Preprints of the 30th IFAC Workshop on Real-Time Programming and
4th International Workshop on Real-Time Software pp. 143–148

Communication Protocols for Real-Time Systems: Time vs. Events Revisited

Dirk Müller * **Matthias Werner** *

* *Operating Systems Group, Chemnitz University of Technology,
Chemnitz, Germany (e-mail: {dirkm,mwerner}@cs.tu-chemnitz.de).*

Abstract: The sharing of a medium for communication between nodes, forming a real network, is much more cost-effective than building a fully connected mesh. Due to the then possible collisions, special techniques have to be applied. Historically, the two main schemes to handle collisions are the *time-triggered* (TT) and the *event-triggered* (ET) approach. While the former one prevents messages from collisions by time sharing, the latter one considers collisions as events and tries to resolve them. For a long time, it has been believed that time-triggered techniques are contradicting event-triggered ones and only time-triggered approaches can provide real-time communication with hard deadlines. Both statements are not true. There are several protocols like TTCAN, TTEthernet and FlexRay combining both approaches. Well-established event-triggered protocols like CAN are capable of real-time communication to a certain degree.
In this article, we provide a survey to systematize on TT and ET in the context of real-time (RT) communication systems. We discuss a wide range of approaches and try to identify relations that cross the border between TT and ET.

Keywords: Real-time communication, Deadlines, Event-triggered, Time-triggered, Time-shared control, Master-slave systems, CAN, Ethernet, Token-ring protocol

1. INTRODUCTION

As a matter of principle, there are two ways to manage network access of nodes. It is an issue to *resolve* conflicts between nodes which try to get access to the network at the same time, or to *ensure* by a pre-defined timing that such conflict situations cannot occur. The first solution can be seen as an optimistic approach and is called *event-triggered*. The second one is pessimistic and was coined *time-triggered*.

For a long time, time-triggered (TT) and event-triggered (ET) approaches in order to achieve real-time communication in networks of distributed embedded systems have been seen as *concurrent* proceedings. At least, there could be observed a heated debate on this matter, which started with the famous paper by Kopetz (1991). However, both approaches have their strengths and weaknesses. Thus, it seems to be very reasonable to integrate them. In the automotive sector, this is exactly the way gone. I.e., TTCAN, ByteFlight and FlexRay, are the most successful ones, suggesting a confirmation of the theoretically anticipated synthesis.

In this paper, we provide a survey on TT and ET approaches in the area of real-time (RT) communication systems. We will show that there the relation of ET and TT are closer than frequently implied. We consider a wide range of communication approaches, even few that were not intended to be used in RT systems in the first run. Figure 1 shows the approach we considered in relations to data rate and intended RT behavior.

The remain of this paper is the following: After a short discussion of connection-oriented protocols in Sections 2 we consider in deep ET and TT approaches in Sections 3 and 4. Then, we identify and discuss approaches that integrate both, ET and TT aspects in Section 5. Finally, we conclude the paper with a discussion of our observations.

2. CONNECTION-ORIENTED PROTOCOLS

The simplest way to manage network access is to establish connections between two neighboring nodes and to route the messages to their respective destinations. These *connection-oriented protocols* were dominating before the advent of LANs. While they exhibit a deterministic behavior between directly connected nodes, latency of messages sent between arbitrary nodes can be high because processing is necessary in the routers. Collisions are resolved by sequentialization immanent to the nodes. All other here presented methods are connectionless protocols allowing multiple nodes connected to one and the same physical transmission medium. Thus, this medium is *shared*, and special care has to be taken to manage the now potentially possible collisions in the medium. Nowadays, only these types are classified as networks.

3. EVENT-TRIGGERED APPROACHES

A first approach to deal with possible collisions is to consider them as *events*. Then, there is a need for an appropriate reaction to them (ALOHA, CSMA/CD), a resolution by prioritization (CSMA/CR) or an avoidance to a large degree (CSMA/CA). These variants will be described in the following.

