D1.1

State-Of-The-Art Document on Drive-by-Data

Project number:	730830
Project acronym:	Safe4RAIL
Project title:	Safe4RAIL: SAFE architecture for Robust distributed Application Integration in roLling stock
Start date of the project:	1st of October, 2016
Duration:	24 months
Programme:	H2020-S2RJU-OC-2016-01-2

Deliverable type:	Report
Deliverable reference number:	ICT-730830 / D1.1 / 1.1
Work package	WP 1
Due date:	December 2016 – M03
Actual submission date:	30th of December, 2016

Responsible organisation:	TTT
Editor:	Mirko Jakovljevic
Dissemination level:	Public
Revision:	1.1

Abstract:	This document provides an overview of state-of-the-art in relevant technologies for deterministic high-bandwidth networking and reveals different use cases in transportation industries.
Keywords:	Deterministic Ethernet, Standards, Technology, Integrated Architectures, Ecosystem, Trends

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

Editor

Mirko Jakovljevic (TTT)

Contributors (ordered according to beneficiary numbers)

Mirko Jakovljevic, Arjan Geven, Astrit Ademaj, Georg Gaderer, Christian Fidi (TTT)

Erik Männel, Jonas Rox, Dr. Donatas Elvikis (IAV)

Bernd Löhr (NEW)

Executive Summary

System integration represents a core capability required for the design of advanced integrated architectures. 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, certifiabilty, robustness and system lifecycle costs.

This document provides an overview of state-of-the-art in relevant technologies for deterministic high-bandwidth networking and reveals different use cases in transportation industries aerospace, automotive, railway, and space.

Proven core technologies for deterministic Ethernet integration which could satisfy requirements of advanced integrated architectures for mission-, time-, and safety-critical applications are described in ARINC664 and SAE AS6802. Their implementations include the properties which correspond to “white channel” communication approach, and provide congestion-free communication with full control of temporal behaviour for all critical dataflows in the system. Formally verified and robust fault-tolerant distributed clock algorithms support the control of system time in the most demanding critical applications.

Currently, IEEE TSN (Time-Sensitive Networking) suite of standards is in development and it could further develop to gain the capabilities relevant for critical applications in automotive, industrial and IoT applications. Other technologies such as software-defined networking (SDN), DetNet or WDM can expand the range of system integration options in critical integrated systems over the longer term (10-15+ years).

With the objective to design scalable, reusable, reconfigurable and certifiable system architectures, system integration and Ethernet networking cannot be seen separately from the software platform. Well-designed generic integrated modular platform are designed as one subsystem, which provides all services and capabilities required for hosting non-critical and critical (SIL0-4) applications.

In addition to safety, the network and system integration security becomes more important. Security issues may lead to safety-related consequences and risks, which must be carefully managed and considered during the design of robust integrated modular platforms.

Contents

List of Figures

List of Tables

Chapter 1 Ethernet Networking

1.1 Brief Overview – Ethernet Principles for Full Duplex Switched Ethernet (IEEE802.1-2000)

1.1.1 Ethernet Basics

Ethernet is a flexible, scalable networking standard for high bandwidth communication which evolves and alters over time to satisfy different application requirements, with different quality of services, determinism and bandwidth.

Ethernet evolves over time, and adapts to new requirements for high-volume applications.

Ethernet must be clearly differentiated from monolithic communication standards such as low-speed fieldbuses i.e. MVB/TWB, PROFIBUS, MIL-1553, CAN, TTP or ARINC429. Monolithic fieldbus (serial databus) standards have been designed for a specific set of applications and simpler systems, and they have been used over decades almost unchanged.

Ethernet is a family of frame-based LAN protocols defined in IEEE802 which share common properties such as:

- Frame format with variable packet size
- Media access with fair arbitration
- Set of physical layers

Since 1999, Ethernet finally evolved from Ethernet bus with hubs/bridges, to become a fully switched full duplex Ethernet network, with routers/switches replacing the bridges between two Ethernet bus sections, and providing only a point-to-point connectivity between end stations and routing network devices (switches, or in purist IEEE terminology “bridge”).

1.1.2 Ethernet Networks: Historical Overview

In due course, Ethernet services are added, upgraded or have become obsolete within the period of 10-20 years. As a family of networking protocols, Ethernet has adapted over time to different industry-specific applications and use cases.

High-bandwidth LAN networks have evolved over 40 years through several evolutionary phases (early IEEE802.1 Ethernet Bus, 802.4, 802.5) since 1980s and the winner since 2000 was switched full-duplex Ethernet (Codename “FastEthernet”). Those networks have very limited commonality with switched full-duplex Ethernet standard today. In the past there were different competing standards in IEEE802. The big technology struggle of the 80's and 90's was Token Ring vs. Ethernet. Even though it was technically superior at that time, Token Ring was overpriced (5-6x higher pricing then Ethernet, due to higher complexity for management, and IBM appetite for revenue) and ultimately did not succeed. Token Ring in reality was rarely a ring. It was a star architecture with a token passing protocol. In the centre of the star there is a unit comparable to today’s Ethernet switch. With acquisition of Crescendo (1993) and Kalpana (1994), Cisco entered the switched Ethernet market [1]. While the slower 16Mbit/s Token Ring was superior in many cases to 100Mbit/s Ethernet due to congestion mitigation, IBM started 100Mbit/s Token Ring development, but it was too late. Switched full-duplex Ethernet resolved many system integration challenges, has added full

duplex communication, increased line speed, and became a commodity on the market in the late 1990s. Both Token Ring, Token Bus and ThinEthernet were obsolete by the early 2000s.

Over time the IEEE802 has disbanded standard working groups in early 2000s which focused on token bus (802.4) and token ring (802.5), and has adopted new amendments (additional functions) in 802.3 focusing on switched Ethernet, and in 802.1 focusing on higher layers.

These two groups (802.1 and 802.3) determine Ethernet technology development and adaptation to new applications. As a result of all evolutionary adaptations since 1999 Full Duplex Switched Ethernet (Fast Ethernet), Ethernet can be used in many different high-volume markets (LAN, IT/enterprise). In other emerging lower-volume niches (industrial, manufacturing, storage, datacenter) it was used with adaptations or special standard extensions.

1.2 More deterministic data sharing and limitations of statistical multiplexing with VLANs and priorities

Over time QoS enhancements have been added to Ethernet. For example IEEE802.1Q has been introduced to include priorities and VLANs, to reduce broadcast traffic in the network and improve performance for high priority streams. However the network bandwidth is still shared among functions by statistical multiplexing. With best effort traffic, incoming packets will be stored into the switch buffer and forwarded whenever the outgoing port is freed. If the volume of messages exceeds the buffer size, packets will be dropped and lost, as a symptom of so called network congestion. Network congestions happen in case of excessive traffic which exceeds the memory size requirements in the buffer memory of Ethernet network switches.

While the VLAN network can be designed to be “more” deterministic, there are no absolute guarantees and temporal boundaries.

If the network:

- implements a number of VLANs and the bandwidth use on highly utilized Ethernet links is low (<<10%),
- and if the applications use only a handful of high priority data flows (typically one per sending ingress port connected to the network with i.e. one frame at 50x/second)
- the number of hops (switches/routers between two end-stations) is limited (1-2, more in special cases with very few data flows)
- consist only of end-stations sending packets periodically and not exceeding their specified bandwidth consumption

then the probability is high that there will be no traffic congestions, assuming the traffic profile does not change. However this scenario is not realistic, as there will be faulty or rogue nodes, and the system scalability will mandate many well defined dataflows per port or Ethernet link which cannot be protected by available mechanisms.

1.3 Faults, faults propagation and impact on deterministic operations of VLANs

In current Ethernet standards, there are no guarantees or mechanism which will ensure that in the case of fault or rogue components, the network will not be congested and transmit data within prescribed temporal boundaries. Without such robust and absolute guarantees for latency and jitter, the communication channel can be seen as a “black” channel – a

communication medium without any guarantee of operational capabilities. Therefore the architecture and applications shall be designed to account for any situation in which the communication network becomes unavailable.

While VLANs with priorities add value in simpler applications, they can hardly enable “gray channel” communication and isolation of functions in complex integrated systems.

By traffic profiling (shaping, policing, application-controlled network access, etc.), modifications and additional services, Ethernet networks can improve their QoS and partitioning of critical and non-critical streams. This could lead to a solid “gray” or plain “white” communication channel performance in more complex integrated systems (see Chapter 4.3.1.1).

1.3.1 Prioritized traffic and traffic isolation

VLANs do not protect or reserve the bandwidth for intra-VLAN traffic operation. They just offer logical separations, which in case of high-priority VLAN traffic consumes the bandwidth of other lower priority VLANs and pre-empts their packets. In the case of faulty or babbling node sending high-priority packets, any VLAN in the network can be influenced. Furthermore it is impossible to test a system sufficiently to guarantee that any change or modification will not adversely influence the operation of the system. This is due to the inability of present mechanisms to support traffic isolation.

1.3.2 Packet transmission and switching approach

The frame forwarding approach which determines how messages are transmitted via a switch plays a significant role.

A store-and-forward switch will compare the last field of the datagram against its own frame-check sequence (FCS) calculations, to ensure that the packet is free of bit and data-link errors. Afterwards the switch will transmit the packet stored in its own memory. Cut-through devices will start the forwarding of a received frame before the end of the frame has been received. Therefore, it cannot perform the same validity checks as a store-and-forward switch and will omit FCS checks.

As cut-through switching does not drop invalid packets, the packet with physical- or data-link-layer errors will get forwarded to other segments of the network. The solution in the IT world to this problem is to utilize store-and-forward mode in all switches in the edge of the network to filter out incorrect packets.

Furthermore cut-through makes sense only in linear architectures with master node. This approach is practicable in linear topologies for automation systems. As a downside, in large networks (20-30+ hops), the packet error rates rise significantly, and this can limit the high-integrity communication. In aerospace systems, the number of hops rarely exceeds five hops, as only switched star or small ring-like network sections are architectures that are deployed (see Figure 8).

1.4 Network Capabilities and Limitations for Advanced Integrated Architectures

1.4.1 Assumptions on integrated system capabilities

Advanced integrated architectures integrate and host a large number of distributed functions on common resources. Theoretically any function can support real-time, including hard real-time and soft time operation, periodically (controls, historians, …), aperiodically (controls, IO signals, …) or one-shot messaging (e.g. alarms, safety signals, …).

The system should be configurable to host any function which can be configured to access any sensor or actuator in the system through the network. Ideally the system architect can configure a system as a cloud of embedded resources, which can be tailored into an application specific topology to support industry-specific use cases in fail-safe and fail-operational systems. Furthermore, the modification of functions should not influence any other critical function in the system – this is important for sustainable V&V activities, incremental and modular certification, reuse and maintenance.

1.4.2 Assumptions on network capabilities

The network shall support:

- Different models of computation and communication to host functions with different properties
- Prevent unintended network congestions (or timing delays) and system behaviour
- the capability to provide robust services and embedded virtualization for safety functions in certifiable systems per design (data flow isolation, internal device design/architecture, service history, …)
- different types of redundancy and topologies to enable a sufficient design space and options for system/safety architecture specialist to create robust communication for critical functions
- Sufficient QoS for any function to access other resources with given maximum (bounded) latency and jitter
- Latency and jitter control
- Robust synchronization to support above

1.4.3 Limitations of the existing Ethernet standard

Ethernet with VLANs and prioritized asynchronous communication alone do not permit any of capabilities listed above, unless the network operation is profiled (and constrained) for a special application and topology, using additional mechanisms at application level or some additional modifications at OSI Layer 2-6. Such network can be scaled for a limited set of applications and limited number of distributed functions, but any change in requirements can lead to additional complexity or additional hardware or subsystems to handle it.

In general, the scalability, reuse and reconfigurability of system architecture or embedded platforms are not guaranteed, if significant networking and system integration capabilities are missed or simply insufficient for the integration of different types of mixed mission, safety-, and time-critical functions.

As an example, the missing capability for strict determinism (synchronous communication) will not enable system-level time partitioning. This keeps control function closer to sensor and actuators, on a specific computing unit. As the function cannot be hosted anywhere in the system, this will influence the reconfigurability and architecture scalability.

Chapter 2 Ethernet for Critical Systems

2.1 Deterministic Ethernet Standards Relevant for Advanced Integrated Systems

2.1.1 Ethernet Standards

2.1.1.1 Overview

The following figure shows different Ethernet networking standards with their target applications (industry niches) and use cases (application criticality). Specific network capabilities are added by modifications or enhancements at layers 3-7. The capabilities which support the integration of different functions and traffic isolation for critical applications, are defined at Layer 2. At this level it is possible to support detailed HW-based control of data flows on every port. The following figure shows the ecosystem of Ethernet standards and protocols which are used in different applications.

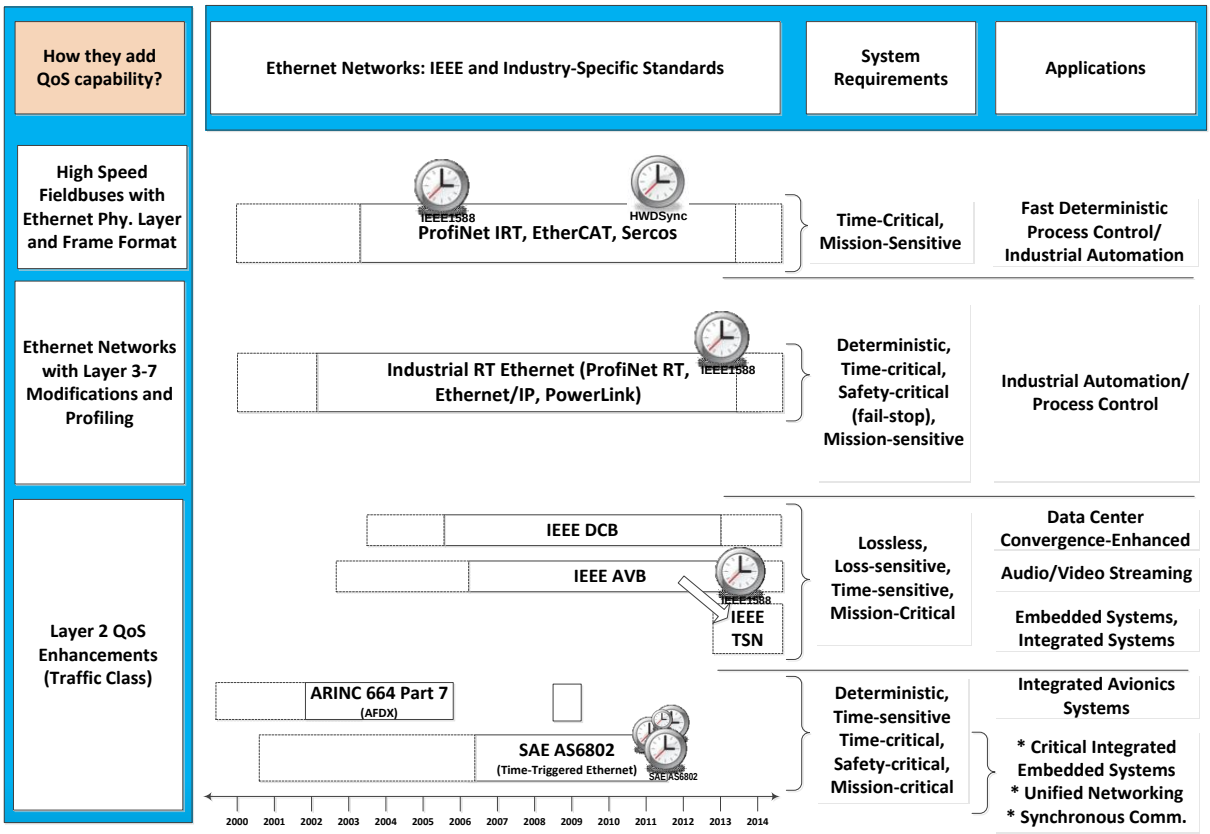


Figure 1: Ethernet variants and protocols addressing different system requirements and application domains

Obviously it is possible to design real-time systems using presented technologies, but very few of their capabilities can be used for design of scalable Ethernet-based integrated architectures which can host mixed-criticality functions in up-to SIL4-rated systems.

2.1.1.2 ARINC664-2005

2.1.1.2.1 Introduction

The ARINC664 standard and its core technology was developed between 1998-2004 by Airbus, Boeing and key avionics suppliers (e.g. Rockwell Collins).

ARINC664 [2] proposes *"profiled networks", which are entitled to adapt the IEEE 802.3, 802.1D and "IP" (RFC 1122) standards in order to fulfill specific performance or safety needs. For instance, a subset of profiled networks, called "deterministic networks", is defined for those aircraft network domains where quality of service (including timely delivery) is the objective.*

The idea was to avoid traffic congestion and prevent the overflow of internal switch queues and memory buffer, and minimize frame drops or loss. The frame drop avoidance was not the only driver, as the AFDX network does not guarantee a fully lossless service, but it minimizes the packet loss probability. The deterministic networking approach applied in AFDX requires a computing model which is insensitive to the occasional loss of frames. The primary driver was to keep maximum latency under control to provide deterministic latency.

Avionics Full DupleX Switched Ethernet (AFDX) defines the protocol specifications (IEEE 802.3 and ARINC 664, Part 7) for the exchange of data between Avionics Subsystems. Airbus has patented the key AFDX (Avionics Full Duplex Ethernet) mechanisms, which can be licensed by semiconductor component providers.

Same as standard Ethernet, AFDX networks contain the following components:

- **AFDX End stations**: Network interface card with AFDX interface to the network, and a host CPU interface to the computer node.
- **AFDX Switches:** network devices which forward Ethernet frames to their target destinations.

2.1.1.2.2 Deterministic Data flows in AFDX

2.1.1.2.2.1 Message forwarding and routing

In standard Ethernet layer 2 switches, Ethernet frames are routed to output ports based on the Ethernet MAC destination address. The destination MAC (Media Access Control) address is also used for routing of messages in AFDX switches. The destination MAC address defines the routes for the switch, based on its configuration. The value of the lower two bytes defines a critical deterministic data flow – in aerospace terminology and ARINC standards it is called a Virtual Link ID (VLID). Virtual Links are unidirectional, multi-casting circuits which are sent from one source (end-station) to one (unicast) or multiple destinations (multicast). No broadcast is allowed in AFDX networks.

2.1.1.2.2.2 Mechanisms to support deterministic network and system design

The determinism for a single deterministic data flow is not provided only by defining its data flow parameters, but also by defining the exact configuration for every data flow crossing the path with the data flow under consideration. The flow for which we define latency and jitter shall be compliant with the properties (periodicity, frame length) of all other data flows competing for the bandwidth.

Therefore the maximum latency for critical data flows cannot be calculated in isolation, but must be analysed by calculating the configuration with the respect to all other known data flows. Any dynamic dataflows or additional bursty traffic, not planned and defined at design time would influence the calculated maximum latency and are thus not permitted, and will be discarded by AFDX switches.

2.1.1.2.2.3 Essential mechanism for AFDX determinism

In AFDX networks this means that the following mechanisms are essential:

- The capability to limit the maximum bandwidth use per virtual link (VL) and to create a data flow with defined periodicity by using BAG (bandwidth allocation gap) and maximum BAG jitter parameters
- To calculate maximum latencies for every stream according to their initial configuration data and priorities (*note: it is not possible to define latency in advance – it is a result of total system calculation!*).
- To precisely police performance of the data flow on every incoming port and prevent discrepancies
- To fine-tune latencies by adjusting priorities for egress(outgoing, sending) buffer selection and group the upper latency boundaries, depending on priorities (this will define which dataflow has the highest priority to get out of the outgoing port buffer faster)

2.1.1.2.3 Topology and redundancy

There are two independent switched networks in an AFDX system, the A and B Networks – defined for higher availability. They are fully independent, and network switches do not know anything about redundancy. Only end-stations are aware of traffic via both A and B networks.

Each packet transmitted by an end-station is sent on both networks. Therefore, under normal operation, each end-station will receive two copies of each packet. End-stations identify corresponding packets (replicas) that arrive on the A and B networks over virtual link ID and a sequence number field (0-255) – this sequence field is positioned after the end of the frame application data payload, before the frame checksum. The receiving end-station checks the order of successive frames in the scope of “Integrity Checking.” together with sequence field. The end-station will pass the first redundant packet to the target end-station without delay, and the second packet will be dropped. There are mechanisms to control and monitor redundant timing message skew, as they can arrive with significant delay difference between the network A and B.

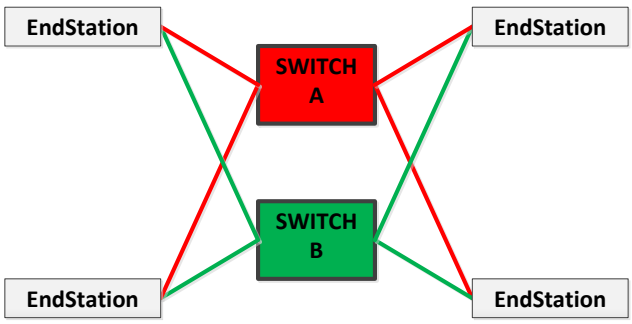


Figure 2: Redundant networks

So the redundancy management is implemented in the end station, which will pass one valid copy of the redundant messages to the application. Therefore the redundancy management is implemented at the communication controller (network interface card), and completely transparent to the application and switching devices.

The sequence order can be less than 255 frames, and can be reset by 0 sequence frame. With system security breaches, rogue end-stations or faults, such mechanism can trigger availability issues (if one faulty channel transmits “correct” frame with zero sequence, and the other channel continues counting, both channels will be down), but with modifications in endstation implementation such issues can be resolved [2]. This example shows how

important is the implementation of the networking device, in addition to communication protocol.

2.1.1.2.4 End-station and API

Avionics subsystems use communications sampling and queueing ports to exchange data. So called “Communication ports” or COMs are defined for access to applications. Queueing ports fill and empty on read/write (buffer size > 1 message), while sampling ports do not shrink or expand – they just contain the last written sampling message for state variables (buffer size = 1 message), and end-stations provide an indication of the message freshness.

2.1.1.3 SAE AS6802-2011

2.1.1.3.1 Introduction

As described in SAE AS6802 standard [3], *“Time-Triggered Ethernet functionality described in the SAE AS6802 standard is a Layer 2 Quality-of-Service (QoS) enhancement for Ethernet networks. It provides the capability for deterministic, synchronous, and congestion-free communication, unaffected by any asynchronous Ethernet traffic load. This occurs via a fault-tolerant, self-stabilizing synchronization strategy, which helps to establish temporal partitioning and ensures isolation of the synchronous time-critical dataflows from other asynchronous Ethernet dataflows. By implementing this standard in network devices (network switches and network interface cards), Ethernet becomes a deterministic network which can be shared by low-latency, low-jitter, and non-time-critical applications. This means that distributed applications with mixed time-criticality requirements (e.g., real-time command and control, audio, video, voice, data) can be integrated and coexist on one Ethernet network.”*

This service represents a time-triggered traffic class for Ethernet networks and can be implemented on asynchronous packet-switching network devices. Every sender node has a transmit schedule, and each Ethernet switch has receive and forward schedule. This traffic is sent over the network with constant communication latency and small and bounded jitter.

2.1.1.3.2 Deterministic Data flows in Time-Triggered Ethernet

2.1.1.3.2.1 Message forwarding and routing in SAE AS6802

In standard Ethernet layer 2 switches, Ethernet frames are routed to output ports based on the Ethernet destination address. As in ARINC664/AFDX standard, the destination MAC (Media Access Control) address is also used for the routing of messages in Ethernet switches with time-triggered traffic.

2.1.1.3.2.2 Mechanisms to support deterministic network and system design

Time Triggered (TT) messages are used for deterministic synchronous Ethernet communication in complex networks. All TT messages are sent over the network at predefined times and take precedence over all other traffic types (e.g. best effort and priority driven rate constrained traffic), based on time progression. This means they message forwarding is not related to message priorities or statistical multiplexing, but relate only to the system time progression. A network switch has a complete overview of time-triggered traffic, arrival and transmission timing. So it has enough intelligence to decide how to handle time-triggered and asynchronous traffic, and avoid violations of the network timing for every TT data flow in the network.

Time-Triggered Ethernet creates periodic synchronous data flows or circuits, which are based on robust clock synchronization and time division. Due to the predefined transmission

time of a message, it is possible to reserve the medium and avoid even minimal delays of transmission if this is required for a specific TT message.

The determinism for a single data flow is defined by the period and latency constraints, and system time accuracy. Similar to AFDX/ARINC664 the maximum latency for critical data flows cannot be calculated in isolation, but must be found out by calculating the configuration with the respect to all other known data flows and their bandwidth requirements. During the network design, all those constraints are taken into account and the network scheduling for all TT messages is created to avoid any congestions. The timing properties of data flows are known in advance and any adaptation to one data flow will not influence the timing of all other data flows.

The advantage of TT traffic is that any incremental changes or modifications can be conducted without any change of timing for already integrated functions, assuming there are sufficient resources available in the system.

2.1.1.3.2.3 Essential mechanism for SAE AS6802 determinism

In TTEthernet networks this means that the following mechanisms are essential:

- The capability to limit the maximum bandwidth use per virtual link (VL) and create a data flow with defined periodicity
- To calculate fixed latencies for every stream according to their initial configuration data
- To precisely police performance of the data flow timing with the respect to system time on every incoming port and prevent congestions and packet drops

2.1.1.3.3 Mixed Time-Criticality Traffic

SAE AS6802 traffic class can operate with other traffic classes such as:

- **Periodic rate-constrained** (RC) traffic – is sent with a bounded latency and jitter ensuring lossless communication. Each sender node gets a reserved bandwidth for transmitting messages with the RC traffic. No clock synchronization is required for RC message exchange. This type of traffic is covered by AFDX standard.
- **Best-effort** (BE) traffic – traffic with no timing guarantees. It is compatible with the IEEE 802.3 standard.

Time-triggered frame format is fully compatible with the standard Ethernet (IEEE 802.3) frame format, and operates at the OSI model Layer 2. It allows the usage of existing layer 3 and upper layer protocols. Messages from higher layer protocols, like IPv4/v6 or UDP, can be sent in a time-triggered way without modification of the message content itself.

2.1.1.3.4 Global (System) Time and Synchronization

The notion of global time in conjunction with scheduled transmission is used to implement a fault isolation (temporal firewall) mechanism. This mechanism prevents that a faulty device affects the network operation of other devices. Based on the global (system) time, an Ethernet switch with AS6802 can block the traffic generated by faulty components, and prevent untimely messages to disrupt the determinism of critical traffic flows.

Clock synchronization among all participants is crucial for the transmission of TT messages. TTEthernet components transmit clock synchronization messages to keep the clocks of the end stations and switches in synchronization. For this purpose TT traffic relies

on a redundant master-slave method that has a distributed fault-tolerant majority of master nodes and master switches to provide the time in the system. This synchronization approach can be combined with other mechanisms such as IEEE 1588. SAE AS6802 takes a two-step approach to synchronization: In the first step, the synchronization masters send protocol control frames to the compression masters. The compression masters then calculate an averaging value from the relative arrival times of these protocol control frames and send out a new protocol control frame in a second step. This new protocol control frame is then also sent to synchronization clients. The decision on which devices are configured as synchronization masters, synchronization clients, and compression masters arises from the requirements on the system architecture.

