

A study on avionics and automotive databus safety evaluation

Andrew Kornecki¹, Janusz Zalewski², Janusz Sosnowski³, David Trawczyński³

Received January 2006

Abstract

This study presents an approach to evaluating safety of computer databuses for use in avionics and automotive industries. It responds to the need for qualification and certification of electronic devices used in safety-related applications. Some of these applications, such as fly-by-wire systems on the aircraft and steer-by-wire systems in cars, are reviewed briefly, first. Then the process of designing for safety is discussed, involving an outline of criteria for databus assessment with respect to the risk involved, followed by descriptions of respective approaches to hazard analysis and failure mode analysis. After presenting typical characteristics of major databuses used in avionics and automotive applications, including ARINC 429, 629 and 659 databuses, MIL-1553, IEEE-1394, CAN, TTCAN, FlexRay, TTP/C, Safe-by-Wire, and others, experimental results are presented and discussed for data throughput and latency evaluations with regard to safety in databus technologies. This is followed by a discussion of databus scheduling experiments for CAN and TTCAN networks with fault injection, which show significantly better databus utilization in case of TTCAN, even though CAN databuses can operate with higher performance in a fault-free environment.

1. Introduction

A definitive trend can be observed across the entire field of transportation that modern electronic systems applied in vehicles tend to be increasingly distributed. Communication among the smart sensors, smart actuators, and independent control units becomes a necessary part of system design [1]. Distribution of functions, control algorithms and data among devices over the in-vehicle networks pose new problems for design engineers, especially with respect to safety assurance for drivers, passengers and the general public, in case of unexpected failures of associated electronic or software components.

¹ Embry Riddle Aeronautical University, Dept. of Computer and Software Engineering, Daytona Beach, FL 32114, USA; kornecka@erau.edu

² Florida Gulf Coast University, Dept. of Computer Science, Ft. Myers, FL 33965, USA; zalewski@fgcu.edu

³ Warsaw University of Technology, Institute of Computer Science, Nowowiejska 15/19, 00-665 Warsaw, Poland; jss@ii.pw.edu.pl and d.trawczyński@ii.pw.edu.pl

The concern for safety in avionics and automotive applications means that assurance must be provided that computer hardware or software does not contribute to situations, which may cause loss of life, injuries or significant property damage. One aspect of this concern is the design and use of short-range networks known in this application domain as databuses, which provide communication to exchange information among various electronics devices on the vehicle. This issue is especially important in a view of certification, when regulatory authorities require vendors to make their products compliant with certain sets of criteria to assure safety.

The authors had previously studied the subject of safety assurance with respect to certification, in relation to software tools used in the development of real-time safety-critical applications [2,3]. A set of consistent criteria was developed, based on existing software engineering standards related to tool use, that help certification authorities to make respective decisions on compliance with respective requirements. In case of databuses, the certification process, although essential for modern aircraft and vehicles, has not been started yet, due to the lack of appropriate, agreed upon, criteria.

The objective of this paper is to present an approach to databus assessment with respect to safety, with potential application in certification. First, in Section 2, we review briefly several case studies taken from the literature, two in avionics and two in automotive applications, for which databus design plays an essential role in achieving safety. Then, in Section 3, we discuss various aspects of the process of designing databuses for safety, focusing on certification concerns, the applicable criteria, hazard analysis and failure mode analysis. In Section 4, we overview important characteristics of selected bus designs from the point of view of potential certification. Section 5 presents evaluation data for selected buses, based on an array of experiments, which is followed by a conclusion in Section 6.

2. Avionics and automotive applications

Before any credible safety assessment of a databus design can be made, the theoretical and engineering analyses of databuses have to be applied to practical cases. Below, we briefly present selected case studies taken from the available literature, to illustrate the level of complexity any safety analysis of the databus design has to deal with.

2.1. Avionics case study - Swedish JAS 39 Gripen

Johansson et al. [4] analyzed a flight control system, fly-by-wire, for controlling flying surfaces of a Swedish JAS 39 Gripen combat aircraft, illustrated in Figure 1. The databus runs along the aircraft connecting actuators and sensors at each surface.

The authors provide communication bandwidth analysis for these primary control surfaces and engine, as well as for four secondary control surfaces and a variety of sensors, including:

- advanced air data sensors (static pressure, Mach number, altitude, angle of attack, angle of sideslip)
- angular gyro sensors (pitch, roll and yaw)
- acceleration sensors in z and y axes
- cockpit sensors (pilot commands for pitch, roll and pedal), and all of the above duplicated.

However, safety related criteria are not listed in this work, and the design process is not reported upon.

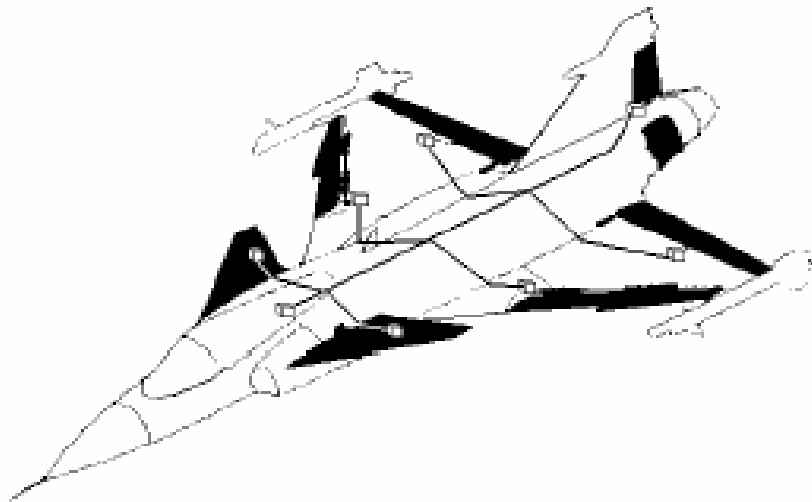


Fig. 1. Distributed flight control system for JAS39 [4].

2.2. Avionics case study – Boeing 777 flight control surfaces

Similar case studies have been presented in the literature for civilian aircraft. Yeh [5] analyzes a flight control system, fly-by-wire, for controlling flying surfaces of Boeing 777 commercial aircraft (Figure 2). Around a dozen of ARINC 629 buses glue together multiple systems consisting of sensors, transducers, actuators, alerts and warnings, electrical and hydraulic power control, information management system, interfaces and other digital electronics.

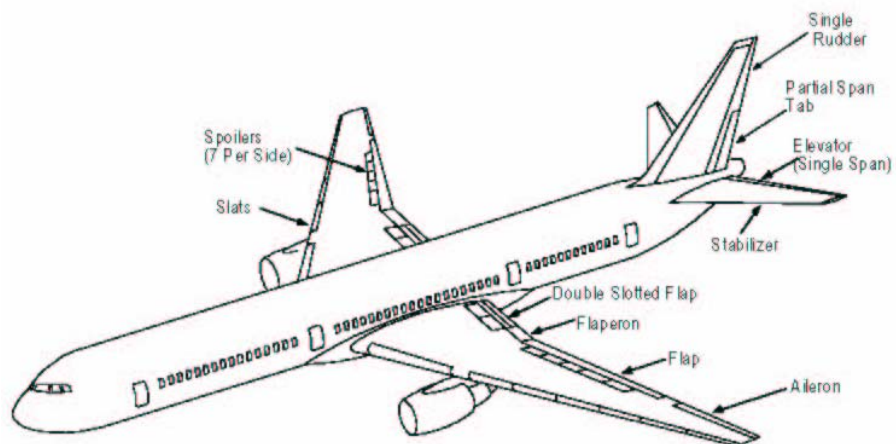


Fig. 2. Boeing 777 flight control surfaces [5].

For this system, a comprehensive safety analysis was performed to assess all potential, significant failures of the fly-by-wire system, including single failures, latent failures, and failure combinations. In the analysis, system separation, partitioning and redundancy were addressed in particular. The process of designing for safety has not been addressed though.

2.3. Automotive case study

Leen and Heffernan [6] present an extended application of three databuses coexisting in a single, road vehicle (Figure 3).

