The Event-Triggered and Time-Triggered Medium-Access Methods

Vilgot Claesson*
Dept. of Comp. Engg.
Chalmers Univ., Sweden

Cecilia Ekelin
Dept. of Comp. Engg.
Chalmers Univ., Sweden

Neeraj Suri
Dept. of Comp. Science
TU Darmstadt, Germany

Abstract

The processes of accessing a shared communication media have been extensively researched in the dependability and real-time area. For embedded systems, the primary approaches have revolved around the event-triggered and the time-triggered paradigms. In this paper, our goal is to objectively and quantitatively outline the capabilities and limitations of each of these paradigms. The event-triggered approach is commonly perceived as providing high flexibility, while the time-triggered approach is expected to provide for a higher degree of predictable communication access to the media. We have quantified the spread of their differences, and provide a summary discussion about suggested best usage for each approach. The focus of our work is on the response times of the communication system, and also on the schedulability of the communication system in collaboration with tasks in the nodes.

1. Introduction

The access to a shared communication media using demand/contention based access (Event-Triggered) and slotted controlled access (Time-Triggered) has garnered its share of strong opinions in the community.

The effects of choosing one design approach over another are not obvious especially in a complex system where the various dimensions of a design decisions cover issues of flexibility, efficiency, predictability and dependability. For example, if choosing the event-triggered system we might be prohibited from using the full bandwidth in order to ensure critical message transfers even in an eventful situation consequently leading to a low utilization of the media.

Thus, our specific objectives in this paper are: • We quantitatively establish the conditions under which Event-Triggered (ET) and Time-Triggered (TT) systems are more appropriate to use concerning the amount of transferred data and the response time characteristics. • We establish schedulability strengths and limitations of each approach. • Based on the above two facets, we outline suggested domains of strength and weakness for each paradigm along with suggested envelopes of operation.

Comparisons of ET and TT approaches exist, although these target specific protocol instantiations and for specific load conditions, e.g., [4] CAN-TTP and [3] CAN-QWIK. Work in [5] investigates min/max delays and generated jitter. Related work concerning communication and task scheduling appears in [6, 8]. However, our work compare the normal behavior of ET and TT architectures under varied load conditions with the intention of achieving better understanding of their fundamental capabilities and limitations.

2. System and Task Model

The system environment consists of n nodes interconnected via a multiple access communication channel. To avoid collisions, one node at a time can send while all nodes can receive the sent message. This requires an ordered access approach to avoid or resolve multiple simultaneous accesses.

Communication takes place between two tasks τ_i and τ_j. The message size is measured in bytes. Tasks are periodic and must execute within the time interval given by the period. The communicating tasks are assumed to be located on different nodes so the communication will take place over the communication media. In the schedule, it is assumed that a message must be sent after the completion of τ_i and received before the start of τ_j. Periodic messages do not have explicit deadlines but must be delivered in time for the receiver task to be able to meet its deadline. Message transmission is always non-preemptive.

2.1. Media-Access Methods

In the *Time-Triggered Architecture* (TTA), access to the media is ordered by time, such that each node is assigned a time-slot in a cyclic schedule. Sporadic messages must be sent in a slot that belongs to the node where the sender process executes.

*Email:{vilgotc,cekelin}@ce.chalmers.se, suri@informatik.tu-darmstadt.de.

Partly supported by the European project NextTTA, IST-2001-32111.

We assume that one frame, i.e., message data including overhead, is sent in each time slot. Frames are used as containers for both periodic and sporadic messages, and have one static part assigned for periodic messages transfer, i.e., the same message data is transferred periodically in that part. The second part of the frame is used for sporadic message transfer. As we do not know the arrival times for sporadic messages, we use a local node queue where all new sporadic data is queued on arrival. Messages are queued based on priorities. When it is time to send a frame, as many messages as possible are packed in the sporadic part of the frame.

To avoid media contention in the Event-Triggered Architecture (ETA) we assume that nodes can resolve potential collision by a priority on the frame. Thus, if two nodes start to send at the same time the node with the highest priority frame will send and the other node will withdraw, see for instance CAN [7]. We assume that periodic messages have the highest priority and sporadic messages the lower priority.

3. Simulation Setup

In this paper we present two investigations. a) The behavior of TT and ET communication under different communication loads, and b) How ET and TT systems handle different task sets regarding scheduling.