2.1.1.3.5 Fault containment

In Ethernet networks with AS6802 and using TT traffic, we assume each component to be a fault-containment unit. This means that a fault will not propagate directly from one device to another one. However, a fault in one device may manifest in an error state and ultimately result in a failure of a network device. This failure may then become visible as faulty or missing Ethernet frames on the interface from the faulty device to the network.

To tolerate faulty Ethernet frames, Ethernet switches with AS6802 specify two ways to construct error-containment units: the central guardian and the high-integrity design. A third type of error-containment is based on triple-modular redundancy.

Some fault-isolation mechanisms of AS6802 can be considered as implicit security mechanisms, as denial-of-service and masquerading attacks are mitigated by means of fault isolation capabilities of the protocol.

2.1.1.3.6 Topology, redundancy, end-station and API

Time-triggered Ethernet uses equivalent principles as ARINC664/AFDX (see 2.1.1.2.3 and 2.1.1.2.4) to design a distributed embedded platform. The high-integrity end-station design counts on precise timing definition, so the message generation field/counter as in AFDX, is not really mandatory for network design, but it can be considered useful.

2.1.1.4 IEC 62439-3-4/5 PRP/HSR

IEC 62439-3 [4] specifies two redundancy protocols based on the duplication of the LAN, designed to provide seamless recovery in case of single failure of an inter-switch link or switch in the network. Critical applications may require much faster recovery on link or path faults and recovery periods of Nx100ms or seconds may be too long for fast processes.

This set of standards is relevant for redundancy management in general switched Ethernet networks (PRP - Parallel Redundancy Protocol) and their linear/circular topology variants (HSR – High-Availability Seamless Redundancy).

The objective is to send redundant frames over two independent paths, and allowing the receiving end station to decide how to handle incoming redundant messages. The sequence number field is attached into the message. Same as in ARINC664, PRP will accept the first available message and discard the second message [5].

The two networks are assumed to be fail-independent. The destination node will always receive at least one packet as long as either one of the two networks is operational. This provides zero-time recovery in case of a single failure, so no frames are lost.

HSR uses an equivalent mechanisms but in a daisy chained ring, by sending frames in opposite directions. HSR adds a new forwarding mechanism to the switch, which behaves as a simple end station.

As Layer 2 mechanisms, PRP/HSR can “cooperate” with other mechanisms used in proprietary Ethernet solutions. However it does not represent a holistic solution for the design of scalable integrated systems with hard RT performance.

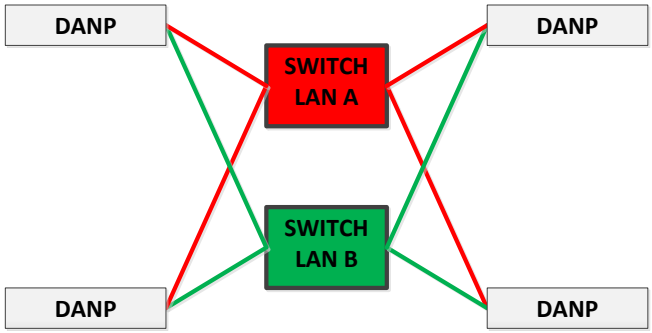


Figure 3: PRP network with DANP (doubly attached nodes)

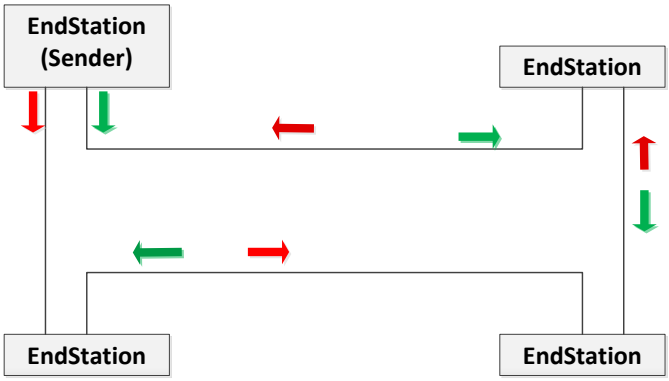


Figure 4: HSR redundant ring

As a conclusion, ARINC664 and PRP provide similar capabilities for redundancy management. The only difference is the position of the sequence counter in the frame. Both PRP and HSR mechanisms do not have any impact on isolation or temporal boundaries in the system, and represent only one of mechanisms for designing highly –available networks.

2.1.1.5 IEEE AVB-2012

2.1.1.5.1 Introduction

IEEE AVB is a set of protocol services which solve the challenge of periodic communication for audio/video applications (AVB = Audio/Video Bridging). It supports smooth traffic shaping on every switch, and allows up to 25% of best effort traffic, with bandwidth reservation. IEEE AVB includes the synchronization of applications via IEEE 802.1AS. This standard also supports stream reservation protocols suitable for data producer and subscribers within the network.

In simple linear architectures with relatively few data flows, AVB network can perform better than standard VLAN network architectures. AVB as a set of network protocols relies on traffic shaping inside switches to provide generic temporal boundaries on latency, prevent stream distortion and microbursts. AVB can support 2ms latency with few controlled data flows and Nx10 A/V channels per port over 7 hops.

The analysis on AVB latency provided in [6] is based on a simplified analytical model of AVB Ethernet switch without technology latency and occasional jitter in end-station transmissions, but it reveals key mechanisms and considerations used in design of IEEE AVB.

This analysis claims that the end-to-end latency/delay can be varied effectively by changing link utilization level and shaping period, but AVB focuses solely on two classes of traffic (high-priority A and low-priority B) with different periods of 125 and 250µs, which limits the maximum number of channels and data flows.

The benefit of end-station synchronization helps to avoid congestions and reduce the switch buffer memory resource requirements. By scheduling packet transmissions much better control of network bandwidth use can be accomplished. Similar to other asynchronous packet-switched network, the number of data flows supported will depend on the link bandwidth, number of data flows, and the number of channels per stream, and will be limited by the configuration tool calculation capabilities.

2.1.1.5.2 Limitations

The Ethernet network devices with this standard do not support an ingress policing for AVB traffic. Only a limited temporal control and monitoring of packets can be configured via coarse port-based policing mechanisms, and traffic policing as we know from ARINC664 and SAE AS6802. Hence, a frame can be sent out on false ports and therefore unintentionally transmitted to multiple devices.

The synchronization in AVB works well if nothing goes wrong, but it is difficult to analyze complex fault scenarios in the case of synchronization faults or best-master search. The predictable initiation of synchronization on network startup can complicate the design of robust integrated systems.

This standard is relevant for non-critical applications, but it does not support any maximum latency monitoring mechanisms for complex architectures and mixed traffic, and does not offer a communication fault containment barrier. Furthermore even if a frame contains multiple channels in one larger frame (e.g. Nx100), only a limited set of maximum frames/data flows [7] can be sent over one link and cross with other data flows over the same link. This limits or prevents its use in complex integrated architectures with hundreds or thousands of data flows, but can be useful for a system with very few data producers (publishers).

IEEE AVB offers benefits against VLANs, in bandwidth reservation, for specific use cases with a limited number of data flows (within a period only a small number of data flows can be transmitted), and therefore it is limited in terms of scalable operation and topology for predictable integration.

2.1.2 Emerging Ethernet Standards and Related Developments

2.1.2.1 IEEE AVB/TSN (2012-2018/2020)

New emerging deterministic Ethernet capabilities enable time-multiplexed bandwidth sharing and a number of services defined for embedded system applications. The IEEE TSN WG will define a number of amendments to IEEE802.1Q and 802.3, which will be later adopted as a part of 802.1Q, together with VLANs. Most probably this set of standards will be completed before 2020, with essential partial capabilities implemented before this date.

The key players behind the IEEE TSN standardization include IT/networking, telecom, automotive, and industrial OEMs (Original Equipment Manufacturers), as well as all major semiconductor and Ethernet network switching companies. Therefore it can be safely assumed that such industry support for this standard will lead to a broad availability of components from different suppliers and affordable pricing for networking solutions.

2.1.2.1.1 Scheduled Traffic and Time-Aware Queueing (802.1Qbv)

At the core of TSN is a time-triggered scheduling principle. In TSN this concept is known as the “time-aware shaper” (TAS), which deterministically schedules traffic in queues through switched networks. The principle operation is depicted below (Fig. 2).

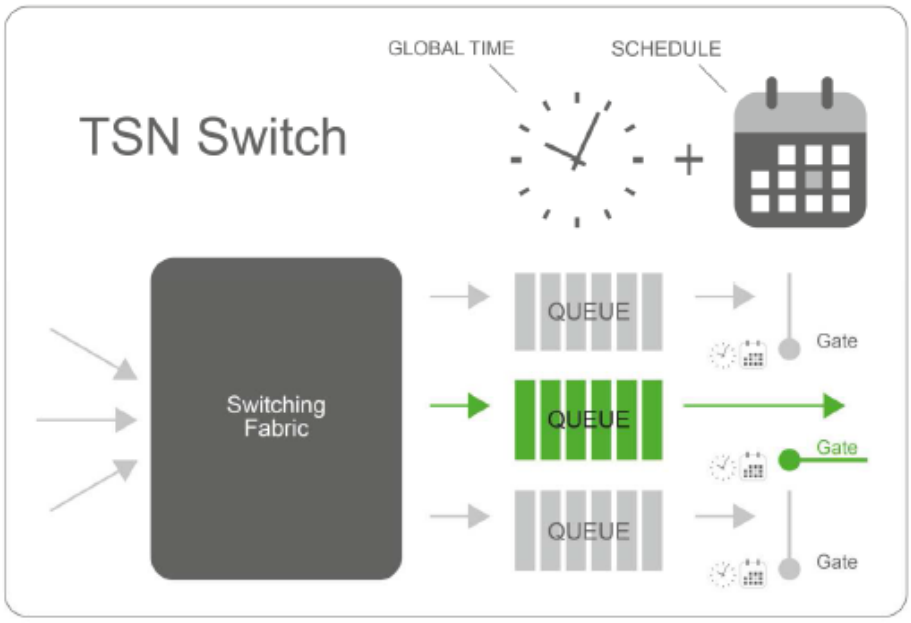


Figure 5: 802.1Qbv time-aware queuing and scheduling

With the time-aware shaper concept it is possible to control the flow of queued traffic from a TSN-enabled switch. Ethernet frames are identified and assigned to queues based on the priority field of the VLAN tag. Each queue is defined within a schedule, and the transmission of messages in these queues is then executed at the egress ports during the scheduled time windows. Other queues will typically be blocked from transmission during these time windows, therefore removing the chance of scheduled traffic being impeded by non-scheduled traffic. This means that the delay through each switch is deterministic and that message latency through a network of TSN-enabled components can be guaranteed.

2.1.2.1.2 Timing and Synchronization (802.1ASrev)

Clock synchronization is a vital mechanism for achieving deterministic communication with bounded message latency in TSN. A robust mechanism for providing global time lays the foundation for the scheduling of traffic queues through each participating network component. The IEEE 802.1ASrev project is working to create a profile of the IEEE 1588 PTP synchronization protocol for TSN. This profile will enable clock synchronization compatibility between different TSN devices, and eventually become a profile of IEEE 1588.

2.1.2.1.3 Frame Replication and Elimination for Reliability (802.1CB)

The IEEE 802.1CB standard implements a redundancy management mechanism similar to the approaches known from HSR (High-availability Seamless Redundancy – IEC 62439-3 Clause 5) and PRP (Parallel Redundancy Protocol – IEC 62439-3 Clause 4). In order to increase availability, redundant copies of the same messages are communicated in parallel over disjoint paths through the network. This feature has similarities with AFDX integrity checking.

2.1.2.1.4 Per-Stream Filtering and Policing (802.1Qci)

Per stream filtering and policing prevents adverse effects on system communication performance, as a result of faulty end-stations which would otherwise violate the engineered bandwidth use. This fine-grained policing capability allows to better control different data flows in complex systems.

2.1.2.1.5 Traffic Preemption (802.1Qbu and 802.3br)

IEEE 802.1Qbu works together with IEEE 802.3br (Interspersing Express Traffic Task Force) on a standardized pre-emption mechanism. This standard addresses the fact that the TAS described in IEEE 802.1Qbv avoids transmission jitter by blocking lower priority queues (for

the duration of one maximum interfering frame) in advance of the transmission point of the critical frame.

Therefore on links where pre-emption as defined by IEEE 802.1Qbu is supported, the transmission of standard Ethernet or jumbo frames can be interrupted in order to allow the transmission of high-priority frames, and then resumed afterwards without discarding the previously transmitted piece of the interrupted message.

Note: Despite broader discussions on its importance for industrial low-latency systems, this feature can be damaging in design of highly critical applications. It can also lead to complex diagnostics of network failures or new unknown failure modes. It is useful for simpler systems, but too tricky for use in advanced deterministic integrated architectures.

2.1.2.1.6 Stream Reservation Protocol (SRP) Enhancements and Performance Improvements 802.1Qcc

TSN also provides mechanisms to improve existing reservation protocols such as SRP (Stream Reservation Protocol – IEEE 802.1Qat) in order to meet the configuration requirements of industrial and automotive systems, such as timing, bandwidth reservation, frame preemption, synchronization, and redundancy. This standard will enable consistent configuration of Ethernet switches from various vendors. In addition it will support the implementation of central configuration models for dynamic scheduling of TSN networks.

2.1.2.1.7 Path Control and reservation 802.1Qca

This protocol relies on IS-IS and collects topology information from nodes (network discovery), to be able to adapt dynamically on network modifications and failures, and contains the mechanism to specify the path, bandwidth reservation and redundancy for data flows.However, the IS-IS algorithm is complex and in-depth formal analysis for arbitrary topologies have not been carried out partly due to this complexity.

2.1.2.2 Other relevant developments

2.1.2.2.1 SDN / OpenFlow

Software-Defined Networking decouples the network control and forwarding functions. This enables the network control to become directly programmable and the underlying infrastructure to be abstracted for applications and network services.

Utilizing SDN capable switches, different communication protocols and their modifications can be implemented on the same hardware. By using a central software-based controller with system know-how, the network performance can be easily adapted and optimized.

This approach is attractive as it moves the ecosystem power toward users and large operators, and reduces the cost of hardware, which is suddenly more customizable and can be sourced from different vendors. The implementation of network device configuration and features can be fully abstracted from the underlying implementation.

OpenFlow and P4 are key elements/instantiations of the SDN concept. OpenFow was designed to create a common approach for software control planes to remotely control lots of different switches with well-defined functions and fixed protocol behavior. OpenFlow is used for networks built from a set of programmable switches which support OpenFlow interfaces. The data path operations reside on the switch, and high-level routing decisions are controlled by a central controller device – this is a high-performance server which can control packet flow forwarding and packet drops.

The challenge with OpenFlow can be the delayed reaction on incoming packet in cases where central decisions must be taken s (>100µs). Furthermore, the safety certification and fault containment are not discussed yet, and some limitations are identified using COTS OpenFlow v1.3 compliant COTS Ethernet switches [8].

The situation for SDN can change with new approaches for the configuration of switching devices. In 2013, the P4 language was defined to supports programming a switch behaviour

for protocol independent switch architectures. P4 tells the switch what it should do, and how it should process packets. It creates a fully customizable switch behavior.

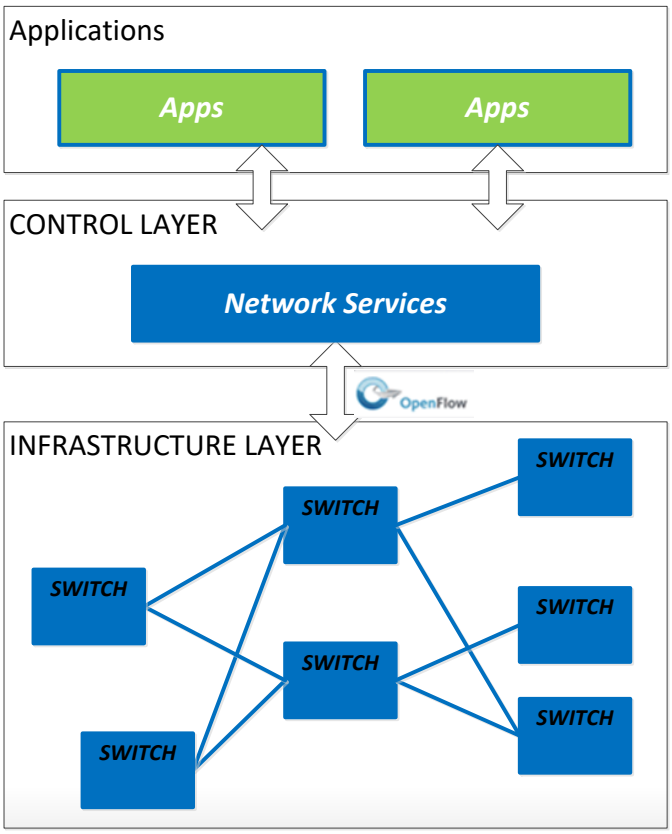


Figure 6: Open Flow-capable switches separate control and data plane

The mix of both OpenFlow and P4 technologies can become a tool in the future which will enable system architects to more easily design a network infrastructure with desired capabilities and custom-made protocols for critical applications.

In the context of data flows, SDN and central control make a lot of sense. However this networking domain and its ecosystem needs to be developed first, and it must be shown that the internal architecture and control plane are not dependent on configuration and programming code.

For critical systems, the technology latency and maximal reaction times are essential, but they could be hard to prove or calculate if defined via software-defined networking. Furthermore, the (formal) verification efforts needed to qualify the technologies for use in safety-critical scenarios are expected to be very high.

2.1.2.2.2 Deterministic Networking (DetNet)

DetNet is a system design philosophy and networking architecture which enables the integration of many asynchronous and synchronous multicast data flows within the complex networked multi-hop system. Data flows can be established via Layer2/3 mechanisms and can coexist on an IP network with best-effort traffic.

It is a step toward defining an integrated architecture with deterministic Ethernet capabilities and predictable latencies for complex networked systems. It describes a network architecture and capabilities which enable fully converged networking.

As presented in [9]:

"Deterministic Networking (DetNet) provides a capability to carry specified unicast or multicast data flows for real-time applications with extremely low data loss rates and bounded latency.
Techniques used include:

1) reserving data plane resources for individual (or aggregated) DetNet flows in some or all of the intermediate nodes (e.g. bridges or routers) along the path of the flow;
2) providing explicit routes for DetNet flows that do not rapidly change with the network topology; and
3) distributing data from DetNet flow packets over time and/or space to ensure delivery of each packet's data' in spite of the loss of a path. The capabilities can be managed by configuration, or by manual or automatic network management.

"

Key mechanisms which are required to implement it in critical systems exist already in mixed SAE AS6802/ARINC664 systems. For general applications such as IoT, real-time Internet or general industrial and process automation IEEE TSN will provide all capabilities to support it.

This work is interesting as a concept which can be tied with broader cross-industry developments. It could be potentially used with SDN networking approaches. For embedded systems with defined temporal boundaries, static network configuration can be more suitable, but it can be uploaded by a central controller, or by the system maintainer.

2.1.2.2.3 Optical Switching and Dense Wavelength Division Modulation (D-WDM)

Optical switching is another approach to bandwidth partitioning over frequency modulation. In fact this approach supports a number of physically separated networks on the same infrastructure. Avionics applications have been researched by DARPA and aircraft OEMs and used in telecom, defense and aerospace systems. Dense WDM enable the integration of 8+ wavelengths over one fibre. The DWDM de-multiplexer tunes into a specific carrier wavelength, removes the carrier wavelength, and presents the transmitted data frames to the end station or applications.

In fact, every wavelength represents a virtual network as a protocol-independent tunnel, which can transfer any network protocol or data format, including Ethernet.

The deployment of optical WDM network depends on [10] the development of networking architecture infrastructure, optoelectronics miniaturization for harsh environments, bend-insensitive optical fiber and optical fiber connectors for target environments (salt, fog, vibration, acceleration, temperature, humidity, …) and long-term reconfigurability.

The key challenge of DWDM technology is to design affordable fiber optic connectors which can be in operation for decades in harsh environments. Today, the market for rugged DWDM components is limited and the ecosystem does not provides reliable low cost solutions for harsh environments. There are solutions provided by TE Connectivity for Coarse WDM with upto 18 wavelengths, but it seems that the major focus of suppliers is on datacentre markets [11] with the growing bandwidth 25-100Gps capacity – but in environmentally controlled environments. As an example at the lower end, Cisco offers DWDM ports (Cisco, 2016) for enterprise applications with 48 channels for 10GBE, operating at 0-70°C.

2.2 Ethernet Networking for Scalable and Reconfigurable Integrated Systems

2.2.1 Basic Ideas and Concepts for Integrated Architectures

Future generic integrated embedded platforms shall host applications with time-, safety- and mission-critical functions deployed on a set of virtualized computing and networking resources in line with ongoing IMA 2G development efforts from the aerospace domain. As an objective, a set of distributed functions shall be hosted on many computers. This approach fully decouples the locality and spatial proximity requirements in control systems (and the hierarchies), and enables the design of flat architectures.

As example two European projects have further developed those concepts for European aerospace industry – SCARLETT [13] and ASHLEY [14].

Such integrated systems aim to support reconfiguration, higher availability and extended maintenance intervals. As a proof of their scalability and full reconfigurability, such systems can also host non-critical functions and support different models of computation and communication. The primary focus of those projects is on ARIN664-based standards and workarounds required to satisfy European aerospace industry requirements.

Similar challenges and generic solutions are relevant to generic open embedded architectures such as reconfigurable next-generation Integrated Modular Avionics (IMA) architectures, vetronics, and specific classes of real-time Internet-of-Things (IoT) systems for critical infrastructure applications. But there are also few subtle differences in the embedded platform design approaches and limitations for all application domains listed above.

Ideally the designer would have all the capability to define a set of resources as a large distributed embedded computer (or embedded cloud), which is so virtualized that all hosted functions have all required computing and networking resources required for their operation, without congestions and resource starvation. The functions hosted on such distributed computer do not need to know about their placing in the system. This would completely decouple software control functions from controlled sensors and actuators, and supporting integration of distributed hard RT controls. In this case, the common set of mechanisms would be more similar to fully scalable and reconfigurable embedded cloud infrastructure.

2.2.1.1 Scalability vs. Proprietary Solutions

2.2.1.2 Problem statement

Typically, an integrated solution is defined with some physical constraints in mind: system size and the number of integrated nodes, bandwidth/QoS requirements, functional domains and hierarchies, safety considerations, etc. Therefore, a system architect with application specific background and understanding of a set of deployed technologies can immediately work on architectural concepts which would satisfy a specific problem.

The system architect cannot create a generic solution for any problem, so the modifications to a problem or system configuration may draw changes to the solution, with additional design, integration, verification and maintenance effort. In addition, if the used set of basic technologies and methodologies changes or requires updates, their cost will be also spread over on the smaller number of systems, and they will be much more expensive.

The objective is to use a set of open technologies with properties which do not change frequently, and have the capability to reconfigure system performance and behaviour at will, by using a small set of computing and networking components.

Today the most promising concepts for generic architecture exist in cloud and fog computing, but they have very limited support for mixed criticality design and real-time functions.

The challenge with any distributed embedded platforms solution which cannot support static or dynamic embedded cloud virtualization, is that it cannot be easily adapted and reconfigured for different use cases, due to a high number of constraints with the respect to topology, bandwidth use, determinism (bounded latency and jitter) and traffic class compatibility.

2.2.1.2.1 Comparing to railway-specific architecture constraints

In railway systems and TCMS, the situation is relatively simplified:

- Train consists have internal hard-wired topology which can change over time for non-critical functions, but will stay unchanged over the 30 years period for critical functions
- Inter-consist (ETB) communication will include critical and non-critical functions
    - ETB contains two redundant lines and by-pass
- Intra-consist (ECB) communication will include critical and non-critical functions

To ensure compatibility among different car makers, the ETB level interfacing should be fully defined. The underlying ECN architecture can be optimized as long as it satisfies system-level safety, security, performance and other embedded platform requirements.

2.2.1.2.2 Safe2Rail / SAFE4RAIL Problem Statement

One of the Safe4RAIL aims is to provide a reconfigurable embedded platform that supports modularity and thus can specify all interface characteristics in temporal domain and logical domain. This will provide assurance of the fitness of the technical concept to support the *modular certification of distributed integrated modular railway architectures* according to EN 50129 or a new electronic system safety standard. This can be accomplished by robust virtualization for mixed criticality functions, while ensuring maximum independence between the TCMS (sub-) system applications hosted on the same generic platform in a dependable and safe way. The platform shall support full isolation of functions hosted on a common computing and networking infrastructure.

Therefore one of key objectives must be to establish a robust “white” (or at least “gray”) communication channel, to be able to handle all those issues and enable system-level time partitioning

2.2.2 Required system integration capabilities

The challenge with any industry-specific Ethernet network, which cannot support embedded cloud virtualization for mixed criticality applications, is that it cannot be easily adapted and reconfigured for different use cases, due to a high number of constraints with the respect to topology, bandwidth use, determinism (bounded latency and jitter) and traffic class compatibility.

Some of the common challenges are:

- System integration does not support specific topology, requires too much bandwidth, is incompatible with other types of traffic, or simply cannot support QoS in a given use case
- Different levels of QoS and determinism cannot be hosted on an Ethernet-based system
- Redundancy management does not support the availability requirements
- System integrity is supported in some use cases and does not scale

- Network use by different functions is application driven (Layer 4-7) and does not scale for different use cases

2.2.2.1 Determinism and predictable communication

At system level, deterministism implies that the system behavior (i.e., a sequence of output signals) will be uniquely defined for some sequence of causes (input signals). The determinism guarantees consistent and predictable performance of system operation under different normal and failure operating conditions and allows accurate understanding of system behaviour. This leads to well-defined “white” and “gray” channel communication.

The determinism can be defined in different terms as bounded latency, jitter, and message order in end-to-end communication. Asynchronous Ethernet communication can support types of determinism (1-2) from Table 1; the type 1-3 are viable with synchronous communication. By mixing both asynchronous and synchronous communication, all variants of determinism are viable in one network. Asynchronous communication supports the relaxation of timing constraints, while synchronous communication allows audio/video and hard-real time controls to operate in one system together with less critical functions (e.g., map or vehicle health-monitoring data upload). In addition this approach can support any type of design paradigm and remove technological limitations to system architecture design.

Type	Description	Implementation viable with:
1	Defined maximum latency, but no absolute guarantees	Standard Ethernet, statistical multiplexing under certain assumptions, “more deterministic”
2	As 1), with defined periodicity, max. latency and jitter	Application-specific modifications to Ethernet, statistical multiplexing and bandwidth partitioning with exact network analysis
3	As 2), with message jitter in (sub)µs due to global timebase, latency is fixed constant	Fixed bandwidth, partitioning synchronous TDM-style communication

Figure 7: Determinism Types and Definition

2.2.2.1.1 Determinism 1: Controlling Packet Latency

As with any asynchronous technology, the network bandwidth sharing is based on statistical multiplexing, so with a sufficient bandwidth margin and overprovisioning (i.e. using only a small bandwidth percentage), different traffic loads scenarios can be handled. As a result, the maximum latency is relatively well defined.