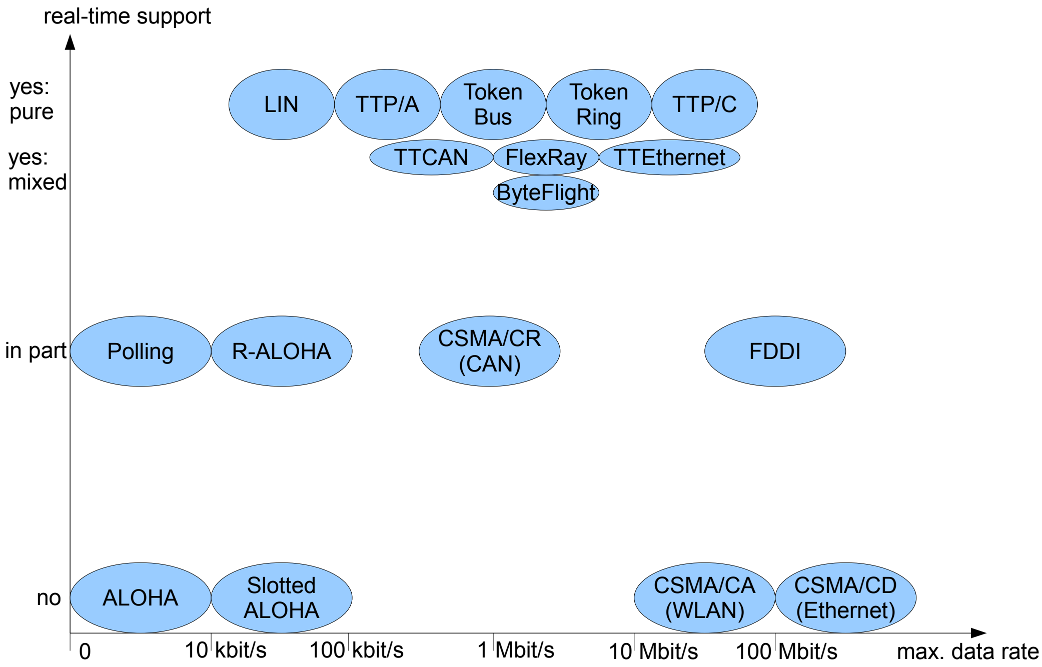

Fig. 1. Communication protocols for real-time systems: real-time and data rate capabilities

3.1 ALOHA

In its highest consistency, the optimistic principle event-triggered approaches are characterized by leads to a simple protocol first applied for connecting the Hawaiian islands, see Binder et al. (1975). A node just tries to send the message in the hope that there won't be any collision. Of course, collisions can occur. The nodes test the consistency of each message with a checksum. In case of a collision, a resending of the message which collided is initiated after a random wait. It has been shown that only 18% of the maximum channel data rate can be used by this protocol. This percentage is the maximum throughput.

Note the analogy to *non-blocking synchronization*, see, e.g., Greenwald and Cheriton (1996), and Tsigas and Zhang (1999). In databases, *optimistic concurrency control (OCC)*, see Kung and Robinson (1981), is closely related. The reason that the optimistic principle has less success in the network field than in the synchronization or database field is that periods of time with collision risk are longer due to the inclusion of transmission times. Improvements of ALOHA, including time-triggered elements, are given in 5.1 and 5.2.

3.2 CSMA/CD

If a node not only sends the message, but additionally listens to the network in order to get some feedback on the situation there, we call this *Carrier Sense Multiple Access (CSMA)*. In the case of collision, a first approach is to detect it, to communicate this to the other nodes by sending a jamming signal, and to retry after a random time in the hope that the randomly chosen pause lengths differ. Otherwise, endless repeating collisions would occur. We call this *collision detection (CD)*. The principle is implemented in the famous *Ethernet*. A specialty of it is the *truncated binary exponential backoff algorithm* which is intended to provide a good chance of a quick *random* prioritization. Thus, pure Ethernet is not suitable at all for real-time messages. On the other hand, data transmission rates up to 10 Gbit/s are possible.

