Enforcing Fail-Silence in the Entire FlexRay Communication Cycle

Kai Wang
Shenyang Institute of Automation of the Chinese Academy of Sciences and Graduate School of the Chinese Academy of Sciences

Hong Wang and Aidong Xu
Shenyang Institute of Automation of the Chinese Academy of Sciences

ABSTRACT

It is desirable that the nodes exhibit fail-silent behavior when designing distributed fault-tolerant systems. Bus Guardian (BG) is adopted to enforce the fail silence property of the node in the FlexRay communication system. However, the BG is effective only in the static segment while in the dynamic segment it does not provide any protection. Therefore, a novel BG is proposed in this paper. The mechanisms of the novel BG are presented along with the node architecture necessary for implementing the presented technique. Mathematical analysis results show that the novel BG can effectively enforce fail silence property in the entire FlexRay communication cycle.

INTRODUCTION

FlexRay is a communication system that will support the needs of future in-car control applications[1]. These sophisticated future vehicle applications like driver assistance or autonomous driving support need computerized control of the driving dynamics. Drivers requests have to be translated into optimum steering, braking, and acceleration manoeuvres while taking into account the current driving conditions and environmental influences. This will greatly increase overall vehicle safety by liberating the driver from routine tasks and assisting the driver to find solutions in critical situations[2]. These future automotive applications, which include safety critical functionality, will demand new features for the communication subsystem. It has to provide reliable and predictable communication service to guarantee the information exchange between the components. And fast error detection in time (babbling idiot avoidance) is one of the primary requirements the communication subsystem has to fulfill[3].

A faulty node that monopolises the common channel by sending messages at erroneous points in time is called a babbling idiot. The babbling idiot disrupts the communication between all properly operating nodes and can thus cause a complete system failure[4]. It is therefore important to avoid babbling idiot failure in safety critical real time system. The cause of babbling idiot failure is that the
nodes of the system fail in an uncontrolled way. So, it is advisable to require that a node should exhibit only simple failure modes. In the ideal case a node has only a single, fail-silent failure mode[5]. In general a node is considered to be fail-silent if it produces either correct results or no results at all [6]. Fail-silent behaviour assures that an error within one node cannot propagate to other nodes within the system so that each node can be regarded as a separate fault containment region. Using fail-silent nodes greatly reduces the complexity of designing distributed fault-tolerant systems[5, 7]. Therefore, this property is appropriate to the automotive industry, which has to satisfy the requirement of low cost in mass production.

Fail-silence is traditionally enforced by adding redundancy within the node or by introducing behavioral error detection mechanisms[5]. Methods based on replication for achieving fail-silence suffer from the high overheads caused by replicating the subsystems and implementing a comparator (or voter) for reaching agreement. Due to cost-constraints such techniques are not possible to use in mass produced systems, such as automotive. Adding an extensive set of error detection mechanisms to each node is an alternative approach for enforcing fail-silence. The major advantage of using error detection mechanisms is the low cost associated with the mechanisms compared to replicating the processing systems within the node. Bus guardian (BG) is based on this method and has been adopted widely to improve the reliability of communication in both Time-triggered system and Event-triggered system.

BG is incorporated in Physical Layer to enforce the fail silence property of the node in the FlexRay communication system. The BG enables transmission according to its configuration parameters (set of numbers of active static slots) in the static segment, while it enables transmission for the entire duration of the dynamic segment without any limitation. Thus the BG does not provide any protection in the dynamic segment. To enforce fail-silent behavior of the node in the entire FlexRay communication cycle, a novel BG is proposed. The mechanisms of the novel BG are presented along with the node architecture necessary for implementing the presented technique. The BG is working as the FlexRay Specification describes in the static segment. While as far as the dynamic segment is concerned, the BG is designed according to the dynamic mini-slotting based scheme which is used to arbitrate transmissions in the dynamic segment. The BG provides protection based on Semi Scheduling Table specific to each node and a key parameter named Max_Write_Time. The novel BG has been verified by mathematical analysis. Analysis results shows that although the novel BG can not detect all babbling idiots due to the essential characteristics of the mini-slot based scheme, fail silence can be enforced outside of the allotted slots while in the slots allotted to the node the maximum effect of undetected faults is bounded. Therefore, the novel BG can effectively enforce fail silence property in the entire cycle. And at the same time, the real time performance of message transmission in the FlexRay dynamic segment can also be guaranteed by the BG.

