PREDICTABILITY OF MESSAGE TRANSFER IN CSMA-NETWORKS

J. KAISER

University of Ulm, Department of Computer Structures, Germany
E-mail: kaiser@informatik.uni-ulm.de

M.A. LIVANI

Siemens AG, Information and Communication Mobile, Germany
E-mail: Mohammad.Livani@icn.siemens.de

W. JIA

City University of Hong Kong, Department of Computer Science, China
E-mail: weijia@cs.cityu.edu.hk

The paper describes a novel approach to establish predictable message transfer between a set of co-operating nodes controlling a physical process. The protocol allows the coexistence of different classes of criticality in a real-time communication network. The approach exploits a priority scheme to guarantee static preplanned message slots for hard real-time communication and enforces a dynamic earliest deadline first scheduling scheme for SRTMs. Soft real-time or SRTMs can be sent at any time without interfering with the hard real-time traffic. The mechanism is embedded in the media access control (MAC) layer of the communication system and realised by a new variant of the virtual time CSMA (VTCSMA) scheme. One of the benefits of our approach is that it can be applied to all kinds of CSMA-networks including Ethernet and wireless LAN.

1 Introduction

Distributed Real-Time Systems comprising a large number of intelligent processing nodes which co-operatively control real world processes like industrial plants or traffic systems become the popular architecture for such applications. All processing nodes are autonomous and work concurrently. To avoid a single point of failure, these nodes are not co-ordinated by a single central control entity but exchange messages to disseminate critical information and co-ordinate joint actions. Therefore, these systems need a predictable communication subsystem. Usually there are highly time-critical messages which require bounded transmission times otherwise the system may fail unpredictably causing serious effects or damage to the environment. These messages are called hard real-time messages (HRTM) and the transfer of each individual HRTM must be certain. Besides time critical message transfer there may be also less critical information which should be accommodated by the communication system. This information can be visual data from camera supervising a plant, statistical data gathered by a smart sensor or navigational information in a robot which is used to create a map for long term route planning rather than for short term steering. Although the transmission of the data may have timing constraints e.g. in the example of transferring frames from the camera, these constraints are soft in the sense that a temporary violation of the timing constraints leading to a quality reduction can be tolerated. Messages of this class are refered to as soft real-time messages (SRTM).

To accomodate hard real-time traffic, conflicting timing requirements between HRTMs resulting in a temporary overload situation have to be omitted. This can only be achieved by proper off-line scheduling of the communication resources. If HRTMs and SRTMs share the same communication medium, it has to be guaranteed that the SRTMs by no means interfere with the hard part. Some static hard scheduling schemes take the approach to consider every message as a periodic HRTM [1, 2]. This method is safe but it does not only consume more resources than actually needed, but also may end up with an infeasible system. Combining static pre-planned message scheduling and a dynamic message scheduling would be beneficial. Protocols like the IEEE 802.11 [3] for wireless LAN communication, the shared channel concept [4] and recently, the TTP/A protocol [5] implement this approach. However, they use a centralised approach with a single master [3] or an ubiquitous global time [2] [4] to guarantee exclusive communication of messages. We did not consider a central master because it constitutes a single point of failure and consumes additional bandwidth. The other approaches require global time in all nodes which has to be provided by a respective protocol. Simple nodes like the camera sensor, the sensor distributing navigational information or actuators which may not transmit critical messages at all, have to participate in the clock synchronisation protocol. If a node does not obey the global time discipline, it will inevitably corrupt the communication. This paper proposes an approach to realise a bus arbitration scheme which supports the coexistence of messages with different real-time requirements. It provides pre-planned time slots for HRTMs. For SRTMs, dynamic EDF-scheduling is applied. SRTMs and even non real-time messages (NRTM) can be sent any time and the scheme guarantees that they will not interfere with HRTMs. To enforce the reservations and the scheduling decisions, we devised a dynamic priority scheme presented in this paper.

