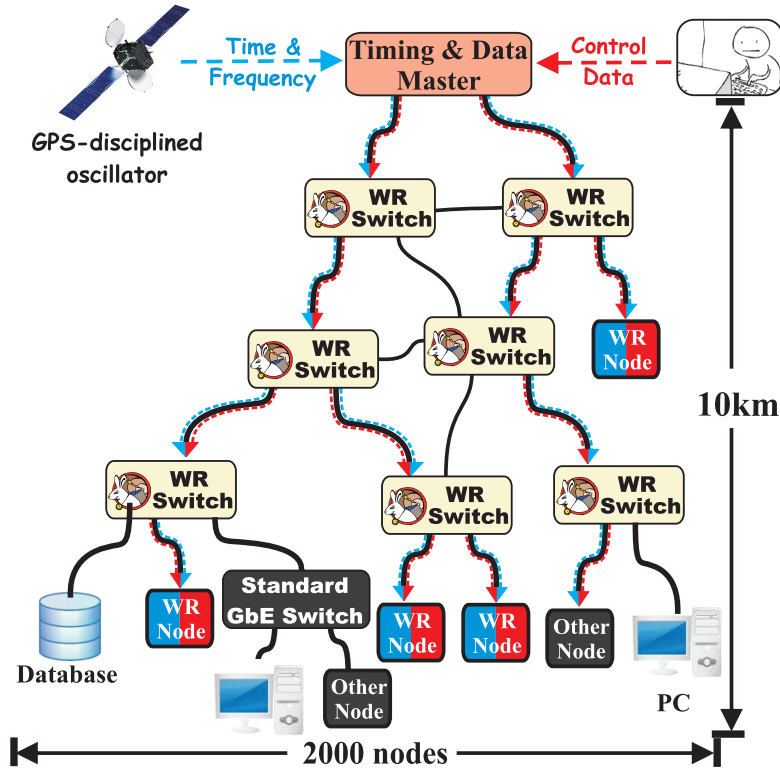


Figure A.1: White Rabbit network.

Access Control (MAC) address. An end station communicates with another end station by sending Ethernet frames with its own MAC address and the MAC address of the other end station, the source and the destination MAC address respectively. The switches forward traffic based on the two MAC addresses contained in the Ethernet frame header. An L2 switch learns forwarding rules by associating the source MAC address of a frame received on its port with that port. This information is then used to forward other received frames to the proper ports according to their destination MAC addresses. If a frame with an unknown destination MAC address is received, it is usually forwarded to all the ports, and its source MAC address is learned.

Ethernet-based LANs are used by the **Precision Time Protocol (PTP) defined in IEEE1588-2008 [19]** to synchronise devices in industrial, telecommunication, measurement and other applications. PTP is a packet-based protocol in which PTP devices establish a tree-like hierarchy of time distribution using information exchanged in *PTP Announce* messages. On

a communication path between two PTP devices, the time of a PTP slave device is synchronised to that of a PTP master device by exchanging PTP messages that are timestamped at reception and transmission. Figure A.2 depicts the exchange of PTP messages<sup>1</sup>:

- The *Sync* message is sent from the master to the slave and timestamped at transmission ( $t_1$ ) and reception ( $t_2$ ).
- The *Follow\_Up* message is used to pass the  $t_1$  timestamp to the slave.
- The *Delay\_Req* message is sent from slave to master and timestamped ( $t_3$  and  $t_4$ ).
- The *Delay\_Resp* message is used to pass the  $t_4$  timestamp to the slave.

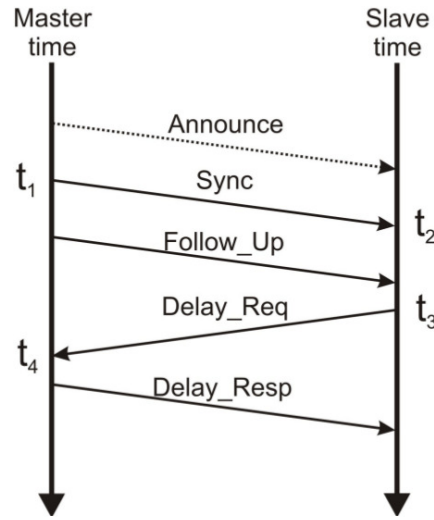


Figure A.2: Standard PTP message exchange.

Assuming medium symmetry, the synchronisation between two devices can be achieved by calculating the one-way link delay

$$link\_delay = \frac{(t_4 - t_2) - (t_3 - t_2)}{2} \quad (A.1)$$

and then correcting the time of the slave for its offset from the master

$$offset\_from\_master = t_2 - t_1 - link\_delay \quad (A.2)$$

The **WR extension to PTP (WR PTP)**, specified in [108] and described in detail in [7], complements the PTP with two ideas: physical Layer 1 (L1) syntonisation<sup>2</sup> and phase detection using a Digital Dual Mixer Time-Difference (DDMTD) scheme.

**L1 syntonisation** allows to transfer frequency with the data stream. The master device encodes the transmitted data using the reference clock signal (frequency). This clock signal is

<sup>1</sup>As the two-step delay request-response mechanism is used in WR, it is depicted in the figure and described. Alternative scenarios of exchanging PTP messages exist, e.g. peer-to-peer mechanism.

<sup>2</sup>Syntonisation is a synchronisation of frequency. Two clock signals are syntonised if their frequencies are identical.

recovered from the received data by the slave device, downstream the network hierarchy. This process is repeated in a hierarchical manner providing a network-wide traceability of frequency to a common reference. Such L1 syntonisation usually provides much better frequency transfer than syntonisation based on the PTP message exchange.

**The Digital Dual Mixer Time-Difference** phase detector is a digital implementation of the analogue Dual Mixer Time-Difference phase detector [109, TN-16]. It uses digital mixing to produce output clock signals of lower frequency than that of the input clock signals. The operation of DDMTD is explained in Figure A.3. The input clock signals,  $clk_{Ain}$  and  $clk_{Bin}$ , are

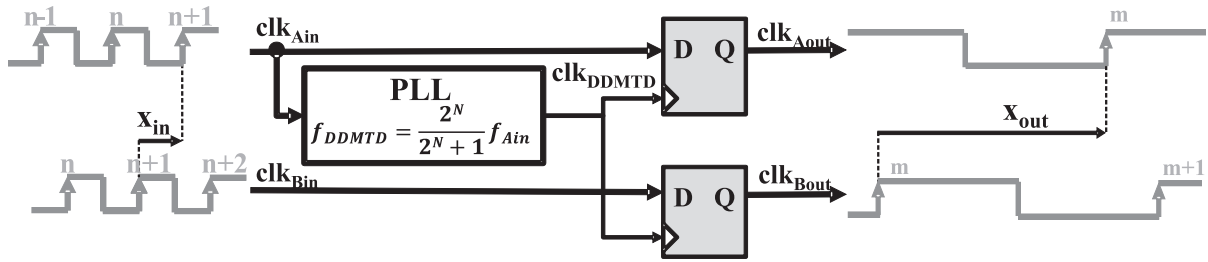


Figure A.3: Digital Dual Mixer Time Difference phase detector.

sampled with D-type flip-flops that are clocked with an offset clock signal,  $clk_{DDMTD}$ , generated from one of the inputs. Its frequency,  $f_{DDMTD}$ , is very close to that of the input clock signal,  $f_{in}$  and is specified as follows:

$$f_{DDMTD} = \frac{2^N}{1 + 2^N} \cdot f_{in} \quad (A.3)$$

where N is an implementation-specific value that is 14 in WR. The sampling operation performed by the flip-flops is similar to analog mixing and low-pass filtering. Thus, the output clock signals,  $clk_{Aout}$  and  $clk_{Bout}$ , are of a frequency that is much lower than and proportional to the frequency of the input clock signals,  $clk_{Ain}$  and  $clk_{Bin}$ . The phase, expressed in radians, between the input signals is equal to that between the output signals. Therefore, the time-difference between the edges of the input and output clock signals is proportional and can be expressed as follows:

$$x_{in}[ns] = \frac{1}{1 + 2^N} \cdot x_{out}[ns] \quad (A.4)$$

The phase between the output clocks  $clk_{Aout}$  and  $clk_{Bout}$  can be precisely detected by a counter running at the  $f_{in}$  frequency and easily converted to the phase time-difference between the input  $clk_{Ain}$  and  $clk_{Bin}$ . This zooming effect allows phase detection with picosecond precision in the WR devices.

All the described technologies and standards are implemented in the main component of a WR network, the WR switch. Its architecture is described in the next section.

## A.2 White Rabbit Switch

Figure A.4 presents the architecture of a WR switch that constitutes an input to this thesis. The methods developed by the author enhance the switch's architecture and implementation, mostly gateware, to provide two specialised services: determinism and reliability. This section briefly explains the architecture and operation of the WR switch which are relevant to this thesis.

The architecture of the WR switch is divided into two parts that execute different types of tasks:

- Time-critical tasks – implemented as gateware (G/W) – are executed in the Xilinx FPGA<sup>3</sup>.
- Non-time-critical tasks – implemented as software (S/W) – are executed in the ARM CPU.

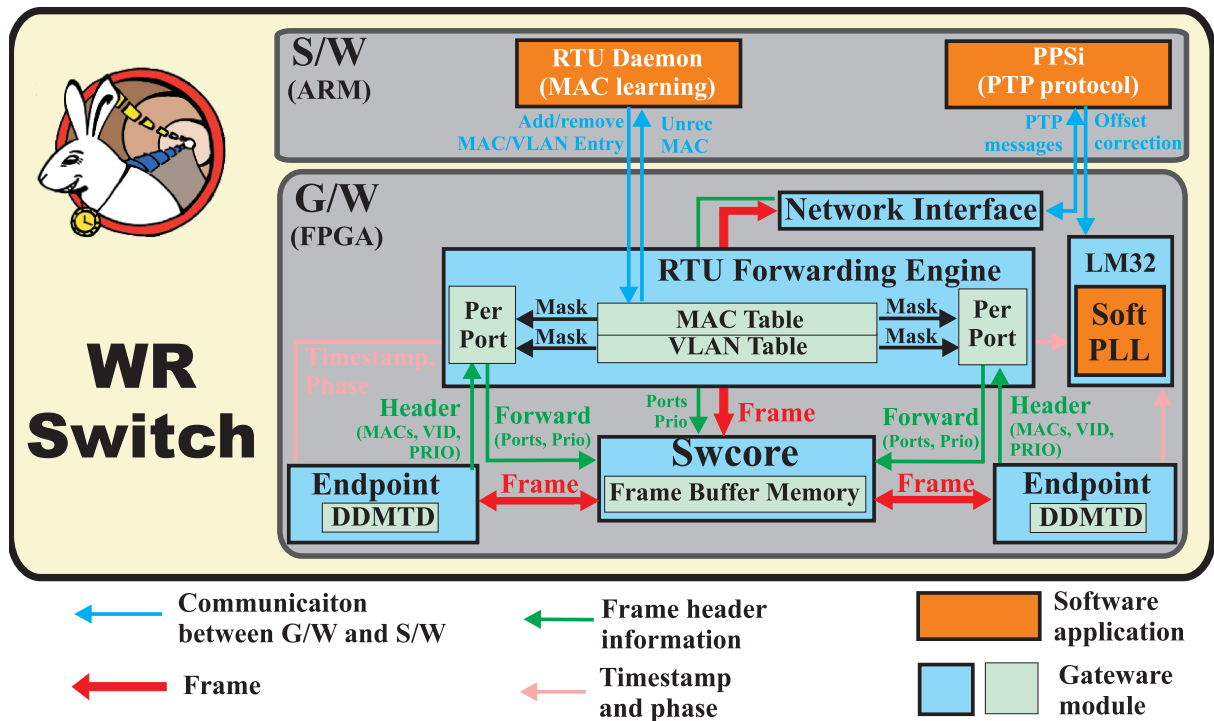


Figure A.4: Simplified architecture of a two-port WR switch including elements taking part in the forwarding of Ethernet traffic between ports.

Similarly to any standard L2 switch, the WR switch forwards Ethernet frames between its ports. An Ethernet frame sent to one of the switch's ports is received by the component called Endpoint that extracts its header information. The Endpoint provides this information (i.e. MACs, VLAN ID, priority) to the RTU Forwarding Engine (RTU) while passing the frame for storage to the Swcore Multi-Access Memory (Swcore). The Swcore is a multi-port and multi-access memory that allows all the ports to write and read frames simultaneously. When the decision from the RTU is ready, it is passed directly to the Swcore that starts providing frames to the appropriate Endpoints for transmission.

<sup>3</sup>FPGA is a Field Programmable Gate Array chip.

The RTU decides to which ports a frame should be sent based on the entries in its Virtual Local Area Network (VLAN) and MAC Tables. These entries are added and updated by the RTU daemon running in the ARM CPU. New entries in the table are added based on the VLAN configuration by the user and the information provided by the RTU Forwarding Engine to the RTU daemon about new source MAC addresses.

In addition to forwarding frames, and unlike any other L2 switch, the WR switch can synchronise with another WR switch or WR node with sub-nanosecond accuracy. Such synchronisation is achieved by using:

- the DDMTD in each Endpoint, and
- the software implementation of WR phase-locked loop (WR PLL), called SoftPLL, in the LM32 CPU[88] embedded in Field Programmable Gate Array (FPGA), and
- the PTP daemon in the ARM CPU.

The PTP daemon, called PTP Ported to Silicon (PPSi) [110]), sends and receives PTP messages on each of the switch ports through the Network Interface Controller (NIC). These messages are precisely timestamped in the Endpoints with the help of the DDMTD. PPSi uses these timestamps to calculate updates for the SoftPLL that controls the frequency and phase of the WR switch.

The elements not present in the above description and figure include diagnostic and management tools such as a web interface, an implementation of Simple Network Management Protocol (SNMP) and frame counters. These elements are not essential in the context of this thesis.

The switch is an open hardware, gateware and software project that is commercially available. It is at the heart of any WR network.

## Appendix B

---

# Explanation of Requirements for the New CERN Control and Timing System

---

The requirements for the new CERN control and timing system, summarized in Table B.1, are briefly explained in this Appedix.

Requirement name	Value(s)
Network size: maximum distance and number of receivers	10km & 2000
Accuracy & Precision [12]	sub-ns & sub-50ps
Control message size	1200-6000 bytes
Maximum number of control messages lost per year	1
Upper-bound latency through a network & a single switch	1ms & 10 $\mu$ s

Table B.1: Requirements for the new CERN control and timing system.

### Maximum link length: 10km

The maximum distance of 10km comes from the requirement of interconnecting devices in the entire CERN accelerator complex which is contained within a circle of a maximum 10km diameter.

### Number of receivers: 2000

It is an estimation based on the current number of timing receivers which is 1934.

### Accuracy & precision: sub-ns & sub-50ps

This synchronisation performance is few orders of magnitude higher than the one provided by the current timing system. It was exceeding CERN needs at the time of project specification in order to anticipate future requirements. It is currently a valid requirement for applications at CERN and other institutes. These applications include diagnostics of beam instabilities in the Large Hadron Collider (LHC) [13] and synchronisation of cosmic ray detectors in the Large High Altitude Air Shower Observatory (LHAASO) [14].



**Control message size: 1200-6000bytes**

The new WR-based and the old General Machine Timing (GMT) systems are intended to inter-work, thus the control messages sent over the WR network must be interoperable with the GMT timing messages. Based on the current content of a single GMT event, the content of a single WR event is anticipated as follows:

- Address, estimated size: 32 bits
- Timestamps, estimated size: 64 bits
- Event Header (IDs), size: 32 bits
- Event Payload, size: 64 bits

The total size of a single WR event is estimated to be 192 bits (24 bytes). In the current GMT system, there are 7 events generated per each of the 7 distribution networks (a network per machine). This gives 49 events sent out every 1ms. The size of a control message to be used for WR network serving all CERN accelerators would currently require 1176 bytes. However, the current number of events is not sufficient. A desired number of events for the new WR-based system is not defined, the more the better. A five-fold increase, to 245 events, should be a sufficient. This gives a control message size of 5880 bytes.

