Fault-Tolerant Time-Triggered Ethernet Configuration with Star Topology

Astrit Ademaj Hermann Kopetz Petr Grillinger Klaus Steinhammer Alexander Hanzlik
Vienna University of Technology
Real-Time Systems Group
Treitlstr. 3/182-1, A-1040 Vienna, Austria
email:{ademaj,hk,grilling,klaus,hanzlik}@vmars.tuwien.ac.at

Abstract: We have shown in our past work that the standard configuration of Time-Triggered (TT) Ethernet unifies real-time and non-real-time traffic within a single coherent communication architecture. In this paper we present the design of a fault-tolerant configuration of time-triggered Ethernet, which unifies the real-time traffic of fault-tolerant applications as well as the real-time and non real-time traffic of non fault-tolerant applications into the same communication platform. TT Ethernet guaranties the temporal properties of the real-time traffic and prevents error propagation in the presence of component and communication faults. The advantage of this solution is that Ethernet traffic is handled in conformance with the existing Ethernet standards of the IEEE. The design of fault-tolerant time-triggered Ethernet has been driven by the requirement of certification of safety-critical configurations and by an uncompromising stand with respect to the integration of legacy applications and legacy Ethernet hardware.

1 Introduction

During the past decades, Ethernet has established itself as the most successful local area network of the world. Due to its open nature it is difficult to guarantee strict temporal properties in standard Ethernet-based systems. Therefore many researchers have looked into the problem of extending Ethernet such that it can be deployed in applications where temporal properties are important and the system is able to operate in the presence of faults.

Deterministic component operation contributes on building up systems with temporal guarantees, and fault tolerance can be achieved by component replication. Deterministic operation is also a precondition for certification of safety-critical systems [ARI04]. Fault-tolerant systems like drive-by-wire and fly-by-wire require a communication subsystem that provides deterministic and fault-tolerant data transmission with specified bandwidth requirements. The comfort electronics in vehicles requires a communication system with high bandwidth and flexible communication scheduling to achieve effective utilization of communication resources. There is an ongoing trend in the industry for building up systems that integrate predictable message transfer for critical functions and flexible message

transfer for other non-critical system functions. It must be guaranteed by design that changes can be made in one part of the system without having to redesign the whole system.

This paper presents the design of a communication system that tries to achieve this ambitious goal. We present a novel communication system, denoted as fault-tolerant time-triggered Ethernet, that allows the integration of different communication traffic categories, the real-time traffic and the non-real-time traffic in one core communication architecture by providing: i) predictable transmission for real-time traffic, ii) flexible communication schedule for non real-time traffic, iii) global time base, iv) fault-isolation, and v) diagnosis service.

2 TT Ethernet Overview

Time-Triggered (TT) Ethernet is a communication system that supports distributed non-real-time and real-time applications. The TT Ethernet design allows the existing Ethernet applications to be ported to TT Ethernet without any modification in software and hardware. We distinguish between two classes of configurations:

Standard TT Ethernet configuration can be used for real-time applications that require guaranteed message transmission delays, for example multi-media streaming applications.

Fault-tolerant TT Ethernet configuration for safety-critical real-time control applications that require predictable transmission delays and that shall tolerate component failures.

A standard TT Ethernet system consists of a set of *computer nodes* that are connected to a *TT Ethernet switch*. A computer node consists of a *host computer* and of a *communication controller*. The communication controller can be either an Ethernet controller or a TT Ethernet controller that is connected to the TT Ethernet switch by a bidirectional point-to-point link (see Figure 1). The TT Ethernet distinguishes between the following two message categories:

Time-Triggered Ethernet messages (*TT messages*) are sent periodically by a set of cooperating nodes according to an a priori established conflict free communication schedule that is coordinated by the progression of time.

Standard Event-Triggered Ethernet messages (*ET messages*) are handled by the network according to the *store-and-forward paradigm* of the standardized switched Ethernet.