2.2.2.1.2 Determinism 2: Controlling Packet Jitter

The packet jitter is the major part of the bounded latency in asynchronous packet-switching communication, and the major jitter driver is the variable delay in outgoing switch buffer. This jitter can be policed with the respect to defined period.

Synchronous communication does not rely on statistical probability of message delivery, but on exactly defined transmission instants relative to common time. The behavior of the system can be defined to follow exact schedules with microsecond jitter and minimize latency. The latency minimization is possible as there is no need for extra margin which is reserved for statistical uncertainties emerging from the lack of synchronization in the system.

2.2.2.1.3 Determinism 3: Controlling Message Order

The message order can be controlled in simpler systems relatively easily via master polling, but this mechanism is not suitable for highly-critical applications.

Message order is tied to the strict determinism and synchronous packet switching, with full alignment between the network and application layer. This means that the application produces data just in time for transmission, at a given rate within a specified repetitive communication cycle. Therefore it cannot happen that messages from multiple sources arrive with different order at destination.

2.2.2.2 Managing and preventing traffic congestion

For critical functions in closed systems, the bandwidth reservation (asynchronous packet switching) and/or frame scheduling (synchronous packet switching) are used for congestion management in complex integrated systems. The objective is to understand the network traffic profile (data length per packet, periodicity, temporal behaviour of functions) and prevent any network device misbehaviour or unexpected performance in relation to the expected and planned traffic profile.

In open systems, additional non-critical functions with unknown network traffic profile may use the remaining computing and networking resources, so that additional switch architecture mechanisms shall be present to support that.

Open systems with integrated functions may require a different setup for security zones, which exchange information via conduits which represent a secure path for the flow of information between zones. The security is enabled by firewalls, secure gateways and virtual private networks (VPNs), which imposes limitations on the system topology and structure.

In order to support the isolation of different functions for safety and security, robust congestion management shall be supported by the correct use of traffic classes and Ethernet protocols at OSI Layer 2 AND the physical network device implementation which supports different safety and security design assurance processes.

Chapter 3 Transportation Industry Solutions for Ethernet Integration

3.1 Aerospace

3.1.1 Networking Standards in Aerospace Industry

Commercial aircraft networks in use are AFDX, CAN, TTP, ARINC429 networks and MIL1553 physical layers. ARINC as an commercial standard body which is part of SAE Industry activities and driven by commercial airline industry, OEMs and 1st Tiers defines:

- ARINC 429 – low speed (100kbit and 12.5kbit) unidirectional databus for transmission of 32 bit words over two wire twisted pairs using bipolar RZ format
- ARINC 664 – switched full duplex Ethernet (100Mbps) profiling for deterministic avionics/integrated architecture applications
- ARINC 825 (CAN) – CAN profiling for aircraft applications
- ARINC629 – Boeing 777 Integrated Architecture Fieldbus @ 2Mbps
- ARINC659 (SafeBUS) – Boeing 777 Integrated Architecture Backbone databus @ 60MBps

SAE Standards also controls the following standards used in commercial, defense and space and automotive applications:

- SAE AS6802 (Time-Triggered Ethernet)
- SAE AS6003 (Time-Triggered Protocol)
- SAE AS15531 (MIL-1553),
- SAE AS5643 (Firewire)
- SAE AS5659 (WDM LAN)
- SAE AS5653A (MIL-1760)
- SAE J1939 (CAN) - vehicle databus recommended practice used for communication and diagnostics among vehicle components for cars and truck industry in USA
- Etc.

All listed avionics databus standards are maintained by SAE (Society of Automotive Engineers) organizations: SAE Standards and ARINC Industry Activities. The only other widely spread TDMA fieldbus in general aviation is ASCB which is managed by GAMA (General Aviation Manufacturer Association).

Low-to-medium speed databuses operating in synchronous mode are used in integrated glass cockpit applications, modular aerospace controls such as flight controls, engine controls, different subsystems, distributed power generation for more electric aircraft, for different commercial and defense systems. With adaptations, a fully integrated and distributed IMA can be designed for smaller aircraft today with ASCB (max. 10Mbit/s) or TTP (max. 20Mbit/s).

SafeBus (ARINC659) became a backbone of the first integrated commercial aircraft architecture in 1990s, but is used only on one commercial aircraft family (Boeing 777) together with ARINC629, and there is no commercial motivation for its deployment in new systems.

Older IEEE802 token-passing buses and rings based on IEEE802.4, IEEE802.5 and FDDI became obsolete by late-1990s due to prevailing switched full duplex Ethernet technology described in IEEE802.1 and IEEE802.3, and they are not relevant today. Furthermore the token passing mechanism can emulate TDMA bandwidth partitioning, but this approach is very sensitive and was never considered robust enough for advanced integrated modular architectures with safety-critical functions.

3.1.1.1 Full-duplex switched Ethernet

The system integration capabilities represented a bottleneck in the design of larger and more complex integrated systems with high bandwidth requirements and tens or hundreds of computing modules. Before the completion of full-duplex switched Ethernet in 2000, and its avionics flavour ARINC664 in 2004, complex integrated architectures were not viable without significant limitations.

Without any mass-market presence of successful synchronous communication networking technology, and with exponentially increasing Ethernet use, there was no real competitor in the high-bandwidth domain to Ethernet. Back in 2005, it seemed that such complex architectures can be defined only by using asynchronous networks with robust partitioning based on statistical multiplexing and careful calculation, configuration and policing of the configured traffic profile in closed systems.

With early 2001, a new class of time-triggered Ethernet networks has been researched, then commercialized since 2005, and standardized in 2011 [3]. This activity has inspired time-aware networking amendments activities in IEEE802.1 since late 2012.

With SAE AS6802, it is possible to define a synchronous traffic class for Ethernet which can operate on top of asynchronous traffic and enable robust isolation of asynchronous and synchronous communication in the network. This approach virtualizes an Ethernet network into two separate virtual entities - one with asynchronous packet switching communication, and the another with synchronous (hard RT) communication capability. Each end-station can use both virtualized networks via synchronous and asynchronous VLs (virtual links), which represent an emulation of unidirectional unicast/multicast connections.

3.1.2 Overview: From Federated to Integrated Architectures

Increased system integration started at a subsystem level in early 1970s with ARINC429 and MIL1553 databuses, but typically one function would have its own network or point-to-point connections.

Integrated architectures emerged in the aerospace domain as a result of size, weight, and power consumption optimization considerations at the system level. They have later led to improvements in maintainability, availability and reuse of common platforms in different aircraft. So a set of separated aircraft systems hosted on private resources, was moved to an integrated architecture and hosted on common computing and networking resources, together with other functions. Aerospace industry has worked on integrated aircraft architectures since early 1980s, but the embedded system and networking technologies could not support such system architectures.

Initial work on integrated systems has been done by Honeywell and Boeing on integrated Boeing 777 architecture in late 1980s. The Boeing 777 architects managed to deploy an integrated architecture with sophisticated (but from today's perspective very limited) means. More than 15 functions are deployed on central computing resource cabinets (AIMS) using a 60Mbit time-driven backbone SAFEbus (ARINC 659), back in 1992. They used a time-triggered architectural (TTA) model with a partitioning RTOS DeOS (DDC-I Embedded Operating System) to align and integrate functions on common resources. As described in [14]: "*Memory is allocated before run time, and only one application partition is given write-access to any given page of memory. Scheduling of processor resources for each application is also done before run time, and is controlled by a set of tables loaded onto each CPM and IOM in the cabinet. This set of tables operates synchronously, and controls application*

scheduling on the CPMs as well as data movement between modules across the SAFEbus™."

Boeing 777 is one of most reliable and safest aircraft in worldwide airline fleets, with scheduled reliability >99% [15], [16] .

3.1.2.1 Ethernet-based integrated modular architectures and technology baseline

New avionics architectures and Complex IMA [17] since 2006 are based on Ethernet are based on the following technology baseline:

- ARINC664 (Avionics Full Duplex Ethernet)
- ARINC653 (Avionics Application Standard Software Interface)

The idea of ARINC 664 was to deploy Ethernet developments and enhance them to be applicable in larger integrated modular architectures (IMA). ARINC 653 specifies space and time partitioning for safety-critical avionics real-time operating systems (RTOS), and allows the hosting of multiple applications of different software levels on the same hardware in the context of an Integrated Modular Avionics architecture. Details on ARINC 653 are provided in in the Safe4RAIL deliverable D2.1 (Report on state-of-the-art of 'functional distribution architecture' frameworks and solutions).

The technology baseline for ARINC 664 supports the L-TTA (Loosely TTA) [19], as no synchronous networking is supported, and the prevalent internal system architecture philosophy of early adopters, such as Airbus has historically focused on asynchronous system design to avoid time generation as a single point of failure. Boeing has adopted similar approach, but they add additional mechanisms for message integrity and communication error detection with timeliness checking.

3.1.3 Evolution of High-Bandwidth Networking for Integrated Systems

3.1.3.1 System Topology

The Airbus 380 IMA [20] consists of 16 AFDX/ARINC664 switches in two networks, Network A (8x) and Network B (8x). The number of end stations and subscribers is 80. They can be remote data concentrators (a PLC with IO and internal processing) or functional units/ECUs or sensors/actuators. A switch is seen as a physical separation/isolation point between functions, and this “domain-based IMA” concept, offers a strict functional separation in addition to network-level bandwidth partitioning mechanisms.

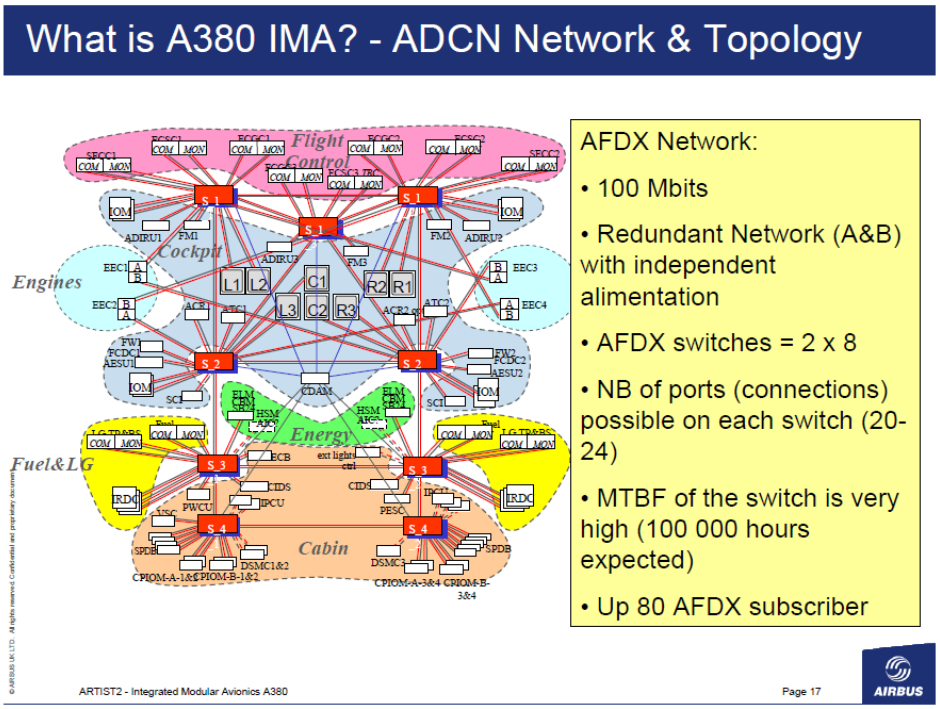


Figure 8: Airbus 380 IMA Architecture [17]

An essential component in this architecture is a CPIOM (computing and IO resource), which can be compared to an industrial PLC with processing and IO capability. The CPIOM supports strong partitioning, BIST and fault monitoring for certifiable systems.

The Boeing 787 approach is slightly different insofar that it represents a more generic variant architecture variant which does not have switches as additional “firewall” between functions. Functions are not separated into domains, but reside in CCR (common comuting resources) and can integrate with low-level functions hosted on RDC (Remote data concentrator) units. The network can be therefore more heavily loaded due to higher utilization of the backbone among different functions.

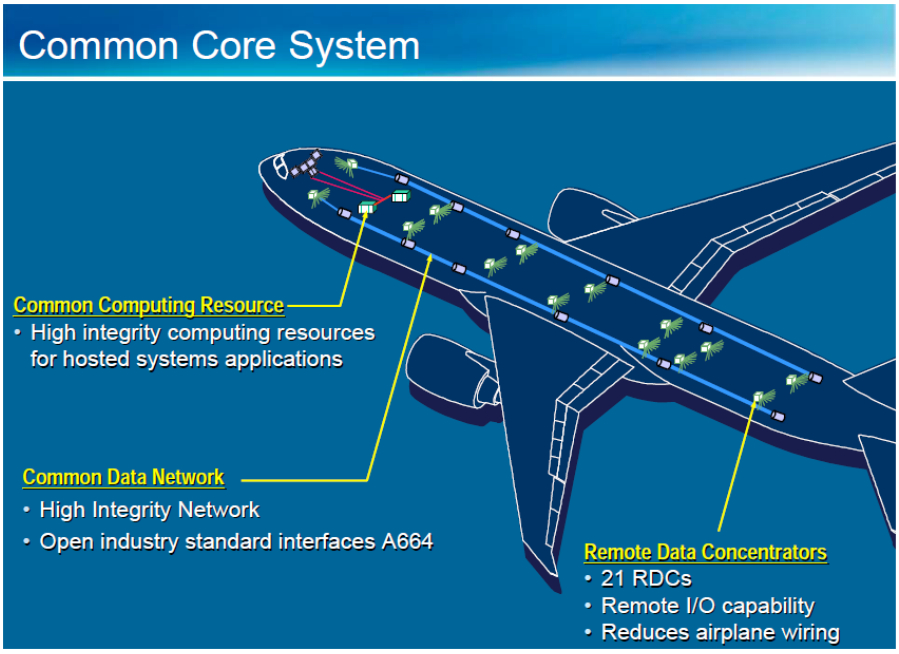


Figure 9: Boeing 787 IMA – High-level Overview

3.1.4 Integration of Software Platform (ARINC653) and Network

Both IMA architectures deploy ARINC653 API and add proprietary services and mechanisms at application/middleware layer. By establishing a dedicated link via the ARINC664 network, with integration via sampling and queueing ports in ARINC653, it is possible to establish deterministic inter-task communication on one computer or in a distributed system or among several computers, with full software abstraction from system architecture and topology details.

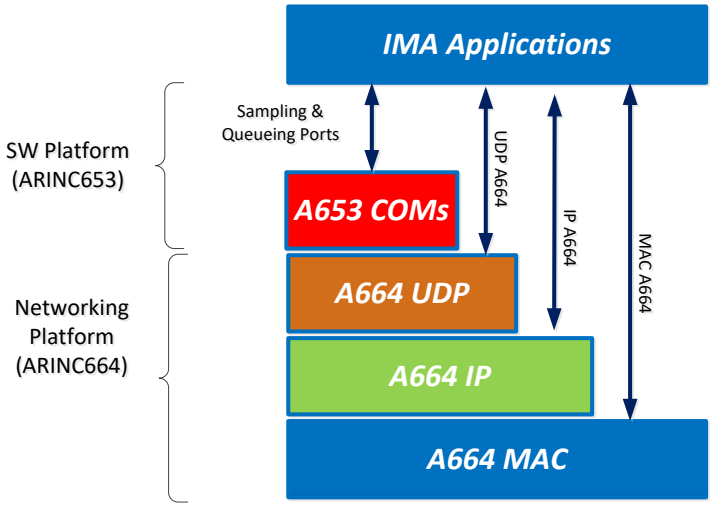


Figure 10: Integration of ARINC653 and ARINC664 data flows

3.1.5 Future Outlook on System Architectures and System Integration

New Ethernet architectures in commercial aviation are going to utilize ARINC664 for a long time, due to high capital, ecosystem, V&V, technology and embedded platform investments. Furthermore, the architectures and platforms patents are tied to the selected baseline

technologies - ARINC664 an ARINC653. For the aerospace industry, the next developments will focus on adding incremental improvements, higher bandwidth networks and integrate the latest technological advances in semiconductor technology.

Airbus has developed its own lightweight AFDX variant which is designed for faster real-time subsystems, and it can support up to 8 end-stations at lower frame transmission periods (0.5ms). This variant relies on much simpler devices and utilizes a very simple approach to the bandwidth sharing. This helps to further integrate subsystems within the IMA perimeter.

SAE AS6802 can be added as a network service to the existing systems and expand IMA-architecture scalability and capabilities. Other smaller OEMs and newcomers cannot simply copy the architecture from early entrants (patents!), so there could be a space for faster introduction of cross-industry technologies, assuming those standards support the criticality and safety design assurance relevant for high integrity IMA in aerospace (up to DAL A).

3.2 Automotive

3.2.1 Current Status and Standards

Depending on the application domain, different network technologies are used in today's automotive E/E-Architectures.

The LIN bus (Local Interconnected Network, standardized in ISO 17987) is used for the inexpensive integration of sensors and actuators in vehicle networks. The CAN bus (Controller Area Network), standardizes in ISO 11898, enables the networking of a large number of ECUs. As of today, it is still the prevailing vehicle Bus system due to its low cost for the relative high bandwidth provided. The wide adoption of the technology, together with the accompanying spreading of corresponding tools and Knowledge will assure that the technology will stay relevant in automotive E/E-Architectures for some time to come. With CAN FD (Flexible Data Rate), standardized in ISO 11898-7, data rates up to 8Mbit/s are supported.

The FlexRay technology (standardized in ISO 17458) provides data rates up to 10Mbit/s. This databus operates on a time cycle which is divided into a static segment and a dynamic segment. In the static segment, communication time is preallocated, providing the means for strong real-time guarantees. The dynamic segment caters for event-driven communication, since it operates similar to CAN.

With the development of the BroadR-Reach technology which enables Ethernet/IP communication with data rates of 100Mbit/s over unshielded single twisted pair cable, while still meeting the automotive EMV requirements, Ethernet has become an important communication technology in current and future automotive E/E-Architectures. BroadR-Reach has been standardized as 100BASE-T1 in IEEE 802.3bw-2015 Clause 96. Just recently, on 30 June 2016, work on IEEE P802.3bp 1000BASE-T1 PHY completed, which defines Gigabit Ethernet over a single twisted pair for automotive and industrial applications.

An important organisation regarding Ethernet in the automotive domain is the OPEN Alliance (One-Pair Ether-Net) Special Interest Group (SIG) (www.opensig.org), which is a non-profit, open industry alliance of mainly automotive industry and technology providers. The members of the group collaborate to encourage wide scale adoption of Ethernet-based networks as the standard in automotive networking applications.

On layers 3 and above, usually the standard communication protocols, i.e. IP, UDP and TCP as defined in the corresponding IEEE RFCs, for communication over Ethernet are used. There are however some automotive specific protocols in use, namely DoIP (Diagnostic over IP) and SOME/IP (Scalable service-Oriented Middleware over IP).

As the name suggests, DoIP is a protocol for diagnostic communication over an IP network. It is standardized in ISO 13400.

SOME/IP is part of the AUTOSAR specification (since AUTOSAR 4.0). It supports certain middleware features and was specifically designed to fulfil automotive requirements. Among the features supported by SOME/IP are Service Discovery, Remote Procedure Call and Publish/Subscribe communication pattern.

3.2.2 Evolution of High-Bandwidth Networking in Integrated Automotive Systems

Today, modern driver assistance systems and infotainment systems are the main drivers for high-bandwidth networking in automotive systems. But in 2008, the main driver behind the introduction of Ethernet as an automotive network was the significant reduction in time, needed to reprogram an ECU when using Ethernet, compared to using CAN.

For Example, BMW has been using Ethernet to reprogram the calibration software for their engine control units since 2008. According to [21], in the 4th generation BMW 7 series, to upload 81 MB via CAN 10 approximately 10 hours were required. In the 5th generation BMW 7 series, to upload 1 GB via Ethernet only approximately 20 minutes were required. In this use case however, there was no Ethernet communication within the car, i.e. between two or more ECUs.

The first implementation of Ethernet within a series car (beyond diagnosis/update) was done by BMW (which is the major driving force for pushing Ethernet into Automotive) in the BMW i5. Figure 11 shows the architecture of a surround view system (SVS) as it was presented by BMW on the 2nd Ethernet &IP@Automotive Technology Day in 2012 [22]. The system uses four cameras connected via Open Alliance BroadR-Reach (OABR), which was the current name of Ethernet over a single twisted pair at that time, to on central ECU where the data by the cameras is processed.

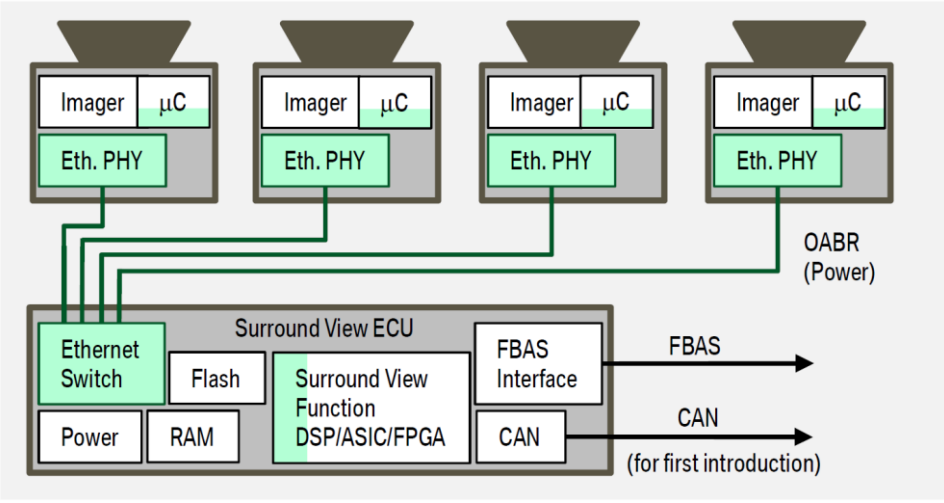


Figure 11: Digital system with OABR (series 2013) [22].

In 2015, BMW introduced Ethernet (100BASE-T1) as a system bus for infotainment and driver assist domains in the new 7-series. The target architecture as presented by BMW is depicted in Figure 12.

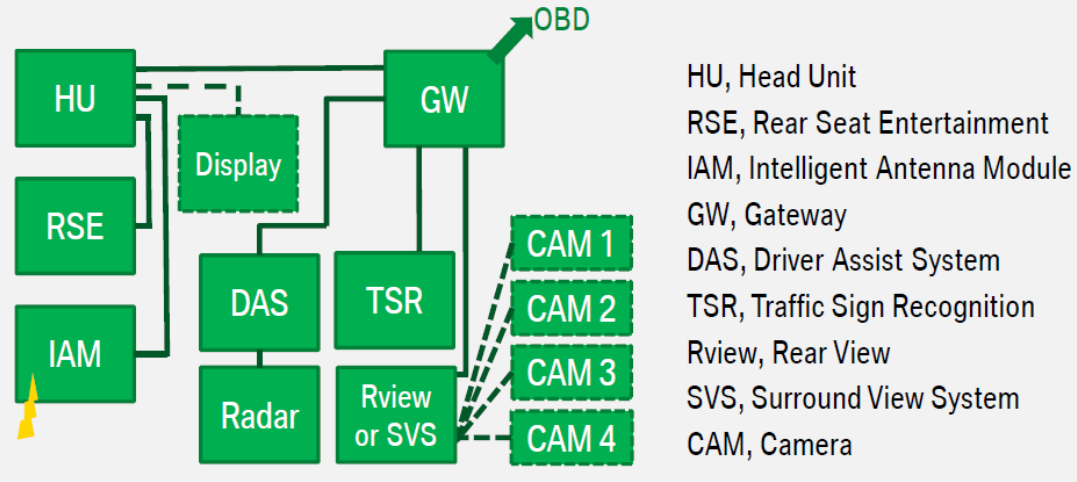


Figure 12: 100BASE-T1 as system Bus in BMW 7-series in 2015 [23]

In this new architecture, the previous SVS from 2013 is a small part of a larger domain, communicating over Ethernet. Obviously, several switches are required to realise this architecture, which introduces multi-hop communication and, depending on the network configuration also routing, e.g. when the picture from one of the cameras should be displayed on the display connected to the head unit. In contrast to the rather simple network shown in Figure 11 network in the Ethernet system bus, network configuration becomes much more important, due to the fact that several of the links are used by several different communicating functions. While service oriented communication with SOME/IP can facilitate the process of network configuration it can also make it more complex to predict the communication behaviour, especially regarding the network timing, e.g. message latencies.

According to BMW and other OEMs, the next step in the evaluation of automotive Ethernet architectures is the introduction of an Ethernet backbone which interconnects the different domains in-car domains with a high speed (at least 1 GB/s) Ethernet connection. The domains in the discussed architectures may use Ethernet for communication or other, more traditional communication technologies like CAN.

3.2.2.1 System Topology

With the introduction of Ethernet as a system bus, the technology has a large impact on the resulting system topologies of automotive E/E-Architectures, at least for the part where Ethernet is used. As can be seen in Figure 12, the architecture of the depicted driver assist and infotainment domain comprises a mix of star topology and daisy chains. While such topologies can increase the complexity when it comes designing the communication, but the possibility to implement such topologies increase the overall flexibility when designing the whole architecture.

As indicated in the previous chapter, in the near future automotive E/E-Architectures will be built around an Ethernet backbone. Figure 13 shows an E/E-architecture built around an Ethernet backbone as envisioned by NXP. Similar future E/E-Architectures are shown by several OEMs.

Besides interconnecting the different in-car domains with high-speed Ethernet, another aspect of these future architectures is the trend to more centralization. This is represented by using so called domain-controllers which are ECUs with a relatively high computing performance (at least for automotive). Additionally, depending on the domain, the ECUs can bring special purpose hardware, e.g. GPUs. This allows all the processing-intensive tasks on these ECUs, on which even virtualisation techniques can be employed to provide different computing environments for different applications. In exchange, some of the other ECUs might become obsolete or are turned into I/O boxes with only the most basic processing capabilities.