Therefore, after rounding the numbers, the minimum control message size provided in the requirements equals 1200 bytes and the maximum size is 6000 bytes.

**Maximum number of lost control message: 1 per year**

An elevated requirement for reliability which pushes the limits. There is no known case of the current timing system to malfunctioning due to a lost event.

**Upper bound latency through a timing network: 1ms**

In the current timing system, events are scheduled for the next millisecond. WR must provide such a functionality, thus the latency through the WR network, including the time for transmission, reception and processing, needs to be below 1ms.

**Upper bound latency through a single switch: 10 $\mu$ s**

This requirement is specified by the application of WR to upgrade the Btrain system [15]. Btrain provides real-time distribution of the value of the main bending magnetic field in all the CERN accelerators, except LHC. The most demanding recipient of this distribution system are the radio-frequency cavities which require maximum 10 $\mu$ s latency.

## Appendix C

# Network Reliability and Availability Values for all Considered Topologies

Table C.1 includes results of reliability and availability calculations for all considered non-redundant and redundant topologies and all three considered Mean Time Between Failures (MTBF) values of the WR switch ( $MTBF_{switch}$ ): 40 000, 100 000 and 650 000 hours. For all calculations, the MTBF of the fibre ( $MTBF_{fibre}$ ) is 3 000 000 hours and the assumed value of Mean Time To Repair (MTTR) is 4 hours.

Network	Topology type	MTBF <sub>switch</sub> [hour]	Reliability R <sub>n</sub> (1 year)	Availability [%]	MTBF		Number of	
					[hours]	[years]	switches	fibres
Non-redundant	Tree	40 000	0.00000000000000318	98.47999	259.2	0.0296	126	2126
		100 000	0.00000003200633452	99.21869	508.0	0.0579	126	2126
		650 000	0.00036653102627936	99.64029	1108.0	0.1264	126	2126
	Line	40 000	0.00000000000000204	98.49965	262.6	0.0300	124	2124
		100 000	0.00000003836316817	99.22682	513.3	0.0586	124	2124
		650 000	0.00037875875839872	99.64178	1112.6	0.1269	124	2124
Redundant	Ring	40 000	0.13061858295062079	99.9072	4306.6	0.4913	252	4254
		100 000	0.70507456897575238	99.9841	25085.0	2.8616	252	4254
		650 000	0.98817186054240402	99.9995	736722.3	84.0432	252	4254
	Parallel	40 000	0.66464943168557056	99.9814	21459.2	2.4480	252	4252
		100 000	0.93347463909753481	99.9969	127336.0	14.5261	252	4252
		650 000	0.99752149895820696	99.9999	3532430.3	402.9695	252	4252
	Mesh	40 000	0.98681133610478600	99.99939	660269	75.3216	288	4574
		100 000	0.99699777804916600	99.99986	2915452	332.5864	288	4574
		650 000	0.99971843849711800	99.99999	31129134	3551.1219	288	4574

Table C.1: Reliability and availability for all types of considered topologies and Mean Time Between Failures values.



## Appendix D

# Network Reliability Calculations for Doubly-Redundant WR Network

This appendix explains step-by-step the reliability calculations of the doubly-redundant WR network depicted in Figure D.1. This network can be used to redundantly connected a master node with 2048 client nodes using the mechanisms described in this document. The reliability calculations are performed for the WR network only, i.e. the grey background, and exclude nodes connected to the network.

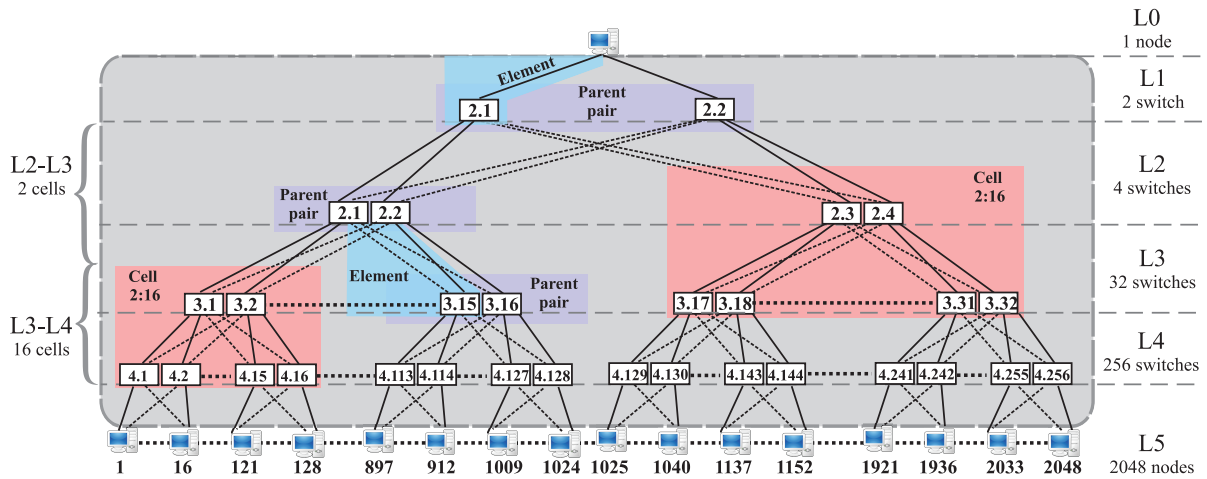


Figure D.1: WR network for which reliability calculation are performed and explained.

The network has regular construction in which three types of building blocks can be distinguished: *element*, *pair* and *cell*. An element contains a WR switch and the upstream fibre(s) connected to it. Elements in a single layer are arranged in pairs. Each pair is redundantly connected with switches or nodes downstream. The network works as long as at least one of the elements in each pair works. The elements belonging to two layers, 2 and 3 as well as 3 and 4, are organised in *cells*. Each cell connects a pair in upper layer with 8 pairs lower layer.

## D.1 Probability Laws Used in the Calculations

The probability calculations are made in probability space  $(\Omega, F, Pr)$ , with sample space  $\Omega$  and  $Pr(\Omega) = 1$ , event space  $F$  and probability measure  $P$ . The following probability laws are used in the calculations:

- **Addition law of probability**

$$Pr(A_1 \cup A_2) = Pr(A_1) + Pr(A_2) - Pr(A_1 \cap A_2) \quad (D.1)$$

where  $A_1, A_2, \dots, A_n$  are events in the same probability space.

- **Conditional probability**

$$Pr(A | B) = \frac{Pr(A \cap B)}{Pr(B)} \quad (D.2)$$

where  $A$  and  $B$  are events from the same probability space  $(\Omega, F, P)$ ,  $Pr(B) > 0$ , and  $Pr(A | B)$  is the conditional probability of  $A$  given  $B$ .

- **Law of total probability**

$$\begin{aligned} Pr(A) &= \sum_n Pr(A \cap B_n) \quad \text{and alternatively} \\ Pr(A) &= \sum_n Pr(A | B_n) \cdot Pr(B_n) \end{aligned} \quad (D.3)$$

where  $B_n : n = 1, 2, 3, \dots$  is a finite or countably infinite portion of a sample space  $\Omega$ , each  $B_n$  is measurable, and  $A$  is any event in the same probability space  $(\Omega, F, P)$ .

## D.2 Symbols Used in the Calculations

$s = Pr(S)$  - switch reliability, the probability of event "S" that switch operates successfully.

$f = Pr(F)$  - fibre reliability, the probability of the event "F" that fibre operates successfully.

$SF_{x,n} = z$  - an event that represents operation of the element "n" at layer "x" (example use cases are depicted in Figure D.2-a):

- If  $SF_{x,n} = 1$ , the element "n" is operational which means that the switch and one or both of the upstream fibres belonging to this element operate successfully. In this case, assuming the parent pair of the element operates successfully, the probability of the event is:  $Pr(SF_n = 1) = Pr(F_1 \cup F_2) \cdot Pr(S) = [f + f - f \cdot f] \cdot s = (2 - f) \cdot f \cdot s$ .
- If  $SF_{x,n} = 0$ , the element "n" is non-operational which means that the switch or both of the upstream fibres belonging to this element do not operate. In this case, assuming the parent pair of the element operates successfully, the probability of the event is:  $Pr(SF_n = 0) = 1 - Pr(SF_n = 1) = 1 - (2 - f) \cdot f \cdot s$ .

$EN_{x,n} = z$  - an event that represents operation of the WR network for a single end node "n" at layer "x":

- If  $EN_{x,n} = 1$ , the WR network provides connectivity between the master node and the end node "n" at layer "x".
- If  $EN_{x,n} = 0$ , the WR network does not provide connectivity between the master node and the end node "n" at layer "x".

$LxP_m$  - an event number "m" that is considered at layer "x", its details are described in an appropriate table.

$PP_{x,\{a,b\}} = z$  - an event that represents operation of parent pair consisting of elements "x.a" and "x.b" at layer "x" (example use cases are depicted in Figure D.2-b):

- If  $PP_{x,\{a,b\}} = 1$ , the pair is operational, element "x.a" and "x.b" operate successfully.
- If  $PP_{x,\{a,b\}} = \frac{1}{2}$ , the pair is partially operational, one particular elements, "x.a" or "x.b", is operational while the other is not operational.
- If  $PP_{x,\{a,b\}} = 0$ , the pair is non-operational, neither "x.a" nor "x.b" operates successfully.

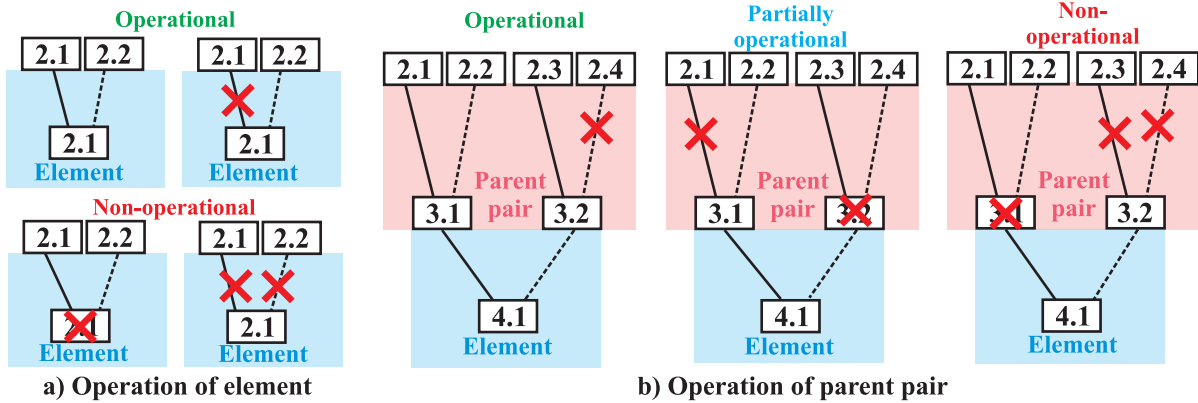


Figure D.2: Example use cases of an element and a parent pair operation.

## D.3 Calculations

This section describes reliability calculations of the WR network depicted in Figure D.1, given the reliability of the switch and fibre are  $s$  and  $p$ , respectively. The calculations are performed for different layers separately. The general flow of the calculations is as follows:

1. First, the probabilities are calculated that a given parent pair in a given layer is fully operational and partially operational.
2. Second, the probabilities are calculated that out of all the pairs in the lower layer,  $N$  pairs are fully or partially operational, given their common parent pair is fully or partially operational.
3. Finally, having the conditional probabilities and the probabilities of the condition, the total probability is calculated. It provides the likelihood that the considered elements provide connectivity.

The following subsections explain in details the calculations.

### D.3.1 Layer 1

At layer 1, there are two elements, 1.1 and 1.2, connected to the master node. The network is operational if both or one of these elements is operational. The appropriate probability calculations are provided in Table D.1.

Event Symbol	Probability formula	Explanation
$L1P_1$	$Pr(SF_{1.1} = 1, SF_{1.2} = 1) = (s \cdot f)^2$	Both elements at L1 are operational
$L1P_2$	$Pr(SF_{1.1} = 1, SF_{1.2} = 0) = (s \cdot f) \cdot (1 - k \cdot s)$	Element 1.1 is not operational
$L1P_3$	$Pr(SF_{1.1} = 0, SF_{1.2} = 1) = (s \cdot f) \cdot (1 - k \cdot s)$	Element 1.2 is not operational

Table D.1: Probability calculations for layer 1 of the redundant network.

### D.3.2 Layer 2

At layer two, each of the two pairs can be operational or partially operational, given their parent pair at layer 1 is either operational or partially operational. The calculation of the probability for each case and the total probability are provided in Table D.2.

Event Symbol	Probability formula	Explanation
Both pairs at L2 are operational		
$L2P_1$	$Pr(PP_{2,\{1,2\}} = 1, PP_{2,\{3,4\}} = 1 \mid L1P_1) = [(2-f) \cdot f \cdot s]^4$	Given 1.1 and 1.2 are operational
$L2P_2$	$Pr(PP_{2,\{1,2\}} = 1, PP_{2,\{3,4\}} = 1 \mid L1P_2) = (f \cdot s)^4$	Given 1.1 is not operational
$L2P_3$	$Pr(PP_{2,\{1,2\}} = 1, PP_{2,\{3,4\}} = 1 \mid L1P_3) = (f \cdot s)^4$	Given 1.2 is not operational
$L2P_{1-3}$	$Pr(L2P_{1-3}) = \sum_{i=1}^3 Pr(L2P_i) \cdot Pr(L1P_i)$	Total probability that all elements at L2 are operational
One particular element at L2 is not operational, thus one particular pair at L2 is partially operational, e.g. 2.1 is not operational		
$L2P_4$	$Pr(PP_{2,\{1,2\}} = \frac{1}{2}, PP_{2,\{3,4\}} = 1 \mid L1P_1) = [1 - (2-f) \cdot f \cdot s] \cdot [(2-f) \cdot f \cdot s]^3$	Given both elements at L1 work
$L2P_5$	$Pr(PP_{2,\{1,2\}} = \frac{1}{2}, PP_{2,\{3,4\}} = 1 \mid L1P_2) = (1-f \cdot s) \cdot (f \cdot s)^3$	Given element 1.1 does not work
$L2P_6$	$Pr(PP_{2,\{1,2\}} = \frac{1}{2}, PP_{2,\{3,4\}} = 1 \mid L1P_3) = (1-f \cdot s) \cdot (f \cdot s)^3$	Given element 1.2 does not work
$L2P_{4-6}$	$Pr(L2P_{4-6}) = \binom{4}{1} \cdot \sum_{i=1}^3 Pr(L2P_{3+i}) \cdot Pr(L1P_i)$	Total probability that one pair at L2 is partially operational. This probability considers $\binom{4}{1}$ use cases.
Two particular elements at L2, not part of the same pair, are not operational, thus two particular pairs are partially operational, e.g. 2.1 and 2.3 are not operational		
$L2P_7$	$Pr(PP_{2,\{1,2\}} = \frac{1}{2}, PP_{2,\{3,4\}} = \frac{1}{2} \mid L1P_1) = [(2-f) \cdot f \cdot s]^2 \cdot [1 - (2-f) \cdot f \cdot s]^2$	Given both elements at L1 work
$L2P_8$	$Pr(PP_{2,\{1,2\}} = \frac{1}{2}, PP_{2,\{3,4\}} = \frac{1}{2} \mid L1P_2) = (1-f \cdot s)^2 \cdot (f \cdot s)^2$	Given element 1.1 does not work
$L2P_9$	$Pr(PP_{2,\{1,2\}} = \frac{1}{2}, PP_{2,\{3,4\}} = \frac{1}{2} \mid L1P_3) = (1-f \cdot s)^2 \cdot (f \cdot s)^2$	Given element 1.2 does not work
$L2P_{7-9}$	$Pr(L2P_{7-9}) = [\binom{4}{2} - 2] \cdot \sum_{i=1}^3 Pr(L2P_{6+i}) \cdot Pr(L1P_i)$	Total probability that both pairs at L2 are partially operational. This probability considers all possible cases that 2 out of 4 elements can be non-operational $\binom{4}{2}$ , excluding two combinations when two elements of a single pair fail and this pair is non-operational.

