



D1.9

Final Drive-by-Data Concept Design

Project number:	730830
Project acronym:	Safe4RAIL
Project title:	Safe4RAIL: SAFE architecture for Robust distributed Application Integration in roLLing stock
Start date of the project:	1 st of October, 2016
Duration:	24 months
Programme:	H2020-S2RJU-OC-2016-01-2
Deliverable type:	Report
Deliverable reference number:	ICT-730830 / D1.9 / 1.0
Work package	WP 1
Due date:	30 Sept 2018 – M24
Actual submission date:	16 October 2018
Responsible organisation:	TTT
Editor:	Nataša Simanić-John
Dissemination level:	PUBLIC
Revision:	Final 1.0
Abstract:	Describes the concept of design and methodology for next generation TCMS including all relevant technologies and open issue/gap analysis.
Keywords:	IMP Concept, Train Topology, Architecture, Inauguration, Clock Synchronization, Flow Control, Data Transmission, Configuration, Verification/Validation, TSN



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 730830.

Editor

Nataša Simanić-John (TTT)

Contributors (ordered according to beneficiary numbers)

Nataša Simanić-John, Arjan Geven, Mirko Jakovljevic (TTT)

Bernd Löhr, Iris Bosse (NEW)

Tobias Pieper (UNI)

Adrian Szawlowski (IAV)

Disclaimer

The information in this document is provided “as is”, and no guarantee or warranty is given that the information is fit for any particular purpose. The content of this document reflects only the author’s view – the Joint Undertaking is not responsible for any use that may be made of the information it contains. The users use the information at their sole risk and liability.

Executive Summary

The Final Drive-by-Data Concept Design gives an overview of previous work done within Safe4RAIL's WP1 and presents a summary of outlined mechanisms needed for the implementation of the Integrated Modular Platform.

An accent is put on following topics:

- Scheduled traffic, which introduced TSN into the IMP implementation; TSN standards were used in employment of clock synchronization, traffic scheduling, filtering and policing, and redundancy management;
- Topology and architecture, which needed to be adopted to the new circumstances of the scheduled data traffic, in comparison to standardized common practice in use, according to [1] and [2]. This enabled a significant improvement in terms of reliability and safety.
- Inauguration, where the new concept for train composition discovery increased the safety integrity level to SIL4;
- Data Transmission, with an end-to-end concept including the new TSN-PD traffic class;
- Configuration, with regards to TSN needs, together with tooling, development, commissioning and maintenance topics.
- Verification and validation, executed by using a fault injection method.

Contents

Contents	4
Chapter 1 Introduction	9
1.1 Scope.....	9
1.2 Integrated Modular Platform	9
1.2.1 High-level requirements	10
1.2.2 IMP Concept	11
1.2.2.1 <i>Network services offered by Drive-by-Data</i>	12
1.2.2.2 <i>Functional Distribution Framework Middleware</i>	14
Chapter 2 Train Topology and Architecture	17
2.1.1 ETB Topology	17
2.1.1.1 <i>Solution</i>	17
2.1.2 ECN Topology.....	18
2.1.2.1 <i>Topology virtualization</i>	19
2.1.2.2 <i>Preferred Solution</i>	20
Chapter 3 Inauguration Concept	22
3.1.1 ETB Inauguration	22
3.1.2 Operational Train Inauguration.....	23
3.1.3 Safe Inauguration.....	25
Chapter 4 Networking Concept	27
4.1 Clock Synchronisation Concept	27
4.1.1 Introducing the Clocks.....	27
4.1.2 Robust Clock Synchronisation	28
4.1.3 Fault-Tolerant Clock Sync Concept for ETB and ECN.....	30
4.1.3.1 <i>ETB Synchronization</i>	30
4.1.3.2 <i>Distributed Set of High-Integrity Clocks</i>	31
4.1.4 GPS and NTP time as basis.....	34
4.2 Flow Control Concept	34
4.2.1 Enhancements for Scheduled Traffic IEEE 802.1Qbv [23]	34
4.2.2 Per-Stream Filtering and Policing IEEE 802.1Qci [24].....	36
4.2.3 Achieving Traffic Requirements with Traffic Shaping Mechanisms.....	38
4.3 TSN Mapping (ETB – ECN)	39
4.3.1 TSN Gateway.....	40
4.3.2 Configuration needs.....	43

4.4	Redundancy Management.....	43
Chapter 5	Data Transmission Concept.....	46
5.1	End-to-End Concept w. TRDP / TSN-PD.....	46
5.1.1	Network Protocol Layers	46
5.1.2	Communication Profiles	48
5.1.3	Framing.....	49
5.1.4	Addressing.....	52
5.2	Integration to FDF	52
5.3	Safety Concept	53
Chapter 6	Configuration.....	55
6.1	Configuration & Life Cycle	55
6.1.1	Life cycle.....	55
6.1.2	System Communication Integration.....	57
6.2	TSN network configuration.....	58
6.2.1	Loading configurations to network devices.....	59
6.3	Tooling & Development.....	60
6.3.1	EN 50657 [27] and Tooling.....	60
6.3.1.1	<i>System Configuration Tools</i>	61
6.3.2	Configuration flow – FDF Application/Functions driven	61
6.3.2.1	<i>TRDP</i>	63
6.3.3	Configuration flow – Topology/network driven.....	63
Chapter 7	Verification & Validation Concept.....	65
7.1	Dependability Evaluation Methods.....	66
7.2	Network Verification by Fault Injection	66
7.3	Failure criteria for communication.....	67
Chapter 8	Summary and Conclusions.....	68
Chapter 9	List of Abbreviations	69
Chapter 10	Bibliography.....	74

List of Figures

Figure 1: The 8 pillars on which the IMP is conceptualized.....	10
Figure 2: Integrated Modular Platform high-level overview	11
Figure 3: Generic embedded platform network, embedded computers and software platform components for next generation TCMS.....	13
Figure 4 FDF in the context of a train (simplified)	15
Figure 5: ETB in serial redundant layout.....	18
Figure 6: ETB in parallel redundant layout (Variant D).....	18
Figure 7: Topology virtualization approach: Policing to create dual independent lanes out of a ring or ladder network.....	20
Figure 8: ECN layout: physical ring, logical ladder.....	20
Figure 9: ETB/ECN Layout proposed by CTA.....	21
Figure 10: Inauguration with separate ETB lines	23
Figure 11: Operational train directory computation block diagram [15]	24
Figure 12: TTDB Data Model [15].....	24
Figure 13: Cooperation of ETBN and CCU (source: [18])	26
Figure 14: ETB lines usage for validation of inauguration result (source: [18]).....	26
Figure 15: PTP: Two trains with two consists	27
Figure 16: PTP: Trains after coupling and inauguration.....	27
Figure 17: Safe4RAIL system based on IEEE p802.1AS-rev [21] standard's ready mechanisms	29
Figure 18: Failure Sources within Safe4RAIL system	30
Figure 19: GlobalMC's synchronization startup according to assigned priorities.....	31
Figure 20 Option one: ETB and ECNs exclusively asynchronous.....	33
Figure 21 Option two: ECNs synchronous to ETB	34
Figure 22 Scheduling, as defined by IEEE 802.1Qbv, within frame forwarding process.....	36
Figure 23 Per-Stream Filtering and Policing, within frame forwarding process functions of a switch	38
Figure 24: TSN Streams over ETB	40
Figure 25: TSN traffic on ETB.....	41
Figure 26: Dataflow Mapping for ETBN in Leading Consist	42
Figure 27: TSN Mapping for ETBN in Non-leading Consist	43
Figure 28 Stream functions in a switch, with marked peering levels of FRER sublayers.....	45
Figure 29: Protocol Layers	47
Figure 30: ECN Dual Plane Network with sample data flow.....	48
Figure 31: Process Data Communication Pattern – Push	49
Figure 32: Legacy TRDP framing	50

Figure 33: TRDP TSN-PDU framing51

Figure 34: TSN Stream Identification52

Figure 35: Dataflow of FDF/DbD53

Figure 36: Safe Communication54

Figure 37: SDT computation Variant 2.....54

Figure 38: The life cycle of a train.....55

Figure 39: Device view56

Figure 40: Functional view.....56

Figure 41: Bottom-up: Application constraints lead to network schedule (left) and Top-Down:
Network schedule is calculated in the first step (right).....58

Figure 42: TSN network configuration workflow.....59

Figure 43: Network Configuration with NETCONF60

Figure 44: Application driven configuration – Steps62

Figure 45: Application driven configuration – several End Devices63

Figure 46: Example Network Layout.....64

Figure 47 V-Model, a development method for a system lifecycle65

List of Tables

Table 1: Traffic classes and traffic shaping mapping39

Table 2: TRDP PD Frame50

Table 3: TRDP TSN PD-PDU51

Table 4: List of configurations within a consist.....57

Table 5: List of Abbreviations73

Chapter 1 Introduction

The purpose of this document is to present the final, agreed-upon concepts of architectural and design methods for the use of Time-Sensitive Networking (TSN) with the next generation TCMS. The goal of the present document is to ensure a consistent final set of concepts that can be used as guidance for the next steps of the NG-TCMS development related to the IMP and NG-TCN.

1.1 Scope

This document is the main Drive-by-Data paper that is released as “PUBLIC” deliverable and builds extensively on the work performed in the previous two years. It is the result of iterations according to the project plan which started with the requirements and went iteratively through several aspects from initial concepts to refined concepts to final concepts. It incorporates ideas and designs that have been presented partly in previous deliverables Safe4RAIL D1.4 [3], Safe4RAIL D1.6 [4], Safe4RAIL D1.7 [5], and Safe4RAIL D2.3 [6].

The key topics relevant for the final Safe4RAIL system concept, are summarized in the following seven chapters:

Chapter 1 offers a background and presents the Integrated Modular Platform (IMP), Drive-by-Data and Functional Distribution Framework (FDF) concepts. The content is mainly adopted from the refined Drive-by-Data concept design, elaborated in detail in Safe4RAIL D1.4 [3] and Safe4RAIL D2.3 [6], which have been extended and completed to include new insights.

Chapter 2 compiles topics related to train topology and architecture. The content of this chapter is adopted from the network design methodology elaboration of Safe4RAIL D1.4 [3] extended with new concepts and insights.

Chapter 3 offers the inauguration concept based on the Safe4RAIL D1.6 [4].

Chapter 4 explains the networking concept, related to clock synchronization, flow control and redundancy management. These concepts are adopted from the initial descriptions in the documents Safe4RAIL D1.6 [4] and Safe4RAIL D1.7 [5].

Chapter 5 presents the data transmission concept, adopted from Safe4RAIL D1.6 and Safe4RAIL D1.7 [4].

Chapter 6 discusses the configuration of the system with regards to all relevant aspects, as originally described in Safe4RAIL D1.6 [4].

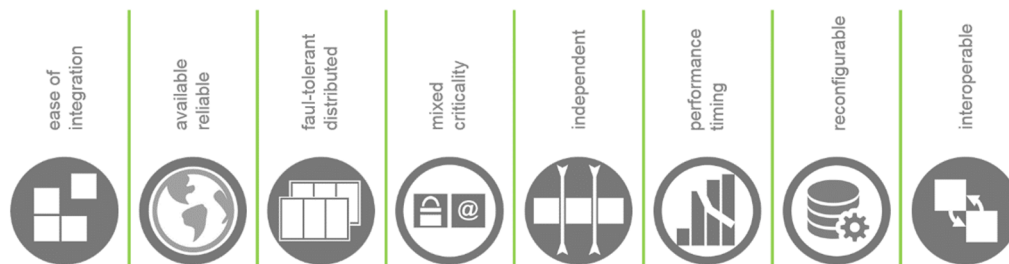
Chapter 7 discusses the verification and validation, based on the results presented in Safe4RAIL D1.5 [7].

1.2 Integrated Modular Platform

System integration represents a core capability required for the design of advanced integrated architectures using a virtualized computing and networking infrastructure. With advanced integrated systems tailored to host many critical and non-critical functions, the system integration gains in importance as it represents a common shared resource relevant for all functions. Its features influence the system architecture, topology and the integrated system

capabilities in terms of performance, functionality, certifiability, robustness and system lifecycle costs.

The baseline technologies include all embedded platform modules and components such as networks, middleware, real-time operating systems, with appropriate models of computation and communication (i.e. layered alignment), which support flexible application hosting and inter-process communication. The capabilities are all means and methodologies to define, configure and assess performance of embedded platform components, to align, verify, model and simulate their performance, and to structure scalable, reconfigurable, generic integrated modular architectures. The simplified design and deployment of advanced integrated architectures for next generation TCMS relies on baseline technologies, embedded platform modules and components. It also depends on capabilities and methodologies to design, integrate, verify and certify complex integrated systems. This means that the methodology for








configuration and design of integrated systems is as important for system lifecycle costs and complexity management, as the embedded platform component properties.

1.2.1 High-level requirements

The system-level high-level requirements for next-generation TCMS are denoted in eight pillars. They are related to different functional and extra-functional features, expected of an integrated modular platform that hosts safety critical and non-safety-critical functions.

Figure 1: The 8 pillars on which the IMP is conceptualized

- 
Pillar 1: Next generation TCMS will **simplify integration** of many functions on common computing and hardware resources to reduce system lifecycle costs for design, integration, V&V, testing, maintenance, upgrades, modifications, extension, incremental certification and modernization and reuse.
- 
Pillar 2: Next generation TCMS will be **highly available and highly reliable** integrated platform, to serve the TCMS functionality.
- 
Pillar 3: Next generation TCMS will operate as a **fault-tolerant distributed computer** hosting all TCMS and other brake-by-wire, signalling, safety line, and non-critical applications.
- 
Pillar 4: Next generation TCMS platform will be able to **integrate all critical and non-critical functions** relevant for train operation, including functional, performance, safety, security, availability and integrity requirements.
- 
Pillar 5: Next generation TCMS will support **independent** design, testing, V&V and certification/homologation of functions.

- Pillar 6.** Next generation TCMS platform will **establish and guarantee timing and performance of all critical functions**, based on system integration configuration.
- Pillar 7.** The IMP will support a robust **(re)configuration management system** that is easy to maintain to allow for (re)deployment of functions and changes in train configuration.
- Pillar 8.** The platform must provide **interoperability** with respect to different and changing train configuration at functional and system integration level.

1.2.2 IMP Concept

The goal of the Integrated Modular Platform (IMP) is the facilitation of system integration, interfacing and information transfer from one application partition to another application partition in the networked system. It focuses on all system integration capabilities required to define an integrated modular platform which can host different TCMS, door control, braking, safety or non-critical functions in one system. The integrated modular platform hosts application functions and provides specific services to critical and non-critical applications, to establish robust software abstraction and provide all resources and timely information (sensors, global variables) access to applications.

The IMP does not depend on applications. Modular applications hosted on an Integrated Modular Platform can be tested in isolation and integrated on the system, without unintended interactions and interdependencies. The IMP represents the lower part of the integrated system, see Figure 2.

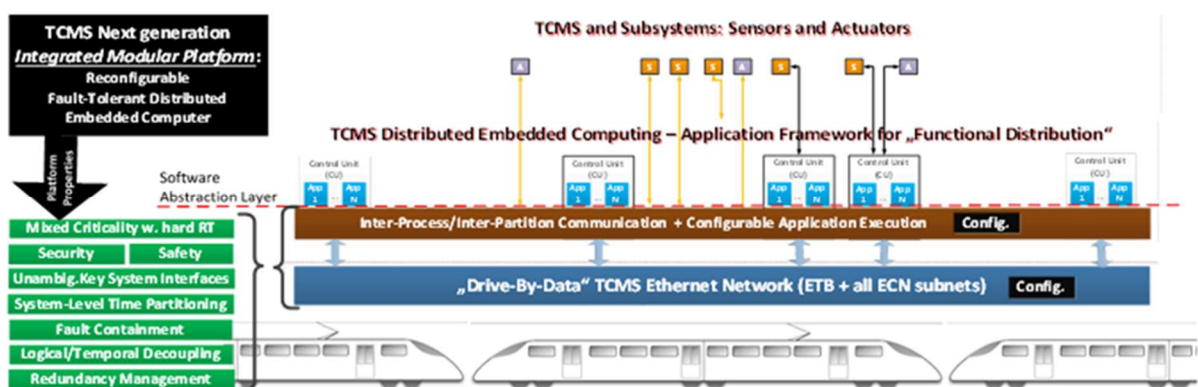


Figure 2: Integrated Modular Platform high-level overview

In the figure, the inter-process and inter-partition communication and configurable application execution are part of the so-called Functional Distribution Framework (FDF, highlighted in brown colour), whereas TCMS Ethernet network represents the inter-node communication system or the so-called Drive-By-Data framework. Both parts are described in detail in the next two sections.

The IMP approach implies a paradigm shift from the current federated architecture of loosely connected applications to a much more integrated view, where resources are fundamentally shared between multiple applications. The IMP is a subsystem, whose only function is to host different applications. The configuration of this subsystem, since it is underlying all other

applications, is of fundamental importance and must ensure the safe and reliable operation of all applications that make use of it.

The usage of the IMP in a safe and reliable context is ensured through its safe-by-design components, combined with a unified methodology to its configuration. The configuration of integrated modular platform components adapts the integrated modular platform to a specific use case and topology or architecture.

The Integrated Modular Platform may use different components for hosting a software platform, such as:

- 1) Computing units with integrated Ethernet Switching and Interfacing
 - Single-core Microcontroller Unit (MCU) with Memory Protection Unit (MPU) and Ethernet interface (smart sensor/actuator)
 - Safety MCU with memory protection unit and Ethernet interface (smart sensor/actuator, remote/safety I/O, central computing unit...)
 - Single Core SoC with MMU (memory management unit) for central computing unit with hypervisor/RTOS
 - Heterogenous multicore MCU, a core with MMU (memory management unit) with Ethernet interface (smart sensor/actuator, remote/safety I/O, switching, central computing unit with hypervisor/RTOS)
 - MPSoC with MMU in different variants, including Ethernet switching and interface
- 2) Computing units without integrated Ethernet Switching and Interfacing, or not using internal peripherals
 - Single-Core MCU with memory protection unit (MPU) and external Ethernet interface (smart sensor/actuator, remote/safety I/O, central computing unit...)
 - Computing multicore with external Ethernet interface (remote/safety IO, switching, central computing unit with hypervisor/RTOS)

Different variants will require different models of alignment between software platform and system integration/networking. In addition, different RTOS kernels and hypervisors, or separation kernels will have different models of driver integration and time/space partitioning capabilities. If SoC components cannot offer evidence on non-interference between different peripherals, safety requirements will demand physical partitioning.

1.2.2.1 Network services offered by Drive-by-Data

The next-generation TCMS concept is inherently distributed in nature and the network is an enabler for functions to reside anywhere in the system, as if they would be hosted in an “embedded cloud”. Hosted functions can have strictly deterministic access to other resources in the system, unaffected by other activities. Drive-By-Data (DbD) translates into predictable and safe technology for fully distributed electronic control and comprises the networking technology to achieve this as well as the configuration methodology on the system level.

The DbD concept relies on deterministic Ethernet as the mixed-criticality technology for ensuring strict separation of traffic streams, guaranteeing controlled bandwidth, latency and jitter. The network in an advanced integrated architecture can host all system function traffic with different timing and safety requirements, assuming there are sufficient network device and bandwidth resources available. It is aimed to support the partitioning of network bandwidth for time-critical & safety-critical functions (up to SIL3/4) and to integrate other less critical traffic.

This is a necessary enablement for the design of integrated TCMS architectures with mixed-criticality requirements.

An expanded and generic view to the next-generation TCMS, where the Drive-By-Data part is shown in more detail is provided in Figure 3.

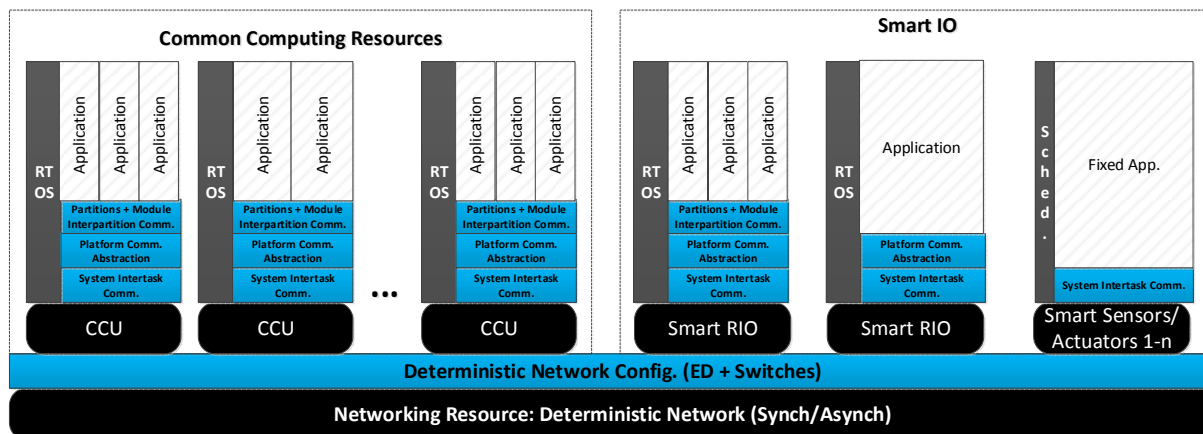


Figure 3: Generic embedded platform network, embedded computers and software platform components for next generation TCMS.

The on-board communication network for train operation is separated in a train-wide backbone network, the Ethernet Train Backbone (ETB) and set of Ethernet Consist Networks (ECN) that operate on the local level (one or several train cars). The Ethernet networking technology is a core technology for both ETB and ECN for the design of an integrated architecture and enables the convergence of functional integration in one network.

The network services are grouped in the following sets:

1. Services related to the data communication itself, in the form of time-triggered, rate-constrained and other Ethernet messaging
2. Services related to the train topology and potential changes/reconfiguration, e.g. due to coupling and uncoupling of trains
3. Services related to clock synchronisation and the dissemination of a shared global time
4. Service related to health monitoring and error handling

In terms of data communication services, the DbD concept is derived from the concept of a shared communication schedule, in which all nodes in the network dispatch periodic messages at a priori defined intervals (time-triggered communication). There are two leading mechanisms for this. The first is TTEthernet [8], used in avionics, industrial automation and automotive. The second is Time-Sensitive Networking (TSN) [9] based time-aware shaper for the use of periodic control messages, which have similarities but also differences which may influence system design constraints. In time slots where no scheduled communication is foreseen, other types of periodic messages can be communicated. These can either be bounded in latency through rate-constraining mechanisms such as defined in the avionics AFDX [10] or TSN credit-based shaper standards. These basic services ensure the separation of traffic streams and the strict control of latency and jitter for critical traffic. The services are implemented partly

in the network nodes but mainly in the network switches that police incoming and/or outgoing messages to respect and maintain the defined communication boundaries.

Train topology services handle the required changes in the train network due to changes in the train topology during operation. Trains are not static entities such as airplanes or cars, and the network must consider reconfiguration scenarios, such as trains coupling/uncoupling in many permutations, change of the train direction, or network topology change. The train topology services take care of gathering all topology-related information from the train from a bottom-up perspective. Distributed train topology services are largely standardized in IEC-61375 [11]. This information is subsequently disseminated to all elements of the train and potential addressing issues are resolved.

Clock synchronisation services are required for establishing a shared notion of time which is an elementary feature for scheduled (hard) real-time communication. These services handle the fact that local clocks from the various networking devices drift in time. Particularly, the clock synchronisation services ensure the reliable distribution of timing information as well as the assurance of correct time information under fault conditions.

Health monitoring services assess during run-time of the network the communication system (e.g. network nodes and switches) to ensure that hardware or software errors are detected. In such a case local or global fault reaction can be executed, depending on the type of fault. Examples include the restart and recovery of an individual component or the disabling of a component.

1.2.2.2 Functional Distribution Framework Middleware

Functional Distribution Framework (FDF), the application framework concept for modular integration of TCMS applications, aims to host distributed safety-critical and non-critical application side-by-side on the same hardware platform in distributed next-generation TCMS systems. This solution will have to fulfil functional safety-critical and non-critical requirements and non-functional requirements (including security) that support functional distribution, interoperability, reconfiguration, deterministic inter-partition communication, hardware and communication abstraction and virtual coupling of services. The Functional Distribution Framework for the next generation TCMS needs to fulfil a set of requirements, in order to overcome today's TCMS limitations and provide further functionalities and enhancements.

The main goal is to provide the "Functional Distribution" architecture concept for a mixed criticality embedded platform, offering an execution environment for distributed TCMS safe and secure applications up to SIL4. This execution environment must ensure a strict temporal and spatial partitioning, location transparency and abstraction from the underlying network protocols and hardware. Figure 4 illustrates the FDF layer in the context of two train cars of a train. Two Ethernet Consist Networks (ECN) can be seen, one for each train car and a set of Electronic Control Units (ECU) distributed among the ECNs. Some applications of different SIL run in the ECUs and the Functional Distribution Framework abstracts these from everything underneath.

