On the (non-intrusive) observability of the CAN FD protocol

João de Sousa Alves and José Rufino

LaSIGE, Faculdade de Ciências, Universidade de Lisboa, Portugal,
fc41937@alunos.fc.ul.pt, jmrufino@ciencias.ulisboa.pt *

Abstract. The Controller Area Network (CAN) protocol has been extensively used in many application domains, including industrial control, appliances, medical, and transportation. The last sector includes manned and unmanned vehicles in land, maritime and aerospace.
The definition of the CAN with Flexible Data rate (CAN FD) specification, currently under normalisation, enhances the original CAN protocol in two ways: it extends from 8 to 64 bytes the maximum payload size of a data frame; it enhances the bus signalling rate, while maintaining the determinism of node network access arbitration.
The CAN FD data frame includes in its header an *Error State Indicator* (ESI) flag. Thus, one fundamental question is whether or not the ESI flag will be useful for building highly reliable distributed real-time embedded systems based on the CAN FD protocol?
This communication formulates the problem in terms of provisioning node failure detection and membership services for CAN FD systems.

Keywords: Real-time and embedded systems; Real-time reliable communications; Controller Area Network; Dependability and adaptability; Non-intrusive system observability.

1 Introduction and Motivation

The Controller Area Network (CAN) [4, 8] is a widely used simple and robust protocol. Despite the large application base, the original CAN protocol specification exhibited two main shortcomings: a data frame payload size limited to a maximum of 8 bytes; a low data signalling rate on the bus, intended to allow a quasi-stationary operation of the network which is exploited to implement a deterministic node network access arbitration scheme.

The CAN with Flexible Data rate protocol [9] aims to overcome those limitations: allowing message encapsulation with payloads up to 64 bytes and allowing nodes to switch to a faster bit signalling rate after network access, through the

* This work was partially supported by FCT, through funding of LaSIGE Research Unit, ref. UID/CEC/00408/2013, and FCT/CAMPUS FRANCE (PHC PESSOA), through the transnational cooperation project NORTH. This work integrates the activities of COST Action IC1402 - Runtime Verification beyond Monitoring (ARVI), supported by COST (European Cooperation in Science and Technology).

original CAN deterministic node/message arbitration scheme, has been decided. In the redefinition of frame's format, CAN FD includes in its data frame header an *Error State Indicator* (ESI) flag, transmitted dominant by *error active* nodes and recessive by a node in the *error passive* state. The ESI flag does not exist in the original CAN protocol [8, 9].

Reasoning on the utility of the ESI flag, one first observation is that it allows an higher observability of CAN FD based systems. Moreover, this enhanced observability is (almost) non-intrusive, since it requires only a bit (out of many) per transmitted CAN FD data frame. Non-intrusive observation and runtime verification of CAN based systems have been addressed recently in [6, 5].

On the other hand, the CAN protocol has been used for building highly reliable distributed real-time embedded systems [11], through the definition and design of useful constructs such as bus media redundancy [12], fault-tolerant broadcast communication [14], clock synchronization [10], node failure detection and membership [13]. These works need to be revisited and reformulated in the context of the CAN FD protocol. This communication goes towards that goal, reasoning on the utility of the CAN FD ESI flag in terms of provisioning node failure detection and membership services for CAN FD based systems.

2 Controller Area Network

CAN is a multi-master network that uses a twisted pair cable as transmission medium [4, 8, 3]. The network maximum length depends on the data rate. The maximum values are: 40m @ 1 Mbps [3]. A single bit can be transmitted in the bus at a time. The signalling of a bit in the bus takes one out of two values: *recessive*, also the state of an idle bus; *dominant*, which always overwrites a recessive value. This behaviour, together with the use of unique frame identifiers, is exploited for bus arbitration. A *carrier sense multi-access with deterministic collision resolution* policy is used [4, 8]. If several nodes compete for bus access, the node transmitting the frame with the lowest identifier gets the bus.

CAN FD: CAN with Flexible Data rate

The CAN FD protocol is allowed to mix the nominal quasi-stationary operation with non-stationary data transfers, where a node has permission to partially transmit a data frame at a secondary ("higher") bit signalling rate. During this non-stationary period: a single node may be transmitting at a time; several bits may be travelling in the bus at the same time. CAN FD [9] enables data rate switching at a pre-determined control bit, the Bit Rate Switch (BRS) bit of the data frame structure illustrated in Figure 1. Therefore, the CAN FD BRS bit establishes a boundary separating:

- *arbitration-phase* - which uses the "normal" bit signalling rate, allowing the CAN deterministic arbitration scheme to operate properly;
- *data-phase* - enabling the use of a "higher" rate for bit signalling.

A third phase, performed at the normal bit signalling rate, intended for frame acknowledgement, terminates the CAN FD data frame transmission.

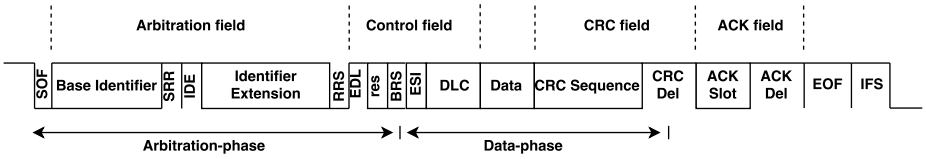

Fig. 1. CAN FD data frame structure and format.

