First results of the assessment of the improvement of error containment achieved by CANcentrate

Manuel Barranco, Julián Proenza
Dpt. Matemàtiques i Informàtica
Universitat de les Illes Balears, Spain
manuel.barranco@uib.es, julian.proenza@uib.es

Luís Almeida
DET/IEETA
Universidade de Aveiro, Portugal
lda@det.ua.pt

Abstract

Although the Controller Area Network (CAN) is widely used as the communication subsystem of many distributed control systems, it presents dependability limitations that are inherent to its bus topology. In particular, it lacks of the tailored mechanisms to avoid that the errors generated by a single fault jeopardizes the communication capabilities of many nodes, possibly causing a general failure of the system. In order to overcome this limitation, we developed an innovative star topology, called CANcentrate, whose hub includes advanced error-containment mechanisms. This paper explains how to model both CAN and CANcentrate by means of stochastic Petri Nets, in order to assess the error-containment improvement achieved by CANcentrate over CAN, and further shows some first results.

1 Introduction

Controller Area Network (CAN) fulfills the communication requirements of many distributed control systems. In particular, CAN includes an event-triggered data link layer that provides high reliability and good real-time performance with very low cost.

The requirements that must be met to consider that a communication subsystem is dependable are application dependent. Many applications require a communication infrastructure, in which a minimum number of nodes can communicate with each other throughout a complete interval of time. For instance, in a factory plant it is required that a fault in any of its components jeopardizes the communication capabilities of the less number of nodes as possible. Moreover, other applications can simply accept that it exists a minimum number of nodes that can still communicate. For example, in the intra-building communication network of an hotel, the main objective is to provide service to the maximum number of rooms, even when faults occur.

In the context of these applications the use of CAN is controversial due to dependability limitations. One such limitation arises from its bus topology, which lacks of adequate error-containment mechanisms for preventing the propagation of errors. Consequently, a bus topology includes multiple points such that a single fault of any of them can make it impossible for more than one node to communicate with any node of the system, possibly causing a general failure. We refer to these points as points of severe failure.

In order to eliminate the existence of multiple of such points, we have developed a new communication infrastructure called CANcentrate [1] that relies on a star topology. CANcentrate incorporates an active hub provided with enhanced fault-treatment capabilities that prevent the errors from one part of the star from propagating to other parts. In this way, it reduces the multiple points of severe failure of a bus topology to a unique single point of failure: the hub.

The key issue of CANcentrate is that its hub enforces the necessary error containment to ensure that a given fault prevents a maximum of one node from communicating with the rest of nodes. In this way, CANcentrate would improve dependability of any system, specially those that include tailored mechanisms to tolerate (or accept) that some nodes cannot communicate (e.g. by means of node replication)

The aim of the present work is to model different CAN bus networks and CANcentrate in order to mathematically assess the improvement of error containment achieved when using CANcentrate. For this purpose, we compare the probability that a failure does not invalidate the communication capabilities of more than N nodes.

In a first phase, we are using a value of $N = 1$, so that we measure the probability of not suffering a severe failure. In this way, we first study the potential benefits of CANcentrate for systems that can tolerate (or accept) that at most one node cannot communicate. In future papers, we will address greater values of N in order to asses the benefits of CANcentrate for applications that can tolerate or accept that more than one node cannot communicate.

2 CANcentrate basics

Due to the lack of adequate error-containment mechanisms, faults at different points on a bus may generate errors that propagate along the communication subsystem,

 doi:10.1109/WFCS.2006.1704127

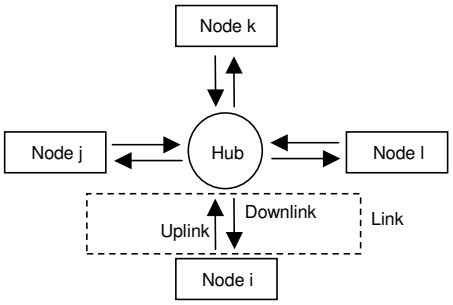

Figure 1. Architecture of CANcentrate

i.e. a bus has many points of severe failure. We have focused our efforts in preventing faults related to the Physical Layer of CAN from causing a severe failure of communication. These faults are:

- Stuck-at-recessive and stuck-at-dominant faults, which can occur within nodes or in the medium.
- Medium partition fault that occurs whenever the network is physically broken into several subnetworks called network partitions.
- Bit-flipping fault that occurs whenever a network component, either node or medium, exhibits a fail uncontrolled behavior, sending random erroneous bits with no restrictions in value or time domains.