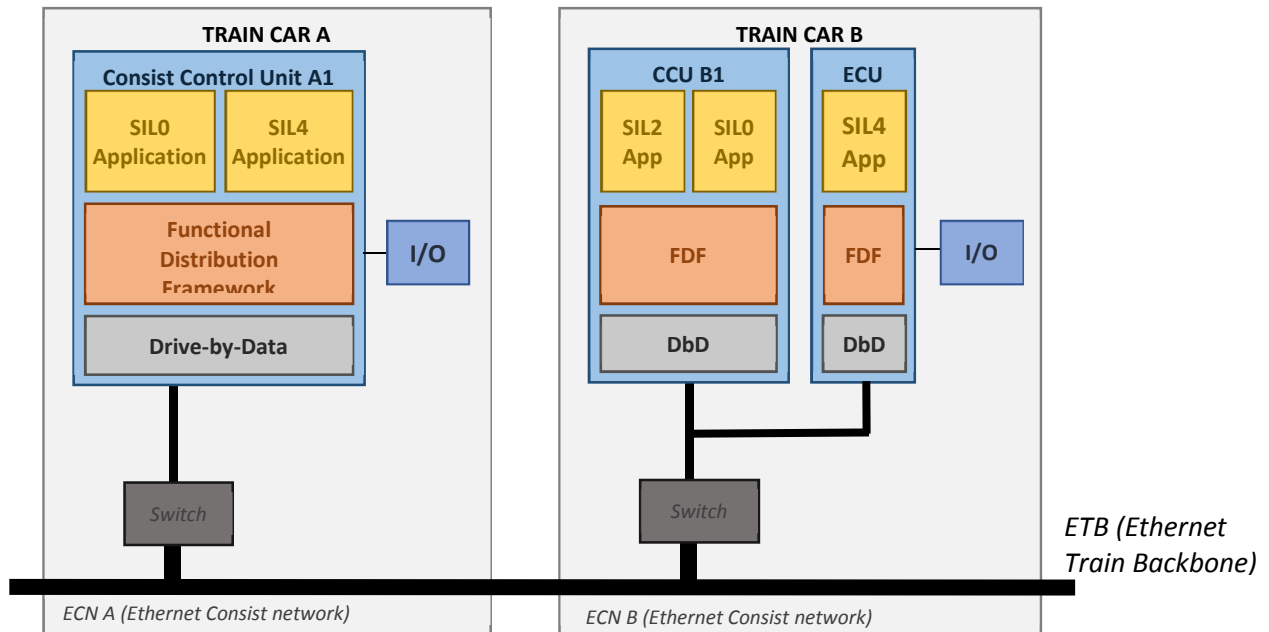


Figure 4 FDF in the context of a train (simplified)

In order to achieve the previously mentioned goals, the Functional Distribution Framework must provide a set of services:

- Initialization.
- Global clock synchronisation.
- Scheduled execution of applications (of different SIL).
- Safe local data distribution.
- Safe and secure remote data distribution.
- Transparent I/O reading and writing.
- Health-monitoring.
- Remote monitoring.
- Logging.
- Deployment, which means providing the ability to update an application without altering the rest.

The Functional Distribution Framework for the next generation TCMS needs to fulfil a set of requirements to overcome today's TCMS limitations and provide further functionalities and enhancements. These are described in the following lines.

- **Technical Characteristics**

- *Configuration and management services*
 - Configuration services
 - Partition management
 - Process management
 - Function management
 - Time management
 - Memory management
 - Communication management
 - Data exchange management
 - Incremental certification and re-certification of applications
- *Time services*
- *Input/output services*
- *Real-time support*
- *Fault isolation*

- *Health monitoring and error-handling*
- *Safety services*
- *Security services*
- *Requirements for underlying hardware:*
 - sufficient processing capacity
 - granted processor access to required I/O, memory and time resources
 - processor providing atomic operations for implementing processing control constructs and a mechanism to transfer control to the OS if the partition attempts to perform an invalid operation
 - interrupts strictly forbidden from disturbing the time partitioning
- **Non-technical characteristics**
 - *A need for System Architecture Engineering Method*
 - *Safety and the relevant standards*
 - *Security and the relevant standards*

The FDF abstracts the applications from the Input/Output management and the network. Therefore, in the end, the FDF is an abstraction layer from the I/O Management and communication. The FDF also abstracts the applications from the synchronisation of the different FDF nodes within a consist, since it is responsible for getting the global clock from the network and updating the system clock of each concrete ECU.

Chapter 2 Train Topology and Architecture

The basic network topology originates from the classic train car/consist notion, where, in opposite to the automotive or avionic use cases, the overall network topology is not constant. Cars or consists can be coupled, train composition may change. A train usually exists of one or multiple consists, equipped with the same network (and device addresses).

2.1.1 ETB Topology

This subchapter addresses the physical topology and wiring considerations related to the interconnection between consists, i.e. the Ethernet Train Backbone (ETB).

The ETB topology considerations, at this lowest level of the system, are linked to the first two pillars of the high-level requirements (see 1.2.1)

- Pillar 1: Simplification of the integration, fewer cables and components
- Pillar 2: Increased availability and reliability of the network

The simplification requirements relate here to the reduction of the number of cables and components in order to reduce cost and simplify the overall topology for the entire communication architecture within the train (including not only Ethernet, but also WTB, MVB, safety lines).

Availability is important to keep the system in operation. The availability of a system depends on the system's design reliability and maintainability. Therefore, a core topology requirement relates to fault-tolerance: no single component failure shall lead to the unavailability of the network. For this reason, redundancy of all components must be taken into consideration in the topology.

2.1.1.1 Solution

The physical layout and connectors of the Ethernet-based TCN backbone (ETB) is described and defined by the standard IEC61375-2-5 [1], whereas IEC61375-3-4 [2] describes the use of Ethernet on consist layer (ECN). The currently defined 100BASE-T (FastEthernet) variant must be extended to at least 1000BASE-T to account for future bandwidth needs. Especially train wide video surveillance needs more bandwidth than the current standards provide. There are several limitations originating from the current design and they must be adapted for future train network layouts.

To execute the inauguration as defined in IEC61375-2-5, the ETB must contain two Ethernet lines and two ETBNs in series, for redundancy. The two ETBNs of each consist act as a redundancy pair (leader/follower) and a bypass relay ensures ETB communication in case the ETBN fails, as shown in Figure 5. The redundant pairs have the same static configuration for non-scheduled traffic. However, failure of an ETBN will not lead to an inauguration, but will lead to changes in packet timing, in case of ECN <-> ETB scheduled traffic. A reconfiguration of the ECN switches will be necessary.

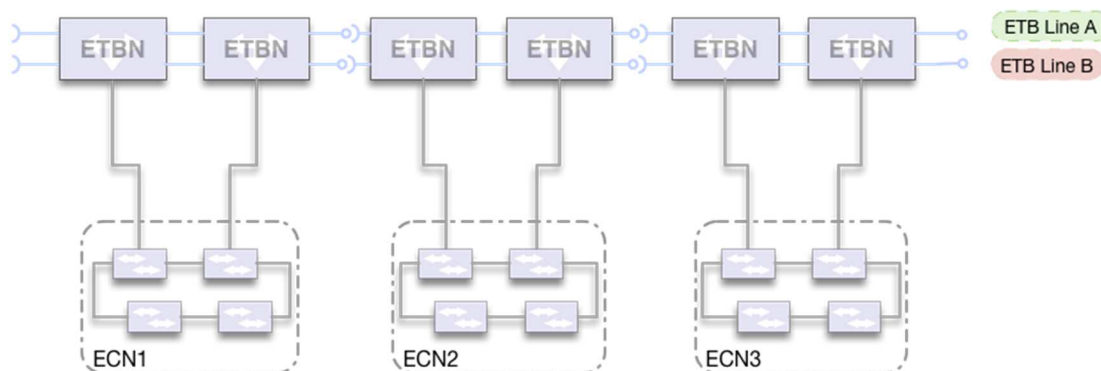


Figure 5: ETB in serial redundant layout

To counteract this problem and adjust the ETB to statically configured deterministic scheduling needs, the two Ethernet lines and two ETBNs will be connected in parallel, for extended redundancy. Each ETBN of each consist acts as a separate connection to the connected ETB line, see Figure 6.

A bypass relay is not necessary; failure of one ETBN will partially omit communication on that ETB line only.

The redundant pairs have different configurations for non-scheduled traffic, and ETB inauguration as defined in IEC61375-2-5 must be modified to prevent inauguration and/or reconciliation in case of a single ETBN failure.

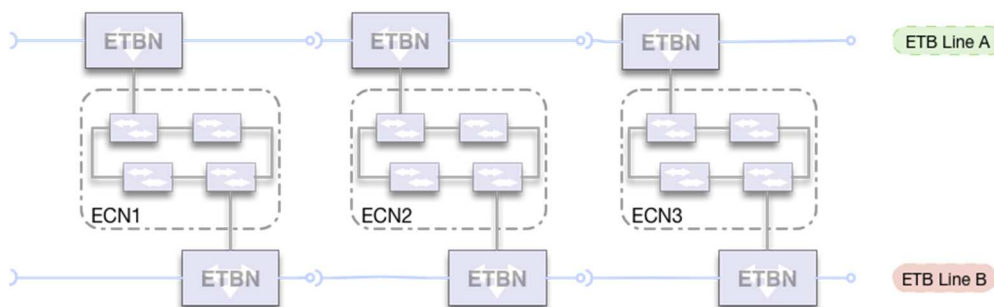


Figure 6: ETB in parallel redundant layout (Variant D)

ETBNs connecting Line B will work as redundancy follower for non-scheduled traffic (hot standby, keeping the same state).

2.1.2 ECN Topology

The next-generation consist network (ECN) shall be able to handle all communication services within the consist. The requirements to a consist network in terms of real-time performance are higher than they are for the backbone network, as it must be able to handle distributed control systems with tight control loops.

Consists can be divided into two major types with different requirements related to functionality and safety:

- Active Consist - can be a stand-alone train, providing at least one leading cab. This includes commuter trains, locomotives, metros, LRV, high-speed trains
- Passive Consist - loco-hauled coaches, no leading cab, usually a single coach. This includes traction-less coach or group of coaches.

The TCMS relevant functions provided in a consist network are of rather static nature and will be changed on initial commissioning of a train, only – not during normal operation. There are exceptions to this depending on the train vendor; removing from or inserting coaches into a consist must be accounted for and will lead to a reconfiguration of the normally static consist network.

In this chapter, the aspects that are highlighted are topology considerations, taking into account the virtualization that the network offers. Note that this analysis does not consider specific technological capabilities from different standards and implementation choices are left to system designers.

The critical point in this design aspect is the integration between the consist network and the train backbone, which is where interoperability plays a crucial role.

2.1.2.1 Topology virtualization

Multipath and network redundancy can be deployed in complex networks by selective policing of traffic on some links. Therefore, a fully meshed network can for some applications behave either as a ladder, ring or dual redundant line, or something else depending on the configuration. This will enable multipath redundancy and virtual network redundancy.

Applications with different levels of safety-integrity will anticipate only a specific virtual topology, even if it is configured on top of a different physical topology. Different applications will not “see” or anticipate virtual topologies configured for other applications. Due to topology virtualization, different applications will act as if they would work on their own physical network separated from all other applications and dedicated resources.

Topology virtualization is based on the policing of all deterministic dataflows (streams) on every switch in the network. Furthermore, a policing mechanism needs to be periodically tested for its function, to avoid unintended transfers or mutual interactions between virtual topologies.

The policing configuration determines which path can be supported for different dataflows, and in addition those dataflows are switched at L2 as defined by configuration.

Therefore, the topology virtualization approach relies on:

- Configured L2 switching so that data can be forwarded over a fixed path
- Per-flow policing to monitor if the preceding switch on the path really forwarded the right frame to the right port, with appropriate timing and periodicity

so that different applications will act as if they would work on their own physical network separated from all other applications and dedicated resources.

Principally, different consist topologies can be chosen to connect the end-devices (EDs) and critical End Devices (cEDs) to the network. Recommendation of one topology over another is very much dependent on the physical placement of the end-devices in the train. Important to point out is the fact that the virtualization approach offered by the network as well as by the integration in the End-Device does not restrict the choice in a topology between the various topologies.

Even hybrid solutions can be used, e.g. as depicted in Figure 7, where a virtualization concept is used in order to virtually separate two networks, combining virtually the different Networks (A and B) which are physically interconnected

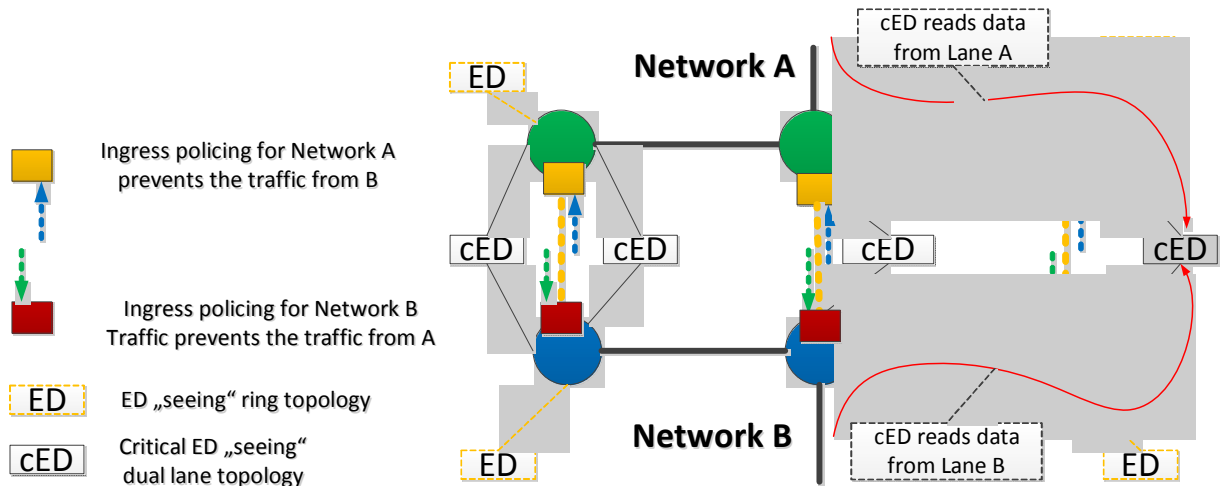


Figure 7: Topology virtualization approach: Policing to create dual independent lanes out of a ring or ladder network

2.1.2.2 Preferred Solution

The favourite network topology on ECN level is a combination of a ladder and a ring layout as depicted in Figure 8. Safety related devices with high RAMS demands will be connected by two physically separated Ethernet ports to separate switches with separate logical planes.

In case of a break-up of the ring, the focal point will close (the left-most switch) and non-scheduled traffic might be redirected. In case the connection to ETBN-A is affected, ETBN-B must take over and provide the routing between ECN and ETB Line B. Devices with high RAMS requirements (using new TSN features) do not need to switch the communication because scheduled data will always be forwarded on both planes simultaneously (see Deliverable D1.4 [3], Figure 50: Topology virtualization).

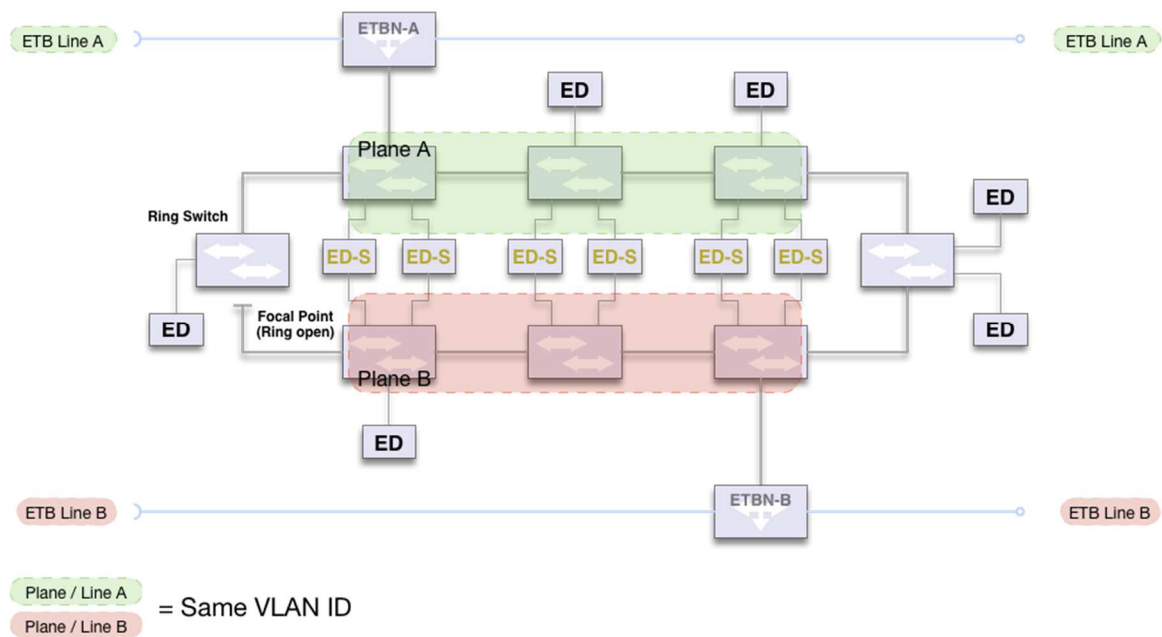


Figure 8: ECN layout: physical ring, logical ladder

For redundancy and reliability, there will be two physical connections between the ECN and the ETB:

- 2 ETBNs as a redundant pair (master/slave or leader/follower) for common lane traffic (non-safe). Access points for wireless sensors (e.g. bogie temperature status) can also be redundant and attached to Plane A and Plane B via separate access points.
- The ETBNs will also provide the mapping between local Plane A resp. B traffic and ETB Line A resp. B. The second ETBN will be a redundant node for non-scheduled traffic; scheduled traffic will always be routed (mapped) to the ETB on both ETB lines.

The network topology as seen by CTA is provided as presented in Figure 9.

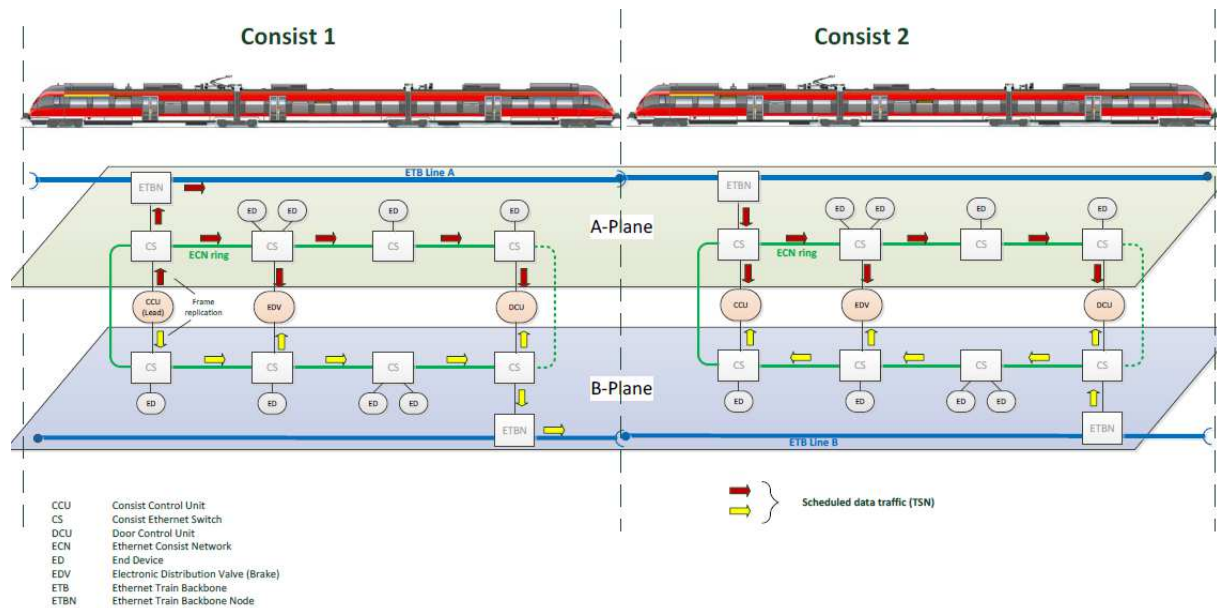


Figure 9: ETB/ECN Layout proposed by CTA

Chapter 3 Inauguration Concept

Definition: Train inauguration in the sense of this document is the process of determining the sequence and orientation of all vehicles and consists which make up a train.

Two stages or kind of inauguration are defined:

- ETB inauguration - discovering the ETB topology and generate the train network directory (TND), inhibiting train inauguration on demand, indicating train lengthening/shortening.
- Operational train inauguration - computing the TTDB after train composition change or after train leadership change.

As the inauguration determines the addressing of direction dependent functions in the train, it will determine traffic routes (left doors/right doors, traction distribution) and thus will lead to a change of the TCN configuration.

3.1.1 ETB Inauguration

The ETB train inauguration is defined and described in IEC61375-2-5 [1] and can be used on the NG-TCN without modifications, if the Variant B layout (see chapter 3.1.1.1 of D1.6 [4]) is to be used.

ETB inauguration uses two different TTDP frames, the “HELLO” frame (IEEE 802.1AB [12], LLDP) and the “TOPOLOGY” frame (IEEE 802.1D [13], multi cast). HELLO frames are sent to the direct neighbours, only, through directed switch ports (direction 1 and direction 2) with time interval of 100 ms or 15 ms for SlowPeriod or FastPeriod, respectively.

TTDP TOPOLOGY frames are sent with a 100 ms interval to all ETBNs in both directions.

All TTDP messages shall be VLAN tagged in accordance to IEEE 802.1Q [14]:

- VLAN identifier (VID) shall be set to 492 (= ‘1EC’H),
- VLAN priority shall be set to highest priority 7.

To support TSN traffic over the ETB, priority schemes and values for TTDP messages have to be adjusted according to Chapter 3.5.2 of D1.6 [4].

The Variant D of the ETB layout demands special handling because of the physical separation of the two redundant ETB lines. This is the current proposal (from CTA):

The modified ETB inauguration needs synchronization between the two ETB nodes. The ‘Hello’-frames, which each ETBN sends to its neighbour, must carry the same MAC address for each direction and each ETB line. This ensures that the topology counters for both sides of the ETB are the same.

There is a mutual exchange between ETBN-A (Left) and ETBN-B (Right) MAC addresses across the ECN (dotted line [some sort of “proxy” function]). ETBN-A and ETBN-B both show the MAC address for ETBN-A towards Dir1 and MAC address for ETBN-B towards Dir2. This is true if redundancy is available (both ETBNs are alive). If only one of the redundant ETBNs is alive its MAC address is used towards both directions (Dir1 and Dir2) instead. The ETBN-A provides its MAC address to ETBN-B (see dotted blue line in Figure 10).

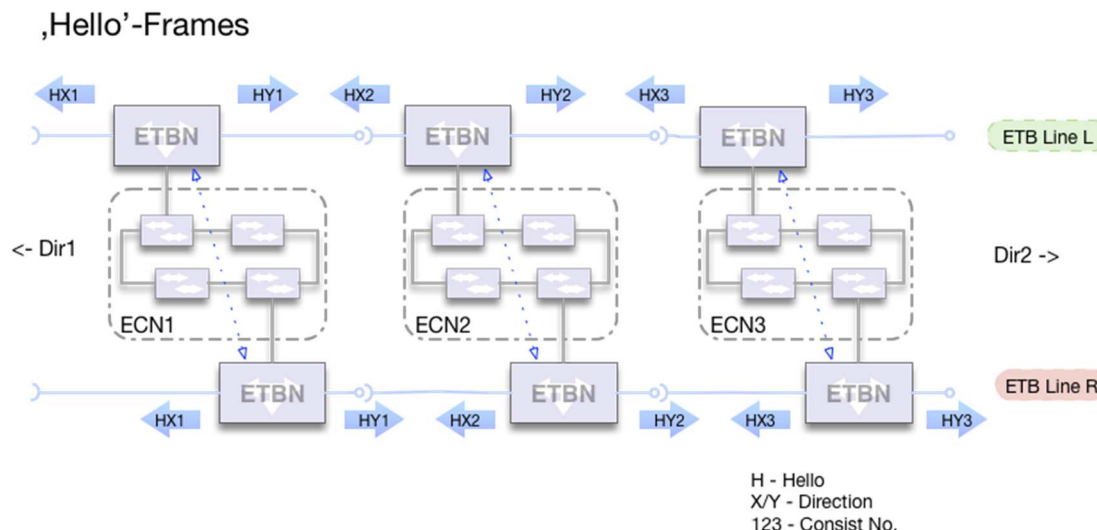


Figure 10: Inauguration with separate ETB lines

An ETB inauguration will result in the same:

- 'Connectivity table' with the MAC addresses of all nodes on the ETB line
- 'Train Network Directory', listing all consists (UUID) attached to the ETB
- etbTopoCount, a CRC over the Train Network Directory

for all ETBNs.

A switchover between the redundant ETBNs will not lead to a change in the etbTopoCount (which would indicate a new inauguration), because they both use the same MAC address.

To prevent traffic congestions, LLDP traffic and TOPOLOGY multicasts will be policed on ingress (and egress ports), in order to prevent a bandwidth isolation breakdown.