To obtain synthetic task sets and sets of sporadic messages, a uniform distribution was used to randomly generate tasks and sporadic messages in a two-step process: (1) a period was generated and (2) within each period an activation time of the task/message was generated. Thus, the generated period controls the basic rate of sporadic messages and tasks. Each message and task are assigned three properties (1) a period (2) a length and (3) a priority, i.e., a triplet $< period, length, priority >$.

The system is run with different communication load, consisting of periodic and sporadic messages, in order to compare the TT and ET approaches. Periodic messages are generated at a constant rate, while sporadic messages are generated with a varied rate. Thus, the amount of periodic messages is fixed and the communication load is determined by the amount of sporadic messages. Hence, the *generated sporadic traffic* is randomly generated to achieve a certain *load*, which is controlled by the message length, number of messages and their basic periods. We define *load* as: $Load = \frac{Generated\ sporadic\ traffic}{Available\ sporadic\ bandwidth}$

The available bandwidth is the amount of data that can be transferred per time unit on our media. Thus, the *Available sporadic bandwidth* is the residual bandwidth, after the bandwidth for periodic traffic has been excluded.

3.1. Task and Communication Scheduling

In TTA, periodic messages (and tasks) follow a predefined schedule and sporadic messages are included when there is slack available. Hence, sporadic messages will not affect the pre-scheduled tasks and messages or cause them to miss deadlines.

In ETA, the scheduling of tasks and messages is performed on-line. Tasks are normally preemptive, but messages sent on the media are not preemptive and can delay a higher priority message, which may lead to a missed message deadline or task deadline.

Scheduling In general, the media-access method reflects the intended use of the system. It would then seem natural that the motivation for choosing a particular communication approach is also applicable for how the tasks should execute, i.e., how to perform task scheduling. Hence, using TT communication, the schedule constitutes a static time-table dictating the start times for the tasks which are not affected by any sporadic messages. Similarly, the schedule for ETA is generated dynamically by considering task priorities.

Experimental Setup For this part of the simulations we have used a constraint programming framework that was developed previously in [2]. We generated three studies representing systems with 6, 12 and 18 nodes. In each study we generated task sets with varying sporadic communication load. The variation was obtained by successively increasing the message sizes while the number of messages remained constant. The deadline for a task equals its period. To get the load evenly distributed in the schedule, the tasks also have randomly generated activation times. For each experiment, 20 task sets were generated.

Experimental Results The first experiment investigates how well ETA and TTA handle sporadic messages. A message/task is regarded as missed, if it cannot be sent/executed in such a way that it is received/finished before its deadline. Figure 1 shows the number of missed messages for TTA and ETA. The figure also shows that ETA is able to accommodate a larger proportion of sporadic messages than TTA. However, as the load increases, the gap between ETA and TTA decreases. The explanation for this transition is that at low load, the ETA has the flexibility to use the slack for any node. This is not the case for the TTA where the number of missed messages grows almost linearly with the load. In contrast, when the load increases, ETA still tries to accommodate all messages before their deadline. However low priority messages can now easily be delayed by messages from all nodes, such that they miss their deadline. This is an effect which increases with load. This problem occurs in ETA as the communication queue is a *global* one. In

TTA this effect is limited as the communication queue is *local* and thus fewer messages are affected. As seen in Figure 1, the transition occurs at a load of approx. 0.6–0.7.

We do note for TTA that in our experiments the sizes of the messages are rather large compared to the size of the slots. This means that sporadic bandwidth may potentially be wasted, as many messages will not fit into the remainder of the designated slot. Hence, if we had varied the load by increasing the number of messages instead of increasing the message size, the performance of TTA would probably improve.

We have assumed small frame overhead for addressing in both ETA and TTA, i.e., approximately 1 and 2 byte for TTA and ETA, respectively. If we should include other overhead, e.g., for checksums etc., we would likely effect the ETA negatively, see [1].

Note, in Figure 1 the amount of missed messages in the ETA curves that corresponds to periodic messages, ranges between 5% and 15%. A missed periodic message also means a missed (receiver) task deadline.

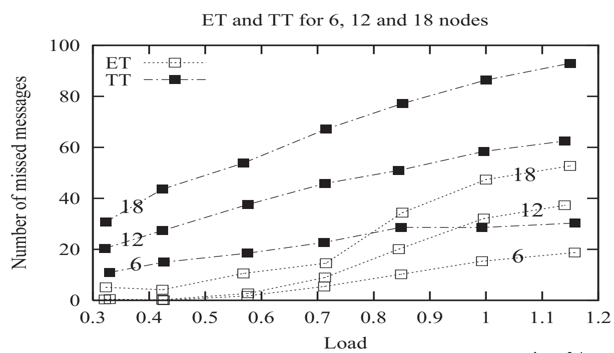

Figure 1: Event-Triggered and Time-Triggered (30% periodics, 30% task utilization).