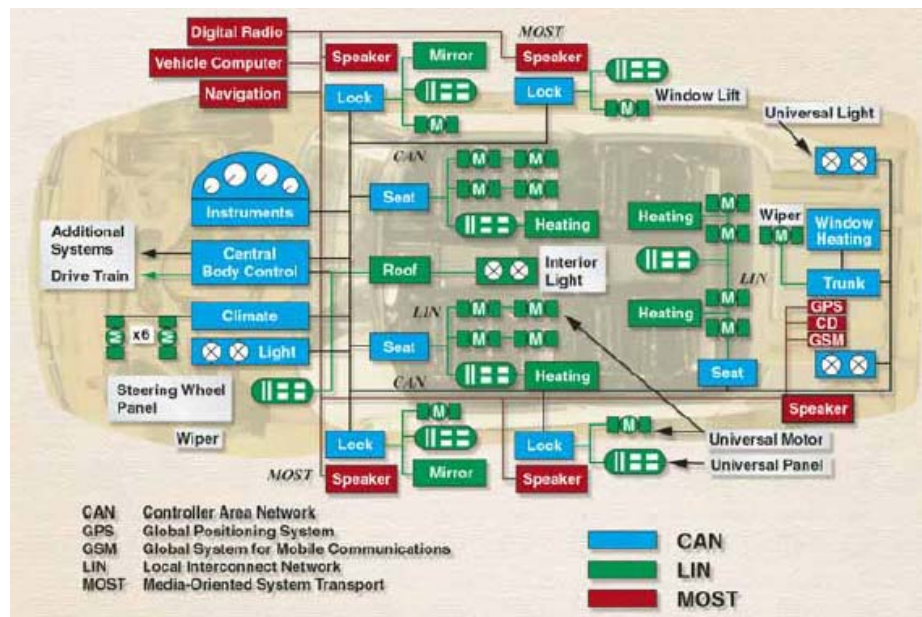


Fig. 3. Example of a modern vehicle network [6].

Various databuses connect subsystems of various levels of complexity and relation to safety. The high speed MOST (Message Oriented Systems Transport) fiber optics bus connects a variety of entertainment and communication devices. LIN (Local Communication Network) bus, based on a master-slave principle, is used for handling low speed on-off devices, such as seats, door locks, sunroofs, windows, rain sensors, door mirrors, etc. CAN (Controller Area Network) bus is used to connect devices performing more sophisticated functions, such as data communication and control between the instrument cluster, body and climate controllers. On top of it, X-by-wire functions can be connected by a fourth type of bus. Despite the comprehensive description of a hierarchy of diverse databuses, no safety analysis has been addressed in this paper.

2.4. Automotive case study – steer-by-wire

Waern [7] studied a system for steer-by-wire application, as an example of X-by-wire system, illustrated in Figure 4. All its individual components are connected electronically via a databus and include an array of sensors (steering wheel sensors, wheel angle sensors,

environment sensors, etc.) and respective actuators (steering actuators, driver feedback actuators, etc.). All X-by-wire systems, where X stands for brake, steer, shift, throttle, etc., are extremely demanding, since their functions are extremely critical for safety. Safety analysis, however, has been only done here in a fragmented way, without addressing the design process.

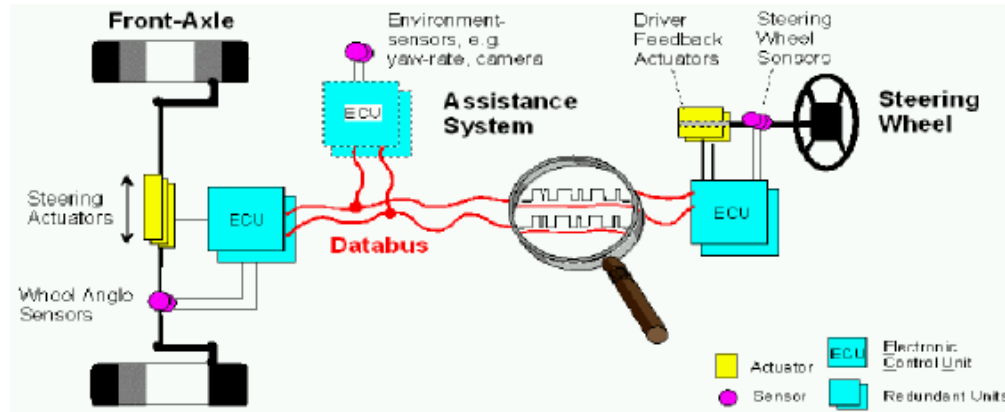


Fig. 4. Steer-by-wire system [7].

3. Fundamental aspects of designing databus for safety

Safety is a property of computer systems that relates to the operation of a computer in a certain physical environment. In principle, a computer or its software does not have to fail to contribute to the violation of safety and cause an accident. Its operation may be perfectly well adhering to specifications, but the chain of unanticipated external events may cause the entire system (of which a computer is a part) to enter some unpredictable state, for which the computer was not designed. In this view, it is not sufficient to focus only on reliability of the final product, such as computer hardware or software (and that of databus, in particular) to assess its suitability for a safety critical-application. The entire development process has to be addressed. The question is, though, what needs to be addressed in this process.

A recent study conducted by the authors on the criteria to evaluate software tools for the development of real-time safety-critical systems, from the viewpoint of prospective certification [2,3], indicated that the selection of the criteria and formulation of respective metrics to assess them has to be a part of the broader effort involving risk assessment with both hazard and failure mode analyses for a specific application. The same conclusion can be applied to the design and use of databuses in safety-critical systems. From the certification perspective, all three aspects of the risk assessment process: multicriteria-based safety assessment, hazard analysis and failure mode analysis, are briefly reviewed below for selected avionics and automotive applications found in the subject literature.

3.1. Certification concerns

Avionics and automotive manufacturers consider using faster and lighter databuses available on the market, in an effort to improve the functionality of systems. However, in these highly regulated industries, the critical point is to assure the equipment safety. For example, the rigorous certification process following the guidelines of RTCA/DO-178B [8], for software, and RTCA/DO254 [9], for hardware, is the basis for acceptance of any new or modified computing systems in the aircraft.

RTCA document DO-178B was developed by the avionics industry to establish software considerations for developers, installers, and users, when aircraft equipment design is implemented using software controlled computing devices. RTCA/DO-254 is an evolving industry specification to apply the same rigor as RTCA/DO-178B does for software to airborne hardware environments. DO-254 was developed by the avionics industry to establish hardware deployment guidelines for developers, installers, and users, when microcomputer hardware, including FPGAs, PLDs and ASICs, are deployed in aircraft equipment designs. Thus, there is a significant emphasis on the design process, when addressing product safety.

DO254 defines three basic categories of lifecycle processes: planning, development, and CCC (correctness, confidence and control). It also defines the required documents to be produced by an applicant. They include: Plan for Aspects of Certification, Development Process, Verification and Validation Process, Process Assurance, Configuration Management, Status Reporting, Requirements Standards, Coding Standards, and a few others.

DO-178B and DO-254 constitute guidelines for the design/development assurance. For validation and testing there is a need to conform to environmental qualification, as per RTCA/DO-160D [10], and need for rigorous and complete testing of variety of failure recovery situations. The system safety considerations, due to lack of well-established metrics, are most difficult to evaluate. Two automotive industry standards, SAE ARP 4754 [11] and SAE ARP 4761 [12], give additional guidelines to the system safety considerations.

Recognizing that it is critical to establish specific measurable criteria to help in assessment of the databuses, the initial selection of the criteria has been proposed in the CAST Position Paper #16 [13]. The major issues to consider when assessing the bus operation include: safety, data integrity, performance, design/development assurance, and validation/testing approaches. Data integrity and performance can be demonstrated by specific array of tests. In the process, the allowed error rate per byte should be defined and means to recover from the errors should be provided. The load analysis and related bus capacity should be also specified. The extreme cases of bus loss, shortening, and opening should be considered in the analyses and tests.

The applicant and certification authority must assure that the evaluation criteria, similar to those listed, are considered. Each specific databus may have details that need to be addressed by a particular method discussed in advance between the applicant and the appropriate certification authority.

3.2. Criteria for assessment of avionics databuses

Based on the CAST paper, Rierson and Lewis [14] provide a set of preliminary criteria to certify avionics databuses on civil aircraft. Their analysis, although not an official position of certification authorities, is aimed at providing aircraft manufacturers with some initial data on the ways to approach the certification process, when developing, selecting, integrating or

approving a databus technology in the context of a civil aircraft project. The suggested criteria are divided into several categories listed in Table 1.