The TT Ethernet switch handles TT messages according to the *cut-through paradigm*, and ET messages according to the *store and forward paradigm*. In case of a conflict between an ET message and a TT message, the ET message is preempted, so that the TT message can be transported by the switch within an *a priori known* constant delay to its receivers. After the transmission of the TT message is finished, the switch autonomously retransmits

the previously preempted ET message out of its memory. The communication schedule, which is created off-line, must avoid conflicts between TT messages.Conflicts among ET message are handled like in standard Ethernet switches. The distinction between an ET Ethernet message and a TT Ethernet message is made on the basis of the contents of the *frame type* fi eld in the header of each message that is defi ned in the Ethernet standard. The Ethernet standard authority of the IEEE [Eth05] has assigned the pattern 0x88d7 for the identifi cation of TT Ethernet messages. More detailed information about the operation of the standard TT Ethernet confi guration can be found in [KAGS05].

3 Fault-Tolerant TT Ethernet

In a safety-critical application the communication system must provide its services in the presence of component and communication channel failures. In this paper we present the design of the fault-tolerant confi guration of TT Ethernet, which can provide communication services and prevent error propagation from components that can fail in an arbitrary way. In order to provide these services, the fault-tolerant TT Ethernet confi guration deploys two independent communication channels by using two independent TT Ethernet switches. The input and output ports of each TT Ethernet switch are controlled by a separate unit called *guardian*. A fault-tolerant TT Ethernet confi guration deploys specifi c TT Ethernet controllers, which can send and receive messages using two bidirectional Ethernet compatible ports.

3.1 The Safety-Critical Time-Triggered Ethernet Controller

In order to tolerate communication channel faults, a safety-critical TT Ethernet controller transmits and receives redundant messages using two communication channels. Messages that are transmitted redundantly on two communication channels are denoted as *protected TT messages (PTTM)* because transmissions of these messages are monitored and controlled by guardians.

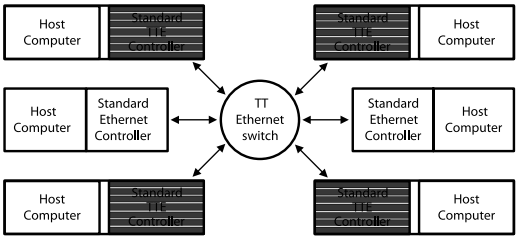

Abbildung 1: Standard TT Ethernet configuration

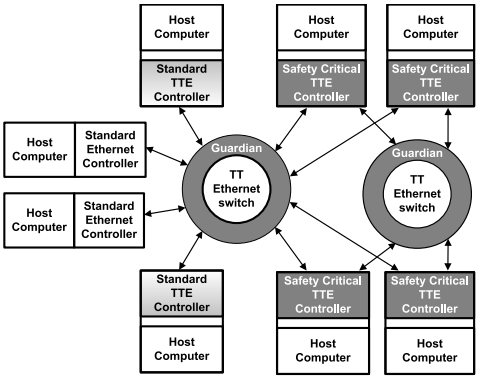

Abbildung 2: Fault-Tolerant TT Ethernet

3.2 The Guardian

The TT Ethernet switch in a fault-tolerant TT Ethernet confi guration is similar to the TT Ethernet switch in standard TT Ethernet, but the input and output ports of each switch are monitored and controlled by its guardian. The connection between the guardian and the switch consists of an Ethernet port and a set of parallel wires that allow the guardian to observe and disable the inputs and outputs of the switch. The guardian has its own fault-tolerant clock synchronization sub-system. It receives all TT messages from the switch over the Ethernet port but is allowed to send only ET messages to the switch via this port. An example of an ET message from the guardian is a diagnostic message informing about the behavior of the controllers. The switch will not accept TT messages generated by the guardian. The guardian has knowledge about the schedule of *protected TT Ethernet messages* (PTTM) and disables the inputs of the point-to-point links to controllers that could possibly interfere with the PTTM schedule.