Table D.2: Probability calculations for layer 2 of the redundant network.



### D.3.3 Layer 3

At layer 3, the probability calculations concern two cells that include elements at layer 2 and layer 3. Each cell consists of a parent pair at layer 2 and 8 pairs at layer 3. The total number of pairs at layer 3 is 16. The probability is calculated given the two parent pairs at layer 2 are both operational, or given one of them is partially operational, or given both are partially operational. The probability is calculated for a number  $N = 0..16$  of partially operational pairs at layer 3. The calculations for each case are provided in Table D.3, where

$PP_{3,\{y,y+1\}}$  is an event that represents operation of a pair at layer 3 where  $y \in Y$  and  $Y$  is set of  $N$  elements that are partially operational

$PP_{3,\{z,z+1\}}$  is an event that represents operation of a pair at layer 3 where  $z \in Z$  and  $Z$  is set of  $32-N$  elements that are operational

and each pair at layer 3 can be partially operational in two cases:  $SF_{3,y} = 0, SF_{3,y+1} = 1$  or  $SF_{3,y} = 1, SF_{3,y+1} = 0$

Event Symbol	Probability formula	Explanation
N pairs at layer 3 are partially operational given both pairs at layer 2 are operational ( $y \in Y$ and $Y$ has $N$ elements, $z \in Z$ and $Z$ has $32-N$ elements).		
$L3P_1, N = 0..16$	$Pr(PP_{3,\{y,y+1\}} = \frac{1}{2}, \dots, PP_{3,\{z,z+1\}} = 1 \mid L2P_{1-3}) = \binom{16}{N} \cdot 2^N \cdot [(2-f) \cdot s \cdot f]^{32-N} \cdot [1 - (2-f) \cdot s \cdot f]^N$	Each of $N$ pairs can be partially operational in two ways, there are $\binom{16}{N}$ use cases.
N pairs at layer 3 are partially operational given one pair at L2 is partially operational and the others is operational Out of $N$ partially operational pairs in layer 3, "i" are under partially operational pair in layer 2.		
$L3P_{2a}, N = 0..8$	$Pr(PP_{3,\{y,y+1\}} = \frac{1}{2}, \dots, PP_{3,\{z,z+1\}} = 1 \mid L2P_{4-6}) = \sum_{i=0}^N \binom{8}{i} \cdot 2^i \cdot (f \cdot s)^{16-i} \cdot (1-f \cdot s)^i \cdot \binom{8}{N-i} \cdot 2^{N-i} \cdot [(2-f) \cdot f \cdot s]^{16-(N-i)} \cdot [1 - (2-f) \cdot f \cdot s]^{N-i}$	$N \leq 8$ , i.e. all $N$ partially operational pairs at layer 3 can be connected to the partially operational pair at layer 2.
$L3P_{2b}, N = 9..16$	$Pr(PP_{3,\{y,y+1\}} = \frac{1}{2}, \dots, PP_{3,\{z,z+1\}} = 1 \mid L2P_{4-6}) = \sum_{i=N-8}^{16} \binom{8}{i} \cdot 2^i \cdot (f \cdot s)^{16-i} \cdot (1-f \cdot s)^i \cdot \binom{8}{N-i} \cdot 2^{N-i} \cdot [(2-f) \cdot f \cdot s]^{16-(N-i)} \cdot [1 - (2-f) \cdot f \cdot s]^{N-i}$	$N > 8$ , i.e. $N-8$ partially operational elements at layer 3 are always connected to the operational pair at layer 2.
N pairs at L3 are partially operational while both pairs at L2 are partially operational		
$L3P_3, N = 0..16$	$Pr(PP_{3,\{y,y+1\}} = \frac{1}{2}, \dots, PP_{3,\{z,z+1\}} = 1 \mid L2P_{7-9}) = \binom{16}{N} \cdot 2^N \cdot [(f \cdot s)^{32-N} \cdot [1 - f \cdot s]^N]$	
$L3P_{1-3}, N = 0..16$	$Pr(L3P_{1-3}, N) = Pr(L3P_1, N) * Pr(L2P_1) + (Pr(L3P_{2a}, N) + Pr(L3P_{2b}, N)) \cdot Pr(L2P_2) + Pr(L3P_3, N) * Pr(L2P_3)$	Total probability that layer 3 provides connectivity between master node and all pairs in layer 4.

Table D.3: Probability calculations for layer 3 of the redundant network.

### D.3.4 Layer 4

At layer 4, the probability calculations concern a single cells XY that include elements at layer 3 and layer 4. The cell XY consists of parent pair  $3.x$  and  $3.(x+1)$  at layer 3 and elements  $4.y$  to  $4.(y+16)$  at layer 4. The probability is calculated for a number  $N = 0..8$  of partially operational pairs at layer 4, given the pair at layer 3 is operational or partially operational. The calculations for each case are provided in Table D.4, where

$PP_{3,\{x,x+1\}}$  is an event that represents operation of a pair at layer 3, it is the parent pair in the cell XY

$PP_{4,\{y,y+1\}}$  is an event that represents operation of a pair at layer 4 where  $y \in Y$  and  $Y$  is set of  $N$  elements in cell XY that are partially operational

$PP_{4,\{z,z+1\}}$  is an event that represents operation of a pair at layer 4 where  $z \in Z$  and  $Z$  is set of  $16-N$  elements in cell XY that are operational

and each pair at layer 4 can be partially operational in two cases:  $SF_{4,y} = 0, SF_{4,y+1} = 1$  or  $SF_{4,y} = 1, SF_{4,y+1} = 0$

Event Symbol	Probability formula	Explanation
In a single cell XY, N pairs at L4 are partially operational given the pair at L3 is operational ( $y \in Y$ and $Y$ has $N$ elements, $z \in Z$ and $Z$ has $16-N$ elements).		
$LAP_1, N = 0..8,$	$Pr(PP_{3,\{y,y+1\}} = \frac{1}{2}, \dots, PP_{3,\{z,z+1\}} = 1 \mid PP_{3,\{x,x+1\}} = 1) = \binom{8}{N} \cdot 2^N \cdot [(2-f) \cdot s \cdot f]^{16-N} \cdot [1 - (2-f) \cdot s \cdot f]^N$	
In a single cell XY, N pairs of elements at L4 are partially operational given the pair at L3 is partially operational		
$LAP_2, N = 0..8$	$Pr(PP_{3,\{y,y+1\}} = \frac{1}{2}, \dots, PP_{3,\{z,z+1\}} = 1 \mid (PP_{3,\{x,x+1\}} = \frac{1}{2} \cup PP_{3,\{x,x+1\}} = \frac{1}{2})) = \binom{8}{N} \cdot 2^N \cdot (s \cdot f)^{16-N} \cdot [1 - (s \cdot f)]^N$	There are two cases as one or the other element of the parent pair can be non-operational.

Table D.4: Probability calculations for a single cell XY that has a parent pair at layer 3 and 8 children pairs at layer 4.

### D.3.5 Layer 5 - WR Network Reliability

The entire network operates successfully if all the 2048 nodes at layer 5 are connected through operational fibres to operational or partially operational parent pairs at layer 4. First, the probability of successful operation of 128 nodes connected to a single cell XY is calculated. It is used then to calculate the probability for all the 2048 nodes, thus the entire WR network. The probability calculations are provided in Table D.5, where:

$PP_{3,\{x,x+1\}}$  is an event that represents operation of a pair at layer 3 that is the parent pair in a cell XY and  $x = 1, 3, 5, \dots, 31$

$PP_{4,\{y,y+1\}}$  is an event that represents operation of a pair at layer 4 where  $y \in Y$  and  $Y$  is set of  $N$  elements in cell XY that are partially operational

$PP_{4,\{z,z+1\}}$  is an event that represents operation of a pair at layer 4 where  $z \in Z$  and  $Z$  is set of  $16-N$  elements in cell XY that are operational

and each pair at layer 4 can be partially operational in two cases:  $SF_{4,y} = 0, SF_{4,y+1} = 1$  or  $SF_{4,y} = 1, SF_{4,y+1} = 0$

Event Symbol	Probability formula	Explanation
WR network is operational for all 128 end nodes connected to a single cell XY given N pairs at layer 4 are partially operational		
$LSP_1, N = 0..8$	$Pr(EN_{1...128} = 1 \mid PP_{4,\{y,y+1\}} = \frac{1}{2}) = f^{16 \cdot N} \cdot [(2 - f) \cdot f]^{16 \cdot (8 - N)}$	
WR network is operational for all 128 end nodes connected to a single cell XY given the pair at L3 is operational		
$LSP_2, N = 0..8$	$Pr(EN_{1...128} = 1 \mid PP_{3,\{x,x+1\}} = 1) = \sum_{i=0}^8 Pr(LSP_1, N) \cdot Pr(LAP_1, N)$	
WR network is operational for all 128 end nodes connected to a single cell XY given the pair at L3 is partially operational		
$LSP_3, N = 0..8$	$Pr(EN_{1...128} = 1 \mid (PP_{3,\{x,x+1\}} = \frac{1}{2} \cup PP_{3,\{x,x+1\}} = \frac{1}{2})) = \sum_{i=0}^8 Pr(LSP_1, N) \cdot Pr(LA_2, N)$	
WR network is operational for all 2048 nodes connected to single 16 cells given N pairs at L3 are operational and 16-N pairs are partially operational		
$LSP_4, N = 0..16$	$Pr(EN_{1...2048} = 1 \mid N \text{ pairs at layer 3 are partially operational}) = Pr(LS_2)^N \cdot Pr(LS_3)^{16-N}$	
$LSP_{all}$	$Pr(EP_{1...2048}) = \sum_{i=0}^{16} Pr(LSP_4, N = i) \cdot Pr(L3P_{1-3}, N = i)$	The total probability – the reliability of the WR network in Figure D.1

Table D.5: Probability calculations for layer 5 – the reliability of WR network.

## D.4 Probability Calculations Results and their Interpretation

The probability that the WR network is operational for 2048 nodes has been calculated using the described method for representative values of switch reliability.

Step	Formule	Explanation	Calcauted values for representative MTBFs		
0	$MTBF_{switch}$	Mean time between failure (MTBF) of a switch	40 000h	100 000h	650 000h
	$MTBF_{fibre}$	Mean time between failure (MTBF) of a fibre	3 000 000h		
	$MTTR = 12h$ Mean time to repair/replace (MTTR) a redundant a switch/fibre when the network operates using its spare. Once the broken switch/fibre is replaced, the network is fully redundant again.				
1	$R_{switch}(MTTR) = e^{-\lambda t} = e^{-\frac{MTTR}{MTBF_{switch}}}$	Probability that a switch is operational over MTTR	0.999700045	0.999880007	0.999981539
	$R_{fibre}(MTTR) = e^{-\lambda t} = e^{-\frac{MTTR}{MTBF_{fibre}}}$	Probability that a fibre is operational over MTTR	0.999996000	0.999996000	0.999996000
2	$R_{network}(MTTR) = Pr(L5P_{all})$	Probability that the entire network is operational over MTTR	0.999981826	0.999995884	0.999999615
3	$MTBF_{network} = -\frac{MTTR}{\ln(R_{network}(MTTR))}$	Mean time between failures of the entire network, assuming that any failure of a redundant switch/fibre takes MTTR to repair	660269	2915452	31129134
4	$R_{network}(1\ year) = e^{-\frac{8766\ h}{MTBF_{network}}}$	Probability that network is operational over a year, assuming that any failure of a redundant switch/fibre takes MTTR to repair	0.986811336	0.996997778	0.999718438
5	$A_{network}k = \frac{MTBF}{MTBF+4h}$	Availability of the network, assuming that any switch/fibre failure that causes network failure is fixed in MTTR=4h	0.999993942	0.999998628	0.999999614
$MTTR = 48h$					
3	$MTBF_{network} = -\frac{MTTR}{\ln(R_{network}(MTTR))}$	See description above	82608	364593	3891649
4	$R_{network}(1\ year) = e^{-\frac{8766\ h}{MTBF_{network}}}$	See description above	0.899320657	0.976243550	0.997750020
5	$A_{network}k = \frac{MTBF}{MTBF+4h}$	See description above	0.999951581	0.999989029	0.999998972
$MTTR = 168h$					
3	$MTBF_{network} = -\frac{MTTR}{\ln(R_{network}(MTTR))}$	See description above	47305	259508	3522129
4	$R_{network}(1\ year) = e^{-\frac{8766\ h}{MTBF_{network}}}$	See description abovee	0.830849214	0.966784785	0.997514259
5	$A_{network}k = \frac{MTBF}{MTBF+4h}$	See description abovee	0.999915450	0.999984586	0.999998864

Table D.6: Final calculations of the Mean Time Between Failures, reliability and availability for the redundant network.



## Appendix E

---

# Forward Error Correction Header

---

In the proposed schema, each Ethernet frame carries a single Forward Error Correction (FEC) block that contributes to recovering the original control message:

- N original blocks are equal parts of the original control message.
- M parity blocks are generated from the N original blocks.

Reception of any N out of all the (N+M) blocks allows to recover the original control message. The receiver of an Ethernet frame containing a FEC block must be able to:

- identify the block
- recognise whether any block of a particular control message is lost
- recognise the encoding algorithm that was used to generate the parity blocks
- know the size of the original control message to be recovered.

This information is provided in the header proposed in Table E.1. This header is to be placed between the header of the transport layer used to carry the FEC payload (e.g. the Ethernet Header) and the FEC block in the payload. Effectively, a FEC encapsulation is created.

Information	Starting Bit	Size [bit]
FEC scheme ID	0	4
Fragment ID	4	4
Message ID	8	32
Original length	40	13
Fragment length	53	11

Table E.1: Header proposed to be prepended to the FEC block in the payload of an Ethernet frame.



## Appendix F

# Latency of Control Messages in a WR-Based Control and Timing Network

This Appendix provides detailed description of the control message latency in a WR-based control and timing network. It describes different contributors to this latency, including the configuration of Forward Error Correction (FEC) and other traffic in the network.

The latency requirement of 1ms in Table 1.1 is interpreted as the ability of scheduling events by the data master to be executed in the next millisecond by any receiver node in the WR network. As depicted in Figure F.1, during 1ms, the control message needs to be properly generated and encoded into FEC frames by the data master, transmitted over the WR network, then decoded, interpreted and executed by the receiver node. This thesis is concerned only with the

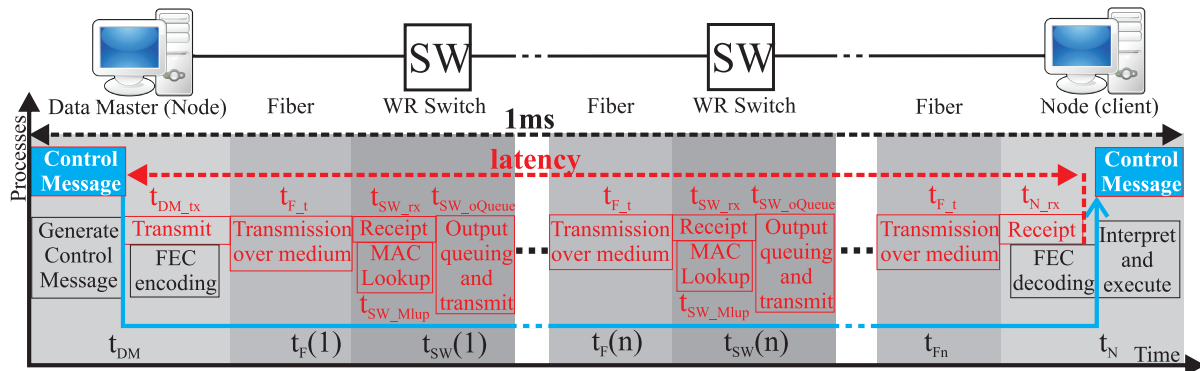


Figure F.1: Contributors to the latency between generating control message by the data master and triggering event at a node.

latency of data transmission over the WR network which is measured between the transmit of the first bit and the receipt of the last bit of the frame(s) carrying a single control message. The contributors to this latency are depicted in red in Figure F.1 and described in details below.