3.3 CSMA/CA

In wireless local area networks (WLANs), it is impossible to send and to listen in parallel. Hence, CSMA/CD is not an option. In order to avoid as much collisions as possible, a less greedy behavior in terms of medium access should be used. A node listens to the medium prior to sending. If the medium is busy, the transmission is deferred for a random interval. This approach called *collision avoidance (CA)* cannot provide any real-time guarantees. Data transmission rates of 54 Mbit/s are achieved.

3.4 CSMA/CR

The requirements in car electronics in the 1980s lead to the development of a real-time capable CSMA approach. The Controller-Area Network (CAN) strives for *collision resolution (CR)*. It uses a non-destructive bitwise arbitration where 0 is the dominant bit value. There is an analogy to fixed-priority scheduling of jobs with access to a common resource. This leads to a blocking by lower priority jobs. They cannot be preempted since first a transmission of a message must be completed. Hence, the concept behind is non-preemptive fixed-priority scheduling.

Up to the mid 1990s, it has been believed that real-time guarantees are only possible for the highest priority message. At that time, this changed since Tindell et al. (1995) provided a message response time analysis. Later on, Davis and Burns (2007) showed that this analysis contains a flaw, making it too optimistic. On the other hand, it could be pointed that the older analysis works consistently if there is a single maximum length diagnostic message omitted. Since CAN design includes such rarely sent diagnostic messages in practice, real-time guarantees can be determined by the optimistic analysis in these cases, too.

So, CAN enables for real-time communication. Nevertheless, a blocking of lower priority messages by a burst of high priority messages is possible. The problem is solved by restricting the frequency of messages which corresponds to the periodic task model. In time-triggered approaches, the frequency of real-time messages is limited due to the time slots designed before. This is a matter of bandwidth, a technical limit. Maximum data transmission rate is 1 Mbit/s.

Up to now, we were assuming the absence of faults. CAN has several fault-tolerant mechanisms. A problem remains with the *babbling-idiot fault*, see, e.g., Buja et al. (2007). This type of fault leads in the worst case to a blocking of the entire system. It could happen if a faulty node sent useless data in high-priority messages at tightly packed faulty timings. It is important to distinguish between hardware and software babbling-idiot faults. The former ones can be fought by node or bus replication (see, e.g., FDDI in 4.6), the latter ones can be attacked by a *bus guardian*. A bus guardian is easy to implement in (F)TDMA approaches as done in TTP/C (4.4), ByteFlight (5.5) and FlexRay (5.6). It is much harder to provide it in event-triggered approaches like CAN, see Broster and Burns (2001). The during normal operation advantageous property of a flexible timing turns into a problem for the babbling-idiot fault. So, more complex (sub-)protocols are required to eliminate a great deal of babbling-idiot faults.

4. TIME-TRIGGERED APPROACHES

Opposed to the optimism of event-triggered approaches, time-triggered approaches are pessimistic as they use *time slots* at least for important (probably real-time) messages. The time slots can be generated by a central master (polling and TTP/A), by synchronized clocks at all nodes (TTP/C), or by a token passed from node to node (Token Ring and Token Bus). These protocols will be described subsequently.

4.1 Polling

Polling is the simplest and oldest protocol among connectionless ones. A centralized approach with a master and several slaves is realized. There, the slaves are asked by the master for communication in a predefined sequence. A problem is the only indirect support of slave-to-slave traffic and the high bandwidth consumption even at lower loads. The overhead of the polling messages is significant. Another problem –as with all centralized architectures– is the master as a Single Point of Failure. Redundant master hardware is an option, but an expensive one. Real-time capabilities are offered to a certain degree, but nowadays, there are more sophisticated derivatives which remedy the weaknesses partially.

4.2 LIN

The Local Interconnect Network (LIN), see DeNuto et al. (2001), uses a centralized architecture based on a master and several slaves. The master manages communication including sending timing data for synchronization purpose to the slaves. It supports real-time messages. On the other hand, data transmission is limited to 20 kbit/s and, thus, slowly. The master is a Single Point of Failure.