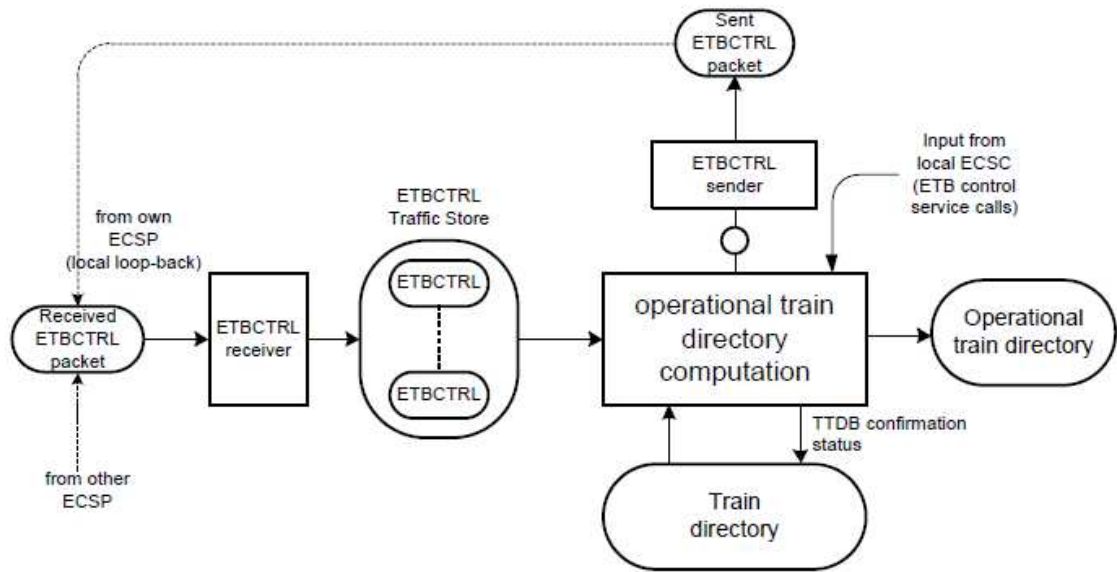
3.1.2 Operational Train Inauguration

The result of the ETB inauguration is the static, physical directory and sequence of the consists, assigned ETBNs and the orientation of the consist (direct or inverse). There is no information about the number, kind or orientation of vehicles inside a consist nor is there any information about functions or device addresses and properties. This information will be distributed and collected during the operational train inauguration (or UIC inauguration) as laid down in IEC61375-2-3 [15] and used to build up the TTDB, the train topology database.

Each ECSP in every consist computes the operational train directory by exchanging ETBCTRL packets periodically via the ETB. These ETBCTRL packets carry a number of flags, which represent the state of each consist. A change of these flags may trigger a state transition, which in turn will be reflected in the ECSP's own transmitted ETBCTRL packet.

These state flags will also be changed by a local ECSC function, which can be seen as the driver's control panel. When the driver operates the train (e.g. by activating the leading cab), the leading bit in the corresponding ETBCTRL packet will be set and all other consists have to confirm this. This way the leading (driving) or operational reference direction, opposed to the ETB reference direction, is determined.

The resulting data structure is the operational train directory and its status info is sent periodically to all EDs in the consist (PD OpTrainDirectoryState ComId 100).

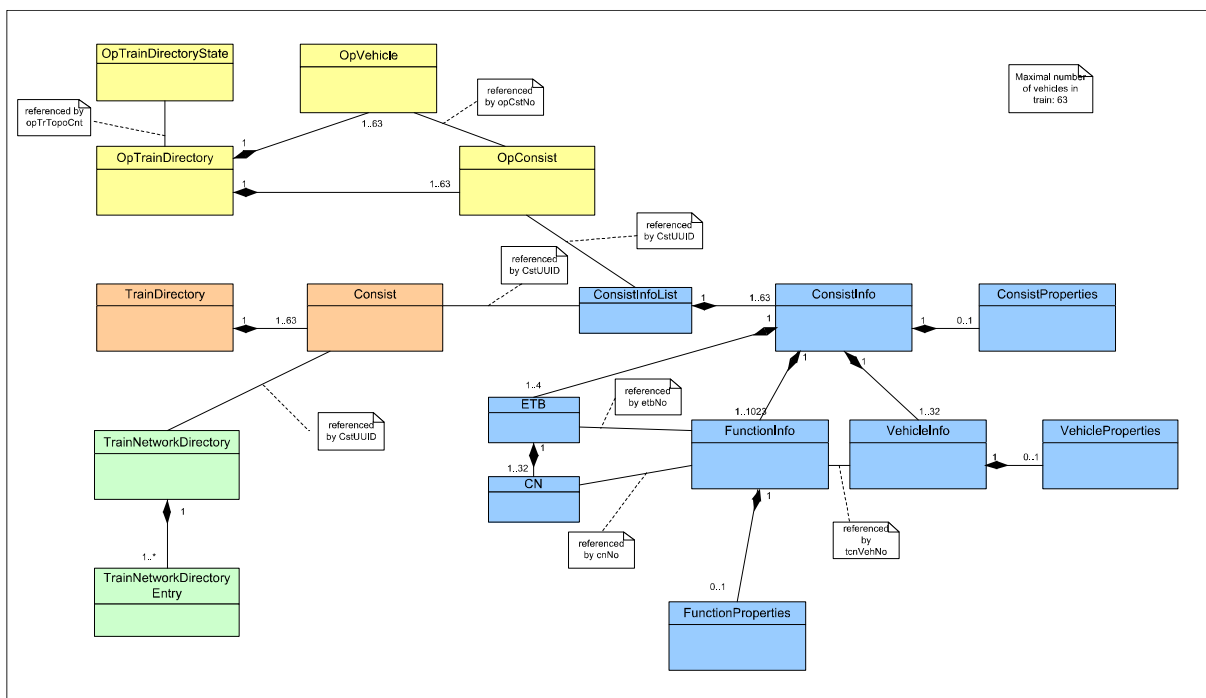


IEC

Figure 11: Operational train directory computation block diagram [15]

After a successful ETB inauguration, each consist sends its CSTINFO as a TRDP multicast message notification to all other ECSPs. Each ECSP collects these (and can request re-transmission, in case the packet has not been received) and, combined with the local consist info, builds up the TTDB.

The TTDB holds the static and dynamic view of the whole train (all consists, all vehicles). Figure 12 shows the data model (from IEC61375-2-3 [15], Figure 16):



IEC

Figure 12: TTDB Data Model [15]

Each ConsistInfo structure holds statically defined (among other):

- consist info and properties (UUID, type, owner, etc.)

- list of vehicle info and properties (type, orientation, number, etc.)
- list of function info and properties (name, fctId, multicast group, cstVehNo, consist network it belongs to)

The field 'fctId' corresponds to the last 14 bits of the IP address or multicast address, if set. The content of the function info can be used to address dedicated functions in remote consists and is used to feed the TCN-DNS service (URI/address resolving).

The meaning and content of the property field is application defined.

Although the size of one consist info telegram is limited to 64kB, the size of the memory to hold the TTDB for a maximum sized train can occupy several MBytes of storage. This is due to a.) MISRA rules and SIL compliant software requiring the use of statically allocated memory, only and b.) variable length arrays in several data structures. It is required (for SIL2) that the memory must be protected, or integrity checked (safe memory).

3.1.3 Safe Inauguration

Inauguration as defined in the current standard can only be implemented as a safe function up to safety level SIL2 – in current implementations the result of the inauguration (sequence of consists and their orientation) depends on the reliability of the switch core's port discrimination. The commonly used switch cores are COTS, and not necessarily designed following even EN50657 [16] basic integrity (former SIL0 in EN50128 [17]).

1. If the inauguration function (SW) on an ETBN fails because of a systematic error, it would not pass a conformance test (it could successfully inaugurate with ETBNs using the same software, though!) → **Detected fault**
2. If the inauguration function (SW) on the ETBN fails because of an HW error (memory corruption, CPU failure, etc.), it would indicate the failure of the inauguration (different/wrong topology counter/CRC) → **Detected fault**
3. If the inauguration function (SW) on the ETBN computes the same topology counter over a different, corrupted train directory than all other ETBNs, the error would not be revealed → **Undetected HW fault**, probability for CRC32 failure depends on HW design
4. If the inauguration function (SW) on the ETBN computes the train network directory with an undetected inverted direction (random or permanent error of the switch core / port exchange), it would lead to a valid topology counter and would not be detected → **Undetected HW fault**

While bullet points 1 and 2 can safely be ignored, point 3 (undetected bit flips in Train Network Directory) and point 4 (wrong switch port assignment) must be considered for the required safety level of the ETBN software. Additionally, swapping the switch port connectors on commissioning will lead to the same undetected fault!

However, when considering variant D of the ETB layout, a random or systematic direction inversion on one of the ETBNs of a consist can be detected – it will lead to different inauguration results for the two ETB lines.

The output of the ETB inauguration, the train network directory, is one of the inputs for the operational train directory computation, performed by the ECSP. The ECSP's result must be SIL2 conformant; most of the communication over the network is SDTv2 secured. A link between the ETB inauguration handler and the ECSP computation logic has been defined in the current standard (see Annex E.6 ETBN control interface of [15]) using SDTv2 protected message data communication. The used communication scheme unfortunately requires polling the ETBN for inauguration events.

➔ It is recommended to use a push or notification instead (ETBN → ECSP).

The safety relevant output of the operational train inauguration is

- the driving or operating direction relative to the train reference orientation (1 or 2)
- the orientation of the consists relative to the train reference orientation
- the orientation of the vehicles inside of the corresponding consist
- the position (address) of direction dependant functions (doors)

One SIL4 function identified so far is the train wide direction dependent operation of the doors. This function depends on the driving direction and the sequence and orientation of the consists.

Until now, a separate physical train line was used to verify the inauguration result independently to exclude systematic and minimise random errors. The new concept eliminates the need for such physical train line by the use of the Safe Train Inauguration concept which was introduced in CONNECTA D3.5 [18].

The concept is based on the split of inauguration and validation function between the ETBN itself with low-SIL capability (i.e. SIL2) and a device with high-SIL capability (i.e. SIL4) to perform the necessary validation of the inauguration result.

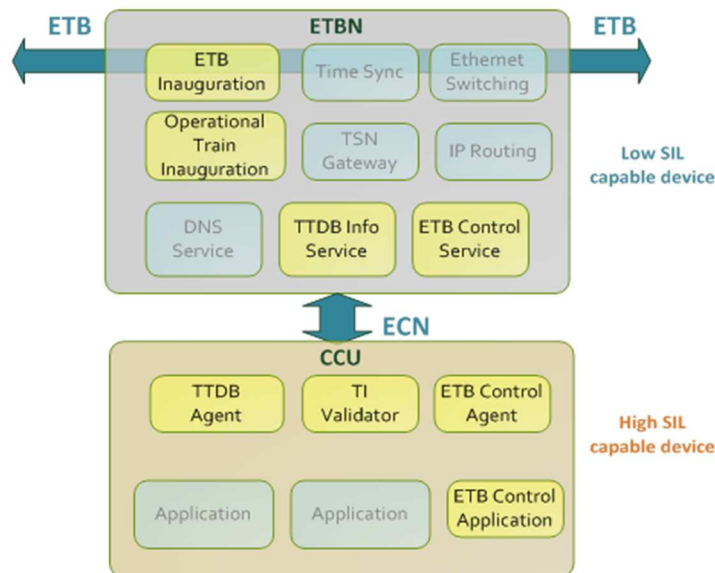


Figure 13: Cooperation of ETBN and CCU (source: [18])

To achieve an independent validation of the results of the inauguration, the fact that the topology requires two independent lines (ETB-R and ETB-L) can be utilized for performing this independent validation. This is performed by sending separate beacon messages that verify that each consist has the same understanding of which side is left and right and thus whether the result is correct, enabling a diverse approach with a very high reliability.

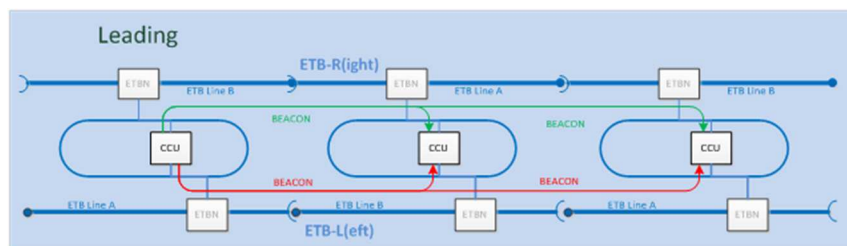


Figure 14: ETB lines usage for validation of inauguration result (source: [18])

Chapter 4 Networking Concept

4.1 Clock Synchronisation Concept

The demands of the real time communication in any train network are high precision clocks and synchronisation thereof. While the ECN topology is static, and the timing can be predicted, real time communication over ETB requires additional measures.

4.1.1 Introducing the Clocks

For TSN support, an active ETBN must carry a gPTP high precision clock (IEEE1588 [19]/IEEE802.1AS [20]) and be able to work as a standalone time domain, but also provide a synchronisation with other clocks on the ETB, if coupled.

One solution: Resulting from the computation of the ETB inauguration, i.e. the train network directory, at least one gPTP clock in the ETB is a grandmaster clock, providing the system time.

In accordance to PTP, two types of clocks will be used, OC (ordinary clock) and BC (boundary clock). The ordinary clock is an end-device, which operates either as a (grand-)master or as slave. The boundary clock acts as both – it has one port in slave status, receiving time from the master clock, and all the other ports have a master role, disseminating time to downstream slave clocks.

On ETB inauguration the role of the gPTP clock in each ETBN will be determined and the clock in consist 1 will become the grandmaster (see Figure 15 and Figure 16).

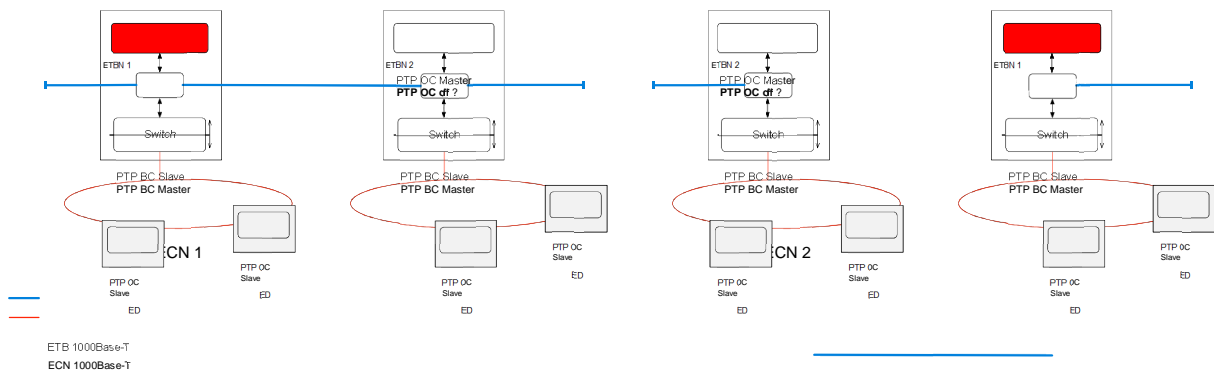


Figure 15: PTP: Two trains with two consists

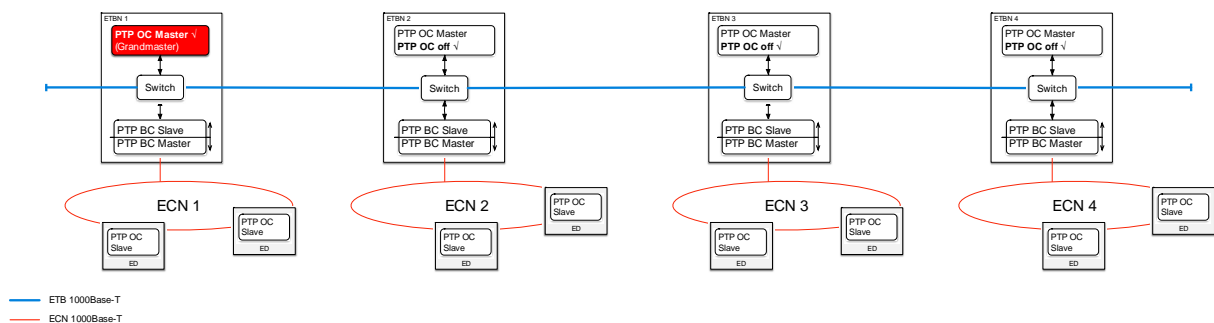


Figure 16: PTP: Trains after coupling and inauguration

Depending on the accuracy of the time basis, there could be a leap in time during inauguration and syncing of the clocks, which shall be tolerated by ECN networks and applications.

4.1.2 Robust Clock Synchronisation

A more robust clock synchronisation mechanism demands multiple clock masters, and flexible introduction of system clock at any point in the system, e.g. in ECN network or within ECN computers as described below. Robust synchronization is of exceptional importance for:

- mixed criticality communication
- defined deterministic performance of many real-time functions integrated on the system
- high-performance operation and optimized resource use
- incremental verification and certification

The approach to synchronization is designed to support high robustness and system availability in complex architectures with tens of hops.

A first (but not satisfactory) approach to clock synchronisation may use exclusively the clock synchronisation concept as defined in the 802.1AS-rev [21] standard. Such a concept revolves around 1-4 clock masters, but ETB network and related system-relevant integrated functions can also operate without any clock in the system, in a degraded mode. All clock slaves receive the available set of clocks, select correct clocks and isolate faulty clocks. In fault-tolerance terms, the limits of the TSN's 802.1AS-rev [21] are set to a relatively simple Grandmaster Clock's fail-silent type of failure (i.e. the system either provides the correct service, or provides no service at all), based on a redundant grandmaster, or through BMCA. In short, the standard supports two redundant master clocks – one primary grandmaster clock and one secondary, hot-standby grandmaster clock, synchronized to the primary clock. If the primary GMC becomes silent, the hot-standby clock can take over the role of the main clock source in the system.

Figure 17 shows the Safe4RAIL clock synchronization concept with these basic redundancy mechanisms. Each redundant ETB line is operating as a separate clock domain, i.e. it is synchronized within itself, with a primary Grandmaster clock, *pGM*, and secondary Grandmaster clock, *hsGM*. ECNs as well are separate clock domains each, and communication with both ETB lines is asynchronous.

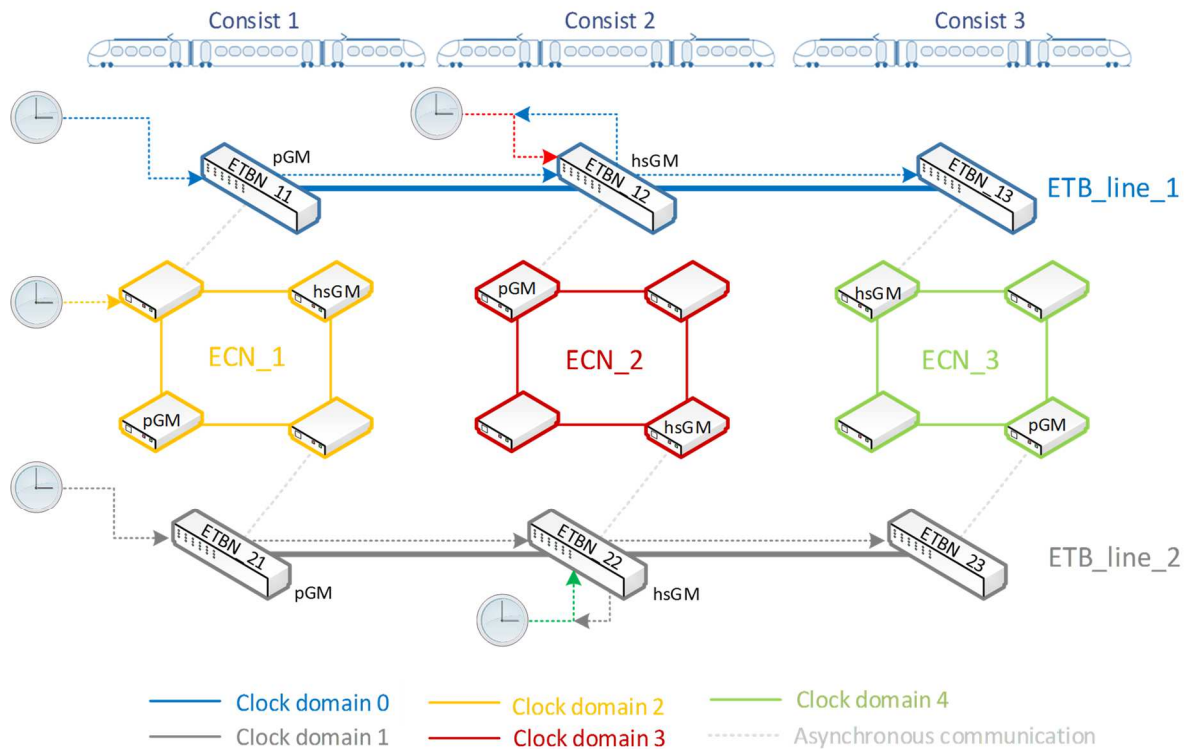


Figure 17: Safe4RAIL system based on IEEE p802.1AS-rev [21] standard’s ready mechanisms

As indicated above, these mechanisms alone are not enough to satisfy the fault tolerance requirements for clock synchronisation. Implementing only standard’s ready-mechanisms, this system has no tolerance to failures of another nature, such as frame omission failure (transient/intermittent transmission, reception or relay), invalid frame failure (producing faulty CRC, calculating faulty CRC or changing CRC value), incorrect frame failure (transmitting/receiving faulty data, changing correct data into incorrect), untimely frame failure (transmit/reception out of time, or faulty delay). [22]

Each component of the Safe4RAIL system – clock source, ECN, ETBN/ETB - can potentially impose faults, as shown on a Figure 18, with combinatorial and accumulating effects. A detailed assessment of the failure modes is presented in deliverable D1.7 [5].

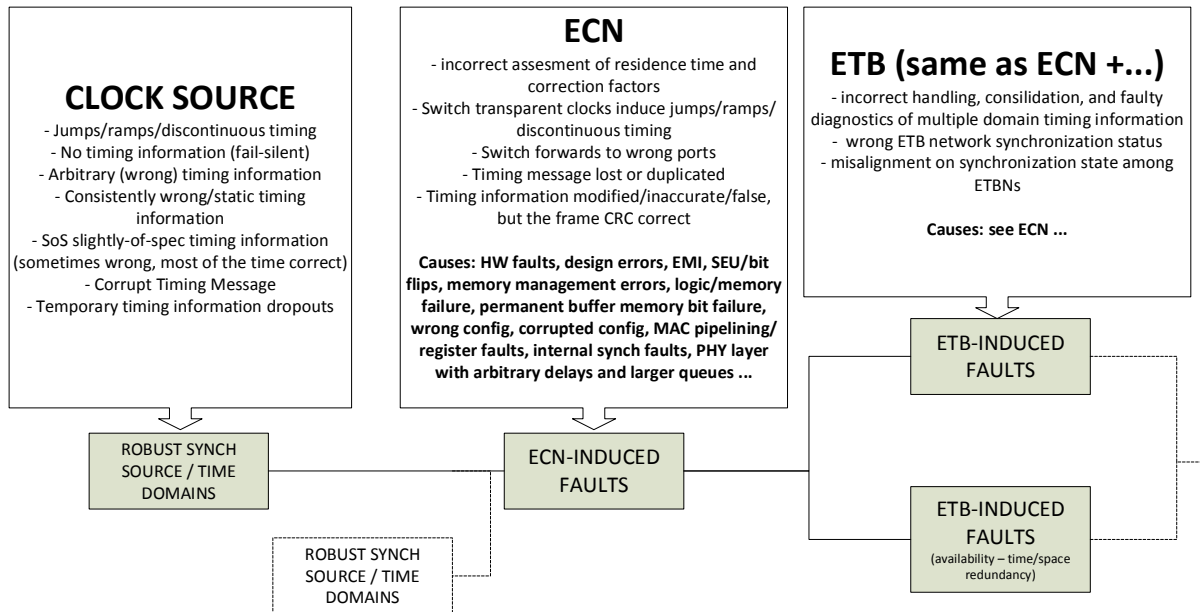


Figure 18: Failure Sources within Safe4RAIL system

The following two chapters describe the mechanisms introduced by Safe4RAIL to achieve the required fault-tolerance, i.e.

- (1) a mechanism to fall back to deterministic communication in the presence of faults, the so-called degraded mode, and
- (2) the establishment of a fault-tolerant “synchronisation domain” based on multiple gPTP domains that span the ETB.

4.1.3 Fault-Tolerant Clock Sync Concept for ETB and ECN

Based on previous fault tolerance considerations, the old concept, relying exclusively on 802.1AS-rev [21] ready mechanisms, was adjusted to enable failure detection beyond the simple fail-silent case.

4.1.3.1 ETB Synchronization

On power-up, the ETB lines/ETBNs are asynchronous and exchanging critical traffic only. Any other traffic is blocked, to ensure the guaranteed latency and bandwidth for high-importance frames in an asynchronous system.

With the appearance of the first sync messages coming from system’s GlobalMC (a Global Master Clock is defined as a clock source that contributes to the establishment of the synchronisation domain that spans the ETB), each ETBN will go through an iterative synchronization process, leaning on identification of correct data information from available clocks. After a successfully executed synchronization process, every ETBN is confirmed to be synchronized and all the nodes of an ETB line have the same synchronization perception. The ETBs are now considered synchronized and fully operational. In this state, they are exchanging mixed criticality traffic (as opposed to critical-traffic-only so far).

Very important for deterministic performance is to warrant the simultaneous start of the synchronized operation for all ETBNs.

In case of the synchronization disturbance, where an ETBN is unable to synchronize its clock to the global GMC time due to its own failure or due to an absence of sync messages, the

ETBN in question must become inactive after a predefined grace period, during which the ETBN's local clock runs freely but with a known maximum drift. In inactive state, the ETBN will not communicate further with the rest of the network and waits for re-inauguration.

4.1.3.2 Distributed Set of High-Integrity Clocks

Each ECN in the Safe4RAIL network has one or two clocks which are designed as high-integrity clocks, meaning that they:

- identify and mitigate abrupt time changes, jumps (transient) and ramps (permanent)
- identify their own faults
- minimize and avoid chance of injecting synchronization faults into the system and impeding system operation

Design of high-integrity clocks may require design assurance methods which can guarantee high-integrity behaviour, and transition into a fail-silent state until the next power-up. Each clock will send data over several gPTP domains for redundancy. In a system with 4 clocks this implies that at least 8 gPTP domains will be configured.

Multiple GlobalMCs are needed to increase the ability of fault detection and fault distribution mitigation.