In order to tolerate these faults in CAN, several approaches have been proposed. The benefits of CANcentrate over these solutions are thoroughly described in [1].

In CANcentrate, each node is connected to the hub by a dedicated link that contains an uplink and a downlink (Figure 1). The hub receives each node contribution through the corresponding uplink, couples all non-faulty contributions with a logical AND function, and broadcasts the resultant coupled signal through the downlinks.

The use of an uplink and a downlink for each node allows separating the contribution of each node from the coupled signal, so that the hub can monitor each node contribution separately and detect faulty transmissions. This feature allows the hub to diagnose the location of faults with more precision than the typical error counters of CAN [2]. Permanently faulty contributions are disabled, and so not propagated to the coupled signal, thus being confined to the port of origin.

Since CANcentrate is fully compliant with CAN, it keeps all CAN dependability properties and can be built using CAN COTS components.

3 Modelling

The dependability properties of a communication subsystem are determined by the following aspects: the components that are considered to constitute it, their respective failure rates, how they are interconnected, as well as by the way in which they interact to provide fault treatment. Therefore, the error containment of a system can be assessed by modelling these aspects.

To carry out our models, we are using Stochastic Activity Networks (SANs), which are stochastic extension to Petri Nets [3]. A SAN basically includes tokens, places, activities, input gates and output gates. The number of tokens located in each place, i.e. the marking of the places, determines the state of the modelled system. Activities are used to change the marking of the places, thereby modelling the transitions of the system through different states. Input and output gates are used to define enabling and completion rules for activities.

Each component of the system is modelled by means of a specific SANs submodel whose marking indicates whether the component is faulty or not. In order to measure the probability of the overall system suffering a severe failure, two aspects are taken into account: the probabilities of the markings indicating faulty situations of components, as well as how components interact to provide fault treatment. Such strategy is very similar to the one proposed in [4].

The tool we have used to built our models is Moebius [3]. Moebius incorporates a simulator and several numerical solvers to calculate the measures of interest. All models we have carried out were analytically solved.

3.1 Components of CAN and CANcentrate

We have considered that both CAN and CANcentrate are constituted by the following types of components: micro controllers, CAN controllers, transceivers, segments of cable (UTP-CAT5), connectors (9-pin DSUB), and an FPGA (in the case of the hub)

In order to consider realistic failure rates of these components, we have used a software of prediction of failure rates, called Relex [5], taking into account the MIL-HDBK-217F [6] standard, as well as the *AT&T Reliability Manual* [7].

The calculus of the major part of these rates required to specify parameters of the components. For them, we have assumed not optimistic values, p.e. category $\rightarrow$ integrated circuit, technology type $\rightarrow$ MOS, quality level $\rightarrow$ commercial, package type $\rightarrow$ hermetic, years in production ≤ 2. In the case of the hub and each CAN controller, it was also necessary to specify their number of logic gates. For that we used real values obtained from synthesized implementations of both types of components [8].

3.2 Model and fault assumptions

In the first phase of our work we are taking into account the following fault assumptions.

Firstly, we consider that each component can independently fail. The *Mean Time to Failure* (MTTF) of each of them is considered to be exponentially distributed with mean $1/\lambda$, where λ is the failure rate expressed in number of failures per hour. Secondly, we assumed that the MTTF distribution is Non-Defective [4], which implies that the

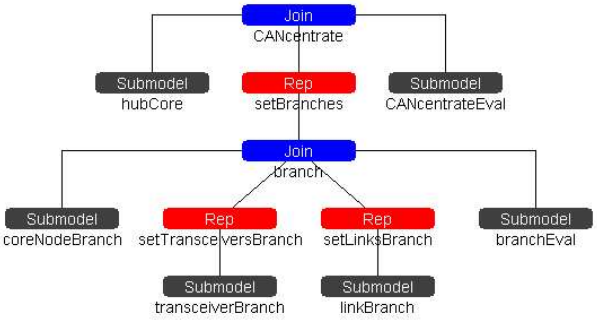

Figure 2. Hierarchical model of CANcentrate

probability of failure at instant of time t is 0 when $t = 0$, $1-exp(-\lambda * t)$ when $t = X$, and 1 when $t = \infty$. Thirdly, we consider that when a component fails, it can exhibit any of the following faults: a stuck-at-recessive, stuck-at-dominant or a bit-flipping fault. In addition, cables and connectors can also suffer a physical disruption, which may provoke a network partition. Fourthly, all faults that a component may suffer are equiprobable and permanent.