4.3 TTP/A

The Time-Triggered Protocol Class A (TTP/A), see Kopetz et al. (2000), is based on a centralized architecture with Time Division Multiple Access (TDMA). It supports real-time messages. The time slots in which messages are exchanged between the master and a slave are predefined in a so-called Round Description List (RODL). A redundant design with a backup master is used. So, there is no longer a Single Point of Failure. Data rates up to 500 kbit/s are supported.

4.4 TTP/C

The Time-Triggered Protocol Class C (TTP/C) uses a decentralized (including synchronization) TDMA architecture, see Kopetz and Grünsteidl (1993). Message sending and receiving periods are predefined in a so-called Message Descriptor List (MEDL). Data transfer rates up to 25 Mbit/s are supported. Real-time traffic is possible. A problem common to TTP/A and TTP/C is the lack in flexibility. Future modifications in the set of nodes are hard since they require a complete redesign.

A good feature in terms of reliability is the bus guardian protecting the system from babbling-idiot faults, see also 3.4. Only during proper operation, a node is granted network access. Hence, problems are isolated, avoiding an affection of the entire network. Additionally, a membership protocol even enables the detection of faults not included in the error model. Faulty cliques are avoided by periodic checks for membership in the overall clique. Nodes not belonging to it will be switched into passive mode.

4.5 Token Ring

In a Token Ring network, see Bux (1989), the functionality of the master is transferred to a token rotating in a physical ring structure of all nodes. Overhead is relatively small both under light and heavy traffic. Token Ring allows for real-time messages since it is deterministic. A problem is the lack of robustness. A single cut in a cable or a node crash makes the entire network to crash. This can only be solved by an expensive redundancy with node bypasses and a second ring. Token Ring networks allow for data rates up to 16 Mbit/s.

4.6 FDDI

A doubling of the ring as mentioned in 4.5 is implemented in the Fiber Distributed Data Interface (FDDI). There, two counter-directed rings are reconfigured to a single ring in case of a cable cut. FDDI has been designed for optical fiber as the underlying medium. It became a standard in 1989 and supports 100 Mbit/s. With some additional effort, it is capable of real-time traffic, see Zheng (1993).

4.7 Token Bus

Token Bus systems use a logical ring on a physical bus structure. It is deterministic as well, but slower than Token Ring since there is no parallelism in the connections. On the other hand, hardware failures do not make the entire network to crash. Additionally, adding or removing nodes is easier. Data rates up to 12 Mbit/s are possible.

5. HYBRID APPROACHES

5.1 Slotted ALOHA

Slotted ALOHA improves pure ALOHA by introducing time slots. The basic idea is borrowed from Time Division Multiple Access (TDMA) approaches. By such a discretization, the chance of overlapping reduces significantly, and maximal throughput is increased from about 18% to about 37%, see Binder et al. (1975). On the other hand, real-time traffic remains impossible. Note further that a clock synchronization in order to establish the slots is necessary.

5.2 Reservation ALOHA

A further improvement can be achieved by applying an implicit or explicit reservation scheme for the slots. So, we obtain Reservation ALOHA, see Crowther et al. (1973). R-ALOHA allows for real-time messages to a certain degree. In particular, it is useful for multimedia data since the reservation of future slots can be based on the streamed nature of this type of data. A problem is that a node can permanently block all other nodes by implicitly reserving all future slots for itself. This can be circumvented by explicit reservation rules interdicting such an impolite behavior.

5.3 TTEthernet

TTEthernet, see Plankensteiner (2008), enables real-time messages by placing a layer on top of Ethernet. Messages are categorized into time-triggered, rate-constrained and best-effort messages. Time-triggered ones can be delivered in a real-time manner. Data rates reach already 1 Gbit/s, see TTTech Computertechnik AG (2009).