2 Controlling the access to the network

The media access control layer responsible for arbitration and collision control in a network plays a key role for the predictable message transfer. We can distinguish between mechanims which statically avoid collisions [2, 4, 6, 7] and collision resolution mechanisms. It has to be noted that static collision avoidance schemes need complete off-line resolution of conflicting temporal requirements.

The most popular protocol based on collision resolution is the CSMA/CD (Carrier Sense Multiple Access/ Collision Detection) protocol of Ethernet. This protocol inherently suffers from unpredictable delays when collisions occur. Therefore, it is less suited for critical real-time applications. However, there are CSMA protocols which try to avoid collisions dynamically. Whenever the medium is free, the nodes do not immediately start transmitting messages but delay sending for a predefined amount of time. This technique is e.g. used in the IEEE 802.11 wireless communication standard [3]. A more general solution which realises a priority based network arbitration is provided by the virtual time CSMA protocol (VTCSMA) [8]. The basic idea is that different priorities can be realized on a CSMA bus by waiting times. The higher the priority of a message, the shorter its waiting time. The VTCSMA algorithm is a general approach, which maps different message parameters (e.g. deadline or arrival time) onto waiting periods, hence prioritizing messages and scheduling them according to different strategies (e.g. earliest deadline-first or first-come-first-serve)[9].

3 Scheduling of messages

Our protocol exploits VTCSMA as a basic mechanim to prioritize messages and thus enforce proper message scheduling. Additionally, we take account of the fact that HRTM traffic needs preplanning. For HRTM a time slot (Fig. 1) is statically reserved.

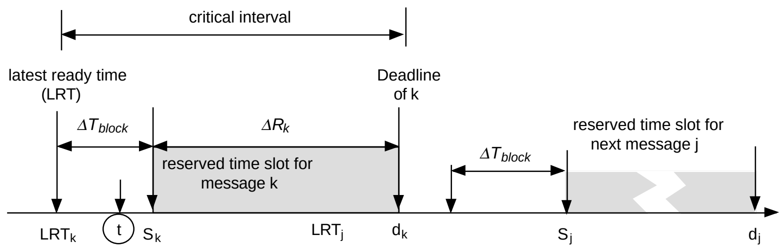

Fig. 1 Reserved time slots and critical interval for HRTMs

The reserved time slots are maintained in a data structure called calendar. The calendar is available at all nodes of the system which transmit HRTM. In contrast to the TDMA approach, however, the enforcement of the schedule is not dictated by global time but by the priority scheme. The benefit is that nodes which do not send HRTM do not need access to a global time base and can send their messages any time. Our scheme uses three priority classes. A HRTM gains the highest possible priority at the beginning of its reserved time slot. When a HRTM is in the sending queue and the reserved time-slot arrives, a HRTM is sent without any waiting time. Because no other HRTM is scheduled during this interval and all other messages always have a non-zero waiting period, no collision will corrupt the transfer.

However, due to the non-preemptive nature of message transmission on CSMA buses, every message in the worst case may be delayed by one message, which just has been started before the ready time of the respective HRTM. We define ΔT_{block} as the longest possible message transmission time. A HRTM *k* (with a reserved time-slot beginning at S_k) must be ready before $S_k - \Delta T_{\text{block}}$ (Fig. 1). Between $S_k - \Delta T_{\text{block}}$ and S_k, k is transmitted with a waiting time of $\Delta Cmin$, which is the minimum waiting time that can be distinguished from zero. This assures that a HRTM k gains the bus after $S_k - \Delta T_{\text{block}}$. Consider the situation illustrated in Fig. 1. If at time t the bus becomes idle, and HRTMs k and a SRTM l compete for the bus, then k is transmitted with no contention, and finishes before d_k. If k has already been transmitted, then l is transmitted with a waiting time $2*\Delta C_{\text{min}} \leq C_l \leq K*\Delta C_{\text{min}}$ (K is a system constant with $K > 2$. (see below)). Before its latest ready time, the waiting time for a HRTM is a random period between $(p+1)*\Delta C_{\text{min}}$ and $\Delta C_{\text{max}} = 2*p*\Delta C_{\text{min}}$. This assures that outside the reserved lot, a HRTM has a lower priority than a SRTM.