Table 1. Potential criteria for avionics databus certification.

Criterion	Selected Evaluation Factors
Safety	Availability and reliability, partitioning, failure detection, common cause/mode failures
Data integrity	Maximum error rate, error recovery, load analysis, bus capacity, security
Performance	Operating speed, bandwidth, schedulability of messages, bus length and max. load, retry capability, data latency
Electromagnetic compatibility	Switching speed, wiring, pulse rise and fall times, lightning & radiation immunity
Design assurance	Compliance with standards (such as DO-254 & DO-178B)
Configuration management	Change control, compliance with standards, documentation, interface control, etc.
Continued airworthiness	Physical degradation, in-service modifications and repairs, etc.

3.3. Hazard analysis for automotive electronics

Hazard analysis for complex automotive systems involving electronic communication devices (such as databuses) has been done recently by Debouk et al. [15]. They present a list of potential hazards that need to be taken into account at the beginning of safety analysis of X-by-wire systems, consisting of steer-by-wire, brake-by-wire, electronic throttle, and active safety systems. They divide associated risks according to critical, moderate and low consequences. Table 2 includes the hazards with highest associated risk (critical) and their possible controls.

Table 2. Hazard analysis for an x-by-wire system.

Potential Hazard	Possible Mitigation
Loss of power	Dual power system
Loss of communication	Dual communication system
Loss of steering	Backup system, Reduced functionality Steer by braking active safety system
Loss of braking	Backup system Reduced functionality Brake by steering active safety system
Loss of electronic throttle	Backup system Reduced functionality
Loss of actuators	Backup actuators Reduced performance actuator
Loss of sensors	Backup sensors Reduced performance sensor

3.4. Failure mode analysis for a space application

Chau et al. [16,17] describe and discuss typical failure modes for a highly reliable bus architecture for space applications. Their study is related to the use of commercial-off-the-shelf products, such as those compliant with IEEE Std 1394 and SpaceWire, to be used in high availability avionics systems. They identified those failure modes that are fairly frequent or critical to the survival of the spacecraft. Summary of the discussion is presented in Table 3.

Table 3. Failure modes for a space application.

Failure Mode	Description
Invalid messages	Messages sent across the bus contain invalid data
Non-responsive	An anticipated response to a message does not occur or return in time
Babbling	Communication among nodes is blocked or interrupted by uncontrolled data stream
Conflict of node address	More than one node has the same identification

4. Selected databus designs

Databus applications and case studies, such as those described in Section 2, as well as others presented in the literature (steer-by-wire [18], safe-by-wire system [19], car entertainment platform [20], give a broader context for developing databus evaluation criteria with respect to safety. Essential characteristics of databus description from the safety standpoint do not differ much from conventional bus specifications, which must include mechanical, electrical and logical elements of the bus design [21]:

- *mechanical* properties concern bus wiring, connectors, pinout, module design and dimensions,
- *electrical (or optical)* properties are related to signal levels and their dynamics to carry information, including electromagnetic characteristics, and
- *logical* properties concern the protocol of exchanging information over a bus.

Specifics of the bus protocol must include separate descriptions of three phases of bus operation:

- *bus arbitration* (competing for bus access)
- *data transfer*, how devices exchange data once they obtain bus access, and
- *fault handling* (dealing with bus errors).

Bus protocols are typically described in terms of a layered approach, defining various aspects of bus operation according to the respective layers of the ISO/OSI Reference Model, especially Physical, Data Link and Application layers.

Since historically, two types of buses, ARINC 429 and 629, have been used first in most commercial aircraft, we mention them first in the description below. They may not be

adequate for the future avionics applications due to the limited speed and bandwidth. Therefore, a number of new databuses currently are being considered. For general aviation aircraft (business jets and smaller aircraft) and automotive applications, a number of different communication technologies are being developed, including: CAN, FlexRay, TTP/C, SAFEBUS, and others, which are considered next.

4.1. Traditional avionics databuses

This is the oldest category of databuses, well documented and researched, with a multitude of applications worldwide, on both military and civilian aircraft [22].

ARINC 429 is a general purpose avionics databus, the most used databus in commercial aviation, with the following characteristics:

- data rate: 1 Mb/s or 12-14.5 kb/s with 1% tolerance
- type: serial, unidirectional (two buses needed for bi-directional operation), point-to-point
- medium: two signal wires, wired transformer coupling
- bit encoding: Return-To-Zero bipolar, 10V, trilevel
- architecture: serial point-to-point one way protocol with only one transmitter on a wire pair and one to twenty receivers
- protocol: Williamsburg "bit oriented" and Numeric Data, Discrete Data, File Data
- frame length: 32 bit sequential words separated by at least four bit times of zero voltage (NULL) eliminating the need for a separate clock signal wire (self-clocking)
- frame format: typically includes five primary fields – one Parity bit, SSM, SDI, and 8-bit System Address Label, leaving 18-20 bits for payload data
- max. length or electrical load: 20 receivers
- fault tolerance features: only parity bit.

MIL-STD-1553B/1773 is an Aircraft Internal Time-Division Command/Response Multiplex Data Bus, in use since 1973, widely applied in military avionics, with the following parameters:

- data rate: 1 Mb/s (20 Mb/s for MIL-STD 1773, and optical version)
- type: serial, bi-directional, self synchronized
- medium: twisted-shielded pairs of wires (a second path for bus traffic should one of the buses be damaged); fiber for MIL-STD-1773
- bit encoding: biphase Manchester II; logic one is transmitted as a positive pulse followed by a negative pulse; logic zero is a negative pulse followed by a positive pulse; serial digital pulse code modulation
- architecture: single master (bus controller), the only device that can initiate communication; three functional modes of terminals allowed on the data bus: the bus controller, the bus monitor, and the remote terminal (only one controller may be active at a time)
- protocol: serial digital multiplex data bus system shall function asynchronously in a command/response mode, and transmission occurs in a half-duplex manner
- frame length: 20 bits (16 bits command data and status, 3-bit sync, 1 bit parity)

- frame format: twenty one-microsecond bit times allocated for each word (3-bit time sync pattern, a 16-bit data field specified differently for each word type, and 1 parity check bit); three types of words: command, status, and data
- max. length or nodes: 32 (master + 31)
- fault tolerance features: (a) accuracy and long-term stability of +/- 0.1% (short-term stability is less than 0.01%); (b) nominal characteristic impedance of the cable (Z_0) within the range of 70.0 to 85.0 ohms at a sinusoidal frequency of 1.0 MHz; may be transformer coupled.

ARINC 629 is an upgrade and modification of 429, used only on Boeing 777, with the following characteristics:

- data rate: 2 Mb/s
- type: serial, bidirectional, distributed control, without the need for a bus controller (avoiding single-point failure mode)
- medium: two signal wires (twisted pair)
- bit encoding: Manchester II
- architecture: multi-master, autonomous terminal
- protocol: digital autonomous terminal access control (collision avoidance); Basic protocol and Combined protocol (for periodic and non-periodic traffic, with priority)
- frame length: 16 bits of data, 1 parity bit and 3 sync bits
- frame format: inherited from MIL-STD-1553
- max. load: up to 120 terminals
- fault tolerant features: (a) each terminal monitors its own transmissions; (b) non-intrusive, inductive coupling.

ARINC 659/SAFEbus is a Backplane Data Bus for Integrated Modular Avionics, developed by Honeywell, and installed only on Boeing 777:

- data rate: 60 Mb/s (with clock rate 30 MHz)
- type: serial unidirectional tightly coupled synchronous backplane bus
- medium: 2 data lines and one clock per bus (12 lines total on a backplane)
- bit encoding: uses backplane transceiver logic
- architecture: quad-redundant
- protocols: physical and data link layers; Table Driven Proportional Access for medium access
- frame length: 32 bits
- frame format: compatible with ARINC 629; Frame Description Language
- max length: up to 42 inches
- fault-tolerant features: fault detection, fault containment and redundancy.

4.2. New generation avionics and automotive databuses

Increasing demand for fast and reliable communication causes rapid proliferation of relevant databus standards. Some recent, most important designs are described below.