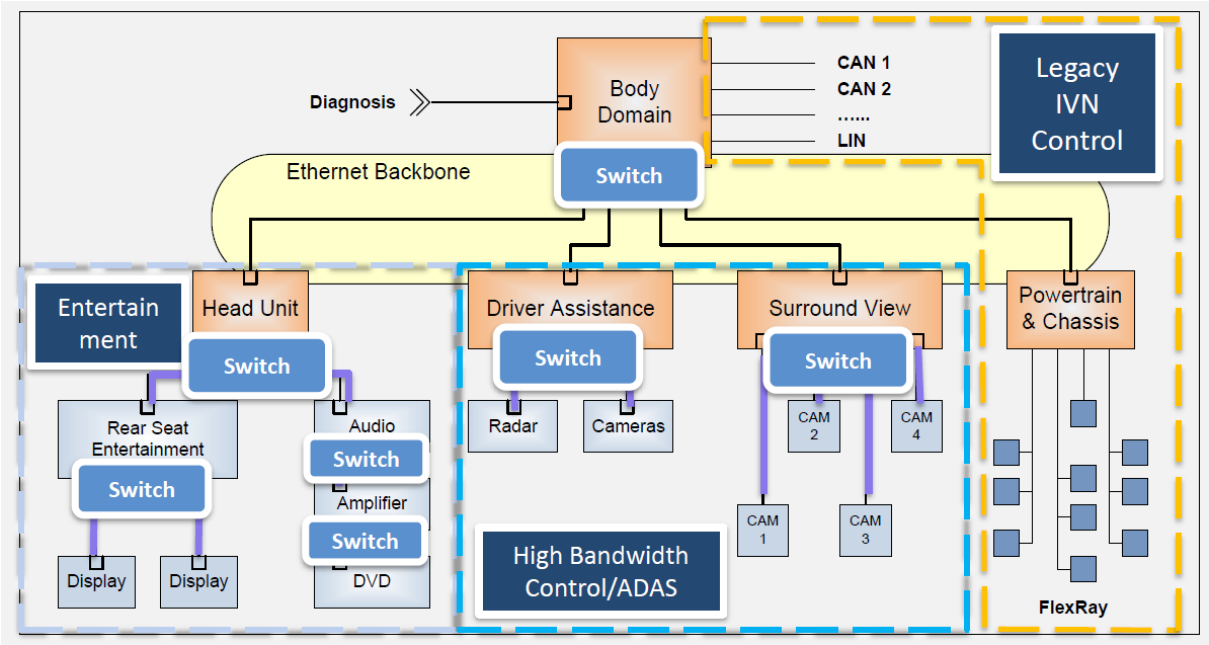


Figure 13: Architecture with Ethernet backbone envisioned by NXP [23]

The traditional bus systems, especially CAN and LIN will be still a part of automotive E/E-Architectures for quite some years, due to low cost and proven service history.

3.2.2.2 Hosting of Critical Functions

With the introduction of Ethernet and the trend towards centralized ECUs with high computing resources, different techniques are used with respect to safety critical functions.

To compensate failures in the communication, the mechanisms provided by standard protocols, e.g. TCP/IP (RFC 793), are used for functions which are not time-critical, but have strict requirements regarding the completeness of the communicated data.

For time-critical functions, where TCP/IP may not be suitable, OEM-specific techniques on application layer are implemented to provide end-to-end protection against frame loss. These are usually comprised of a message counter and some timeout mechanism which communicate in different frames than the application data to be protected.

Physical network redundancy is implemented very rarely in automotive E/E-architectures, due to the high cost and weight this entails. Instead, when a function detects an error in the communication which impacts its correct behaviour, it informs the driver that it is not available and turns itself “Off”. Thus, as long as the basic driving functions, e.g. steering, accelerating and braking, still work the driver is left in charge to bring the car to a safe hold if necessary. However, with the advent of autonomous driving, this fail-safe behaviour may no longer be sufficient as a driver may not be available to overtake in case of a failure. Physically redundant communication may then be required to build the necessary fail-operational systems.

While redundant communication networks are rarely implemented, redundant execution of critical software functions on different processing components is more common for critical functions. The different processing components may be different CPUs within the same ECU or different ECUs. The hosting of mixed-criticality functions on the same ECUs is often supported by using virtualization techniques. Hypervisors for managing the virtual partitions are also available for embedded computing platforms.

3.2.3 Integration of Software Platform (AUTOSAR) and Network

3.2.3.1 AUTOSAR Aims and Objectives

The application scope of AUTOSAR (AUTomotive Open System ARchitecture) is dedicated for Automotive ECUs, where ECU is treated as one microcontroller with peripherals and the according software with its configuration. ECUs considered have: strong interaction with sensors and actuators; connection to vehicle networks like CAN, LIN, FlexRay or Ethernet; microcontrollers have limited resources of computing power and memory; and finally Real Time Systems are considered.

AUTOSAR aims to be a key technology to manage growing electrical/electronic complexity of innovative systems that further improve performance, safety and environmental friendliness through increased reuse and exchangeability of software modules between OEMs and suppliers. The objectives of AUTOSAR are [23]:

- Implementation and standardization of basic system functions
- Scalability to different vehicle and platform variants
- Transferability of functions throughout network
- Integration of functional modules from multiple suppliers
- Maintainability throughout the whole “Product Life Cycle“
- Increased use of “Commercial off the shelf hardware“
- Software updates and upgrades over vehicle lifetime

3.2.3.2 Overview of AUTOSAR System Architecture

The system architecture of AUTOSAR uses top-down approach to describe hierarchical structure of AUTOSAR software and defines on the highest abstraction level three layers [19]:

- Application Layer
- Runtime Environment (RTE)
- Basic Software Layer (BSW)

All these layers run on top of microcontroller and can communicate only with adjacent layer by means of well-defined interfaces. One exception to this rule makes Complex Device Driver.

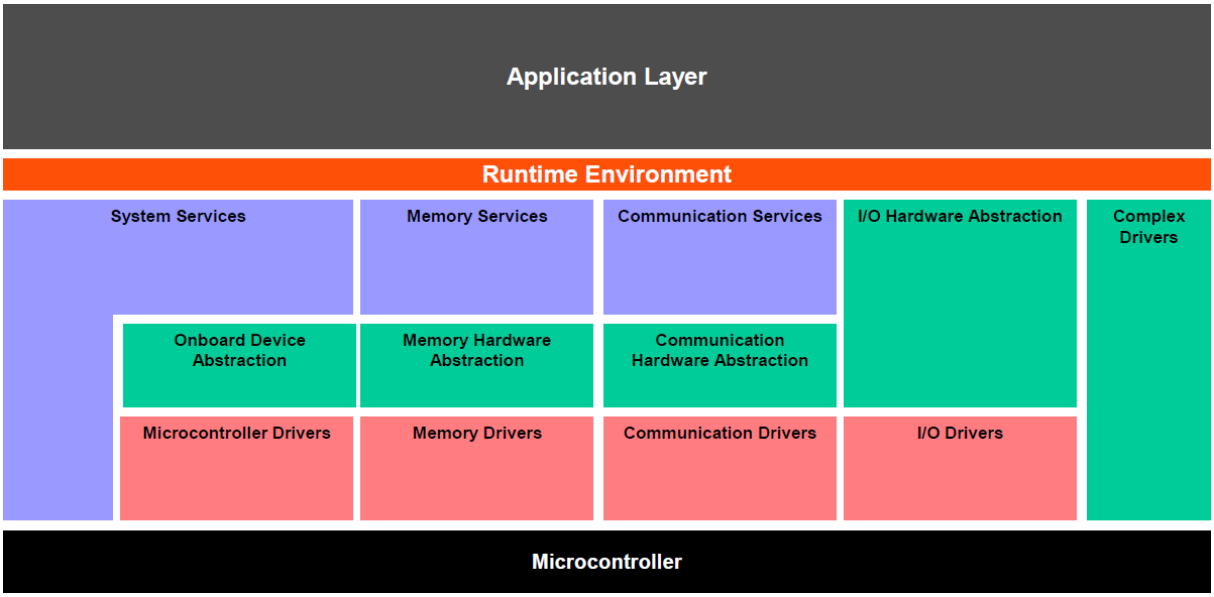


Figure 14: AUTOSAR ECU Layered Software Architecture

The Application Layer hosts Software Components (SWCs) which are decoupled from ECU hardware manufacture and can be independently developed and provided by different vendors. These SWCs represent project-specific functionality whereas all communication of these SWCs with each other and with the Basis Software is carried out over the RTE through well-defined connection points, called *PortPrototype*s [20]. By using this methodology AUTOSAR creates prerequisites for highly automated integration environment for Software Components which are independent of the actual hardware implementation or used communication bus.

Decoupling of hardware and software is done within AUTOSAR by the Basic Software Modules (BSW) [21] further organized in four layers (see Figure 14): Services, ECU Abstraction, Microcontroller Abstraction and Complex Drivers.

Services Layer implements abstraction for operating system, communication and memory management. Hardware abstraction takes place in Microcontroller and ECU Abstraction layers whereas special application requirements, which do not fit to the layered AUTOSAR structure, can be implemented in vertical Complex Drivers layer.

A middleware between Application and Basic Software Modules is represented by the Runtime Environment. RTE. The RTE is specific for the selected software and hardware configuration, so it will be individually generated for every ECU Configuration as needed. Details about the AUTOSAR architecture are provided in Safe4RAIL deliverable D2.1 (Report on state-of-the-art of 'functional distribution architecture' frameworks and solutions).

3.2.3.3 Virtual Functional Bus and Communication Interfaces

With regards to communication, the Virtual Functional Bus (VFB) is the communication mechanism within AUTOSAR that allows individual software components to interact with each other, see [22]. The concept of the VFB allows for a strict separation between application and infrastructure such that software components implementing the application are largely independent of the communication mechanisms through which the component interacts with the other components or with hardware.

The VFB specifies concepts for the following infrastructure-services that are used in automotive applications for implementing component communication:

- Communication to other components in the system
- Communication to sensors and actuators in the system
- Access to standardized services, e.g. read/write to non-volatile RAM
- Responding to mode-changes, e.g. changes in the power-status of the local ECU
- Interacting with calibration and measurement systems

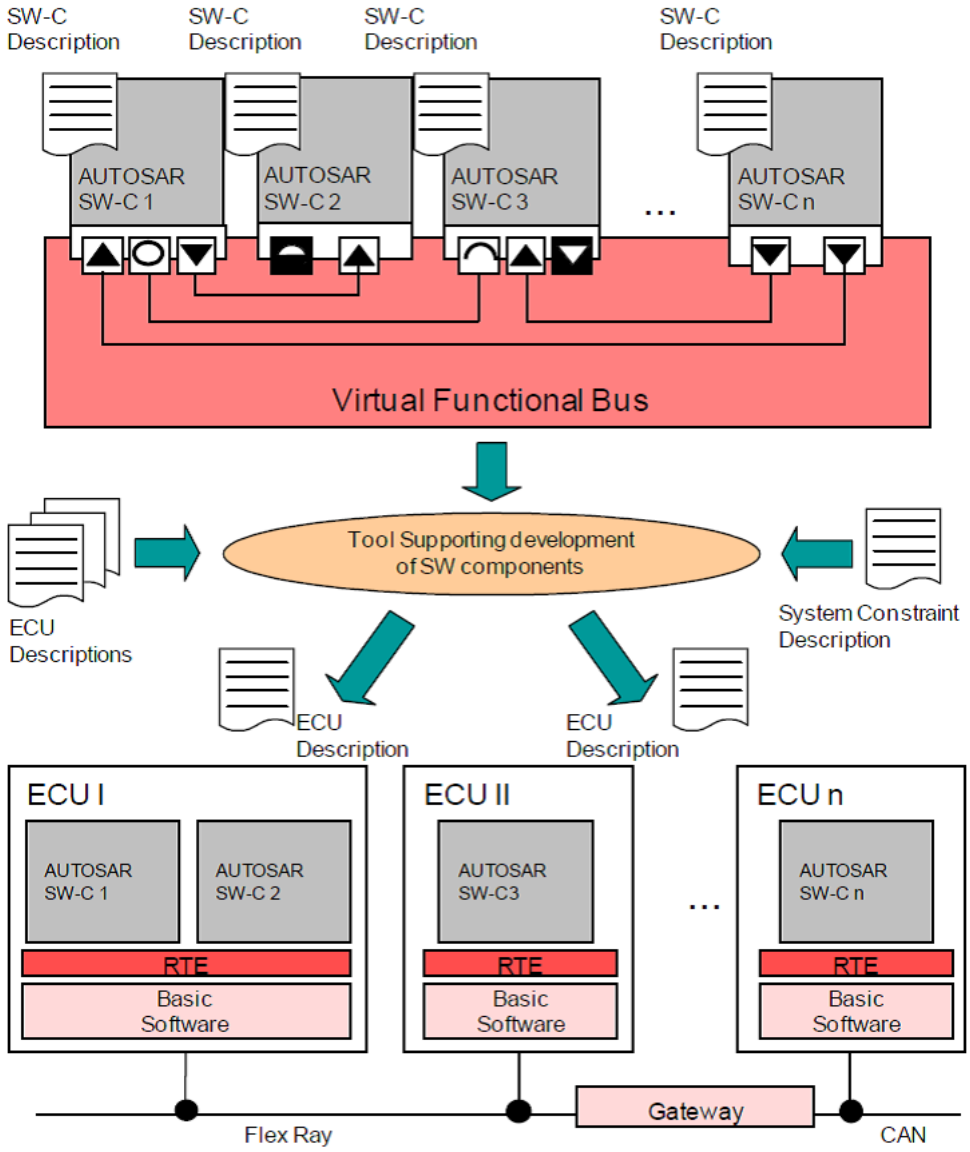


Figure 15 shows a representation of VFB starting with System Design (top of the graphic) down to realization on hardware (bottom of the graphic). The software components and their virtual connections from the early design step are mapped on the system resources, i.e. ECUs such that these connection between the components are then mapped onto local connection (within a single ECU) or on some network-topology realized by specific communication mechanisms. Hence concrete interface between components and BSW Modules is implemented over the RTE. Furthermore, Service Layer within the BSW contains a block of Communication Services which represents a group of modules for vehicle network communication. These interface with the communication drivers (Microcontroller Abstraction Layer) via the communication hardware abstraction on the ECU Abstraction Layer.

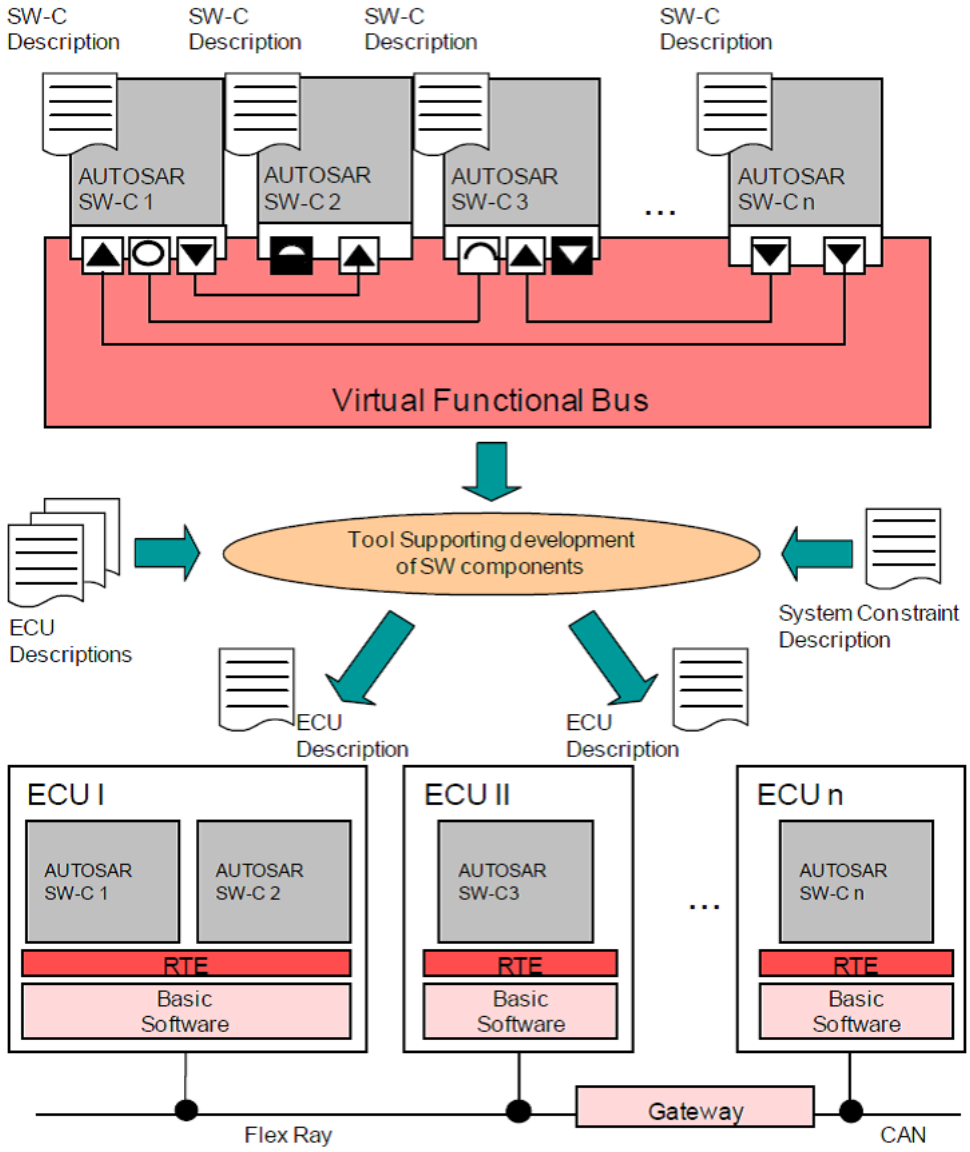


Figure 15: AUTOSAR Virtual Functional Bus: From System Design to Realisation

In order to implement strict separation between application and infrastructure using the VFB concepts, AUTOSAR defines several structural elements like software *components* with well-defined *ports*, through which the component interact with the other components. One or several such ports, where each port belongs to exactly one component and represents a point of interaction, represent a communication interface of a component.

There are basically two types of communication mechanisms available for atomic software components over the VFB, namely: *Sender-Receiver* and *Client-Server* communication and three types of data which may be sent are: *data*, *events* and *modes*. Additionally data validity, infrastructure and application error information will be communicated using the concepts of VFB.

AUTOSAR specifies support for several network communication protocols (listed in Table 1) which are commonly used in automotive.

Name	*BSW Modules*

Controller Area Network (CAN)	Can, CanIf, CanTrcv, CanNm, CanSM, CanTp, CanTSyn
Ethernet	Eth, EthIf, EthSwt, EthTrcv, EthSM, EthTSyn
FlexRay	Fr, FrIf, FrTrcv, FrNm, FrSM, FrTp, FrArTp, FrTSyn
Local Interconnect Network (LIN)	Lin, LinIf, LinTrcv, LinNm, LinSM

Table 1: Communication Protocols in AUTOSAR COM Stack

AUTOSAR supports placements of both arbitrary and fixed communication matrix signals into Protocol Data Unit (PDU), whereas fixed communication matrix allows an optimized usage of low playload networks like CAN or LIN, since e.g. a boolean data can be configured to occupy only one bit in the PDU. Both cases are presented in Figure 16.

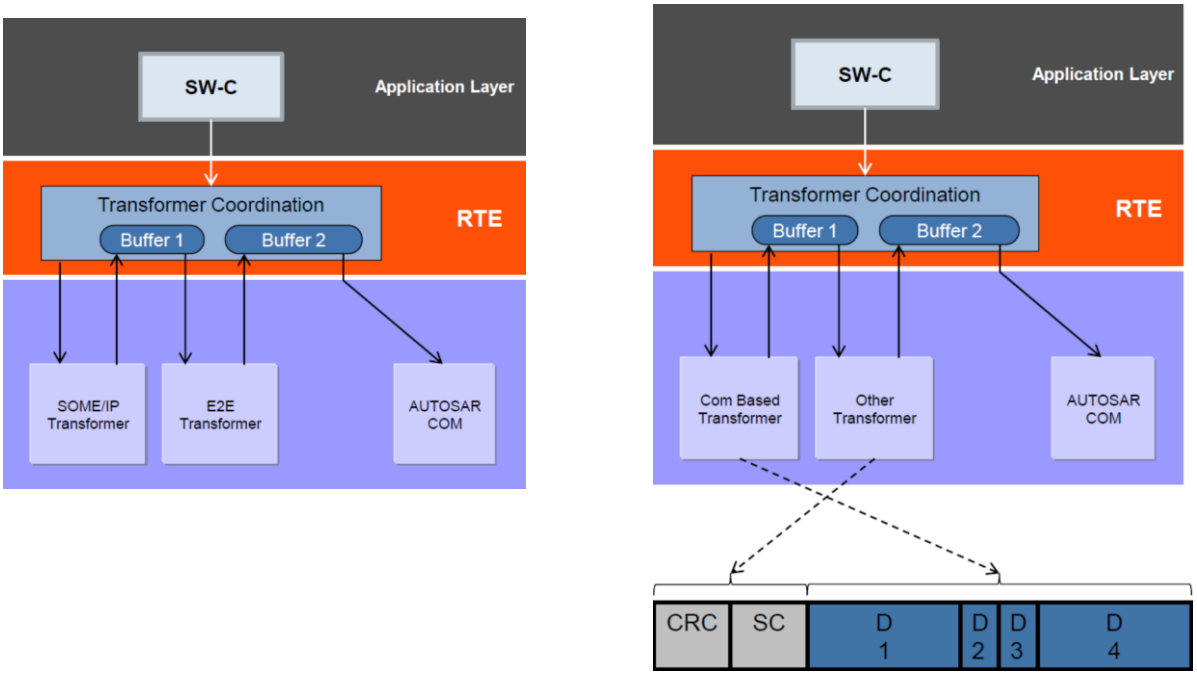


Figure 16: Data Transformation in AUTOSAR Communication Modules

On the left side SOME/IP Transformer is present for serialising arbitrary data signals and on the right side serialisation is done based on the COM configuration, i.e. communication matrix. Other transformers may further enhance the playload to have CRCs and sequence counters (SC) or, like End-2-End Communication Protection Transformer [23] on the left side, enable safety-related data exchange such that faults in the communication link can be detected and handled at runtime.

As one can see, AUTOSAR encapsulates in the automotive industry commonly used communication networks and protocols such that standardized and transparent communication interfaces are supplied for the software components in Application Layer. Hence application software does not have to take care of the backbone network architecture which is designed at system level.

3.2.3.4 Synchronised Time-Base Manager

One important feature for a distributed functional architecture is the global time synchronisation provided by the BSW module “Synchronized Time-Base Manager” (StbM)

[24], such that time bases of multiple nodes of a distributed system are synchronised. AUTOSAR considers two use cases of StbM: synchronisation of runnable entities and provision of absolute time value. The first one takes care that execution of an arbitrary number of runnable entities is done on the same time-base, while the second is responsible for a temporal correlation of signal or event data from different sources.

As the StbM does not provide any network time protocols or time agreement protocols to synchronise its local time bases to the bases on other nodes, it interacts with the BSW communication modules to handle these protocols. Currently there are time synchronisation modules for the CAN, Ethernet and FlexRay protocols specified within the AUTOSAR.

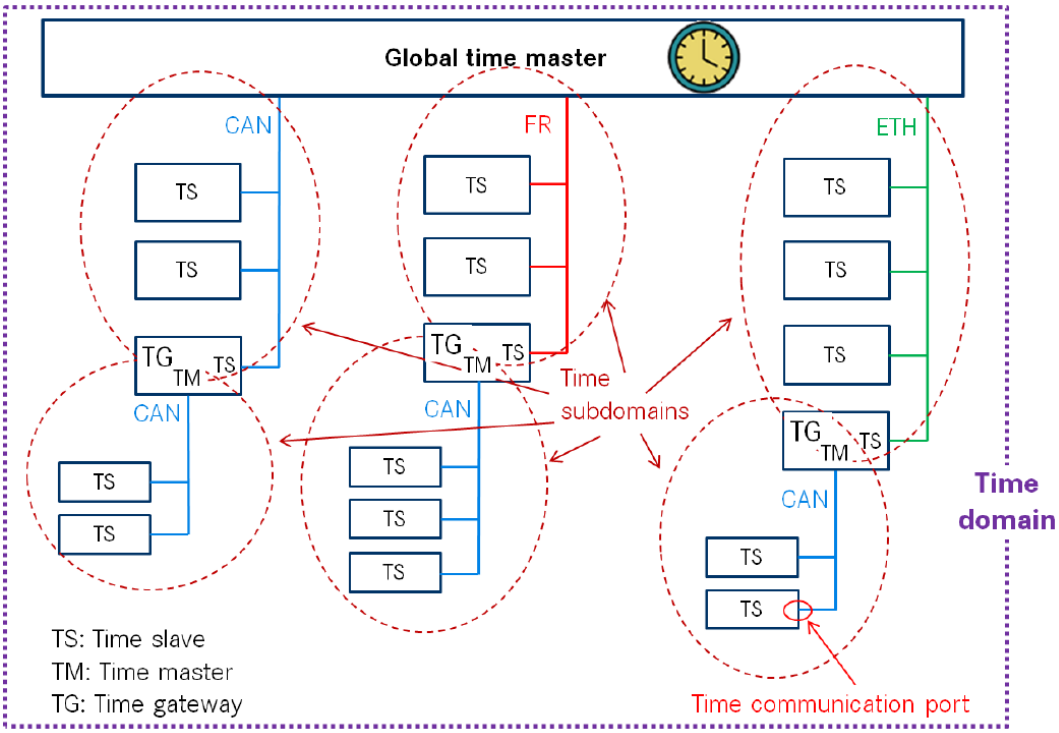


Figure 17: Network Topology of the Synchronised Time-Base

Figure 17 shows an example of the network topology for synchronised time-bases. Here we see one *time domain* managed by the *global time master* which is the owner of a certain time base. Global time master defines network protocol specific *time masters* (TM) and hence specifies some certain *time subdomains*. Furthermore, these subdomains may have multiple *time gateways* (TG) containing one *time slave* (TS) acting as time base recipient and multiple time masters which distribute this time base to sets of time slaves e.g. other time subdomains. Time gateway can be connected to different types of bus systems.

3.2.4 Future Outlook on Automotive System Architectures and System Integration

There are two trends which will shape the future automotive system architectures, namely autonomous driving and connected cars.

Regarding autonomous driving, safety aspects, especially support for fail-operation systems will become much more important. Such a future architecture is demonstrated for example in the German RACE (**R**eliable **A**utomation and **C**ontrol **E**nvironment) project [30].

Fehler! Verweisquelle konnte nicht gefunden werden. shows the system architecture roposed by the RACE project for safety critical functions, especially autonomous driving. The architecture features redundant ECUs and, additionally, the network is organized in a redundant ring topology. Thus, there are always two physical network paths between any two ECUs providing redundancy on the network level.

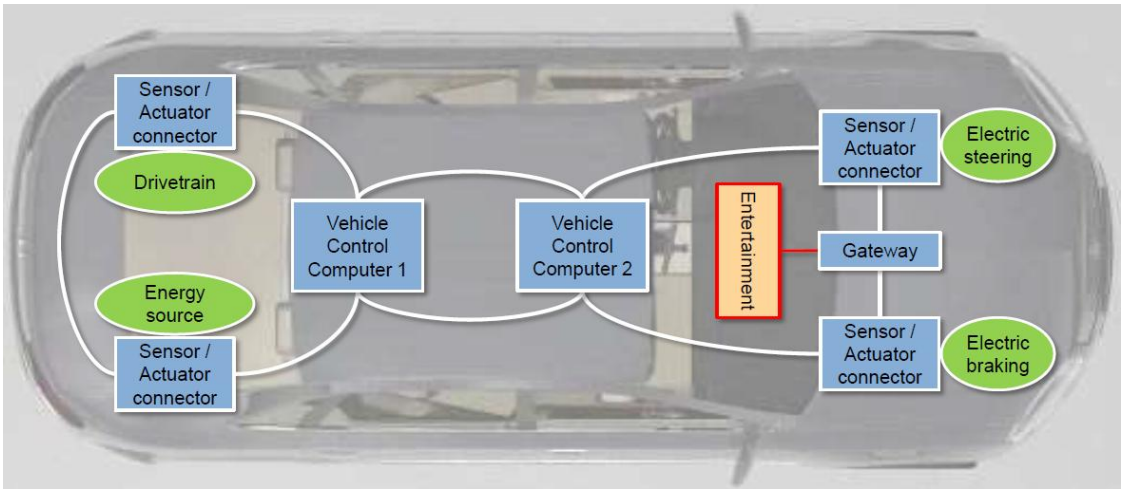


Figure 18: System architecture envisioned by the RACE project [30]

With the future cars becoming more and more “connected”, new requirements for their system architectures are emerging. First, the system architectures must support online updates. This comprises updating existing functions, but also the download of completely new functions (as long as the necessary hardware is available in the system). Naturally, this is easier with a more centralized architecture that allows for complete separation of logical functions and physical devices.