Furthermore, the guardian will control how the switch delays an outgoing TT message, such that the instant of the start of frame transmission is *reshaped*. This is done in order to avoid the occurrence of a *Slightly-Off-Specification* (SOS) failure with respect to the start of frame transmission time [Ade02]. Similar functionality is implemented in the central guardian [BKS03] for the time-triggered architecture (TTA).

3.3 Time-Triggered Ethernet Node

A TT Ethernet node has a structure as presented in Figure 3. It consists of (i) host computer, (ii) HAL–the hardware abstraction layer, (iii) CNI–the communication network interface, (iv) TT Ethernet controller. The TT Ethernet communication protocol is executed in the TT Ethernet controller, whereas an application is executed at the host computer. The CNI is a data exchange interface between the host computer and the TT Ethernet controller. The TT Ethernet controller can either be a standard TT Ethernet controller or a safety-critical

TT Ethernet controller. Different implementations of the TT Ethernet controller may have different hardware interfaces to the host computer. The hardware abstraction layer (HAL) hides these hardware interface details, and therefore different controller implementations will not affect the application.

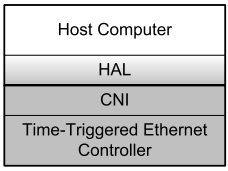

Abbildung 3: TT Ethernet Node Structure

3.4 Global Time

The global time in TT Ethernet is based on the *Uniform Time Format* (UTF). This time-format has been standardized by the OMG in the small transducer interface standard [OMG02]. A digital time format can be characterized by two parameters: *granularity* and *horizon*. The *granularity* determines the interval between two adjacent ticks of a digital clock, i.e., the smallest interval that can be measured with this time format. The *horizon* determines the instant when the time will wrap around. The time format of TT Ethernet is a 64-bit binary time-format that is based on the physical second. Fractions of a second are represented as 24 negative powers of two (the granularity is approx. 63 nanoseconds), and full seconds are presented as 40 positive powers of two (the horizon is about 30,000 years).

4 Message Format

The TT-Ethernet message format is based on the *IEEE Ethernet standard*. An Ethernet frame consists of the following fi elds: *preamable* (7 bytes), *start of frame delimiter* (1 byte), *destination address* (6 bytes), *source address* (6 bytes), *frame type/frame length* (2 bytes), *data field* (46 to 1500 bytes), frame check sequence *FCS* (4 bytes), and *inter-frame gap* (12 bytes). A TT Ethernet message is identifi ed by the value 0x88d7 in the Ethernet *frame type* fi eld. The *data field* of the standard Ethernet message contains the *TT Ethernet header* followed by the *data field* of the TT Ethernet message.

The fi rst byte of the TT message header, the *control byte*, indicates the message category of a TT Ethernet message. Four categories are reserved for the standard TT Ethernet confi guration. The fault-tolerant TT Ethernet confi guration introduces these additional TT Ethernet message categories:

- *Protected TT Ethernet startup messages* are sent during the start-up phase.
- *Protected TT Ethernet synchronization messages* are sent periodically during the operational phase in order to support clock synchronization.

- *Protected TT Ethernet messages* are used to periodically send user data.

The second byte of the header denotes the size of data contained within a TT message in 8-byte blocks. The size of the headers is not included in this fi eld. The length of transmitted user data is always a multiple of 8 bytes.

The third and the fourth byte contain the *message ID* of a TT message. The *message ID* contains the information about the period and the phase of transmitted TT messages, and this information is also used to uniquely identify a TT message. TT Ethernet supports transmission of messages with 16 different periods. The difference of two TT Ethernet message periods is always a power of two. For a detailed explanation of the structure of the *message ID* fi eld, the reader is referred to [KAGS05].