Frame Formats

A *data frame* is a piece of encapsulated information disseminated on the network, which contains as payload a *message*, a user-level piece of information. A *remote frame* has no payload. Accordingly with the CAN specification several nodes may simultaneously transmit the "same" remote frame[1]. Since one cannot have more than one transmitting node in CAN FD, no remote frame formats are defined for this protocol [9]. Protocols above CAN FD must resort to CAN remote frames if their use is required, as it happens in [14, 13].

Fault Confinement

Fault confinement mechanisms are based on two (transmit/receive) error counters, with each error causing an increment larger than the decrement resulting from a successful data/remote frame transfer. With a very high probability, the node causing the error will experience the highest error count increase.

Upon reaching a given error threshold a node enters into the error passive state, where: it is allowed to transmit and receive frames; after transmitting a data or remote frame, the node is obliged to an additional wait period before it can start a new transmission; only succeeds to signal errors while transmitting.

Otherwise, the node is in the error active state, the normal operating mode, or it is bus off, where it does not participate in any bus activity, being unable to send or receive frames.

3 Node Failure Detection and Membership

Informally, a membership service provides consistent information, at any given time, about "who is present" and "who is absent" in a distributed system. Though this may generically refer either to processes or to processors (nodes), this communication is specially concerned with site/node membership. Namely, in the context of the CAN FD protocol, we aim to revisit, re-analyse and reformulate the node failure detection/site membership protocols described in [13].

As a general rule, each CAN or CAN FD node may signal its active state through the broadcast of a life-sign message, which works as an heartbeat. To save network bandwidth (a scarce resource in CAN) normal traffic is implicitly

[1] Meaning, nodes must specify the same identifier and Data Length Code (DLC) field.

used in [13] as a life-sign. The use of CAN FD ESI flag allows to extract, (almost) with no overhead, a sign that a node is correct or has been severely affected by errors. A node signalling an error passive state through the ESI flag is in fact sending a *dying-sign*. This can trigger a combination of strategies such as: placing the node in quarantine; early declaration of the node's crash failure; apply failure prediction techniques in order to forecast the node's failure/recovery [1].

Node crash failures still need to be detected, but the lack of an heartbeat from a given node will (non-intrusively) trigger hardware-based timing failure detectors, as proposed in [7, 2]. Dissemination of this failure notification to all active nodes should resort to CAN remote frames, even in CAN FD systems.

Exploitation of CAN FD data frame's characteristics (increased payload and lower message deliver latency) will have a positive impact in the execution of [13] internal *reception history* and *site membership view* agreement protocols.

References

1. Almeida, K., Pinto, R.C., Rufino, J.: Fault detection in time- and space-partitioned systems. Communication at the 5th Simpósio de Informática (INFORUM), Évora, Portugal, (Sep 2013)
2. Casimiro, A., Gouveia, I., Rufino, J.: Enforcing timeliness and safety in mission-critical systems. In: Proceedings of the 22nd International Conference on Reliable Software Technologies (Ada-Europe). Vienna, Austria (Jun 2017)
3. CiA - CAN in Automation: CAN Physical Layer for Industrial Applications - CiA Draft Standard 102 Version 2.0 (Apr 1994)
4. ISO: International Standard 11898 - Road vehicles - Interchange of digital information - Controller Area Network for high-speed communication (Nov 1993)
5. Kane, A., Chowdhury, O., Datta, A., Koopman, P.: A case study on runtime monitoring of an autonomous research vehicle (ARV) system. In: Proc. 15th Int. Conf. on Runtime Verification. pp. 102–117. LNCS, Springer, Vienna, Austria (Sep 2015)
6. Kane, A.: Runtime Monitoring for Safety-Critical Embedded Systems. Ph.D. thesis, Carnegie Mellon University, USA (Feb 2015)
7. Pinto, R.C., Rufino, J.: Towards non-invasive run-time verification of real-time systems. In: Proc. 26th Euromicro Conf. on Real-Time Systems - WIP Session. pp. 25–28. Euromicro, Madrid, Spain (Jul 2014)
8. Robert Bosch GmbH: CAN Specification Version 2.0 (Sep 1991)
9. Robert Bosch GmbH: CAN with Flexible Data-Rate Version 1.0 (Apr 2012)
10. Rodrigues, L., Guimarães, M., Rufino, J.: Fault-tolerant clock syncronization in CAN. In: Proc. 19th Real-Time Systems Symposium. IEEE, Spain (Dec 1998)
11. Rufino, J.: Computational System for Real-Time Distributed Control. Ph.D. thesis, Technical University of Lisbon, Inst. Superior Técnico, Lisboa, Portugal (Jul 2002)
12. Rufino, J., Verissimo, P., Arroz, G.: A Columbus' egg idea for CAN media redundancy. In: Digest of Papers, The 29th International Symposium on Fault-Tolerant Computing Systems. IEEE, Madison, Wisconsin - USA (Jun 1999)
13. Rufino, J., Verissimo, P., Arroz, G.: Node failure detection and membership in CANELy. In: Proc. of the 2003 International Conference on Dependable Systems and Networks. pp. 331–340. IEEE, San Francisco, California, USA (Jun 2003)
14. Rufino, J., Verissimo, P., Arroz, G., Almeida, C., Rodrigues, L.: Fault-tolerant broadcasts in CAN. In: Digest of Papers, The 28th Int. Symposium on Fault-Tolerant Computing Systems. pp. 150–159. IEEE, Munich, Germany (Jun 1998)