$t_{DM\_tx}$ : **Transmit time of control message.** It depends on the size of the control message,



FEC configuration and spacing between frames. Assuming that the FEC encoding is done on the fly (while sending) and the control message is divided into equal blocks of size  $Block_{size}$  [bytes], the size of each transmitted frame is:

$$F_{L1size} [bytes] = EthOvhd_{size} + FEChdr_{size} + Block_{size} \quad (F.1)$$

where each block is prepended with a FEC header of  $FEChdr_{size}$  [bytes] and the Ethernet overhead of size  $EthOvhd_{size}$  [bytes] includes preamble, start of frame delimiter, header, Virtual Local Area Network (VLAN) tag and Cyclic Redundancy Check (CRC). The transmission time of the entire control message using FEC configuration of  $N$  original blocks and  $M$  parity blocks is

$$t_{DM\_tx} [\mu s] = ((N + M) \cdot F_{L1size} + (N + M - 1) \cdot IFG_{size}) \cdot 8 \cdot 10^{-3} \quad (F.2)$$

where  $IFG_{size}$  is the size of the Inter-Frame Gap (IFG).

$t_F$ : **Transmission time over the medium.** For fibre optic, it is roughly:

$$t_F [\mu s] \approx 5 \left[ \frac{\mu s}{km} \right] \cdot fibre\_length [km] \quad (F.3)$$

In the calculations for CERN, a total length of 10km is assumed, thus  $t_{F\_total} = 50 [\mu s]$ .

$t_{SW\_rx}$ : **Switch receipt time.** A standard Layer 2 (L2) switch forwards a frame only after completing its reception and determining its destination in a Media Access Control (MAC) lookup process ( $t_{SW\_Mlup}$ ). The receipt time is the time of the process that dominates:

$$t_{SW\_rx} [\mu s] = \max(t_{SW\_Mlup}, F_{L1size} \cdot 8 \cdot 10^{-3}) \quad (F.4)$$

The MAC lookup process should, ideally, complete within the receipt time.

$t_{SW\_oQueue}$ : **Queuing time.** The worst-case queuing time depends on the size of the maximum allowed frame ( $F_{max\_size}$  [bytes]) and the maximum possible number of frames in an output queue of the same priority ( $queue_{max}$ ):

$$t_{SW\_oQueue} [\mu s] = (queue_{max} + 1) \cdot (F_{max\_size} + IFG_{min\_size}) \cdot 8 \cdot 10^{-3} \quad (F.5)$$

Latency of transmit and receipt in a serial connection do not accumulate. The transmit of the frame is accounted for in the receipt time of the downstream switch or node.

$t_{N\_rx}$ : **Node receipt time.** In the worst-case scenario, all the FEC frames used to encode a control message need to be received:

$$t_{N\_rx} [\mu s] = t_{DM\_tx} \quad (F.6)$$

The receipt time of a node is accounted for in the transmit time of the data master in the final equation.

The final equation for the total latency of a control message through a network of  $n$  switches is:

$$latency [\mu s] = t_{DM\_tx} + t_{F\_total} + n \cdot (t_{SW\_rx} + t_{SW\_oQueue}) \quad (F.7)$$

## Appendix G

---

### PTP Support Unit

---

The PTP Support Unit (PSU) has been developed by the author to allow fast ( $\sim 1ms$ ) and standard-compatible notification about holdover between WR switches. It is a VHDL module that is placed between the Network Interface Controller (NIC) and the Swcore Multi-Access Memory (Swcore) modules in the switch's Field Programmable Gate Array (FPGA), as depicted in Figure G.1<sup>1</sup>.

Placing the PSU between the Swcore and the NIC allows a single module to handle Precision Time Protocol (PTP) communication on all the switch ports, thus save resources. Any Ethernet communication between one of the 18 ports of the switch and the embedded Linux running on the ARM CPU is performed through a single NIC which is multiplexed between 18 network interfaces exposed to the operating system (wr0 to wr17). The PTP daemon uses these interfaces to transmit and receive PTP messages. All these PTP messages pass through the PSU, in particular two identical *PTP Announce snooper & dropper* sub-modules, placed on transmission and reception paths.

The *PTP Announce snooper & dropper* on the transmission path monitors all the egress traffic and detects PTP Announce messages transported using one of two mappings: Ethernet or User Datagram Protocol (UDP) over Internet Protocol version 4 (IPv4). A detected message is stored in the *TX RAM* to be used as a template for transmission. It is assumed that the same mapping is used on all the ports, in which case the PTP Announce messages sent on all ports contain the same information except the sequence number (seq\_id) and PortIdentity (port\_id). Thus, the common part is saved for all the ports while the sequence number and PortIdentity are saved per-port in *TX RAM*. The memory is organised in banks which are swapped when the entire frame is successfully sent out.

The PSU module receives directly from the SoftPLL an indication that holdover is entered (*holdover\_on*) and a PTP Announce message with degraded clockClass needs to be sent to notify the downstream WR switch. When *holdover\_on* becomes TRUE, the *Frame Injector* sub-

---

<sup>1</sup> A Full switch gateway architecture is available here [111], it does not include the PSU

module reads the stored PTP Announce message template from the *TX RAM* and transmits it with an appropriate sequence number and PortIdentity, and a pre-configured clockClass number. The injected frame is snooped and stored as if it were sent by the PTP daemon. Therefore its sequence number is remembered. When an Announce message with the same sequence number is sent by the PTP daemon, this message is dropped by the *PTP Announce snoop & dropper* sub-module. In this way, the sequence order is preserved on the wire and the average interval time between Announce messages is compliant with the standard.

All the frames received by the NIC are monitored on the reception path by a second *PTP Announce snoop & dropper* sub-module in order to detect PTP Announce messages with the clockClass indicating holdover. The *PTP Announce snoop & dropper* sub-module uses the *RX RAM* to remember the PortIdentity and sequence number of the received messages. This provides verification when detecting the pre-configured clockClass in the PTP Announce messages. When a pre-configured value of the deteriorated clockClass is detected, the PSU module notifies the SoftPLL with a signal that triggers Fault Detection.

The described PSU module allows the delivery of notification about holdover to a downstream switch in an estimated time of 1ms. During 1ms the phase error of a WR switch in holdover is expected to deviate around 10ps. Thus, this mechanism allow to maintain sub-ns synchronisation during holdover notification, even if it needs to be transmitted over a cascade of switches.

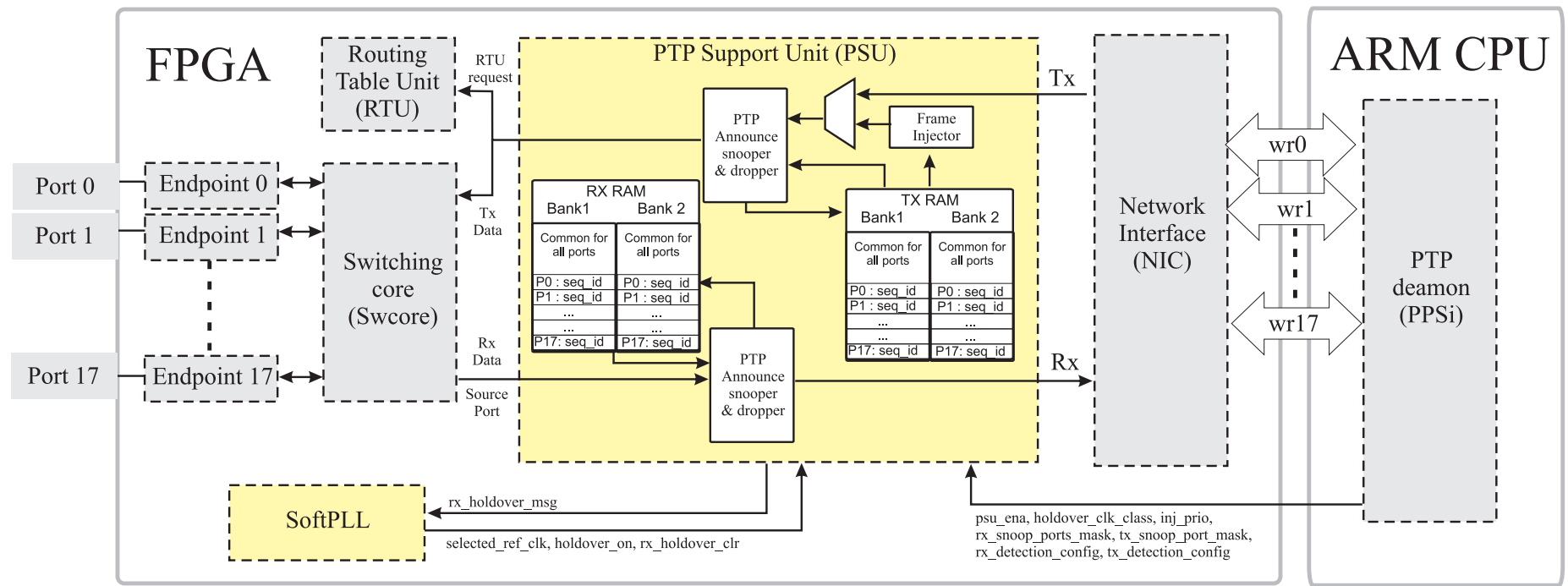


Figure G.1: Architecture of the PTP Support Unit (PSU) and its interactions with other components of the WR switch (elements with yellow background were developed or modified for this thesis).



## **Appendix H**

---

### **Time Switchover Test Results**

---

This Appendix provides detailed results of the time switchover measurements that have been performed by the author and summarised in section 5.6. Refer to section 5.6 for the description of each test scenario.

## H.1 Direct Redundant Connection (scenario a)

Table H.1 presents results of the measurements that are summarized in Figure 5.16 of subsection 5.6.1.

Test 0 in the table present the reference measurement of the mean skew as well as its standard deviation (sdev) and maximum time error (MTE), all measured over a period of 60 seconds. This measurement is performed for each of the ports that are used in this scenario. It includes synchronisation over an attenuator with 0dB attenuation. The reference measurement is used to estimate the *phase jump* in test measurements 1 to 10.

Tests 1 to 10 presents results of synchronisation performance (mean, sdev, MTE) during the switchover measured with the oscilloscope as well as by the Main PLL (mPLL) and the Backup PLL (bPLL). The last two columns in the table provide an estimation of the *phase jump* during switchover.

Test <i>N</i> <sup>o</sup>	Short Info	Failure	Phase skew oscilloscope			Phase err mPLL		Phase err bPLL		Estimated phase jump	
			mean	sdev	MTE	sdev	MTE	sdev	MTE	SoftPLL	Scope
			[ps]			[ps]		[ps]		[ps]	
0	reference p1		7.4	13.0	109						
0	reference p2		-7.6	11.8	102						
0	via atten p2		-25.8	11.5	96						
0	reference p3		-18.1	12.2	96						
1	p3 → p2	discon	-14.5	24.6	402	26.1	447	14.3	117	280 : 337	293
2	p2 → p1	discon	-30.5	29.1	225	14.4	207	28.2	140	157 : 171	116
3	p1 → p2	discon	-30.4	16.0	186	15.3	304	10.4	75	130 : 181	77
4	p1 → {p2,p3}	discon	-11.1	29.7	138	10.3	119	9.8	88	51 : 79	29
5	p2 → {p3,p1}	discon	-48.5	13.0	109	13.0	143	10.6	78	61 : 73	neg
6	p3 → {p1,p2}	discon	-39.4	24.5	152	10.6	126	10.3	69	71 : 78	43
7	p2 → p1	attenu	-11.2	16.5	98	9.3	83	10.8	71	29 : 40	neg
8	p2 → p3	attenu	-42.2	11.9	94	10.4	84	9.7	79	29 : 38	neg
9	p2 → {p3,p1}	attenu	-33.2	14.0	92	10.0	74	10.9	80	24 : 39	neg
10	p2 → {p3,p1}	attenu	-7.1	13.5	85	10.5	81	8.7	62	30 : 40	neg

Table H.1: Measurement results during switchover between direct redundant connections in scenario a (pX: port number X, discon: disconnected, attenu: attenuated, neg: negligible).

Figure H.1 shows the raw phase error data from the SoftPLL. In particular, the input phase error to Proportional-Integral (PI) controller of the mPLL is depicted in the upper plots and the phase error measured by the bPLL is depicted in the lower plots. The input to the mPLL is the phase error  $\phi_{errA}$  before switchover and  $\phi_{errB}$  after switchover, see Figure 5.5. The input to the bPLL depends on the test:

- when one backup port exists, the input is the phase error measured at the backup port that becomes active during switchover, i.e.  $\phi_{errB}$ , thus the data ends at the switchover instant
- when two backup ports exist, the input is the phase error measured at the backup port that remains backup during switchover, i.e.  $\phi_{errC}$ .

The test cases which result in a substantial *phase jump* are grouped in the left plots, the test cases which resulted in a negligible *phase jump* are grouped in the right plots of Figure H.1.

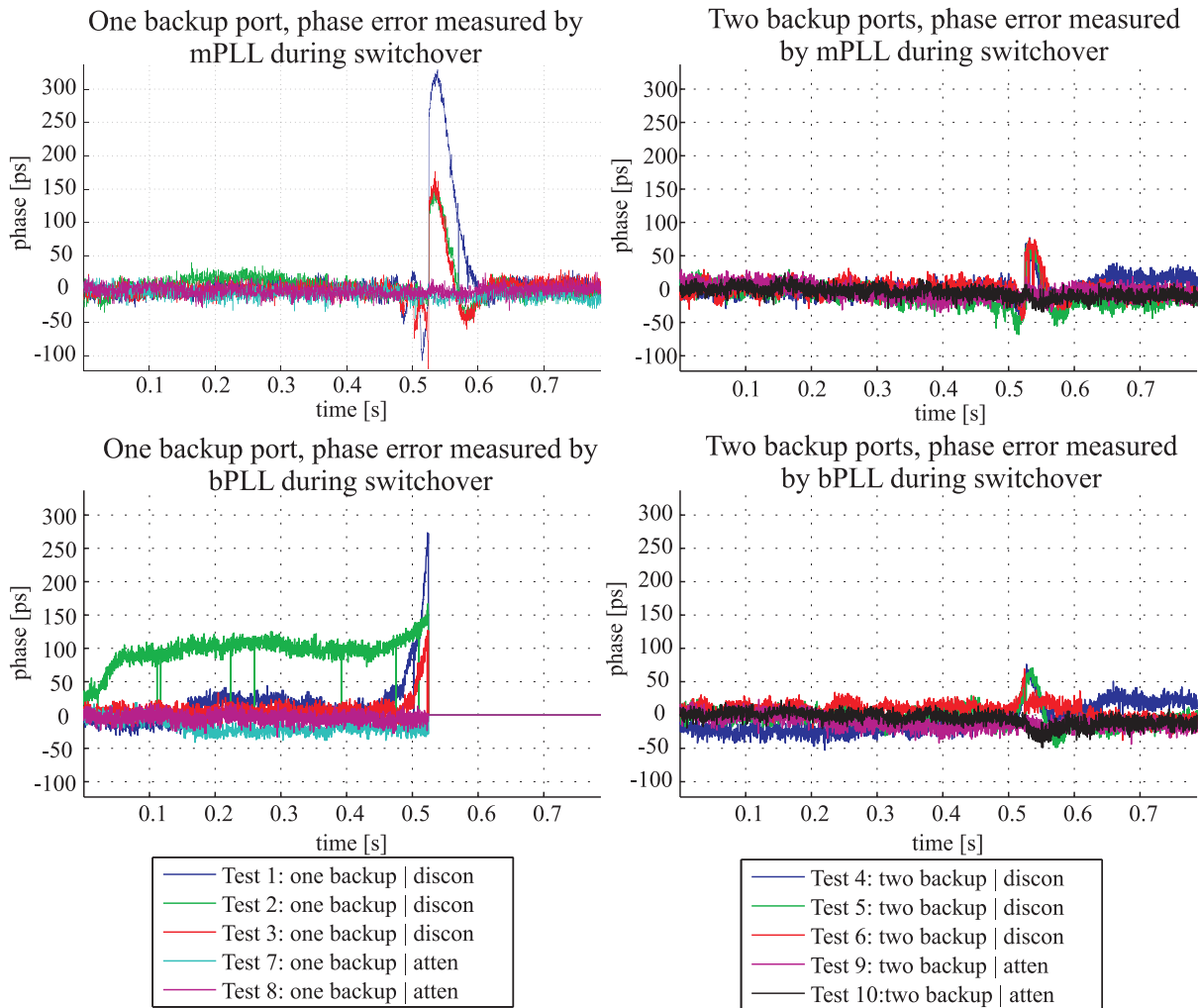


Figure H.1: The phase error at the Main PLL and the Backup PLL for scenario a.