The proposed scheduling approach for hard real-time communication requires access to a global time reference with bounded inaccuracy. To guarantee that messages k and j do not collide, their senders must agree that t is inside of the reserved time-slot of k, and outside of the reserved time-slot of j. This agreement is guaranteed by leaving a gap between different reserved time-slots. A more detailed discussion of the temporal properties is ound in [10, 11].

For the soft real-time distributed activities the approach guarantees optimal scheduling by applying VTCSMA-D on the bus which complies to EDF scheduling. The waiting period $\Delta C_l(t)$ of a SRTM l at the time t is determined as function of the deadline in the interval: $2 \bullet \Delta C_{\min} \leq \Delta C_l(t) \leq K \bullet \Delta C_{\min}$. K is a system constant selected to assure that the waiting time for a SRTM is always shorter than for a non real-time message or an early HRTM. Thus, SRTMs have always priority except in the reserved slots for HRTMs.

Non real-time messages (NRTM) do not change their importance by the passage of time. Therefore, a fixed waiting time is assigned to NRTMs. In order to fulfil the requirement that a NRTM only use the bus in absence of real-time messages, the waiting time for a NRTM is $\Delta C_{\text{NRT}} > \Delta C_{\text{min}} + \Delta C_{\text{max}}$.

4 Simulation

To evaluate our scheme, we selected the SAE benchmark for electric vehicles [12]. It is defined for a couple of distributed sensors and actuators communicating via a network. The benchmark comprises 42 messages classes exchanged by 7 computing nodes. We distinguish respective sets of periodic and sporadic hard and SRTMs. The total data load caused by the messages, including headers, CRC, inter-frame spacing, etc. is about 0.466 Mbit/s. Because we wanted to use the SAE benchmark which reflects a realistic load instead of using a synthetic load generator, we could not increase the bus load by simply increasing the number or length of messages. Instead, we varied the bus bandwidth. This is equivalent to varying the load at constant bus speed. In order to simulate the worst case situation for scheduling, the message arrival times have been adjusted such that deadlines of SRTMs are equal, hence resulting in the maximum number of collisions. Although very pessimistic, this case must be taken into account to show whether message loss for HRTMs is possible or not. We simulated the bus scheduling for a period of 1000 seconds[1]. The simulation (cf. Fig. 2) shows that our scheduling mechanism is superior to the pure EDF scheduling based on VTCSMA-D, because in the worst case of equal dead-lines, VTCSMA-D results in timing failure even under an average bus load of about 25%.

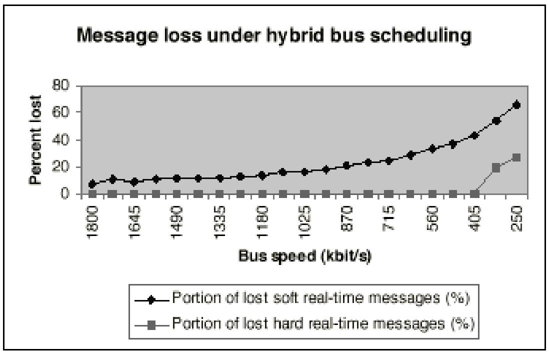

Fig. 2. Simulation of the hybrid bus scheduling algorithm

In contrast, the hybrid scheduling mechanism guarantees timely message delivery of HRTMs even under loads above 100%. In this case, of course, a great portion of SRTMs will miss their deadlines. The point at which HRTMs were lost was at the bus speed of 0.405 Mbit/s (about 115% average bus load). Although only 0.07% of HRTMs were lost, this would not be acceptable in a safety critical system as a electric vehicle. In our scheme, however, HRTMs are not really lost because of the unpredictable bus arbitration. Rather, at higher bus loads some HRTMs could not reserve any time-slot. Because the calendar for HRTMs is assumed to be constructed off-line, this violation would be recognized before the system actually is in operation. It has to be emphasized that this extreme bus utilization usually makes no sense in a real system. Here, it served as a worst case scenario to prove the guarantees preserved by the priority-based arbitration mechanism.