Second, with the connected car becoming part of the internet of things, functions may be partly or as a whole be implemented “outside” the car, i.e. in the cloud. Thus, parts of the system on which the functions execute are not in control of the OEM. This requires the system architecture of the cars to provide adequate fall-back mechanisms in case the required systems “outside” the car is not available, e.g. no internet connection in a tunnel. Additionally, security becomes very important requirement, when the car is always connected. **Fehler! Verweisquelle konnte nicht gefunden werden.** shows a possible future system architecture of connected cars in which the system is divided in to a connected layer and an automotive layer.

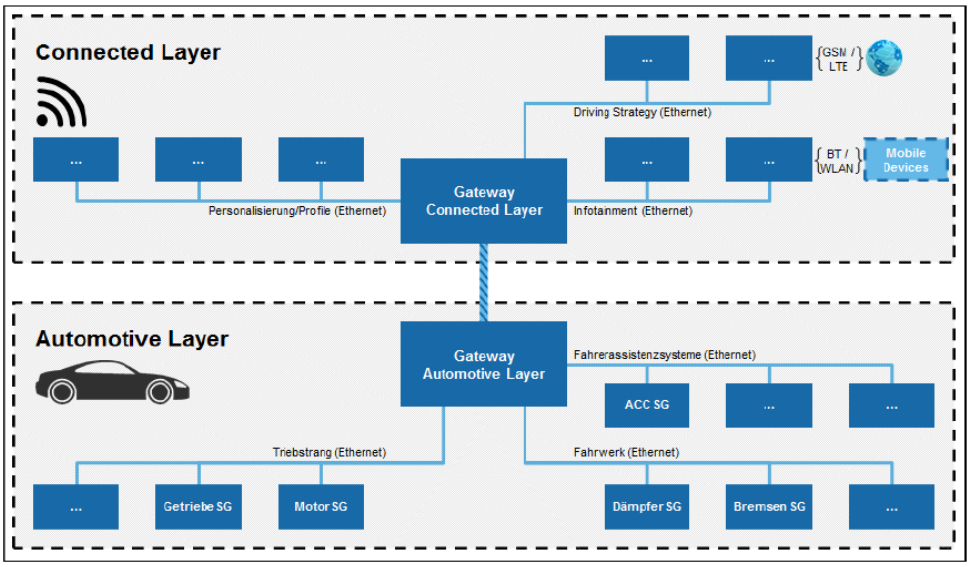


Figure 19: Possible future system architecture of connected cars [31]

This separation not only facilitates the access control to the automotive layer from outside the car, it also decouples the development within the two domains. Therewith faster development cycles can be supported in the connected layer while still maintaining slower development cycles in the automotive layer.

3.3 Railway

3.3.1 Current Status and Standards

Communication systems in the railway industry exist mostly as proprietary solutions factorized to meet the requirements of national railway operators and vendors. Apart from the analogue Multiple-unit train control systems mainly used for traction control only the slow WTB/MVB based TCN described in IEC 61375-1, IEC 61375-2-1ff can be called standardized.

Recent additional parts of the evolving IEC 61375-family define a faster TCN based on standard 100Mbit/s Ethernet conformant to IEEE 802.3 and IEC 61076-2-101. Ethernet communication is used as a backbone (ETB) and in car/consist networks (ECN) with proprietary higher layer protocols, e.g. TCP/IP, IPTCom or CIP, already. Aside from being standardized in IEC 61375-2-5 (ETB), Train Switches/ETB nodes from several manufacturers already support Gigabit-Ethernet (1Gbit/s).

While current trains using these proprietary protocols implement their own coupling methods (inauguration), the evolving new parts IEC 61375-2-5 and IEC 61375-2-3 standardize the TCN in a way that shall allow the coupling of consists built by different manufacturers.

In the current state, it will resolve IP address conflicts and allows leading cab and direction determination. TRDP has been defined as the inter-consist communication protocol, providing Layer 3 + 4 services using UDP/IP and TCP/IP. For leading-direction related communication, a safety layer (SDTv2) has been standardized, which allows for safety functions up to SIL2.

IEC reference	CLC reference	Title	IEC situation	CLC situation
IEC 61375-1 Ed.3	EN 61375-1:2013	Part 1: General Architecture	Published (2012)	Published (2013) following // vote
IEC 61375-2-1 Ed.1	EN 61375-2-1:2013	Part 2-1: WTB – Wire Train Bus	Published (2012)	Published (2013) following // vote
IEC 61375-2-2 Ed.1	EN 61375-2-2:2013	Part 2-2: WTB - Wire Train Bus Conformance Testing	Published (2012)	Published (2013) following // vote
IEC 61375-2-3 Ed.1	EN 61375-2-3:2016	Part 2-3: Communication Profile	Published (2015) Corrigendum published in 2015	Published (2016) following // vote
TS 61375-2-4 Ed.1	-	Part 2-4: Application Profile	Work in progress	-
IEC 61375-2-5 Ed.1	EN 61375-2-5:2015	Part 2-5: Ethernet Train Backbone	Published (2014)	Published (2015) following // vote
IEC 61375-2-6 Ed.1	-	Part 2-6: On-board to Ground Communication	Work in progress	-
TR 61375-2-7 Ed.1	-	Part 2-7: Wireless Train Backbone	Published (2014)	-
IEC 61375-3-1 Ed.1	EN 61375-3-1:2013	Part 3-1: MVB - Multipurpose Vehicle Bus	Published (2012)	Published (2013) following // vote
IEC 61375-3-2 Ed.1	EN 61375-3-2:2013	Part 3-2: MVB - Multipurpose Vehicle Bus Conformance Testing	Published (2012)	Published (2013) following // vote
IEC 61375-3-3 Ed.1	EN 61375-3-3:2013	Part 3-3: CCN - CANopen Consist Network	Published (2012)	Published (2013) following // vote
IEC 61375-3-4 Ed.1	EN 61375-3-4:2015	Part 3-4: ECN - Ethernet consist network	Published (2014)	Published (2015) following // vote

Figure 20: TCN standards

3.3.2 Overview: From Fieldbus to More Integrated Ethernet-based Architectures

The first fully digital and standardized electronic control systems in trains were based on the WTB, the Wired Train Bus, as a backbone and the MVB, the Multifunction Vehicle Bus for car/consist internal communication. Both bus systems use RS-485 as the physical layer.

The interoperability of trains using the WTB is standardized: IEC 61375 defines the physical, lower layer interface specifications and the train inauguration process – assigning to each node its sequential address and orientation. Additionally, several UIC leaflets (e.g. UIC556) define the meaning of exchanged information.

	WTB	**MVB**
Media:	Shielded Twisted Pair, Fibre Optics	Shielded Twisted Pair, Fibre Optics, 2-channels
Signalling:	RS-485	
Data rate:	1 Mbit/s	1.5 Mbit/s
Max. no. of nodes:	32	
Max. segment length:	860m	200m (2000m optical fibre)
Max. data packet size:	1024 bit	16 … 256 bit process data
Coding:	Manchester 2, HDLC	
Typ. Latency:	<25ms	<16ms

Table 2: WTB/MVB Basic Specs

MVB supports Process Data (cyclic transmission of relative small chunks of data) and Message Data (event-driven, up to several 100kB) transmissions.

It is currently used for all operational control and status functions inside a consist: Traction control, brakes, doors, lighting, HVAC, heating. Message Data is mainly used for on-line-diagnostics, event recorder, maintenance and passenger information.

“The MVB standard was introduced to replace the multitude of field buses in the train equipment. This was noted to be not the case for several reasons. While the CANopen and PROFINET are controlled by international manufacturer associations targeting wide application range this is not the case for MVB which allows no options and is used only in railways and in some electrical substations. As a result, MVB modules are more expensive than for instance CANopen components. This is not due to the communication technology itself: most devices implement the MVB protocol machine in a small area of an FPGA which is today anyhow present, and the most costly component remains the connector. But railways certification is costly and not always needed for uncritical applications such as comfort and passenger information. When total cost of ownership is considered, the cost of the hardware elements can easily be outweighed by additional engineering costs in the railways market with its small series.

This has led to the observation that – despite the advantages of the MVB field bus – many train vehicle buses are still built from CANopen and PROFIBUS components. Additionally more and more components are added to rail vehicles that need far more bandwidth than any field bus can provide (e.g. for video surveillance), so switched Ethernet IEEE 802.3 with 100 Mbit/s is being introduced into train sets (according to the EN 50155 profile). Still all the alternate vehicle buses are connected to the Wire Train Bus.”

Therefore, fieldbus usage inside a car/consist is not limited to the MVB – for example a hybrid traction subsystem, where the recuperation and charge control is connected to a MVB device via CAN-bus.

Another variant on consist level, the Ethernet-based PROFINET is used to extend the bandwidth of the MVB by some vendors.

WTB Limitations

- The UIC command set is limited and difficult to extend without braking compatibility and interoperability. This limits development of new functionality.

- The maximum data rate of 1 Mbit/s limits the usage of the backbone to control and status commands, only. Video surveillance, for example, needs a much higher bandwidth – passenger comfort functions may demand even more.

Fieldbus Limitations

- Data rate
- Number of nodes
- High cost of interfaces (MVB)

3.3.3 Evolution of High-Bandwidth Networking in Railway Systems

To overcome some of the limitations of the WTB/MVB based TCN, vendors have extended their systems by adding higher speed Ethernet networks in parallel to the existing low speed busses.

The safety related functions (traction, doors, brakes) are still controlled via WTB/MVB, while the higher bandwidth of the ETB/ECN allows more sophisticated functions for comfort and passenger information (e.g. audio messages), surveillance, and more.

Figure 21 shows the layout of such a system in principle. There are two backbones, each equipped with two lines for redundancy.

The MVB is connected to the WTB by TCN-Gateways (green lines). One of the two redundant VCUs is the MVB-bus master and serves as the main controlling instance. MVB enabled devices are connected directly to the MVB lines, e.g. Multiple Input Output (MIO) or other subsystems.

The comfort network in this example consists of a ring structure (this is a redundancy concept, see Figure 28) with ring switches. End device can only be connected to the ECN via Ethernet switches – opposed to devices on the MVB, which can be attached directly.

The ECN is connected to the ETB via train switches. The label ‘switch’ is misleading – beside switching Ethernet packets on the backbone, it must also route packets between ETB and ECN and also be able to manage a train inauguration.

This leads to a major problem in such mixed designs with multiple backbones: The inauguration on both backbones need to be synchronized to keep the same numbering of the consists and cars (a faulty train switch could lead to a different assumption on the number of consists). One common solution is to implement a ‘reconciliation’ – layer within the gateways/routers.

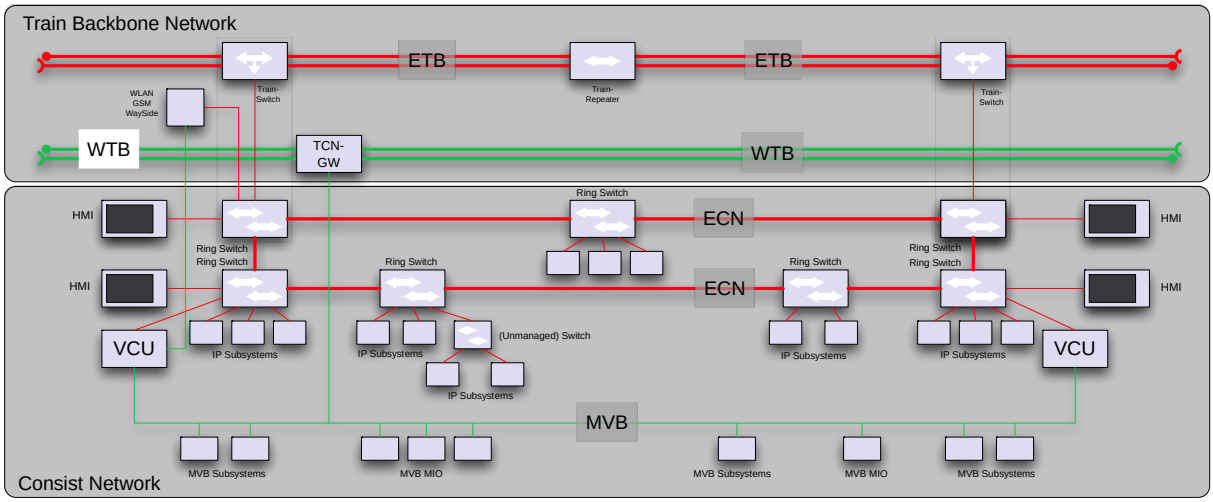


Figure 21: Mixed WTB/ETB Consist

The higher protocols used in such an Ethernet network are mostly IP related: DHCP, DNS, TCP/IP, UDP/IP (IPTCom), RTP, NTP, VRRP, NAT, RNAT, LLDP (ETB neighbour detection) and more. Allowing the use of COTS technology for non-safe functions is a big plus.

Limitations

- synchronization between ETB/WTB necessary (reconciliation)
- non-standard inauguration on ETB, no interoperability
- extended cable length per consist -> weight and costs
- maintenance costs for two different network technologies

3.3.3.1 Topology

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 fleet usually exists of a series of consists or cars, equipped with the same network (and device addresses).

Although a car is the smallest item from the physical view, it will not necessarily show up as a separate network item. The smallest network part is usually the consist, a group of coupled cars not separated or changed during normal operation.

IEC 61375-1 defines a train as a composition of closed trains and consists, each consist having one or several vehicles, and each closed train having one or several consists.

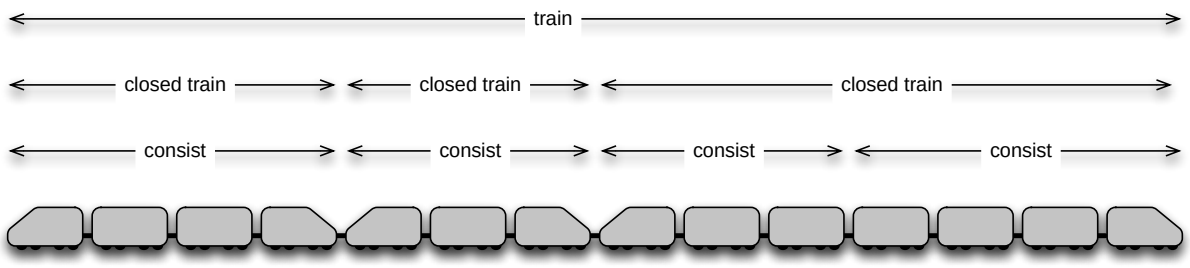


Figure 22: Train composition and hierarchy

3.3.3.1.1 Ethernet Train Backbone topology

Each Ethernet train backbone consists of two redundant 100Mbit/s lines with using link aggregation (IEEE 802.1AX).

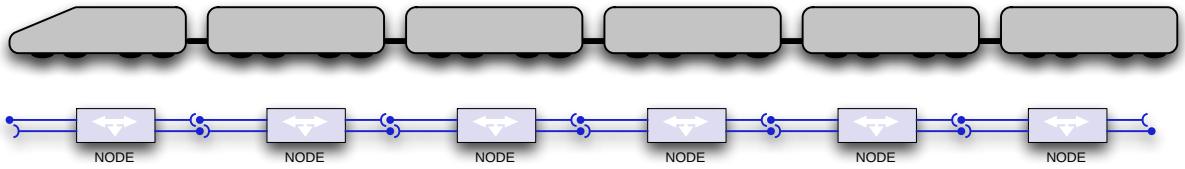


Figure 23: Redundant train backbone archtiecture

Link aggregation as described in IEEE 802.1AX is managed at OSI layer 2 and allows one or more lines to be aggregated together to form a logical group, able to manage the link redundancy.

Link aggregation combines several individual lines, each having a physical and MAC layer. From the MAC client, a single MAC interface is provided.

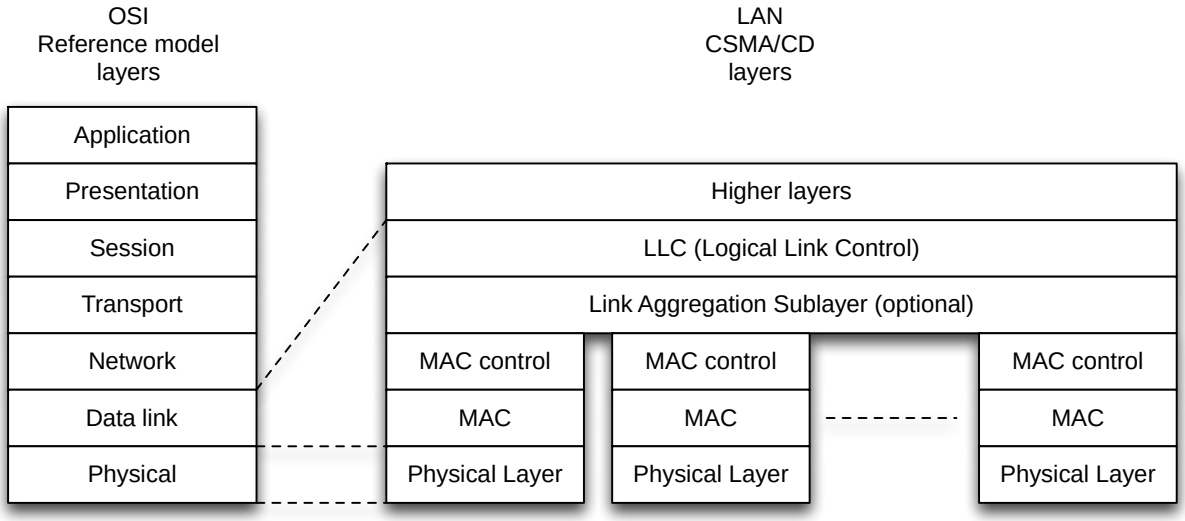


Figure 24: Link Aggregation

Between two ETBNs, there is only one link aggregation group, which contains the redundant Ethernet segments. The link aggregation process is only defined as a relation between 2 ETB nodes.

The nodes on the train backbone are actual switches (on the ETB-side) and routers between the backbone network and the consist network. To ensure high reliability, there should always be a redundant switch in each consist – also to overcome the maximum Ethernet cable length of 100m. In case of a malfunction (e.g. power supply failure), each ETBN must provide failsafe relays bypassing ETB traffic.

In passive bypass setting, the ETB lines will bypass the ETB switch, which then is decoupled from the ETB lines (see Figure 25). The Passive Bypass Setting is the default setting in the powerless state and the ETB switch is out of order.

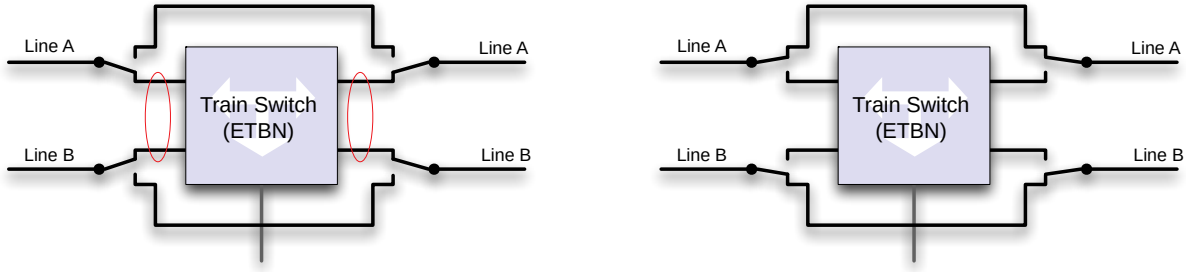


Figure 25: Fallback on ETBN – left: active, right: passive mode

If the cable length between two ETBNs exceeds 50m, a repeater needs to be provided (if one ETBN fails and is bypassed, the resulting effective distance would exceed the 100m limit).

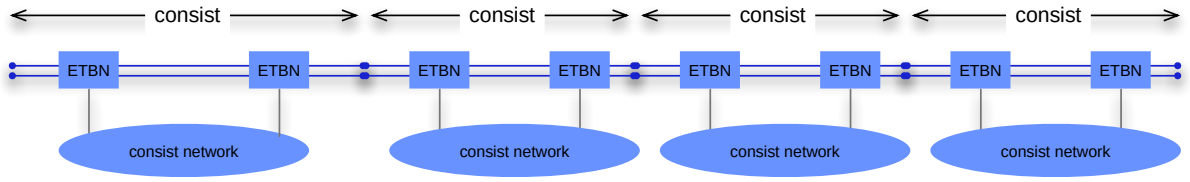


Figure 26: Consists on ETB

IEC 61375-2-3 and 2-5 define up to 4 parallel backbones. ETB0 is defined as attached to the operational network, providing control services and devices for door control, HVAC, lighting, drive control etc. (up to SIL2). ETB1 is defined as backbone for a multimedia network and is supposed to be used for CCTV and passenger information systems (PIS).

This parallel structure makes it necessary to pay attention to topology differences in case of faults on one ETB: Inauguration results from ETB0 must be forwarded to ETB1 to keep up correct addressing between both networks (reconciliation).

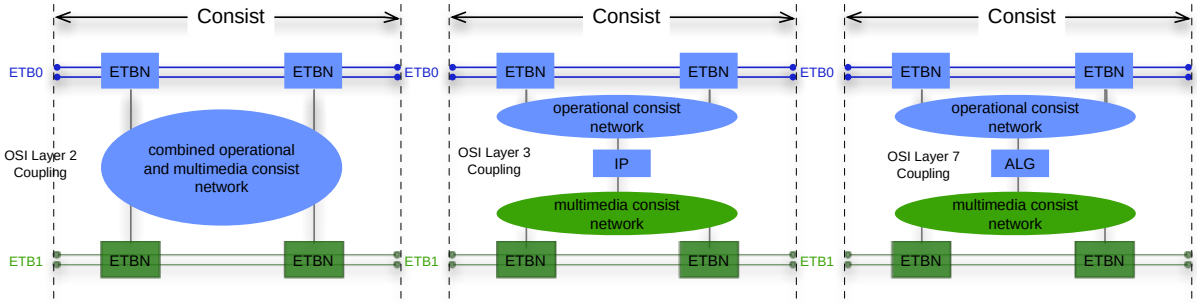


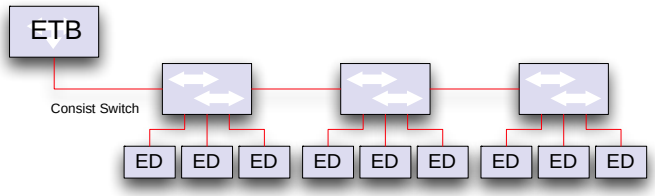
Figure 27: Dual ETB topology

3.3.3.1.2 Ethernet Consist Network

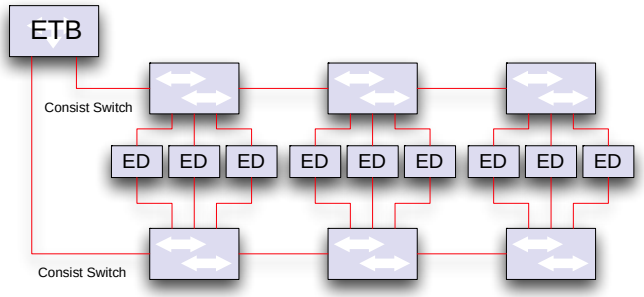
The physical layer of the ECN is defined in IEC 61375-3-4, while addressing and ETB-related control services (ECSP, ECSC) are laid down in IEC 61375-2-3.

The topology of the ECN can be quite different and depends on the vendor's preferences:

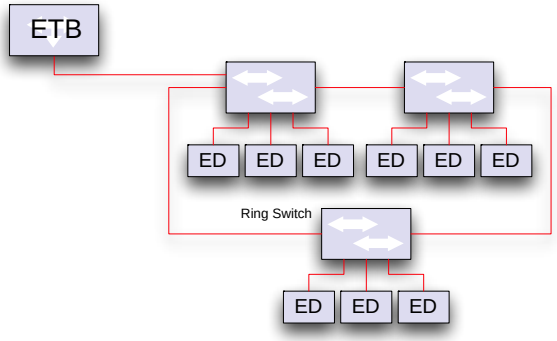
While some vendors prefer the ladder topology, where each end device is connected to two lines, others use a ring topology or favour a hierarchical approach. See Figure 28 for some example topologies.



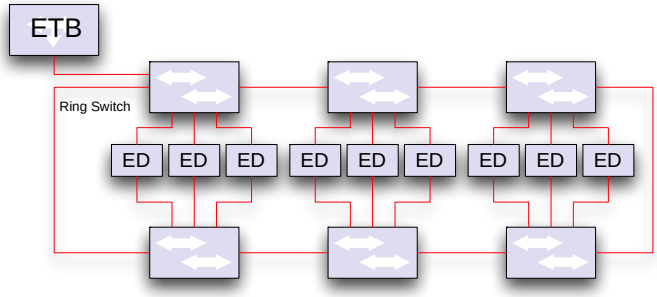
Linear Topology



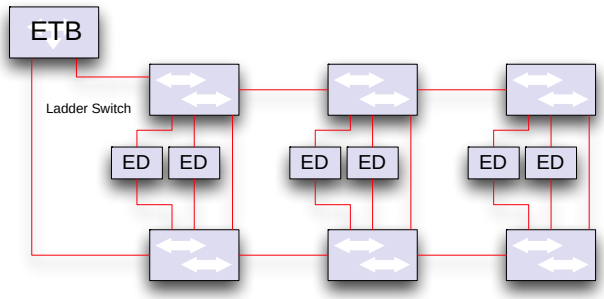
Linear Topology with dual homing



Ring Topology



Ring Topology with dual homing



Ladder Topology with dual homing

Figure 28: Topologies for the ECN

3.3.3.2 Hosting of Critical Functions

Looking at current implementations of Ethernet-based train networks, for example the ones using the IPTCom protocol, mixed WTB/ETB networks host safety related functions on the WTB/MVB side of the network, only. Gateways between the two networks have to ensure proper isolation of critical functions. On the MVB side of the TCN, door controllers, consist controller (VCU), traction controller (e.g. MIO), HMI (traction & door control) must be developed conformant to at least EN50128 SIL2.

On an ETB/ECN network, where SIL2 functions are implemented, the train switches play a key role for safety critical functions and must be seen as playing a centre role in a safe train network.