## H.2 Direct Redundant Connection in Cascade of Switches (scenario b)

Table H.2 presents results of the measurements that are summarized in Figure 5.17 of subsection 5.6.2.

Test 0 in the table presents the reference measurement of the mean skew, its sdev and MTE, all measured over a period of 60 seconds. This reference measurement is used to estimate the *phase jump* in test measurements 1 to 10.

Tests 1 to 10 present results of synchronisation performance (mean, sdev, MTE) during the switchover measured with the oscilloscope as well as by the mPLL ( $\phi_{errA}$ ) and the bPLL ( $\phi_{errB}$ ). The last four columns provide an estimation of the *phase jump* during switchover.

Test $N^o$	Short Info	Failure	Phase skew measured with oscilloscope									Estimated phase jump			
			First Slave			Second Slave			Third Slave			SoftPLL	Scope		
			mean	sdev	MTE	mean	sdev	MTE	mean	sdev	MTE	Slave 1	Slave 1	Slave 2	Slave 3
			[ps]			[ps]			[ps]			[ps]			
0	reference p2		-38.3	12.5	94	-17.4	18.0	133	-5.1	25.1	194				
1	p3 → p2	discon	-27.4	20.5	354	-6.1	24.9	428	6.3	29.4	463	276 : 328	260	295	269
2	p2 → p1	discon	-13.5	15.6	561	-17.5	18.7	618	0.8	26.9	969	228 : 470	467	485	775
3	p1 → p2	discon	-14.5	34.2	179	-17.3	37.1	202	0.5	40.6	243	102 : 121	85	69	49
4	p1 → {p2,p3}	discon	-30.0	12.4	89	-12.6	15.4	111	-1.6	20.1	181	52 : 79	neg	neg	neg
5	p2 → {p3,p1}	discon	-20.6	17.6	139	0.4	23.2	167	12.6	28.2	226	49 : 66	45	34	32
6	p3 → {p1,p2}	discon	-27.9	11.8	97	-36.1	16.7	127	-20.0	23.9	186	57 : 62	3	neg	neg
7	p2 → {p3,p1}	attenu	-22.0	18.9	113	-9.2	22.7	137	1.6	28.8	180	23 : 41	19	4	neg
8	p2 → {p3,p1}	attenu	-26.7	18.3	99	-33.3	22.0	129	-16.0	26.0	159	32 : 40	5	neg	neg

Table H.2: Measurement results during switchover between direct redundant connections in a cascade for scenario b (pX: port number X, discon: disconnected, attenu: attenuated, neg: negligible).

### H.3 Indirect Redundant Connection (scenario c)

Figure H.2 presents results of the measurements that are summarized in Figure 5.18 of subsection 5.6.3.

Figure H.2 shows the raw phase error data from the SoftPLL of the WR Switch 2 depicted in Figure 5.18. In particular, the phase error input to PI controller of the mPLL is depicted in the upper plots and the phase error measured at the bPLL is depicted in the lower plots. The input to the mPLL is the phase error  $\phi_{errA}$  before switchover and  $\phi_{errB}$  after switchover, see Figure 5.5. The input to the bPLL is the phase error measured at the backup port that becomes active during switchover, i.e.  $\phi_{errB}$ , thus the data ends at the switchover instant. The test cases which result in a substantial *phase jump* are grouped in the left plots, the tests which resulted in a negligible *phase jump* are in the right plots of Figure H.2.

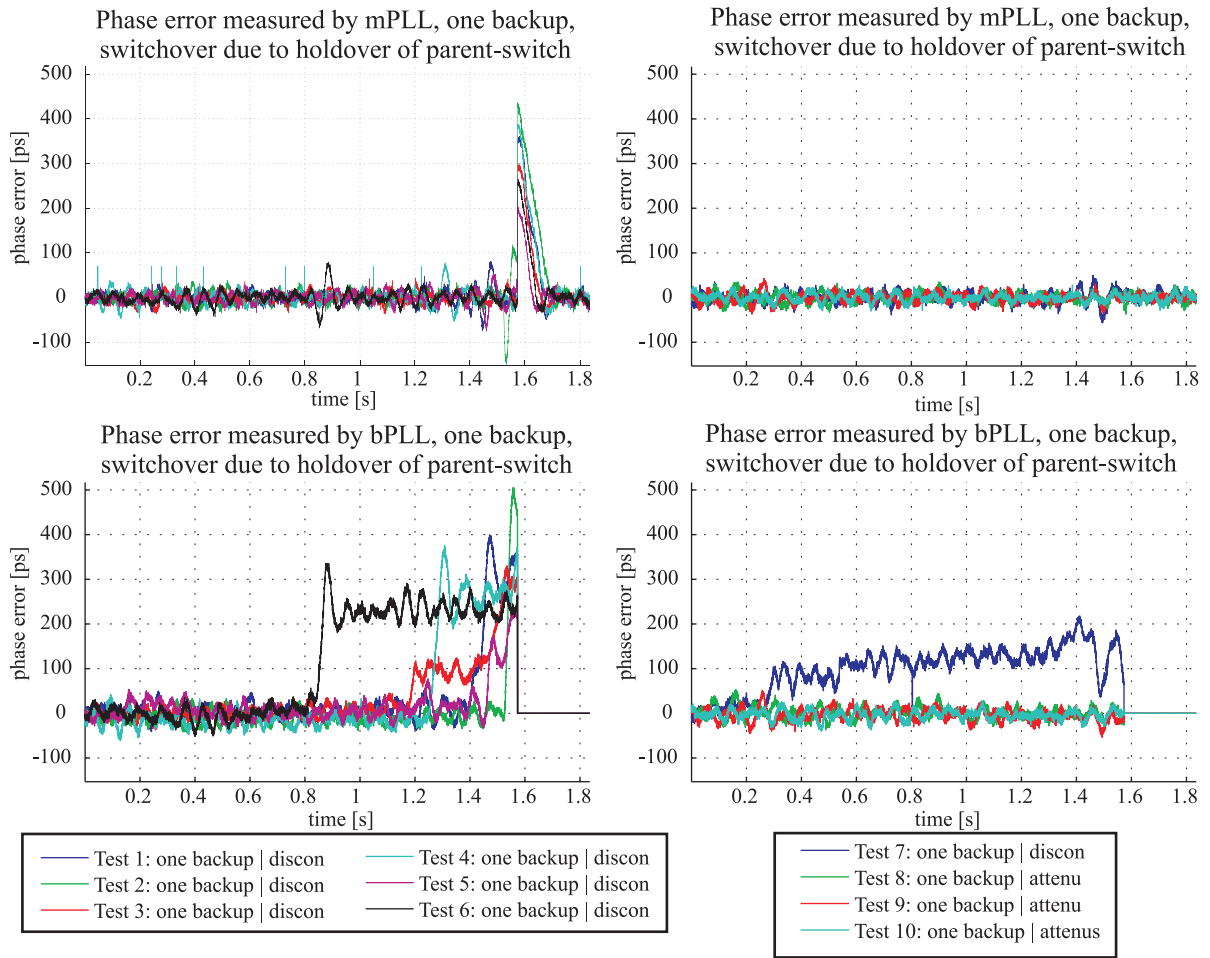


Figure H.2: The phase error at the Main PLL and the Backup PLL for scenario c.

## H.4 Indirect Redundant Connection in Cascade of Switches (scenario d)

Figure H.3 presents results of the measurements that are summarized in Figure 5.19 of subsection 5.6.4.

Figure H.3 shows the raw phase error data from the SoftPLL of the WR Switch 3 depicted in Figure 5.19. In particular, the phase error input to PI controller of the mPLL is depicted in the upper plots and the phase error measured at the bPLL is depicted in the lower plots. The input to the mPLL is the phase error  $\phi_{errA}$  before switchover and  $\phi_{errB}$  after switchover, see Figure 5.5. The input to the bPLL is the phase error measured at the backup port that becomes active during switchover, i.e.  $\phi_{errB}$ , thus the data ends at the switchover instant.

The dynamics of phase error captured by the bPLL before switchover are similar to scenario c for the cases where the fibre is disconnected. A deterioration of switchover performance for the cases where the fibre is attenuated can be observed.

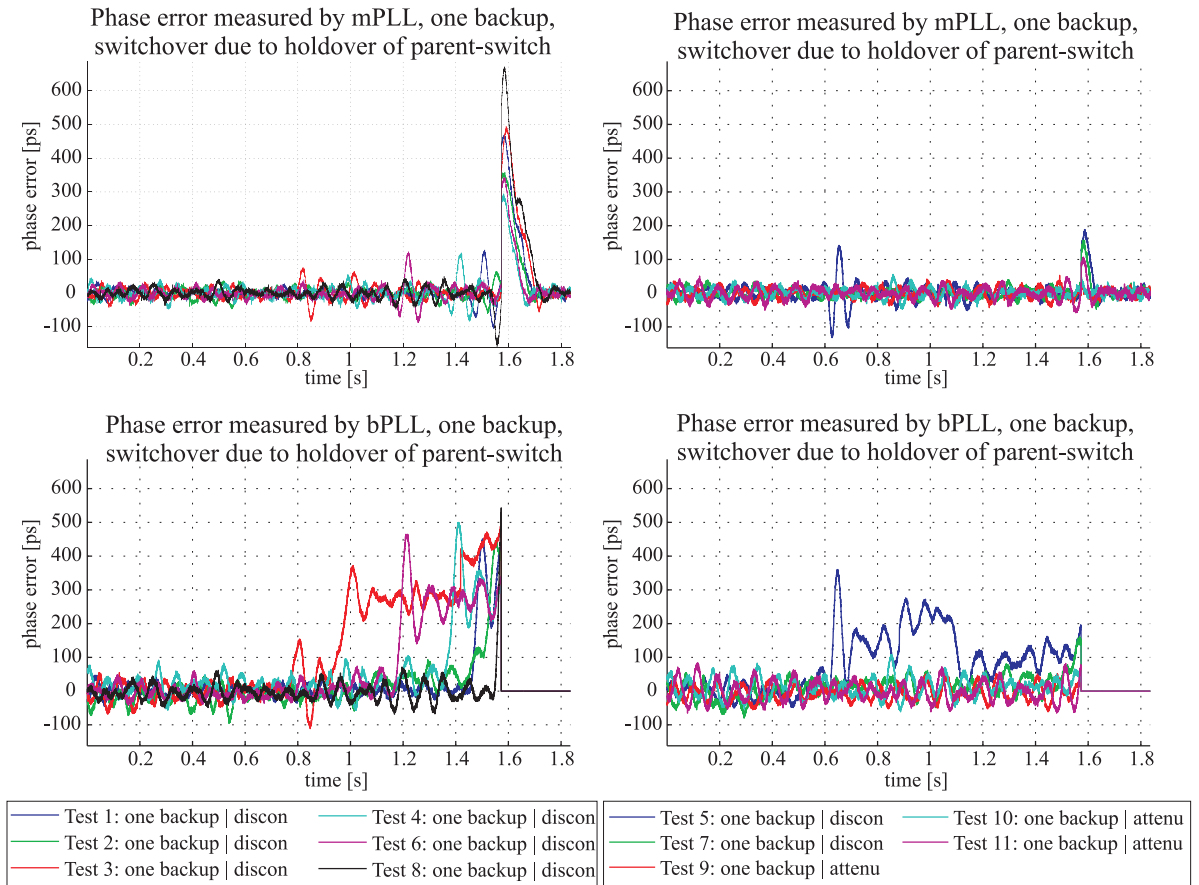


Figure H.3: The phase error at the Main PLL and the Backup PLL for scenario d.

## Appendix I

---

# Basic Configuration of the Reference WR-Based Control and Timing Network

---

This Appendix proposes a basic configuration in terms of data distribution for the reference WR-based control and timing network. This includes configuration of Virtual Local Area Networks (VLANs), port tagging QMODE<sup>1</sup> and untagging mask<sup>2</sup>, as well as usage of priorities and multicast addresses.

Firstly, in order to ease the management of the network, separate ranges of VLAN ID (VID) are allocated to VLANs, Ethernet Trees (E-TREES) and shortest path VIDs (SPVIDs) as follows:

- **0x001-0x1FF**: 509 VIDs to create VLANs, each might be translated into SPVIDs.
- **0x200-0x3FF**: 255 VIDs to create E-TREES, each E-TREE needs two VIDs.
- **0x400-0x3FF**: 3070 VIDs for SPVIDs which are dynamically allocated for VLANs.

All the switches in the network are configured to treat traffic with priority 6 and 7 as critical. Priority 6 is dedicated to Precision Time Protocol (PTP) traffic, priority 7 is dedicated to transmission of control messages between Data Masters and nodes. A multicast address is configured for each of the Data Masters. frames sent by nodes to Data Masters should be destined to the appropriate multicast address so that a network-wide reconfiguration is avoided when the Data Master is replaced. A basic set of VLANs and E-TREES for the network is specified in Table I.1 and presented in Figure I.1.

