http://dx.doi.org/10.1016/j.ress.2015.01.005
0951-8320/

Quantitative characterization of the reliability of simplex buses and stars to compare their benefits in fieldbuses

Manuel Barranco[a,*], Julián Proenza[a], Luís Almeida[b]

[a] *manuel.barranco@uib.es, julian.proenza@uib.es*
Dpt. Matemàtiques i Informàtica, Universitat de les Illes Balears, Spain
[b] *lda@fe.up.pt*
DEC-FEUP, Universidade do Porto, Portugal

Abstract

Fieldbuses targeted to highly-dependable distributed embedded systems are shifting from bus to star topologies. Surprisingly, despite the efforts into this direction, engineers lack of analyses that quantitatively characterize the system reliability achievable by buses and stars. Thus, to guide engineers in developing adequate bus and star fieldbuses, this work models, quantifies and compares the system reliability provided by simplex buses and stars for the case of the Controller Area Network (CAN). It clarifies how relevant dependability-related aspects affect reliability, refuting some intuitive ideas, and revealing some previously unknown bus and star benefits.

Keywords:
Reliability analysis, Stochastic Activity Network, fieldbus, Controller Area Network, star topology, bus topology.

1. Introduction

Dependability of wired fieldbuses is a matter of raising concern as complexity and criticality of distributed control systems are rapidly increasing [1, 2].

One key aspect of wired fieldbuses that has deserved special attention is the influence of the underlying network topology on the guarantee with which nodes communicate among them [3]. Specifically, in order to improve dependability in general and reliability in particular, fieldbuses targeted to critical applications offer the possibility of substituting the traditional bus by stars. This is seen in protocols like TTP/C, FlexRay, Switched Ethernet and the Controller Area Network (CAN) [3].

The central element of a star, e.g. an active hub, can provide better error containment by simply isolating faults at their ports of origin [3]. Moreover, stars are more robust since they are less prone to common-mode failures [4]. However, stars also

*Principal corresponding author. Phone: +34 971 172 992, Fax: +34 971 173 003, e-mail: manuel.barranco@uib.es

have more hardware components, which increases the probability that faults and errors occur and, thus, can have a negative impact on the final system reliability.

Network reliability analysis has been a traditional topic of interest of dependability evaluation. Most literature addresses large and medium scale networks, such as common information exchange WANs/LANs, smart/power grids, intra communication in power substations, wireless networks and multiprocessors interconnection networks, e.g. [5, 6, 7, 8]. In this sense, the major part of analyses focus on the connectivity among different nodes thanks to the structure of the topology [9], or on the Quality Of Service (QoS) and reliability of different network paths depending on resources constraints [10]. As concerns dependability analyses of fieldbuses, most of them tackle the correctness or the real-time guarantees of specific protocol mechanisms, e.g. [11, 12]. Thus they are not intended to asses how different topologies affect the capacity of nodes for communicating among them. Also, many fieldbus dependability evaluations consist of either qualitative studies or simulation-based/experimental fault-injection assessments, e.g. [13, 3, 14]. However, simulation is normally computationally more expensive than analytical approaches, thereby limiting the precision that can be achieved in practice. Fault injection, on the other hand, allows characterizing some dependability-related features, but it cannot ultimately quantify their contribution to reliability. Anyway, there are a number of works that do analytically quantify how the wired fieldbus topology influences the capacity of nodes for communicating. However, even the most recent ones abstract away important dependability-related features, e.g. [15, 16]. To the authors best knowledge, the only one of these analytical works that does not abstract important features is [4]. Unfortunately, it explains almost no detail about how it models the dependability, and it focuses on the integrity and availability impact of faults in a redundant ring topology.

It is surprising that, despite the efforts in developing star-based fieldbuses, the reliability benefits of buses and stars have not been adequately characterized yet. Thus, the objective of the present work is to throw light on this issue by analytically quantifying the system reliability achievable by these two fieldbus topologies depending on several important aspects. In fact, the influence of the aspects addressed here has been unknown (and even controversial) so far. Thus this paper can help system engineers in assessing when it is adequate to substitute a bus by a star and, in that case, what are the relevant aspects that have to be considered for attaining high reliability levels.

To achieve this objective, the authors of this paper take as a basis the models they presented in [17] and [18]. These models represent the reliability of two functionally-equivalent systems, one relying on a bus and the other one on a star. To the authors' best knowledge, no one has modelled a system relying on a bus or a star with the same level of detail. Moreover, these models include parameters that numerically represent many relevant dependability-related aspects. Thus, they are adequate to perform sensitivity analyses with respect to these aspects and, hence, to quantitatively characterize the system reliability achievable by a bus and a star.

Additionally, this paper further extends these models to include the influence of the reliability of power supplies. [17] and [18] did not consider this aspect because they focused on the communication subsystem, and power supply failures are addressed by mechanisms other than those provided at the communication level. However, the reliability of power supplies is an issue of increasing interest in highly-reliable systems, specially in the automotive domain [19].

Note that our models assume the *Controller Area network* (CAN) as the underlying communication technology. CAN was chosen because there is a clear interest in improving its dependability and real-time features. It is one of the most widespread,

mature and low-cost fieldbus technologies. Its application is expected to grow in several domains [20], including those in which criticality, and thus reliability, is an issue [1, 3]. In particular, the CAN star modelled here is CANcentrate [21]. It is the star that provides the best error containment for CAN. Also, it is fully-compatible with CAN applications and hardware components, which makes the comparison easier.

Anyway, the aspects analyzed here are not specific to CAN, but they influence reliability when using any other fieldbus technology. Specifically, it is always mandatory to analyze the reliability of any system with respect to the effectiveness of its fault-tolerance mechanisms, as this effectiveness is well known to have a big impact on its dependability. On the other hand, a star includes extra components when compared with a bus. Thus studying the impact of these components' reliability allows identifying which of them are worth to be improved and deserve a higher investment in terms of quality. Finally, it is mandatory to quantify the actual impact of the hub reliability. The hub is the only single point of failure of a star. Hence to invest in its quality or protection has been assumed as a key issue deserving of attention [22].

The paper is organized as follows. Sections 2 and 3 respectively summarize the characteristics of CANcentrate, and the way in which the reliability of systems relying on this star and on the CAN bus are modeled. The results of the sensitivity analyses are thoroughly described in Section 4 and summarized in Table 2. Finally, Section 5 further discusses the practical suitability of the models and Section 6 concludes the paper and points out future work.

2. CANcentrate basics

As depicted in Fig. 1 CANcentrate includes one hub to which each node is connected by means of a dedicated link containing an uplink and a downlink. The hub receives the contribution to each bit value of each node through the corresponding uplink, couples all non-faulty contributions with a logical AND function, and broadcasts the resulting signal back through the downlinks. The hub performs the AND-coupling in a fraction of the bit time. So it keeps the dominant/recessive transmission and the in-bit response properties of CAN [23], i.e. a dominant bit '0' prevails over a recessive bit '1' and nodes quasi-simultaneously observe every single bit on the channel. Moreover, each node is constituted by COTS components only: a microcontroller, a CAN controller and two transceivers [21].

The separation between the uplink and the downlink allows the hub to diagnose any faulty contribution and to disable it by isolating the corresponding uplink port. The fault model of CANcentrate includes faults that manifest as the transmission of syntactically incorrect frames, i.e. faults that generate stuck-at-recessive (STR), stuck-at-dominant (STD) or bit-flipping (FLIP) streams [21].

It is noteworthy that a STR fault that affects the medium prevents nodes from communicating in a CAN bus. But not in a CAN star as each node has a dedicated link to the hub. In contrast, a faulty node that transmits a STR stream does not affect the communication among the other nodes neither in a CAN bus nor in a CAN star, i.e. a STR node exhibits a *fail silent* behavior [17]. STD and FLIP faults, on the other hand, are considered as *blocking* or *disturbing faults* [17] independently of whether they occur in the media or in a node. This is because if these faults are not contained, the errors they generate propagate throughout the media.

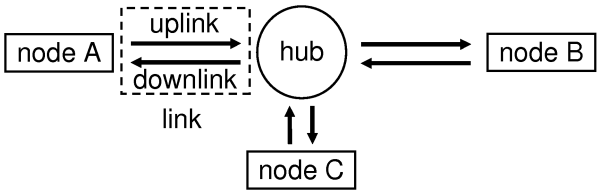

Figure 1: CANcentrate architecture

3. Modeling rationale

3.1. Reliability metric

Reliability is the probability with which a system continuously delivers its intended service throughout a given interval of time. In particular, to analyze the reliability of a distributed control system relying on a given topology it is necessary to choose a metric that takes into account not only the probability with which nodes operate, but also the probability with which they communicate among them. In other words, the metric has to properly reflect the contribution of the underlying communication infrastructure to the system reliability

Moreover, to show the benefits of any error-containment mechanism provided by the network, the metric also has to consider the system's ability to continue providing its service despite the loss or disconnection of nodes. Particularly, a CAN node disconnects itself from the network to not propagate errors. Additionally, the hub of CANcentrate disables a node if that node does not succeed in isolating itself. However these error-containment mechanisms become useless if the system does not accept/tolerate the disconnection of that node.

To reflect these considerations, we introduce the concept of *Fault-Tolerant-Accepting* (FTA) system [18], i.e. a system that correctly operates while accepting or tolerating the failure or the disconnection of up to k out of N nodes. These systems range from highly-reliable ones that tolerate faulty or disconnected nodes that are replicated, to non-critical systems that accept graceful degradation and continue operating despite some of their parts are inoperative. Then, the metric we use to calculate the reliability of those system is the so-called FTAR_k. This metric is defined in terms of k. It is the probability with which at least $N - k$ of the N nodes of a system can correctly operate and communicate among them throughout a given interval of time.

3.2. Models limitations

An important limitation of dependability modelling is that it is hardly possible to find numerical values that accurately quantify specific systems characteristics related to dependability. To mitigate this problem this work characterizes most of the system's dependability-related features by taking into account real implementation and technological aspects, as well as widely spread prediction standards (Section 3.3). Moreover, the sensitivity analyses herein presented consider a wide range of values for quantifying most of these system's characteristics. Thus they reveal what actually matters, i.e. the influence of the different features on dependability.

Our models consider permanent hardware faults caused by the aging or the stress suffered by components. They do not take into account faults provoked by external events. This is because it is extremely difficult to numerically quantify these events'

probabilities, e.g. the probability of a collision. Thus, the models do not show certain advantages of a star, e.g. the fact that it is more resilient to proximity faults than a bus. The models do not either consider temporary faults. Hence, they do not show the benefits a star can yield for this kind of faults, e.g. the hub of CANcentrate is able to isolate (and then reintegrate) temporarily faulty ports, thereby reducing the interference they generate. This work excludes temporary faults because the main goal of star topologies is to improve the system behaviour in the presence of permanent hardware faults. Transient ones are normally handled by mechanisms that are independent from the underlying topology, e.g. by the native CAN frame retransmission mechanisms. Moreover, the dependability impact of the temporary unavailability provoked by transient faults necessarily depends on the application, e.g. on the scheduling of a hard real-time system. Thus they are somehow beyond the scope of this work.