On inauguration, any node on the backbone (ETBN) connecting an ECN, takes part on determining the operational train directory, which in turn determines the position of the leading car/consist, number of vehicles, and operating direction of the train. At least the computation of the operational direction is a safety-critical function (e.g. operating the doors on the right sight).

Communication over ECN is considered as a 'grey channel' – IPTCom and TRDP use standard UDP/IP with an optional SDTv2 safety layer for SIL2 functions. Both protocols require QoS options from the underlying socket layer (default 5 for Process Data) and a minimum of 4 priority queues in any switch connecting devices using SDTv2 protected communication.

Aside from QoS, sufficient bandwidth and latency for critical functions have to be considered by the projector of the train / consist.

3.3.4 Advanced Architectures and System Integration Requirements

The current standard defines up to 4 ETBs in parallel; mainly to allow separating safety related traffic from non-critical devices, and to partition available bandwidth for future functions. It would also ease upgrading / adding more non-critical options without demanding re-certification, if the TCMS network is clearly separated.

Clearly, this approach has a major drawback: Each additional ETB adds not only the backbone cables plus train switches, but also needs a separate ECN with all the necessary infrastructures (cabling, switches -> weight). Reconciliation and commissioning of multi-backbone networks are additional issues.

One solution to reduce the need for parallel ETBs is the move to Gigabit-Ethernet (1000BASE-X) instead of 100BASE-TX. Several vendors offer such high-speed train switches, although it is not standardised.

Another approach proposes the coupling of car/consists via a wireless connection (WLAN) – it is mentioned here for completeness, only.

If there is one physical train network (ETB/ECN), access needs to be controlled by all switches providing ports with different priority properties.

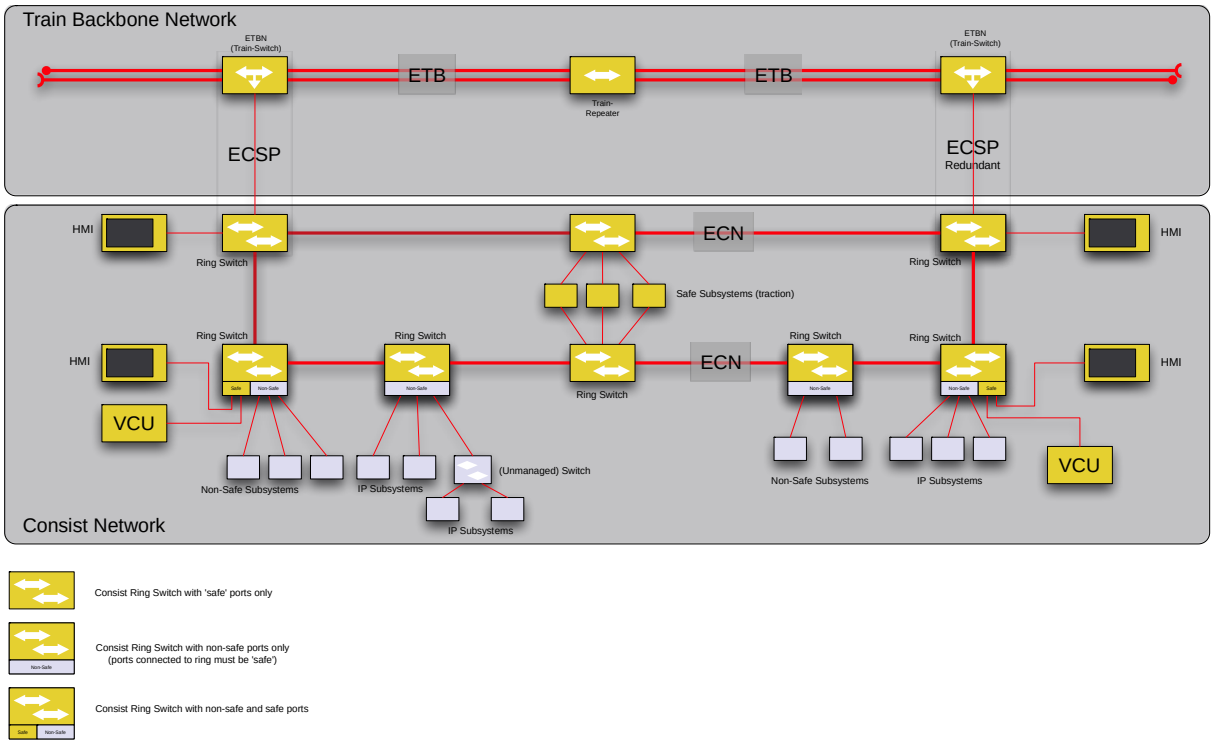


Figure 29: Sample mixed consist network with safe/non-safe functions

The ring switches in the example (Figure 29) need to prioritize safe ports over non-safe ports, possibly overriding or ignoring priority values of packets on certain switch ports and thus providing a partitioning between the two domains. Additionally, to enhance the partitioning, VLAN tags could be used to limit access of non-safe devices to safe devices and provide absence of interaction.

3.3.5 Integration of Software Platform (TRDP) and Network

Access to TCMS functions of the ETB and ECN is provided by the TRDP, which is defined in IEC 61375-2-3 Annex A, and an optional safety layer SDTv2, defined in Annex B of the same standard documents.

TCNOpen is an open source initiative and follows the Open Source scheme, as the software is jointly developed by participating companies, according to their role, so as to achieve cheaper, quicker and better quality results – and accelerate acceptance of the standard's new parts.

Figure 30 shows the basic layout of the software layers of a safe device. The yellow line divides the device into two partitions:

- the 'safe'-TCN application, which usually resides on a separate processor
- the 'un-safe' or 'grey' communications stack handler

Non-safe TCN applications, like the TRDP Handler, do not need separate partition. The TCNopen TRDP stack has been designed to support several operating and target systems by using a so-called virtual operating system (VOS). Ports exist to vxWorks, Embedded Linux, BSD, QNX, Mac OS, Windows32 and rcX. Supported hardware architectures are ARM, PowerPC, Intelx86, netX (SDTv2 and the netX port are not yet open source).

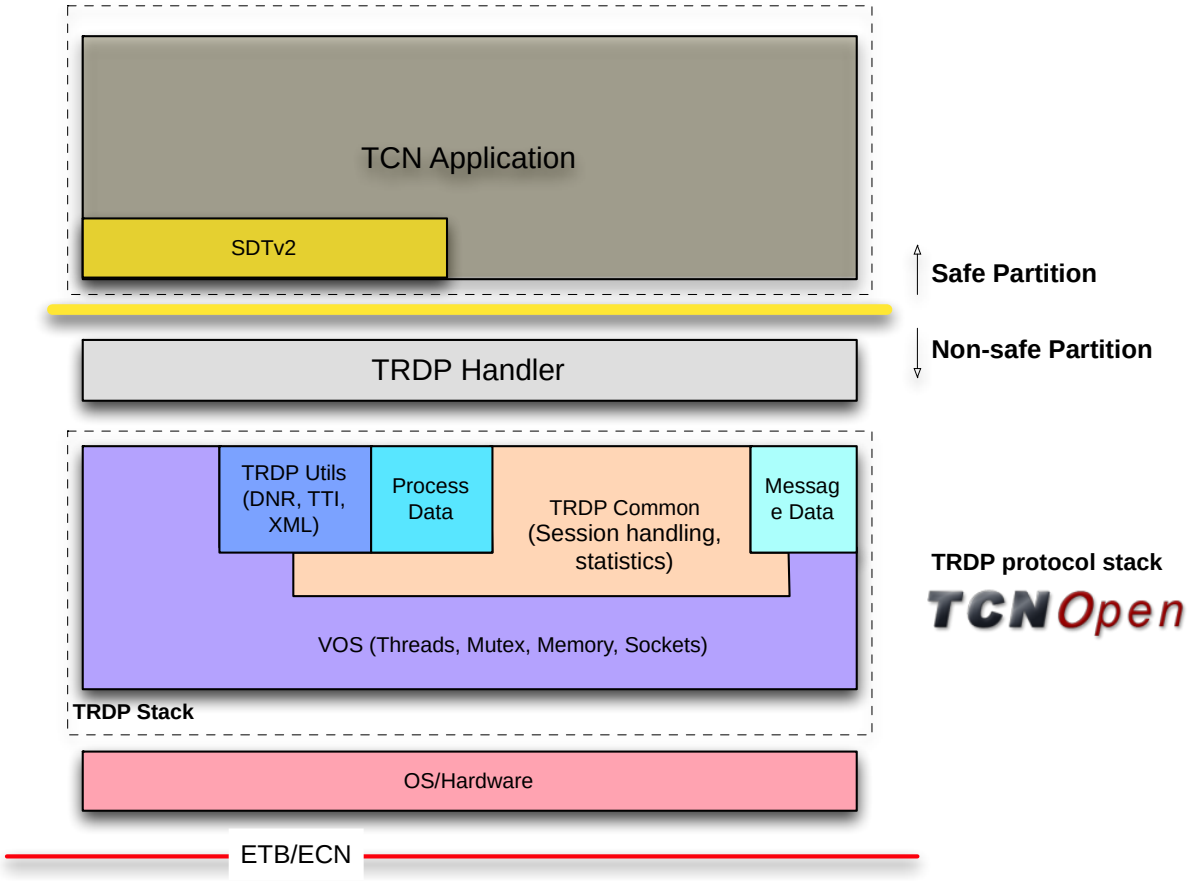


Figure 30: Safe TCN application using the TRDP stack

TRDP supports

- process data communication (cyclic transmission of control and status data, UDP/IP, ≥ 10ms, ≤ 1432Bytes payload, timeout supervision)
- message data communication (event driven, UDP or TCP/IP, ≤ 64kByte, timeout supervision)
- push, pull (PD), notify, request/reply, request/reply/confirm (MD) communication patterns
- TCN addressing (inauguration, topology verification, functional & hostname URIs)
- SIL2 safety layer (SDTv2)
- Various redundancy concepts

3.3.5.1 Communication Pattern

TRDP provides several ways of communication between network devices. The major pattern used in the TCMS is Process Data (PD), which supports cyclical publishing of data to one or more subscribers using unicast or multicast IP addressing.

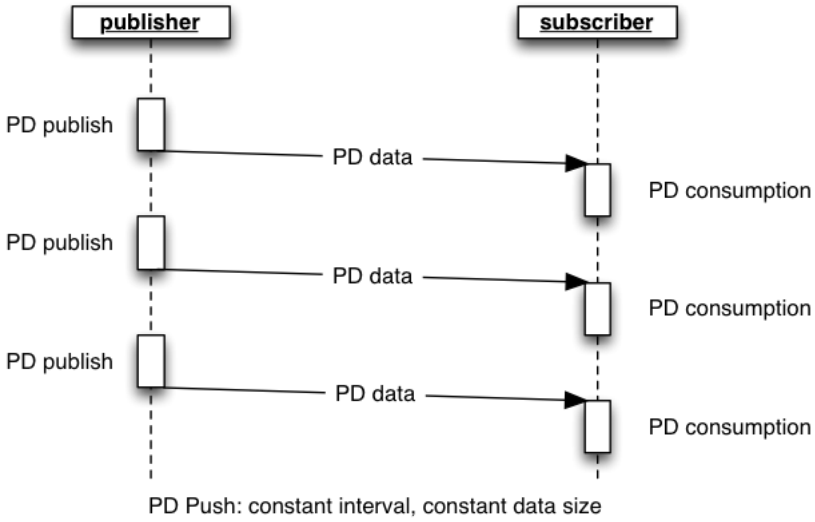


Figure 31: PD Push unicast

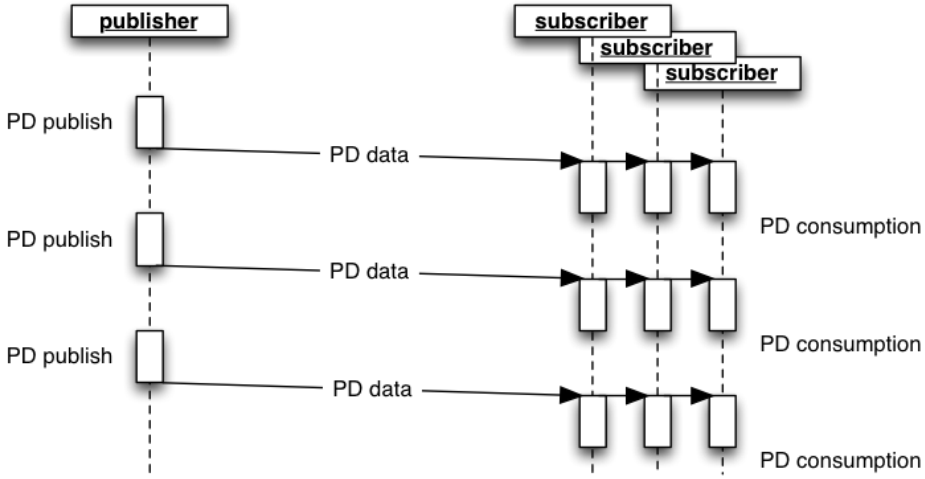


Figure 32: PD push multicast

For event driven data exchange, Message Data over UDP or TCP can be used, which allows for larger data transmission.

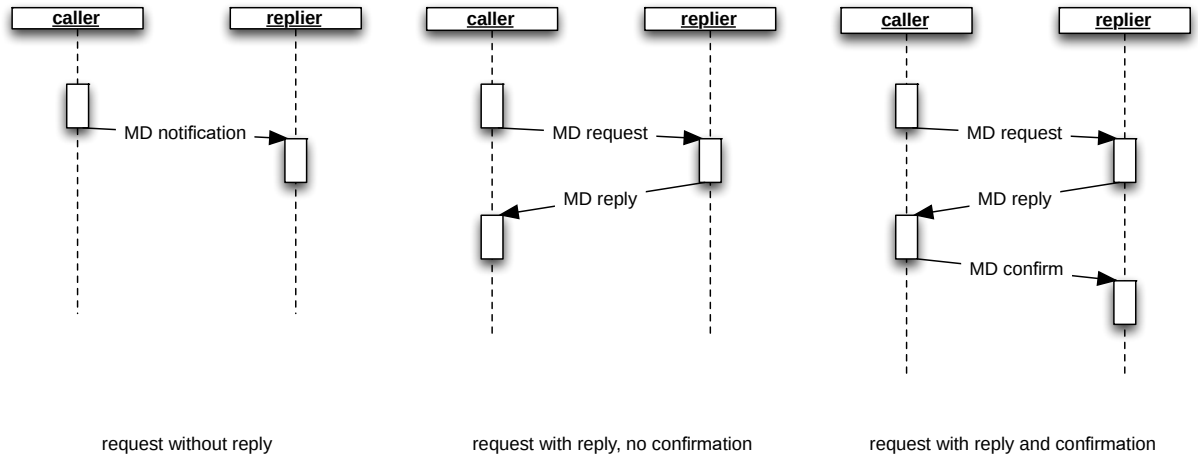


Figure 33: Message Data exchange

PD and MD data can optionally be secured by the SDTv2 protocol, which adds a SIL2 safety layer on top. From the user point of view, data exchange may look like depicted in Figure 34. The publisher prepares a transmitter by defining all relevant parameters (ComId, cycle time,

destination) by calling the TRDP publisher function. After providing the first data TRDP starts transmitting the telegram to the destination IP address using the provided cycle time.

An interested device needs to create a receiver by subscribing to that ComId (and possibly to the source address) and can then either receive data by means of an indication (callback) or start polling for data. In case packets are lost and a defined timeout has been set, the subscriber will receive a time out.

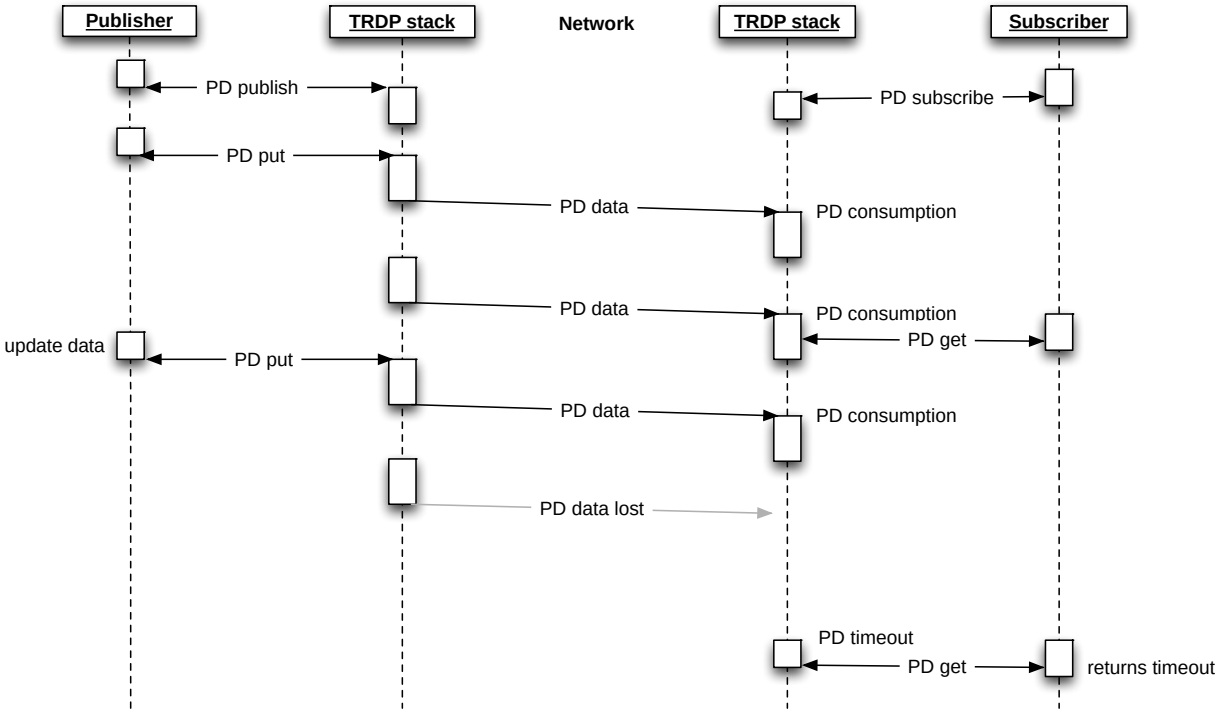


Figure 34: Process Data exchange - push pattern

3.3.5.2 Addressing and Train Topology

Especially for TCN addressing over the ETB to work, every consist needs to have an ECSP, an ETB Control Service Provider, which, in addition to the ETBN, provides DNS and TCN specific services. The ECSP computes the train directories, both static and dynamic (operational), and distributes the current valid topology counters. To simplify design, the ECSP should reside as a service task on each active ETBN (IEC 61375-2-3).

It can be controlled by a local ECSC, the ETB Control Service Client, if the local consist has leading capabilities and it will align with all other ECSPs in a train (other consists) to compute the operational train directory. The CRC over that directory is the operational topology counter, which must be transmitted in the header of every direction related (SIL2) TRDP telegram. Telegrams where one of the topology counters do not match the locally known values, will be discarded by the TRDP stack and eventually lead to communication timeout. This is to prevent misrouted telegrams during inauguration (de/coupling or direction change).

3.3.6 Future Outlook on Railway System Architectures and System Integration

3.3.6.1 Overview

Regarding Ethernet-based communication networking in the TCN this is the current state:

- There are several vendor specific implementations of Ethernet-based networks inside consists and as ETB.
- There do exist SIL2-certified versions, where drive control is realized using the IPT-Wire protocol (using IPTCom as predecessor to TRDP on ETB and ECN).

- The new parts of the IEC 61375 standard define ETB and ECN networks suitable for reliable TCMS functions up to SIL2.
- The new standard provides interoperability for static (IP) and dynamic (TCMS) addressing (train inauguration).
- A crucial part of the standard is missing: 61375-2-4 shall define common application profiles, which would allow TCMS interoperability between trains of different vendors.

3.3.6.2 Safety

The current TCN is considered to provide grey-channel communication. The network topologies provide ways to enhance reliability by several redundancy concepts (which are until now dependent on the train projectors, but there are proposals laid out in the current standard).

Hardware Provisions and Requirements:

- Each ETB comprises at least two Ethernet lines.
- Each consist network should at least be connected through two redundant ETB nodes.
- Train switches / ETBNs on the backbone must have fall-over capabilities to keep ETB communication alive. Repeaters ensure reliable communication in case of extended segment length caused by single ETBN failure.
- Safety and non-safety functions can be separated through up to 4 distinct ETBs.
- Safety critical devices need to have extended failure protection (memory surveillance, watchdogs, redundant power supply, separate ‘safe’ and communication CPUs)
- Precedence queuing of high priority process data in all train switches is required (minimum 4 queues).
- Two redundant interfaces (dual homing) are supported by several ECN topologies.

Software / Protocol Provisions:

- The TRDP is standardized and freely available (TCNopen).
- The SDT safety layer provides additional timeout supervision, data integrity and source verification on top of TRDP or IPTCom.
- IP standard protocols supporting fast redundancy switchover like VRRP (Virtual Router Redundancy Protocol) or RSTP (Rapid Spanning Tree Protocol) for fast topology recovery can be used (COTS).
- Certified operating systems (including network stacks and tools) like ‘Integrity’, ‘vxWorks’ or ‘QNX’ can be used.

Critical Issues / Gaps:

- The computation of the operational train directory is complex and must be carried out in the safe domain (ECSP).
- Application profiles are not standardized yet.
- SIL4 functions inside the consist network require the separation of non-safe data (closed network concept)

Chapter 3.3.4 (Figure 29) describes one way of separating safe and non-safe functions in a critical network using ECN switches with safe and non-safe ports – another way could be attach SIL4 conformant devices to the ETB directly. This could leave the SIL4-related safety domain strictly to the ETB. The ETB makes up an own IP network – the addressing of the nodes is crucial, though.

3.4 Other industry examples

3.4.1 Space

3.4.1.1 Networking Standards in Space Industry

Space applications are split into three classes: launchers (rockets), spaceships/flight crew modules, and satellites. Due to long product lifecycles for embedded platforms and integrated systems which are typically in operation for 30-40 years, the majority of current systems in operation use Firewire, SpaceWire or RapidIO for distributed processing and MIL-1553 for controls.

In the past, launchers would use MIL-1553 or analog systems, whilst satellites would combine point-to-point RS422, Firewire/RapidIO/Ethernet/SpaceWire for high bandwidth and distributed payload/sensor processing, while CAN and MIL-1553 are used for platform controls.

3.4.1.2 Space Avionics Architectures

Typical architectures are designed as triple voting architectures, with three computers collecting data from many sensors. Distributed fault-tolerant real-time systems as they can be known from the past are depending on the application field used, designed to be single or dual fault tolerant leading to architectures with three or four independent communication channels[3]. Such as space shuttle avionics [34] in Figure 35.

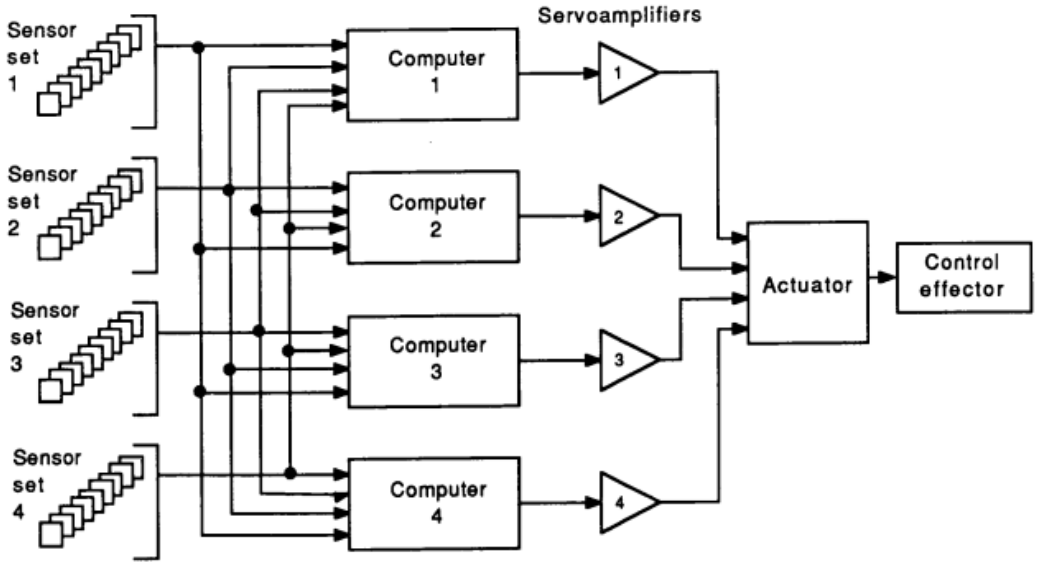


FIGURE 3-3.—Baseline system approach.

Figure 35. Space Shuttle Avionics

The respective on-board computers of such architectures are designed in a triple- or quad-voting paradigm meaning that the data of the three or four independent networks is received by all on-board computers to ensure a consistent system state view for each of them as illustrated in Figure 36.

However each on-board computer (OBC) is only able to transmit on one of the channels to ensure that a fault of a single on-board computer will not propagate to another channel. In case of a fault in one of the on-board computers a maturity-voting of the on-board computers (2 out of 3 or 3 out of 4 depending on the fault-hypothesis[1]) will trigger a physical disconnect of the sending line of the faulty OBC. The communication on the network needs to be very

simple so that all other receivers (typically sensors and actuators which can be very simple devices) are able to vote between the data they are receiving on the three or four independent network channels.

