iCC 1996

3rd international CAN Conference

in Paris (France)

Sponsored by

Motorola Semiconductor
National Semiconductor
Philips Semiconductors

Organized by

CAN in Automation (CiA)
international users and manufacturers group
Am Weichselgarten 26
D-91058 Erlangen
Phone +49-9131-69086-0
Fax +49-9131-69086-79
Email:headquarters@can-cia.de
URL: http://[illegible]

Sandra Schneider
Daimler-Benz AG
Abt. F1M/ET
70546 Stuttgart

Performance Analysis for Automotive CAN Systems

Abstract:
In recent years systems of CAN-connected automotive control units have been increasing rapidly both in complexity and size. Therefore not only functional but also performance aspects must be considered during development.
For the system to be feasible, all communication timing constraints for all functions have to be proven. This feasibility has to be assured in early design stages already.
This paper presents simulation results of a bus model as a basis for performance analysis of automotive CAN systems. The single server queue with non-preemtive priorities was chosen as a simple model of the bus. Each priority class represents a message of a certain identifier. Bus arbitration is implicitly modelled by the non-preemtive priorities of the service discipline.
The simulation model considers not only cyclic messages (messages with constant interarrival times) but also sporadic traffic sources. Those sources emit only when their corresponding function is active.
The obtained simulation results give the mean waiting time for the different priorities and can be used to tell if the system with the planned communication load can work acceptably for all realized functions. If necessary the design can be changed in an early stage to improve network performance.

Introduction

Electronic systems currently under development at Mercedes-Benz consist of several communication buses interconnected by gateways. These systems comprise more than 25 communicating control units. Systems of this size and complexity require a formal design method during development. A special need for performance analysis exists for the CAN Class B bus. It has the constraint of a low transmission speed combined with a high number of body electronic control units. Customarily functions do not correspond to control units. As shown in figure [REF: funcdec]functions are usually divided into sub-functions which are realized on different hardware units. Due to the high number of functions, tool supported executable specifications of these functions are necessary. The design method in section [REF: sysspec]describes how to get from the functional view of a system to a specification of the distributed system. The necessary CAN communication parameters are also defined during the design steps.
Given the whole system specification and all the communication needs for all messages, a simulation model is built (section [REF: simmodel]) which gives insight into the performance of the system.

System Specification

During this specification stage the designers of the different functions use decomposition in terms of internal sub-functions. This method is supported by the Statemate $^{©}$ tool .
The specifications comprise :
 a behavioural description,
 a definition of interfaces between the different functions and to sensors/actors,
 performance requirements like execution times that have to be satisfied.
For implementation on hardware devices the functions and their internal sub-functions have to be distributed. Sub-functions of a function are possibly put on different devices due to specific needs like proximity to sensors/actors.
At this point there is a need for communication at the interfaces between functions as well as at the new interfaces between sub-functions.
When distributing functions or parts of them on different control units in the car body, also the parameters for CAN communication have to be given. These parameters are determined by the function and include:
 message id = priority,
 transmission type: the most frequently used types are cyclic messages and messages that are sent on demand only,
 cycle times.
Time constraints on communication for the different messages have to be given along with these parameters.
Given the whole system specification and the communication needs for all messages, the feasibility of the system from the performance point of view has to be checked.
In the following section a simulation model for this system is presented.

Simulation

[LABEL: simmodel]Simulation Model
For CAN-Communication transmission time is always constant for a message (error free transmission assumed), but for the waiting time until transmission no fixed value can be given. It depends highly on the busload and the priority of the message. Worst case waiting times can only be given for high-priority messages. For lower priority messages mean values and statistical techniques have to be used to assure feasibility of the design under the given network load.
The simulation model consists of a single server queue with a non-preemptive priority service discipline. Items arriving to the queue correspond to CAN messages demanding transmission. The priorities of arriving items correspond to CAN-Identifiers. Arbitration is modelled implicitly by the non-preemptive priority discipline. Messages of the same identifier belong to the same priority class. The bus is represented by the server where the service time of a message corresponds to its transmission time. The service time of messages of one priority class is constant due to their fixed message length.

The simulated non-preemptive priority queue is analytically tractable for Markovian arrival patterns. These analytical results, given in are valuable for comparison purposes to validate the simulation and to compare them to the obtained results for other arrival patterns.
Sources that transmit their messages cyclically (fig. [REF: source]) generate the baseload of the system. The cycle time of these messages has been specified during the distribution step described in section [REF: introduction]. The transmission requests for these messages are not subject to significant statistical fluctuations - except for the jitter which is introduced to lower average waiting times, see section [REF: results].
Other types of sources in the system are more difficult to model. The transmission of these messages is requested by the application function. For these sources a workload characterization is done using an ON-OFF-source model, shown in figure [REF: bursty]. These sources follow an ON-OFF pattern where the length of both the ON and the OFF phase is exponentially distributed.
The simulated system consists of 88 identifiers of which 52 are cyclic and 36 come from bursty sources. The cycle times for the messages vary between 20 and 2500 ms.
The simulation was done using SimPack's © C++ queueing libraries which offer convenient random number generation, future event list and priority queue administration.
The first 900 s of simulation did not enter in the statistics in order to observe the steady state behaviour of the queue only , . It is not easy to decide on the length of the "warm-up-phase". For our choice of 900 s the simulated results of the analytically tractable queue with Markovian arrivals correspond nicely to the calculated results.