IEEE 1394/FireWire [23] was designed for high-speed entertainment applications, but due to its real-time features, such as guaranteed message delivery time, it is often considered in safety-related applications. Its major parameters are as follows:

- data rate: up to 400 Mb/s
- type: serial, asynchronous and isochronous
- medium: two shielded twisted pairs, two power conductors (entire cable shielded)
- bit encoding: IEEE Std 1569 LVDS (Low Level Differential Signaling)
- architecture: multi-master, daisy-chain or tree
- protocol: physical, link and transaction layer
- frame length: 16B for isochronous transmission
- frame (packet) format: varies depending on transmission mode
- max. nodes: 63 on a segment
- fault tolerant features: (a) data and header packet CRC; (b) timeout conditions; (c) error code in the acknowledgement and response packets; (d) enabling/disabling individual ports for reconfiguration; (e) isochronous transmission guaranteeing message delivery
- bus highlights: asynchronous transmission for reliable message delivery; multicast isochronous transmission; tree topology.

CAN and TT-CAN [24] is a world standard in automotive electronic control, with wide component manufacturing and support bases; typical application involves 2-10 control units with soft real-time requirements. The major characteristics of CAN are as follows:

- data rate: 10 kb/s to 1 Mb/s
- type: serial, bi-directional, multi-master
- medium: differential twisted pair, single wire, optical fiber
- bit encoding: NRZ with bit stuffing, 5V/50mA transceivers
- architecture: multi-master bus, bitwise priority arbitration, event-triggered with no clock synchronization, multicast transmission with message filtering
- protocol: Physical Layer + Data Link Layer (Logical Link Control + Media Access Control = Object + Transfer Layers)
- frame length: varies by frame type, for data 107-bit (max. 64 data bits, 11 bits address id, 15 bit CRC, 6 bit control field)
- frame types: data, remote, overload, error
- max length: 40 m for 1 Mb/s, 100 m for 500 kb/s, 200 m for 250 kb/s, 500 m for 125 kb/s, 6 km for 10 kb/s; max 2048 nodes theoretically for CAN 2.0A (average 2-10 nodes per network)
- fault tolerant features: (a) frame recognition determined by polarity of the RTR bit, data and remote frames separated by interframe spacing; (b) error detection: stuff, bit timing, data, 15 bit CRC, format and message acknowledgement error detection
- bus highlights: immediate message retransmission, low message latency for small traffic loads, wide support network of manufactures and suppliers, highly tested.

TT-CAN adds session layer on top of CAN, uses TDMA as medium access protocol, disables retransmission and provides global clock synchronization via master reference message, with 1-8B of data per frame. Bus highlights include: support for deterministic messages and fault handling, low jitter transmission, 25-35% typical data efficiency, error detection and redundancy management; improved bus utilization for higher traffic loads as compared to CAN.

FlexRay [25] has been designed for future generation high-speed control applications in vehicles, as a replacement of CAN, TTCAN, and as a contender for newer designs. It has the following parameters:

- data rate: 10 Mb/s
- type: serial, bi-directional, half-duplex transmit
- medium: differential pair, 2-channel redundancy
- bit encoding: NRZ8N1 & Frame Start Sequence
- architecture: multi-master, fault tolerant, single/dual channel, bus/star or mixed topology; collision-free arbitration via unique ID and slot counting (during startup collisions may happen)
- protocol: time-triggered and event-triggered supported by static (TDMA) and dynamic slots
- frame length: 264 bytes
- frame format: 5B header, 254B data, 3B CRC
- max. length 24 m; max. nodes 22 (or 64)
- fault tolerant features: (a) frame recognition with 11b frame ID; (b) error detection/data validation with 11b header CRC and 24b frame CRC; (c) clock sync: offset and rate correction (fault tolerance midpoint)
- bus highlights: two independent physical layer channels; recurring communication cycle in guaranteed time; static segment for polling and dynamic for temporal events.

TTP/C [26] has been designed for avionic flight control (Airbus) and automotive control systems (drive-by wire, steer-by wire, chassis and body control), and has the following parameters:

- data rate: 25 Mb/s over Ethernet, 5 Mb/s over optical fiber and twisted pair RS-485, 2 Mb/s over twisted pair ISO 11892-2
- bus type: serial, bi-directional, with bus-guardians
- medium: twisted pair, optical, Ethernet
- bit encoding: Modified Frequency Modulation
- architecture: time triggered (TTA) for fault tolerance, replica determinism, fail-silence, composability; topologies include active star, passive bus, combinations with bus guardians; three communication modes - startup, download, normal
- protocol: Physical Layer + Data Link Layer + Protocol Service Layer + FT-COM Layer; strict TDMA slotting by means of rounding, with ET traffic possible if scheduled statically
- frame length: 4-8 bit header, 240B total data length, 24-bit CRC
- frame types: I-frame (initial synchronization), N-frame (application data)
- bus length: depends on medium; max. nodes 64; average 4-32 nodes with high safety requirements
- fault tolerant features: (a) redundancy with two separate channels (duplicated nodes and buses); (b) error detection: 24 bit CRC, with clique detection for all asymmetric communication faults, different dividing polynomial seeds for dual channel operation; (c) distributed clock synchronization with offset correction in microseconds range; (d) frame recognition via Frame Type Identifier.
- bus highlights: fault tolerance, replica determinism, fail silence, error containment by bus guardian, guaranteed message latency and jitter, dual redundant TT messages,

consistent membership and clique detection, high data efficiency 60-80%, effective long propagation delay handling; static scheduling implying safe bandwidth utilization through the MEDL data structure.

Safe-by-Wire [19] has been developed very recently for application in an automotive passenger restraint system for deployable devices and for sensors, and has the following parameters:

- data rate: variable between 20..200 kb/s
- type: serial, bi-directional with integrated power distribution
- medium: unshielded differential pair
- bit encoding/signaling: three data levels (0, 3, and 6 V) and power level (11/30 V)
- architecture: master-slave (multi-master optional); bus, tree, ring, or mixed topology; fast polling with interrupt possibilities, asymmetrical or symmetrical daisy-chain configuration (shut-down individual bus sections)
- protocol: physical, data link & application layer
- frame length: 30 bits including command, addresses, data, CRC, and one error bit
- frame format: 4b command, 6b address
- max. length 25-40 m; max. nodes: 64 (3 reserve)
- fault tolerance features: (a) shorts of bus wire, open circuits, and shorts between the bus wires; (b) immunity to “babbling idiots”; (c) multilevel protection against inadvertent deployment; (d) CRC and bit error data validation
- bus highlights: sensor can interrupt current message for exclusive communication of time-critical data; switches in the slaves can split bus into sections; recovery from short made by software and hardware; special D-frame to collect one-bit data from several slaves (30 bits only); latency time for interrupts from smart sensors: 2 bit times at low speed + 1 bit time at high speed ($\cong 105 \mu\text{s}$).

Other databuses. There is a multitude of other databuses used in avionics and automotive applications, of which the following are worth mentioning:

- ROBUS (Reliable Optical BUS) is a fault tolerant bus being developed at NASA as a part of a SPIDER project for high-reliability space missions [27]
- SpaceWire is a serial, point-to-point, full-duplex bus based on IEEE Std 1355, modified for space applications, with minimum data rate of 2 Mb/s (no max.), and predetermined jitter [17]
- Byteflight has been designed by BMW to support both event and time driven messaging in passive safety systems; it has been incorporated into FlexRay’s dynamic protocol segment [28]
- Bluetooth and its derivative, ZigBee, are wireless short-range (< 100 m) networks that may play an important role in this application domain in the future, because of their ability of quick reconfiguration [29].

In the following Table 4, selected databus characteristics important to safety are summarized, addressing the low-level aspects of respective databus designs. The focus is on selected hardware issues, such as medium, data rate, and encoding schemes, leaving out specific upper-layer protocol characteristics.

Table 4. Selected databus characteristics.