The remaining header fi elds contain information about the *node identification*, *schedule identification*, *membership data*, etc. The membership is a 64 bit long vector because TT Ethernet supports a single cluster confi guration of 64 nodes with safety-critical TT Ethernet controllers. Each bit in the membership vector mirrors the operational status (*operating/not operating*) of one controller in the cluster. The membership and the clique avoidance mechanisms [BP00] are used to prevent and resolve the creation of different subsets (cliques) of nodes with inconsistent views about the global state of the system.

5 Clock Synchronization

The clock synchronization of fault-tolerant TT Ethernet uses the combination of a distributed *fault-tolerant clock state synchronization* algorithm with a *central clock rate correction synchronization* algorithm, as described in [KAH04]. The clock synchronization mechanisms used for the TT Ethernet extends the clock synchronization mechanism in [KAH04] by using not only the fault-tolerant distributed clock state correction algorithm but also a *fault-tolerant clock rate correction* algorithm.

The fault-tolerant confi guration of the TT Ethernet uses the *fault-tolerant average* algorithm (FTA) [LL84] for clock state correction, which is resilient against Byzantine faults [LSP82]. The FTA is suitable for systems that use periodic exchange of messages with a single period. The TT Ethernet supports 16 different periods for TT messages, and the periods of TT Ethernet messages depend on the application design. It is possible that an application design has a confi guration where nodes send periodic messages with different periods, and therefore it is diffi cult to achieve the clock synchronization of desired precision with the FTA. Therefore, TT Ethernet uses *synchronization messages*, such that there is always a confi guration of 5 nodes that send periodic messages with the same period (synchronization cycle). The FTA requires a minimum of 4 clock readings (that are retrieved implicitly by the periodic transmission of messages), and in the case of a single node failure, there will always be 4 clock readings for every synchronization cycle.

There exist at least two nodes in each cluster: They are denoted as *rate-master nodes* and have high-quality oscillators with a relatively low clock drift rate. Other nodes are denoted as *time-keeping nodes*. In each confi guration of TT Ethernet, the rate-master nodes always send synchronization messages.

Each TT Ethernet controller performs clock-rate correction based on the clock readings from the rate-master nodes. Since the *message reception instants* of synchronization frames of rate-master nodes are known a priori, each controller can perform *clock reading* from rate-master nodes based on the measured time difference between the expected and actual *message reception instant*. The measurement of the time difference between a rate-master node clock and a local clock is performed every time a *synchronization message* is received. Clock readings of both rate-master nodes are stored in separate buffers. One of the rate-master nodes is denoted as *primary rate master* and the other node as *secondary rate master*. The rate correction term is calculated based on the clock reading of the primary rate-master node. Since clock readings of both rate-master nodes are stored in separate buffers, in case of a failure of the primary rate-master, the rate correction term is calculated using the measured time difference values of the secondary rate-master node (which is classifi ed as the new primary rate master due to the failure of the former primary rate master). Each node sends the rate-master membership (a two bit fi eld denoting the primary rate-master node), and all nodes set the membership bit of the primary rate-master before performing the clock correction. The rate-master membership ensures that all nodes have a consistent view of which rate-master node is the primary rate-master node.

5.1 Start-up

In the fault-tolerant confi guration of the TT Ethernet there exist 5 nodes that perform the system startup. These nodes can send *startup messages* on the redundant channels with the period denoted as *startup period*. Each node listens on both channels for the startup frames for a time denoted as *listen timeout*. The listen timeout interval must be larger than the startup period. If no correct startup message was received during the listen timeout interval, each startup node initiates the *startup timeout* and if no correct startup message is received after the startup timeout interval, the node will send the fi rst startup message. To minimize message conflicts between two startup nodes, the startup timeout is different for each node. However, a startup message collision is always possible if two startup nodes are switched-on at signifi cantly different instants. The conflict of TT messages is resolved by the switch by assigning static priorities to the switch ports. The TT message in the port with the higher priority will be forwarded by the switch, whereas the other TT message will be dropped. After a successful reception of a startup message, the guardian will disable that input port of the switch for the interval of one startup period. In this way the guardian will prevent a faulty startup node, which may send messages at unspecifi ed instants, and therefore disturb the startup phase.