By using two clocks, detection potential is available but fault-tolerant operation is very limited. Clocks can be compared, and it can be detected that something is wrong, but it cannot be determined which clock is failing, so the operation cannot be continued safely. All the nodes should retract to Degraded Mode. With three clocks in a system, 2oo3 voting can be applied and work well in case of simple and consistent failures, such as when the failure appears the same way at all nodes. If four clocks are used to provide time in a system, mechanisms such as fault-tolerant averaging can be applied to rule out wrong timing information. In large systems such as a train with several consists and thus a high number of hops, chances of multiple errors are higher and different units can perceive the same master's time differently. With help of the averaging, unexpected glitches in time can be isolated and prevented from influencing the time dissemination in the system.

After the power-up, GlobalMCs may synchronize to a trusted NTP or GPS clock to get an initial time base and function as an unsynchronized system. After successful inauguration, they will start the initial startup and comply to hierarchy-based on priorities rather than a Best Master Clock Algorithm. All of them have a priority assigned and the one with the highest priority will initiate the synchronization by disseminating its local time as a global time, and transit to a clock integration state. The other master clocks will follow sequentially, based on their own priorities, as shown in Figure 19. As soon as a lower priority clock (e.g. GMC 2) receives the time information from a higher priority one (e.g. GMC 1) and synchronizes to it, it can forward it to all the other master clocks, with lower priority than its own (e.g. GMC 3 and GMC 4).

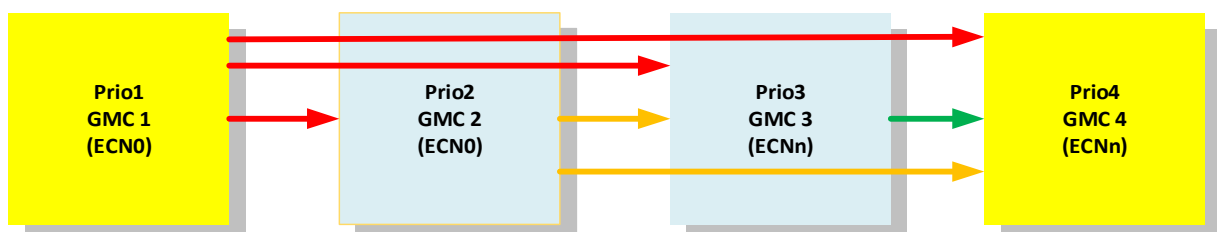


Figure 19: GlobalMC's synchronization startup according to assigned priorities

Once all have adjusted their time to their higher priority master clocks and after each of them has the same perception of synchronization status(es) in the system, they can be considered synchronized. During synchronized operation, priorities lose their role, and the clocks behave democratically, calculating a fault-tolerant average of time based on their local clock and clocks provided from the other clocks. Plausibility of the calculated average will be proofed, and the valid value will be used to adjust the local clock.

There are two options for formation of synchronization domains within a train, with their drawbacks and benefits:

Separate synchronisation on ETB and on ECN, no synchronisation between the domains

ETB and ECNs are communicating exclusively asynchronously. This option is enabling an undisturbed re-coupling/de-coupling of ECNs, without time jumps. The master clocks are placed in pairs per line on ETBNs in the first and the last ECN of the train, in order to establish disjunct paths for the clock information distribution, as shown in Figure 20.

These 4 clocks disseminate the time information through eight clock domains to the entire ETB, i.e. to both communication lines. Other ETBNs also hold a pair of MCs which will not actively participate in the ETB synchronization as long as they do not become the leading or tailing ECN. Since each GlobalMC is providing time on both ETB communication lines and through two different dissemination paths, it also creates two gPTP clock domains. ECNs themselves also host four MCs each, for their local synchronization.

ECNs can communicate synchronously to ETB.

This option enables a fully synchronized train communication and therefore very tight end-to-end latency across the network, even over many hops. However, as a drawback it complicates the re-coupling/de-coupling of ECNs, due to unavoidable time-jumps in ECNs that accompany it .

The difference to the first option is the placement of master clocks - in pairs within the first and the last ECN of the train, therefore making these ECNs synchronized to the ETB, as shown in Figure 21. Every other ECN of the train also hosts four local MCs, which can but must not synchronize to the ETB. If synchronizing, their role is passive, i.e. they do not contribute to the synchronization. All four GMCs are disseminating time on both redundant ETB lines.

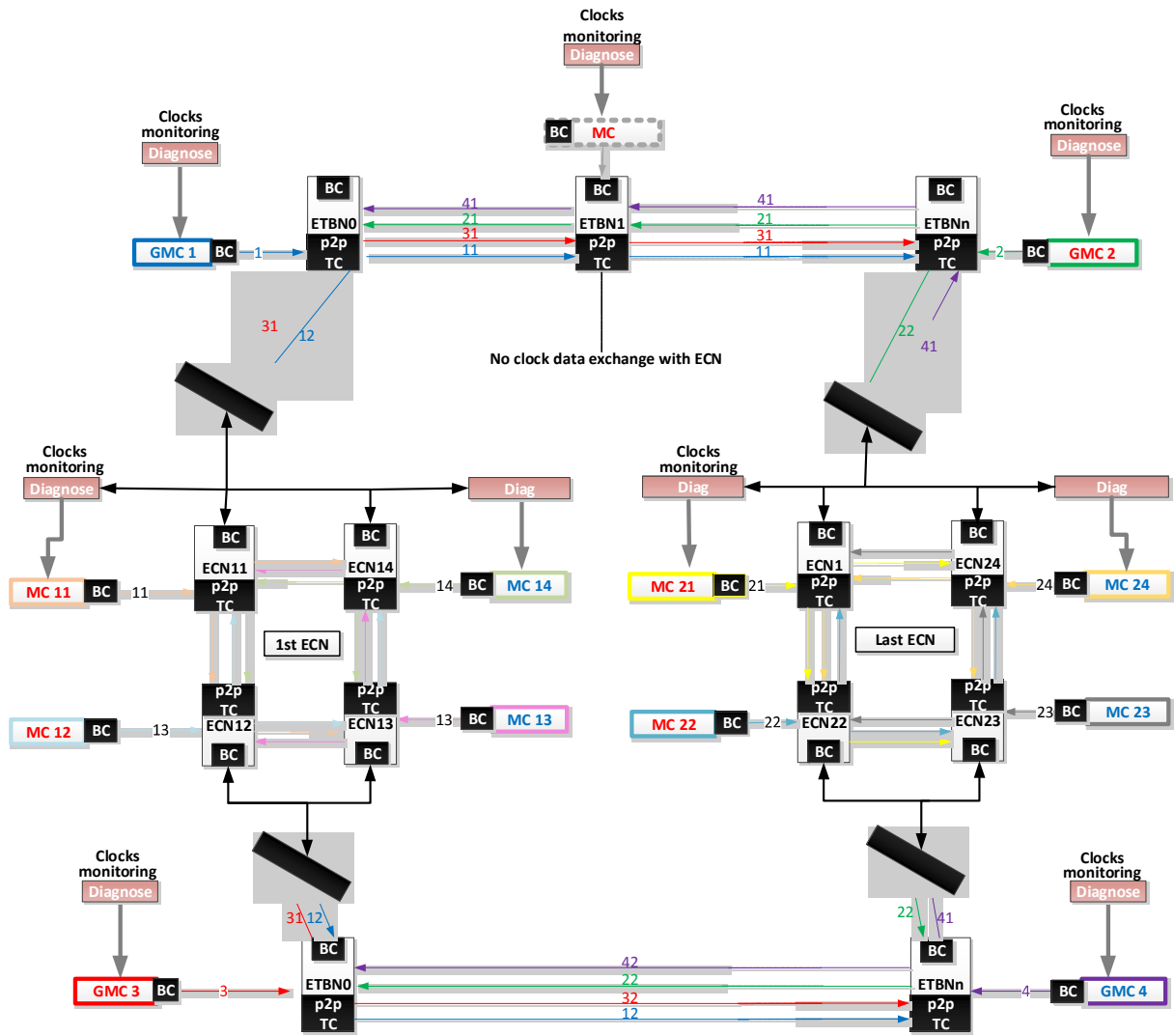


Figure 20 Option one: ETB and ECNs exclusively asynchronous

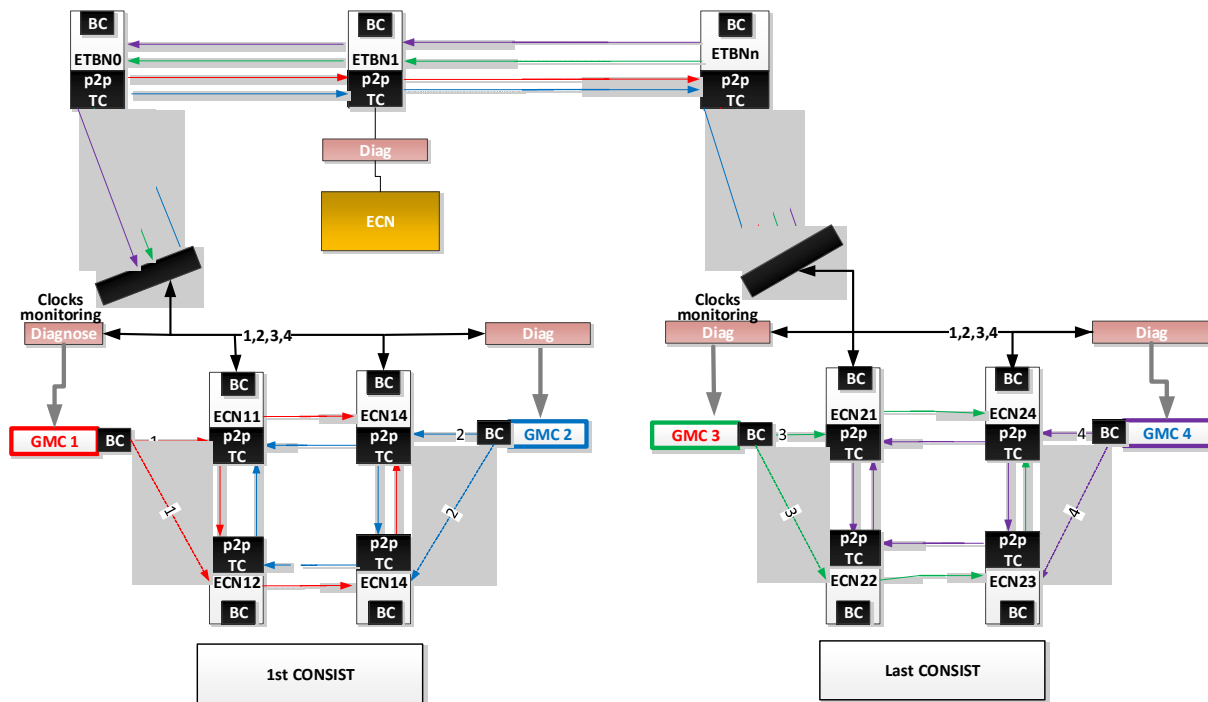


Figure 21 Option two: ECNs synchronous to ETB

4.1.4 GPS and NTP time as basis

If available on power-up, either an NTP or GPS clock shall be taken as a reference time. The 1st consist and ETB network shall operate on the same time, meaning they need to synchronize to the external clock. The master clock shall be accurate enough so that it can hold a train system time at less than 250ms difference to the ETCS and railway system signalling time until the next power-up.

Having precision clocks associated with ETBN switches could make an ETB switch more expensive and might even not be possible due to the physical position of a train switch. An MCG with GPS option could be provided for each consist and could host the synchronization point.

Nevertheless, the master clock needs a low drift ($\leq 1\text{-}5\text{ppm}$, TCXO class) to allow for synchronous scheduling when operating without GPS signals or NTP (e.g. in tunnels).

4.2 Flow Control Concept

On a network, it is meant for different traffic categories to co-exist. They can have different priorities, and/or different requirements regarding bandwidth and end-to-end latency. This is made possible through scheduling and traffic shaping, i.e. through flow control. IEEE 802.1Q [14] enables eight priorities to distinguish the importance of the packets, visible in the 802.1Q VLAN tag of a standard Ethernet frame, but no definitive assurance can be warranted related to latency questions. These non-determinism issues can't be avoided with standard Ethernet switches. Even the highest priority frame will not have precedence over a frame whose transmission has already started, and it will wait in the switch buffer for the ongoing process to complete.

4.2.1 Enhancements for Scheduled Traffic IEEE 802.1Qbv [23]

In contribution to the solution of issues related to bound latency in Safe4RAIL, TSN's time-aware scheduler **IEEE 802.1Qbv** [23] will be used, with a goal to reach the lowest possible delays. Its approach is to line up the network communication into cyclical time slots with fixed length. Within these slots, different time allotments can be assigned by configuration to one or more Ethernet priorities for a limited-time exclusive use. The transmission medium is completely reserved for assigned traffic classes, the transmission conditions are guaranteed, and the transmission can't be interrupted. These virtual time-constrained communication channels isolate time-critical communication from non-critical, and eliminate non-deterministic interruptions, such as buffering in the Ethernet switches.

To achieve this, to each traffic class queue (TC) of a switch port a *transmission gate* is assigned, as shown in Figure 22. It is enabling the control over the frame's transmission - its state signalizes if the queued frames of the port can be sent out or not:

- State '*Open*' allows the transmission selection algorithm of the queue to select the frames within that queue for transmission;
- State '*Closed*' prevents transmission of the frames within the queue.

The *transmission selection algorithm* determines which traffic class (out of those with frames available for transmission) will be providing the next frame to be sent.

A sequence of transmission gate states makes for a *gate operation*. The gate operation changes the gate transmission state for gates of each port's queue.

Every port has a list of gate operations, i.e. a *gate control list*. Without scheduled traffic, by default, all gates are permanently open.

Gate operations from the gate control list are managed by three standard defined state machines. Together they:

- initiate the execution of the gate control list
- ensure that the gating cycle time defined for the port is maintained
- execute the gate operations in the gate control list, in sequence
- establish the appropriate time delay between each operation
- manage the process of updating the current active schedule
- interrupt the operation of the other state machines while the update process is performed and re-start them once the new schedule has been installed.

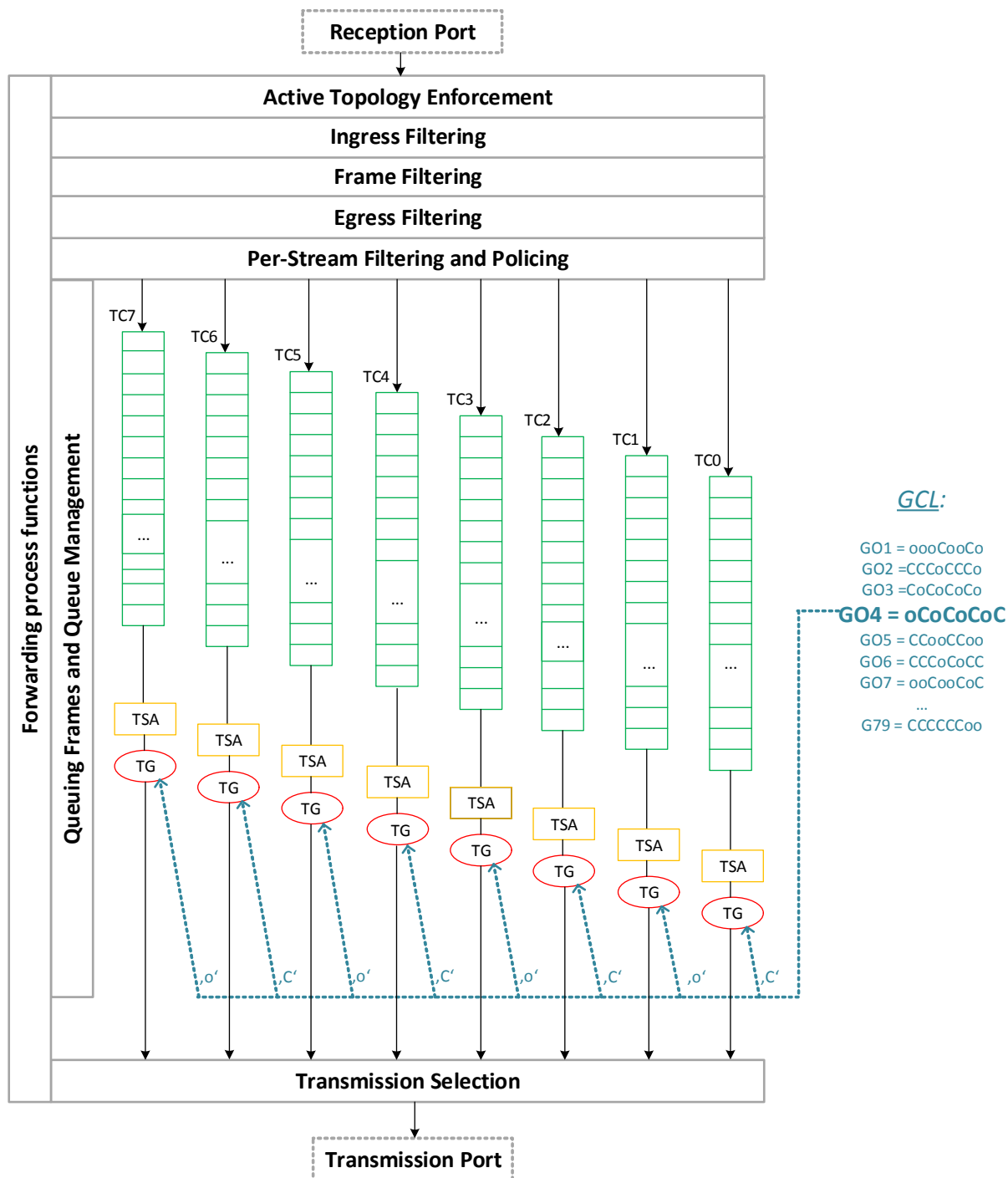


Figure 22 Scheduling, as defined by IEEE 802.1Qbv, within frame forwarding process

4.2.2 Per-Stream Filtering and Policing IEEE 802.1Qci [24]

To further enhance IEEE 802.1Q [14] to enable a guaranteed delivery for the traffic, the **IEEE 802.1Qci** [24] standard is used. It specifies procedures to perform frame counting, filtering, policing and service class selection for a frame, based on a data stream (to which the frame in question belongs) and synchronized cyclic time schedule. Policing and filtering operations perform the detection and mitigation of disturbing transmissions by other systems in the network by limiting the reception rate, and therefore improve the robustness. For example, devices could exceed the dedicated bandwidth per stream, which can affect all the other streams, not just the offending one.

Support of PSFP requires implementation of the Stream identification function specified in Clause 6 of IEEE Std. 802.1CB, as the parameter *stream_handle* provided by this function is used in the policing and queuing decisions taken by PSFP.

PSFP is made possible with the following mechanisms, integrated into the frame forwarding process, as shown in Figure 23:

1. Stream filters,
2. Stream gates, and
3. Flow meters.

The *stream filters* determine the filtering and policing actions that are to be applied to frames received on a specific stream. Each stream filter contains the following elements:

- a. A stream filter identifier, to uniquely identify the filter instance,
- b. A *stream_handle* specification (a single [25] or wild-card value).
- c. A priority specification (a single [25] or wild-card value).
- d. A stream gate identifier, for the stream gate used by the stream filter.
- e. Zero or more filter specifications. The actions specified in a filter specification can result in a frame passing or failing the specified filter. Frames that fail a filter are discarded. The following filter specifications are currently defined:
 - Maximum SDU size. Frames that exceed this SDU size do not pass the stream filter; frames that do not exceed this SDU size can pass the stream filter if all other filter conditions are met.
 - Flow meter identifier of a flow metering function. Flow metering is always applied after any other filter specifications that could result in frame discard.
- f. Frame Counters (frames matching both the *stream_handle* and priority specifications, frames that passed/did not pass the stream gate, frames that passed/did not pass the Maximum SDU size filter, frames that were discarded due to the flow meter operation).
- g. A *StreamBlockedDueToOversizeFrameEnable* parameter, to operate the corresponding function.
- h. A *StreamBlockedDueToOversizeFrame* parameter (for reference, see Chapter 8.6.5.1.1 of [24]).

The *stream gates* are allowing or preventing frames from passing to the flow meters and further to queuing algorithms, by alternating their states, opening and closing. Each stream gate is determined by its parameters:

- a. A Stream gate identifier.
- b. Stream gate state (open or closed).
- c. Internal priority value specification, IPV, which determines the traffic class associated with a frame.
- d. *GateClosedDueToInvalidRxEnable* parameter, to operate the corresponding function.
- e. *GateClosedDueToInvalidRx* parameter (for reference, see Chapter 8.6.5.1.2 of [24]).
- f. *GateClosedDueToOctetsExceededEnable* parameter, to operate the corresponding function.
- g. *GateClosedDueToOctetsExceeded* parameter (for reference, see Chapter 8.6.5.1.2 of [24])
- h. Optionally, a stream gate control list.

The *flow meters* instance table contains a set of parameters for each flow meter instance. The parameters for each flow meter are as specified in Bandwidth Profile Parameters and the Algorithm of [26] plus some additionally:

- a. Flow meter identifier, identifying the flow meter instance.
- b. Committed information rate (CIR), in bits per second.
- c. Committed burst size (CBS), in octets.
- d. Excess Information Rate (EIR), in bits per second.

- e. Excess burst size (EBS) per bandwidth profile flow, in octets.
- f. Coupling flag (CF), which takes the value 0 or 1.
- g. Color mode (CM), which takes the value color-blind or color-aware.
- h. *DropOnYellow*, which takes the value TRUE or FALSE. A value of TRUE indicates that yellow frames are dropped (i.e., discarded); a value of FALSE indicates that yellow frames will have the drop_eligible parameter set to TRUE.
- i. *MarkAllFramesRedEnable*, to operate the corresponding function.
- j. *MarkAllFramesRed* (for reference, see Chapter 8.6.5.1.3 of [24]).

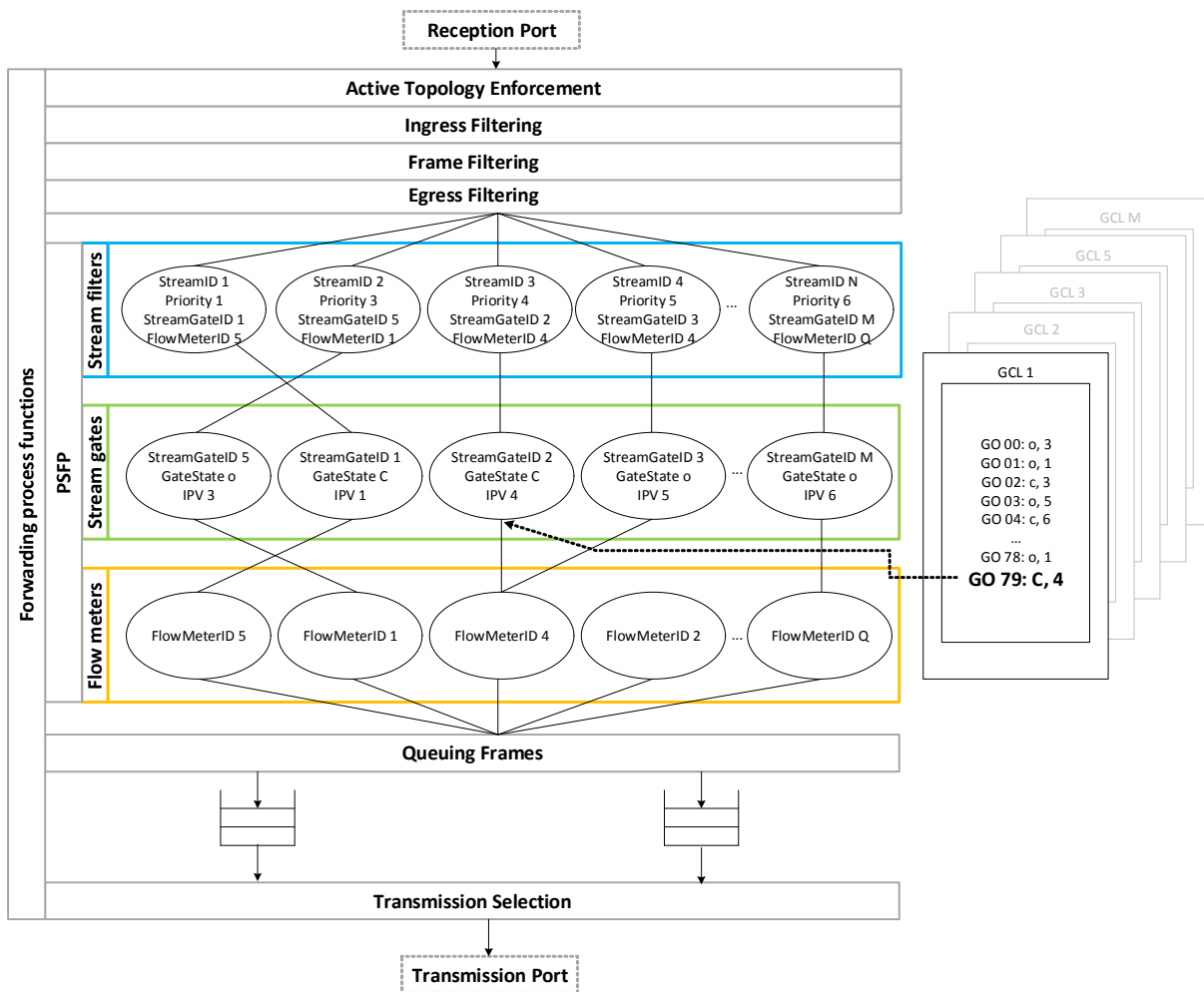


Figure 23 Per-Stream Filtering and Policing, within frame forwarding process functions of a switch

4.2.3 Achieving Traffic Requirements with Traffic Shaping Mechanisms

The use of different queues provides isolation between traffic with different timing requirements. An overview is offered in Table 1. Control traffic has the strictest traffic requirements, and demands the highest priority among the traffic classes, therefore must be mapped to the highest priority queues. In order to control the exact time at which messages are relayed, the time-aware shaper (Option B) is used for Control traffic so that the latency is guaranteed, and the jitter is kept as low as possible.