The analyses presented here must be understood as an estimation of the reliability advantages of simplex stars over simplex buses. They are not devoted to showing the full potential of a simplex star. Their purpose is to guide engineers in designing adequate fieldbuses based on simplex stars/buses depending on the quality of the hardware components, the way in which they fail, and the effectiveness of different fault-tolerance mechanisms.

3.3. Modeling assumptions

Some assumptions the models rely on determine their structure, whereas other ones are reflected in their parameters. Specifically, the starting point of this paper's sensitivity analyses is the case of reference we proposed in [17]. This case specifies default parameter values that can be considered as reasonable but that, at the same time, were chosen to favor the bus in the comparison. The models' main assumptions for this case are summarized next. Table 1 shows the default value of some of the parameters that are common to the CAN bus and CANcentrate, e.g. main coverages, the number of required/total power supplies, and the failure rate of the most representative components. A detailed explanation of all model assumptions and parameters can be found in [18].

The system is considered to be composed of the following components: power supplies, microcontrollers, CAN controllers, transceivers, memory ICs, oscillators, PCBs, segments of cable, connectors, network terminations, and an ASIC (in the case of the hub). It is assumed a bus length / star diameter of 100 m. The nodes of the bus are equidistant and they are interconnected following a daisy chain configuration, which minimizes the number of connectors [18]. In the star, every pair of nodes is supposed as separated by a length equal to the star diameter, with all links being 50 m long.

There is a wide variety of ways to power the nodes and their communication interfaces (transceivers and controllers) in fieldbuses like CAN. The specific cabling configuration of the power supply infrastructure is application dependant and, in some domains, the most reliable configuration is still a topic under research, e.g. in the automotive one [19]. What would be the best power cabling configuration to achieve a given degree of reliability is out of the scope of this paper. However, the models include the most unreliable elements of the power infrastructure, i.e. the power supplies. In particular, since the use of redundant power supplies is widely accepted to mitigate their high unreliability [24], the models assume that the system requires p power supplies and that it is provided with R of them, i.e. that up to $R - p$ power supply failures can be tolerated. Finally, the models do not abstract away the fact that the hub is an extra element that needs to be connected to the power supply infrastructure. In this sense, to keep the power supply cabling as orthogonal as possible to the communication network, it is assumed that each node (and the hub) is connected to this infrastructure by

Table 1: Most relevant models parameters common to CAN and CANcentrate

Parameter	Default value	Meaning
sysFauTolCov	1.0	System fault-tolerance coverage
ctrlFlipCov	0.95	CAN controller's FLIP error-containment coverage
hubFlipCov	0.95	Hub's FLIP error-containment coverage
powSupCov	0.95	Power supply fault-tolerance coverage
p, R	1, 2	Required power supplies and total number of them
powFR	$1.00000 \cdot 10^{-5}$	Power supply failure rate
connectFR	$2.07774 \cdot 10^{-8}$	Connector failure rate
wireFR	10^{-7}	Wire failure rate per kilometer
txrxFR	$6.73258 \cdot 10^{-7}$	Transceiver failure rate
ctrlFR	$1.25537 \cdot 10^{-6}$	CAN controller failure rate
microFR	$3.25312 \cdot 10^{-6}$	Node's microcontroller (and memory) failure rate
hubElecFR	$1.20843 \cdot 10^{-6}$, $1.87019 \cdot 10^{-6}$	Hub electronics failure rate for 3 and 15 nodes

means of a dedicated link of 3.5 m. This is half the maximum length of a devicenet dropline connecting a node to the main trunk line.

Component failures are considered as permanent, independent and not near-coincident. Each component's *Time To Failure* distribution is supposed to be exponential and Non-Defective, with mean $1/\lambda$, where λ is the failure rate expressed in hour^{-1}. Failure rates were calculated with a software [25] based on the MIL-HDBK-217 prediction standard [26] and the Tellcordia *Method I Case I* calculation method [27], while assuming a Mobile Environment and 40 Celsius degrees.

There is not a real consensus on the components failure mode proportions. Thus from the channel point of view a faulty component is assumed to manifest a STR, a STD or a FLIP fault [21] with equal probability. There are two exceptions: the microcontroller, which can only cause its node to transmit a STR stream; and the CAN controller, which delivers a STR and a STD/FLIP stream to the channel with proportions 66.6% and 16.6% respectively [18].

It is considered that every component can also cause an *out-of-fault-model* (*OFM*) failure. This mode was introduced to model the fault assumption coverage (*fauAsuCov*), i.e. to gather all faults that are treated by neither CAN nor CANcentrate. Anyway, the proportion of OFM failures is considered as 0% (*fauAsuCov* $= 100\%$). A larger value would mask the contributions of CAN and the star to the overall system reliability by the effect of faults they do not address. In fact, to benefit from any topology, the system must include mechanisms that deal with OFM faults. This is because it is impossible to increase its reliability by improving only one of its parts, i.e. the network and the power supply in this case.

The effectiveness of the fault-tolerance mechanisms were characterized by means of parameterized coverages. The first one is called *sysFauTolCov* and it represents the system ability to actually accept or tolerate the failure or disconnection of up to k of N nodes, provided that the system is able to accept or tolerate such a situation. In principle, it is assumed that sysFauTolCov $= 100\%$ since this is the most representative value for both: (1) FTA systems that intrinsically accept the failure or disconnection of up to k nodes, and (2) highly-reliable FTA systems, in which the probability of success of the fault-tolerance mechanisms is virtually of 100%, e.g. [28].

Other coverages were defined for the fault-tolerance mechanisms of the communication subsystem itself. In CAN and CANcentrate these coverages represent the probability with which a CAN controller and the hub successfully contain errors. Specif-

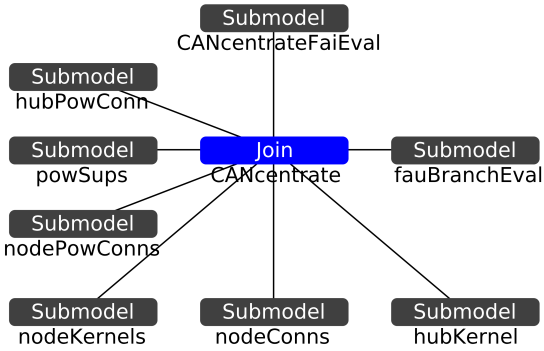

Figure 2: CANcentrate model

ically, both the controller and the hub are supposed to provide the same coverages: 100% for STR and 95% for STD and FLIP. A 100% coverage for STR is realistic, as it is extremely easy to detect an STR fault in both native CAN and CANcentrate. A 95% coverage for STD/FLIP was chosen because it is a common reference and conservative value in dependability evaluation. Moreover, to assume a 95% STD/FLIP coverage in both CAN and CANCentrate is detrimental for the star. The hub has a privileged view of the communications and, thus, it can detect STD/FLIP errors with higher accuracy than CAN nodes [18]. Finally, the coverage with which a power supply failure is tolerated by the redundant ones (*powSupCov*) is also assumed to be of the 95%.

3.4. Modeling strategy

The models were built using the *Stochastic Activity Networks* (SANs) formalism and the Moebius software [29]. SANs are an extension to *Stochastic Petri Nets* (SPNs) [29] that allow building a model as an intuitive and compact hierarchical composition of easily parameterizable submodels. SANs make it possible to flexibly specify and retrieve the results of different metrics. Moebius is able to transform models whose delays are exponentially distributed into a *Continuous Time Markov Chain* (CTMC) and, when so, to solve them by means of an exact (not approximate or statistical) method. This is the case of our models.

The modeling strategy proposed here consists in using three sets of SANs submodels [17]. They share specific places by using the *join* primitive [29]. The first set of SANs models the occurrence of faults and the kind of errors they generate. Then, when a fault occurs, the second set carries out what was called the *coverage process* [18], i.e. it evaluates how the errors propagate and are contained. Once the coverage process is finished, a SAN submodel evaluates whether or not the system is still able to deliver its service.

This strategy was implemented in two different manners [18] that yield the same reliability figures, but that differ in terms of performance. [17] describes the least efficient of these two implementations. [30] describes the most efficient one for the case of a system based on a replicated star called ReCANcentrate. Next, we summarize this last and most efficient implementation for the case of CANcentrate, as well as the extensions made in the current work to model the power supplies.

Fig. 2 shows the overall model of CANcentrate. The SANs are connected to each other by means of a join primitive that allows them to share common places.

The submodels at the bottom and at the left of the join are the SANs that model the occurrence of faults. The left ones has been added in the current work in order to model faults affecting power supply.

Note that one could explicitly model faults affecting each single hardware component. But then the resulting CTMC would track the state of each one of them, thus leading to an explosion of the state space. Fortunately, some modeling patterns have been proposed to mitigate this problem in Markov models [31]. One of them consists in not tracking the state of single components but of groups of them (subsystems). So what is actually represented is whether or not the subsystem as a whole is faulty. To lump components' states together in such a way is adequate as long as it is not necessary to differentiate between the state of each individual component of a given subsystem. Another common practice that allows reducing the state space further is not to represent the state of each component/subsystem, but to model the number of components/subsystems being in a given state [31], e.g. the number of processors that are either faulty or not faulty [29].

In the specific case of the models presented here, the different components are grouped into what we call Error-Containment Regions (ECRs). An ECR is a part of the system that includes different components that are isolated as a whole to prevent the propagation of errors generated by a fault affecting any of them. When a component of a given ECR fails then either, the component is isolated together with the rest of the components of its ECR, or the system fails. Note that in the best case all the components of a given ECR are isolated from the rest of the system when any of them fails. Thus, what is relevant to model is not the state of each individual component of an ECR, but whether or not the ECR as a subsystem is faulty or not.

The components of CANcentrate are grouped into several of the following ECRs: (1) *Node Kernels*, each of which basically includes a node's microcontroller and memory ICs; (2) *Node Connections*, each of which comprises all the components a given node uses to connect to the hub (e.g. a CAN controller, four transceivers, and the cables and connectors of the uplink and downlink); (3) the *Hub Kernel*, which basically represents the electronic components (ICs) that implement the hub coupling and fault-treatment functionalities; (4) *Power Supply*, which represents a power supply; and (5) *Node Power Connection* and *Hub Power Connection*, which respectively represent the cables and connectors that connect each node and the hub to the power supplies.