Databus	Type	Archit.	Medium	Rate	Encoding
Arinc 429	serial unidirectional	single master	2 wires	100kb/s	RTZ bipolar
MIL1553	serial bi-directional	single master	twisted pairs	1 Mb/s	2phase Manchester
Arinc 629	serial bi-directional	multi-master	twisted pairs	2 Mb/s	Manchester II
Arinc 659	serial bi-directional	quad redun.	twisted pairs	30MHz	2phase Manchester
FlexRay	serial bi-directional	fault toler.	optical or wire	10Mb/s	undefined
CAN	serial bi-directional	multi-master	twisted pairs	1 Mb/s	NRZ + bit stuffing
TTP/C	serial bi-directional	double redun.	twisted pairs	25Mb/s	MFM
IEEE1394	serial	daisy or tree	twisted pairs	400Mb/s	LVDS
Safe-Wire	serial bi-directional	master-slave	twisted pairs	200 kb/s	3-level
SpaceWire	serial bi-directional	master-slave	2 wires	Min. 2Mb/s	undefined

5. Experimental approach to databus safety evaluation

With such a wide spectrum of avionics and automotive databus designs of very different characteristics, as presented in Section 4, it is not very clear what specific criteria would be optimal for data bus evaluation with respect to safety. As pointed out in Sections 2 and 3, there are only a few engineering studies on the subject of databus safety, but they do not go deeply enough into analyzing the product safety. Moreover, to the authors' knowledge, there are neither empirical data nor simulation studies on the subject, even in the most recent literature [30]. Some safety oriented databus comparison studies exist, but they are based only on verbal protocol analysis, without experimental backing that would validate the findings [31].

Even assuming that the process issues in databus design can be adequately addressed, as described in Section 3, there is still an open question what specific criteria need to be met by the databus, as a product, for certification purposes. To shed some light on databus safety evaluation, we conducted for the last couple of years a number of experimental studies on various databus designs and configurations.

In principle, the traditional approach to evaluate a databus design is to measure its two most important performance parameters: data throughput and data latency. *Data throughput* can be defined as the number of information bits per second transmitted by a network. Data throughput is dependent on a variety of factors, may be very sensitive to network load, and cannot exceed a theoretical limit known as channel capacity (that is, the theoretical maximum number of bits transmitted in unit time, also known as data rate). *Data latency*, on the other

hand, also known as message delay, packet delay, etc., is usually defined as the time interval elapsing between making a request by a network node, and the completion of a response to this request. As such, it depends on such factors as medium characteristics, overall load, transmission time, propagation time, protocol overhead, queuing delay, service time, and others. Normally, one would change the bus load and measure both parameters to derive some conclusions on bus performance. However, being very common measures of databus performance, they do not have much applicability to safety evaluation. This is because in safety related applications, the main issue is to predict when the databus may become the source of errors and when its failure may initiate or contribute to a chain of events that would cause a hazard or danger, leading to an unsafe behavior of an embedded system. The above mentioned parameters address only databus performance, not the consequences of failures, which are directly related to safety.

Nevertheless, since databus by itself is neither safe or unsafe, unless put in the context of an application, databus performance parameters can be used, to a limited extent, as indicators of databus safety, if reflecting the broader context of databus use. In this view, to study safety a databus should be placed in a broader context, for example:

- system oriented, such as another network or different modes of operation, or
- software oriented, such as a driver software or higher level protocol.

Considering these factors, to address the issue of databus safety, we conducted a series of experiments using both plain simulation for well described models of databus networked configurations, as well as empirical tests relying on actual data transfers along a modern bus. The ultimate objective was to acquire more comprehensive information on databus behavior than just from straightforward performance evaluations.

5.1. Previous research

VMEbus and RACEway experiments [32]. Historically, VMEbus was one of the first databuses used in the military and industrial applications, including transportation, so it is well understood and has multiple installations worldwide. Although VMEbus is not currently considered suitable for avionics or automotive applications, we intended to use it as a vehicle for creating a benchmark application for subsequent databus protocol comparisons.

When safety is of major concern in databus design, one wants to have a good hold on databus performance under heavy load conditions. With this in mind, several experiments were designed to understand the behavior of a VMEbus based server under high utilization, and to compare its performance with a circuit-switched interconnect (RACEway), with both immersed in a bigger network of multiple nodes.

Sample results from this study are shown in Figure 5. A VMEbus server hooked to an FDDI network shows a significant degradation of performance (measured as access delay) with higher load (channel utilization). Its performance pattern is typical to any other configuration of interconnects, as shown by the shape of respective curves, and demonstrates significant improvement of behavior if a newer databus architecture is used (RACEway). In a view of databus safety research, the experiments verify the suitability of conducting access delay vs. bus load simulations for determining databus response of specific bus designs. The implications for safety may be significant, if the results show meaningful increase in access delay for utilizations far from bus saturation (100%), as in the three cases from Figure 5.

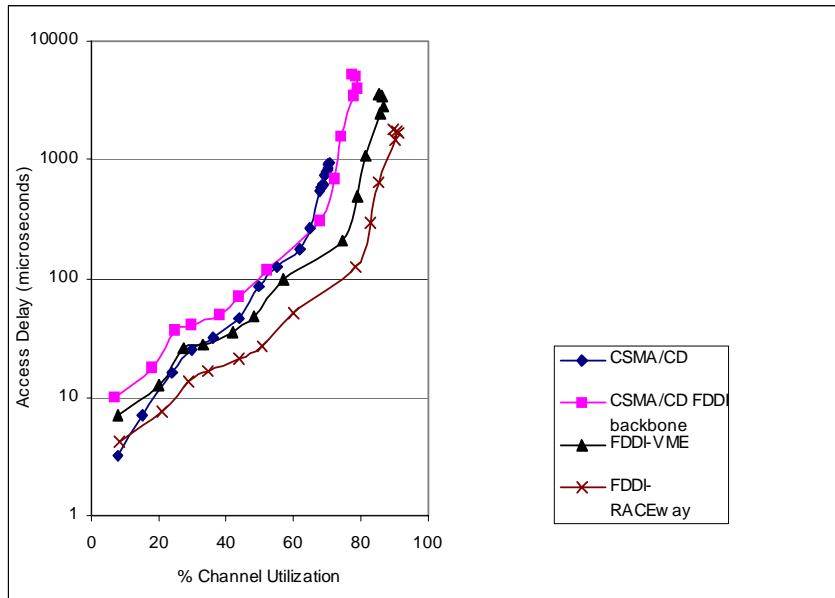


Fig. 5. Server access delay for 64-byte packets.

IEEE Std 1394/FireWire experiments [33]. Another set of experiments was aimed at conducting actual measurements of data transfers over the bus to see the effect of internal databus properties, such as protocol variations, packet/block size, dependence on software driver properties, etc. Of modern databuses available to the authors, IEEE Std. 1394, known as FireWire bus, was selected, because of its diversity of configurations that can be experimentally set up and rearranged at the convenience of experimenters.

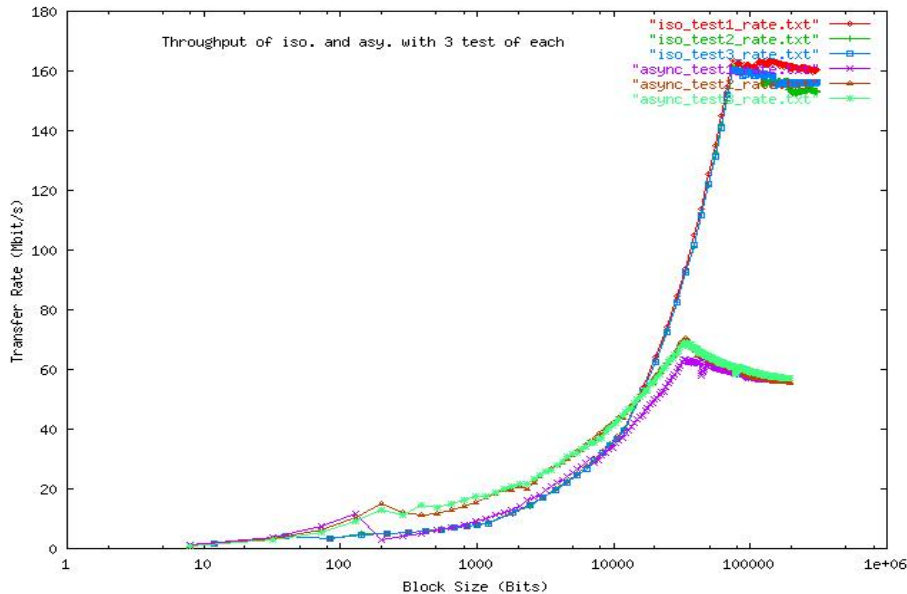


Fig. 6. IEEE 1394 throughput over raw driver.