Streaming traffic can have different timing requirements, varying from hard real-time to non-real-time, with a queuing allocation varying from high to low priority respectively. High priority streaming traffic can use the time-aware shaper (Option B), whenever a guaranteed latency needs to be provided.

Low-priority streaming traffic and best-effort traffic can be mapped to the remaining priority queues, without the use of traffic shaper.

Traffic class	Features	Characterization	TSN traffic shaping policy
Control	Latency: guaranteed, low Jitter: guaranteed, low	<ul style="list-style-type: none"> - Period - Deadline - Data length - Source - Destinations 	Time-aware shaper (option B)
Streaming	Latency: guaranteed Jitter: guaranteed, low	<ul style="list-style-type: none"> - Minimum interarrival time (=period) - Deadline (=period) - Data length - Source - Destinations 	Time-aware shaper (option B)
	No guarantee		None (option A)
Best-effort	No guarantee	-	None (option A)

Table 1: Traffic classes and traffic shaping mapping

From the set of up to 8 queues, the number of queues used by every type of traffic is assigned with the following criteria:

- Traffic using TAS will use a timing gate scheduler that uses as few queues as possible while still satisfying the traffic requirements.
- Best-effort traffic will use the remaining queues.

The TAS mechanisms provide the opportunity to meet the latency requirements for critical flows. The corresponding configuration for all devices along a time-aware path must be calculated before it can be used and then loaded into the devices. For simple applications, such a schedule can be computed by hand. However, the computational complexity for more complex applications scales rapidly and is considered an NP-hard problem. This problem has sparked a research field on its own, searching an optimal solution to match all scheduling constraints in an efficient amount of time. The scheduler makes use of the parameters offered by the different TSN mechanisms, e.g. priority queues, and defines the schedule for the GCLs for each port in each switch in the network.

Note that the TAS scheduler provides a schedule fulfilling the requirements of the traffic, including the deadline requirement. This means that the message delay is constrained to be no larger than the relative deadline.

4.3 TSN Mapping (ETB – ECN)

While the topology of an ECN network is statically defined and all Eds and their network traffic requirements can and will be defined on commissioning, the number of ETBNs on the ETB network will change as soon as consists separate or couple. ETB inauguration will handle this event by setting up the ETBN's IP address and providing the NAT/R-NAT routing for IP based

traffic from/to the connected consist networks. TSN (IEEE 802.1Q..) is below IP and cannot be routed the same way because of its timing and different addressing properties.

- The ETB and each ECN will be in a different time domain (see 4.1) unless the synchronization of consists to the leading consist and ETB is introduced.
- TSN needs predefined paths/links – ECN addresses may vary

Figure 24 shows an example of synchronous traffic for the maximum number of ECNs between the end consists of a train. For synchronous operation, time slots are allocated and the ETBNs are statically configured to distribute and use these time slots for the predefined amount of time. Data is transmitted at three different intervals at fixed times in the ETB time domain. Each interval hosts 32 separate streams with datasets to read and write – this way the NG-TCN can guarantee reserved bandwidth and minimum latency for the longest train.

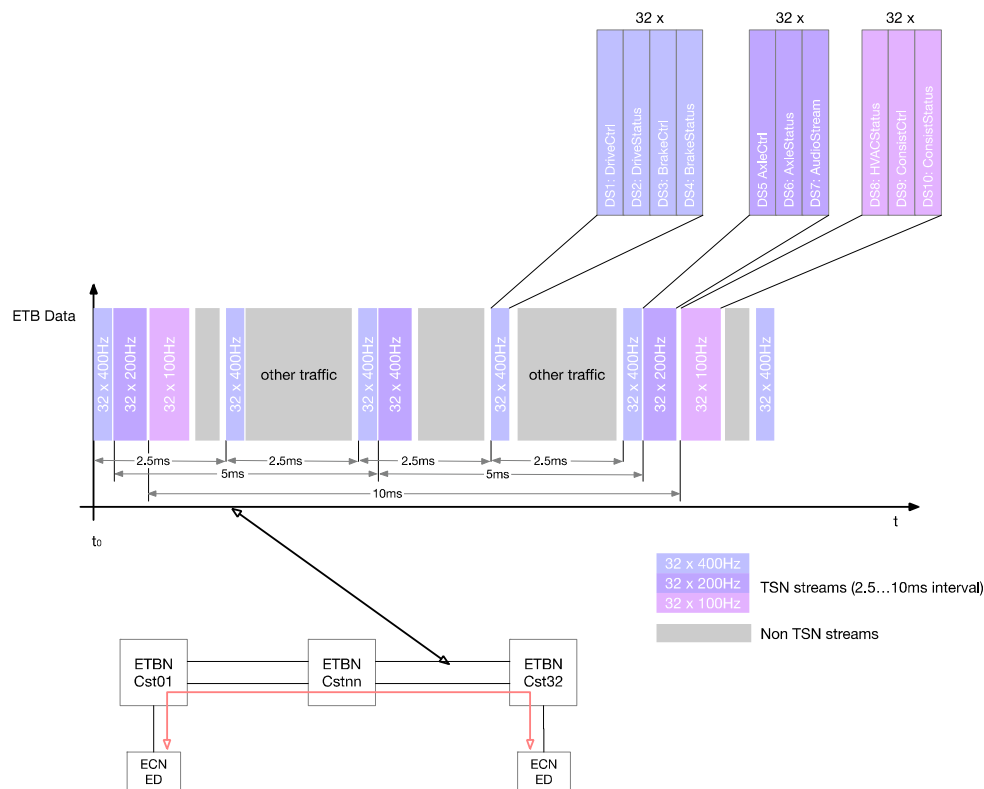


Figure 24: TSN Streams over ETB

TSN traffic on the ETB must be common for all trains and consists, which might be coupled. These common communication schemes and requirements are to be defined in the CONNECTA project, CTA WP4 – Functional Open Coupling/Application Profiles. All telegrams defined in FOC shall be foreseen in the ETBNs and the ETB/ECN Gateway as well as in the bandwidth calculation for the ETB.

TSN communication between arbitrary devices in different ECNs is permitted, but possible in special OEM-proprietary configuration

4.3.1 TSN Gateway

To support scheduled deterministic communication between consists of different classes and vendors, a common definition on ETB traffic (frame format, periodicity, number of frames in

each periodicity class) must be provided. This commonly defined scheduled traffic can be mapped to the local ECN via a deterministic ECN-ETB gateway.

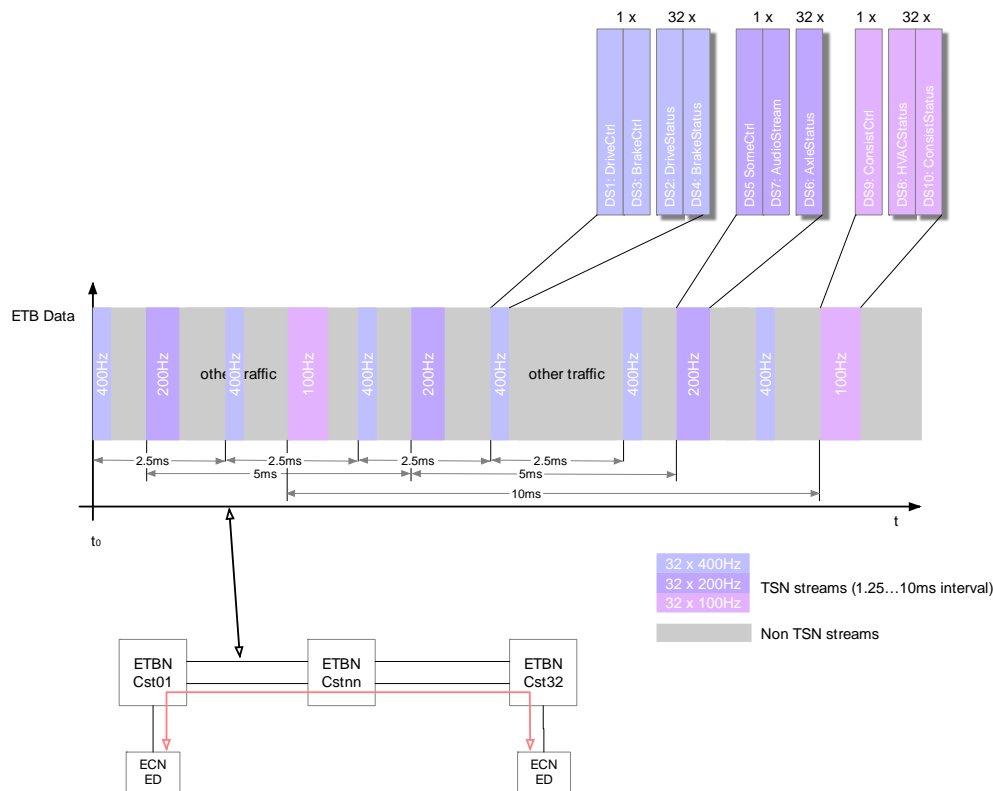


Figure 25: TSN traffic on ETB

On trains consisting of less than 32 consists this will waste a lot of bandwidth on the ETB, because for interoperability the worst case has to be accounted for – even with 2 consists the ETB must be laid out for 32! Figure 25 shows a reduced scheme, where only the leading consist writes to a single control stream.

The leading consist:

- transmits all control data to all other consists
- receives status data of all other consists
- provides common audio streams (PA) to any consist

Requirements needed from Functional Open Coupling:

- data definitions for consist egress and ingress flows
- scheduled interval times / frequency
- scheduled data size

An NG-TCMS ready ETBN must provide in parallel to the IP-NAT router, a deterministic gateway, which maps the predefined scheduled ETB traffic to the corresponding ECN.

Figure 26 shows the needed data flow in such a gateway for the leading consist, Figure 27 shows the data handling for a non-leading (led) consist:

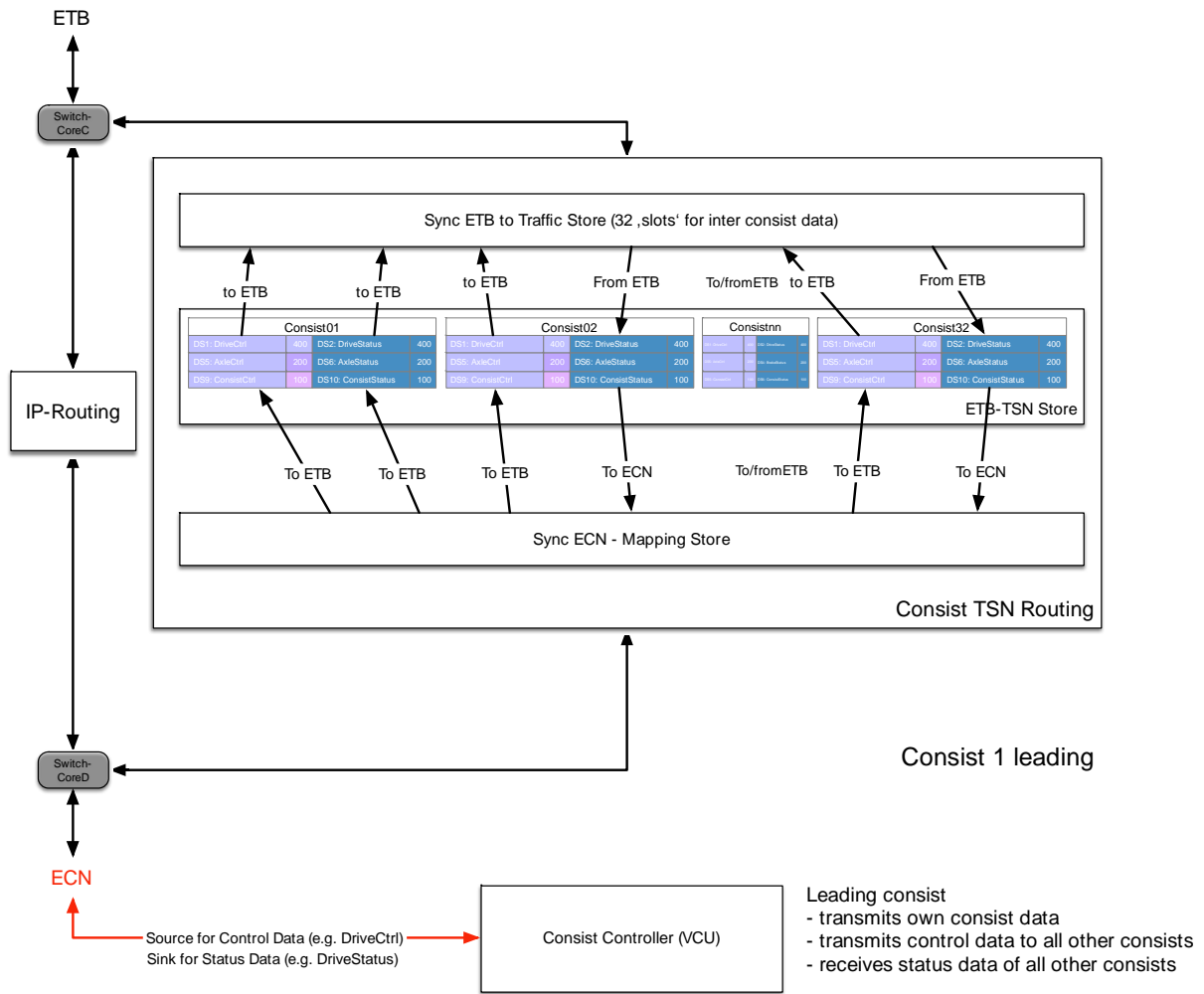


Figure 26: Dataflow Mapping for ETBN in Leading Consist

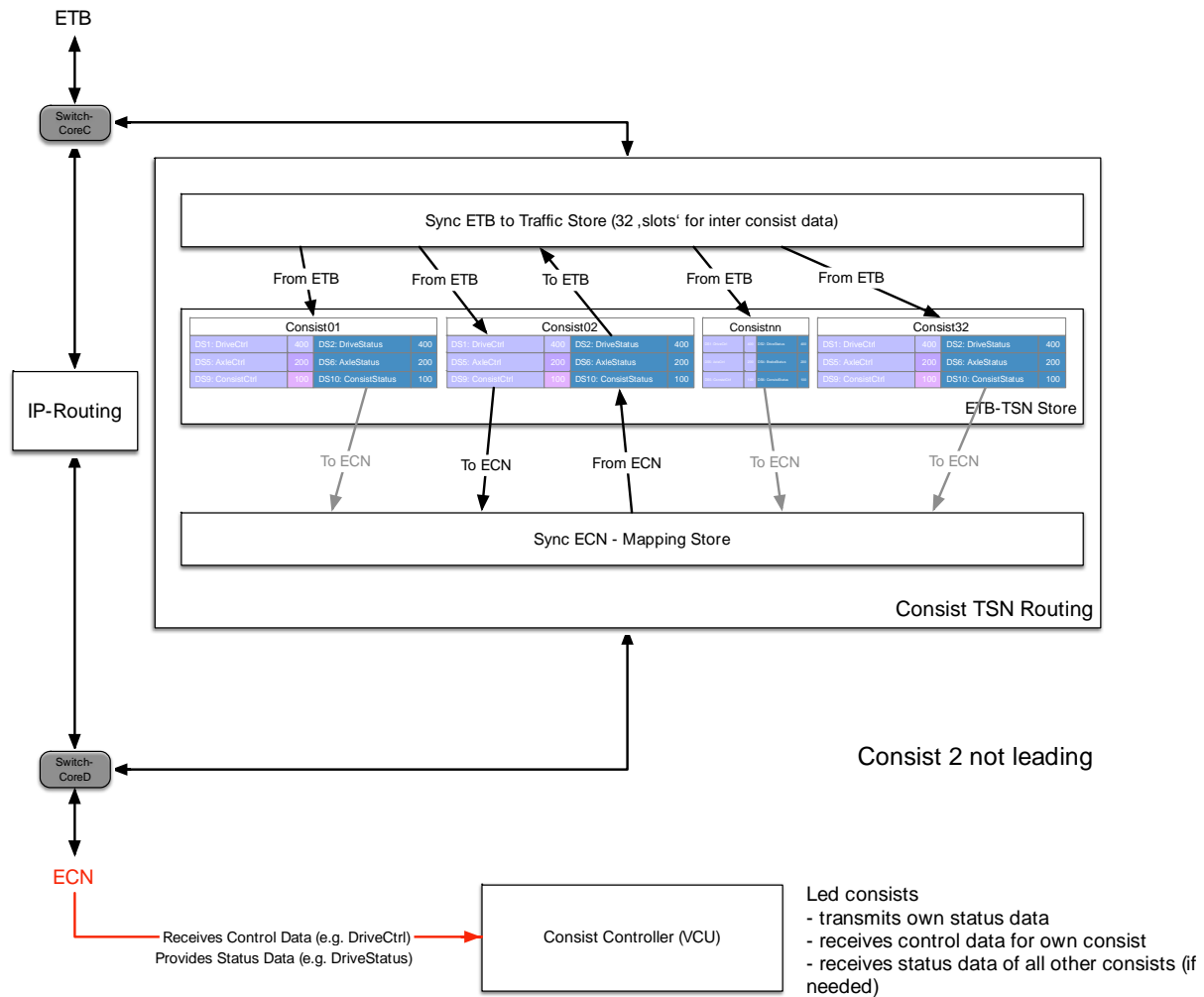


Figure 27: TSN Mapping for ETBN in Non-leading Consist

4.3.2 Configuration needs

The TSN mapping adds to the configuration of an ETBN:

- Static (standardized, from FOC):
 - ETB side properties of inter-consist data
- Dynamic (standardized):
 - Train Consist No.
 - Operational Train Consist No.

Because the latter information is available from the TTDB (i.e. ECSP), the synchronization and synchronous TSN traffic over ETB is available only after an operational train inauguration and synchronization initialization.

4.4 Redundancy Management

When discussing mission-critical networks, fault tolerance or increased availability is an imperative, which necessitates the use of redundant structures. Safety-relevant communication must be protected from hardware and media faults, especially avoiding single

points of failure, i.e. disabling the communications network by a single technical fault. This can be prevented by redundant structures - additional media connections enable the system to switch over to a secondary network path on a failure in the primary path. That way the communication in the network remains undisturbed and a reparation can be carried out to restore the failing elements and continue the operation in a fault-free state.

Also, introducing redundant connections makes load balancing possible. The effective bandwidth of the original connection increases, since the data load within a specific time over a redundant path is greater than the data load that a single cable can handle.

One difficulty for the implementation of redundancy in any Ethernet network is an Ethernet-native requirement for loop prohibition. Between a transmitting and a receiving point in an Ethernet network, only one active path at time is allowed. Having more than one active paths between two devices will create loops, which enable an endless circulation of messages within them and, consequently, a network overload. Since alternative paths in a network are a postulate for redundancy, a protocol is needed to ensure that, at any time, only single **logical** path to each device (from application point of view) is active. This does not limit the number of **physical** paths, though, if they are kept in a standby mode. To ensure the desired behavior, the links need to be monitored, communication interruptions detected, and alternative paths activated as soon as a failure is noticed. The switching to alternative paths will pause a traffic for a certain period, which can sometimes be hard to predict.

A protocol for redundancy management used in Safe4RAIL needs to satisfy three elemental concerns:

1. To ensure compatibility, the method it relies on needs to be standardized;
2. Constraints on the network installation (such as topology limitations or maximum number of switching devices) need to be settled;
3. The switchover-time from an active, failing logical path to a redundant, alternative one, during which the network communication is out of operation, needs to be deterministic.

The requirement of determinism is a prerequisite for any time-critical or time sensitive application. Only reliable, measurable values can provide the worst-case network switchover time, critical to fulfill the time limitations of the application that is using the network. Only if the media redundancy protocol can switchover within time limits which enable the application to continue operating without impairment, then its redundancy mechanism is fulfilling the timing requirements of the said application.

The redundancy concept within Safe4RAIL is based on top of the Frame Replication and Elimination for Reliability (FRER) protocol as defined in IEEE 802.1CB [25].

When the duration of the switchover after a failure of the equipment in the network is intolerable, in order to prevent packet loss due to communication interruption, a seamless redundancy is needed. To achieve it, FRER replicates the Ethernet frames at the transmission source and forwards all the copies through the network using multiple active paths. Further, these messages are rejoined at one or more other points in the network, where the replicates are eliminated, and reconstituted unique originals are delivered from those points on. To accommodate existing applications and to support interoperability with similar standards, FRER defines several strategies for identifying replicated packets and distinguishing them from other traffic, all in a seamless manner for the applications.

The creation of multiple paths for transmission of duplicates is out of its scope.

IEEE P802.1CB [25] mechanisms offer the advantage of use in any topology. Additionally, IEEE P802.1CB is not restricted to two redundant paths, but numerous redundant transmission paths can be used. However, for the purposes of time-critical or time sensitive applications, as the case in Safe4RAIL is, it must be ensured that all redundant paths can support the required latency guarantees.

For FRER to function, a stream identification is required. Stream identification utilizes a single Service Access Point (SAP) to the layers below it (e.g. MAC and physical), and an array of SAPs to the layer above it, each SAP corresponding to a different Stream.

In Figure 28 a two-port switch is illustrated, with FRER functions in one of its ports.

The *Stream Transfer Function* acts as a two-port packet relay, existing inside the one port of the switch, and relaying packets belonging to Streams.

The *Non-Stream Transfer Function* (NSTF) has the same function but attaches to the “unknown” SAP and relays packets not recognized as belonging to Streams.

The *Lower Stream Identification* functions separate FRER and non-FRER packets; the non-FRER packets are then forwarded to the NSTF.

The *Upper Stream Identification* functions identify the Streams to which FRER packets belong and enable the other FRER functions to execute.

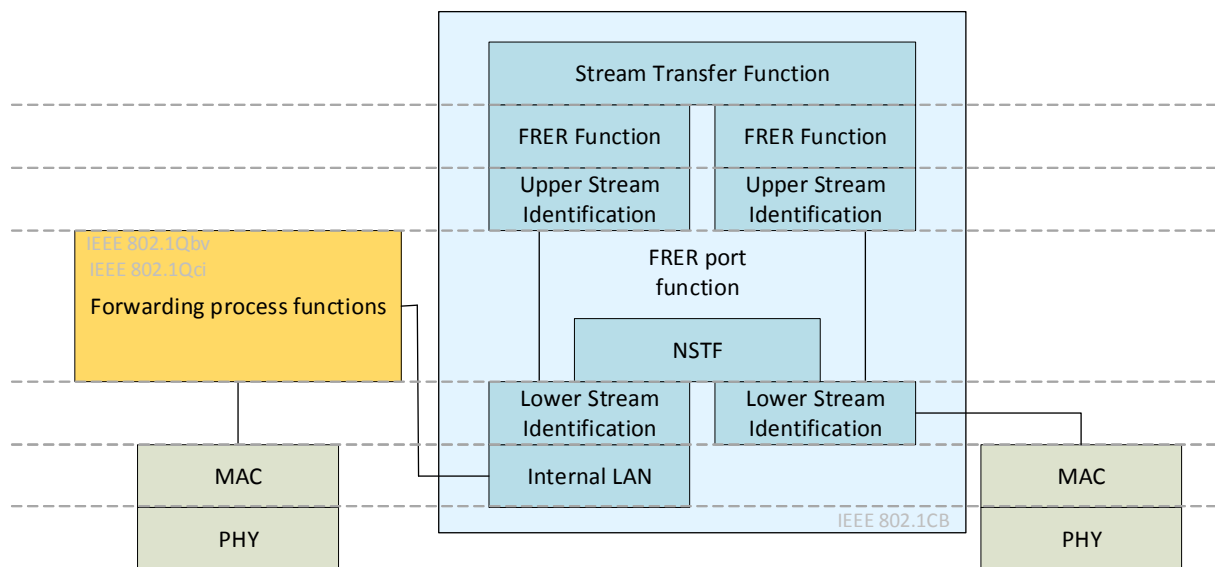


Figure 28 Stream functions in a switch, with marked peering levels of FRER sublayers

Four Stream Identification functions are defined:

1. Null Stream identification (see 6.4 of [25]),
2. Source MAC and VLAN Stream identification (see 6.5 of [25]),
3. Active Destination MAC and VLAN Stream identification (see 6.6 of [25]),
4. IP Stream identification (see 6.7 of [25]).

Chapter 5 Data Transmission Concept

Ethernet networks allow usage of concurrent protocols over the same physical connections. The DbD concept ensures independent bandwidth, latency and data delivery priorities by reserving network resources via network switch configuration and the usage of VLANs and TSN for TCMS and non-TCMS traffic.

Data transmission between end devices connected to the NG-TCN can or must support different communication schemes, according to their function:

- a) Hard real-time TCMS control communication → TRDP v2 (TSN-PDU)
- b) Soft real-time and legacy TCMS management communication → TRDP v1
- c) High bandwidth, low latency video/audio communication → e.g. RTSP
- d) Other – network maintenance, SW updates, monitoring, etc. “Best effort”

While the FDF supports all these schemes, devices like CCTV or audio devices might support c) or d) only.

Additionally, devices with high availability requirements (not necessarily safety related) from the TCMS domain, should connect to the A-Plane and B-Plane through separate hardware (Ethernet PHYs). Redundancy is also supported by the TRDP protocol by means of a sequence counter in the TRDP header field.