The CAN bus includes the same type of components, except those related to the hub. However, the bus error-containment mechanisms are different to those of the star. Thus the bus components are grouped into slightly different ECRs [18]. Specifically, a *Node Connection* of a CAN bus is an ECR that includes the CAN controller and the transceiver that attaches a node to the medium. But conversely to the case of CANcentrate, it does not include any cable or connector. This is because the CAN node can contain errors generated by its transceiver to some extent, but it can do nothing to contain errors generated by a cable or connector. In fact, the cables and connectors of the bus line are grouped into two different types of ECRs, i.e. the *Internal Bus Section* and the *Edge Bus Section*. They compose the bus line, which is assumed to follow a daisy chain (see Section 3.3). Each one of these two ECRs includes a section of the bus cable and a straight connector at both its ends. Additionally, the *End Bus Section* includes a termination resistor and therefore it has a slightly higher failure rate. For the sake of clarity note that in the out-of-date model implementation of [17] the ECRs were different. There, connectors and terminations where included within ECRs that represented the nodes. Thus [17] distinguished between what was called *Internal* and *End* nodes.

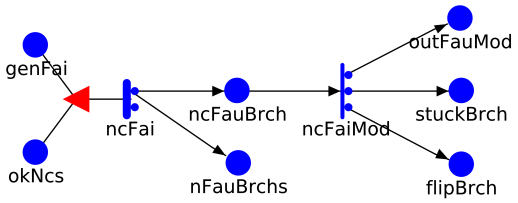

Figure 3: nodeConns submodel

The use of ECRs helps in reducing the state space. But to reduce it even further, we also applied the second strategy pointed out before, i.e. we did not explicitly model the state of each ECR. Instead, what is modelled is how many ECRs of each type is not faulty. As shown in Fig. 2, each SAN at the bottom and at the left of the join primitive does not necessarily represent just one ECR, but all the ECRs of a given type. *nodeKernels* represents all Node Kernels, *nodeConns* all Node Connections, *hubKernel* represents the Hub Kernel (just one), *powSups* all Power Supplies, *nodePowConns* all Node Power Connections, and *hubPowConn* just the *Hub Power Connection*.

To better understand how failures happening in the ECRs are modelled, Fig. 3 shows the details of *nodeConns*. The marking (number of tokens) of place *okNcs* represents the number of surviving Node Connections, whereas activity *ncFai* models the time that elapses until a fault happens in any of these ECRs. The failure rate of *ncFai* is calculated by multiplying the marking of *okNcs* by the failure rate of a single Node Connection. In turn, the failure rate of a single Node Connection is the sum of the failure rates of all its constituent components.

Once *ncFai* fires, it determines whether or not the Node Connection is attached to a hub port that has been already isolated due to a previous fault, i.e. due to a previous fault affecting the Node Kernel that corresponds to that Node Connection. This is fundamental to model how the errors propagate; because if the hub port is already isolated, the errors the faulty Node Connection generates cannot pass through that port.

Each one of these two cases is represented by a small circle attached to activity *ncFai*. The upper one models the situation in which the Node Connection that fails is located in a non-faulty branch, i.e. in a hub port that is not isolated yet. The lower one corresponds to the opposite situation. The proportion with which each case is selected depends on the number of surviving Node Connections (marking of *okNcs*); the number of branches that are already faulty, which is tracked by the marking of place *nFauBrchs*, i.e. $nFauBrchs \rightarrow Mark()$; and the total number of branches (a model parameter called *nBrchs*). For instance, the proportion of the upper case is obtained by dividing the number of surviving branches, i.e. $nBrchs - (nFauBrchs \rightarrow Mark())$ by the number of surviving Node Connections, i.e. $okNcs \rightarrow Mark()$.

No further action is done if *ncFai* takes its second case. This is because errors cannot propagate through and already isolated hub port. Conversely, when the first case is chosen, the model increases the marking of *nFauBrchs* to reflect that a new branch is faulty and, then, sets a token in *ncFauBrch*.

This place enables activity *ncFaiMod*. It instantaneously fires to decide what kind of errors the faulty connection generates, i.e. to model the Node Connection failure mode. This is an important decision as the coverage with which the hub isolates a faulty port depends on the kind of errors it observes at it. Since every surviving Node Connection is equal to each other, the proportion of a given *ncFaiMod* case is the prob-

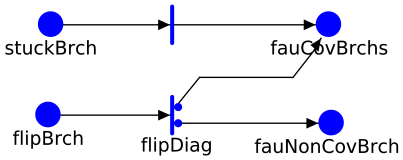

Figure 4: fauBranchEval submodel

ability with which a single Node Connection exhibits the corresponding failure mode. Moreover, faults were considered as not being near coincident in time. Thus, the proportion of an ECR failure mode in general, and of the Node Connection in particular, is the weighted arithmetic mean of the proportions witch which its constituent components exhibit that failure mode. Specifically, each component proportion is weighted considering the contribution of that component to the whole ECR failure rate [18].

The first case of activity *ncFaiMod* represents an OFM failure. The second and the third ones model a STR/STD and a FLIP respectively. When *ncFaiMod* selects its first case, the whole systems is considered as faulty, since the hub cannot isolate an OFM faulty port. Conversely, when *ncFaiMod* sets a token in *stuckBrch* or *flipBrch*, it triggers the execution of the above-mentioned *coverage process*; as it is necessary to determine if the hub contains the errors.

The coverage process is carried out by submodel *fauBranchEval*, which is located at the right of the join primitive in Fig. 2. This submodel shares the places *stuckBrch* and *flipBrch* with submodel *nodeConns* (see Fig. 4), *nodeKernels* and *nodePowConns*. If the hub contains the errors, then *fauBranchEval* transfers the token from *stuckBrch* or *flipBrch* to *fauCovBrchs*, whose marking represents the number of successfully isolated faulty branches. Otherwise, *fauBranchEval* moves the token to *fauNonCovBrch*. A token in this place is used to reflect that there is a faulty hub port that is not isolated and that, thus, pollutes the system with errors.

Since STR/STD failures are diagnosed with a coverage of 100%, a token in *stuckBrch* is always transferred to *fauCovBrchs*. In contrast, a FLIP failure cannot be always diagnosed. Thus, *fauBranchEval* transfers any token set in *flipBrch* to either *fauCovBrchs* or *fauNonCovBrch* in accordance to the corresponding coverage parameter (*hubFlipCov* in Table 1).

When the coverage process finishes, or a hub port exhibits an OFM failure, then submodel *CANcentrateFaiEval* (top of Fig. 2) becomes active. This submodel is the one who evaluates whether or not the system delivers its service. *CANcentrateFaiEval* becomes aware of component failures by sharing different places with the other submodels (see Fig. 5). Then, it takes a decision depending on the number of nodes that can still operate and communicate with each other. For example, imagine that a new token arrives at *fauCovBrchs* from *fauBranchEval*. Then *CANcentrateFaiEval* checks if this place's marking exceeds the value of a parameter called *kSevere*. This parameter specifies the maximum number of faulty or disconnected nodes the system accepts / tolerates. If so, *CANcentrateFaiEval* diagnoses the system as faulty and indicates it to the rest of submodels by setting a token in the shared place called *genFai*. Otherwise, *CANcentrateFaiEval* takes into account the coverage *sysFauTolCov* to evaluate if the system actually tolerates the disconnection of that new branch.

Finally, Fig. 6 shows one of the additional SANs we have included in the current work to model faults affecting the power supplying. This SAN, *powSups*, models the

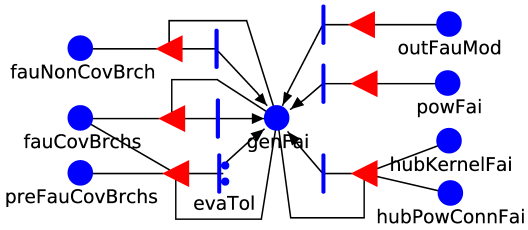

Figure 5: CANcentrateFaiEval submodel

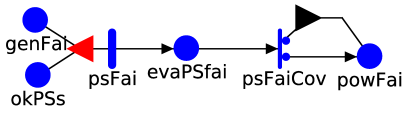

Figure 6: powSups submodel

occurrence and treatment of faults happening at any Power Supply. *okPSs* represents the number of surviving Power Supplies and its initial value is R. Activity *psFai* models the failure of any of them. When it fires, it erases one token from *okPSs* and transfers it to *evaPSfai*. Then, activity *psFaiCov* instantaneously fires. The lower case of this last activity represents the situation in which the faulty Power Supply is not covered. It sets a token in *powFai*, which is immediately transferred to place *genFai* by the SAN *CANcentrateFaiEval* (Fig. 5). The upper case of *psFaiCov* is chosen when the Power Supply failure is covered. This case triggers the execution of the output gate it is attached to (black triangle). If the number of surviving Power Supplies - marking of *okPSs* - is greater or equal to the number of Power Supplies required by the system, p, then it sets a token in *powFai*. Otherwise, the output gate does nothing so that the system remains operational.

4. Results of the sensitivity analyses

This section assesses the sensitivity of the reliability achievable by two functionally-equivalent systems, one relying on CAN and the other on CANcentrate, with respect to several key aspects. To assess the influence of each one of these aspects, this section carries out a *parametric sensitivity analysis* [32]. It takes as a starting point the case of reference summarized in Section 3.3 and Table 1. Then, it varies the value of the model parameter/s that characterize a given aspect.

The reliability metric used in this Section is the FTAR_k (Section 1). It is measured for $k = 1$, as this value is the one for which the star intuitively yields the least benefits when compared with the bus [18]. Note that in [17] the FTAR_1 was referred to as the *probability of not suffering a severe failure*.

To make results as visually clear as possible, the analyses do not plot how the values of the parameters influence the evolution of the FTAR_1 in time. Instead, they show how these values affect the *Mission Time* (MT). The MT is the maximum amount of time during which a system (in this case a system based on CAN or CANcentrate) exhibits a reliability equal to or greater than a certain threshold [33]. The analyses presented here use a reliability (FTAR_1) threshold of 0.99999, which corresponds to the one required

by the least-demanding x-by-wire car applications during a MT of 10 hours [33]. Each analysis considers the cases of systems with 3 and 15 nodes, to address small highly-reliable embedded systems and typical in-vehicle subnetworks [34] respectively.

Since the MT has a direct relationship with the achievable system reliability (FTAR_1), for the sake of clarity each analysis will be discussed in terms of the MT only.

Finally, the models yielded the following performance results. Each experiment (each instantiation of the model parameters) of the CAN bus with 15 nodes was converted into a Markov Chain of 152 states in $1.6 \cdot 10^{-2}$ sec; whereas CANcentrate with 15 nodes resulted in 108 states and $1.2 \cdot 10^{-2}$ sec. In both cases, each experiment was solved in $4 \cdot 10^{-3}$ sec.

4.1. MT vs Number and Coverage of Power Supplies

As already said, power supplies are more unreliable than the other components that constitute a computer system. For instance, Table 1 shows that the failure rate of a power supply is one order of magnitude greater than the rate of devices such as microcontrollers or CAN controllers. A common practice to cope with this problem is to use redundant power supplies [24].