The fi rst startup frame is used to initialize the state of the cluster regarding the global time, and the membership bit of the node that has sent the fi rst startup message. The guardian has a confi guration fi le containing the mapping of *node IDs* to *switch port numbers*. Therefore, a faulty node cannot affect the operation of the cluster by sending a masqueraded message (give false information about its identity), because this faulty message will be recognized by the guardian and preempted before it is fully transmitted.

If during the listen timeout and the startup timeout a correct startup frame is received, the

nodes will always wait for the reception of a second correct startup frame before they send their first startup frame. During the startup phase, the guardian and the startup node will initialize their local view of the global time based on the time information contained in the first correctly received startup frame. Since the clocks of the nodes and the guardians are running free until 2 correct startup frames are received, the clock drifts of startup nodes and the guardians must be bounded.

Rate-master nodes and nodes that send synchronization frames are always chosen as startup nodes, because during the startup phase all nodes perform clock synchronization. All nodes (startup and non-startup) will leave the startup phase and continue with the operation in the *application phase* as soon as 4 startup nodes are operational (membership bits of four nodes are set) and after a predefined minimum number of clock corrections has been performed.

The issues related to a fault-tolerant start-up of a time-triggered system have been thoroughly investigated by Steiner [Ste04]. TT-Ethernet uses the fault-tolerant start-up algorithm described in his thesis. The presented startup differs in two points from [Ste04]: (i) message collisions do not exist in our case since message conflicts are resolved by the switch, (ii) clock synchronization is performed during the startup phase.

5.2 Re-Integration

When a guardian experiences a transient fault and loses its state information, it will restart and obtain the current state information from the incoming messages within a specified number of synchronization rounds.

When a safety-critical TT Ethernet controller experiences a transient fault and loses its state information, it will restart and obtain the current state information in the same ways as the guardian. A safety-critical TT Ethernet controller will use synchronization messages to restore the global time and the membership view. Each controller will restore the state if at least two correct synchronization frames from different nodes are received. No explicit state recovery procedure needs to be established, neither for the guardian nor for the controller.

6 Fault Hypothesis

In any fault-tolerant architecture it is important to distinguish clearly between *fault containment* and *error containment*. Fault containment is concerned with limiting the immediate impact of a single fault to a defined region, while error containment tries to avoid the propagation of the consequences of a fault, the error [Kop03]. It must be avoided that an error from one fault-containment region propagates into another fault-containment region that has not been directly affected by the original fault. A fault-containment region (FCR) is defined as a set of subsystems that share one or more common resources. A fault in any one of these shared resources may thus impact all subsystems of the FCR, i.e. the

subsystems of an FCR cannot be considered to be independent of each other [KBJ00].

In TT Ethernet a smallest unit of failure (or a fault containment region) is considered either a safety-critical TT Ethernet node, a TT Ethernet switch, a guardian, or a communication link. An error containment region in TT Ethernet consist of a safety-critical TT Ethernet controller, a TT Ethernet switch and a guardian.

Fault-Tolerant TT Ethernet tolerates a single arbitrary FCR failure at one point in time without an impact on the operation of the safety-critical part of the system. A faulty controller can affect only the operation of nodes with a standard TT Ethernet controller, whose messages are not protected by the guardian.

7 Existing Real-Time Ethernet Solutions

Several different real-time Ethernet solutions were developed by the industry and the academic research community [VC94, KS00, PAG02]. A summary of the state of the art in real-time Ethernet can be found in [Dec05, Fel05] and at the web-page *http:www.real-time-Ethernet.de*.

POWERLINK (see *www.ethernet-powerlink.org*) is an Ethernet extension that is intended as a new generation fi eld bus. It deploys hubs instead of switches in order to provide constant transmission delays. To prevent collisions among the participants, a TDMA method is used instead of the standard CSMA/CD (Carrier Sense Multiple Access with Collision Detection). The solution does not require specifi c hardware, and the achieved precision and performance depends on the host computers. It is not possible to mix standard Ethernet nodes in the real-time network without sacrifi cing the real-time guarantees.