3.2. Communication Load

In this section we present an investigation of the average delay for a message, from its release time until it is actually transferred. The system is only limited by the communication. The tool Matlab have been used for these simulations using the basic triplets $<$ *period*, *length*, *priority* $>$ as described in section 3.

As there is no natural termination in this type of simulation, we want to determine a time where the behavior of the system has stabilized. We have found, see [1], that the system is stable after 100 rounds. That is, the average delay is not affected even if the system is run for a longer time.

A First Comparison In Figure 2, we can see the results after simulations with six nodes, where the bandwidth is divided equally between periodic and sporadic messages. The left y-axis depicts the average delay of messages. The right y-axis shows the amount of messages not sent, i.e., instance i of message a (a_i), is deleted if a_{i+1} arrives before a_i.

It is important to note that in TTA, periodic messages will not contribute to the average delay or be deleted, as they are sent in predetermined time slots.

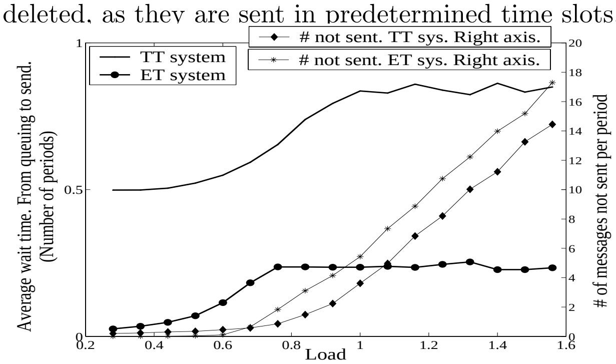

Figure 2: Average send delay of messages. 6 node systems. Note, for TT, periodic messages never contribute.

Number of Nodes In this section we have varied the number of nodes in the system and studied changes on the average message delay. We have changed the number of nodes between 6, 12, and 18. Thus in Figure 3 we show the trends when we increase the system size, both for TTA and ETA.

In Figure 3 we can see that the trend scales with the system size. The only changed parameter is the number of nodes, which contributes to the size of the TDMA-period (for TTA). Furthermore, we use a fixed average size on messages and this will affect the period for sporadic messages.

Thus, the TTA will in this case get larger delay as the system size increase, as the average delay is mainly dependent on the TDMA period.

The ETA has low average delays as long as the load is low. However, the delays will grow earlier, and at high load the average delays are still low for messages accessing the media.

The amount of missed messages is not affected by the system size. We have also studied the effect when varying the preassigned share of periodic messages [1].

4. Discussions and Conclusion

Prior to summarizing our result, note that the use of randomly generated messages are not adaptable to any real-life situation. However, similar loads can be found in many real-time systems, e.g., high priority control loops generating periodic messages and other system parts generating sporadic messages.

We argue that the ETA and TTA approaches have certain inherent properties that makes them more suited for a specific context. We summarize our observations that can aid proper solutions and designs of ETA and TTA.

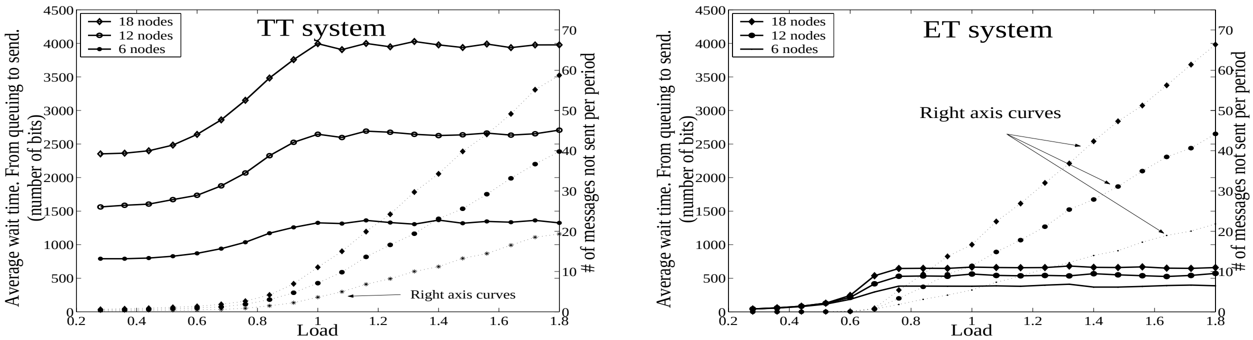

Figure 3: The number of nodes in the system is changed from 6,12 and 18 nodes.