Prior to decide how many power supplies to use, R, it is necessary to determine what is the number of power supplies the system requires, p. Note on the one hand that typical power supplies for computer systems provide a voltage from less than 1 V to 12 V and an amperage of less than 1 A to 20 A. On the other hand, devices such as microcontrollers and vehicle *Electronic Control Units* (ECUs) require voltages in the range of $[2.5, 5.0]$ V and amperages of the order of 300 mA [35]. Surprisingly, CAN transceivers consume relatively high amperages around 70 mA [36]. This means that with an average single power supply of 10 A, and since the hub core is much simpler than a microcontroller, it would be possible to power a CAN bus of up to 27 nodes and a CANcentrate of up to 17 nodes. This difference is mainly due to the fact that each CANcentrate node needs four transceivers (a pair located within the node and another pair at the hub), whereas each CAN node only needs one.

For the number of nodes considered in this work, one power supply of 10 A suffices. For a higher number of nodes, one could use a power supply of 20 A, thus being able to power a CAN system of 54 nodes and a CANcentrate one of 34. A value of $p = 1$ is assumed hereafter (Table 1); analyses for higher number of nodes and power supplies is postponed for future work.

Given $p = 1$, Fig. 7 shows the MT achieved with CAN and CANcentrate when using $R = 1, 2, 3$ power supplies. It considers two values for the coverage with which a faulty power supply is covered by the surviving ones, namely the default reference value ($\text{powSupCov} = 95\%$), and a perfect coverage ($\text{powSupCov} = 100\%$).

Results show that it is fundamental to use redundant power supplies for CAN and CANcentrate, as the MTs that can be achieved with just one power supply is very low in both cases. The coverage is also very important. When the number of nodes is low or the underlying topology is a star, the MT significantly increases when using redundant power supplies and a perfect coverage. In fact, Fig. 7 reveals that the benefits of a star when compared with a bus increase when the power supply coverage increases.

Results indicate that it is not worthy to invest in more than 2 power supplies. When the coverage is perfect, 100%, the improvement of MT achieved when using 3 power supplies instead of 2 is negligible. Moreover, if the coverage is imperfect, 95%, to triplicate the number of power supplies is even counter-productive. Thus, it can be concluded that when $p = 1$, the best is to have $R = 2$ power supplies. Therefore, from

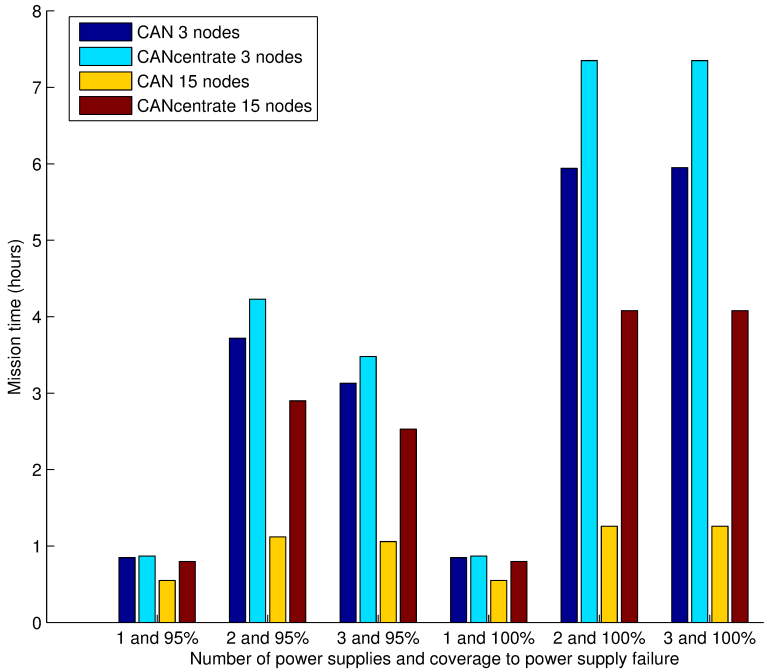

Figure 7: Mission Time (MT) vs Number of power supplies and coverage to power supply failure

now on, the rest of analyses are carried out considering this value of R as a reference (see Table 1).

Another interesting result is that CANcentrate only achieves a noticeably better MT than CAN when the system includes redundant power supplies. Otherwise, the benefit of CANcentrate is almost negligible, specially when the number of nodes is low.

4.2. MT vs System fault-tolerance coverage

This section analyzes the MT with respect to the ability of the system to actually tolerate the disconnection/loss of a node. The parameter that represents this coverage is called *sysFauTolCov* (Table 1). It is noteworthy that in highly-reliable systems this coverage is typically achieved by means of fault-tolerance techniques such as active node replication, in which each node replica executes the same code. Thus, since CANcentrate is transparent for any CAN-based application [18], the *sysFauTolCov* a system provides is independent from whether its underlying communication network is CAN or CANcentrate.

Fig. 8 shows that the star is better than the bus when *sysFauTolCov* $> 97\%$ and $> 89\%$ for 3 and 15 nodes respectively. This indicates that, in order to take advantage from a star topology, a system (specially a small one) must include effective enough fault-tolerance mechanisms.

However, note that it is not necessary for the system to provide an extremely high *sysFauTolCov*. The MT achieved with a *sysFauTolCov* of 99.9% is very close to the one achieved with a *sysFauTolCov* of the 100%. This is a significant result as it shows that typical coverages of highly-reliable systems are more than enough for taking advantage from a star topology over a bus. For instance, this would be the case of a military Self-

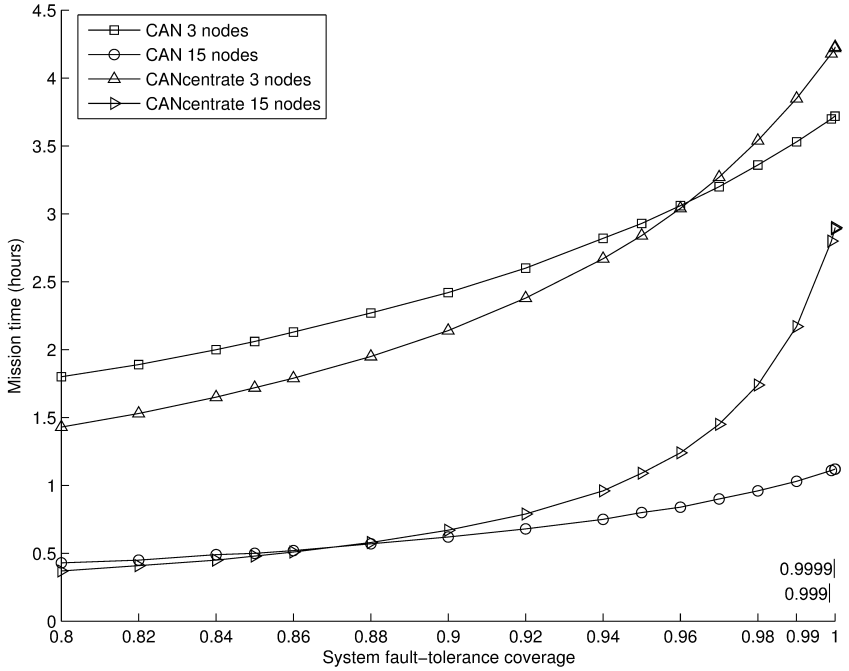

Figure 8: Mission Time (MT) vs System fault-tolerance coverage

Repairing Flight Control System (SRFCS), which normally achieves fault-tolerance coverages of the order of 99.99% to 99.9992% [28].

4.3. MT vs Hub's error-containment coverage

The benefits of a simplex star over a bus are mainly due to the ability of the hub for containing errors. Thus this section assesses the sensitivity of the star reliability with respect to this ability. Note that the coverage with which a hub contains errors depends on the kind of errors. The hub of CANcentrate diagnoses stuck-at faults with a perfect coverage [17]. Hence, the only type of errors that can actually propagate from one node to the others in CANcentrate are those generated by bit-flipping faults. Therefore, this section analyzes how the reliability benefits of CANcentrate vary depending on the coverage with which its hub contains bit-flipping streams, i.e. with respect to the bit-flipping coverage (*hubFlipCov* in Table 1).

Fig. 9 indicates that the sensitivity of the star with respect to *hubFlipCov* is significant and that it increases with the number of nodes, e.g. if *hubFlipCov* drops from 100% to 0%, the MT is reduced by 47% with 3 nodes and by 77% with 15. This confirms the intuition about the relevance of the error-containment coverage provided by the hub; specially when the number of nodes increases, as a higher number of nodes implies a more frequent occurrence of faults.

Furthermore, results qualify this coverage relevance by revealing two previously unknown features. First note that the star presents a higher reliability than the bus as long as *hubFlipCov* $> 79\%$ with 3 nodes and $> 41\%$ for 15 nodes. The lower required value of *hubFlipCov* for 15 nodes may seem counterintuitive. This is because, as just said, the sensitivity of the star reliability with respect to this coverage increases with

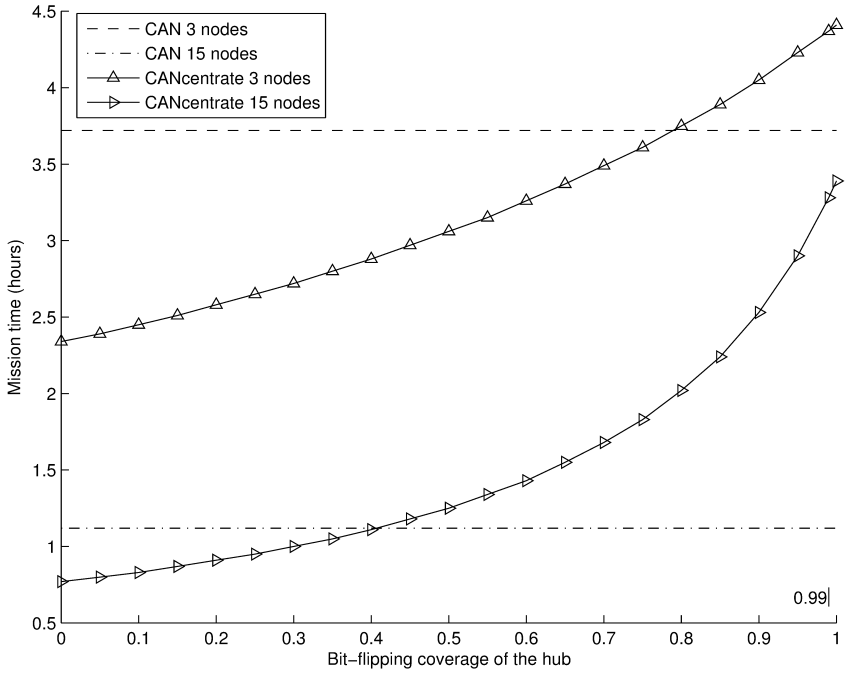

Figure 9: Mission Time (MT) vs Hub's error-containment (bit-flipping) coverage

the number of nodes. However, what these results indicate is that the bus reliability is much more sensitive than the star to the number of nodes due to the bus scarce error-containment features. Thus, in practice, the efforts devoted to improving the coverage provided by the hub in order to outperform the bus can be somehow relaxed as the number of nodes increases.