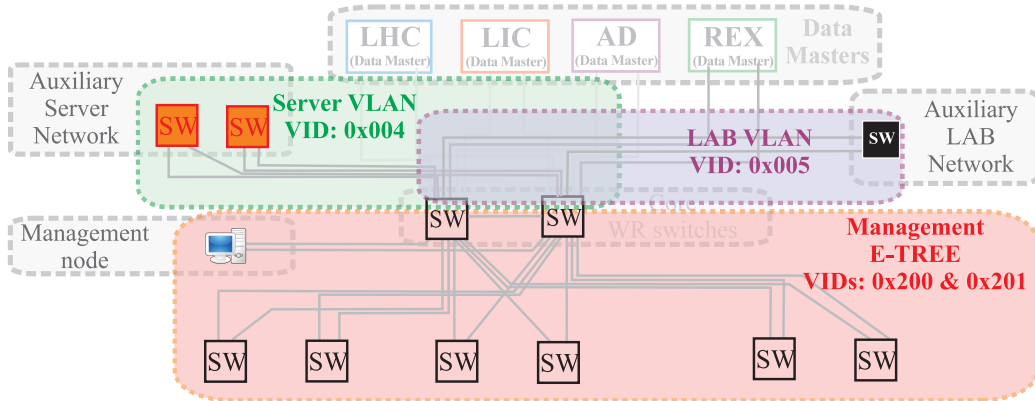
The ports of the switches are configured as specified in Table I.2. All the access switches, by default, have their ports (not connected to aggregate switches) configured to be in the "access"

---

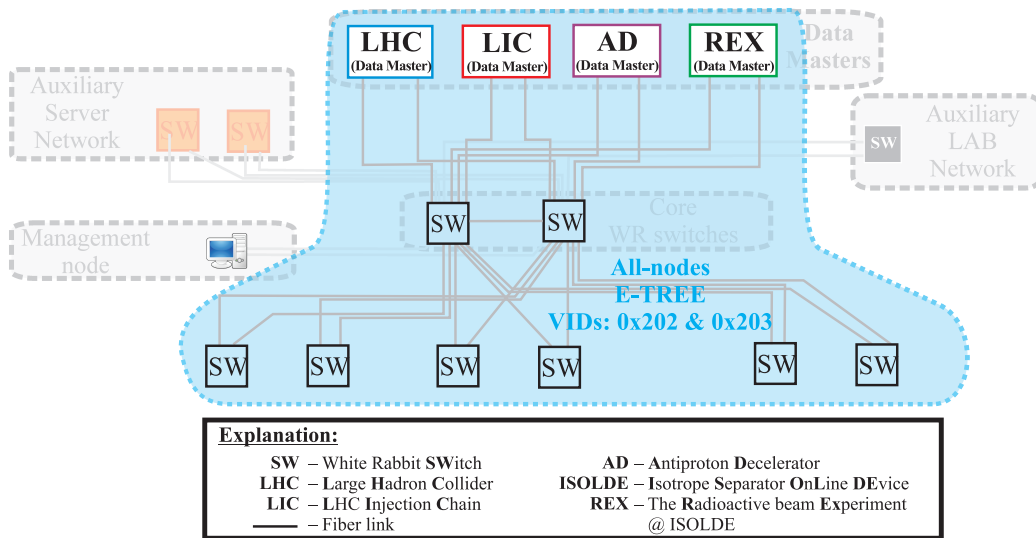
<sup>1</sup>There are 4 modes of ingress port tagging (see [112]): **Access** – received tagged frames are discarded, untagged frames are admitted and tagged with PVID and Priority; **Trunk** – received untagged frames are discarded, tagged frames are admitted; **Unqualified** – received frames are admitted, untagged frames are tagged with PVID and Priority; **VLAN disabled** – received frames are passed unmodified.

<sup>2</sup>Tags of any number of VIDs can be specified for removal at egress [112].

a) Best-effort traffic: Server and LAB VLANs, Management E-TREE



b) Critical traffic: all-nodes E-TREE



c) The highest priority best-effort traffic: DM-to-DM

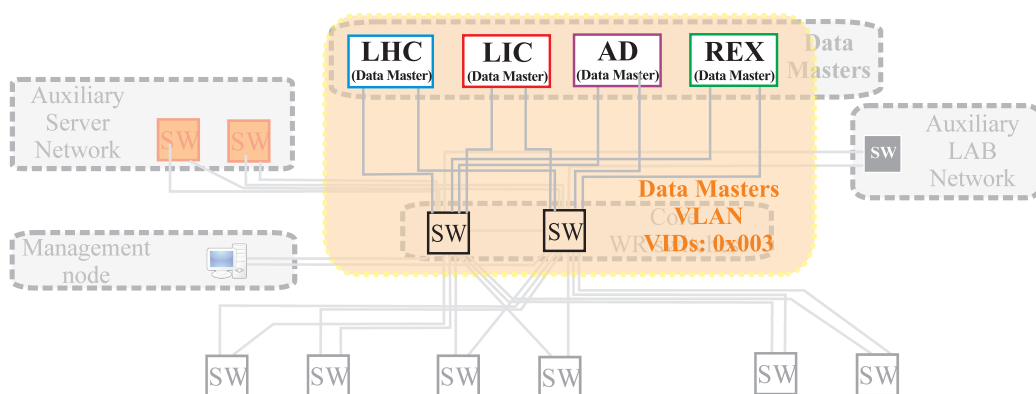


Figure I.1: Basic VLAN configuration in the proposed WR-based control and timing network design.

VIDs	Name	Recommended Priority	Traffic type
0x002	PTP VLAN	6	critical
0x003	DMs VLAN	5	best-effort
0x004	Servers VLAN	0	best-effort
0x005	LAB VLAN	0	best-effort
0x200	Management E-TREE trunk	0	best-effort
0x201	Management E-TREE branch	0	best-effort
0x202	All nodes E-TREE trunk	7	critical
0x203	All nodes E-TREE branch	7	critical

Table I.1: Basic VLAN configuration and a recommended priority for each VLAN.

QMODE. Consequently, these ports discard any received tagged frames and tag with the "Management E-TREE branch" (VID=2) any received untagged frames. The "Management E-TREE" is depicted in Figure I.1 a). Only when the management node verifies the newly attached node, the ports of the switch that the node connects to are configured as "unqualified". An unqualified port accepts tagged frames and tags untagged frames with the "All-nodes E-TREE branch" (VID=5). The "All-nodes E-TREE" is depicted in Figure I.1 b). This E-TREE is used by the Data Masters to send control messages (critical traffic) to the nodes and it can be used by any node to send critical traffic to the Data Masters. This traffic has the highest priority (i.e. 7), it is deterministic and reliable.

Switches	Switch ports connected to	Port QMODE	Ingress tagging with		Egress VIDs untagging
			PVID	Prio	
Core	Data Masters	Unqualified	0x202 (All-nodes trunk)	7	0x202, 0x002
	Aggregate switches	Trunk	-	-	0x005
	Server Network	Access	0x004 (Server VLAN)	0	0x005
	LAB network	Access	0x005 (LAB VLAN)	0	0x005
	Management node	Access	0x200 (Management trunk)	0	0x200, 0x002
Aggregate	All switches	Trunk	-	-	-
Access	Aggregate switches	Trunk	-	-	-
	Unknown nodes	Access	0x201 (Management branch)	0	0x201, 0x002
	Recognized nodes	Unqualified	0x203 (All-nodes branch)	7	0x203, 0x002

Table I.2: Port configuration of the switches.

The Data Masters communicate using "DMs VLAN", as depicted in Figure I.1 c). Communication using this VLAN requires Data Master to send frames tagged with proper VID, such frames are admitted by the ports of core switches which are set to the unqualified mode.

The auxiliary networks have their own VLANs to provide isolation of their traffic from the main network. The ports of core switches connecting to these networks are set to access mode. Thus, any tagged frame received at the port from LAB and Server networks is discarded; untagged frames are tagged with VID which is set to be discarded. The ports connected to the auxiliary networks are members of "All-nodes trunk" (VID=4), therefore all the critical traffic is unidirectionally forwarded to these networks.

PTP traffic is sent by the ports in "trunk" QMODE using "PTP VLAN" (VID=6) with priority 6. At unqualified and access ports, PTP traffic is sent with the VID used for tagging ingress traffic.

This basic configuration can be extended to provide E-TREES dedicated to each accelerator or to provide other logic division, as described in [98].

## Appendix J

---

# Reflections and Recommendations

---

The author considered the following factors while working on this thesis, proposed for further investigation:

- It might be worth installing a better oscillator in the White Rabbit switch to further increase reliability. A more stable oscillator can significantly improve failure detection and synchronisation performance during the switchover between alternative paths. As a result, sub-nanosecond synchronisation accuracy could be guaranteed in a large redundant network with a single backup path.
- The measured latency through a WR switch with the PTP traffic meets the initial requirements. However, the latency added by the PTP traffic is greater than expected from the calculations and the simulation. This indicates that further optimisation is possible if required by an application.
- The set of standards prepared by Time-Sensitive Networking (TSN), once completed, could potentially supersede some of the solutions proposed. The White Rabbit network could and should be seamlessly upgraded in the future with the newly developed standard enhancements.
- Currently, the most widely used mechanisms to configure redundant paths in Ethernet-based Local Area Networks (LANs) are protocols based on distributed algorithms. These protocols are not appropriate for the configuration of critical networks, such as CERN's control and timing network. The author recommends Software-defined networking (SDN) as the most appropriate approach for such networks. SDN's OpenFlow standard has recently been upgraded to support time-triggered configuration and seems the best candidate for network configuration protocol in CERN's future White Rabbit control and timing network.





---

# Bibliography

---

- [1] European Organization for Nuclear Research (CERN). <https://www.cern.ch/>.
- [2] J. Serrano. CERN General Machine Timing System: status and evolution. <https://indico.cern.ch/event/28233/contribution/1/material/slides/1.pdf>, 2008. CERN Presentation.
- [3] J.Serrano, P.Alvarez, and J.Lewis D.Dominguez. Nanosecond Level UTC Timing Generation and Stamping in CERN’s LHC. In *Proceedings of International Conference on Accelerator and Large Experimental Physics Control Systems (ICALEPCS)*, Gyeongju, Korea, 2003.
- [4] J. Lewis, P. Alvarez, J-C. Bau, S. Deghaye, I. Kozsar, and J. Serrano. The CERN LHC central timing, a vertical slice. In *Proceedings of International Conference on Accelerator and Large Experimental Physics Control Systems (ICALEPCS)*, Knoxville, USA, 2007.
- [5] White Rabbit Project. <http://www.ohwr.org/projects/white-rabbit>.
- [6] J. Serrano, P. Alvarez, M. Cattin, E. G. Cota, P. Moreira J. H. Lewis, T. Włostowski, et al. The White Rabbit Project. In *Proceedings of International Conference on Accelerator and Large Experimental Physics Control Systems (ICALEPCS)*, Kobe, Japan, 2009.
- [7] T. Włostowski. Precise time and frequency transfer in a White Rabbit network. Master’s thesis, Warsaw University of Technology, Warsaw, Poland, May 2011.
- [8] Roman G. CERN Accelerator Control System. <https://indico.cern.ch/event/273998/>. CERN Presentation.
- [9] Julian L., J-C. Bau, J. Serrano, D. Dominguez, and P. Alvarez Sanchez. The Evolution of the CERN SPS Timing System for the LHC Era. In *Proceedings of International Conference on Accelerator and Large Experimental Physics Control Systems (ICALEPCS)*, Gyeongju, Korea, 2003.

- [10] IEEE Standard for Ethernet. *IEEE 802.3-2012*.
- [11] Electronic Industries Alliance. ANSI/TIA/EIA-422-B Standard: Electrical Characteristics of Balanced Voltage Digital Interface Circuits (RS-422).
- [12] Accuracy and precision. [https://en.wikipedia.org/wiki/Accuracy\\_and\\_precision](https://en.wikipedia.org/wiki/Accuracy_and_precision).
- [13] T. Wlostowski. LHC Instabilities Trigger Distribution. <https://indico.cern.ch/event/295937/contribution/1/material/slides/0.pdf>. CERN Presentation.
- [14] G. Gong, S. Chen, Q. Du, J. Li, and Y. Liu. Sub-nanosecond Timing System Designed And Developed For LHAASO Project. In *Proceedings of International Conference on Accelerator and Large Experimental Physics Control Systems (ICALEPCS)*, Grenoble, France, 2011.
- [15] M. Buzio, R. Chritin, D. Giloteaux, D. Oberson. PS Booster B-train upgrade. <https://indico.cern.ch/event/346235/contribution/6/material/slides/1.pdf>. CERN Presentation.
- [16] IEEE Standard for Local and Metropolitan Area Networks: Media Access Control (MAC) Bridges. *IEEE 802.1D-2004*.
- [17] IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries. *IEEE 1991.106963*.
- [18] K. Dooley. *Designing Large Scale Lans*. O'Reilly Media, 2001.
- [19] IEEE Standard for Precision Clock Synchronization Protocol for Networked Measurement and Control Systems. *IEEE 1588-2008*.
- [20] ITU-T Recommendation. ITU-T X.200 (07/94): Open Systems Interconnection - Basic Reference Model: The basic model.
- [21] P. Moreira, P. Alvarez, J. Serrano, I. Darwezeh, and T. Wlostowski. Digital Dual Mixer Time Difference for Sub-Nanosecond Time Synchronization in Ethernet. *IEEE International Frequency Control Symposium (FCS)*, 2010.
- [22] T. Fleck, C. Prados, S. Rauch, and M. Kreider. FAIR Timing System. Technical report, GSI, Darmstadt, Germany, 2009. v1.2.
- [23] HiSCORE Gamma-Ray and Cosmic-Ray experiment. [http://tunka-hrjrg.desy.de/e98279/index\\_eng.html](http://tunka-hrjrg.desy.de/e98279/index_eng.html).
- [24] A multi-km<sup>3</sup> sized Neutrino Telescope (KM3NeT). <http://km3net.org>.

- [25] M.A. Weiss, G. Petit, and Z. Jiang. A comparison of GPS common-view time transfer to all-in-view. In *Proceedings of the 2005 IEEE International Frequency Control Symposium and Exposition*, 2005.
- [26] Collaboration: MIKES/CSC/FUNET. White Rabbit time-transfer experiment between Espoo and Kajaani in Finland. [http://www.ohwr.org/attachments/2250/MIKES-CSC-WR\\_time\\_link.pdf](http://www.ohwr.org/attachments/2250/MIKES-CSC-WR_time_link.pdf).
- [27] PRESS RELEASE: VSL, Dutch Metrology Institute. Always on time with the White Rabbit protocol. <http://www.vsl.nl/en/about-vsl/news/always-time-white-rabbit-protocol>, 2014.
- [28] N. Kaur, F. Frank, P. Tuckey, and P-E. Pottie. White Rabbit to disseminate time on an active telecom network? [http://www.ohwr.org/attachments/4272/5\\_WR4TFdisseminationNational-PEPottie.pdf](http://www.ohwr.org/attachments/4272/5_WR4TFdisseminationNational-PEPottie.pdf), 2016.
- [29] NI sets out plans for real-time distributed cyber-physical systems. <http://www.techdesignforums.com/blog/2014/08/09/labview-white-rabbit-distributed-real-time/>.
- [30] G. Daniluk. White Rabbit PTP Core the sub-nanosecond time synchronization over Ethernet. Master of science thesis, Warsaw University of Technology, Warsaw, Poland, 2012.
- [31] IEEE Standard for Local and metropolitan area networks – Link Aggregation. *IEEE 802.1AX-2014*.
- [32] The Tolly Group. Nortel Test Summary Ethernet Routing Switches, 2005.
- [33] G. Prytz. Redundancy in Industrial Ethernet Networks. *2006 IEEE International Workshop on Factory Communication Systems*.
- [34] GarrettCom, Inc. Standards-based Approaches to Redundancy and Fault Tolerance Using Industrial Ethernet LANs. *A White Paper for Network Engineers in Factories, Transportation Systems, Utilities, and Other Industrial Networking Applications*, 2008.
- [35] H. Weibel. Tutorial on Parallel Redundancy Protocol (PRP).
- [36] D. Allan and N. Bragg. *802.1aq Shortest Path Bridging Design and Evolution: The Architect's Perspective*. John Wiley & Sons.
- [37] Transparent Interconnection of Lots of Links (TRILL). <http://datatracker.ietf.org/wg/trill/charter/>.
- [38] J. Farkas and Z. Arato. Performance Analysis of Shortest Path Bridging Control Protocols. *IEEE Global Telecommunications Conference*, 2009.