For a more general view on different protocols on top of Ethernet, see, e.g., Felser (2005). The EtherCAT, see, e.g., Jansen and Buttner (2004), abandons software protocol stacks and implements them completely in hardware. Although this is much quicker on average, the here highlighted aspect is the colloquial meaning of real-time: fast. It has to be stressed again that real-time means the compliance with deadlines and not necessarily quick processing or transmission. This becomes apparent in the two independent axes in Figure 1 which will be discussed in Section 6 in more detail.

5.4 TTCAN

TTCAN, see Saha and Roy (2007), uses a layer on top of CAN. Again, messages are categorized into groups. Here, they are called exclusive and arbitration messages which are sent in the respectively named windows (slots). The slots are established by a time basis which is constructed either by a time master (extension level 1) or a distributed approach (extension level 2). So, for exclusive messages real-time service is possible. Data rates up to 1 Mbit/s are supported.

5.5 ByteFlight

ByteFlight, see Berwanger et al. (2000), can be seen as a combination of TTP/C and CSMA/CR. It uses advantages of both approaches. The application of flexible TDMA (FTDMA) techniques makes such an integration possible. There are data rates up to 10 Mbit/s possible. For higher priority messages, real-time behavior can be guaranteed.

5.6 FlexRay

FlexRay, see Pop et al. (2008), is one of the most sophisticated known media access protocols. It is based on ByteFlight and improves it further by using a distributed clock synchronization as TTP/C does. So, it is more fault-tolerant since there is no longer a synchronization master as a Single Point of Failure. Data rates of up to 10 Mbit/s can be achieved. Real-time messages are supported.

6. DISCUSSION

In Figure 2 we show the relations of the discussed approaches and protocols. The arrows in the graph signify an *is-basis-for* relationship. Note that the connection from TTP/C to R-ALOHA is dashed since R-ALOHA uses TDMA principles, but is not directly based on TTP/C.

If one compares these relations with the classification with respect to intended RT behavior and data rate as provided in Figure 1 in Section 1, there is no clear correlation between both diagrams. Although more of the non-RT approaches are on the TT side, there is no deterministic contingence between real-time and time-triggered concepts.

In general, event-triggered approaches allow for shorter average latencies than time-triggered ones. Additionally, systems based on them are easier to build and–in particular–to extend. Typically, new nodes can be integrated easily. On the other hand, maximum latency in event-triggered systems is higher than in time-triggered configurations or just non-existent. Thus, in safety critical embedded systems as in means of transport (car, train, airplane, etc.) and in medical or industrial environments, time-triggered systems with real-time capabilities are preferable. During the last years, new protocols combining the advantages of both –at first glance conflictive– approaches have been developed. As in the case of the state-of-the-art FlexRay protocol, this shows that flexibility can be combined with deadline guarantees.

On the other hand, it would be naïve to see FlexRay as the ultimate solution. Of course, a lot of older, less sophisticated protocols like LIN, CAN, Ethernet and others are still in use for good reasons. These reasons are costs and the reputation of older technologies. The latter is clearly visible in the attempts to improve widespread protocols by