Second, results also reveal that it is not worth the effort to achieve a hub with a nearly perfect coverage, i.e. 100%. In particular, for the case of a CANcentrate-based system, Fig. 9 shows that there is no significant improvement in the MT when *hubFlipCov* is increased above the 99%, for both 3 and 15 nodes.

4.4. MT vs Disturbing faults proportion

Previous section shows that the advantage of a star over a bus significantly depends on the hub's ability for containing errors that, otherwise, would propagate through the network. Thus, from an intuitive point of view, this suggests that the benefit of a star also depends on the proportion with which faults actually manifest by generating those errors, i.e. on the proportion of *disturbing faults*. Intuitively, the benefit of the star should be higher as the proportion of disturbing faults increases.

To clarify this issue current section analyzes the sensitivity of both, the bus and the star, with respect to the proportion of bit-flipping faults. *flipProp* was varied while keeping stuck-at-recessive and dominant equiprobable with respect to each other. The proportion of bit-flipping faults was chosen because they are the ones that are more likely to propagate (and thus disturb the communication) in the CAN bus. This is because each node's CAN controller contains with a perfect coverage most of stuck-at faults happening in its node [18]

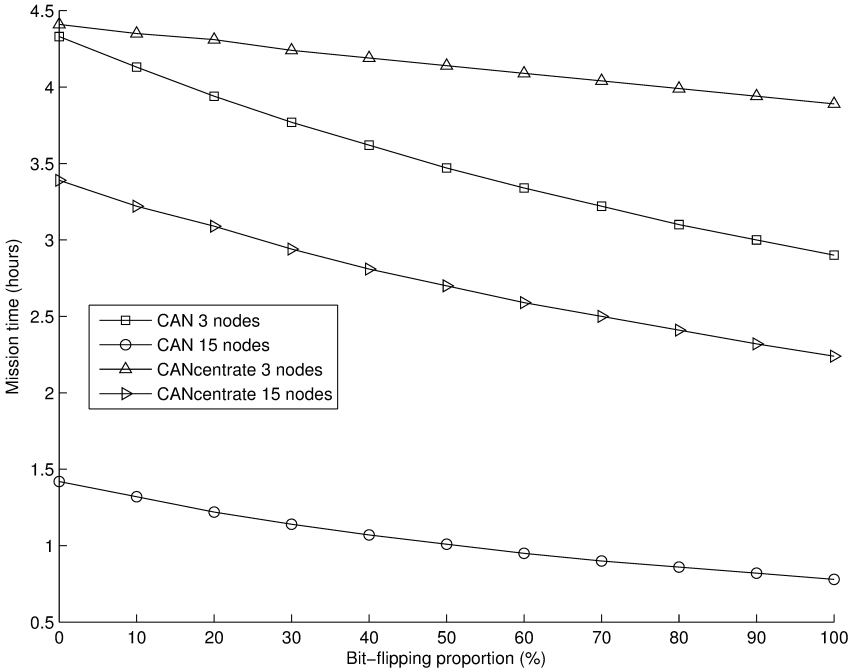

Figure 10: Mission Time (MT) vs Disturbing faults (bit-flipping) proportion

flipProp was varied for all components except for the node microcontroller (and memory). The microcontroller is supposed to fail only in a stuck-at-recessive manner (Section 3.3), so that it cannot pollute the channel with errors. This is because to assume that a faulty microcontroller can compel its CAN controller to send a faulty bit stream other than a stuck-at-recessive one is extremely unrealistic.

Fig. 10 confirms the intuitive idea that the MT improvement the star yields whit respect to the bus is more evident as the proportion of disturbing faults (*flipProp* in this case) increases. Nevertheless, it also reveals that this improvement decelerates as the number of nodes grows. Specifically, when 3 nodes are considered and *flipProp* varies from 0% to 100%, the MT improvement the star provides increases from the 1.8% to the 34%. However, for the case of 15 nodes, the MT improvement only increases from the 139% to the 187%

To better understand this last result note that the bus can barely contain errors and, hence, a single fault is enough to block the communication. This makes the bus specially vulnerable to the proportion of disturbing faults regardless of the number of nodes. For instance, Fig. 10 shows that when *flipProp* increases from 0% to 100%, the MT of the bus decreases by the 33% with 3 nodes and by the 45% with 15. In contrast, the hub's error-containment mechanisms make the star more resilient to disturbing faults. Therefore, the star MT is hugely affected by the proportion of disturbing faults only when the residual probability with which the hub does not isolate them is high enough. In particular, Fig. 10 reveals that the number of nodes has a noticeable impact on this residual probability. A star with 15 nodes is more vulnerable to *flipProp* than a star with 3, e.g. when *flipProp* increases from 0% to 100%, the star MT decreases by the 12% with 3 nodes, but by the 34% with 15.

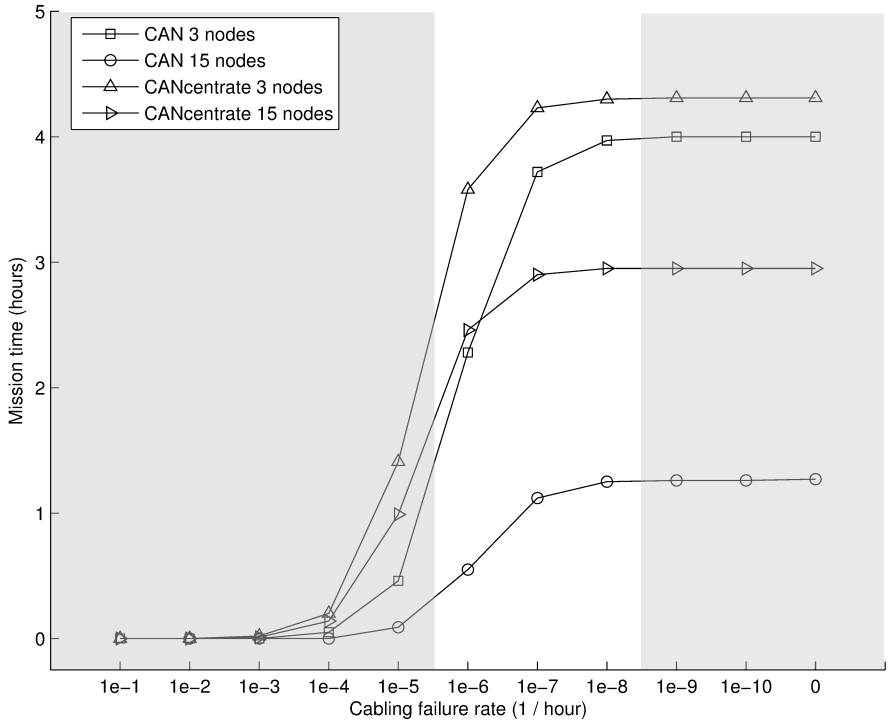

Figure 11: Mission Time (MT) vs Cabling failure rate

Finally another interesting result is that, independently of the number of nodes, the star outperforms the bus for any proportion of disturbing faults. In fact, the star is better even when this proportion is of the 0%. This is an argument in favor of using stars instead of buses, as it is not necessary to suffer from a huge proportion of disturbing faults to benefit from the star's better error containment.

4.5. MT vs Cabling failure rate

As mentioned, the star includes extra hardware when compared with the bus. Since a higher number of components implies a higher probability of fault occurrences, it would be easy to assume that the star is not adequate for substituting the bus when the quality of the star's extra hardware components is poor. This section is devoted to clarifying whether or not this statement holds for the case of the *cabling*.

The cabling includes two types of components, namely wires and connectors. Specifically, each CAN cable, i.e. each uplink and downlink, and each bus segment connecting two adjacent nodes, includes 2 wires [23] (CAN_H, CAN_L) and two connectors. Likewise, each power cable comprises 2 wires (Vcc, GND) and has a connector at both its ends. In the analysis presented here the order of magnitude of the cabling failure rate is varied. To keep the relative weight of wires and connectors as in the default case, the failure rate of the connectors is always considered one order of magnitude lower than that of the wires. Note from Table 1 that in the default case the failure rates are $2.07774 \cdot 10^{-8}$ hour^{-1} per connector and 10^{-7} hour^{-1} per km of wire.

For the sake of clarity, the x-axis legend of Fig. 11 refers to the order of magnitude of the wire failure rate only. The figure considers these failure rates ranging from

10^{-1} to 0 hour^{-1}. The rates are arranged in descendent order in the x-axis, so that a failure rate of 0 hour^{-1} corresponds to the ideal case in which the cabling cannot fail. Anyway Fig. 11 shades in the area corresponding to the values of the range that cannot be considered as realistic according to the MIL-HDBK-217 standard [26].

Fig. 11 shows that CANcentrate leads to a higher MT than CAN for any failure rate of the cabling. Specially when this failure rate is one order of magnitude higher than the default case, i.e. when it increases from 10^{-7} to 10^{-6} hour^{-1}. This result is enlightening. It refutes the intuitive idea that, given its extra cabling when compared with a bus, a star is inappropriate when using wires and connectors of poor quality.

This result can be explained by the fact that in a star the hub provides containment to errors generated at faulty wires and connectors of the communication network, whereas in a bus nodes almost can do nothing to contain them; as a consequence the hub clearly compensates the star additional cabling.

Moreover, Fig. 11 shows that almost for any failure rate the benefits of the star are more evident with 15 nodes. This demonstrates that the hub is fundamental for containing errors generated by an increasing number of wires and connectors.

Another important result is that a star-based system exhibits sustained high reliability before the cabling failure rate reaches a high level of 10^{-5} hour^{-1}, which is already a non-realistic (too high) value. This indicates that the star can improve reliability without much concern on the cabling.

Conversely, the cabling quality is crucial in CAN. On the one hand, the reliability of the bus is significantly reduced when the order of magnitude of the cabling failure rate is increased by one order of magnitude with respect to the default case, i.e. from 10^{-7} to 10^{-6} hour^{-1}. On the other hand, when few nodes are considered, the bus reliability could be noticeably improved by reducing this order of magnitude from 10^{-7} to 10^{-8} hour^{-1}. However, this potential enhancement of the bus presents practical limitations, as wires and connectors are mechanical parts and it is hardly possible to significantly improve their reliability beyond the default case.

4.6. MT vs Transceiver failure rate

The current section assesses the influence on the reliability of the other type of component whose number is larger in a star than in a bus, i.e. the transceiver. In general, note that in a star two transceivers per node are needed (each one at a given extremity of its link), whereas in the bus each node only needs one transceiver to connect to the bus line. In the particular case of CANcentrate, each node needs four transceivers; a pair for connecting the node to the uplink and the downlink, and another pair to connect the uplink and the downlink to the hub [18].

Transceivers are analyzed apart from the cabling because, conversely to wires and connectors, they are not mechanical components, but electronic ones. This is a relevant difference because the failure rate of electronic devices can be decreased much more easily than such of mechanical parts by investing in their quality. Thus, conversely to the cabling, any conclusion related to the potential benefits of improving the reliability of the transceivers should be taken into account from a practical point of view.