As a part of a bigger study, we focused in particular on investigating the limits of data transfer speed for two distinctive transmission modes of IEEE Std 1394 databuses:

asynchronous and isochronous transmission. An isochronous transmission mode provides reserved bandwidth for real-time data, every 625 microseconds, which is highly relevant for safety-critical applications, because of the message delivery guarantees.

The results of experiments, conducted for PCILynx boards under Linux and a variety of different device drivers, confirmed the superiority of isochronous transmissions over the traditional asynchronous mode, as shown in Figure 6. But at the same time, the experiments revealed significant degradation in the maximum data transfer rate achievable in practice, versus the theoretical one (163 Mb/s as opposed to theoretical 400 Mb/s for IEEE Std 1394 databuses in isochronous transmission mode).

In a view of databus safety, the experiments proved the necessity of conducting practical measurements of data transfer rates, and other databus parameters, to verify vendor's claims and compliance with standard's specifications. When facing a decision on selecting a specific databus for safety-critical applications, performance data are of primary significance with respect to making the right choice. They are also crucial in finding how widely various databus designs differ in meeting application requirements.

Bluetooth experiments [34]. One of the objectives of these experiments was to get a grasp on the suitability of wireless networks for automotive applications, since the ability of having a mobile connection allows not only obtaining real-time information on traffic conditions, but also joining the traffic by a vehicle, which then becomes a part of the entire community sharing information on the road conditions, potential congestions, and other factors related to safety. One of the main problems with wireless networks is reconfiguration of the network when nodes join or leave the network. In case of Bluetooth, this boils down to the development of a suitable routing protocol that could be successfully applied to Bluetooth scatternet networks. The effective application of routing to Bluetooth depends on careful consideration of two additional protocols: topology formation and link scheduling.

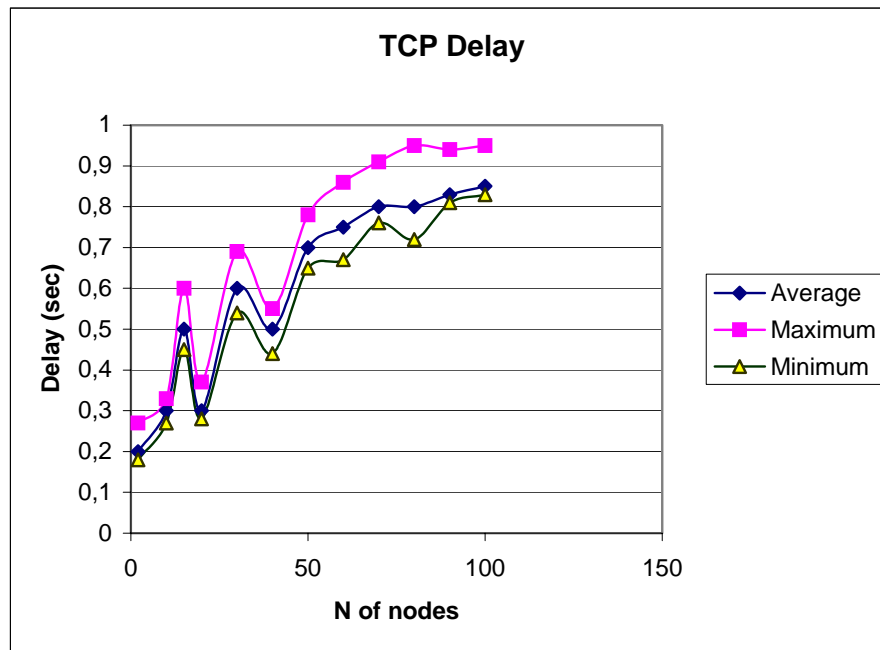


Fig. 7. TCP delay versus the number of nodes in scatternet.

In general, the topology formation protocol has been shown to provide relatively low connection latencies at the expense of higher nodal density. The free node connection delay decreased by 75%, from the maximum value of 6 seconds, as the number of nodes connected to the scatternet increased to over 16. The scatternet formation delay increases linearly for scatternet sizes up to 16 nodes. With more than 16 nodes, the graph is more logarithmic making the algorithm suitable for larger scatternets.

The healing delay in our simulations was measured under 20 seconds for larger networks. Such a delay could pose a significant threat to service up-time in highly dynamic environments. Considering however the fact that clock re-synchronization to one lost slave node may take about 6 seconds, the algorithm was still performing relatively well under increased network occupancy because the healing delay decreases with increased network partitions (increased node density).

Figure 7 shows that as the number of nodes in the scatternet increases, the average delay a packet experiences increases just a little due to scheduling conflicts and routing table lookups. One may speculate that this delay shall increase at a higher rate, if network loses some of its members and becomes unstable. But a more important observation is an unexpected bump in delays for certain numbers of nodes between 10 and 40, which can be attributed to the limits on the TCP buffer size. The implications for safety are that higher level protocols may have detrimental effect on the stability of bus performance and, thus, on safety.

5.2. CAN/TTCAN experiments: databus scheduling in presence of faults

Previous experiments show that, although databus performance measures such as data throughput and latency alone can hardly provide useful information on databus safety, if combined with other factors they can form more meaningful quality indicators of an embedded system in a safety related application. To evaluate safety more thoroughly, different criteria are needed that relate more directly to potential databus failures. One such parameter studied by the authors is databus scheduling in the presence of faults. To illustrate this capability, we built and implemented a model of fault injection into CAN and TTCAN networks [35]. Below, we present the effects specific faults, called Remaining Time faults, have on the node message scheduling for both databuses.

Methodology. Figure 8 shows the network topology used in the simulation model. The network in all simulations contained 8 nodes communicating on a CAN or TTCAN logical databus. This setup is a good representation of a physical network, and may model a distributed control network in a vehicle active suspension system, for example.

In this network topology, data transmission is realized through network interface functions of the simulator. When a task is ready to send data, it calls the respective network function. When new data arrives at its destination, an interrupt is triggered in the receiving node and the real-time kernel invokes an interrupt handler. The handling task then determines how the received data are processed. In this particular simulation model, the incoming data are forwarded to the next node in the network. For example, if node i receives data from node $i-1$, node i after the triggering of its interrupt handler function, sends the same data to node $i+1$ in the network. This type of data flow proceeds in a circular fashion but still respects the databus scheduling policy of its respective protocol.

This means that a message generated by a periodic task in node 1 eventually comes back to this node after all nodes in the network have processed it. This type of traffic flow can model sensor traffic for a shock absorber in an automotive suspension control system. All

controller nodes that control the mechanical actuators process this sensor reading and compute control variables with their respective control laws. To more realistically model network traffic, we added three interference tasks into three network nodes. These tasks generate dummy data, once every 10 milliseconds with varying time phase shifts.

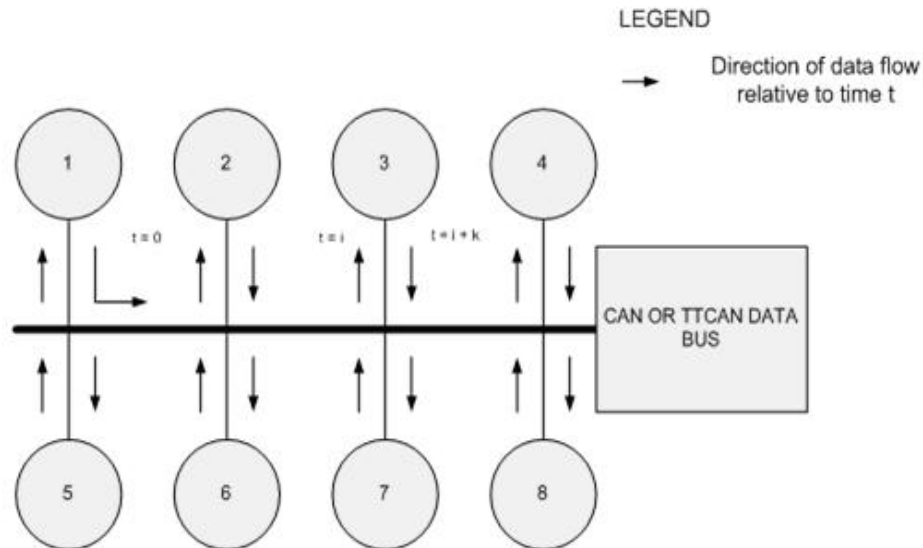


Fig. 8. Simulated network topology.

