Probably Right, Probably on Time: An Analysis of CAN in the Presence of Host and Network Faults

Arpan Gujarati*, Akshay Aggarwal†, Allen Clement‡, Björn B. Brandenburg*
*MPI-SWS, Germany †IIT-Kanpur, India ‡Google Inc., Zurich

***Abstract*—Safety-critical real-time systems that operate in electromagnetically noisy environments must be designed to withstand electromagnetic interference. Typically, networked systems handle this issue by retransmitting corrupted messages and by replicating critical processes on independent hosts. However, there exists a fundamental tradeoff between the two techniques: to increase a system's resiliency to message retransmissions without violating any deadlines, a low bus load is favorable; whereas adding replicas requires more messages to be sent, which in turn increases the bus load. This paper presents our ongoing work on a probabilistic analysis that quantifies this tradeoff in CAN-based systems, which enables system designers to select an optimal number of replicas that maximizes the probability of a correct *and* timely operation of a distributed real-time system.**

I. Introduction

Automotive embedded systems are surrounded by spark plugs and electric motors. Industrial embedded systems may be deployed in close vicinity to high-power machinery. Robots may need to operate in environments exposed to hard radiation. All of the above are examples of safety-critical real-time systems that are susceptible to electromagnetic interference (EMI) and must be designed to withstand its effects; including hangs, crashes, or incorrect outputs (*node faults*), and, in networked systems, also corrupted messages (*transmission faults*) [1, 4].

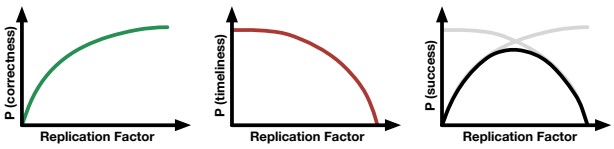
Presently, networked systems tolerate transmission faults by retransmitting corrupted messages (*e.g.*, CAN controllers automatically retransmit a message if any host connected to the bus reports a transmission fault). Node faults on the other hand are typically tolerated by replicating critical processes on independent nodes. In the context of real-time systems, however, both these techniques fundamentally conflict with each other.

While a low bus load is favorable to ensure that deadlines are not violated due to retransmissions (*i.e.*, the more slack, the better), replication increases the bus load as more messages are sent (*i.e.*, it reduces the available slack). Thus, beyond a certain point, adding replicas actually hinders a system's ability to meet all deadlines, and can ultimately decrease its overall reliability.

Our research aims to develop a probabilistic analysis that quantifies the fault-tolerance versus timeliness tradeoff in CAN-based safety-critical real-time systems. The proposed analysis will enable system designers to select an optimal number of replicas that maximizes the probability of a correct *and* timely execution of a distributed real-time system.

II. Key Idea and Challenges

Key idea: We expect the probability of a correct execution to improve with replication due to better tolerance against node faults, but the probability of a timely execution to degrade with replication due to higher bus load, as illustrated in Figs. 1(a) and 1(b). With the proposed analysis, it is possible to determine the probability of a correct *and* timely execution for different replication factors (*i.e.*, number of replicas per tasks), and hence, quantify the benefits of replicating any given task or compute the optimal replication factor (see Fig. 1(c)).



(a) (b) (c)

Fig. 1. *Replication factor* denotes the number of independent replicas of critical tasks. The illustration shows that: **(a)** probability of correct execution improves with replication; **(b)** probability of timely execution degrades with replication; and **(c)** probability of correct *and* timely execution initially increases with replication due to better fault-tolerance, but then starts degrading with replication due to increasing bus load. The graphs are just a conceptual illustration, *i.e.*, they do not represent actual probabilities or empirical results.

Challenges: Although Tindell et al. [5] and Broster et al. [2] have extensively investigated the problem of fault-aware timing analysis for the CAN message bus, a simultaneous analysis of both timeliness and correctness in a system with replicated tasks is significantly more involved. Below, we list some of the major challenges that we have identified in this work.

Assume two tasks A and B running on independent hosts networked via a CAN bus. A regularly sends message m_a to B. Suppose that A is replicated in order to ensure that B's output is always based on a correct input from A, *i.e.*, each replica of A, denoted by A_i, sends a separate message m_a^i to B. Thus, B must implement an aggregation procedure to select/generate the correct input value. In addition, since the hosts may or may not have access to a synchronized clock, and since each message m_a^i may be generated either periodically or sporadically, the aggregation procedure needs to be implemented differently for each scenario (*e.g.*, B cannot know the absolute deadline of each message m_a^i if the clocks are not synchronized).