Moreover, note that in general it is more likely to suffer from a permanent fault affecting a transceiver than from a permanent fault occurring at the cabling. The default failure rate of a transceiver is of the order of 10^{-7} hour^{-1}. The failure rate of an uplink (or downlink) of 50 m is of the order of 10^{-8} hour^{-1}, taking into account that the default failure rates of connectors and wires are of the order of 10^{-8} hour^{-1} and 10^{-7} hour^{-1} per km respectively. This means that the sensitivity of the bus and the star

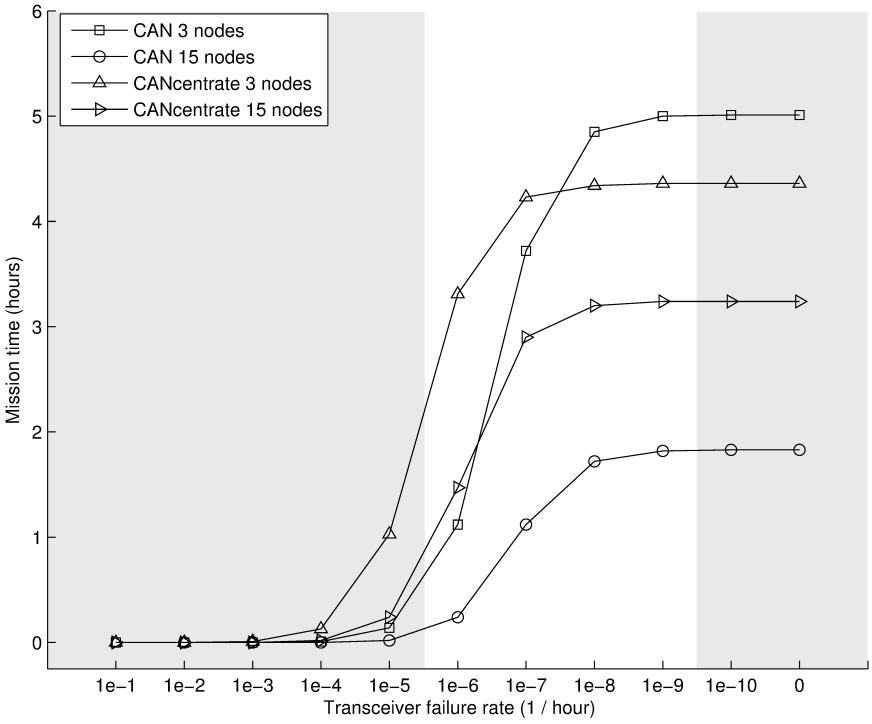

Figure 12: Mission Time (MT) vs Transceiver failure rate

with respect to the transceivers' reliability could be different from the sensitivity with respect to the cabling.

Fig. 12 shows how the MT is affected when the order of magnitude of the failure rate of the transceiver is changed. The x-axis specifies the order of magnitude of these rates (in descendent order) and shades in the area of unrealistic failure rates.

Surprisingly, results show that the bus is the topology that benefits the most from improving the reliability of the transceivers when the number of nodes is low. In this case the bus outperforms the star when the transceiver failure rate is decreased by one order of magnitude with respect to the default case (from 10^{-7} to 10^{-8} hour^{-1}). This greater sensitivity of the bus may seem counterintuitive, as the star includes a higher number of transceivers. However, the bus can scarcely contain errors generated by transceivers when compared with the star. Thus, the weight of the transceiver reliability on the reliability of the overall system is greater when using a bus than when using a star.

The fact that a small bus-based system is extremely sensitive to the transceivers should be taken into account in practice. As already pointed out, it is relatively easy to decrease the transceivers' failure rate below the default case by investing in their quality. Indeed, to improve the reliability of a small system, it could be preferable to invest in the quality of the transceivers than to substitute the bus by a star topology.

Anyway, Fig. 12 also reveals that the star is still the best choice in terms of reliability when the number of nodes increases. As can be seen, the degree in which the bus reliability can be improved by means of better transceivers decreases with an increasing number of nodes. In fact, with 15 nodes, the bus is worse than the star for any

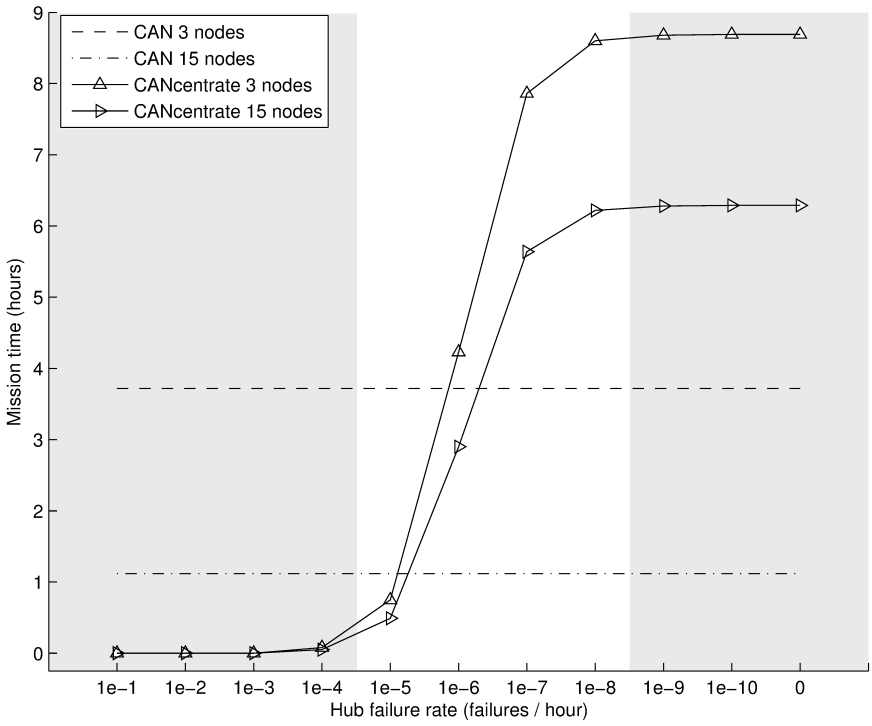

Figure 13: Mission Time (MT) vs Hub failure rate

given transceiver failure rate. To better understand this, note that transceivers are much more reliable than other components of the nodes, e.g. the microcontroller. Therefore, the transceivers reliability, even when it is improved further, becomes less relevant as the number of these other components also increases.

Another important result is that to use transceivers of good-enough quality is an issue deserving of attention, independently of both the topology and the number of nodes. Fig. 12 shows that when the transceiver failure rate is increased from 10^{-7} to 10^{-6} hour^{-1} the MT achieved by a bus-based system is reduced by 70% and 79% for 3 and 15 nodes respectively. Conversely, the MT of a star-based one is decreased by 22% and 49% for 3 and 15 nodes respectively.

4.7. MT vs Hub failure rate

Since the hub is the only single point of failure of a star, it has been traditionally assumed that it is a key factor in the reliability achievable by a star. In fact it is normally placed in a well-protected zone or it is even replicated [22]. However, the actual impact of the hub has not been quantified yet.

To fill this gap, the current section analyzes the reliability achievable by a star-based system for a wide range of hub failure rates (parameter *hubElecFR* of Table 1). Note that these failure rates correspond to the part of the hub that implements its actual coupling and fault-treatment functionalities, i.e. the set of electronic components that were referred to as the *Hub Kernel* in Section 3.4. The influences of other electronic and mechanical parts of the hub, e.g. the transceivers and connectors, have been taken into account in previous sections.

Although the influence of the hub has been considered as relevant, results of Fig. 13 are outstanding as they quantitatively expose that the hub reliability is actually of utmost importance. They show that the MT can be hugely improved by reducing the magnitude of the hub failure rate. For small systems, e.g. with 3 nodes, decreasing the magnitude of the failure rate from its default value, 10^{-6}, to 10^{-7} increases the MT from 4.23 to 7.86 h, a 86% improvement with respect to the default case. Further reducing it to 10^{-8} yields a MT of 8.60 h, i.e. 103% improvement. Likewise, for larger system, e.g. with 15 nodes, when the magnitude of the hub default failure rate decreases from 10^{-6} to 10^{-7} and 10^{-8}, the MT improves from 2.90, to 5.64 and 6.22 h respectively, i.e. the MT approximately improves by 95% and 114%.

Moreover, recall from Section 4.1 and Fig. 7 that the benefits of a star are higher as the power supply coverage (*powSupCov*) increases. Although not shown in Fig. 13, with a nearly perfect *powSupCov*, the MT of the default case of CANcentrate improves by 916% and 281% for 3 and 15 nodes if hub failure rate is of the order of 10^{-8}.

These results encourage the use of a simplex star in practice. The internal part of the hub is mainly constituted by electronic components. Thus, very low failure rates can be achieved, either by internally replicating it, or by using high-quality components in its construction. Moreover, by using low but still realistic hub failure rates, the MT is very close to that achieved with a perfect hub with a failure rate of 0 hour^{-1}.

Finally, Fig. 13 quantitatively corroborates that the quality of the hub cannot be disregarded, e.g. with 15 nodes the MT is drastically reduced from 2.90 to 0.49 h (around 83%) when the hub failure rate magnitude increases from 10^{-6} to 10^{-5} hour^{-1}. This result shows that the star becomes worse than the bus if the hub is not reliable enough.

5. Discussion

The sensitivity analyses presented above reveal for the first time how different relevant aspects of systems relying on simplex buses and stars affect reliability. Current section, instead, is devoted to further discuss how the models can guide in the design of reliable systems based on these topologies.

We have shown that nodes are the most unreliable elements of a system and that, as a consequence, to attain a high degree of reliability the system must be provided with the capacity to accept/tolerate their failure/disconnection. We refer these systems to as FTA systems. For instance, building/home automation systems can still provide a proper degree of service despite the failure/disconnection of a certain number of nodes. On the other hand, critical systems accept/tolerate the failure/disconnection of nodes thanks to either node replication or intrinsic redundancy. Node replication is extensively used in many domains such as nuclear power plants, medical equipment, railway switching, fly-by-wire and drive-by-wire. Intrinsic redundancy can be exploited in domains such as x-by-wire and robotics. For example, the laws that control the operation of a car brake-by-wire system, multicopters and autonomous underwater/remotely-operated vehicles can be dynamically adapted to respectively tolerate the failure of brake actuators, rotors and thrusters [37, 38, 39].