All simulations were performed with the TrueTime simulator [36], extended for fault injection by the authors at Warsaw University of Technology, executed in Matlab 7 and Simulink 6 environment on a single Intel Pentium 4, 2.8 GHz microprocessor under Microsoft Windows XP operating system. Fault injection was done by fault code insertion and recompilation of the TrueTime source code with Microsoft Visual C++ 6.0 compiler.

Experimental results. Our fault injection model disturbs the simulation network model explained above, by introducing disturbances to the databus operation. The effect of the fault injection can be observed on the network throughput, message delay and scheduling. Those three parameters can be compared against the total network traffic or total number of transmissions completed by all nodes at any given simulation time t , to assess databus performance. Since for safety evaluations, message scheduling is a more important indicator of databus performance than the other two, the simulation results presented below show the effects of the remaining transmission time fault on this particular parameter. This type of parameter determines how much time is needed for the entire transmission of a message.

Figures 9 and 10 present the results of message scheduling under fault presence conditions for CAN and TTCAN protocols. The experiments show the effect a *Remaining Time* fault has on schedules of both databuses. For CAN, we notice much lower bus utilization within a simulation window of 50 ms. This utilization is less than 10%, while TTCAN achieves utilization over 40%. Therefore this experiment shows that although in fault-free conditions CAN achieves better bus utilization (under simulated traffic conditions), it performs poorly under a presence of faults. TTCAN, due to its scheduling mechanism, performs better with respect to efficient bus access in the presence of this type of fault, but its bus utilization also decreases considerably. This is an important result because it indicates

that utilization can decrease drastically in the presence of a fault, while in fault free conditions, theoretically, the utilization can be higher.

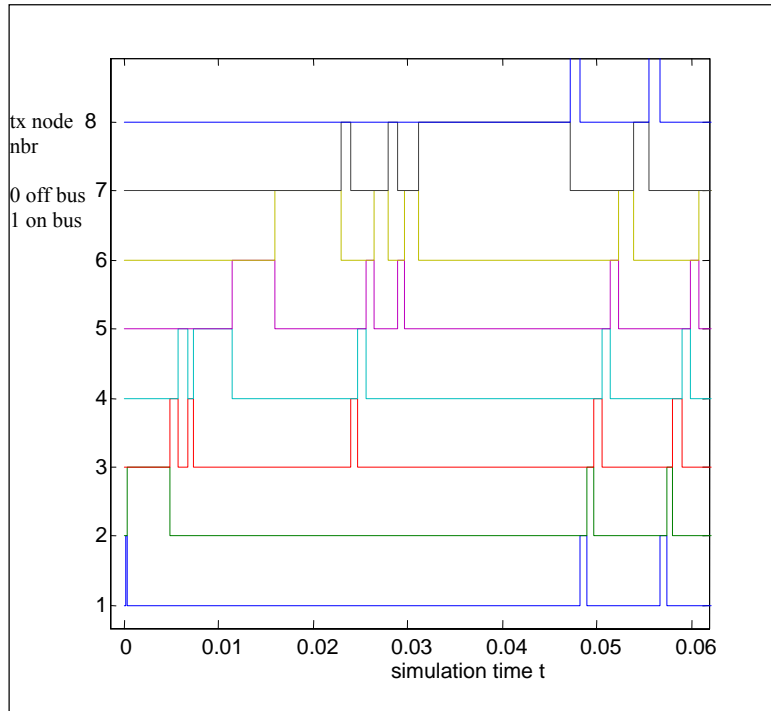


Fig. 9. Schedule of CAN databus under the Remaining Time fault.

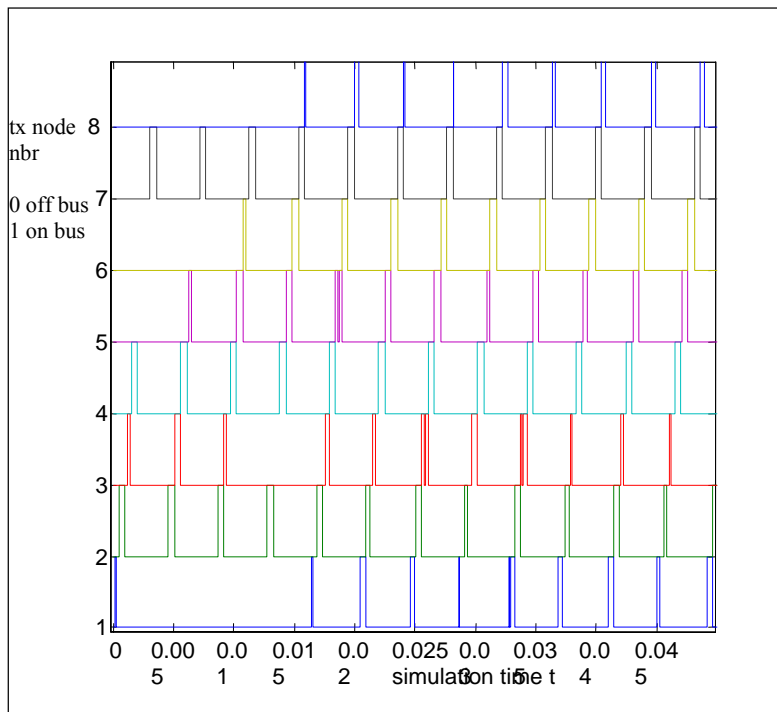


Fig. 10. Schedule of TTCAN databus under the Remaining Time fault.

6. Conclusion

Because of the risks involved in using computer equipment and software in safety-critical applications, specific industries, such as civil aviation or automotive industries, are highly regulated. As a result, databuses with their hardware and software components need to be certified for use in these critical applications. Therefore an urgent need exists to develop a consistent set of criteria for databus evaluation that can be used by certification authorities and applicants to assess respective bus designs.

In this paper, we described an approach that can be used as a starting point to develop a comprehensive set of criteria and corresponding measurement procedures for databus evaluation. It has to be a part of the overall risk assessment process for a safety-critical application, and include a databus component as an important part of computer hardware and software design, both from the process and product perspective. In the process aspect, corresponding hazard analysis and failure mode analysis should be accompanied by the evaluation of safety criteria for a specific application and the databus under consideration.

Specific results of this study include the determination to what extent simulations and empirical data collection can be useful in databus evaluation for safety-related purposes. Simulation experiments, conducted for a variety of bus configurations with network interconnects, confirmed the usefulness of typical performance parameters to determine databus response of specific bus designs. On the other hand, practical experiments, conducted for various configurations of databuses, also proved the validity of measuring data throughput and other databus parameters, to verify vendors' claims and compliance with standards' specifications. When facing a decision on selecting a specific databus for safety-critical applications, performance information on data throughput and latency, whether obtained via simulation or via practical measurements, is of high importance with respect to making the right choices, if placed in a broader context of databus use.

More relevant information on bus behavior, however, can be obtained from simulating faulty conditions and observing their consequences on databus performance. From the simulations conducted for CAN and TTCAN models, we conclude that neither one is completely immune to faults. We found that under certain fault types CAN protocol ensures better performance, while for another set of faults TTCAN protocol behaves better. In general, more information is transferred by CAN than TTCAN due to the event driven nature of the latter. TTCAN transfers less information because of its strict scheduling policy. According to this policy if a message misses its slot it must wait at minimum one cycle before it can be transmitted over the network. Effectively this causes decreased global network throughput in TTCAN, but ensures predictability, which is more important for safety.

The fault injection experiments also showed that certain faults could lead to completely unpredictable results. Such was the case when we injected a fault into the *Remaining Time* parameter. This injection resulted in an increased and randomized delay values for TTCAN, which contradicts to the main benefit of TTCAN under fault-free operation – predictability of delays. Under presence of this fault, the observed delay randomization, to some extent, falsifies the notion of uniformity and predictability of delay in time-triggered systems. This effect shall be studied in further research. Future research should also involve effects of injections of other fault types on performance, as well as modeling other databuses and safety-critical network interfaces and comparing them with the two discussed in this paper.