In the following section, the design principles of technique in time-triggered( TT) and Event-triggered( ET) communication system are illustrated. In Section 3 we outline the schemes of FlexRay media access control at first. And then the BG Schedule described in FlexRay-BG Specification is analyzed and the existing problems are illustrated. In
section 4 we present the design of a novel BG to enforce fail-silent behavior of the node in the entire FlexRay communication cycle. The structure of the novel bus guardian is presented along with the principle of operation. Section 5 gives the performance analysis of the novel BG. Section 6 concludes the paper.

DESIGN OF BG IN TT AND ET COMMUNICATION SYSTEM

A bus guardian is a device designed to protect a bus from failure of some component attached to the bus. The concept of a bus guardian to isolate a node in the event of failure is not new, it has been used at least as far back as the FTMP architecture in 1978 to safely interface nodes to a multiple-redundant bus[10]. It is common for a bus guardian to either disconnect a node from the bus, or to shutdown the node, in order to prevent the faulty node from interfering with the correct operation of the other nodes in the system[8].

Time-triggered bus protocols such as TTP/C [11] and FlexRay[12], dictate that a node is only allowed to write to the bus at predefined times. Therefore it is relatively easy to detect if a node becomes a babbling node by watching when it writes to the bus. Furthermore, it is possible to build a bus guardian which only connects the node to the bus at these predefined times, hence ensuring that the node can never write to the bus when it should not do so.

With event-triggered communication, it is not possible to exactly specify when a message will be sent. This flexibility is an advantage in many systems, but it makes babbling idiot detection (and hence the design of a bus guardian) significantly more difficult[8]. Usually, it is designed based on probability theory. A few methods have been proposed.

I.Broster and A.Burns gives an analysable BG for Event-Triggered Communication[8]. The BG proposed cannot detect all babbling idiots, but the maximum effect of undetected faults is bounded and small. Giuseppe Buja etc. design another BG based on the FlexCAN architecture[9]. The BG is proposed to cope with the software babbling-idiot faults in the FlexCAN environment. Laboratory test shows that the babbling-idiot fault is not removed but masked by the BG and the system works properly.

MEDIA ACCESS CONTROL AND BG SCHEDULE OF FLEXRAY PROTOCOL

FlexRay is a communication system that will support the needs of future in-car control applications. At the core of the FlexRay system is the FlexRay communications protocol. The protocol provides flexibility and determinism by combining a scalable static and dynamic message transmission, incorporating the advantages of familiar synchronous and asynchronous protocols[1]. A physical layer incorporating an independent BG provides further support for error containment. And the BG Schedule is described in FlexRay-BG Specification.

MEDIA ACCESS CONTROL OF FLEXRAY PROTOCOL

In the FlexRay protocol, media access control is based on a recurring communication cycle. A communication cycle contains the static segment, the dynamic segment, the symbol window and the network idle time as shown in Figure 1[12]. Within one communication cycle FlexRay offers the choice of two media access schemes. These are a static time division multiple access (TDMA) scheme,
and a dynamic mini-slotted based scheme. Within the static segment a static time division multiple access scheme is applied to coordinate transmissions as shown in Fig 2. Within the dynamic segment a dynamic mini-slotted based scheme is used to arbitrate transmissions as shown in Figure 3. The dynamic mini-slotted based scheme works in the following manner. Each node maintain a mini-slot counter. At the beginning of dynamic segment, mini-slot counter value in each node is m. All nodes start to check to see if any node is trying to broadcast a message and thus determine whether to increase their mini-slot counters or not. If no nodes have message to transmit on the bus at current slot, mini-slot counter in every node is increased by 1 every one mini-slot duration. If there is a node on the network that has a message with an ID equal to the current slot count and the message has already in the transmit buffer before the communication cycle begins, it is broadcast over the network. At this time all nodes halt the incrementing of their mini-slot counter until they detect the end of the message. This process continues until the dynamic segment concludes.

![Figure 1 Structure of the FlexRay communication cycle](image1)

**Figure 1 Structure of the FlexRay communication cycle**

![Figure 2 Structure of the static segment](image2)

**Figure 2 Structure of the static segment**

**Figure 3 Structure of the dynamic segment**

**BG SCHEDULE IN FLEXRAY-BG SPECIFICATION**

Figure 4 shows the bus guardian schedule for one communication cycle described in FlexRay Communication System Preliminary Node-Local Bus Guardian Specification Version 2.0.9[13]. In the static segment, the BG enables transmission to the communication medium for all configured slots according to bus guardian configuration parameters (set of numbers of active static slots). And the BG enables transmission for the entire duration of the dynamic segment. It disables transmission for the entire duration of the symbol window and the NIT.