5.1 End-to-End Concept w. TRDP / TSN-PD

5.1.1 Network Protocol Layers

Figure 29 shows the principle protocol and functional layers of a sample end device using the DbD platform.

On top, a TCMS application (Level **A**) reads and writes to consist- or train-wide pre-defined variables or datasets (**B**). These values are transferred locally in case of mapped local I/O ports and also transmitted or read as process data using TRDP (**C**).

If variables are configured with real hard time requirements (TSN), the application’s task has to update these values according to the configured time schedule of the corresponding TSN dataset. Before that data is sent out on the wire, it must be marshalled and probably secured by SDTv4 into a VDP payload (**D**). SDTv2 will be used for ECSP/SIL2-related traffic.

The marshalled, packed and secured/validated dataset will be framed as TRDP TSN-PD (**E**) and transferred via a raw socket interface to the related NIC. In the NIC the payload will be embedded into an UDP/IP frame and timely put on the wire (**F**). All processing from levels B to F add to the negative time offset the application task needs to account for.

Also, the NIC is responsible to block or stop non-TSN traffic if its transfer would collide with scheduled TSN traffic (e.g. with ‘Best Effort’).

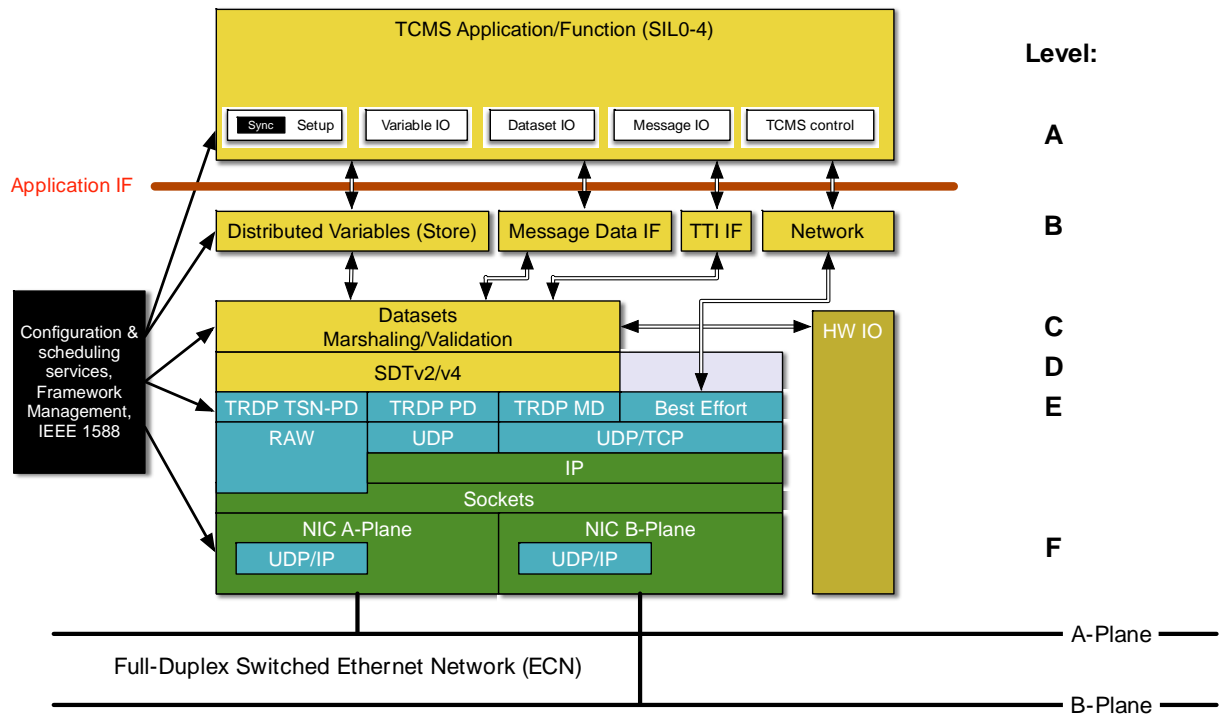


Figure 29: Protocol Layers

This sample provides two Ethernet interfaces to be connected to separate switches in the consist network. For safe end devices (ED-S) using scheduled TSN traffic, these ports provide the same packets to the A-Plane and the B-Plane using a pre-defined VLAN and destination MAC address reserved for TSN.

Legacy TRDP traffic may also use both interfaces for redundancy but may use only one interface; if 'best effort' is used, one interface will suffice as well.

Figure 30 shows the physical connections and basic network layout of a consist. The ring is logically separated into two planes and open at one arbitrary point (focal point) to prevent circulating frames. Devices (ED, ED-S) sharing the same set of variables or datasets, exchange and align this data through TRDP-TSN streams (source → multiple sinks). ED-S 1 as data source shares a set of variable values with ED-S 3 on the same switch and ED-S 7 and 8 on another switch in the ECN. This data is sent using a network-wide defined time schedule as TRDP-TSN Process Data to all interested (subscribed) devices. Packets on each plane will be identical; end devices will discard duplicates¹.

¹ The duplicate frame could also be used to enhance communication error detection but would then lower the reliability by misusing the redundancy features for safety.

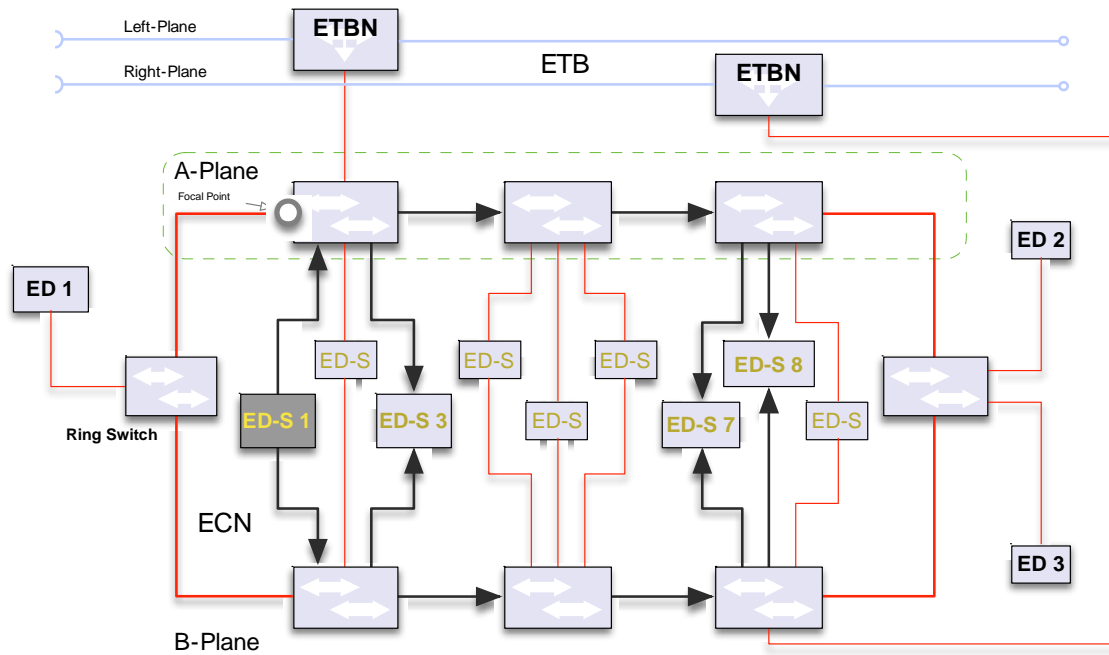


Figure 30: ECN Dual Plane Network with sample data flow

All network switches between source device and sink devices (and including these) must be configured for the used TSN stream(s), see Chapter 6.

A special case applies to consist-to-consist communication. TSN communication over the ETB requires passing at least two different time domains and is not (IP-) routable. Train-wide TSN communication is therefore restricted to predefined (standardized) streams on the ETB and a special ETBN extension, the ETB/ECN TSN gateway, will interface the different scheduled frames (see Chapter 4.2).

5.1.2 Communication Profiles

The standard TRDP protocol supports several communication patterns, which are used for TCMS maintenance traffic and non-TSN based communication:

- **PD Push** point to point
- **PD Push** point to multipoint
- **PD Pull** point to point
- **PD Pull** point to multipoint
- **MD Notification** point to point
- **MD Notification** point to multipoint
- **MD Request/Reply** point to point
- **MD Request/Reply** point to multipoint
- **MD Request/Reply/Confirm** point to point
- **MD Request/Reply/Confirm** point to multipoint

TSN communication in the NG-TCN uses the PD Push pattern only (Figure 31); by the way how TSN is defined, multicast addressing (in addition with VLAN tagging) is used to provide point-to-point or point-to-multipoint channels/links by switch and ED configuration.

Because of their event-based characteristics, the PD Pull pattern and all MD patterns cannot use TSN communication channels.

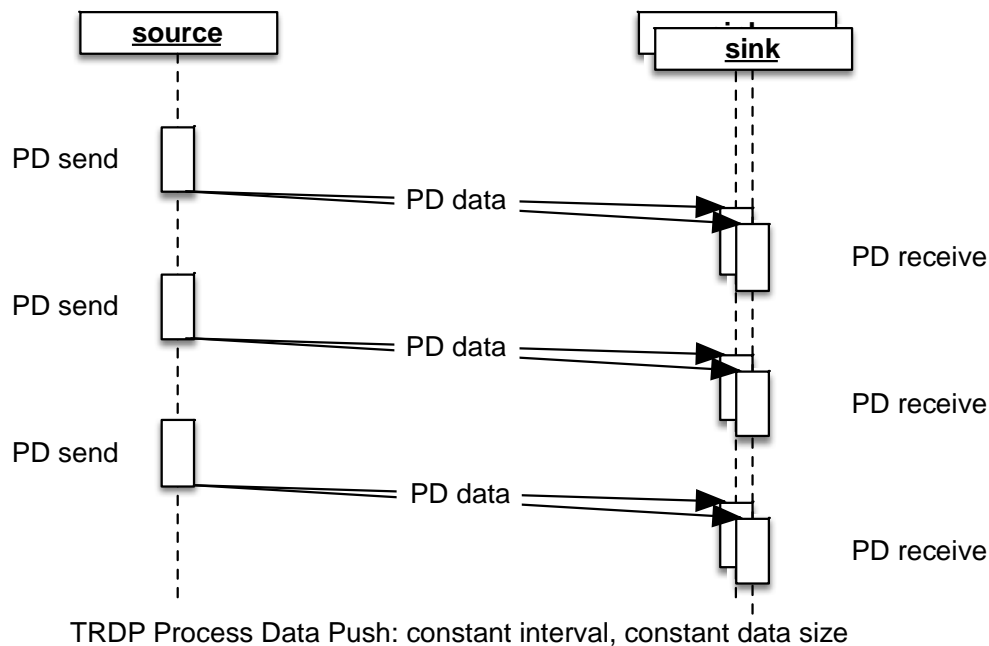


Figure 31: Process Data Communication Pattern – Push

TSN in synchronous mode demands an exact time when to put a frame on the wire (or switch). This schedule is taken from the configuration and maintained by a high precision timer (see 4.1). For a real implementation, the time used for changing a variable, preparing a dataset, validating and framing of the data must be considered as offset before sending the data. See 5.2 “Integration to FDF” for details.

5.1.3 Framing

TRDP process data telegrams use UDP/IP for transport and add an additional header (Table 2) to identify the payload, the message type and topology information to support train-wide communication. The message data header adds some more fields, like session identifiers and user URIs.

0		7	8		15	16		23	24		31
SequenceCounter											
ProtocolVersion						MsgType					
ComId											
etbTopoCnt											
opTrnTopoCnt											
DatasetLength											
Reserved											
ReplyComId											
ReplyIpAddress											
HeaderFCS											
... Dataset[0...1432Bytes] ...											

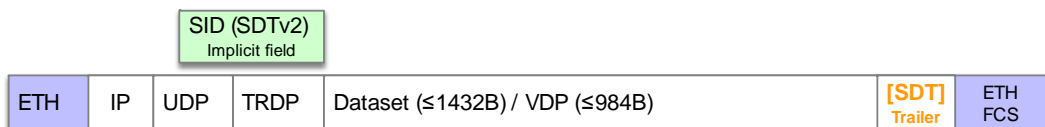
Table 2: TRDP PD Frame

For TSN, the necessary header fields can be reduced, because:

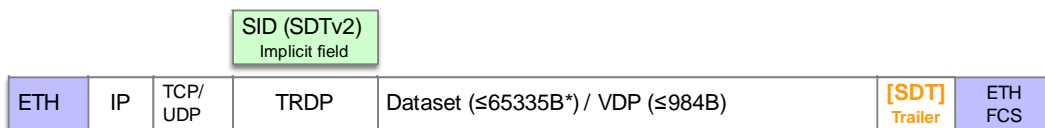
- a) there is only one message type (PD Push)
- b) addressing is fixed and pre-configured (TSN stream, VLAN ID)
- c) train-wide communication uses pre-configured ETB/ECN gateway (no topography counter).

Although TSN streams use Layer 2 addressing, and do not need IP or UDP, for the sake of easy monitoring and development, the IP and UDP headers are included.

TRDP PD



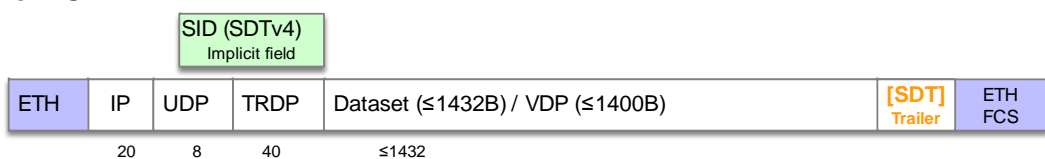
TRDP MD



* TRDP MD exceeding MTU-size are split into fragments (UDP) or stream chunks (TCP)

Figure 32: Legacy TRDP framing

TRDP PD over TSN



TRDPv2 PD over TSN

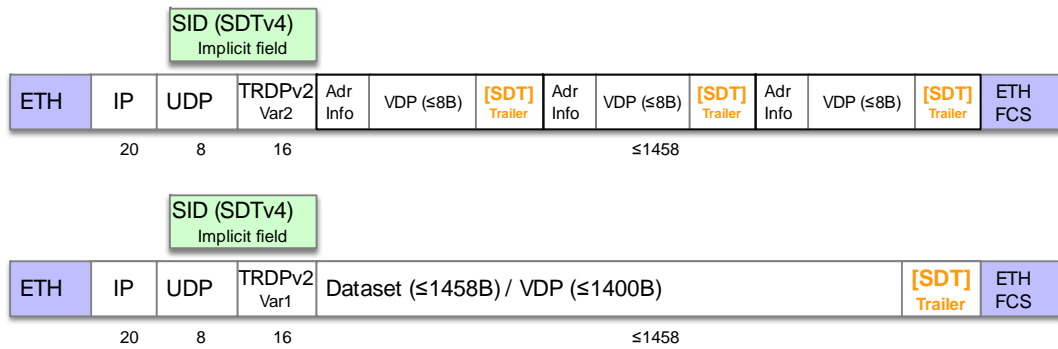


Figure 33: TRDP TSN-PDU framing

Discarding superfluous fields, the header size can be reduced from 40 to 16 octets. The TSN PD-PDU looks like shown in Table 3 (also labelled TRDPv2).

0	7	8	15	16	23	24	31
SequenceCounter							
VersionMsgType				DatasetLength			
ComId							
HeaderFCS							
... Dataset [0...1458Bytes] ...							

Table 3: TRDP TSN PD-PDU

The fields of TRDP TSN PD-PDU are:

- SequenceCounter: For non-safe payloads, to detect redundant frames, UINT32
- VersionMsgType: 0x0201 (non-safe), 0x0202 (Safe Data), 0x0203 (for multiple SDTv4 frame), 0x0204 reserved
- DatasetLength: Net size of payload, UINT16 [0...1458Bytes]
- ComId: Unique Identifier for payload, 0 for multiple SDTv4 frames
- HeaderFCS: CRC32

The major version number is at the same location as in the standard TRDP header – thus it will be recognized and discarded by all current standard TRDP stacks in case of erroneous reception. The header size would sum up to a total of 16 octets.

The TCNOpen TRDP stack implementation (as of V1.4) will discard the packet anyway, because the FCS would not compute correctly. The CRC check is done before enumerating the major version number.

The payload of a TRDP telegram is defined by a unique communication identifier, ComId. The structure of the payload data is defined by configuration using unique DatasetIds to allow a mapping between ComIds and DatasetIds.

To avoid the protocol overhead by transferring single variables in single frames over the network, variables should be grouped into datasets (as structures/arrays) before being

validated and protected by the safety layer (SDT). These DatasetIds must be mapped to Comlds. IEC 61375-2-3 reserves Comlds 1...1000 for internal use and also defines Comlds for TCMS telegrams. Payload definitions for the Application Profiles and Functional Open Coupling could add to these.

5.1.4 Addressing

TRDP-TSN frames are primarily addressed by the Ethernet Layer 2 properties “destination MAC” and “VLAN tag” (Figure 34):

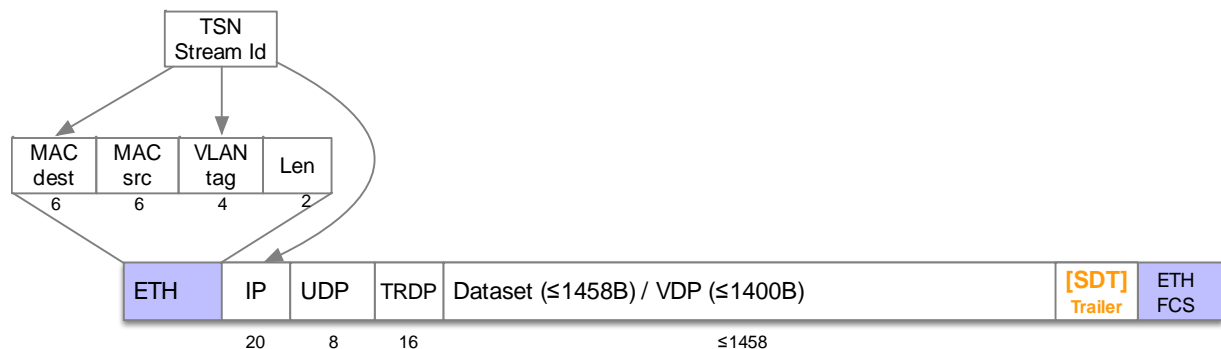


Figure 34: TSN Stream Identification

Depending on the used network layer (e.g. POSIX Socket interface), the IP-Multicast destination address is coded into a MAC multicast group and should thus also be used to determine and configure a TSN stream.

TSN addressing must be statically defined and, depending on the concrete instantiation (e.g. FDF) must be handled by the TRDP layer, the network (socket) layer and the NIC synchronously.

See Chapter 6 for an example of TSN stream addressing for the TRDP stack.

5.2 Integration to FDF

The current definition (IEC 61375-2-3, Annex A [15]) and implementation (TCNOpen) of TRDP does not demand tight timing for sending or receiving, which is necessary for real-time Ethernet.

Protocol processing in the TCNOpen TRDP implementation uses linked queues which are traversed periodically (~ 1 ms) and the lowest interval time is 10ms for process data.

To handle TRDP TSN-PD, the TCNOpen TRDP stack has to be extended to:

- allow scheduled transmission of TSN PD-PDU with high precision
- allow scheduled or event-driven reception of TSN PD-PDUs
- use TSN and non-TSN traffic concurrently (with TSN taking precedence over non-TSN).

Figure 35 depicts data transmission roughly from application to network. After processing some input values, the application updates ‘Var B’ and gives control to the FDF (in case of a single CPU). The FDF will prepare the data by creating the VDP (Vital Data Packet) payload and forward it to the network driver/card. Depending on the interface between the NIC will send out the frame immediately or time triggered.

In any case the sending task must provide the updated data in sync and with an offset (FDF-process time) before T_0 , the scheduled egress time configured for that TSN PD-PDU.

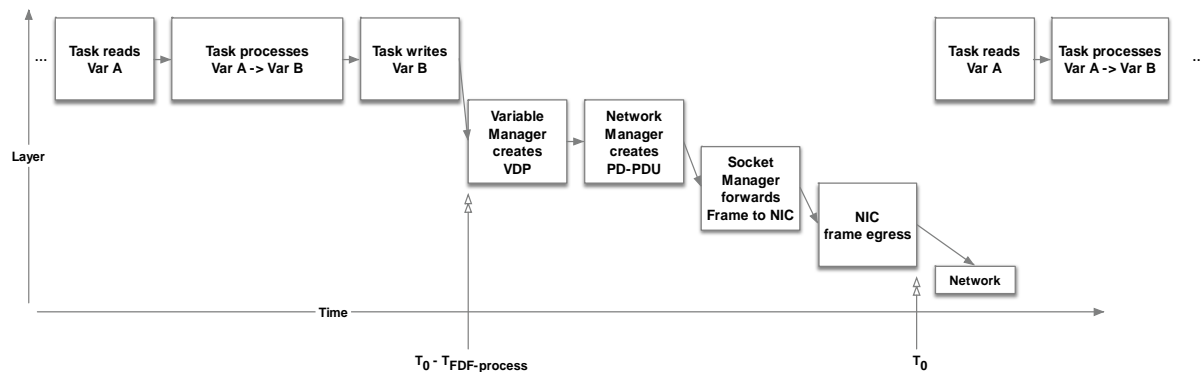


Figure 35: Dataflow of FDF/DbD

The latency generated by marshalling, validating and framing of arbitrary data by the communication layer (including sockets), requires that the NIC needs an exact trigger event to timely transmit the frame and match the switch's ingress schedule. It is the FDF's responsibility to execute the application's task in time and in sync and account for that offset.

For the current (v1.4) TCNOpen TRDP implementation to be used with TSN and the FDF, the following modules and functions must be changed or extended:

- a TSN Process Data publisher and subscriber must be defined
- receiver and sender must be prepared to receive and transmit v2 telegrams
- a direct-send function must be implemented
- the XML parsing utility should support additional TSN related tags and attributes
- depending on the underlying OS, the VOS (Virtual Operating System) layer must support direct access to the NIC

The values for QoS (and TTL) defined in the configuration are already used to set the corresponding fields of the network options, if the used Socket implementation allows it.

5.3 Safety Concept

To keep network infrastructure devices cheap and to allow the use of COTS, communication over the NG-TCN and most of the DbD infrastructure is considered unsafe (black channel approach).

Safe communication must be accomplished by securing transmitted data over an additional safety layer on top of e.g. TRDP. SDTv2, as defined in [23], has been used in the current TCMS but is limited to SIL2 functions. For SIL4, the residual error probability of any communication channel involved must be less than 10^{-11} .

CTA proposes and describes SDTv4 in detail [18], which extends SDTv2 safety measures by supporting two VDP variants and an additional 32-bit CRC for larger payloads (≤ 1432 octets).

SDTv4 must be implemented on each device involved in safety related functions (ED-S) and must be executed within a safe partition of the FDF – clearly separated from non-safe parts, e.g. communication stacks and network drivers.

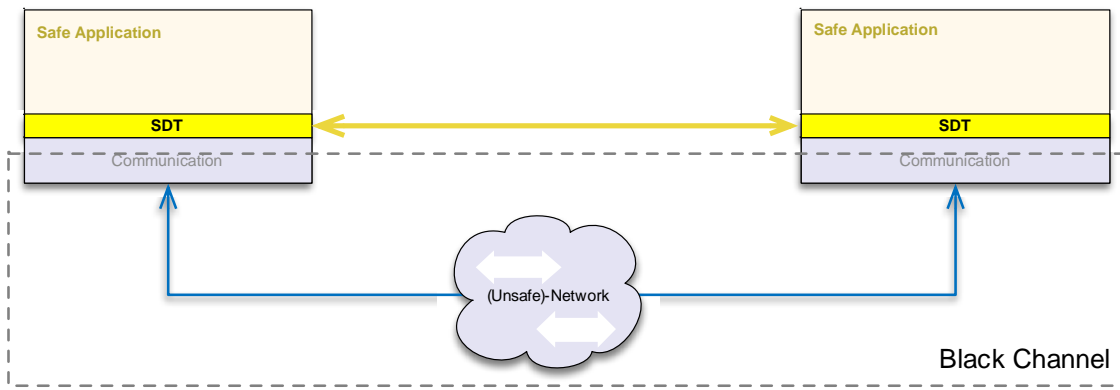


Figure 36: Safe Communication

Although the DbD concept provides black channel communication only, redundancy adds to RAMS and finally also to the safety requirements.

The SDT layer adds a trailer to each VDP and contains 16 octets (variant 1 only 12 octets, no CRC2) of validation data:

Reserved01, UserDataVersion (UDV), Safe-Sequence Counter (SSC), CRC1, CRC2

CRC1 is computed over the SID, the VDP, reserved01, UDV and SSC using the polynomial 1F4ACFB13 as defined in IEC 61784-3-3.

CRC2 is computed over the SID, the VDP, reserved01, UDV, SSC and CRC1 using the polynomial 1A833982B (CRC-32/5.1, as proposed by [33]).

CRC2 is used for variant 2, only. With variant 1, VDPs must be ≤ 8 Bytes.