Acknowledgement

Previous version of this paper was presented at the ESREL'05, European Safety and Reliability Conference, in Gdańsk, Poland, 2005. The authors would like to thank Dr. Marek Śnieżek for contributing to the discussion of bus characteristics in Section 4, and Professor Krzysztof Kołowrocki for the encouragement to extend the conference version of this paper.

References

- [1] Kiencke U. et al. The impact of automatic control on recent developments in transportation and vehicle systems. *Annual Reviews in Control*, vol. 30, issue 1, 2006 (in print).
- [2] Kornecki A., J. Zalewski. Experimental evaluation of software development tools for safety-critical real-time systems. *Innovations in Systems and Software Engineering – A NASA Journal*, vol. 1, no. 2, September 2005, pp. 176-188.
- [3] Kornecki A., J. Zalewski. Qualification of software development tools for airborne systems certification. *Proc. SoftCeMent'05 Workshop on Software Certificate Management*, Long Beach, Calif., November 8, 2005, pp. 13-18.
- [4] Johansson R. et al. On communication requirements for control-by-wire applications. *Proc. 21st International System Safety Conference*. System Safety Society, Unionville, Va., 2003, pp. 1123-1132.
- [5] Yeh Y.C. Design considerations in Boeing 777 fly-by-wire computers. *Proc. Third IEEE International High-Assurance Systems Engineering Symp.* Washington, DC, November 13-14, 1998, pp. 64-72.
- [6] Leen G., D. Heffernan. Expanding automotive electronic systems. *IEEE Computer*, vol. 35, no. 1, 2002, pp. 88-93.
- [7] Waern M. *Real-Time Communication: Evaluation of Protocols for Automotive Systems*. Master Thesis, Royal Institute of Technology, Stockholm, Sweden, 2003.
- [8] RTCA/DO-178B. *Software Considerations in Airborne Systems and Equipment Certification*. RTCA Inc., Washington, DC, 1992.
- [9] RTCA/DO-254. *Design Assurance Guidance for Airborne Electronic Hardware*. RTCA Inc., Washington, DC, 2000.
- [10] RTCA/DO-160D. *Environmental Conditions and Test Procedures for Airborne Equipment*. RTCA Inc., Washington, DC, 1997.
- [11] SAE ARP 4754. *Certification Considerations for Highly-Integrated or Complex Aircraft Systems*. Society of Automotive Engineers, Warrendale, Penn., 1996.
- [12] SAE ARP 4761. *Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment*. Society of Automotive Engineers, Warrendale, Penn., 1996.
- [13] CAST Position Paper CAST-16. *Databus Evaluation Criteria*, 2003. URL: <http://www.faa.gov/certification/aircraft/av-info/software/CAST/cast-16.rtf>

- [14] Rierson L., J. Lewis. Criteria for certifying databuses on civil aircraft. Proc. DASC'03, 22nd Digital Avionics Systems Conference, Indianapolis, Ind., October 12-16, 2003, vol. 1, pp. 1.A.2-1/9.
- [15] Debouk R., T. Fuhrman, J. Wysocki. Architecture of by-wire systems: design elements and comparative methodology. In: In-Vehicle Networks, Safety Critical Systems, Accelerated Testing, Reliability, SP-1783. Society for Automotive Engineers, Warrendale, Penn., 2003, pp. 171-182.
- [16] Chau S.N. et al. Design of a fault-tolerant COTS-based bus architecture. IEEE Trans. Reliability, vol. 48, no. 4, 1999, 351-359.
- [17] Chau S.N. et al. A design-diversity based fault-tolerant COTS avionics bus network. Proc. 8th Pacific Rim International Symp. On Distributed Computing. IEEE Computer Society Press, Los Alamitos, Calif., 2003, pp. 35-42.
- [18] Wilvert C. et al. Evaluating quality of service and behavioral reliability of steer-by-wire systems. Proc. ETFA 2003 IEEE Conf. on Emerging Technologies and Factory Automation. Lisbon, Portugal, 2003, pp. 193-200.
- [19] Boys R. Safe-by-wire: the leading edge in vehicle airbag control. In: In-Vehicle Networks and Software, Electrical Wiring Harnesses, and Electronics and Systems Reliability, SP-1852, Society for Automotive Engineers, Warrendale, Penn., 2004. Paper No. 2004-01-0205.
- [20] Lessard M. IDB-1304 automotive reference platform – enabling in-vehicle entertainment. Mindready Solutions, Research Triangle Park, NC, 2003.
- [21] Zalewski J., ed. Advanced Multimicroprocessor Bus Architectures. IEEE Computer Society Press, Los Alamitos, Calif., 1995.
- [22] Newport J. Avionic System Design. CRC Press, Boca Raton, Fla., 1995.
- [23] Teener M. A bus on a diet - the serial bus alternative: an introduction to the P1394 high performance serial bus. In: [21], pp. 180-194.
- [24] Leteinturier P., N.A. Kelling, U. Kelling. TTCAN from applications to products in automotive systems. In: In-Vehicle Networks, Safety Critical Systems, Accelerated Testing, Reliability. SP-1783, Society for Automotive Engineers, Warrendale, Penn., 2003, pp. 75-84.
- [25] Fuehrer T. et al. FlexRay – the communication system for future control systems in vehicles. In: In-Vehicle Networks, Safety Critical Systems, Accelerated Testing, Reliability. SP-1783, Society for Automotive Engineers, Warrendale, Penn., 2003, pp. 35-41.
- [26] Maier R. et al. Time-triggered architecture: a consistent computing platform, IEEE Micro, vol. 22, no. 4, 2002, pp. 2-11.
- [27] Miner P.S., M. Malekpour, W. Torres. A conceptual design for a reliable optical bus. Proc. DASC'02, 21st Digital Avionics Systems Conf., Irvine, Calif., October 27-31, 2002, vol. 2, pp. 13D3-1/11.
- [28] Homer M. Handling event driven messaging in distributed flight critical systems. Proc. DASC'02, 21st Digital Avionics Systems Conf., Irvine, Calif., October 27-31, 2002, vol. 2, pp. 13.C.4-1/6.

- [29] Nolte T., H. Hansson, L. Lo Bello. Automotive communication: past, current and future. Proc. ETFA'05, 10th IEEE International Conference on Emerging Technologies and Factory Automation, IEEE, New York, 2005, vol. 1, pp. 985-992.
- [30] Navet N., Y. Song, F. Simonot-Lion, C. Welwert. Trends in automotive communication systems. Proceedings of the IEEE, vol. 93, no. 6, June 2005, pp. 1204-1223.
- [31] Rushby J., A Comparison of Bus Architectures for Safety-Critical Embedded Systems. Report NASA/CR-2003-212161, Langley Research Center, Hampton, Va, March 2003.
- [32] Jonnalagadda V., M. Mathure, A. Kornecki, J. Zalewski. Considering local bus traffic in network performance simulations, Proc. CNDS'03 Communication Networks and Distributed Systems Modeling and Simulation Conference. Society for Modeling and Simulation, San Diego, Calif., 2003, pp. 109-114.
- [33] Williamsson C., D. Williamsson, J. Zalewski. 2004. A study of cluster computing over IEEE 1394. Proc. SAWAMAS-2004, 2nd Swedish-American Workshop on Modeling and Simulation. Cocoa Beach, Fla., February 2-3, 2004, pp. 209-217.
- [34] Trawczynski D., J. Sosnowski, J. Zalewski. A study of routing for the Bluetooth scatternet, Proc. PDS2004 IFAC Workshop on Programmable Devices and Systems, Krakow, Poland, November 18-19, 2004, pp. 473-478.
- [35] Sosnowski J., P. Gawkowski, A. Lesiak, Software implemented fault inserters. Proc. PDS2003, IFAC Workshop on Programmable Devices and Systems, Ostrava, Czech Rep., February 11-13, 2003, pp. 293-298.
- [36] Henriksson D., A. Cervin, K. Arzen. TrueTime: real-time control system simulation with MATLAB/Simulink. Proc. Nordic MATLAB Conference, Copenhagen, Denmark, October 2003.