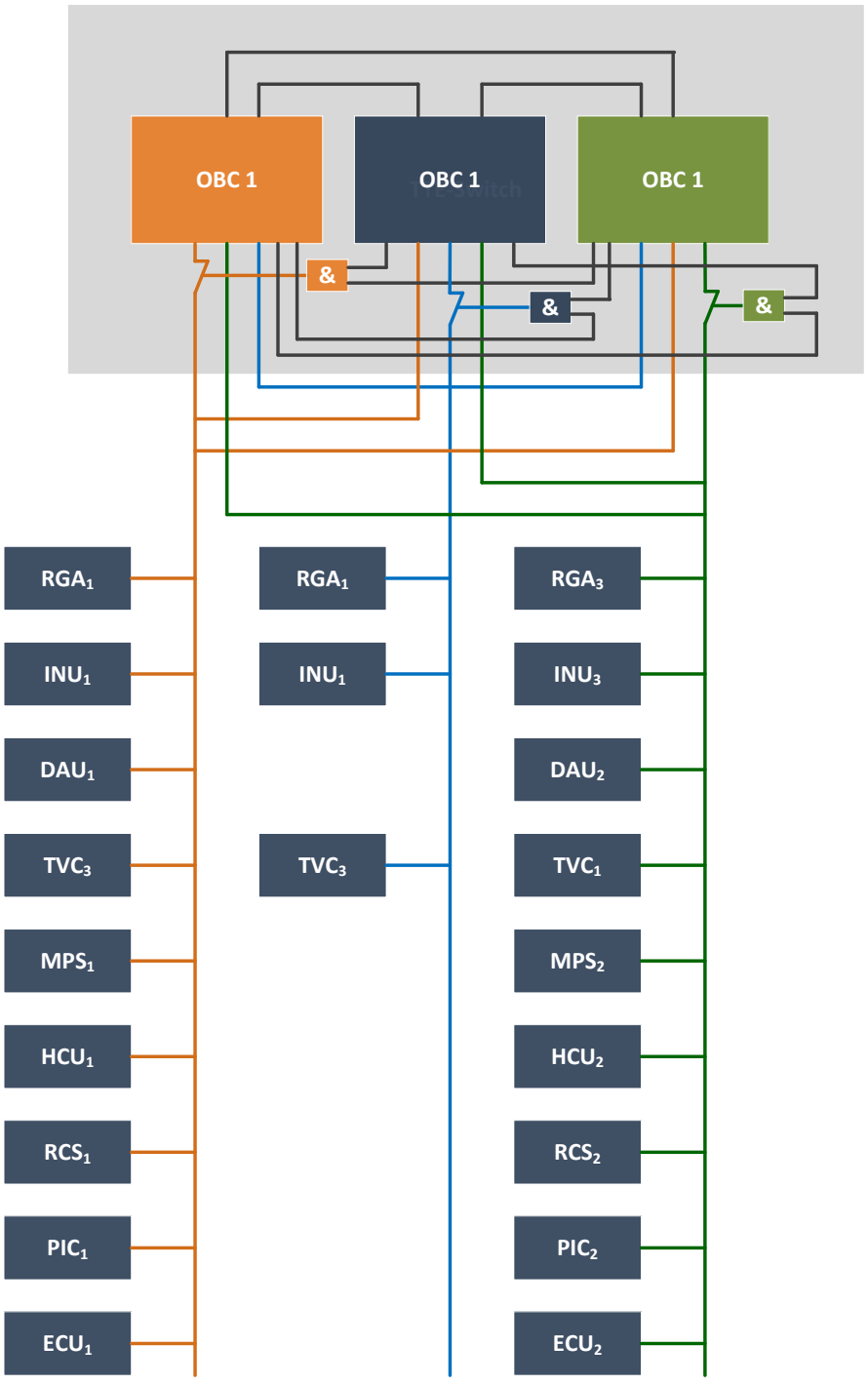


Figure 36: Tripple-Voting Architecture

If these independent communication channels do not provide a deterministic high speed capability, the OBCs need a separate communication interface to exchange the voting results and for the synchronization. This is typically achieved via an additional high speed point to point communication. Depending on the application a physical separation of the OBCs for fault-tolerance reasons can lead to very demanding environmental requirement such as EMC or lightning to fulfill the high bit error rate needed.

The approach of currently used distributed fault-tolerant systems in spacecraft applications can be described as following:

- Replication of the OBCs depending on the fault hypothesis e.g. one fault-tolerant, byzantine faults, …
- Three- or four independent communication channels to the sensors and actuators
- Only one OBC is master of one communication channel
- A maturity voting allows the healthy OBCs to remove a faulty OBC from its transmit channel
- All others are listening to have the same state information
- An additional networks e.g. point to point is used to exchange voting data and to synchronize the OBCs tightly allowing to vote on timely precise data

Such architectures based on different communication technologies result in:

- The support of different communication technologies for deterministic sensor and actuator communication, for the exchange of voting results and for the synchronization of the OBCs
- Software drivers for each of them
- The support of different physical layers e.g. MIL1553, RS422, IRIG-B, …
- Different and lots of harness need for the different interface

With NASA using TTEthernet switches which can integrate ARINC664 and SAE AS6802 services, the technology baseline supports integrated modular spacecraft architectures, which are closer to commercial IMA 2G in capabilities, and can also integrate “open system”(non-critical, a priori unknown traffic profile) and “closed system” (well defined, periodic critical controls and alarms) functions with full isolation between critical and non-critical functions.

3.4.1.3 Future Outlook on Space Avionics

New human-rated space avionics and commercial launcher platforms are based on triple-redundant SAE AS6802 and Ethernet architectures. Below you can find a simple generic launcher architecture which can contain several independent networks (dual or triple). The interface (dotted line) represents a separation point for different stages.

The system contains at the beginning (before launch) of ground, 3rd, 2nd and 1st stage, service module and crew module/capsule. After launch it separates from the ground systems, and loses additional systems during the flight. At the end of mission the service or crew modules can connect to other modules or space station components.

This approach enables the integration of a large number of computers and sensors, and depending on the system design philosophy and key principles, the system can be defined with dual or triple redundant Ethernet network. Furthermore, by configuration the system can be adapted for incremental integration of new capabilities or modified for reuse in new applications.

New concepts consider next generation reconfigurable architectures with triple-redundant computers and Ethernet networks for synchronous communication. The “composability” is relevant to avoid any additional V&V issues after we test the system in isolation, and connect them together – either as modular vehicles or functional subsystems.

Finally a high-level example with triple redundant computer and dual redundant deterministic Ethernet networks is provided.

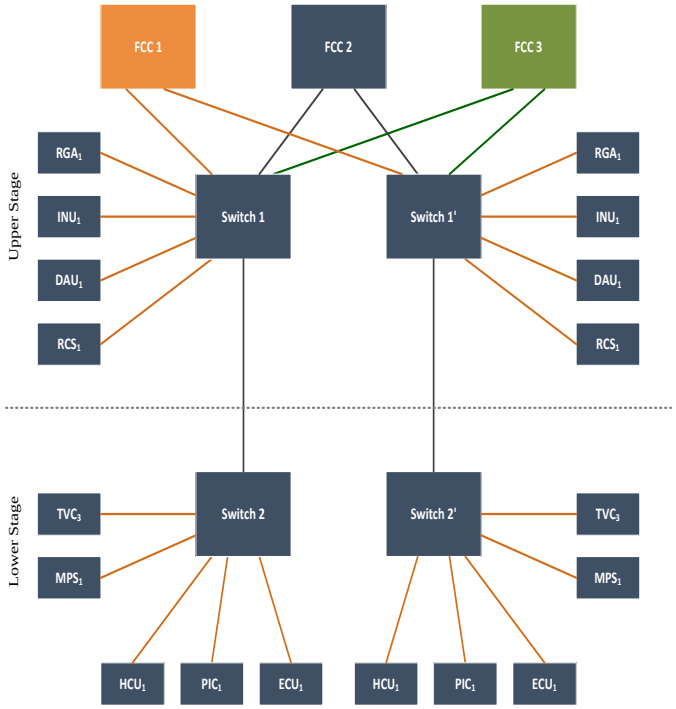


Figure 37: Example Launcher Architecture using Ethernet Technology with SAE AS6802 services.

3.4.2 Energy Production Automation

3.4.2.1 Networking standards in energy and substation automation

3.4.2.1.1 Nuclear Energy

Nuclear energy systems are very conservative in design, and very rigorous about any potential common causes and related faults. Their architectures and topologies are not similar to anything built into vehicles or transportation systems, and they keep functions and systems separated.

Ethernet is used in safety-critical and real-time (control loops at 20Hz cycle) nuclear applications since 1990s by Westinghouse/Rolls Royce. All other controls today still use analog lines or fieldbus based systems. Token Ring (or logical token ring i.e. token bus!) which is today obsolete, is used as deterministic bus with token passing. IEEE802.5 Token Ring is disbanded in 2008 due to lack of interest. Cisco discontinued sales of Token Ring switches in 2003. Nervia is a Token Ring network [35].

25 years ago, token ring was anticipated as a deterministic network of choice, in comparison to Ethernet bus and non-deterministic CSMA/CD approach to network bandwidth sharing. Unfortunately the approach deployed in Token bus would not be applicable in complex integrated systems, as the token passing has its own challenges:

- one malfunctioning station can create a problem for the whole network (solution: dual ring, physical star)
- token can be lost or suddenly multiplied, a station with token can die and delay operation
- adding devices can affect network operation and cycle speed
- rings with many nodes (>30) nodes can be too sensitive and can to frequently loose frames (see study on nuclear networks [25], [26], calculated token loss of 2% in Nervia)

A TUV report from 2005 [27] describes different variants for white, gray and black channel communication in nuclear domain and anticipates Token Ring as obsolete without going into its technical and safety properties.

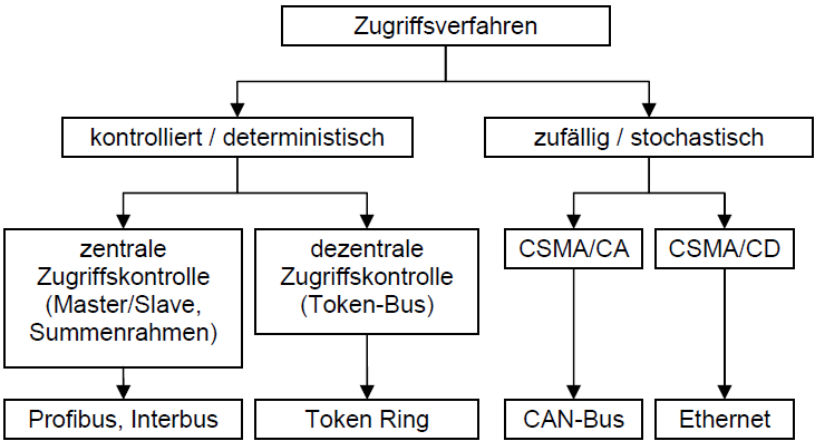


Figure 38: Nuclear industry anticipation of deterministic networking

For nuclear industry which is using asynchronous communication and models of computation, the most interesting networking alternative could be a certifiable network with ARINC664 services, due to DAL A design processes applied in the design of switches and networks.

3.4.2.1.2 Substation automation

Energy automation is relatively conservative and slow-moving market, with systems in operation which can be older than 70 years. With IEC61850 [27], substation automation has started with design and integration of embedded platforms which share computing and networking resources, and become similar to advanced integrated systems designed for aerospace and defense applications.

Their approach targets the interoperability and relies largely on an object model to configure the system and a set of mechanisms which can be used to support the transmission of sampling values and low-latency alarm messages (GOOSE) with bounded latency. Furthermore this standard supports a set of reporting schemes.

GOOSE uses VLANs and priority tagging and defines separate virtual networks within the same physical energy station network with appropriate message priority level. GOOSE enables retransmission with varying intervals to prevent message loss or increase chances of messages getting through the network. The logic behind GOOSE retransmissions is to improve integrity, and get critical messages to the end-station within the 3-sigma probability of delivery within a 4 msec event horizont. With three messages the probability of missing a message is lower than the fault in CRC calculation probability. This also represents a sort of temporal redundancy, which can be helpful in complex EMI (electro-magnetic interference) environments.

In this case GOOSE solves one problem, but IEC61850 is very limited in terms of traffic isolation (VLANS do not really solve it in case of faults!) or temporal guarantees and determinism.

The system is as deterministic as it is verified in the field, but the platform does not offer any real support for deterministic platform design, any system modification can make the V&V effort obsolete. The system configuration and its maintenance is critical as small configuration errors can have hazardous consequences. This can be a disadvantage in terms of system integrity, availability and sustainable maintenance.

Typical substation can have 40-50 end stations (or IEDs – integrated electronics devices), and in very large systems which can be tightly integrated with power plants the number of nodes can exceed Nx100 IEDs.

3.4.2.1.2.1 System architectures and future outlook

Substation has a hierarchical architecture with two backbones - station bus (station LAN) and process bus (field LAN).

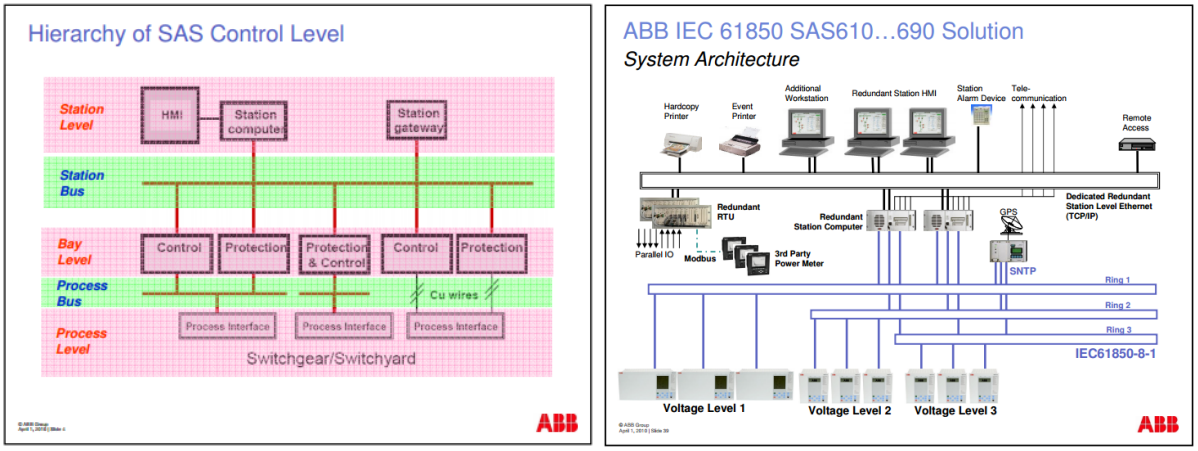


Figure 39: Station and process bus for substation automation architectures [28]

While the companies such as Schweitzer Lab and GE rely more on proprietary optical TDMA networks and provide Ethernet connectivity on IEDs and IEC61850 object model, ABB and Siemens use PRP/HSR and Ethernet switching networks with optical and copper physical layers.

In the future, there is a trend toward Ethernet backbone for station and process bus. This is driven by customer (electric utility companies) requirements, as they prefer more open Ethernet-based solutions

Another long-term topic is the convergence in substation automation and integration of the station and the process bus into a flat integrated architecture. This convergence will require a different set of mechanisms, which are closer to commercial aerospace or space capabilities, to be effective.

3.4.2.1.3 Wind energy controls

Different solutions has emerged in domain of wind energy controls – they are based on EtherCAT, Profinet, and other industrial Ethernet solutions.

In a specific niche of large off-shore turbines the market leader Vestas deploys the TTEthernet (SAE AS6802)-based system which can be considered similar to aerospace architectures in terms of complexity and the number of integrated end stations. The objective of such an architecture is to support high availability and safety at once.

Wind turbine control systems are in many cases a complex mix of fieldbus and point-to-point networks, with safety and control functions physically separated. This can lead to expensive integration, higher material costs and limited access to data from critical systems. By integrating a system in a flat integrated and modular architecture it is possible to reduce complexity and simplify reuse and adaptations for different platforms.

However it must be noted that the reasoning of this industry niche is not so much on the safety side as in aerospace – it is about operation and construction optimization and

scheduled maintenance [40], which are essential lifecycle costs drivers in off-shore applications for the largest turbines under harsh environmental conditions.

3.4.2.1.4 Other

Energy systems use Ethernet for different tasks in different configurations, and in many cases with point-to-point connections (synchronous 100MBps or 1GbE links) which are sufficient for low-cost centralized controls for energy production systems or power plants. Such systems typically cannot be categorized as highly integrated.

3.4.3 Defense

3.4.3.1 Historical Overview

Defense industry was one of early adopters of Ethernet in aerospace and ground systems due to high-bandwidth capabilities. Other high-bandwidth networks such as FibreChannel and Firewire have successfully competed with Ethernet for a long time, but Firewire haven't made it in commercial markets and is obsolete today. For FibreChannel the gap is closing with new 10GbE to 100GbE Ethernet variants, and it will become obsolete over the next 5 years. A common converged fabric that can be virtualized and turned into software-defined network is very desirable for scalability. Ethernet meets that goal very well, while Fibre Channel does not support it [29]. Therefore, legacy high-bandwidth defense applications are ultimately going to slowly converging on Ethernet over longer term.

Early fault-tolerant integrated mission systems have been built specially adapted Ethernet variants based on HSRB (High-Speed Ring Bus) [30], defined in SAE AS-2 committee. This fault-tolerant, real-time high speed (80MBps) data communication standard defined a counter-rotating ring topology with the use of a token passing access method with distributed control (time-division) on network access. Particular attention has been given to low message latency, deterministic message priority and comprehensive reconfiguration capabilities. However it is used only in a handful of advanced integrated systems developed in late 1980, early 1990s (e.g. Comanche Rotorcraft). This standard does not have any similarities with full duplex switched Ethernet, and it is completely obsoleted by new developments and proprietary adaptations of full duplex switched Ethernet. New avionics systems are simply upgraded every 20-25 years, and all old hardware and networks are replaced together with legacy networks.

3.4.3.2 Limitations in the use of standard Ethernet

VLAN priorities or more complex Ethernet capabilities are rarely used for avionics applications, as this guarantees simpler obsolescence management and independence of the device implementation. Furthermore proven MIL-1553 databus is used for critical controls and deterministic applications. As military applications do not dictate the development of semiconductor components and networking technologies for the last 25 years, this is a workaround to blend high-bandwidth communication with determinism in one system. This may translate into other problems in terms of scalability and reuse for similar systems with slightly different capabilities. Other applications such as radars and payloads typically use synchronous point-to-point connection links or tactical switches with standardized components.

3.4.3.3 Future Outlook on Defense Aircraft and Vehicle Architectures

Currently, a new class of more integrated vehicle, aircraft/rotorcraft and vehicle electronics architectures emerges with Gigabit-Ethernet at its core, and new considerations on higher levels of integration, interoperability and embedded virtualization [31].

Defense industry is not in the position to mandate commercial or embedded networking developments since 1970s, and relies largely on existing commercial IT and automotive technologies, accompanied with additional defense-specific features.

An example of a generic mission and controls system architecture (Figure 40) with deterministic Ethernet is presented [32] in the following figure. Ethenret networks provide access to all sensors/payloads and vehicle subsystems, and can be configured for different applications. It looks like there are two physical deterministic Ethernet backbones, but this is not correct – it is one flat network with backplane and backbone sections.

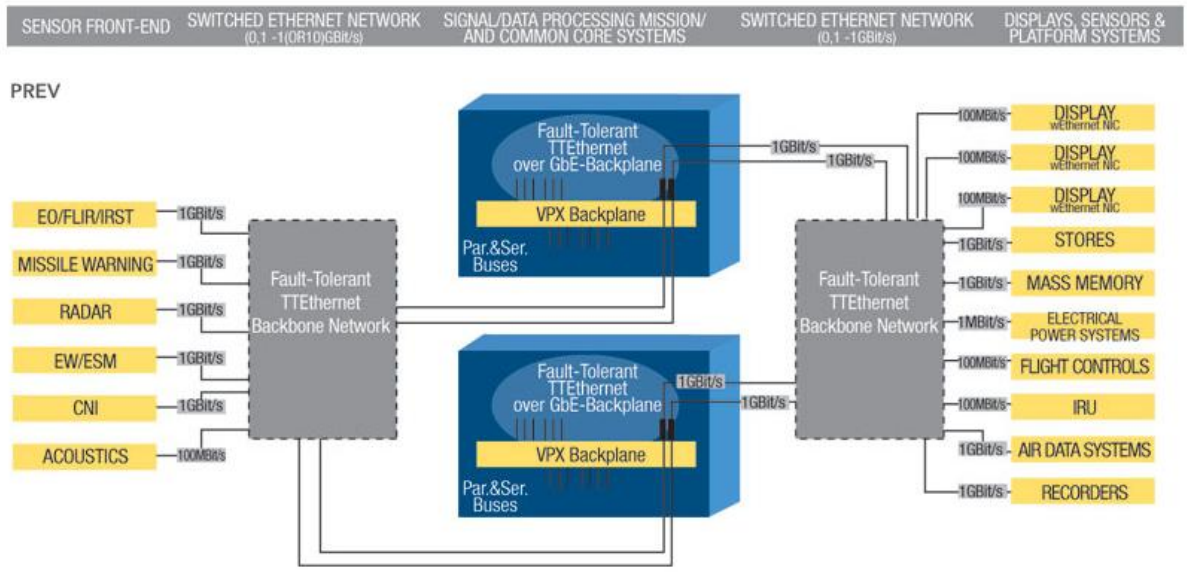


Figure 40: Generic mission and control architecture with sensor and vehicle systems integration

3.5 Conclusion

The described approaches to network and platform design reveal the differences between fail-safe and fail-operational approaches, and integrated and federated architectures. Ethernet networks have been typically used for less critical applications, in parallel to dedicated (also redundant) networks defined for critical functions. The major roadblock has been the resource sharing and the capability of Ethernet to handle sufficient isolation of the traffic for different functions. Typically for fail-operational system design with Ethernet tailored for highly critical integrated applications, both safety (with high-integrity and fault isolation) and availability are considered – meaning that redundancy, monitoring and handling of different faults should be carefully designed to match required safety and reliability objectives.

Defense industry focused more on mission-criticality, availability or survivability of systems. Space industry focus is on mission-criticality and crew safety with new human-rated avionics systems, typically with triple and quad redundant systems.

In the past, the automotive industry has not used redundant systems due to costs and the predominantly assumed fail-safe nature of automotive systems. The automotive industry

progresses toward highly-critical integrated architectures and designs implicated by the advent of more-autonomous driving applications and respective safety measures.

The railway industry hosts SIL2 functions on separated Ethernet networks, and integrates different dedicated MVB- or CAN-based controls, and dedicated safety lines.

In the nuclear industry, until now, Ethernet networks have been accepted in critical applications, but only as one of dissimilar protection systems.

The aerospace and space industry have worked with advanced integrated systems since the early 1980s, with fully established mechanisms and approaches for IMA (Integrated Modular Avionics) since early 1990s. It uses double redundant Ethernet networks, but also uses additional dedicated systems for flight or power generation controls.

Chapter 4 Distributed Embedded Platform Integration for Critical Applications

4.1 Introduction

There is a plethora of mechanisms which support the system integration determinism, system redundancy management, system state agreement and interactive consistency, robust partitioning, layering and software abstraction, startup and recovery. All of them rely on system integration technology capabilities and directly impact the system availability and dependability. The distributed system capabilities in terms of high reliability, availability, integrity, safety, maintainability are all related to system integration technologies.

A solid grip on the coordination and interfacing in complex systems is required to control resource sharing in the system. This can be accomplished by space and time partitioning of computing and networking resources. The alignment among different layers is essential for system resource sharing among many distributed functions. The approach on access of system networking and computing resources depends on selected models of computation and communication, which also mandate the approaches to network and application alignment, or even system design methodology.

4.2 System Integration and Integrated Embedded Platforms

4.2.1 Objectives: Scalable Embedded Computing and Networking for Critical and Non-Critical Functions

In an ideal case, system integration technology should not limit the possible design space for advanced system architectures. The definition of determinism would support predictable (with “gray”/”white” channel communication) operation for all critical functions under any bandwidth use scenario. System integration technology should help to simplify system design, reduce complexity and allow scalable integration.

The set of integrated functions includes functions with known performance and resource requirements (see Figure 41):

- periodic real-time controls (upto 100Hz -Nx1kHz sampling rate)
- data streaming (continuous diagnostics, periodic multimedia streams …)
- transient alarms, safety functions with deterministic response time in 100µs to Nx1 milliseconds
- sporadic applications (maintenance, updates, inspections, and event driven applications executed in sporadic manner)

Critical control functions typically exchange data periodically, and shall have a non-blocking behavior without busy waiting or resource starvation which can cause unpredictable system performance and affect other integrated control functions. The objective of such embedded control design is to avoid unintended interactions if some functions fail or arbitrary access embedded resources. Similarly periodic data streaming multimedia application provide logical links among functions with well-understood behavior.

Transient and event-driven alarm and protection functions shall have guaranteed deterministic performance with maximum latency, when triggered by some special critical system event and transients.

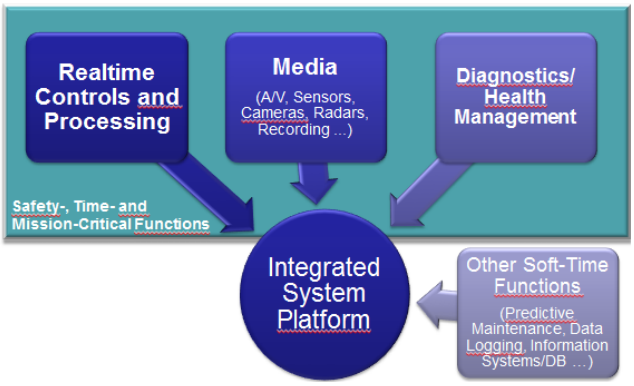


Figure 41. Application types integrated in reconfigurable “embedded clouds”

In addition, non-critical soft-time functions with unknown resource requirements and workload profile can be integrated. In this case we can expect transient bursts of activity at any time, but the impact of those functions on critical functions must be prevented by embedded platform properties and models of computation and communication, described in the next chapter.

4.2.1.1 Scalable and reconfigurable hosting of software functions

Based on previous considerations it is obvious that both synchronous and asynchronous Ethernet packet-switching can enable deterministic system operation, but the definition of determinism and QoS capability will be different. The resulting system architectures will have different capabilities, which may determine the spatial proximity between the controlled objects and the computers with hosted software functions. Furthermore, the impact of non-critical applications on critical applications must be carefully addressed.

4.2.2 Integrated Modular Embedded Computing and Networking Platforms

With the large scale integration of different functions, the platform becomes a subsystem, which does not perform any application-specific function.

The Integrated Modular Platform hosts application functions and provides specific services to critical and non-critical applications, in order to establish robust software abstraction and provide all resources and timely information (sensors, global variables) access to applications.

Integrated Modular Platforms are a part of an integrated system. The configuration of integrated modular platform components, adapts the Integrated Modular Platform to a specific use case and topology or architecture.

Safe4RAIL deals with Drive-By-Data and system integration baseline as well as an application hosting framework and software platform services. The drive-by-data part is focused on anything related to system integration, interfacing and information transfer from one application partition to another application partition *in the networked system*.

Drive-by-data thus focuses on all system integration capabilities required to define an Integrated Modular Platform which can host different TCMS, door control, braking, safety or other non-critical functions in one system. An Integrated Modular Platform does not depend on applications, and can be separately certified. Modular applications hosted on an integrated modular platform can be tested in isolation and integrated on the system, without unintended interactions and interdependencies.

4.2.2.1 New roles in design, integration and maintenance of integrated architectures

Integrated modular platforms host different systems, and it is not possible to make separation of roles and functions in functional subdomains, as it has been done in the past.

Therefore, different roles emerge with the definition of integrated modular platforms.

Integrated system integrators (ROLE 1) integrate and verify applications from different application suppliers (ROLE 2) on top of the integrated modular platform provided by platform providers/integrators (ROLE 3).

All three roles can reside within one company, or may be split between OEMs and 1st Tier system as integrators, or OEMs/1st Tiers/3rd parties as application suppliers.

The generic integrated modular platform consists of SW/HW components (modules) such as RTOS, middleware and network devices. They are delivered by the platform or component/module suppliers (ROLE4). This role can be established by OEMs, 1st Tier or a dedicated integrated platform supplier.

OEMs are considered responsible to submit a request for homologation/certification to regulatory authorities. The certification and homologation authorities (ROLE5) work with OEMs to certify and homologize the whole system, and may rely on the feedback from all other roles.