Fifthly, we assumed that the total length of the CAN bus, as well as the distance between every pair of nodes of CANcentrate are of 70 meters (which is the maximum star diameter achieved in our experiments at 690kbit/sec [8]) This is a pessimistic assumption over CANcentrate, since it represent the worst layout of a star topology for interconnecting an ensemble of nodes in which the two farthest nodes are separated 70 meters. Sixthly, among several possible ways to attach the nodes to the bus medium by means of different types of connectors, we have chosen the most optimistic one for the CAN bus, which includes no stubs and the less number of connectors.

Finally, we made two assumptions regarding the hub. On the one hand, we considered that when a fault occurs in a part of the hub different from its transceivers, it stops providing interconnection, thereby provoking the fault of the overall communication subsystem. This is a pessimistic assumption, since the hub could suffer a more benign fault, e.g. it could stop performing fault confinement or it could unfairly isolate a correct port. On the other hand, we consider that the error detection and the fault confinement coverage provided by the hub are of the 100 per cent. Our experimental results [8] indicate that this assumption is not unrealistic.

3.3 Modelling strategy

As said before, each component is modelled by means of a dedicated submodel whose state indicates whether the component is faulty or not. In order to model the influence of the interconnection of the components on the error-containment capacities of the overall system, the whole model is built as a hierarchical composition of different types of submodels. This approach is similar to that one proposed in [9].

Figure 2 shows the hierarchical model of CANcentrate as an example. As can be seen in such Figure, the submodels are labelled either as *Submodel*, *Rep* and *Join*.

A submodel of the type *Submodel* models whether a given component or a set of components are faulty or not, as well as which kind of fault present. The *Submodel* that models a given component basically consists of the following elements. First, one place that is initially marked with one token, indicating that the component is not faulty. Second, a set of places related to different types of faults. A token in one of these places indicates that the component suffers a specific type of fault. Third, a set of activities where the firing of each one of them has an associated *Mean Time to Failure* and is aimed at modelling the occurrence of a given fault. When a fault occurs, the corresponding activity erases the token from the initial state and sets a specific marking at the adequate place.

In contrast, the structure of a submodel of type *Submodel* that models a set of components is more complicated. It is basically aimed at evaluating the type of fault that the set of components presents as a whole, taking into account the kind of faults suffered by its constituent components. For instance, submodel *CANcentrateEval* takes into account the faults the hub and the different branches present, and decides if a severe failure has occurred.

A *Rep* submodel models a set of components of the same type. It is built as the replication of a submodel of type *Submodel*. For instance, *setLinksBranch* in Figure 2 models the two links that attach a node to the hub. It is built replicating two times the submodel *linkBranch*, which models either the uplink and the downlink.

Finally, a *Join* submodel models a part of the network that is constituted by a set of components of different type that are interconnected. In addition, a *Join* has an associated *Submodel* that evaluates whether the set of components is faulty or not and the type of fault it presents. For example, submodel *branch* in Figure 2 represents a given hub branch. It joins submodels *coreNodeBranch*, *setTransceiverBranch*, *setLinksBranch* and *branchEval*. The first three submodels respectively model the CAN controller, four transceivers, and the uplink and downlink. In contrast, *branchEval* evaluates the way in which the branch fail as a whole.

4 First error-containment results

One of the first experiments we carried out is aimed at comparing the error-containment capacities of CAN and CANcentrate by measuring the probability of not suffering a severe failure in both. In this way, we could assess the dependability improvement achieved by CANcentrate for systems that can tolerate or accept that at most one node cannot communicate.

In particular, we assessed the evolution of the probability of not suffering a severe failure throughout the time, taking into account different number of nodes. We considered as representative enough to measure the probability

of not suffering a severe failure for: 8, 16 and 24 nodes.

As stated above, we used realistic failure rates of the components to obtain values neither optimistic nor pessimistic. These failure rates expressed in faults per hour are depicted in Table 1.

Component	Failure rate
Micro controller	9,65432E-7
CAN controller	4,14319E-7
Transceiver	2,57493E-8
Cable (per meter)	1E-10
Connector	1,8E-9
Hub with 8 ports	2,96078E-7
Hub with 16 ports	5,26466E-7
Hub with 24 ports	5,96267E-7

Table 1. Failure rates

The results of this experiment are depicted in Figure 3. As can be seen, for a given number of nodes, CANcentrate is clearly better than CAN throughout all the time.