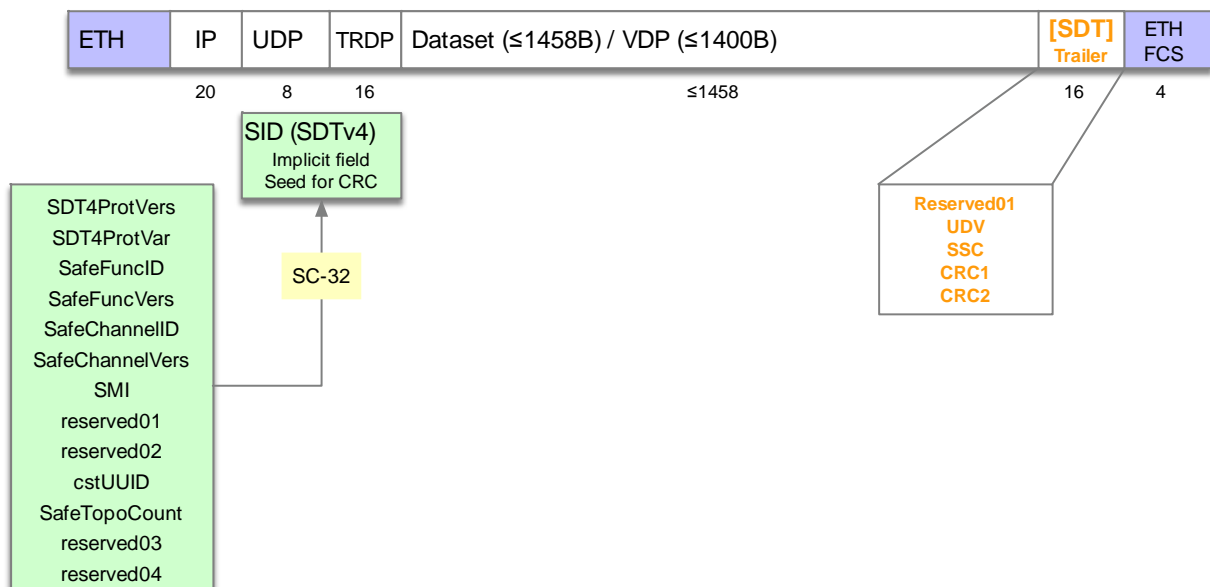


Figure 37: SDT computation Variant 2

The SID is an implicit field and its CRCs will be computed by sink and source prior to sending or receiving SDT data using both polynomials. The CRCs are then used as seed values for the transmitted VDP plus trailer.

The values required to compute the SID are partly preconfigured (e.g. from the SDT XML-tag) or must be retrieved from the TTDB, e.g. cstUUID and SafeTopoCount.

Chapter 6 Configuration

6.1 Configuration & Life Cycle

Common definition:

A configuration is an arrangement of functional units according to their nature, number, and main characteristics. It influences system function and performance, and often pertains to the choice of hardware, software, firmware, and documentation.

Our more specialised formulation:

Configuration is the set of definitions needed for a system to interact or communicate with other (sub-) systems.

6.1.1 Life cycle

The life cycle of a train typically spans over +30 years and over five phases, as shown in Figure 38, from first planning to end of life. After the first operational phase, trains will be refurbished and upgraded to new standards and operator's requirements and put into operation again (Homologation / incremental certification).

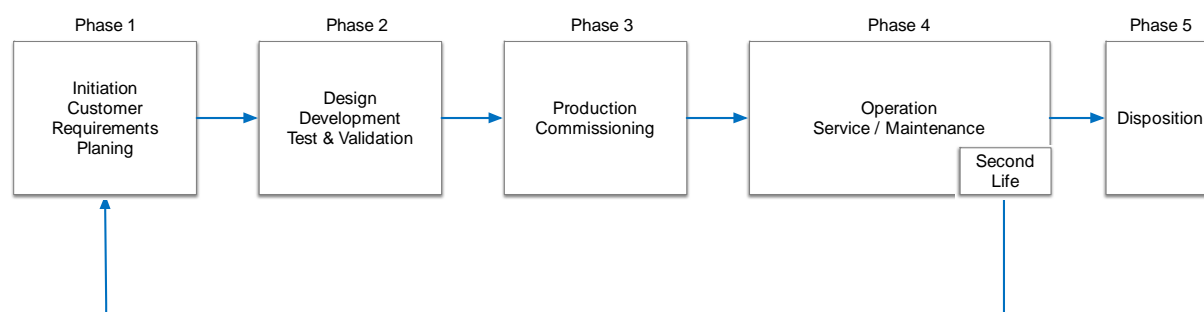


Figure 38: The life cycle of a train

Such a complete overhaul will surely lead to changes in safety critical functions and new configuration of the TCN. Depending on the new/changed requirements from phase 1, the scope of phases 2 and 3 will be less.

In phase 4, the updates of SIL- and non-SIL functions (IMP) must be possible without re-assessing the complete TCMS. In case of updating (or adding) non-SIL functions it has to be ensured that there will be no interference with already certified SIL-functions. The same needs to be proven if SIL-functions are updated.

The configuration of the TCN covers several aspects, depending on the current phase in the life cycle of a train and its project:

- Requirements & planning
- Development
- Lab: simulation & testing, validation & assessment
- Production: commissioning & testing
- Operation: dynamic changes (coupling, inauguration, redundancy / faults handling)
- Operation: regular tests (start-up, BIT, daily, weekly tests etc.)
- Service: commissioning (security) updates, functional upgrades

Dependencies and areas of configuration can be seen from a physical point of view as in Figure 39, where devices as they are attached to the TCN are seen as individual computing devices.

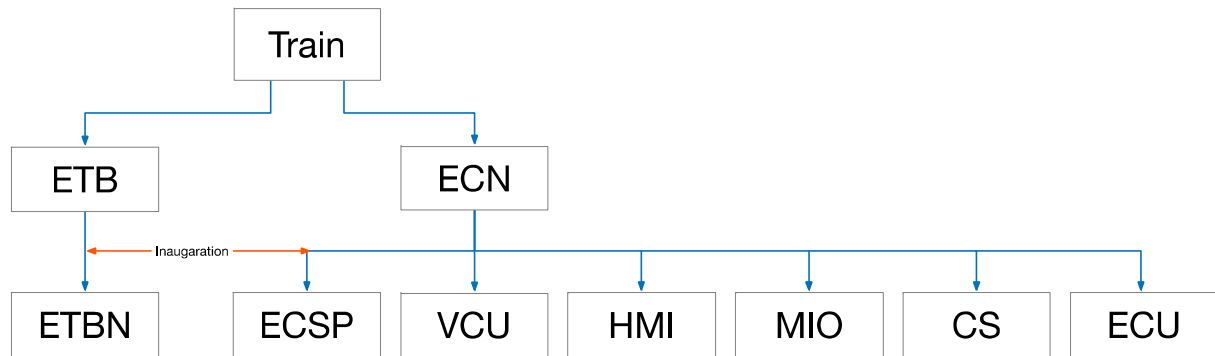


Figure 39: Device view

Incorporating the Functional Distribution Framework (FDF) and the underlying network and its protocols to the Integrated Modular Platform (IMP), configuration must cover the functional view, as a TCMS function will be hosted on different and/or several devices (refer also to Figure 44 and Figure 45).

Functional configuration contains two areas:

- Network infrastructure (switches, routers, couplers, access points)
- EDs (I/O, sensors and actors), Control Units (VCU)

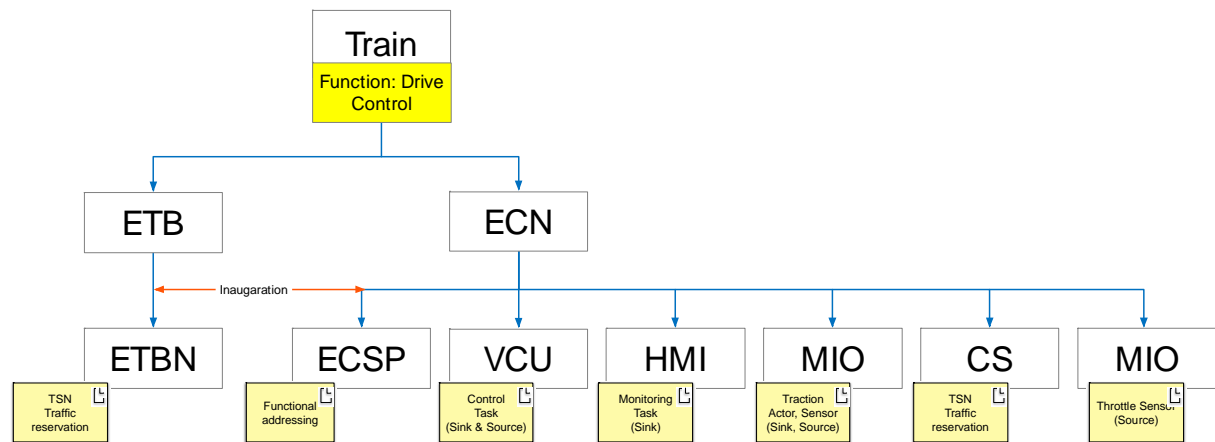


Figure 40: Functional view

Figure 40 shows a sample distribution for a traction control function. This function will be distributed and adds to the configuration for the:

- ETBN (TSN traffic reservation)
- Consist switch (TSN traffic reservation, switch port precedence and addressing)
- Consist Info in ECSP (addressing)
- MIO sensing the throttle lever (ComId)
- VCU control function (TSN, processing throttle reading, controlling traction safety function)
- MIO traction (actor for controlling engine, sensor for actual speed, ComId)

- HMI display and monitor for current speed (ComId)
- More: ETCS safety function, remote control, forwarding status & control to other consists
- ...

Complexity of a consist configuration is shown in the Table 4.

Label	Function	Location	Target	Volatile/Safety	Comment
Consist Info	Defines all TCMS functions/devices in the local consist	ETBN (ECSP)	ECSPs/TTDBs of all consists in a train	Static SIL 2	Defined in IEC61375-2-3
TTDB	Holds TCMS functions/devices of all consists in a train, Defines sequence of cars, driving direction, leading cab...	ETBN (ECSP)	Any device in need for URI addressing (DNS) and operational train information (driving direction, vehicle/consist orientation, train-wide communication etc.)	Dynamically generated Currently SIL 2 Needs to be SIL 4	Result of train inauguration Defined in IEC61375-2-3
Switch -TSN	Timed forwarding of TSN traffic, Switch port usage/assignment	e.g. SDN-Controller	Consist Switch	Static (Precomputed Set) SIL 0	Changed on redundancy failover
ED-TSN	Timed transmitting/receiving of TSN data	End device (FDF)	Functional Distribution Framework	Static SIL 0	
End Device Config	Shared variables / IO / Functions setup Security Credentials, Key exchange	End device (FDF)	Functional Distribution Framework, (TCMS-) Applications	Static SIL 0-2-4	SIL of the highest SIL of addressed function
FW-Updates	Maintenance updates of software	Any networked device	FDF or application (function)	Static SIL 0-2-4	SIL of the highest SIL of a device function

Table 4: List of configurations within a consist

6.1.2 System Communication Integration

During the requirements and design phase, it is needed to carefully plan the amount of traffic transported via ETB and ECN. Since different communication classes (deterministic/non-deterministic) are used on the same communication channel, the whole scheduling of the

traffic/network load needs to be balanced. Also, the application specific constraints - such as which data needs to be sent/received by which frequency – need to be considered.

For the certification of applications exchanging deterministic data, the chosen timing needs to be guaranteed. Also, the configuration of all network devices (switches, EDs, VCUs/CCUs) needs to be derived from such a schedule.

It is possible to choose between two different approaches, as shown in Figure 41:

1. Each application is certified for well-defined limits/constraints, giving the class and amount of data, and its maximum cycle times. For example, data with a maximum cycle time of 10ms can be delivered in a 2,5ms slot.
 - ➔ The overall network schedule is calculated by using all application constraints and is then certified as shown below:

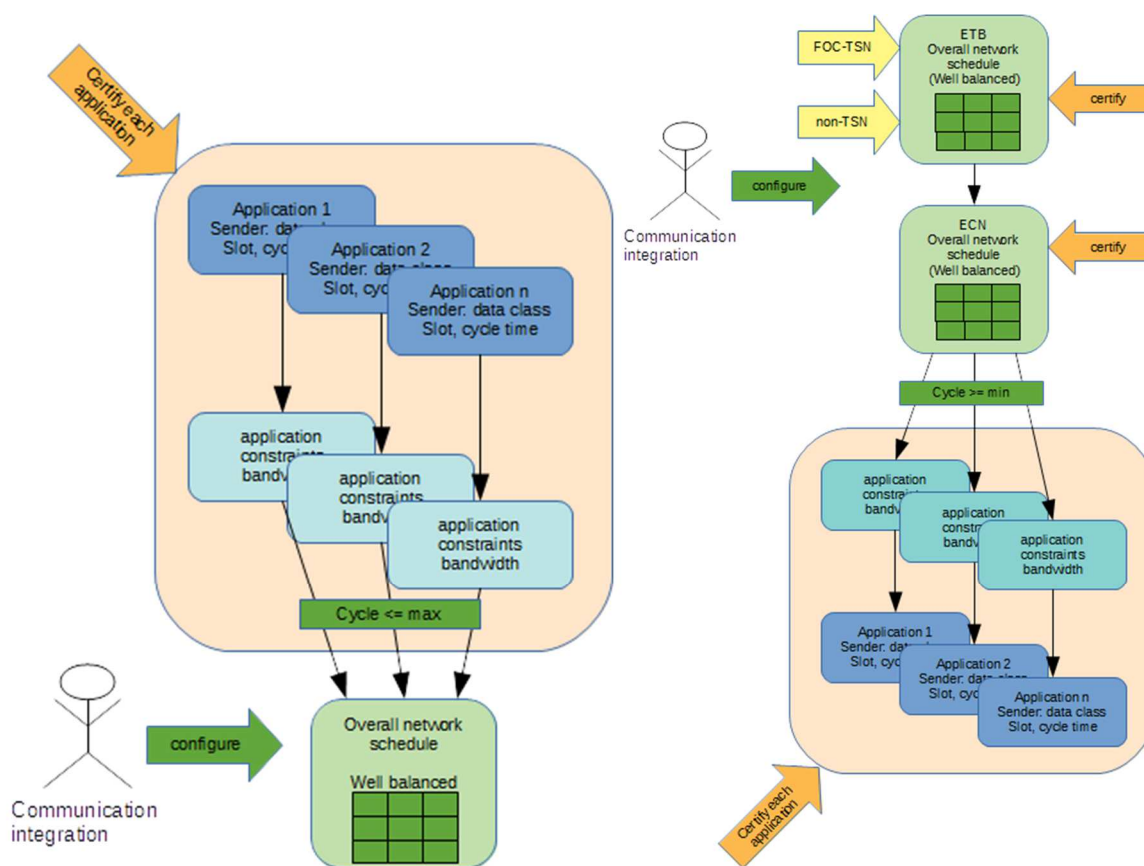


Figure 41: Bottom-up: Application constraints lead to network schedule (left) and Top-Down: Network schedule is calculated in the first step (right)

2. An overall well-balanced network schedule is predefined and reserves a certain amount of communication slots for each application. Then, the application constraints need to be matched to those reserved slots.
 - ➔ Network and applications can be certified incrementally.
 - ➔ The bandwidth and performance of the network is well balanced.

6.2 TSN network configuration

The TSN network requires a configuration to set up how and when the network is being used and that the load is predictable especially for the time-critical communication. In addition, the clock synchronisation must be configured. The overall configuration workflow of such a network is depicted in Figure 42. The configuration is used to perform the queue assignment, define the scheduling tables for the Time-Aware Shaper mechanism and the set up the clock synchronization mechanism.

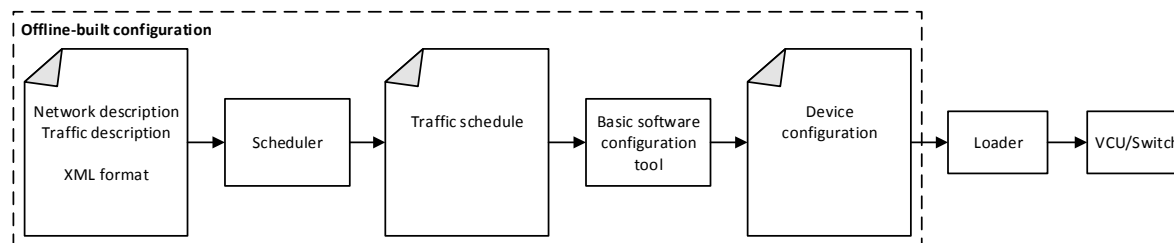


Figure 42: TSN network configuration workflow

The input to the configuration process is a file in XML format containing the parameters that describe the network and the traffic. A scheduler is in charge of taking this input parameters obtaining a schedule for the traffic (schedule for IEEE 802.1Qbv [23]), which is then translated to the device configuration. The device configuration is built-offline.

6.2.1 Loading configurations to network devices

For configuration purposes, the TSN network is based on an information exchange protocol, the Network Configuration Protocol, NETCONF (IETF RFC 6241²). NETCONF provides mechanisms to install, manipulate, and delete the configuration of network devices.

NETCONF is a client-server protocol. A server is typically a network device, whereas the client runs as an application on one central or a set of distributed network managers. In a network with a known set of devices or mainly set up with a configuration-generated offline, a central network manager is the simplest solution. A server could be placed also on central computing resources or vehicle computers.

A typical configuration session follows the following approach, as pictured in Figure 43:

- The application/network manager requests the current configuration of the network devices in the network
- Each network device in the network responds to this request with their current configurations
- The application updates the configuration
- The application distributes the configuration to the network devices
- The network devices load the new/modified configurations

Secure communication is mandated by NETCONF in terms of authentication, data integrity, confidentiality, and replay protection. This is handled by the underlying transport protocol (e.g. SSH or TLS).

² <https://tools.ietf.org/html/rfc6241>

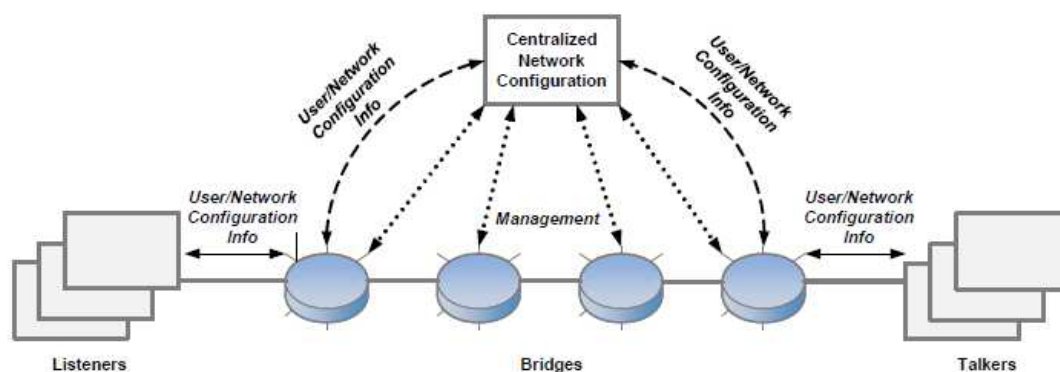


Figure 43: Network Configuration with NETCONF

For the specification of the configuration, NETCONF utilizes XML for all operations. Although it defines the protocol for exchanging information between clients and servers, it does not specify the semantics of the information exchanged. To improve the interoperability of various TSN devices, the semantics of these exchanges can be defined in a YANG model. YANG (Yet Another Next Generation) is a data modelling language for the definition of data sent over the NETCONF network configuration protocol, which is defined in IETF RFC 6020.

YANG defines exactly how NETCONF-based operations, including configuration, state data, Remote Procedure Calls (RPCs), and notifications are represented to allow a complete description of all the data that is sent between the configuration manager and the network devices.

Whereas YANG models for basic network configuration, e.g. topology-related topics have existed for several years, the definition of the models required for the features that are being added by the TSN working group are still ongoing. The configuration of scheduled data transmission according to 802.1Qbv [23] and ingress policing according to 802.1Qci [24] were recently completed. YANG models for other important configuration parameters, for clock synchronization according to 802.1AS-rev [21] or for redundancy support according to 802.1CB have not yet been completed.

6.3 Tooling & Development

Dedicated tools used for configuring the standard hardware and software with application dependent data (parameter values) and algorithms (block diagrams, state charts) underlie the same requirements for the development as the development of the generic software (EN 50657 [27], chapter 8.1.1)

A rigid separation between the generic software and the application data/algorithms shall be enforced. For example, it shall be possible to recompile and update either the generic software or the application data/algorithms without needing to update the other, unless there has been a change to the defined interface.

6.3.1 EN 50657 [27] and Tooling

The EN 50657 requires:

- it needs to be applied also for “supporting tools”
- automatic testing tools as well as integrated development tools are recommended
- tools should be available as early as possible
- tools need to fulfil their intended purpose

- tools need to be validated (compliance to SIL level, appropriate documentation or other means of conformance evidence)

EN 50657 [27] classifies tools as follows:

- T1: generates no outputs, which can directly or indirectly contribute to the executable code (including data) of the software
EXAMPLE: a text editor or a requirement or design support tool with no automatic code generation capabilities; configuration control tools.
- T2: supports the test or verification of the design or executable code, where errors in the tool can fail to reveal defects but cannot directly create errors in the executable software
EXAMPLE: a test harness generator; a test coverage measurement tool; a static analysis tool.
- T3: generates outputs, which can directly or indirectly contribute to the executable code (including data) of the safety related system
EXAMPLE: a source code compiler, a data/algorithms compiler, a tool to change set-points during system operation; an optimizing compiler where the relationship between the source code program and the generated object code is not obvious; a compiler that incorporates an executable run-time package into the executable code.

Configuration management shall ensure that for tools in classes T2 and T3, only justified versions are used. Software tools shall be selected as a coherent part of the software development activities.

6.3.1.1 System Configuration Tools

When configuring the network (switches, controller, TSN, FDF...), dedicated tools, most of them running on central processing units, have to be used. They all belong to the category of “application data tools that produce or maintain data which are required to define parameters and to instantiate system functions” referenced in the chapter above.

6.3.2 Configuration flow – FDF Application/Functions driven

Figure 44 depicts the configuration flow from the application to the network devices/switches. There is the need to configure the FDF with its different layers as well as the network consisting of different switches on consist level as well as on ETB level.

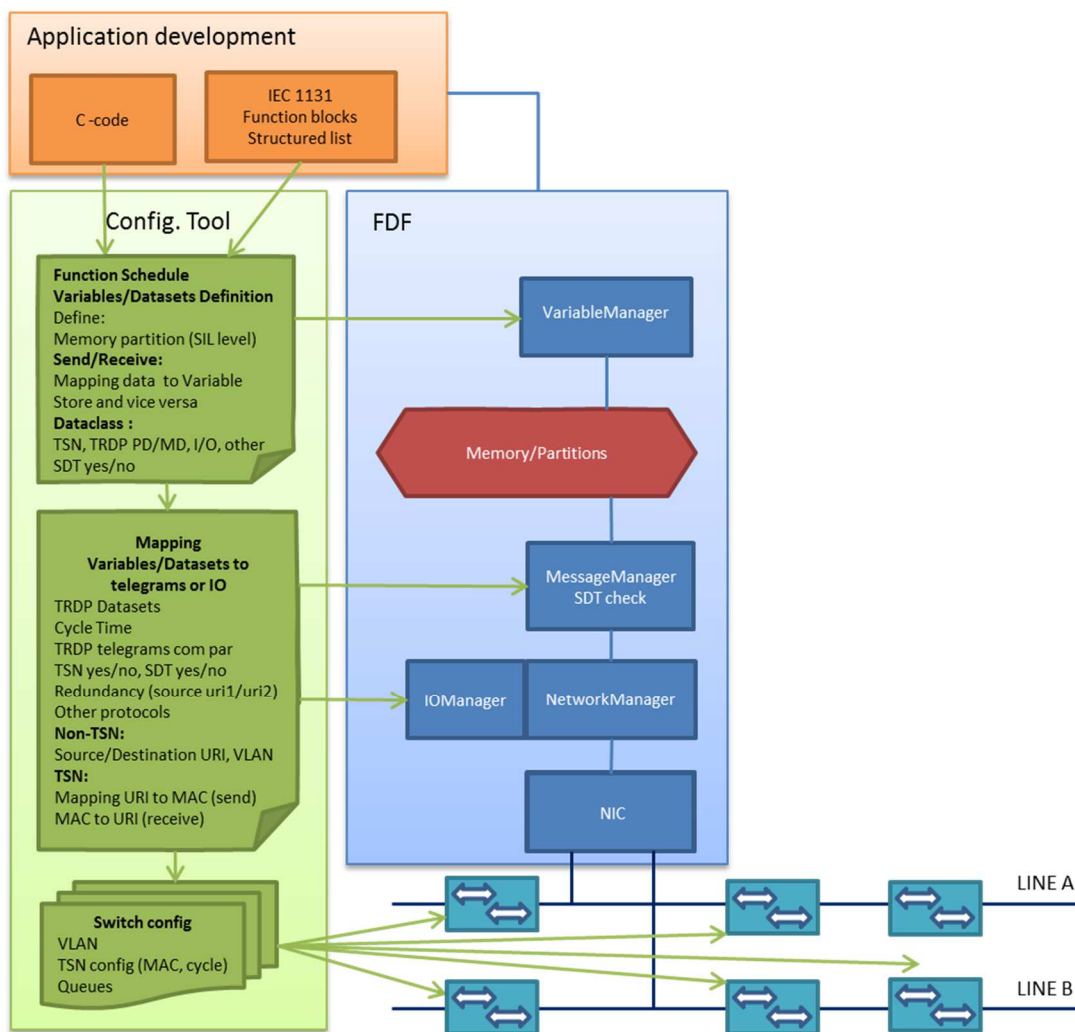


Figure 44: Application driven configuration – Steps

The generation of those configuration files needs to be application driven. Train applications can be programmed in different ways such as by using an IEC1131-interface [28] (e.g. function blocks) or plain C/C++ code using already implemented library functions. When combining those predefined functions to implement an application, they shall also create the respective configuration files or deliver lists for the respective tools. The main task is the function registering and the mapping of data to the memory (Variable Store) and vice versa. Also, the mapping between data and telegrams or HW-I/O is needed. Generally, the source and destination URIs are used to directly address remote functions inside a consist.

