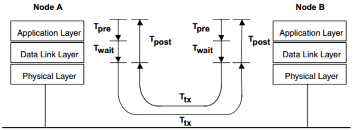
CAN message response time

Embedded systems engineering

David Kendall

Northumbria University

Introduction



$$T_{delay} = T_{pre} + T_{wait} + T_{tx} + T_{post}$$

- We have seen that in general the end-to-end delay of an interaction between two nodes in a distributed system can be decomposed as above.
- Here we focus on the *message response time* ($T_{pre} + T_{wait} + T_{tx}$) for CAN.
- This problem was first considered by Tindell and Burns in 1994. Their analysis has been widely used.
- Their original analysis was revised in 2007 by Davis et al.

Fixed-priority non-preemptive scheduling of CAN

- At the start of every arbitration, each node enters the highest priority message in its queue
- Message m is queued by a software task taking a time between 0 and J_m to queue the message (*queuing jitter*)
- Event that causes queuing of message occurs with *minimum inter-arrival time* T_m (*message period*)
- Each message has a *deadline* D_m – maximum allowed time from occurrence of initiating event to end of successful transmission of message
- R_m is the *worst-case response time* of the message – longest time from occurrence of initiating event to reception of message
- Message m is *schedulable* if $R_m <= D_m$. System is *schedulable* if all messages are schedulable.

CAN message response time (Tindell and Burns 1994)

$$R_m = J_m + w_m + C_m$$

- J_m – queuing jitter
- w_m – worst-case queuing delay (T_{wait})
- C_m – worst-case message transmission time

Message transmission time

$$C_m = \left(g + 8s_m + 13 + \left\lfloor \frac{g + 8s_m - 1}{4} \right\rfloor \right) \tau_{\text{bit}}$$

- C_m – transmission time
- g – 34 for standard CAN frame
- s_m – number of data bytes in message
- τ_{bit} – transmission time for 1 bit
- For 11-bit identifiers, formula simplifies to

$$C_m = (55 + 10s_m)\tau_{bit}$$

Queuing delay

Queuing delay w_m consists of

- *Blocking time* B_m – maximum time that message might be delayed waiting for completion of transmission of a lower priority message
- *Interference* – maximum time that message might be delayed by higher priority messages (that win arbitration)

Maximum blocking occurs when a message m is queued just as the longest lower priority message begins transmission

$$B_m = \max_{k \in lp(m)} (C_k)$$

- where $lp(m)$ is the set of messages with lower priority than m

Queuing delay

Tindell and Burns 1994 gives

$$w_m = B_m + \sum_{\forall k \in hp(m)} \left\lceil \frac{w_m + J_k + \tau_{bit}}{T_k} \right\rceil C_k$$

This equation can be solved using the recurrence relation

$$w_m^{n+1} = B_m + \sum_{\forall k \in hp(m)} \left\lceil \frac{w_m^n + J_k + \tau_{bit}}{T_k} \right\rceil C_k$$

where a suitable starting value is $w_m^0 = B_m$

Iterate until either

- $J_m + w_m^{n+1} + C_m > D_m$: m not schedulable, or
- $w_m^{n+1} = w_m^n$: worst-case response is $J_m + w_m^{n+1} + C_m$

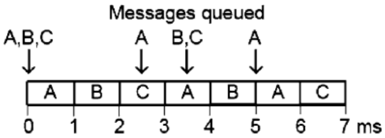
A problem with the analysis

Davis et al. 2007 showed that the previous analysis is flawed.

Consider this simple example

Message	Priority	Period	Deadline	TX time
A	1	2.5 ms	2.5 ms	1 ms
B	2	3.5 ms	3.25 ms	1 ms
C	3	3.5 ms	3.25 ms	1 ms

and its analysis for worst-case response of message C



Analysis gives $R_C = 3$ but example shows $R_C = 3.5$!!!

Sufficient but not necessary test

- Davis et al. 2007 give a detailed account of the reasons for the flaw in the original analysis
- They provide a revised, exact analysis that is not flawed
- The simplest revision that they propose is to assume that the blocking time for each message is given by the longest possible transmission time of a message on the bus, leading to the formula below

$$w_m^{n+1} = B^{\mathrm{MAX}} + \sum_{\forall k \in hp(m)} \left\lceil \frac{w_m^n + J_k + \tau_{bit}}{T_k} \right\rceil C_k$$

- where B^{MAX} is the longest possible transmission time of a message on the bus

Acknowledgements

- Tindell, K., Burns, A. *Guaranteeing message latencies on Controller Area Network (CAN)*. In: Proceedings of 1st International CAN conference, pp 1–11, 1994
- Davis, R., Burns, A., Bril, R., Lukkic, J., *Controller Area Network (CAN) schedulability analysis: Refuted, revisited and revised*, Journal of Real-time Systems, 35:239-272, 2007
- M. Di Natale, H.Zeng, P. Giusto, A. Ghosal, *Understanding and using the Controller Area Network Protocol*, Springer Verlag, 2012