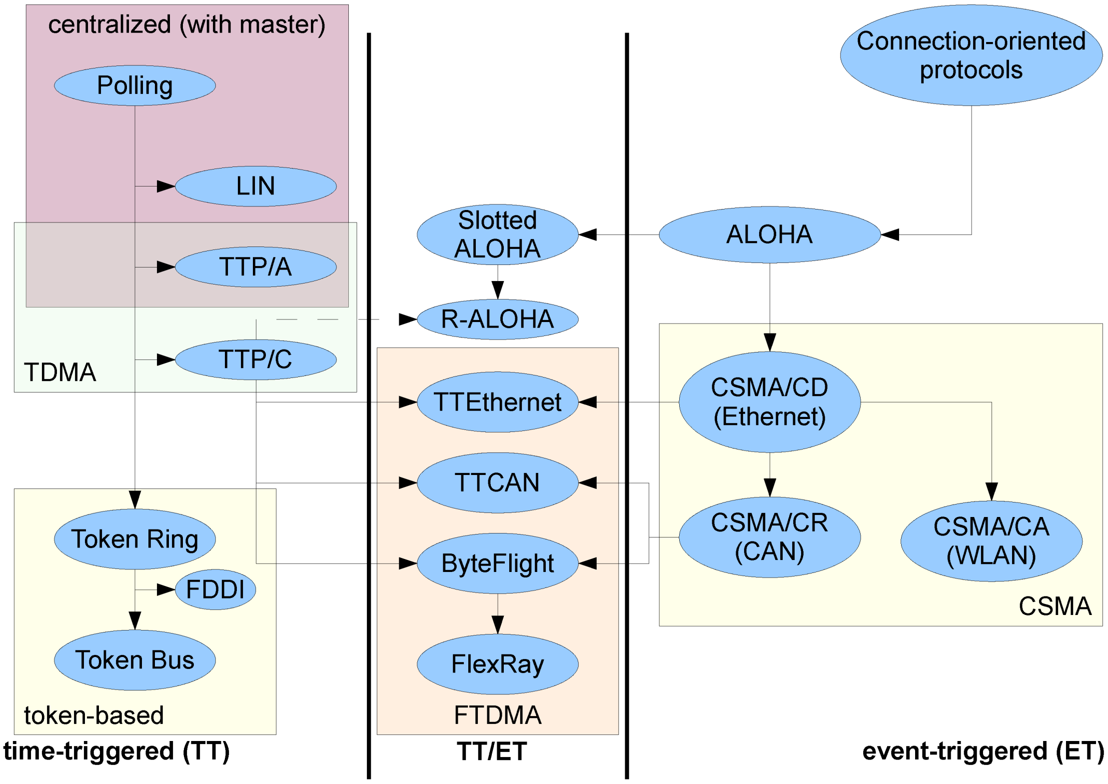

Fig. 2. Communication protocols for real-time systems: relationships

extensions, keeping them compatible to existing systems. Examples are TTEthernet and TTCAN.

As a rule of thumb, time-triggered systems are better in real-time guarantees, and event-triggered systems are better in medium utilization and flexibility. Kopetz (2001) argues, that there is always a trade-off between safety and flexibility. That might be true for a number of applications. However, there are applications where flexibility provides the means to improve safety.

Also, combinations of both approaches are possible. Then, systems providing both the advantages are obtained. From a technical point of view, such systems might the best ones. On the other hand, in practice, we have to leave the ivory tower and costs come into play. Thus, a cost-benefit analysis is important in order to find the appropriate system for your application.

However, the success of combined approaches as TTEthernet or FlexRay shows that the TT vs. ET wars belong to the past.

REFERENCES

Berwanger, J., Peller, M., and Griessbach, R. (2000). byteflight—a new protocol for safety critical applications. Seoul 2000 FISITA World Automotive Congress.

Binder, R., Abramson, N., Kuo, F., Okinaka, A., and Wax, D. (1975). ALOHA packet broadcasting: a retrospect. In *AFIPS '75: Proceedings of the May 19-22, 1975, national computer conference and exposition*, 203–215. ACM, New York, NY, USA. doi: http://doi.acm.org/10.1145/1499949.1499985.

Broster, I. and Burns, A. (2001). The babbling idiot in event-triggered real-time systems. In G. Fohler (ed.), *Proceedings of the Work-In-Progress Session, 22nd IEEE Real-Time Systems Symposium, YCS 337*, 25–28. IEEE, Department of Computer Science, University of York.

Buja, G., Pimentel, J.R., and Zuccollo, A. (2007). Overcoming babbling-idiot failures in CAN networks: A simple and effective bus guardian solution for the FlexCAN architecture. *IEEE Trans. Industrial Informatics*, 3(3), 225–233. doi: http://dx.doi.org/10.1109/TII.2007.904144.

Bux, W. (1989). Token-ring local-area networks and their performance. *Proceedings of the IEEE*, 77(2), 238–256. doi:http://dx.doi.org/10.1109/5.18625.