Furthermore, CANcentrate is better than CAN during significative intervals of time, even when CANcentrate is provided with more nodes than CAN. For instance, the probability of not suffering a severe failure when using CANcentrate with 16 nodes is higher than when using CAN with 8 nodes during more than 4000h. Moreover, in a CANcentrate network with 24 nodes, such probability is higher than in a CAN network of 16 nodes during more than 5000h, and higher than in CAN network with 8 nodes during more than 1000h.

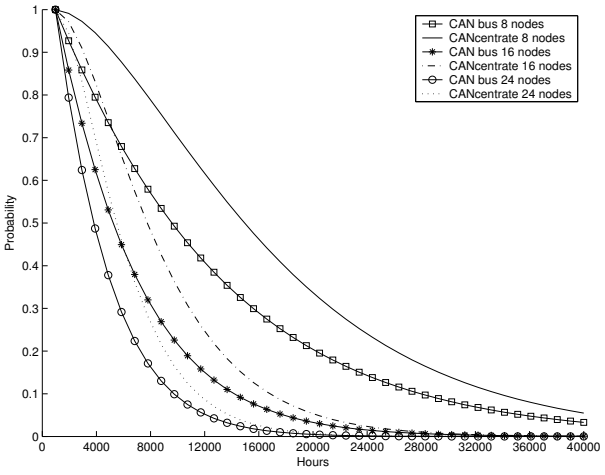

Figure 3. Evolution of probability of not suffering a severe failure

Similarly, if we focus on the interval of time during which the probability of not suffering a severe failure is very high (≥ 0.9999), we obtain the following results for 8, 16 and 24 nodes: 10-15h, 5-10h, $<$ 5h in the case of CAN; and 315h, 170h and 145h in the case of CANcentrate. The interval of time throughout the CAN bus provides high dependability does not exceed the 15 hours, even in the case in which the system is constituted by 8 nodes only. In contrast, CANcentrate is high dependable during 145 hours even when it is provided with 24 nodes.

5 Conclusions and future work

In this paper we propose to compare the dependability properties of CAN and CANcentrate by means of measuring their error-containment capacities. For this purpose, we model both communication subsystems by means of stochastic Petri Nets, and evaluate which is their probability of not suffering a severe failure.

The first results show that, for a given number of nodes, CANcentrate is clearly more resilience to faults than a CAN bus throughout the time. Moreover, even when considering a CANcentrate network with more nodes than a CAN bus, CANcentrate is better than CAN during some thousands hours.

In future work, we will study the impact of the fault-coverage percentage provided by the hub on the error-containment improvement achieved by CANcentrate, as well as the relationship between such improvement and different failure rates of the hub and the cable. Furthermore, we will study the benefits of CANcentrate over CAN when the faults that components may suffer are not equiprobable, as well as when these faults are not permanent. Finally, we will compare the error-containment capacity of CANcentrate and other architectures, such as replicated transmission media and bus guardians.

References

[1] M. Barranco, G. Rodríguez-Navas, J. Proenza, and L. Almeida, "CANcentrate: An active star topology for CAN networks", *WFCS'04. IEEE Workshop on Factory Communication Systems, Vienna, Austria*, 2004.

[2] ISO, "ISO11898. Road vehicles - Interchange of digital information - Controller area network (CAN) for high-speed communication", 1993.

[3] W. Sanders and T. B. of Trustees, "Moebius User Manual Version 1.6.0", 2004.

[4] M. Mahotra and K. S. Trivedi, "Dependability Modeling Using Petri-Nets", *IEEE Transactions on Reliability*, vol. 44, no. 3, September 1995.

[5] R. Corporation, "Relex Reliability Software", http://www.relex.com, 2006.

[6] DOD, *MIL-HDK-217f Military Handbook, Reliability Prediction Of Electronic Equipment*, Department of Defense Washington DC, 1991.

[7] D. J. Klinger, Y. Nakada, and M. A. Menendez, *ATT Reliability Manual*, Van Nostrand Reinhold, 1990.

[8] M. Barranco, J. Proenza, G. Rodríguez-Navas, and L. Almeida, "A CAN hub with Improved Error Detection and Isolation", *10th International CAN Conference*, March 2005.

[9] P. Portugal and A. da Silva, "A Framework for Dependability Evaluation of Fieldbus Networks", *WFCS'04. IEEE Workshop on Factory Communication Systems, Vienna, Austria*, 2004.