Simulation Results

What we present here are mean values for waiting times until bus access for all priority classes. These mean values give a general idea if the planned configuration and distribution of functions is feasible.
Simulations of bursty sources are done using active to passive time ratios of 1:2. The actual mean duration of the active time of the bursty sources varies between 15, 30 and 60 s. Comparing the results for these three different values (fig. [REF: burstcomp]), no significant differences are noticable. As long as the active to passive time ratio stays fixed the lengths of the bursts do not influence much the result.
The simulation results in figure [REF: burstcomp]give very high waiting times for a number of messages. Interestingly enough only cyclic messages experience this high delay. It was further found that the messages that experience this delay depend on the random number stream. This is an effect of several cyclic messages starting transmission at (almost) the same moment and having cycle times which are either the same or multiples of each other. The lower priority message always loses arbitration and has to wait for the higher priority message to finish transmission.
To avoid this effect a jitter can be introduced for the transmission request times for cyclic messages. This requires functions that tolerate this minor shift in transmission request times. When introducing a jitter, the times between two transmission requests are no longer constant but vary between +2%and -2%(fig. [REF: srcjit]). Figure [REF: effofji]shows that this smoothes the curve and avoids the very large mean waiting times for some cyclic messages.
In figure [REF: jitter]the the simulation results for the system with jitter are compared to calculated values for a queue with a Poisson arrival stream. The mean values of the Poisson arrival stream were taken to be the mean arrival rate of each priority class. The results show that jitter not only reduces mean waiting times significantly but also that the theoretical results for the queue with Markov arrivals and constant service times give a good approximation for mean waiting times in the system. Simulated waiting times are even lower than the calculated ones.
For a quick estimation of the waiting times in the system it is sufficient to know the mean arrival rate of messages for each priority class and to calculate their waiting times as if it were a priority queue with Markov arrivals. As the simulation shows, these calculated values will give a good approximation of real waiting times - they will even give scarcely larger values due to the greater variance of the Poisson arrivals.
If this analysis shows that the performance is not sufficient, there are the following possibilities of changing the system to improve performance, even before setting up the system in hardware:
introducing a jitter for transmission request times of cyclic messages,
repeating the design step of distributing functions and sub-functions including reorganization of cycle times and message priorities - this time with attention to crucial performance points.

Conclusion

From given specifications and performance requirements a performance simulation model is built. The simulation of this model gives insight into performance aspects of the system. These aspects, defined during the design steps preceeding the simulation, can now be checked and the design can be corrected early in the development process.
Through the introduction of a jitter for the transmission request times of cyclic messages their dependence on the random number seed can be eliminated. Moreover their waiting times are reduced significantly. The simulation results tempt to suppose that waiting times for this kind of system can be approximated nicely by the analytical results a priority queue with Markov arrivals.
To get more information on system performance than mean waiting times it will be interesting to look at higher moments of waiting times to be able to give statistical estimations or bounds on performance.

References

\bibitem{calvez}
J.~P. Calvez.
\newblock {\em Embedded Real Time Systems}.
\newblock Wiley series in software engineering practice. John Wiley \& Sons,
  Chichester, 1993.

\bibitem{fishwick}
Paul~A. Fishwick.
\newblock {\em Simulation Model Design and Execution}.
\newblock Prentice Hall, Englewood Cliffs, New Jersey, 1995.

\bibitem{statemate}
D.~Harel, H.~Lachover, A.~Naamad, A.~Pnueli, M.~Politi, R.~Sherman, and
  A.~Shtul-Tauring.
\newblock Statemate: A working environment for the development of complex
  reactive systems.
\newblock {\em IEEE Transactions on Software Engineering}, 16(4), 1990.

\bibitem{kant}
Krishna Kant.
\newblock {\em Introduction to Computer System Performance Evaluation}.
\newblock McGraw-Hill computer science series. McGraw-Hill, New York, 1992.

\bibitem{kleinrock2}
Leonard Kleinrock.
\newblock {\em Queueing Systems, Volume II: Computer Applications}.
\newblock John Wiley \& Sons, New York, 1976.

\bibitem{lawkelton}
Averill~M. Law and David Kelton.
\newblock {\em Simulation Modeling and Analysis}.
\newblock McGraw-Hill, New York, 1991.

\bibitem{mitrani}
Israel Mitrani.
\newblock {\em Simulation Techniques for Discrete Event Systems}.
\newblock Cambridge University Press, Cambridge, 1982.