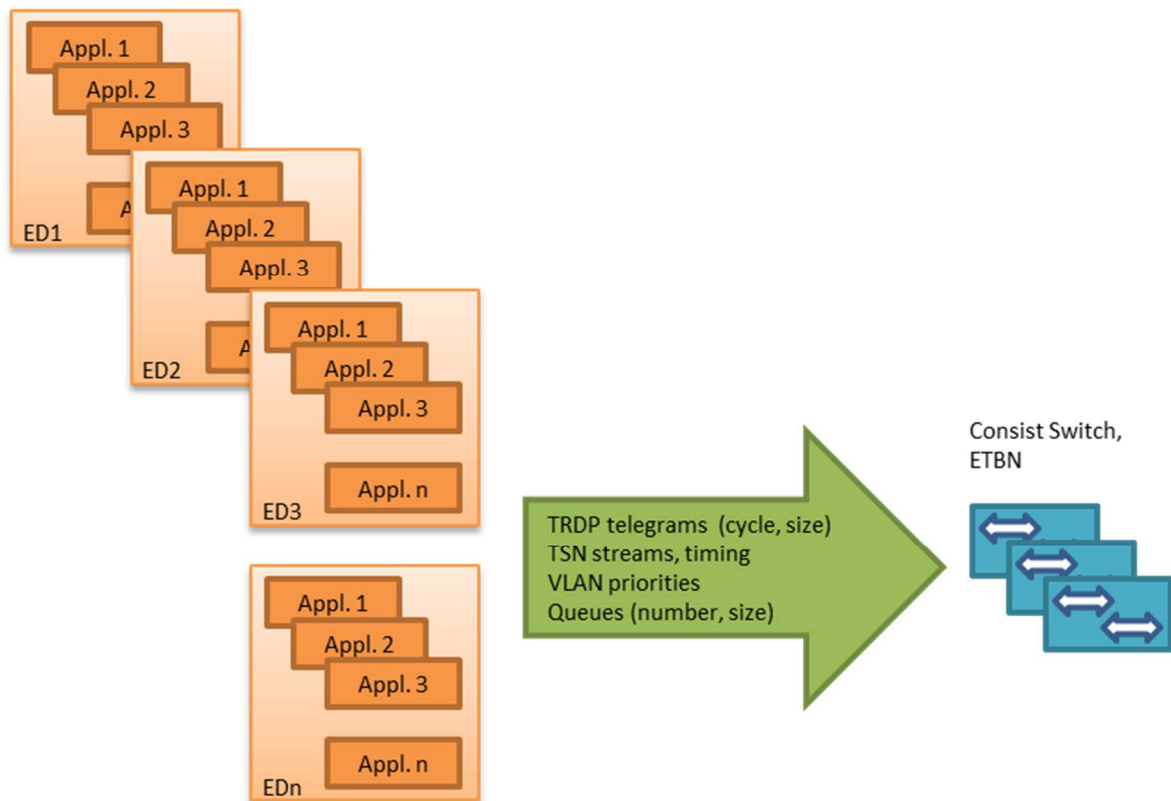


Figure 45: Application driven configuration – several End Devices

6.3.2.1 TRDP

For the standard TRDP a well-defined communication description in XML format has been defined in [15]. This XML scheme allows the definition and configuration of all TRDP related communication parameters, including SDT.

The configuration covers:

- Device resources (memory)
- Bus/Ethernet interface parameters (interface/IP address)
- Process parameters (task cycle time, priority)
- Process & message data communication parameters (IP port, QoS, TTL, cycle time, timeouts, retries, etc.) – both as defaults and per telegram
- Telegram parameters (ComId, datasetId, Source/Dest URIs, SDT...)
- Dataset definitions (data element type, size, unit, scale...)
- Debug/logging configuration

The basic definition of this scheme can easily be extended to cover TRDP over TSN by supplying the necessary TSN parameters (e.g. virtual interface selection) to the communication parameter XML tag.

The TRDP XML configuration scheme currently does not support a mapping between datasets and I/O-variables. Such an addition to the scheme must be included by the FDF configuration and should also be defined in the revised standard (IEC 61375-2-3 [15], Annex C).

6.3.3 Configuration flow – Topology/network driven

Besides the application driven part of the system configuration there is the need to configure from the topology point of view.

Figure 46 below shows an example of a Train network topology (Variant D)

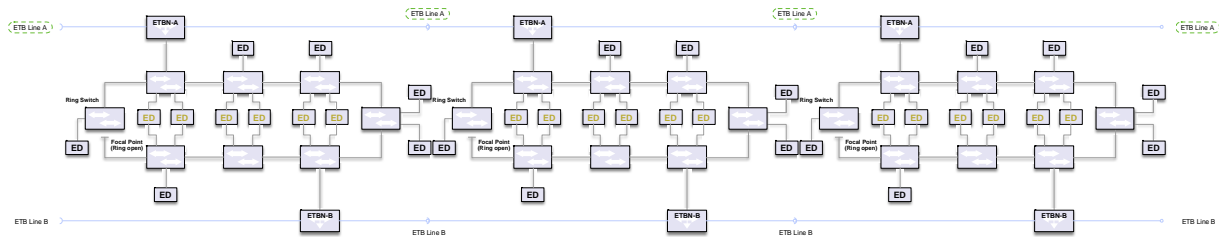


Figure 46: Example Network Layout

The maximum number of ETBNs, Ring Switches and Devices is as follows:

- ETBN: Ethernet Train Backbone Node (1...32 * 2 for Variant D)
- RS: Ring Switch in CN: 1...63
- Veh: Vehicle 1...32
- Dev: Devices 1...16382

A tool configuring the whole network needs a network topology list as follows:

- Layout
- Addressing scheme for each Switch and Device
- TSN configuration for all Switches and Devices (MAC addresses)
- VLAN-Tags

There is the need for a central calculation of network load (operational data: TSN, other), at least during commissioning.

Chapter 7 Verification & Validation Concept

Opposite to waterfall method of system development lifecycle, where the testing starts *after* the development has been finished, the method chosen in Safe4RAIL - V-model method - encourages testing preparation to run in *parallel* with requirements, design, and development phases. The name of this method comes from its visual representation – diagram in the shape of a letter “V”, as shown in Figure 47. The left side of the “V” stands for the project definition (concept, requirements, architecture and low-level design), whereas the right side represents the project integration and test (integration of parts of the system, verification, validation, operational system maintenance).

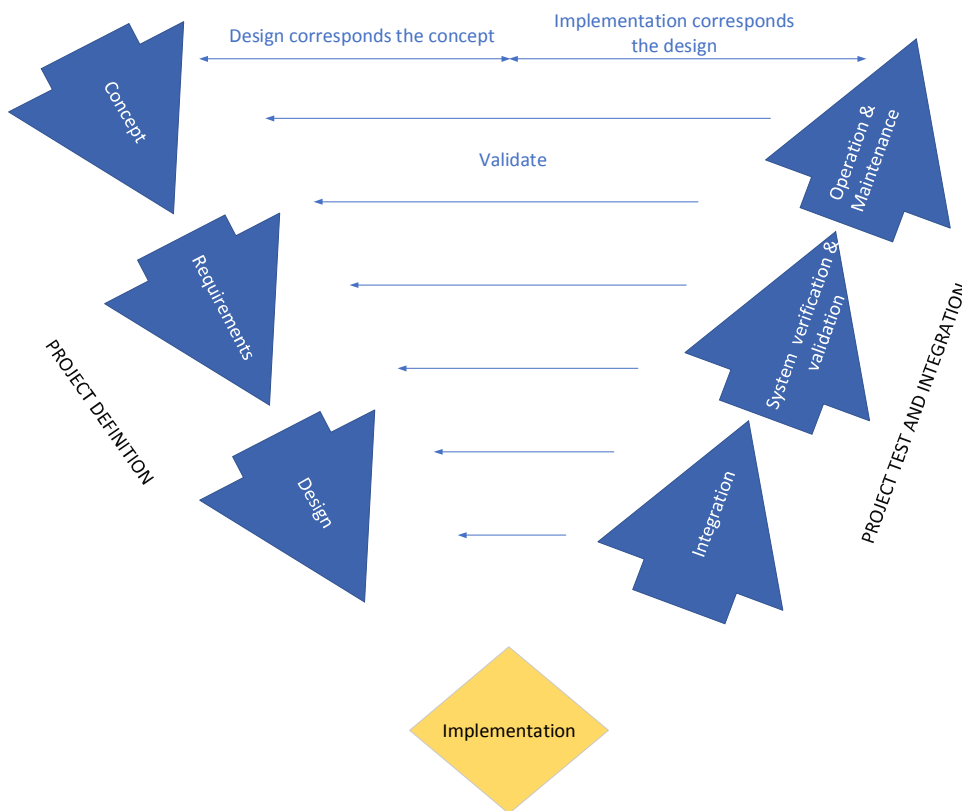


Figure 47 V-Model, a development method for a system lifecycle

The objective of verification is to demonstrate that the requirements of each life-cycle phase have been fulfilled. Validation is used by manufacturers and suppliers to test the functionality of the product and to gain confidence that certain safety requirements are met. Validation can be done in many stages of a system life cycle, once a prototype or the real system is available. Dependability evaluation is carried out for validation purposes to get the required evidence that all has been done to build a safe system as far as reasonably practicable. CENELEC standard EN50126 [29] requires the verification and validation in all phases of a system life-cycle.

The verification and validation (V/V) processes of ETBN, consist switches and end-systems which are compliant to the TSN protocol is expensive and time-consuming. Therefore, simulation tools are used as a cost and time efficient options for analyzing the temporal and non-temporal attributes of TSN-capable DbD. The simulation models provide an opportunity for train manufactures and operators to simulate various train network topologies, train

components (e.g ETBN) and the railway use cases at low cost and with high precision (e.g. nanoseconds).

The verification and validation framework developed for the next generation TCMS focuses on the validation of the communication network protocol. The framework provides the platform for validating communication protocols with the generic capability of targeting deterministic communication protocols such as TTEthernet [8] and TSN [9].

7.1 Dependability Evaluation Methods

Based on the verification of dependability of a system, an evaluation could be done by:

- Analysis,
- Field Experience,
- Testing.

When requirements are verified by measuring product performance and function under various simulated environments, the method is referred to as “Test”. When verification is achieved by performing theoretical or empirical evaluation by accepted techniques, the method is referred to as “Analysis”. Dependability evaluation by field experience involves the observation of a system during its operational phase. However, such observation could take a long time to gain the appropriate confidence required for validation. Considering timeliness and strong evidence in attaining the required confidence level for a dependable system, testing mechanisms such as fault injection testing satisfies this requirement.

A fault injection framework is hereby developed for the next generation TCMS targeting the communication network protocol.

7.2 Network Verification by Fault Injection

Fault injection is a dependability evaluation technique that observes a Unit Under Test (UUT) in a controlled experiment by accelerating faults to probe the systems response. The next generation TCMS network is designed as a highly dependable system and must tolerate any potential faulty scenario.

Dependability measurement relies on controlled fault injection experiments that are able to observe the behaviour of a system under the effect of faults. Fault injection thereby provides the platform for robustness assessment, test of error handling and fault tolerance of a system, and for assessment solutions to improve dependability.

In everyday language, the term fault, failure and error can be used interchangeably. However, these terms have distinctive meaning in fault tolerance computing parlance. A fault can be a physical defect or imperfection in software (bug). An error is the manifestation of the fault. A failure is when an error is propagated and results in an incorrect output. To state clearly, a fault produces an error which may then propagate through the system thereby causing the complete or partial failure of a system.

Over the past few years various fault injection techniques have been proposed. They can be basically grouped into hardware-based, software-based, simulation-based, emulation-based and hybrid fault injection [30]. The fault injection framework developed in this work is a hardware-based injection technique.

Hardware-based fault injection involves altering the parameters of the UUT at the physical level. Faults are injected into a UUT with or without contact. A non-contact fault injection would involve changing the physical environment, such as the use of heavy ion radiation or electromagnetic interference. The hardware fault injection implemented for this project injects faults into data transiting over the link. An external hardware, a field programmable gate array

(FPGA) was connected according to a cut-through paradigm over the physical link that connects the network components of the network under test.

7.3 Failure criteria for communication

IEC 61508 recommends the following failure modes when data communication is used in implementing safety functions, as is intended for the next generation TCMS communication system:

- repetition,
- deletion,
- insertion,
- re-sequencing,
- corruption,
- delay and
- masquerade [31].

Additional failure modes that could be considered:

- **omission failure** - a sending end system fails to transmit a frame, or the receiving end system fails to receive a transmitted frame.
- **babbling idiot** - transmission of arbitrary messages at random point in time
- **link failure** – a discontinuity imposed on the link due to physical damage or other factors on a link connecting network components together.
- **crash failure** - transient or permanent crash failure of an end system or switch component, where the affected end system or switch does not produce any interaction temporarily or permanently with the other network participants.
- **clock synchronisation failure** - the loss of synchronisation
- **stuck-at failure** - a continuous retransmission of a particular frame stuck at a transmitter.

The fault-injection framework is able to inject these type of faults and measure their effect on the system behaviour.

Chapter 8 Summary and Conclusions

This deliverable summarizes the concepts developed to enable the implementation of the Drive-by-Data Technology for the Integrated Modular Platform. The main achievements are:

- New network architecture dealing with effective use of network and highest reliability standards
- Flow control concept for scheduled data traffic on ETB and ECN for bounded latency
- New TRDP data traffic class (imposing solutions for clock synchronization, flow control and redundancy management)
- Configuration methodology that takes into account the dimension of time (and latency) for NG-TCN networks.

The new network topology defined within Safe4RAIL is made adequate for the required traffic scheduling. On ETB level, two detached ETB lines have been introduced as opposed to the previously used aggregated ETB lines, creating two virtual TSN planes. On ECN level, dual-homed critical end devices are connected to both planes and communicate seamlessly also in the case of a network fault. Backward compatibility to existing solutions is ensured for conventional and legacy devices. Their connection to the ECN ring network is a single Ethernet interface.

The NG-TCN modified some of old services, but also defining new ones. The train inauguration is modified to handle the new ETB topology and support SIL4 data communication. The service for scheduled data traffic is introduced, as well as a new safe data transmission protocol, classified for SIL4. These services needed new or modified protocols, which were included in the system, like the protocol for FRER (IEEE 802.1CB), per-stream filtering and policing (IEEE 802.1Qci) or the precise time synchronization protocol, based on the IEEE 802.1AS standard. With these changes, most network components had to evolve and upgrade to provide additional functionalities. The highest influence had the clock synchronization and scheduled traffic, affecting most of them, including TSN-aware end devices.

Some general rules and guidelines on configuration of the network devices are summarized in Chapter 6.

The verification and validation framework for the NG-TCN was also developed. It is the platform which supports validation of deterministic communication protocols such as TTEthernet [8] and TSN [9]. It provided means to evaluate the fault tolerance mechanisms (FTMs) implemented in the network. The next phase is focused on extending the framework to use an efficient synchronisation mechanism, ensuring the same global time. Future works would also enable the fault injection framework to utilize the extensions opportunities provided for the TSN framework.

Chapter 9 List of Abbreviations

AFDX	Avionics Full-Duplex Switched Ethernet
API	Application Programming Interface
BC	Boundary Clock
BIT	Built-In Tests
BMCA	Best Master Clock Algorithm
BSP	Board Support Package
CBS	Credit-Based Shaper
CCU	Central Computing Unit, functional name of VCU
cED	critical End Device
CENELEC	Comité Européen de Normalisation Électrotechnique (fr., European Committee for Electrotechnical Standardization)
ComId	Communication Identifier
COTS	Commercial off-the-Shelf
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
CS	Consist Switch
cstUUID	Universally Unique Identifier of a consist
CTA	CONNECTA
DbD	Drive-by-Data
DHCS	Data Handling and Communication System
DNS	Domain Name System
ECN	Ethernet Consist Networks
ECSC	ETB Control Service Client
ECSP	ETB Control Service Provider
ED	End Device
ED-S	Safe End Device

EN	European Norm
ETB	Ethernet Train Backbones
ETBN	Ethernet Train Backbone Node
FDF	Functional Distribution Framework
FI	Fault Injection
FOC	Functional Open Coupling
FPGA	Field Programmable Gate Array
FRER	Frame Replication and Elimination for Reliability
FSM	Finite State Machine
FT AVG	Fault-Tolerant Averaging
GCL	Gate Control List
GMC	Grandmaster Clock
GPS	Global Positioning System
gPTP	Generalized PTP
HMI	Human Machine Interface
hsGM	hot-standby Grandmaster
HTML	HyperText Markup Language
HVAC	Heating, ventilation, and air conditioning
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IMP	Integrated Modular Platform
I/O	Input/Output
IP	Internet Protocol
IPV	Internal Priority Value
LLDP	Link Layer Discovery Protocol
LRV	Light Rail Vehicle
MAC	Media Access Control
MCU	Microcontroller Unit

MD	Message Data
MISRA	Motor Industry Software Reliability Association
MIO	Multiple Input/Output
MMU	Memory Management Unit
MPSoC	Multiprocessor System-on-Chip
MPU	Memory Protection Unit
MTBF	Mean Time Between Failures
MVB	Multifunction Vehicle Bus
NAC	Network Access Controller
NAT	Network Address Translation
NETCONF	Network Configuration Protocol
NG TCMS	New Generation Train Control and Management System
NIC	Network Interface Controller
NP-hard problem	Nondeterministic Polynomial time Problem
NSTF	Non-Stream Transfer Function
NTP	Network Time Protocol
NUT	Network Under Test
NWIP	New Work Item Proposal
OEM	Original Equipment Manufacturer
OC	Ordinary Clock
OS	Operating System
PA	Public Address
PCF	Protocol Control Frames
PD	Process Data
PDF	Portable Document Format
PD-PDU	Process Data Protocol Data Unit
pGM	primary Grandmaster
PNG	Portable Network Graphics

PoC	Proof of Concept
PSFP	Per-Stream Filtering and Policing
PTP	Precision Time Protocol
QoS	Quality of Service
RPC	Remote Procedure Call
RTOS	Real-Time Operating System
RTSP	Real-Time Streaming Protocol
Safe4RAIL	Safe4RAIL
SAP	Service Access Point
SDK	Software Development Kit
SDT	Safe Data Transmission <i>End-to-end protocol over an untrusted communication channel</i>
SDN	Software-Defined Networking
SDTv2	Safe Data Transmission v2
SID	Source Identifier
SIL	Safety Integrity Level
SMI	Safe Message Identifier
SSC	Safety Sequence Counter
SSH	Secure Shell
TAP	Terminal Access Point
TAS	Time-Aware Shaper
TCMS	Train Control and Management System
TCXO	Temperature Compensated Crystal Oscillator
TLS	Transport Layer Security
TND	Train Network Directory
TRDP	Train Real Time Data Protocol
TSN	Time Sensitive Network
TTDB	Train Topology DataBase

TTL	Time To Live
UDV	User Data Version
UIC	Union internationale des chemins de fer (fr., International Union of Railways)
URI	Uniform Resource Identifier
UUID	Universal Unique Identifier
UUT	Unit Under Test
VCU	Vehicle Control Unit, implementation of CCU
VDP	Vital Data Packet
VID	VLAN Identifier
VLAN	Virtual Local Area Network
VOS	Virtual Operating System
V/V	Verification/Validation
WTB	Wire Train Bus
XML	Extensible Markup Language
XPATH	XML Path Language
XSLT	eXtensible Stylesheet Language Transformations
YANG	Yet Another Next Generation

Table 5: List of Abbreviations

Chapter 10 Bibliography

- [1] IEC, *IEC61375-2-5 Electronic railway equipment – Train communication network (TCN) – Part 2-5: Ethernet train backbone*, 2015.
- [2] IEC, *IEC61375-3-4 Electronic railway equipment – Train communication network (TCN) – Part 3-4: Ethernet Consist Network (ECN)*, 2014.
- [3] M. Jakovljevic, *Safe4RAIL Deliverable D1.4 - Refined Drive-by-Data Concept Design*, Safe4RAIL, 2017.
- [4] B. Loehr, "Safe4RAIL D1.6 - Network Design Methodology and (Re)-Configuration," 2017.
- [5] D. M. Saatci, "Safe4RAIL D1.7 - Safety Concept for Ethernet Networks with Recommendations for Regulatory and Standardization Activities," 2018.
- [6] I. Odrizola, *Safe4RAIL Deliverable D2.3 - Report on 'TCMS framework concept' design, security concepts and assessment*, Safe4RAIL, 2018.
- [7] D. Onwuchekwa and M. Pahlevan, *Safe4RAIL D1.5 Network Modelling and Simulation Concepts for NeXT Generation TCMS networks and Verification Concepts for Next-Generation TCMS*, 2018.
- [8] SAE International, "SAE AS6802 Time-Triggered Ethernet," SAE Standards, Warrendale, PA, 2011, 2011..
- [9] Time-Sensitive Networking Task Group, "TSN - Time-Sensitive Networking".
- [10] AEEC, "ARINC Project Paper 664," in *ARINC Data Networks, Part 7, AFDX Network*, Aeronautic Radio Inc., Nov. 2003.
- [11] IEC, *IEC61375-2-5:2015: Electronic railway equipment – Train communication network (TCN)*, 2012.
- [12] IEEE, *IEEE 802.1AB Station and Media Access Control Connectivity Discovery*, 2005.
- [13] IEEE, *IEEE Std 802.1D IEEE Standard for Local and Metropolitan Area Networks: Media Access Control (MAC) Bridges.*, 2004.
- [14] IEEE, *IEEE Std 802.1Q IEEE Standard for Local and metropolitan area networks--Bridges and Bridged Networks*, 2014 .
- [15] IEC, *IEC61375-2-3: Electronic railway equipment – Train communication network (TCN) – Part 2-3: TCN communication profile*, 2016.

- [16] CENELEC, *EN 50657:2017 Railway applications – Rolling stock applications – Software on Board Rolling Stock*, CENELEC, 2017.
- [17] CENELEC, “EN 50128:2011. Railway applications - communications signaling and processing systems - software for railway control and protection systems.,” 2011.
- [18] G. Hans, *CONNECTA D3.5 - Drive-by-Data Architecture Specification*, 2018.
- [19] IEEE, *IEEE Std 1588 IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems*, 2008 .
- [20] IEEE, *IEEE Std 802.1AS IEEE Standard for Local and Metropolitan Area Networks - Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks*, 2011.
- [21] IEEE, *IEEE P802.1AS-rev/D7.0 Timing and Synchronization for Time-Sensitive Applications*, 2018.
- [22] M. S. W. Sorea, “Classification and analysis of failure modes for time-triggered systems.,” 2007.
- [23] IEEE, *IEEE P802.1Qbv/D3.1 Bridges and Bridged Networks—Amendment: Enhancements for Scheduled Traffic*, 2015 .
- [24] IEEE, *IEEE P802.1Qci/D2.1 Bridges and Bridged Networks—Amendment: Per-Stream Filtering and Policing*, 2016 .
- [25] IEEE, *IEEE P802.1CB Frame Replication and Elimination for Reliability*, 2017 .
- [26] MEF, *MEF 10.3 Ethernet Services Attributes Phase 3*, 2013.
- [27] DIN, *EN 50657 Railways Applications - Rolling stock applications - Software on Board Rolling Stock*, 2017.
- [28] DIN, *EN 61131-3 Programmable controllers - Part 3: Programming languages*, 2014.
- [29] CENELEC, “EN 50126-1:2015. Railway applications - the specification and demonstration of reliability, availability, maintainability and safety (rams),” 2015.
- [30] A. V. Ziade, *A Survey on Fault Injection Techniques*, The International Arab Journal of Information Technology, vol.1, no. 2, 2004.
- [31] IEC, *IEC 61508 Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems*.
- [32] CENELEC, “EN 50129:2016. Railway applications - communication, signaling and processing systems - safety related electronic systems for signaling,” 2016.
- [33] CENELEC, “EN 50159:2011. Railway applications - communication, signaling and processing systems - safety-related communication in transmission systems.,” 2011.

- [34] "TSN," Time-Sensitive Networking Task Group, [Online]. Available: <http://www.ieee802.org/1/pages/tsn.html>. [Accessed Aug 2017].
- [35] German Commission for Electrical, Electronic and Informational Technologies, GK 14, *DIN EN 61508 Functional safety of electrical/electronic/programmable electronic safety-related systems*, 2011.
- [36] CENELEC, *CLC/SC 9XA Electrical and electronic applications for railways - Communication, signaling and processing systems*, 2016.
- [37] ISO/IEC, *ISO/IEC 15408 - Common Criteria*, 2009.
- [38] DIN, *DIN VDE V 0831-104 Electric signaling systems for railways - part 104: It security guideline based on IEC 62443, draft*, 2015.
- [39] DIN, *DIN VDE V 0831-102. Electric signaling systems for railways - part 102: Protection profile for technical functions in railway signaling, draft*, 2013.
- [40] W. Steiner, G. Bauer, B. Hall and M. Paulitsch, "TTEthernet: Time-Triggered Ethernet in Time-Triggered Communication,," pp. pp. 181-220, Aug 2011.
- [41] IEC, "IEC61375-1:2012 Train Communication Network (TCN) - part 1: TCN general architecture," IEC, 2012.
- [42] IEC, *IEC62443 Industrial communication networks – Network and system security*, 2018.
- [43] IEEE, *IEEE Std 802.1Qav - Forwarding and Queuing Enhancements for Time-Sensitive Streams*, 2009.