As concerns the network topology, note that replicated ones are used in pure electronic control systems that need to be fully fail-operational, and in which nodes are usually triplicated or quadruplicated [33, 40]. However, to fully replicate an electronic distributed control system and its network has associated higher production and certification costs. In many applications these extra costs do not compensate the gain in terms of reliability [33, 41]. Different alternatives can be used instead. One possibility is to use a simplex topology, replicate just the most critical nodes and, then, use a

backup strategy either to provide graceful degradation or to ensure that the system fails in a safe manner. Specifically, it is possible to use an emergency hydraulic/mechanical backup [33, 42, 37] or to rely on the partial compensation provided by other electronic control subsystems [37, 42]. Another option is to take advantage from the natural redundancy/symmetry of the system and, hence, connect half of the nodes to one simplex network and the other half to another simplex one e.g. [43]. In case one of the networks fails, half of the nodes can still operate and communicate to provide a gracefully degraded service or to reach a fail-safe state.

In any case, to use a robust-enough simplex topology is essential to attain the reliability required by systems in which extra costs due to full redundancy do not pay off. In particular, error containment is vital in those systems, since accepting/tolerating a node failure becomes useless if the errors a fault generates are not prevented from propagating to the rest of the nodes. In this context, simplex stars are supposed to outperform buses, as the star's center has a privileged location within the system to contain errors.

From the above discussion we believe that, from an engineering point of view, it is interesting to assess how the reliability of FTA systems that rely on simplex buses and stars can be improved just by enhancing the simplex topology itself, specially error containment.

Next, as an example, we sketch how the models we propose can guide in the design of a reliable brake-by-wire (BBW) system. Automotive is an industry in which the investment in terms of full redundancy is clearly controversial [33], and in which customers are still somehow reluctant to rely on pure electronic-based control systems [44]. In fact, nowadays intra-vehicle networks mostly rely on simplex topologies [45, 2] and make use of hydraulic/mechanical backups [45], e.g. as in hybrid regenerative vehicles. Thus, let us consider a classical BBW system relying on a simplex topology and in which safety is enforced by means of a physical backup. Specifically, consider a typical BBW [41] composed of a Central Brake Management (CBM) ECU, a Brake Pedal Sensor (BPS) ECU and four Wheel Brake (WB) ECUs. Since brake control algorithms can intrinsically compensate the loss of at least one WB ECU [37], imagine that the BBW is able to do so. In addition, to increase the BBW reliability let us assume that the CBM, the BPS and the power supply are duplicated. With this configuration, the BBW system tolerates the failure/disconnection of 1 out of 8 nodes and the failure of one power supply.

As concerns what reliability metric to use, note that the FTAR_k is defined to be as general as possible. In this sense, it assumes that the system accepts/tolerates k node failures, independently of the node that fails. In some cases this assumption may not exactly reflect the fault-tolerance capacity derived from a given node redundancy strategy. In those cases, though, an appropriate value of k can be chosen to use the FTAR_k as a lower bound to the system reliability. Moreover, given the flexibility of SANs and their reward formalism, our models can be adapted to adjust the metric to the reliability of a specific redundancy configuration. Anyway, the FTAR_1 matches the reliability of the BBW proposed here.

At this point, imagine that although the BBW is provided with a backup, we are interested in assessing the achievable Mission Time (MT) when imposing the typical reliability requirement of a BBW with no backup, i.e. 0.99999999 [33]. When trying to achieve the highest MT as possible for this system, it is important to recall that it is almost unfeasible to decrease the connectors and wires failure rates (FRs) below their default case values, i.e. below 10^{-8} and 10^{-7} hour^{-1} respectively. Thus, we discard investing in these components. Conversely, we consider investing in the quality of the electronic components (microcontrollers, CAN controllers, transceivers and the hub

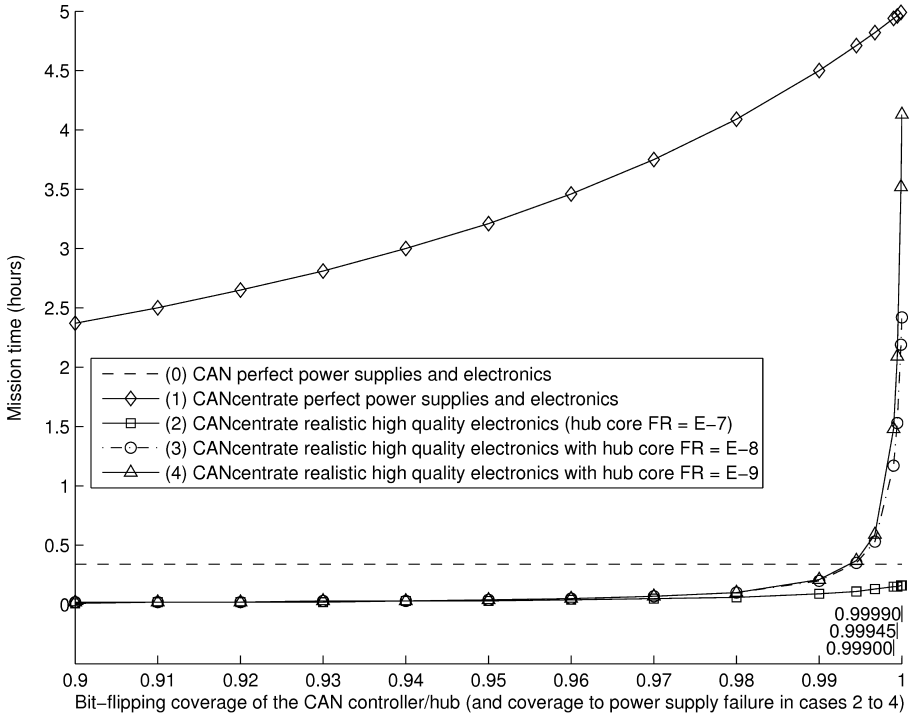

Figure 14: BBW Mission Time (MT) vs coverage

core), as well as on the coverage of the bus/star fault-tolerance mechanisms.

For the bus the coverage that can be improved is the one corresponding to the error containment provided by each node CAN controller (or by each node's bus guardian in case nodes were provided with them). Fig. 14 shows the BBW MT when this coverage ranges from 90% to 100%, assuming an ideal case in which the power supplies and the electronic components do not fail. Results reveal that the MT is kept in a constant low value regardless of the coverage. This indicates that the bus MT is strongly limited by the connectors and wires reliability and, thus, it is not a good option for a BBW system.

On the other hand, we measured the BBW MT depending on the coverage with which the hub contains bit-flipping errors. We assessed the following 4 star configurations (Fig. 14): (1) using ideal power supplies and electronic components that do not fail; and (2 - 4) using power supplies of average quality, but electronic components of the highest one. Cases (2) to (4) consider the same components FRs, which are around 10^{-7} hour^{-1}; thus they only differ from each other in the hub core FR, i.e. (2) 10^{-7}, (3) 10^{-8} and (4) 10^{-9}. Also note that conversely to the case of the bus and the first star configuration, (2) to (4) assume imperfect power supplies of average quality, i.e. they consider the default supply FR of 10^{-5}. To compensate the supplies unreliability, (2) to (4) increase the coverage with which a supply failure is tolerated together with the hub error containment, i.e. both coverages simultaneously range from 90% to 100%. Taking this last aspect into account (2) to (4) reveal that, conversely to the bus, it is possible to improve the MT by enhancing the error-containment coverage provided by the hub. However, they show that to significantly improve the MT in such a way, it is manda-

tory to build up a highly-reliable hub. This could be achieved by internally replicating the hub, as it is almost unfeasible to achieve hub core FRs lower than 10^{-7} hour^{-1} by just investing in its quality. Another interesting result is that the MT in (3) and (4) approach the one of (1) as the hub core reliability, the hub error-containment and the fault-tolerance coverage of the supplies increase. This suggests that it is possible to reach a high BBW reliability by investing in these parameters without much concern on the power supplies quality. Finally, whether or not the MT potentially achievable by a star suffices would depend on the specific BBW application.

6. Conclusions and future work

This paper characterizes the reliability of fieldbus systems relying on simplex bus / star topologies when permanent hardware faults occur. It carries out parametric sensitivity analyses that quantify, for the first time, the system reliability that can be achieved with a bus and a star, depending on several dependability-related aspects. It builds upon the models previously proposed by the authors for the case of CAN and CANcentrate and, then, it extends them to further consider an issue of increasing concern in highly-reliable systems, i.e. the negative impact of power supplies unreliability.

This work provides important insights from both a theoretical and practical point of view. The actual influence on the reliability of the aspects studied here was still unknown. This paper clarifies this influence, refuting some intuitive ideas and revealing some unexpected effects. Results can be used as a practical guide to help system engineers in deciding when is it better to use a star than a bus, as well as in designing simplex bus and star fieldbus systems that are adequate in terms of reliability. In this sense, this work is not intended to demonstrate the superiority of stars over buses, but to identify the key aspects that affect the reliability that can be achieved with them.

Table 2 summarizes the main conclusions of the analyses. First, results highlight the need of providing the system with redundant power supplies. Second, they qualify the importance of the fault-tolerance and error-containment coverages. Third, they further advocate investing in the error-containment mechanisms at the star's central element. Fourth, they refute the intuitive idea that the star is not appropriate when the reliability of the components that are more abundant in it than in a bus is low; and identify the cases in which the bus outperforms the star. Finally, results reveal the huge influence of the hub reliability on the benefits of the star.

The conclusions can be somehow extrapolated to other fiedlbus technologies. Other fieldbuses use components with similar failure rates as those used in CAN. Moreover, although failure modes differ from one technology to another, the models and analyses presented here reflect what really matters of each type of fault, i.e. its proportion and the capacity of the topology for containing the errors the fault generates.

This work is being extended to tackle the issue of temporary faults, as well as the negative impact that external events such as collisions have on dependability.

7. Acknowledgements

This work was supported by project DPI2011-22992 (Spanish *Ministerio de Economía y Competividad*), by FEDER funding, and by the Portuguese government through FCT grant Serv-CPS PTDC/EEA-AUT/122362/2010.