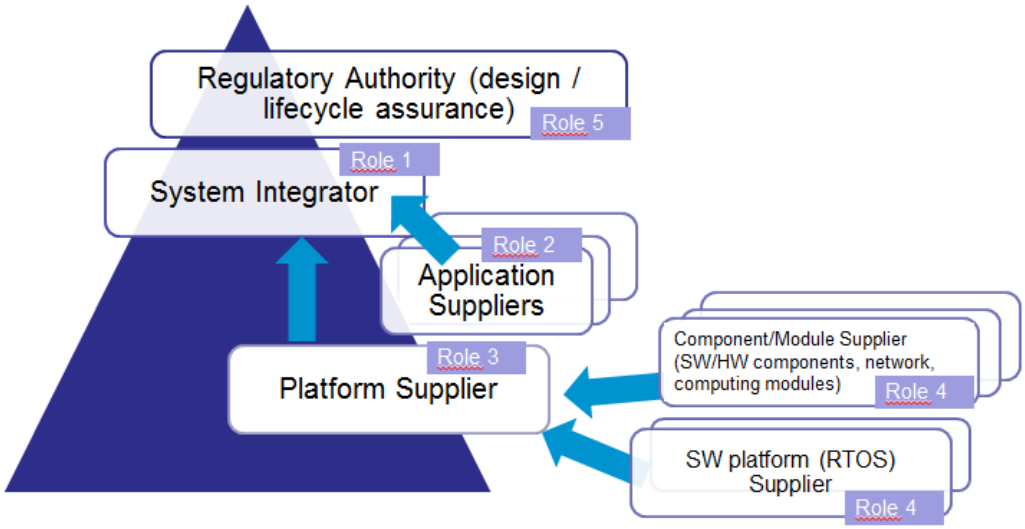


Figure 2. Application types integrated in embedded clouds

4.3 System Integration and Safety Assurance

Safety case and safety assurance are determined in relation to industry-specific concepts of risk, i.e. it must be shown that the probability of catastrophic failures and unacceptable risks is reduced below minimum tolerances.

In different industries, such risks are assessed using different assumptions, but they typically lead to a similar set of methods and processes to ensure that system faults are avoided and the system safety is suitable for a specific application. The gradation and brief comparison of different safety system integrity levels is provided in [36].

Domain	Domain specific Safety Levels				
General (IEC-61508) Programmable electronics	(SIL-0)	SIL-1	SIL-2	SIL-3	SIL-4
Automotive (26262)	ASIL-A	ASIL-B	ASIL-C	ASIL-D	-
Aviation (DO-178/254)	DAL-E	DAL-D	DAL-C	DAL-B	DAL-A
Railway (CENELEC 50126/128/129)	(SIL-0)	SIL-1	SIL-2	SIL-3	SIL-4

Figure 42. A high-level comparison of safety integrity levels in different standards

An appropriate real-time networking architecture can tolerate and mask faults before they generate further errors and system failure. The communication medium dependability can be seen as a “white” or “black” channel, or something in between (“grey” channel).

4.3.1.1 White Channel vs. Black Channel

In systems where a fail-safe state can be turned off on any fault of hazard, it is possible to design critical applications by using a black channel communication approach. In addition, the applications are designed not to rely on network for its operation, and may also include some backup or graceful degradation strategies.

With distributed complex (Ethernet-based) systems which host many functions on different computers, “white channel” designs can provide predictable performance, high integrity, availability and reliability required for safe system operation. To become certified, network components are designed using safety assurance processes which support the system safety objectives and high dependability, or the devices should have sufficient operating history in similar critical applications. “White channel” (Figure 43) will require also protocols services and network components to be designed in line with IEC 61508-2.

Figure 7 (a) White channel

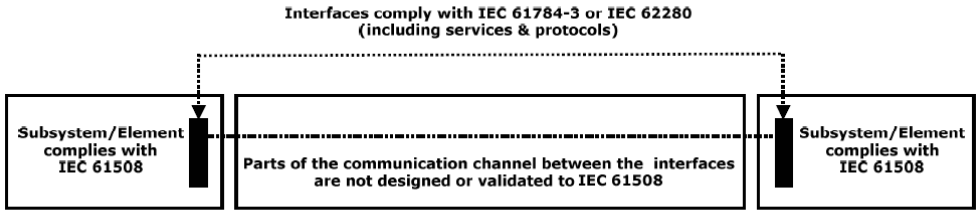


Figure 7 (b) Black channel

Figure 43. IEC61508 Black channel vs White Channel (ref. IEC61508-2:2010)

In aerospace industry, the continuous operation is the only safe state, so the systems are fail-operational. DAL A applications with Ethernet require “white channel” with mandatory deployment of DO-254/DO-178C DAL A design practices (similar to SIL4 safety design assurance practices for software and electronics) on network devices, switches and end-stations.

The platform for complex IMA architectures is seen as a subsystem which provides a “hosting” service to all other functions. Any system integration faults and errors can create potential dependencies between functions hosted by different hardware modules. Therefore an uncontrolled failure of one Ethernet switch could potentially induce a partial failure of the

channel, platform and other hosted functions. It is necessary to assess the effects of cumulative functional failures and effects due to single and multiple resource failures at system level [38].

For IMA architectures, the aerospace industry avoids the use of components which do not have well-understood internal architecture and functionality, can fail unpredictably, and do not provide unambiguous fault diagnostics. Great value is placed on the evidence showing that safety analyses and design assurance for network components has been completed according to best practices and can be accepted by regulatory bodies.

4.3.1.2 Highly-dependable networking and system integration

In advanced integrated architectures, system integration represents the 'glue logic' for many system functions. Therefore for highly critical applications the network is designed to have less than 10e-9 (or 10e-10) failures per operating hour. The components used in such a system should use components with less than 10e-6 transient and permanent failures/hrs.

For SIL 4 functions, the probability of functional failure is less than 10e-8/hour, and if transmission uses 1% of the permissible probability of failure, the probability failure rate for the safety bus system must be 10e-10/hour. By selecting appropriate CRC polynomials for the intended frame length, the resulting residual error probabilities of the undetected corrupt data packets may meet the required limits, but for an FCS of 24 bits, the likelihood that a different data word will produce the correct CRC is 6x10e-8 (under the assumption of uniform distribution), which is typically questionable. Therefore additional data integrity mechanisms might be required for Ethernet-based networks (e.g. additional CRCs for application data fractions). The data integrity is not the only cause of communication failure - there are many other issues at device architecture and protocol implementation level, which must be taken into account (see Figure 45).

All higher software layers and mechanisms are built with the assumption of high dependability of the underlying system integration layer, and the loss of communication would represent a total system failure. For SIL4 functions, the loss of the communication on the critical path could cause a catastrophic failure, unless some additional mitigation mechanisms can be deployed.

Integrated Modular Platforms need to provide capabilities and services to host SIL0-SIL4 functions.

The network shall implement the mechanisms to support the fault avoidance, prevention, detection, isolation, prediction and recovery Figure 44). The large majority of errors should be detected to initiate a recovery process, avoid arbitrary byzantine faults and initiate orderly fail-silent power-off of the component and communication channels.

With system integration and networking capability at the center of an safety critical application the failure rate of message data and the availability of the communication process will be estimated to support SIL-rating of hosted functions. The failure rate calculation may go deeper into the definition and implementation of protocol mechanisms and complex device architecture.

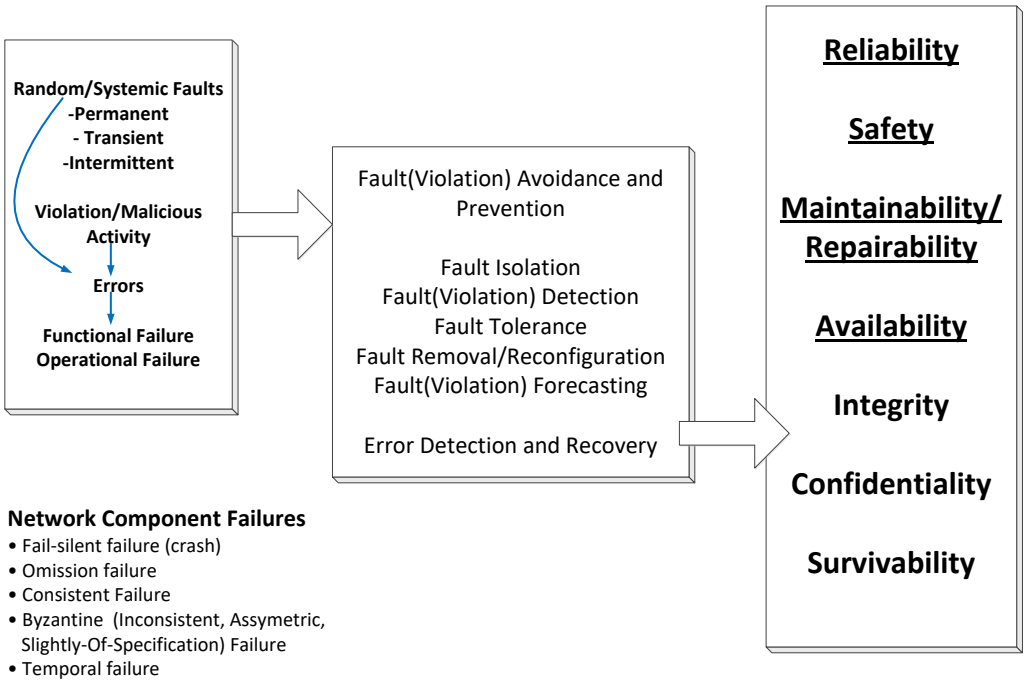


Figure 44. System attributes relevant for the design of advanced integrated architectures

Measures / Comm. Error	Seq. Number	Time Stamp	Time Out	CRC / Checksum	ACK (not applicable in non-blockings integrated systems)	Membership Control	Identifiers for Sender and Receiver	Path replication	Time Replication	Periodicity checking (AFDX)	Time Policing and Time-Aware Shaping (TSN/AS6802)	Priorities	Data Protection / Crypt
Repetition	Y	Y			na					Y	Y	na	
Loss	Y	Y			na			Y	*		*	na	
Insertion	Y				na		Y	*		*	Y	na	
Incorrect Sequence	Y	Y			na			*			Y	na	
Corrupted Message / Incorrect Data				Y	na			*	*			na	*
Delay		Y	Y		na					Y	Y	na	
Masquerade	*			Y	na	Y	Y					na	Y
	*=	limited effect											
	Y=	effective											

Figure 45. Communication errors and measures

Critical functions using system integration are typically designed to be tolerant against transient omission failures and limited data loss. As an example, deterministic avionics networks are not designed to be lossless. While great care and effort is invested in configuration which supports congestion-free and lossless communication, it is not assumed that every frame will arrive to the receiver. In the case of network timing faults or transient

disturbances, packet drops are possible, and packet statistics are available on every Ethernet switch and end-station.

4.3.2 Resource partitioning for critical systems evidence for robust partitioning and non-interference

One of the key issues in system integration for safety critical systems is the evidence of non-interference between functions, or in other words it the robust partitioning of system integration resources, i.e. network bandwidth partitioning.

For integrated architectures, there is a number of relevant aspects which should be taken into account. First of all the embedded platform is a system on its own, with its only function being to host all other critical and non-critical functions.

By enabling a system-level time partitioning, it is possible to slice the resources in an embedded system, to have a system-wide hard real-time performance for critical functions and to use the remaining resources for other less critical applications. At the system integration layer it is necessary to support time-partitioning to be able to integrate mixed criticality traffic and both switches and end-stations shall support high integrity and availability architectures.

Deterministic Ethernet switches need an internal architecture with constant technology latency and full control over the network bandwidth use, as well as the ability to monitor every critical dataflow. Furthermore the switches must be able to synchronize with all other network devices in the system to enable desired temporal performance. Its architecture must ensure isolation of critical streams and monitors the switch buffer utilization by different dataflows, and prevents per design any unintended interactions or side-effects which can cause an application to lose or delay data. Switches must support the network synchronization with defined upper bounds on timing and failure. Switches provide a selection of mechanisms and statistics on transient faults, and if required enable the use of external monitoring to support fail-silent operation on faults. This mandates the availability of fine-grained fault detection mechanisms on the switch, and its accessibility for external monitoring units.

Network isolation and separation relies on:

- Deployed synchronization mechanisms
- Deployed communication protocol mechanisms
- Internal fault detection and health monitoring
- Safety conform implementation of communication protocols and synchronization mechanisms
- Internal switch architecture designed for safe and secure operations of implemented communication protocols, fault detection, health monitoring, and memory partitioning and protection

Very similar requirements are needed for network interfaces. They can be designed as low-integrity end-stations for simple sensor data transmissions over multiple channels, or as high-integrity end-stations which transfer many critical dataflows from one or many functions, and require design approaches similar to the switching devices. Their complexity depends on use cases, and target applications.

Chapter 5 Summary and Conclusion

System integration represents a core capability required for the design of advanced integrated architectures. 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, certifiabilty, robustness and system lifecycle costs.

This document provides an overview of state-of-the-art in relevant technologies for deterministic high-bandwidth networking and reveals different use cases in transportation industries aerospace, automotive, railway, and space.

Proven core technologies for deterministic Ethernet integration which could satisfy requirements of advanced integrated architectures for mission-, time-, and safety-critical applications are described in ARINC664 and SAE AS6802. Their implementations include the properties which correspond to “white channel” communication approach, and provide congestion-free communication with full control of temporal behaviour for all critical dataflows in the system. Formally verified and robust fault-tolerant distributed clock algorithms support the control of system time in the most demanding critical applications.

Currently, IEEE TSN (Time-Sensitive Networking) suite of standards is in development and it could further develop to gain the capabilities relevant for critical applications in automotive, industrial and IoT applications. Other technologies such as software-defined networking (SDN), DetNet or WDM can expand the range of system integration options in critical integrated systems over the longer term (10-15+ years).

With the objective to design scalable, reusable, reconfigurable and certifiable system architectures, system integration and Ethernet networking cannot be seen separately from the software platform. Well-designed generic integrated modular platform are designed as one subsystem, which provides all services and capabilities required for hosting non-critical and critical (SIL0-4) applications.

In addition to safety, the network and system integration security becomes more important. Security issues may lead to safety-related consequences and risks, which must be carefully managed and considered during the design of robust integrated modular platforms.

Chapter 6 List of Abbreviations

VL	Virtual Link a data flow with known QoS, bounded latency and/or jitter boundaries in ARINC6664/SAE AS6802 terminology
Black Channel	A communication channel without any known safety properties and capabilities. Middleware and functional application assumes it cannot rely on communication network for any safety-relevant activity, and provides mechanisms for the identification of communication faults.
Gray Channel	A communication channel with partially known safety and performance properties and capabilities, with devices and protocols already used in safety-critical applications with documented certification evidence and service history and/or with some evidence of design or validation according to IEC 61508, but not fully sufficient for the desired SIL levels. Additional safety checks are required in safety middleware.
White Channel	A communication channel has well understood properties relevant for safety applications, and consists of devices and communication protocols designed to specific safety assurance/integrity levels. Redundant safety checks in communication middleware may be conducted in the safety middleware, as an additional safety net.
More deterministic	Average latency is minimized for overprovisioned networks, but there are no guarantees on temporal behaviour or congestion management in the case growing bandwidth use or faults. A Cisco switch with VLANs would fit this definition.
Very deterministic	Traffic congestion is prevented. Maximum communication latency is defined, but there is a very limited control of jitter (max. jitter > avg. latency) or message order. A strictly deterministic data flow acts as an asynchronous point-to-point circuit with defined maximum latency.
Strictly deterministic	Traffic congestion is prevented. Communication latency is almost fixed and the jitter is tightly controlled (max. jitter < Nx1µs). The message order within the cycle is known. A strictly deterministic data flow acts as a synchronous point-to-point circuit.
IMA	Integrated Modular Avionics
2G	Second Generation (2nd Generation)
FCS	Flight Control System
OBC	On-Board Computer

ECU	Electronic Control Unit
SoC	System-on-Chip
MPSoC	Multi-Processor System-On-Chip

Table 3: List of Abbreviations

Chapter 7 Bibliography

[1] J. Duffy, „http://www.networkworld.com/article/2289826/ethernet-switch/133715-The-illustrious-history-of-Cisco-s-Catalyst-LAN-switches.html,“ *Network World,* 12 2013.

[2] ARINC, „664P7-1 Aircraft Data Network, Part 7, Avionics Full-Duplex Switched Ethernet Network,“ SAE ITC, 2009.

[3] M. A. S. V. I. L. Samar Dajani-Brown, „Formal Modeling and Analysis of the AFDX Frame Management Design,“ *Proceedings of 2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC),* pp. 393-399, 2006.

[4] SAE International, „http://standards.sae.org/as6802/,“ SAE Standards, Warrendale, PA, 2011.

[5] IEC, „IEC 62439-3:2016 Standard,“ 2016.

[6] Flexibilis, „Paralele Redundancy Protocol,“ 2015. [Online]. Available: http://www.flexibilis.com/technology/parallel-redundancy-protocol-prp/. [Zugriff am 11 2016].

[7] M. Azarov, „Approach to Latency-Bounded Ethernet,“ IEEE802.1 AVB Group, 2006.

[8] J. Koftinoff, „Max Channel Count in AVB,“ 2013.

[9] G. F. O. R. Heise P, „Deterministic OpenFlow: Performance Evaluation of SDN Hardware for Avionic Networks,“ *CNSM '15 Proceedings of the 2015 11th International Conference on Network and Service Management (CNSM),* pp. 372-377, 2015.

[10] N. Finn, „DetNet: Deterministic Networking Architecture,“ IETF, 2016.

[11] H. Sarry, „WDM LAN Optical Backbone Networks and Standards for Aerospace Applications,“ 2009.

[12] J. Walko, „Data centre interconnect drives optical module advances,“ 2016.

[13] „SCARLETT,“ [Online]. Available: http://www.scarlettproject.eu/publications/posters/SCA-SP0-THA_MNGT-SLDW-SCARLETT.pdf. [Zugriff am 11 2016].

[14] ASHLEY PROJECT, „Achley Website,“ [Online]. Available: http://www.ashleyproject.eu/the-project-summary. [Zugriff am 11 2016].

[15] U. F. T. F. Cary Spitzer, Digital Avionics Handbook, 3rd Hrsg., CRC Press, 2014, p. 848.

[16] M. K.-J. Guy Norris, „Boeing's 777-300 reliability figures are the best for a widebody

introduction," *Flight Global,* 15 Feb 2000.

[17] B. Zhang, „A Boeing 777 just crashed, but it's still one of the safest planes ever to fly," *Business Insider,* 2016.

[18] RTCA, „DO-297 Integrated Modular Avionics (IMA) Development Guidance and Certification Considerations," RTCA, 2005.

[19] A. B. a. P. C. A. Benveniste, „A unifying view of Loosely Time-Triggered Architectures," in *EMSOFT '10 Proceedings of the tenth ACM international conference on Embedded Software*, Arizona, USA, 2010.

[20] J.-B. ITIER, „A380 Integrated Modular Avionics: The history, objectives and challenges of the deployment of IMA on A380," Rome, 2007.

[21] G. D. T. Thomsen, „Ethernet for AUTOSAR," in *. 1st AUTOSAR Open Conference & 8th Premium Member Conference (Detroit, MI, Oct. 2008)*, Detroit, MI, 2008.

[22] B. A. A. Maier, „ Ethernet – The Standard for In-Car Communciation 2nd Ethernet&IP@Automotive Technology Day (Regensburg, Sept. 2012)," in *The Standard for In-Car Communciation 2nd Ethernet&IP@Automotive Technology Day* , Regensburg, 2012.

[23] B. A. K. Wittmack, „Introducing Automotive Ethernet – A Project Manager's Account," in *5th Ethernet&IP@Automotive Technology Day*, Yokohama, Oct 2015.

[24] N. A. C. Otero, „Automotive Ethernet – Enabler for Autonomous Driving," in *Automotive Ethernet Congress*, Munich, Feb. 2016.

[25] D. S. Bunzel, „Overview on AUTOSAR Cooperation," *2nd AUTOSAR Open Conference,* p. 19, 13 05 2010.

[26] AUTOSAR, „Layered Software Architecture," 2015.

[27] AUTOSAR, „Software Component Template," 2015.

[28] AUTOSAR, „Specification of BSW Module," 2015.

[29] AUTOSAR, „Virtual Functional Bus," 2015.

[30] AUTOSAR, „Specification of Module E2E Transformer," 2015.

[31] AUTOSAR, „Specification of Synchronized Time-Base Manager," 2015.

[32] S. A. L. Fiege, „RACE – ECAR," Munich, Jan 2014..

[33] I. A. E. Kenzler, „Einflüsse des Internets der Dinge auf Architekturkonzepte in der Automobilindustrie," TU Braunschweig, Mar. 2016.

[34] R. M. John Hannaway, „Space Shuttle Avionics NASA SP-504," NASA, Washington DC, 1989.

[35] J.-P. BUREL, „Modernisation of I&C system for ANP Dukovany by the use of computer-based equipment,“ in *CNRA/CSNI WORKSHOP ON LICENSING AND OPERATING EXPERIENCE OF COMPUTER-BASED I&C SYSTEMS*, Hluboká nad Vltavou, Czech Republic, 2001.

[36] H. C. N. D. H. K. I. S. K. Ki-Sang Song, „Analysis of Several Digital Networking Technologies for Hard-Real Time Communications in Nuclear Plant,“ *Journal of the Korean Nuclear Society,* pp. 226-235, 1999.

[37] NRC / Rolls-Royce, „SPINLINE 3 NRC Qualification - SPINLINE 3 Digital Safety I&C Platform,“ 2011.

[38] C. B. Christian Michas, „Zusammenstellung sicherheitstechnischer Anforderungen an Interfaces der Mess- und Stelltechnik in software-basierten Leittechniksystemen mit sicherheitstechnischer Bedeutung in Kernkraftwerken,“ TÜV Industrie Service GmbH, TÜV SÜD Gruppe, 2005.

[39] IEC, „IEC 61850:2016 - Communication networks and systems for power utility automation,“ 28 07 2016. [Online]. Available: https://webstore.iec.ch/publication/6028. [Zugriff am 11 2016].

[40] W. Bin, „Substation automation solution with IEC61850,“ 2010. [Online]. Available: http://www02.abb.com/global/seitp/seitp202.nsf/0/9276485464e7953cc125770300133d9a/$file/ABB+Substation+Automation+Solution.pdf. [Zugriff am 11 2016].

[41] P. Dvorak, „High availability advanced controls let Vestas lower material and operating costs,“ *WIndpower Engineering&Development,* 27 September 2016.

[42] J. O'Reilly, „The Sinking Fibre Channel SAN,“ *Network Computing,* 1st April 2016.

[43] SAE Standards, „http://standards.sae.org/as4075/,“ SAE International, Warrendale, PA, 1988.

[44] The Open Group FACE™ Consortium, „The Open Group Releases Future Airborne Capability Environment (FACE™) Technical Standard,“ The Open Group, Burlington MA 01803, USA, 2012.

[45] M. Jakovljevic, "Deterministic Version of Ethernet Offers Real-Time Performance at Low Risk," 2011.

[46] J. L. d. l. V. B. S. V. d. F. Eric Verhulst, „Eric Verhulst1, Jose Luis de la Vara2, Bernhard H.C. Sputh1, and Vincenzo de Florio3,“ 2014.

[47] M. Morel, „Model-Based Safety Approach for Early Validationof Integrated and Modular Avionics Architectures,“ in *F. Ortmeier and A. Rauzy (Eds.): IMBSA 2014, LNCS 8822*, 2014.

[48] H. Kopetz and J. Reisinger, "The Non-Blocking Write Protocol NBW: A Solution to a Real-Time Synchronization Problem," IEEE Proceedings of the 14th Real-Time Systems Symposium, 1993.

[49] Cisco, „Cisco 10GBASE Dense Wavelength-Division Multiplexing SFP Modules Data

Sheet,“ 2016. [Online]. Available: http://www.cisco.com/c/en/us/products/collateral/interfaces-modules/dwdm-transceiver-modules/data_sheet_c78-711186.html. [Zugriff am Nov 2016].

[50] S. Jolly, „Is Software Broken?,“ *NASA ASK MAGAZINE INSIGHT ,* pp. pp. 22-25, Oct 2009.

[51] P. Koopman and B. Upender, "UTRC Technical Report RR-9500470," United Technologies Research Center, 1995.

[52] „SPINLINE 3 NRC Qualification - SPINLINE 3 Digital Safety I&C Platform,“ 2011.

[53] G. B. Hermann Kopetz, „The time-triggered architecture,“ in *Proceedings of the IEEE (Volume: 91, Issue: 1, Jan 2003)*, 2003.

[54] M. Jakovljevic, „Synchronous/asynchronous Ethernet networking for mixed criticality systems,“ in *Proceedings of IEEE/AIAA 28th Digital Avionics Systems Conference*, 2009.

[55] J. G. Mirko Jakovljevic, „Deterministic high-bandwidth networks: Models of computation and system integration capabilities for actuator/sensor networks,“ in *Procedings of International Conference on Recent Advances in Aerospace Actuation Systems and Components (R3ASC)*, Toulouse, 2014.

[56] Institute for Basic Science, „This could replace your silicon computer chips,“ 30 July 2015. [Online]. Available: https://www.sciencedaily.com/releases/2015/07/150730081154.htm. [Zugriff am 2016].

[57] AUTOSAR, „Software Component Template,“ 2015.

[58] The Economist, „AFTER MOORE'S LAW: Double, double, toil and trouble,“ *The Economist,* 12 March 2016.

[59] Xilinx, „Xilinx UltraScale™ MPSoC Architecture,“ Xilinx, San Jose, CA, USA, 2014.

[60] Frost Sullivan, „Major OEMs and Tier I Suppliers Work Towards Making Ethernet A Backbone of In-Vehicle Networks,“ Frost Sullivan, 2014.

[61] O. Krieger, „How Ethernet Helps Building A Scalable Network Architectute,“ in *IEEE-SA Ethernet & IP @ Automotive Technology Day 2016*, 2016.

[62] Strategy Analytics, „Automotive Ethernet Market Set for Rapid Growth; Demand Forecast to Exceed 120 Million Nodes a Year by 2020,“ Strategy Analytics, 2012.

[63] Texas Instruments, „Multicore SoCs stay a step ahead of SoC FPGAs,“ Texas Instruments Inc, Dallas, TX, 2016.

[64] J. Mallet, „Functional Safety and the FPGA World,“ *EETimes,* 24 10 2016.

[65] IEC, „IEC61508:2010,“ International Electrotechnical Commission, 2010.

[66] RTCA, „DO-326A Airworthiness Security Process Specification,“ RTCA SC-216

Committee, 2014.

[67] C. B. Ngai, „Process Control Networks - Secure Architecture Design,“ Honeywell, 2015.

[68] J. Rushby, „The MILS Component Integration Approach to Secure Information Sharing,“ St. Paul MN, 2008.

[69] O. K. Rene Hummen, „Cyber security for TSN in modern automation networks,“ *Electronic Specifier,* 12 2016.

[70] B. D. Wilfried Steiner, „Automated Formal Verification of the TTEthernet Synchronization Quality,“ in *3rd NASA Formal Methods Symposium (NFM), Springer LNCS 6617*, 2011.

[71] ARINC, „ARINC664-P5,“ ARINC, 2001.