Crowther, W.R. et al. (1973). A system for broadcast communication: Reservation-ALOHA. In *6th Hawaii International Conference system science*, 371–374.

Davis, R.I. and Burns, A. (2007). Controller area network (CAN) schedulability analysis: Refuted, revisited and revised. *Refuted, Revisited and Revisedř. Real-Time Systems*, 35, 239–272.

DeNuto, J.V., Ewbank, S., Kleja, F., Lupini, C.A., and Perisho, R.A. (2001). LIN bus and its potential for use in distributed multiplex applications. Technical report.

Felser, M. (2005). Real-time ethernet - industry prospective. *Proceedings of the IEEE*, 93(6), 1118–1129. doi: http://dx.doi.org/10.1109/JPROC.2005.849720.

Greenwald, M. and Cheriton, D. (1996). The synergy between non-blocking synchronization and operating system structure. *SIGOPS Oper. Syst. Rev.*, 30(SI), 123–136. doi:http://doi.acm.org/10.1145/248155.238767.

Jansen, D. and Buttner, H. (2004). Real-time ethernet the EtherCAT solution. *Computing & Control Engineering Journal*, 15(1), 16–21.

Kopetz, H., Holzmann, M., and Elmenreich, W. (2000). A universal smart transducer interface:

TTP/A. In *Object-Oriented Real-Time Distributed Computing, 2000. (ISORC 2000) Proceedings. Third IEEE International Symposium on*, 16–23. doi: http://dx.doi.org/10.1109/ISORC.2000.839507.

Kopetz, H. (1991). Event-triggered versus time-triggered real-time systems. In *Proceedings of the International Workshop on Operating Systems of the 90s and Beyond*, 87–101. Springer-Verlag, London, UK.

Kopetz, H. (2001). A comparison of TTP/C and FlexRay. Research Report 10/2001, Technische Universität Wien, Institut für Technische Informatik, Treitlstr. 1-3/182-1, 1040 Vienna, Austria.

Kopetz, H. and Grünsteidl, G. (1993). TTP - a time-triggered protocol for fault-tolerant real-time systems. In *FTCS*, 524–533.

Kung, H.T. and Robinson, J.T. (1981). On optimistic methods for concurrency control. *ACM Trans. Database Syst.*, 6(2), 213–226. doi: http://doi.acm.org/10.1145/319566.319567.

Plankensteiner, M. (2008). Standards-based realtime ethernet: New entrant proposes new protocol. *Industrial Ethernet Book*, (47). URL `http://ethernet.industrial-networking.com/articles/articledisplay.asp?id=2272`.

Pop, T., Pop, P., Eles, P., Peng, Z., and Andrei, A. (2008). Timing analysis of the flexray communication protocol. *Real-Time Syst.*, 39(1-3), 205–235. doi: http://dx.doi.org/10.1007/s11241-007-9040-3.

Saha, I. and Roy, S. (2007). A finite state analysis of time-triggered can (TTCAN) protocol using SPIN. In *ICCTA '07: Proceedings of the International Conference on Computing: Theory and Applications*, 77–81. IEEE Computer Society, Washington, DC, USA. doi: http://dx.doi.org/10.1109/ICCTA.2007.4.

Tindell, K., Burns, A., and Wellings, A. (1995). Calculating controller area network (CAN) message response times. *Control Engineering Practice*, 3, 1163–1169.

Tsigas, P. and Zhang, Y. (1999). Non-blocking data sharing in multiprocessor real-time systems. In *Real-Time Computing Systems and Applications, 1999. RTCSA '99. Sixth International Conference on*, 247–254. doi: http://dx.doi.org/10.1109/RTCSA.1999.811240.

TTTech Computertechnik AG (2009). Evaluation system with 1 Gbit/s for TTEthernet. URL `http://www.ttagroup.org/ttethernet/doc/TTTech-TTE-Evaluation-System-1Gbps.pdf`.

Zheng, Q. (1993). Synchronous bandwidth allocation in FDDI networks. In *In Proceedings of ACM Multimedia'93*, 31–38.