• Missed/deleted messages increases when the load is greater than (approx. 0.7), as the lowest priority messages never get access to the media. Similarly, considering tasks and messages with their deadlines, there is a noticeable amount of missed deadlines already at a load of 0.6–0.7. This is confirmed by the pure communication investigation as the standard deviation starts to increase around the same load. Thus, the fact that there is a global queue together with big variations in delay, larger message overhead and more messages cause a lot of missed deadlines in the ETA.

• For TTA, the average delay is about half a TDMA-round (large compared to ETA). Note, that in TTA periodic messages have zero delay and only sporadic messages contribute to the average delay. There is also a higher amount of lost messages compared to the ETA when tasks are involved, i.e., when short deadlines are used for the sporadic messages. In the pure communication case, short deadlines are not used and we can see that fewer messages are missed for the TTA, at higher load, than the ETA. Thus, short deadlines on sporadic messages negatively effects TTA. With less short deadlines of sporadic messages, it is noticeable that this very basic approach of transferring sporadic messages still handles sporadic traffic surprisingly efficiently, in the sense that few messages are missed/deleted. Thus, the predictability of a TTA comes at a cost of longer delays, but with less variation in these delays. However, when sporadic message deadlines are relatively long, there is a smaller loss in bandwidth efficiency compared to the ETA. This basic approach of sending sporadic messages works with very small or no changes in any TTA. In our future work we will look at more efficient ways of transferring sporadic and event driven data using TTA.

• As long as a system runs with a load below 0.6–0.7, there is little reason for choosing a TTA. The ETA implies short message delays and most messages are transferred. Even variations between delays are small with a fewer number of missed deadlines below this threshold. The salient reason for choosing a TTA is if we have high requirements on predictability, e.g., dependability. The sporadic traffic handling is one of the main contributing facts to why ETA is considered flexible. However, the TTA can also handle sporadic messages well, as it is not so sensitive to traffic-bursts and benefits from the fact that it has less overhead than the ETA. As we have seen in the figures, both TTA and ETA have similar amount of lost messages. Thus, when considering flexibility of the ETA we would rather point to the ease of integrating new nodes etc.

• When varying load/bursts traffic or high load is expected, i.e., the load often above 0.6–0.7, a TTA should be considered, especially for real-time systems. However, based on our result we can also state that the only reason for chosing the ETA below the 0.6 is if there is need for small average delays of sporadic messages.

References

[1] V. Claesson, C. Ekelin, and N. Suri. The event-triggered and time-triggered medium-access methods. TR-03-02, Dept. of Comp. Engg., Chalmers Univ. of Tech., 2003.

[2] C. Ekelin and J. Jonsson. Evaluation of search heuristics for embedded system scheduling problems. In *Proc. of the Int. Conf. on Principles and Practice of Constraint Programming*, Paphos, Cyprus, 2001.

[3] L.-Å. Johansson, H. Larsson, and S. Lundgren. QWIK vs. CAN-efficiency comparison. TR, Qrtech AB, http://www.qrtech.se/Downloads/, 19999.

[4] H. Kopetz. A comparison of CAN and TTP. *15th IFAC Workshop on Dist. Comp. Control Sys.*, Italy, 1998.

[5] H. Lönn and J. Axelsson. A comparison of fixed-priority and static cyclic scheduling for distributed automotive control applications. In *1th Euromicro Conf. on Real-Time Systems*, York, 1999.

[6] T. Pop, P. Eles, and Z. Peng. Holistic scheduling and analysis of mixed time/event-triggered distributed embedded systems. *Tenth Int. Symp. on Hardware/Software Codesign.*, Estes Park, CO, USA, 2002.

[7] Robert Bosch GmbH. *CAN Spec. V. 2.0*, 1991.

[8] K. W. Tindell and J. Clark. Holistic schedulability analysis for distributed hard real-time systems. *Microprocessing and Microprogramming*, 40(2-3):117–134, 1994.