- [39] Industrial communication networks - High availability automation networks - Part 2: Media Redundancy Protocol (MRP). *IEC 62439-2*, 2010.
- [40] H. Kirrmann, K. Weber, O. Kleineberg, and H. Weibel. HSR: Zero Recovery Time and Low-cost Redundancy for Industrial Ethernet (High Availability Seamless Redundancy, IEC 62439-3). In *Proceedings of the 14th IEEE International Conference on Emerging Technologies & Factory Automation*.
- [41] IEC 62439-3: Industrial communication networks – High availability automation networks – Part 3: Parallel Redundancy Protocol (PRP) and High - availability Seamless Redundancy (HSR). *International Electrotechnical Commission*, 2012.
- [42] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. OpenFlow: Enabling Innovation in Campus Networks. *ACM SIGCOMM Computer Communication Review* , 2015.
- [43] R.D. Vencioneck, G. Vassoler, M. Martinello, M.R.N. Ribeiro, and C. Marcondes. Flex-Forward: Enabling an SDN manageable forwarding engine in Open vSwitch. *10th International Conference on Network and Service Management (CNSM)*, 2014.
- [44] Interface to the Routing System (I2RS) working group. <https://datatracker.ietf.org/wg/i2rs/charter/>.
- [45] Forwarding and Control Element Separation (ForCES) working group. <https://datatracker.ietf.org/wg/forces/charter/>.
- [46] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, and S. Shenker. NOX: Towards an Operating System for Networks.
- [47] A. Tootoonchian, S. Gorbunov, Y. Ganjali, M. Casado, and R. Sherwood. On Controller Performance in Software-defined Networks. In *Proceedings of the 2Nd USENIX Conference on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services*, 2012.
- [48] Z. Cai, A. Cox, and T. Ng. Maestro: A system for scalable OpenFlow control. *Rice University, Houston, Texas, USA*, 2010.
- [49] S. Sharma, D. Staessens, D. Colle, M. Pickavet, and P. Demeester. OpenFlow: Meeting carrier-grade recovery requirements. *Computer Communications*, 2013.
- [50] S.H. Yeganeh, A. Tootoonchian, and Y. Ganjali. On scalability of software-defined networking. *Communications Magazine, IEEE*, February 2013.

- [51] T. Koponen, M. Casado, N. Gude, J. Stribling, L. Poutievski, M. Zhu, R. Ramanathan, Y. Iwata, H. Inoue, T. Hama, and S. Shenker. Onix: A Distributed Control Platform for Large-scale Production Networks. In *Proceedings of the 9th USENIX conference on Operating systems design and implementation*, 2010.
- [52] Transmission Control Protocol. <https://tools.ietf.org/html/rfc793>.
- [53] A. Li. RTP Payload Format for Generic Forward Error Correction. <https://tools.ietf.org/html/rfc5109>.
- [54] L. Vicisano, M. Luby, M. Handley, J. Gemmell, J. Crowcroft, and L. Rizzo. The Use of Forward Error Correction (FEC) in Reliable Multicast. <https://tools.ietf.org/html/rfc3453>, 2002.
- [55] 802.1CB - Frame Replication and Elimination for Reliability. <http://www.ieee802.org/1/pages/802.1cb.html>.
- [56] Time-Sensitive Networking Task Group. <http://www.ieee802.org/1/pages/tsn.html>.
- [57] IEEE Standard for Local and Metropolitan Area Networks—Bridges and Bridged Networks. *IEEE Std 802.1Q-2014*, Dec 2014.
- [58] IEEE Standard for Local and Metropolitan Area Networks - Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks. *IEEE 802.1AS-2011*.
- [59] E. Heidinger, F. Geyer, S. Schneelee, and M. Paulitsch. A performance study of Audio Video Bridging in aeronautic Ethernet networks. *7th IEEE International Symposium on Industrial Embedded Systems*, 2012.
- [60] J. Imtiaz, J. Jasperneite, and K. Weber. Approaches to reduce the latency for high priority traffic in IEEE 802.1 AVB networks. *9th IEEE International Workshop on Factory Communication Systems*, 2012.
- [61] Y. Kim, J. Takeuchi, and M. Nakamura. QoS requirements for Automotive Ethernet backbone systems. <http://www.ieee802.org/1/files/public/docs2011/new-avb-nakamura-automotive-backbone-requirements-0907-v02.pdf>.
- [62] D. Pannell. AVB - Generation 2 Latency Improvement Options. <http://www.ieee802.org/1/files/public/docs2011/new-pannell-latency-options-0311-v1.pdf>.
- [63] M.D.J. Teener. IEEE 802 Time - Sensitive Networking: Extending Beyond AVB. [http://standards.ieee.org/events/automotive/08\\_Teener\\_TSN.pdf](http://standards.ieee.org/events/automotive/08_Teener_TSN.pdf).

- [64] J. Perry, A. Ousterhout, H. Balakrishnan, D. Shah, and H. Fugal. Fastpass: A Centralized "Zero-queue" Datacenter Network. In *Proceedings of the 2014 ACM Conference on SIGCOMM*.
- [65] ITU-T Recommendation. G.8261-Timing and synchronization aspects in packet networks.
- [66] ITU-T Recommendation. G.8262: Timing characteristics of a synchronous Ethernet equipment slave clock.
- [67] ITU-T Recommendation. ITU-T G.8265.1: Precision time protocol telecom profile for frequency synchronization.
- [68] ITU-T Recommendation. G.8275.1: Precision time protocol telecom profile for phase/-time synchronization with full timing support from the network.
- [69] H. Kirrmann, C. Honegger, D. Ilie, and I. Sotiropoulos. Performance of a full-hardware PTP implementation for an IEC 62439-3 redundant IEC 61850 substation automation network. In *Proceedings of International IEEE Symposium on Precision Clock Synchronization for Measurement, Control and Communication (ISPCS)*, Sept 2012.
- [70] FortiGate-620B/-621B Datasheet. [http://www.fortinet.com/sites/default/files/productdatasheets/FG620B\\_621B\\_DS.pdf](http://www.fortinet.com/sites/default/files/productdatasheets/FG620B_621B_DS.pdf).
- [71] Juniper EX4500 Ethernet Switch Datasheet. <http://www.juniper.net/us/en/local/pdf/datasheets/1000322-en.pdf>, 2013.
- [72] Mean Time Between Failure Analysis of RuggedSwitch RS900 using Bellcore TR-332. <http://s-avt.ru/Portals/1/Catalog/RuggedCom/RS900/RS900%20MTBF%20Release.pdf>.
- [73] Cisco Catalyst 3560-E Series. [http://www.cisco.com/c/en/us/products/collateral/switches/catalyst-3560-e-series-switches/product\\_data\\_sheet0900aecd805bac22.html](http://www.cisco.com/c/en/us/products/collateral/switches/catalyst-3560-e-series-switches/product_data_sheet0900aecd805bac22.html).
- [74] M. To and P. Neusy. Unavailability analysis of long-haul networks. *IEEE Journal on Selected Areas in Communications*, 1994.
- [75] M. Lazzaroni. *Reliability Engineering*. Springer, Dordrecht, 2012.
- [76] IEC 60812 – Analysis techniques for system reliability – Procedure for Failure mode and effects analysis (FMEA), 2006.
- [77] B. Todd. *A Beam Interlock System for CERN High Energy Accelerators*. PhD thesis, Brunel University, West London, October 2006.

- [78] MIL-STD-882E: System Safety. <http://www.system-safety.org/Documents/MIL-STD-882E.pdf>.
- [79] IEC 61508 - functional safety of electrical/electronic/programmable-electronic safety related systems. Technical report, 1998.
- [80] Military Handbook: Electronic Reliability Design Handbook. (MIL-HDBK-338B), 1998.
- [81] L.B. James, A.W. Moore, A. Wonfor, R. Plumb, I.H. White, R.V. Penty, M. Glick, and D. McAluey. Packet error rate and bit error rate non-deterministic relationship in optical network applications. *Fiber Optics Communications*, 2005.
- [82] I. S. Reed and G. Solomon. Polynomial Codes Over Certain Finite Fields. *Journal of the Society for Industrial and Applied Mathematics*, 1960.
- [83] M. Lipinski, J. Serrano, T. Wlostowski, and C. Prados. Reliability In a White Rabbit System. In *Proceedings of International Conference on Accelerator and Large Experimental Physics Control Systems (ICALEPCS)*.
- [84] M. Rizzi, M. Lipiński, T. Włostowski, J. Serrano, G. Daniluk, P. Ferrari, and S. Rinaldi. White Rabbit clock characteristics. In *Proceedings of International IEEE Symposium on Precision Clock Synchronization for Measurement, Control and Communication (ISPCS)*, Stockholm, Sweden, 2016.
- [85] M. Buhler-Broglin, K. Elsener, L.A. Lopez Hernandez, G.R. Stevenson, and M. Wilhelmsson. General Description of the CERN Project for a Neutrino Beam to Gran Sasso (CNGS), 2000.
- [86] M. Bruckner, R. Wischniewski, et al. Results from the White Rabbit sub-nsec time synchronization setup at HiSCORE-Tunka. *33rd International Cosmic Ray Conference*, 2013.
- [87] M. Lipiński. Torutre Report. <http://www.ohwr.org/attachments/1911/tortureReport.v3.2.pdf>, July 2012. CERN Document.
- [88] LatticeMico32 Open, Free 32-Bit Soft Processor. <http://www.latticesemi.com/en/Products/DesignSoftwareAndIP/IntellectualProperty/IPCore/IPCores02/LatticeMico32.aspx>, 2016.
- [89] M. Strassler. OPERA: What Went Wrong. <http://profmattstrassler.com/articles-and-posts/particle-physics-basics/neutrinos/neutrinos-faster-than-light/opera-what-went-wrong/>.



- [90] H. Zhou, T. Kunz, and H. Schwartz. Adaptive correction method for an OCXO and investigation of analytical cumulative time error upper bound. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 2011.
- [91] J.K. Hwang, D.H. Shin, W. Han, and Y.K. Kim. A frequency model of OCXO for holdover mode of DP-PLL. In *Proceedings of the 39th SICE Annual Conference. International Session Papers*, 2000.
- [92] C.W.T. Nicholls and G.C. Carleton. Adaptive OCXO drift correction algorithm. In *Proceedings of the IEEE International Frequency Control Symposium and Exposition*, 2004.
- [93] Y.S. Shmaliy, L. Arceo-Miquel, O. Ibarra-Manzano, L. Moralez-Mendoza, O.Yu. Shmaliy, J.Z. de Paz, and J.M. Moreno-Reyes. A holdover algorithm for applications in GPS-based clock synchronization. *IEEE/ION Position, Location and Navigation Symposium*, 2008.
- [94] HP SmartClock Technology. Application Note 1279.
- [95] K. Gentile. The AD9548 as a GPS Disciplined Stratum 2 Clock. AN-1002 Application Note.
- [96] VM53S3-25.000-2.5/-30+75 - Mercury United Electronics. Datasheet.
- [97] A. Czubla, R. Osmyk, M. Lipinski P. Szterk, P. Krehlik, L. Sliwczynski, L. Buczek, W. Adamowicz, M. Marszalec, J. Nawrocki, and T. Widomski. Optical Fiber Time and Frequency Transfer in Poland – State of the Art. *4th International Conference on Quantum Metrology*, 2013.
- [98] J-C. Bau and M. Lipiński. Discussion On A White Rabbit based CERN Control and Timing Network. <http://www.ohwr.org/documents/85>, October 2011. CERN Document.
- [99] J. Koelemeij. WR TWTFT through long-haul duplexed fiber pairs. <http://www.ohwr.org/attachments/1102/>, March 2012.
- [100] EXFO. Variable Attenuator: FVA-60b. [http://www.exfo.com/Documents/TechDocuments/Specification\\_Sheets/EXFO\\_spec-sheet\\_FVA-60B\\_ang.pdf](http://www.exfo.com/Documents/TechDocuments/Specification_Sheets/EXFO_spec-sheet_FVA-60B_ang.pdf).
- [101] The Metro Ethernet Forum. Technical Specification MEF 6.1: Ethernet Services Definitions - Phase 2. [http://www.mef.net/Assets/Technical\\_Specifications/PDF/MEF\\_6.1.pdf](http://www.mef.net/Assets/Technical_Specifications/PDF/MEF_6.1.pdf).

- [102] IEEE 802.3br Interspersing express traffic (IET) Task Force (TF). [http://www.ieee802.org/3/br/Baseline/8023-IET-TF-1405\\_Winkel-iet-Baseline-r3.pdf](http://www.ieee802.org/3/br/Baseline/8023-IET-TF-1405_Winkel-iet-Baseline-r3.pdf), 2014.
- [103] IEEE P802.1Qbu - Frame Preemption. <http://www.ieee802.org/1/pages/802.1bu.html>.
- [104] M. Roda. Real-Time Distribution of Magnetic Field Measurements Over White-Rabbit. [http://www.ohwr.org/attachments/4270/2\\_PS-BTrain-9th-WR-workshop-Marco\\_Roda.pdf](http://www.ohwr.org/attachments/4270/2_PS-BTrain-9th-WR-workshop-Marco_Roda.pdf), 2016.
- [105] T. Wlostowski, J. Serrano, G. Daniluk, M. Lipinski, and F. Vaga. Trigger and RF distribution using White Rabbit. In *Proceedings of International Conference on Accelerator and Large Experimental Physics Control Systems (ICALEPCS)*, 2015.
- [106] M. Sundal. Development of a new Frequency Program in the CERN Proton Synchrotron. Master's thesis, University of Agder, Norway, January 2015.
- [107] Magnetic field and magnet current distribution through White Rabbit. [https://edms.cern.ch/ui/file/1364009/4/WR\\_POPS\\_Report\\_0v3.pdf](https://edms.cern.ch/ui/file/1364009/4/WR_POPS_Report_0v3.pdf), 2014. CERN Document.
- [108] E.G. Cota, M. Lipiński, T. Włostowski, E.V.D. Bij, and J. Serrano. White Rabbit Specification: Draft for Comments. <http://www.ohwr.org/documents/21>, July 2011. CERN Document, v2.0.
- [109] D.B. Sullivan, D.W. Allan, D.A. Howe, and F.L. Walls. Characterization of Clocks and Oscillators. NIST Technical Note 1337.
- [110] PPSi: PTP Ported to Silicon. <http://www.ohwr.org/projects/ppsi/wiki>.
- [111] T. Wlostowski, M. Lipinski, and G. Daniluk. White Rabbit Switch Gateway Architecture. <http://www.ohwr.org/attachments/2966/SwitchGWarchitecture.jpg>.
- [112] G. Daniluk. White Rabbit Switch HDL software interface. <http://www.ohwr.org/documents/281>, 2014. version 4.0.