As shown in Figure 4, in the static segment the BG restricts transmission attempts of the communication controller to the configured time slots. If the BG detects any mismatch between the schedules of the communication controller and the BG, it signals an error condition to the host and inhibits any further transmission attempts. So the babbling idiot failure can be tolerated in the static segment. However, the BG do not provide the media with any protection since it enables transmission for the entire duration of the dynamic segment. Fail silence can not be enforced in the dynamic segment. If any node in the system suffers babbling idiot failure, the message transmission in the whole system will be affected. Therefore, from this point of view safety of FlexRay communication system has been doubted by experts in related fields.
Figure 4  Bus Guardian schedule in normal operation

DESIGN OF A NOVEL BG TO ENFORCE FAIL-SILENT BEHAVIOR OF THE NODE IN THE ENTIRE FLEXRAY COMMUNICATION CYCLE

Aiming at the problem mentioned in the previous section, a novel BG has been designed to enforce fail-silent behavior of the node in the entire FlexRay communication. The BG is working as the FlexRay Specification describes in the static segment. While as far as the dynamic segment is concerned, a novel algorithm is proposed according to the dynamic mini-slotting based scheme. Several core concepts have been proposed in this paper:

- Complete Schedule Table (CST): set of the Frame IDs and corresponding transmitting time allotted to the node in the static segment.
- Semi Schedule Table (SST): set of the Frame IDs allotted to the node in the dynamic segment.
- Max_Write_Time: time parameter which specify the maximal duration a node can occupy the medium in a dynamic slot.

NODE ARCHITECTURE WITH BUS GUARDIAN

As shown in Figure 5, the node architecture adopted in this paper primarily conforms to the FlexRay Communication System Protocol Specification Version 2.0[14]. And definitions of some interfaces between components have been corrected. It consists of one communication controller (CC), one host, two bus guardians, and two bus drivers. The CC is responsible for executing the FlexRay communication protocol. And the host is responsible for executing the distributed control applications. Each communication channel has one bus driver to connect the node to the channel. Bus guardians are applied to each channel in order to protect the channel(s) from improper transmissions in the time domain.

The interfaces between BG and other component are depicted briefly in the following.

The host and bus guardian share an interface that allows the host to send configuration data to the BG and to receive status information from the BG.

The interface between the CC and the BG serves to synchronize the BG to the schedule of the CC, allowing the BG to supervise the transmit operation of the CC. In addition, the CC uses this interface to provide signals to synchronize the BG schedule to the local representation of the global clock. The CC generates the ARM signal to indicate the start of the communication cycle. The MT signal, also generated by the CC, is used as a clock signal for the BG and is aligned to the start of the CC’s internal locally-corrected macrotick. The BGT signal, also generated by the CC, is used by the bus guardian for supervision of the MT period. The TxEN (Transmit Data Enable) signal, also generated by the CC, is used by the BG to detect whether or not the CC tries to transmit. The Error signal is asserted by the bus guardian if a violation of the bus access cycle is detected.
The interface between BG and BD comprises two signals, BGE and RxEN. The BG uses the BGE (Bus Guardian Enable) signal to enable or disable the BD's transmission of data onto the bus. The BGE signal drives a switch located in the transmission path of the bus driver, which either connects the transmission path of the node to the broadcast bus or isolates it from the bus. The BD uses the RxEN (Receive Data Enable) to indicate to the BG whether or not data is currently being received.

In this architecture, the scheduler inside the BG is clocked directly with the MT signal from the CC. This approach of having the CC provide the synchronization information to the BG keeps the complexity of the BG at a minimum. However, the shortcoming of this approach is that it contains a potential common failure mode. This problem is solved by introducing the BGT signal. The BGT signal is directly derived from the clock oscillator of the communication controller by means of a frequency divider and its period is not influenced by the clock synchronization correction. The BG detects the correctness of clock synchronization of the CC by checking the ratio between the MT period and the BGT period. If the ratio exceeds pre-defined configurable limits, error is detected. And supervision of the BGT signal is performed by means of the internal timer of the bus guardian.