Another major challenge is to define a new analysis that establishes timeliness guarantees for a set of messages, *e.g.*, how to compute the probability that a "sufficient" number of messages of a larger message set $\{m_a^i\}_i$ are transmitted "on time", where "on time" and "sufficient" may mean that at least a majority of messages reach before B executes its aggregation procedure. Jitter makes this exercise even more challenging. There is also the problem of finding a replication-aware priority assignment that optimizes the safety of highly critical tasks (which are replicated), but without compromising on the timeliness guarantees of tasks that are not replicated.

In the next section, we explain an initial probabilistic analysis for hosts with synchronized clocks and periodic messages. In particular, we elaborate on the technique to account for both node faults and transmission faults in the same analysis.

III. Replication-Aware Probabilistic Analysis

System model: We assume a CAN-based system consisting of $r+1$ tasks running on independent hosts, including r replicas of task A and a single instance of task B (as in Sec. II). Each

replica A_i periodically sends message m_a^i to B, with period T_a, deadline D_a, and jitter J_a^i. The j^{th} instance of this message is denoted as $m_{a,j}^i$ and its absolute deadline is denoted as $d_{a,j}$.

Assuming that all the hosts have globally synchronized clocks, B implements the following aggregation protocol for the j^{th} round of messages: at time $d_{a,j}$, B collects all messages in $\{m_{a,j}^i\}_i$ that it has received until $d_{a,j}$, computes the payload that is contained in the majority of these messages, and considers this payload as its input value for the j^{th} round. In this initial work, we assume that B can distinguish between any two messages $m_{a,j}^i$ and $m_{a,k}^i$ transmitted by the same replica.

Fault model: We consider three types of faults: omission faults and commission faults (*i.e.*, message omissions and corruptions on the host), and transmission faults (*i.e.*, message corruptions on the bus). It is assumed that the host running B executes fault-free. However, we can do away with this assumption by replicating B and analyzing the multiple replicas of B individually. In addition, we assume that messages omitted on the host running A do not interfere on the CAN bus, an observation that we exploit in the proposed analysis.

Each fault type is associated with a probabilistic fault model. As assumed by Broster et al. [2], we use a Poisson distribution to model the probability that n transmission faults occur in any time period of length t, *i.e.*, $P_t(n,t)$. For the node faults, we use a simpler model that associates fixed probabilities P_o and P_c with an omission fault and a commission fault per message, respectively. For brevity, we assume the same fault probabilities across all hosts; the full analysis can be extended to incorporate host-specific probabilities. The analysis does not cover "babbling idiot" failures, as we rely on the bus guardian solution proposed earlier by Broster et al. [2].

Analysis: Let $P_{su}(M_j)$ denote the probability of "success", *i.e.*, that for any message set $M_j = \{m_{a,j}^i\}_{1 \le i \le r}$, the input value inferred by the aggregation protocol at B is indeed correct. Intuitively, $P_{su}(M_j)$ is a combination of three factors: omission faults and commission faults at A, and retransmission delays on the bus. Thus, in order to compute $P_{su}(M_j)$, we use the following approach. **Step 1:** Suppose that $O_j \subseteq M_j$ denotes the set of messages omitted at A. Let the corresponding probability be denoted as $P_{om}(O_j)$. Since messages omitted at A do not interfere on the CAN bus, we analyze only the timeliness and correctness of messages in $O'_j = M \setminus O_j$. **Step 2:** Given O'_j, we then consider $|O'_j|$ cases, where *case* k implies that only k out of $|O'_j|$ messages have response time less than or equal to D_a. Let the corresponding probability of timeliness be denoted as $P_{ti}(O'_j, k)$. **Step 3:** Finally, for each *case* k, *i.e.*, among all the k messages transmitted on time, we consider all combinations of corrupted and correct messages and for each combination decide whether the value inferred by the aggregation protocol at B is correct or wrong. The corresponding probability of correctness is denoted as $P_{co}(O'_j, k)$. Using the aforementioned steps, $P_{su}(M_j)$ can be defined as $P_{su}(M_j) = \sum_{O_j \subseteq M_j} P_{om}(O_j) \cdot \sum_{k \le |O'_j|} P_{ti}(O'_j, k) \cdot P_{co}(O'_j, k)$.