---

# List of Figures

---

1.1	CERN accelerator complex . . . . .	1
1.2	LHC Injection Chain sequential cycles. . . . .	3
1.3	Concept of CERN accelerator cycles and events . . . . .	5
1.4	White Rabbit network. . . . .	9
1.5	Simplified architecture of a two-port WR switch. . . . .	11
3.1	Different types of redundant network topologies. . . . .	20
3.2	Path configuration in spanning tree protocols and shortest path protocols. . . . .	22
3.3	Erasure communication channel . . . . .	25
4.1	Boundary of a White Rabbit network considered in reliability calculations. . . . .	32
4.2	Possible non-redundant topologies of a WR network and Reliability Block Diagram. . . . .	39
4.3	Different network topologies that can accommodate 2000 nodes. . . . .	41
4.4	Principles of Forward Error Correction operation with an example configuration where $N = 2$ and $M = 2$ . . . . .	46
4.5	Measurement of synchronisation performance in a cascade of switches . . . . .	50
4.6	Contributors to event distribution latency . . . . .	51
4.7	Network latency for a control message . . . . .	52
4.8	Relation between the allowed network reconfiguration time, Forward Error Correc- tion configuration, and Inter-Frame Gap size. . . . .	53
5.1	Synchronisation and syntonisation in WR. . . . .	58
5.2	Overview of the WR phase-locked loop design. . . . .	60
5.3	Considered network arrangements to support redundancy for synchronisation. . . . .	61
5.4	Elements required to support seamless redundancy for sub-ns synchronisation. . . . .	62
5.5	Principles of functioning of support for synchronisation redundancy. . . . .	63
5.6	Changes of the WR phase-locked loop parameters for active and backup port before, during, and after switchover. . . . .	65
5.7	Model of WR phase-locked loop with support for synchronisation redundancy. . . . .	67

5.8	Different parameters of the Proportional-Integral controller applied to the model of WR phase-locked loop with gradual correction of phase error. . . . .	69
5.9	Simulation of setpoint re-adjustment and its contribution to time error. . . . .	70
5.10	Fully redundant (a), and partially redundant (b) White Rabbit networks. . . . .	73
5.11	Allan Deviation of phase error between WR switch and caesium frequency standard	75
5.12	Port PTP states in basic redundant topologies considered in this thesis. . . . .	77
5.13	A generic algorithm for adding a new PTP slave port. . . . .	78
5.14	Implementation of the seamless switchover. . . . .	80
5.15	Test scenarios for which the developed mechanisms were tested and their perfor- mance measured. . . . .	84
5.16	Phase jump measured during switchover between direct redundant connections. . .	86
5.17	Phase jump measured during switchover between direct redundant connections in cascade of switches. . . . .	87
5.18	Phase jump measured during switchover between indirect redundant connection and a backup direct connection. . . . .	88
5.19	Phase jump measured during switchover between an indirect redundant connection and a backup direct connection in a cascade of switches. . . . .	89
5.20	The worst-case phase jumps during switchover for each test in each scenario. . . .	90
6.1	Redundant mesh topology of the White Rabbit network. . . . .	91
6.2	Logic trees in the Rapid Spanning Tree Protocol and the Shortest Path Bridging. . .	94
6.3	Pseudo-multipath broadcast tree rooted at a switch or a node. . . . .	98
6.4	Frame loss during switchover due to variation of transmission latency. . . . .	101
6.5	Topology applicable for the developed methods. . . . .	104
6.6	Unidirectional pseudo-redundant spanning trees applicable for the developed meth- ods. . . . .	105
6.7	Topology with triple redundancy applicable for the developed methods. . . . .	106
6.8	Algorithm to establish pseudo-multipath unidirectional spanning trees. . . . .	108
6.9	Pseudo-multipath Ethernet Tree. . . . .	109
6.10	VID-based loss-less reconfiguration of network. . . . .	111
6.11	Deterministic forwarding of the selected and best effort traffic in the WR switch. . .	114
6.12	Updates and modifications to the WR switch developed for this thesis. . . . .	116
6.13	Architecture of the Topology Resolution Unit. . . . .	117
6.14	Architecture of the RTU Forwarding Engine enhanced for determinism. . . . .	122
6.15	Architecture of the Swcore Multi-Access Memory enhanced for determinism. . . .	123
6.16	Latency over one and two WR switches without intervening traffic. . . . .	128
6.17	Latency over one and two WR switches with intervening PTP traffic. . . . .	130
6.18	Latency of critical traffic for all ports with and without intervening PTP traffic. . .	131
6.19	Latency of critical traffic with intervening best-effort traffic. . . . .	132

6.20	Test of fast switchover between redundant links. . . . .	134
6.21	Estimation of switchover time based on the test results. . . . .	135
7.1	CERN accelerator complex . . . . .	138
7.2	Detailed layered design of WR-based control and timing network. . . . .	140
7.3	Distribution of time and frequency in the WR-based control and timing network. . .	141
7.4	VLANs in the WR-based control and timing network. . . . .	143
7.5	E-TREE rooted at the data masters and spanning the entire WR network. . . . .	145
7.6	Test setup of the proposed WR-based control and timing network design. . . . .	146
7.7	Latency through the WR-based control and timing network. . . . .	148
7.8	Reliability of the WR-based control and timing network. . . . .	150
A.1	White Rabbit network. . . . .	158
A.2	Standard PTP message exchange. . . . .	159
A.3	Digital Dual Mixer Time Difference phase detector. . . . .	160
A.4	Simplified architecture of a two-port WR switch . . . . .	161
D.1	WR network for which reliability calculation are performed and explained. . . . .	167
D.2	Example use cases of an element and a parent pair operation. . . . .	169
F.1	Contributors to the latency between generating control message by the data master and triggering event at a node. . . . .	179
G.1	Architecture of the PTP Support Unit . . . . .	183
H.1	The phase error at the Main PLL and the Backup PLL for scenario a. . . . .	187
H.2	The phase error at the Main PLL and the Backup PLL for scenario c. . . . .	189
H.3	The phase error at the Main PLL and the Backup PLL for scenario d. . . . .	190
I.1	Basic VLAN configuration in the proposed WR-based control and timing network design. . . . .	192

---

# List of Tables

---

1.1	CERN requirements for the new WR-based systems. . . . .	6
3.1	Comparison of existing networking solutions and the CERN requirements for WR. . . . .	30
4.1	Factors that cause failure and their impact on the delivery of control messages and timing. . . . .	33
4.2	Representative values of switch and fibre Mean Time Between Failures used in the analysis. . . . .	35
4.3	Reliability values of safety-critical systems (black) and network elements (grey). . . . .	37
4.4	Results of the reliability calculations for non-redundant topologies. . . . .	40
4.5	Results of the reliability calculations for all the considered network topologies (data in grey is provided from Table 4.4). . . . .	42
4.6	Loss of messages due to Bit Error Rate in a WR-based control and timing network. . . . .	44
4.7	Transmission reliability calculated for different Forward Error Correction parameters. . . . .	47
4.8	Network latency and allowed switchover times calculated for different Forward Error Correction configurations and number of switches between the data master and node. . . . .	54
6.1	Latency through different types of switches for Gigabit Ethernet. . . . .	95
6.2	Classification and configuration of ports on the switch. . . . .	108
6.3	Latency through different types of switches for Gigabit Ethernet. . . . .	113
6.4	Parameters in entries stored for each Filtering ID (FID) in the TRU_TAB. . . . .	119
6.5	Latency values through WR switches estimated based on design evaluation and simulation. . . . .	125
6.6	Long-term latency test with intervening PTP traffic. . . . .	129
6.7	Latency through different types of switches for Gigabit Ethernet. . . . .	133
6.8	Latency measurement performed within the context of the WR-Btrain project. . . . .	133
7.1	Expected synchronisation performance in the proposed WR-based control and timing network. . . . .	147

7.2	Latency values estimated and measured for the WR-based control and timing network. . . . .	147
7.3	Parameters and performance of different Forward Error Correction schemas and for a simple replication of frames. . . . .	149
7.4	Reliability of the reference WR-based control and timing network. . . . .	151
7.5	Comparison of initial CERN requirements and characteristics of the proposed reference WR-based control and timing network. . . . .	152
B.1	Requirements for the new CERN control and timing system. . . . .	163
C.1	Reliability and availability for all types of considered topologies and Mean Time Between Failures values. . . . .	165
D.1	Probability calculations for layer 1 of the redundant network. . . . .	170
D.2	Probability calculations for layer 2 of the redundant network. . . . .	171
D.3	Probability calculations for layer 3 of the redundant network. . . . .	172
D.4	Probability calculations for a single cell XY that has a parent pair at layer 3 and 8 children pairs at layer 4. . . . .	173
D.5	Probability calculations for layer 5 – the reliability of WR network. . . . .	174
D.6	Final calculations of the Mean Time Between Failures, reliability and availability for the redundant network. . . . .	175
E.1	Header proposed to be prepended to the FEC block in the payload of an Ethernet frame. . . . .	177
H.1	Measurement results during switchover between direct redundant connections in scenario a (pX: port number X, discon: disconnected, attenu: attenuated, neg: negligible). . . . .	186
H.2	Measurement results during switchover between direct redundant connections in a cascade for scenario b (pX: port number X, discon: disconnected, attenu: attenuated, neg: negligible). . . . .	188
I.1	Basic VLAN configuration and a recommended priority for each VLAN. . . . .	193
I.2	Port configuration of the switches. . . . .	193





---

# Listings

---

- 1 Pseudo-code showing how the ingress and egress mask in the Topology Resolution Unit are generated. . . . . 119



---

# List of Abbreviations

---

AD	Antiproton Decelerator	2, 3, 138, 139, 142
AVB	Audio-Video-Bridging	26
BC	Boundary Clock	29, 141, 142
BER	Bit Error Rate	33, 38, 44–47, 52
BMCA	Best Master Clock Algorithm	28, 77, 82
bPLL	Backup PLL	81, 186–190, 209
CBCM	Central Beam and Cycle Manager	4
CCR	CERN Control Room	139
CNGS	CERN Neutrinos to Gran Sasso	2, 3
CoS	Class of Service	26, 48
CRC	Cyclic Redundancy Check	95, 96, 112, 180
C-T	cut-through	53, 54
DDMTD	Digital Dual Mixer Time-Difference	12, 60, 74, 159, 160, 162
EEC	Ethernet equipment slave clock	30
E-TREE	Ethernet Tree	108, 109, 142, 144, 145, 191, 193, 194, 208, 209
FAA	Federal Aviation Administration	36
FEC	Forward Error Correction	25, 31, 45–47, 51, 53–55, 97, 99, 101–103, 133, 134, 146, 147, 149, 150, 154, 177, 179, 180, 207, 210, 211
FID	Filtering ID	117–119, 210
ForCES	Forwarding and Control Element Separation	24

FPGA	Field Programmable Gate Array	8, 11, 12, 60, 76, 79, 154, 161, 162, 181
G/W	gateway	11, 161
GM	Grandmaster	141, 142, 146
GMT	General Machine Timing	2–6, 164
GPS	Global Positioning System	4, 8, 13, 36
GSI	GSI Helmholtz Centre for Heavy Ion Research	127
HDL	Hardware Description Language	8, 15, 154
HDL IP	Hardware Description Language Intellectual Property	15
HiSCORE	Hundred Square km Cosmic ORigin Explorer	83
hPLL	Helper PLL	60, 81
HSR	High-availability Seamless Redundancy	24, 25, 29, 83
I2RS	Interface to the Routing System	24
ID	identification	4, 82
IEC	International Electrotechnical Commission	24, 29, 36, 37
IEEE	Institute of Electrical and Electronics Engineers	23, 25, 26, 126
IETF	Internet Engineering Task Force	23, 24
IFG	Inter-Frame Gap	52, 53, 55, 96, 103, 150, 180, 207
IP	Intellectual Property	15
IPv4	Internet Protocol version 4	181
IS-IS	Intermediate System to Intermediate System	23, 107
ISOLDE	Isotope Separator On Line DEtector	2, 3, 138
ISS	International Space Station	36
IST	Inter-Switch Trunk	20
ITU-T	International Telecommunication Union – Telecommunication Standardization Sector	28
KM3Net	Cubic Kilometre Neutrino Telescope	83
L1	Layer 1	12, 28, 58, 159, 160
L2	Layer 2	11, 158, 161, 162, 180
LACP	Link Aggregation Control Protocol	22

LAN	Local Area Network	8, 10, 11, 15, 17, 20, 21, 24, 26, 92, 93, 95, 105, 126, 142, 158, 195
LEIR	Low Energy Ion Ring	3
LHAASO	Large High Altitude Air Shower Observatory	83, 155, 163
LHC	Large Hadron Collider	2–6, 127, 138, 139, 142, 155, 163, 164
LIC	LHC Injection Chain	3, 138, 139, 142
LRE	Link Redundancy Entity	24
LSB	least significant bit	68
MAC	Media Access Control	23, 94, 100, 113, 121, 122, 158, 161, 162, 180
MIL-STD	United States Military Standard	36, 37
MIT	Massachusetts Institute of Technology	27
MLAG	Multi-Chassis Link Aggregation	22
mPLL	Main PLL	60, 81, 186–190, 209
MRP	Media Redundancy Protocol	23, 24
MSTP	Multiple Spanning Tree Protocol	23, 93
MTBF	Mean Time Between Failures	34–37, 40, 42, 43, 150, 151, 165, 175, 210, 211
MTE	maximum time error	61, 85, 186, 188
MTIE	Maximum Time Interval Error	146
MTTF	Mean Time To Failure	34
MTTR	Mean Time To Repair	34, 36, 37, 40–42, 150, 151, 165
NASA	National Aeronautics and Space Administration	36, 37
NIC	Network Interface Controller	162, 181, 182
OSI	Open Systems Interconnection	11
PHY	physical layer	44
PI	Proportional-Integral	60, 68, 69, 187, 189, 190, 208
PLL	phase-locked loop	49, 59, 60
PPS	Pulse Per Second	49
PPSi	WR PTP daemon	80, 162
PRP	Parallel Redundancy Protocol	24, 25, 29

PS	Proton Synchrotron	3–5, 127
PSU	PTP Support Unit	80–82, 181, 182
PTP	Precision Time Protocol	11, 12, 27–29, 44, 45, 48, 49, 55, 58, 59, 61–63, 67, 68, 71, 73, 76–79, 81, 82, 102, 113, 114, 122, 125, 129–133, 136, 153, 154, 158–160, 162, 181, 182, 191, 194, 195, 208, 210
QoS	Quality of Service	26, 48
RBD	Reliability Block Diagram	39–41, 207
REX	Radiation Beam Experiment	138, 139, 142, 144
RF	Radio-frequency	127
RSTP	Rapid Spanning Tree Protocol	23, 28, 93, 94, 100, 110, 208
RTP	Real-Time Transport Protocol	25
RTU	RTU Forwarding Engine	12, 116, 117, 121–124, 161, 162, 208
S/W	software	11, 161
SAE	Society of Automotive Engineers	26
S-and-F	store-and-forward	53, 54
sdev	standard deviation	186, 188
SDN	Software-defined networking	24, 195
SMLT	Split Multi-Link Trunking	22
SNMP	Simple Network Management Protocol	162
SoftPLL	WR PLL implementation	80, 85
SPB	Shortest Path Bridging	23, 28, 93, 94, 107, 108, 110, 154, 208
SPB-MAC	Shortest Path Bridging MAC Mode	23
SPB-VID	Shortest Path Bridging VID Mode	23, 108, 142
SPS	Super Proton Synchrotron	3–5
SPVID	shortest path VID	144, 191
SRP	Stream Reservation Protocol	26
STP	Spanning Tree Protocol	23, 28, 93
SVL	Shared VLAN Learning	117
Swcore	Swcore Multi-Access Memory	12, 116, 121, 123, 124, 161, 181, 208

SyncE	Synchronous Ethernet	28
TC	Transparent Clock	29
TCP	Transmission Control Protocol	25
TE	time error	6, 49, 63, 68–71, 74, 84, 208
TLV	Type-Length-Value	114
TRILL	Transparent Interconnection of Lots of Links	23
TRU	Topology Resolution Unit	116–120, 122, 127, 134, 208, 213
TSN	Time-Sensitive Networking	25, 26, 30, 155, 195
UDP	User Datagram Protocol	181
UTC	Coordinated Universal Time	2, 4, 13
VCXO	Voltage-Controlled Crystal Oscillator	60, 74, 90
VHDL	Very High Speed Integrated Circuits Hardware Description Language	8, 80, 116, 117, 154, 181
VID	VLAN ID	23, 94, 98, 100, 105, 107–110, 117, 122, 142, 161, 191, 193
VLAN	Virtual Local Area Network	26, 93, 95, 100, 105, 112, 113, 121, 122, 126, 128, 137, 139, 142, 162, 180, 191, 193
WR PLL	WR phase-locked loop	12, 59, 60, 62, 65–69, 71, 72, 79, 81, 82, 85, 87, 162, 207, 208
WR PTP	WR extension to PTP	12, 60, 64, 65, 68, 72, 159
WRTD	WR Trigger Distribution	127