**BG OPERATION IN FLEXRAY COMMUNICATION CYCLE**

As shown in Figure 6, the BG consists of schedule table, state machine, and a timer. The schedule table, composed of Complete Schedule Table and Semi Schedule Table, contains the related schedule information of the CC. The state machine controls the operation of the bus guardian. The independent timer is required to avoid temporal coupling between the BG and the CC.

![Figure 5 Node Architecture with Bus Guardian](image)

In the static segment of the communication cycle, the working principle of the BG conforms to the FlexRay BG Specification. If the bus guardian detects any mismatch between the schedules of the communication controller and the Complete Schedule Table of the BG, it signals an error condition to the host and inhibits any further transmission attempts. In the dynamic segment of the communication cycle, BG synchronizes its own mini-slot counter to the node according to the RxEN signal. The BGE signal is controlled by means of the Semi Schedule Table. The BGE signal is disabled in default and thus inhibit the CC from writing to the bus. As the mini-slot counter increase, if the mini-slot counter matches one of the Frame IDs in the Semi Schedule Table, the BGE signal is enabled. Then the CC is allowed to write to the bus. The duration of BGE enabled(TBGE) is
determined by the following equations (equation 1 and 2).

- In case of that the node has no message to transmit in the mini-slot:
  \[ T_{BG} = 1 \text{ mini-slot} \]  
  \[ (1) \]

- In case of that the node has a message to transmit in the mini-slot:
  \[ T_{BG} = \min(\text{Max\_write\_time, Message\_transmit\_time}) \]  
  \[ (2) \]

Where:
- Message\_transmit\_time represents the duration for message transmission.

Compared to Figure 4, Figure 7 gives an example illustrating the BG Schedule in the novel BG. As shown in Figure 7, the novel BG provides protection in the dynamic segment according to the Semi Schedule Table.

![Bus Guardian](image)

**Figure 6 The logic structure of Bus Guardian**

![BG Schedule](image)

**Figure 7 BG schedule in the novel BG**

**PERFORMANCE ANALYSIS**

**IMPROVING THE RELIABILITY OF MESSAGE TRANSMISSION IN THE DYNAMICS SEGMENT**

Due to the essential characteristics (namely, flexibility) of the minislot-based scheme, the novel BG can not detect all babbling idiots in the dynamics segment. However, the BG can restrict transmission attempts of the communication controller to the configured slots by means of the Semi Schedule Table. Therefore, false silence can be enforced outside of the allotted slots. While in the slots allotted to the node the maximum effect of undetected faults is bounded by introducing the time parameter Max\_Write\_Time.

**GUARANTEEING THE REAL TIME PERFORMANCE OF MESSAGE TRANSMISSION IN THE DYNAMIC SEGMENT**

As shown in Figure 8, two situations of the message transmission in the dynamic segment are illustrated. In situation 1, messages with higher priority seldom need transmission (only message ID m+3 is sent). When the slot counter reaches m+10 the message m+10 can be sent. While in situation 2, message m, m+2, and m+3 all need to be sent, which results in that the slot counter only reaches m+3 before the dynamic segment concludes. So the message m+10 cannot be sent in the current communication cycle. It has to be delayed to the next cycle or later. Therefore, transmission latency of messages with lower priority cannot be guaranteed in the dynamic segment. The novel BG can guarantee the real time performance of message transmission in the dynamics segment by the time parameter Max\_Write\_Time. The minimum number of messages which can be transmitted to transmit in the dynamic segment of one communication cycle (NUM\_min) can be calculated by the following
equation (equation 3):

$$\text{NUM}_{\text{inv}} = \frac{\text{Dynamic\_Segment\_Duration}}{Max\_Write\_Time}$$

(3)

Where:

Dynamics_Segment_Duration represents the whole duration of the dynamic segment in one communication cycle.

**Figure 8** Illustration of message transmission in the dynamic segment

**CONCLUSION**

In a distributed hard real-time system based on a broadcast bus for inter-node communication it is important to prevent babbling idiot failure. In FlexRay communication system, BG is used to overcome babbling idiot failure. However, the BG does not provide any protection in the dynamic segment according to FlexRay BG Specification. To enforce fail-silent behavior of the node in the entire FlexRay communication cycle, a novel BG is proposed. The mechanisms of the novel BG are presented along with the node architecture necessary for implementing the presented technique. Mathematical analysis results shows that the novel BG can effectively enforce fail silence property in the entire cycle. And at the same time, the real time performance of messages in the FlexRay dynamics segment can also be guaranteed by the BG.

**REFERENCES**

11. TTA-Group, Time-Triggered Protocol