5 Conclusion

The paper describes a hybrid bus scheduling mechanism, which combines reservations for hard real-time traffic with the EDF scheduling used for soft real-

---

[1] The rate "η"of the virtual clock of the VTCSMA algorithm was set to η=8, i.e. the virtual clock runs eight times faster than real-time. η is a critical value in the VTCSMA protocol which roughly trades shorter waiting periods against the number of collisions. Because of space restrictions a detailed descussion of VTCSMA is not possibel here. The interested reader is refered to [9].

time activities. The contribution of the paper is how to enforce these scheduling decisions in a decentralized way on a general CSMA network like Ethernet or wireless LAN. We adapted the VTCSMA-D [9] algorithm to a specific priority-based arbitration scheme which guarantees transmission of HRTMs under all load conditions and schedules SRTMs according to EDF in a best effort manner. The approach has been validated by simulations which proved that in worst-case settings, HRTMs are transmitted as long as respective reservations can be provided.

References

1. K. Tindell and A. Burns: „Guaranteed Message Latencies for Distributed Safety-Critical Hard Real-Time Control Networks“, *Report YCS-94-229, Department of Computer Science, University of York, May 1994*
2. H. Kopetz and G. Grünsteidl, "TTP - A Time-Triggered Protocol for Fault-Tolerant Real-Time Systems", *Research Report No. 12/92, Inst. für Techn. Informatik, Techn. University of Vienna, 1992*
3. Brian.P. Crow, Indra Widjaja, Jeung G. Kim, Prescott T. Sakai: "IEEE 802.11 Wireless Local Area Networks", *IEEE Communication Magazine, September 1997*
4. M. Mock, e. Nett: "Real-Time Communication in Autonomous Robot Systems", in: *Proc. of the 4th Int. Symp. on Autonomous Decentralized Systems, Tokyo, Japan, 1999*
5. H. Kopetz, M. Holzmann, W. Elmenreich: " A Universal Smart Transducer Interface: TTP/A", in: *Proc. of the 3rd Int. Symp. on Object-Oriented Real-Time Distributed Computing", Newport Beach, California, March 20006.*
6. R.M. Grow: " A timed token protocol for local area networks", in: *Proc. of Electro/82, Token Access Protocols, 1982*
7. S.T. Levi, S.K. Tripathi, S.D. Carson, A.K. Agrawala, "The MARUTI Hard Real-Time Operating System", *ACM Operating Systems Review 23(3):90-105, July 1989*
8. M.L. Molle and L. Kleinrock, "Virtual Time CSMA: Why two clocks are better than one", *IEEE Transactions on Communications 33(9), Sep. 1985.*
9. W. Zhao and K. Ramamritham, "Virtual Time CSMA Protocols for Hard Real Time Communcations", *IEEE Tr. Software Eng. 13(8), 1987*
10. M.A. Livani, J. Kaiser, W. Jia, "Scheduling Hard and Soft Real-Time Communication in the Controller Area Network (CAN)", *Control Engineering Practice, Vol. 7 Number 12 pp. 1515-1523, Dec. 1999.*
11. M.A. Livani and J. Kaiser, "Evaluation of a Hybrid Real-time Bus Scheduling Mechanism for CAN", *Lecture Notes in Computer Science 1586 (Jose Rolim et al. Eds.), pp. 425-429, Springer Verlag Berlin, 1999*
12. SAE, "Class C Application Requirement Considerations", *SAE Technical Report J2056/1, June 1993.*