The individual probabilities in the definition of $P_{su}(M_j)$ are defined as follows. $P_{om}(O_j) = P_o^{|O_j|}(1-P_o)^{|O'_j|}$since it directly depends on P_o. For computing $P_{ti}(O'_j, k)$, we use $P_t(n,t)$ and assume that if x transmission faults occurs in the interval when messages in O'_j are transmitted, all messages are delayed by x retransmissions in the worst case. Due to space constraints, we omit a detailed timing analysis for this step, but the high-level approach is inspired by Broster et al.'s method [2]. For $P_{co}(O'_j, k)$, recall the aggregation protocol at B: if a majority of the k messages transmitted on or before $d_{a,j}$ are correct, then B infers the correct value as input. Thus, using P_o and assuming that l denotes the number of messages with corruptions: $P_{co}(O'_j, k) = \sum_{l=0}^{\lceil k/2 \rceil - 1} \binom{k}{l} \cdot P_c^l \cdot (1-P_c)^{k-l}$. Although the preceding definition pessimistically assumes that $P_{co}(O'_j, k) = 0$ if k is even and $l = k/2$, a more fine-grained analysis of such ambiguous cases is planned. Next, we conclude with a brief discussion of initial experiments and future work.

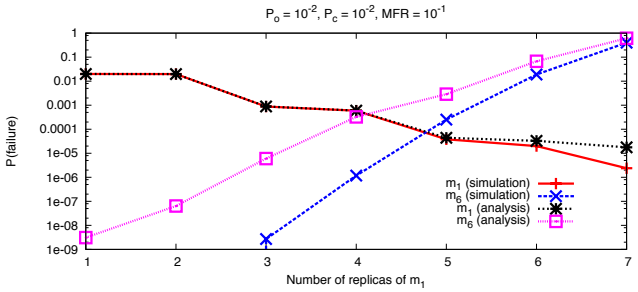


Fig. 2. Analysis and simulation of a set of six implicit-deadline messages $\{m_1, \ldots, m_6\}$, with periods $T_1 = 5ms$ and $T_2 \ldots T_6 = 10ms$. We assume that m_1 can be affected by an omission fault, a commission fault, or a transmission fault, whereas $m_2, \ldots, m_6$ are affected by transmission faults. Thus, only m_1 is replicated. Priorities are assigned using Davis and Burns's algorithm [3], which assigns the highest priorities to m_1 and its replicas. The Poisson distribution for transmission faults assumes a mean fault rate (MFR) of 0.1 faults per *ms*. Results for m_2–m_5 have been omitted to avoid clutter. The CAN bus schedule was simulated for 100,000 seconds and each message instance, *i.e.*, each $\{m_{i,k}\}_{i,k}$, was counted as a separate event. The entire simulation procedure was repeated 640 times.

IV. Example and Future Work

We simulated the transmission of a synthetic message set and compared the observed results with the analytical predictions (Fig. 2). For message m_1, which was replicated, the probability of failure reduced upon increasing its replication factor; whereas for message m_6, which was not replicated, a reverse trend was observed. The results corroborate the expected fault-tolerance versus timeliness tradeoff, and show that the analysis is accurate enough to predict similar trends. Another observation is that if the replicated task has a higher priority, the robust priority assignment algorithm [3] assigns high priorities to its replicas as well, compromising on the timeliness of the lower priority tasks.

In the future, we plan to develop a replication-aware priority assignment scheme for CAN bus messages. In addition, using the proposed analysis, we aim to answer questions such as: what is the optimal replication factor so that the probability of failure for each message is below a given threshold, how does a system's resiliency change when taken from lab conditions to harsher environments, *etc.* Currently, we are working on deriving aggregation procedures and analyses for hosts with asynchronous clocks and for sporadic messages.

References

[1] S. Borkar, "Designing reliable systems from unreliable components: the challenges of transistor variability and degradation," *Micro*, 2005.

[2] I. Broster, A. Burns, and G. Rodríguez-Navas, "Timing analysis of real-time communication under electromagnetic interference," *Real-Time Systems*, 2005.

[3] R. I. Davis and A. Burns, "Robust priority assignment for messages on controller area network (CAN)," *Real-Time Systems*, 2009.

[4] I. Noble, "EMC and the automotive industry," *Electronics & communication engineering journal*, 1992.

[5] K. Tindell, A. Burns, and A. J. Wellings, "Calculating controller area network (CAN) message response times," *Control Engineering Practice*, 1995.