Table 2: Sensitivity Analyses Main Conclusions

Analysed feature	Main conclusions
Power supply redundancy and coverage	(a) To replicate the power supply is fundamental for achieving an acceptable reliability in both the bus and the star. (b) The star is noticeably better than the bus only if the power supply is replicated. (c) When 1 power supply is required, the best redundancy option is to duplicate it, specially when the coverage with which the power supply is tolerated is imperfect. (d) The star can specially benefit from an increase in the coverage with which the power supply failure is tolerated.
System fault-tolerance coverage (tolerance to node failure/disconnection)	(a) In order to benefit from the star over the bus, the system has to provide a high-enough coverage ($> 97\%$ for 3 nodes and $> 89\%$ for 15). (b) The lower the number of nodes, the higher the coverage should be. (c) A coverage $> 99.9\%$ is not necessary. This value is lower than those typically attained by highly-reliable systems.
Hub's error-containment coverage (bit-flipping coverage)	(a) The minimum coverage the hub must provide is lower than such for the system fault-tolerance coverage. (b) The lower the number of nodes is, the higher the coverage should be. (c) A coverage $> 99\%$ is not necessary, except when the number of nodes and the proportion of disturbing faults is high.
Disturbing faults proportion (bit-flipping proportion)	(a) The higher the proportion, the higher the MT improvement the star yields. (b) The bus is very vulnerable to this proportion for any number of nodes. (c) The vulnerability of the star to this proportion increases with the number of nodes and, thus, the MT improvement the star yields when compared with the bus decelerates as this number increases. (d) It is not necessary to suffer from a huge proportion of disturbing faults to benefit from the star's better error containment.
Cabling failure rate (wires and connectors)	(a) It is refuted the intuitive idea that a star is inappropriate when the cabling is unreliable. (b) The higher the number of nodes, the higher the star benefits for almost any cabling failure rate. (c) The star shows sustained reliability without much concern on the cabling reliability. (d) The cabling reliability is crucial for the bus to be reliable enough. (e) In practice it is not possible to make the bus more reliable than the star by investing in the reliability of the bus cabling.
Transceiver failure rate	(a) For a low number of nodes, the bus can outperform the star if highly-reliable transceivers are used. (b) For a high number of nodes, the star is the best choice for any given transceiver failure rate. (c) To use reliable-enough transceivers is fundamental in both topologies.
Hub failure rate	(a) The reliability of the hub is of utmost importance. (b) For a low number of nodes, the system reliability can be improved by 103% in practice by decreasing the hub failure rate. (c) For a high number of nodes, the system reliability can be improved by 114% in practice by decreasing the hub failure rate. (d) It is possible to attain in practice the system reliability that would be achieved by using an ideal hub that does not fail.

References

[1] Munoz-Castaner, J., Asorey-Cacheda, R., Gil-Castineira, F., Gonzalez-Castano, F., Rodriguez-Hernandez, P.. A Review of Aeronautical Electronics and Its Parallelism With Automotive Electronics. IEEE Trans on Industrial Electronics 2011;58(7):3090–3100.

[2] Tuohy, S., Glavin, M., Hughes, C., Jones, E., Trivedi, M., Kilmartin, L.. Intra-Vehicle Networks: A Review. IEEE Transactions on Intelligent Transportation Systems 2014;pp(99):1–12.

[3] Abuteir, M., Obermaisser, R.. Mixed-criticality systems based on time-triggered ethernet with multiple ring topologies. In: Proceedings of the 9th IEEE International Symposium on Industrial Embedded Systems (SIES). Ieee; 2014, p. 170–178.

[4] Paulitsch, M., Hall, B.. Insights into the Sensitivity of the BRAIN (Braided Ring Availability Integrity Network) On Platform Robustness in Extended Operation. In: Conf. on Dependable Systems and Networks, 2007. DSN '07. 37th Annual IEEE/IFIP International. Edinburgh. ISBN 0769528554; 2007, p. 154 – 163.

[5] Amaral, M.C., Costa, C.H.A., Carvalho, T.C.M.B.. REASoN - REliability and / or Availability Evaluation for Sustainable Networking. In: 4th International Workshop on Reliable Networks Design and Modeling. 2012, p. 759–765.

[6] Kumar, S., Das, N., Member, S., Islam, S., Member, S.. Performance Analysis of Substation Automation Systems Architecture Based on IEC 61850. In: Power Engineering Conference (AUPEC). IEEE; 2014, p. 1–6.

[7] Padmavathy, N., Chaturvedi, S.K.. Evaluation of mobile ad hoc network reliability using propagation-based link reliability model. Reliability Engineering & System Safety 2013;115:1–9.

[8] Bistouni, F., Jahanshahi, M.. Analyzing the reliability of shuffle-exchange networks using reliability block diagrams. Reliability Engineering & System Safety 2014;132(0):97–106.

[9] Yeh, W.C., Bae, C., Huang, C.L.. A new cut-based algorithm for the multi-state flow network reliability problem. Reliability Engineering & System Safety 2015;136:1–7.

[10] Kumari, S., Ojha, A.. Maintainable Stochastic Flow Networks with High QoS: A Quick and Practical Approach. In: 4th IEEE International Conference on Advances in Computing and Communications. 2014, p. 260–265.

[11] Rosset, V., Souto, P.F., Portugal, P., Vasques, F.. Modeling the reliability of a group membership protocol for dual-scheduled time division multiple access networks. Computer Standards & Interfaces 2012;34(3):281–291.

[12] Lange, R., Vasques, F., Portugal, P., Oliveira, R.. Guaranteeing Real-Time Message Deadlines In The FlexRay Static Segment Using a On-line Scheduling Approach. In: 9th IEEE International Workshop on Factory Communication Systems (WFCS). 2012, p. 301–310.

[13] Dehbashi, M., Lari, V., Miremadi, S.G., Shokrollah-shirazi, M.. Fault Effects in FlexRay-Based Networks with Hybrid Topology. In: 3rd Int. Conf. on Availability, Reliability and Security. 2008, p. 491–496.

[14] Zimmermann, A., Geyer, F.. Towards Reliability Evaluation of AFDX Avionic Communication Systems With Rare-Event Simulation. In: 12th Probabilistic Safety Assessment & Management conference. 2014,.

[15] Aza-vallina, D., Denis, B., Faure, J.m.. Communications reliability analysis in networked embedded systems. In: European Conf. on Safety and Reliability - ESREL 2011. 2011,.

[16] Aza-vallina, D., Faure, J.m., Aza-vallina, D., An, J.m.F., Expression, A., The, O.F.. An analytic expression of the reliability of transmissions in fieldbuses with propagated failures. In: 4th IFAC Workshop on Dependable Control of Discrete Systems. Elsevier; 2013, p. 103–108.

[17] Barranco, M., Proenza, J., Almeida, L.. Quantitative Comparison of the Error-Containment Capabilities of a Bus and a Star Topology in CAN Networks. IEEE Trans on Industrial Electronics 2011;58(3):802–813.

[18] Barranco, M.. Improving error containment and reliability of communication subsystems based on Controller Area Network (CAN) by means of adequate star topologies. Ph.D. thesis; DMI, Universitat de les Illes Balears (UIB); 2010.

[19] Vidal-Idiarte, E., Maixe-Altes, J., Perez-Solorzano, B., Gil-Dolcet, E.. Controller area network fusing management system in a 14 V/42 V automotive electrical architecture. IET Power Electronics 2009;.

[20] Zeltwanger, H.. Controller Area Network — introduced 25 years ago. CAN Newsletter 2011;:18–20.

[21] Barranco, M., Proenza, J., Rodriguez-Navas, G., Almeida, L.. An active star topology for improving fault confinement in CAN networks. IEEE Trans on Industrial Informatics 2006;2(2):78–85.

[22] Navet, N., Song, Y., Simonot-Lion, F., Wilwert, C.. Trends in automotive communication systems. Proc of the IEEE 2005;93(6).

[23] ISO11898-1. Controller Area Network (CAN) - part 1: Data link layer and physical signalling. 2003.

[24] Reliability considerations for power supplies. Tech. Rep.; CUI inc 2013,; 2013. URL http://www.cui.com/catalog/resource/reliability-considerations.

[25] Relex, . Relex reliability prediction. 2006.

[26] DOD, . MIL-HDK-217f-2 Military Handbook, Reliability Prediction Of Electronic Equipment. 1995.

[27] Telcordia, . SR-332, issue 2, reliability prediction procedure for electronic equipment. 2006.

[28] Wu, N.. Reliability analysis for AFTI-F16 srfcs using ASSIST and SURE1. In: Proc. of the American Control Conf.; vol. 6. 2002, p. 4795–4800.

[29] Sanders, W.H.. Möbius Manual Version 2.4 Rev. 1; 2012.

[30] Barranco, M., Proenza, J., Almeida, L.. Reliability Improvement Achievable in CAN-based Systems by Means of the ReCANcentrate Replicated Star Topology. In: 8th IEEE International Workshop on Factory Communication Systems, Nancy, France. 2010

[31] Sanders, W.H., Malhis, L.M.. Dependability evaluation using composed SAN-based reward models. Journal of parallel and distributed computing 1992;15(3):238–254.

[32] Hamby, D.M.. A review of techniques for parameter sensitivity analysis of environmental models. Environmental Monitoring and Assessment 1994;32(2):135–154.

[33] Morris, J., Koopman, P.. Representing design tradeoffs in safety-critical systems. In: Proc. WADS, St. Louis, MO. 2005, p. 1–5.

[34] Braun, C., Havet, L., Navet, N.. Netcarbench: a benchmark for techniques and tools used in the design of automotive communication systems. In: Proc. of the 7th IFAC International Conf. on Fieldbuses and Networks in Industrial and Embedded Systems (FeT 2007). Toulouse, France; 2007, p. 321–328.

[35] Data Sheet dsPIC30F4011/4012. Tech. Rep.; Microchip Technology Inc.; 2010.

[36] Data sheet PCA82C250. Tech. Rep.; Philips Semiconductors; 2000.

[37] Wang, Z., Yu, L., You, C., Wang, Y., Song, J.. Fail-safe control allocation for a distributed brake-by-wire system considering the driver's behaviour. Journal of Automobile Engineering 2014;228(13):1547–1567.

[38] Freddi, A., Longhi, S., Monteriù, A., Prist, M.. Actuator Fault Detection and Isolation System for an Hexacopter. In: IEEE/ASME 10th Int. Conf. on Mechatronic and Embedded Systems and Applications (MESA). 2014,.

[39] Ahmadzadeh, S.R., Leonetti, M., Carrera, A., Carreras, M., Kormushev, P., Caldwell, D.G.. Online discovery of AUV control policies to overcome thruster failures. In: 2014 IEEE International Conference on Robotics and Automation (ICRA). IEEE; 2014, p. 6522–6528.

[40] Fuchs, C.M., Schneele, S., Klein, A.. The Evolution of Avionics Networks From ARINC 429 to AFDX. In: Proceedings of the Seminars Future Internet (FI), Innovative Internet Technologies and Mobile Communication (IITM) and Aerospace Networks (AN). 2012, p. 65–76.

[41] Sakurai, K.. An Autonomous Decentralized Architecture with Agreement Protocols for Safety-Critical Embedded Distributed Control Systems. Ph.D. thesis; Osaka University; 2014.

[42] Frede, D., Khodabakhshian, M., Malmquist, D.. A state-of-the-art survey on vehicular mechatronics focusing on by-wire systems. Tech. Rep.; Department of Machine Design, KTH Royal Institute of Technology; Stockholm; 2010.

[43] Tibor, K., Gianone, L.. Automotive communication protocols focused on the x-by-wire applications. In: A JÖVÖ JÁRMÜVE 01/02. 2011, p. 46–49.

[44] Cheon, J.S., Kim, J., Jeon, J.. New Brake By Wire Concept with Mechanical Backup. SAE Int Journal on Passeng Cars - Mech Syst 2012;5(4):1194–1198.

[45] Steinbach, T., Korf, F., Schmidt, T.C.. Real-time Ethernet for automotive applications: A solution for future in-car networks. In: 2011 IEEE International Conference on Consumer Electronics (ICCE). 2011, p. 216–220.