PROFINET [Fel04, FFMT04] provides three classes of traffi c: real-time (RT), non-RT, and isochronous RT. Timely delivery of RT packets is supported by the VLAN priority support as specifi ed in IEEE 802.1 with a minimum jitter of about $125\mu s$. Isochronous RT packets are transmitted during the time interval that is reserved only for these packets (this can be guaranteed by a dedicated switch). Transmissions of the isochronous packets are synchronized to a precision of $1\mu s$. Collisions between two RT, or RT and non-RT packets are avoided by using a TDMA scheme. Latencies caused by switches are signifi cant and require the installation of additional hardware for time measurement. The PROFINET switch requires a confi guration for a specifi c application schedule, in order to guarantee constant transmission delays for RT messages.

EtherCAT [JB04] is a different RTE solution presented above. It deploys dedicated network controllers in hosts and standard Ethernet cabling. The assumed topology is a ring formed by a single master and the attached fi eld devices. Real-time data is transported either in raw Ethernet packets or encapsulated in UDP. The latter can pass through the master to other networks, though the real-time properties are lost once the packet leaves the local network. The master initiates every transmission and this message is modifi ed (the responses of slaves are written into it) as it passes through the ring. This minimizes the query-response cycle length and the jitter as well. The EtherCAT protocol does not allow the use of standard Ethernet nodes in the same network.

Most of the real-time Ethernet implementations are based on the configuration of the Ethernet controllers that transmit real-time traffic with time guaranties based on a fixed communication schedule. Some protocols like FFT-Ethernet [PAG02] and PROFINET [Fel04] can integrate the real-time and non-real-time traffic as well, but non-real-time messages are only exchanged in statically allocated time slots. Using a standard Ethernet controller in such systems is not possible without changes in the software. TT Ethernet can integrate time-triggered traffic and standard Ethernet traffic into the same communication network. The TTE switch mechanisms enable this integration without having to change the networking protocols for exchange of standard Ethernet messages which are handled according to the IEEE 802.3 standard [IEE].

8 Conclusion

In this paper we present the design for a fault-tolerant configuration of time-triggered Ethernet that is intended to serve as a communication system for most demanding fault-tolerant control systems that require certification. The design of fault-tolerant configuration of time-triggered Ethernet unifies the real-time traffic of fault-tolerant applications as well as the real-time and non real-time traffic of non fault-tolerant applications into the same communication platform. The TT Ethernet guaranties the temporal properties of the real-time traffic and prevents error propagation in the presence of component and communication faults. TT Ethernet is intended to support all types of applications, from simple data acquisition systems, to multimedia systems up to the most demanding safety-critical real-time control systems which require a fault-tolerant communication service that must be certified. The non real-time traffic in TT Ethernet is handled in conformance with the existing Ethernet standards of the IEEE.

Literatur

[Ade02] Astrit Ademaj. Slightly-Off-specification Failures in the Time-Triggered Architecture. In *Seventh Annual IEEE Workshop on High-Level Design Validation and Test (HLDVT02)*, pages 7–12, Cannes, France, October 2002.

[ARI04] ARINC. Minimal Operational Performance Standards for Avionics Computer Resource. ARINC RTCSA-SC-182/Eurocae WG-48, Washington D.C., June 4, 2004.

[BKS03] G. Bauer, H. Kopetz, and W. Steiner. The Central Guardian Approach to Enforce Fault Isolation in the Time-Triggered Architecture. In *Proceedings of the 6th International Symposium on Autonomous Decentralized Systems*, Pisa, Italy, April 2003.

[BP00] G. Bauer and M. Paulitsch. An Investigation of Membership and Clique Avoidance in TTP/C. In *Proceedings 19th IEEE Symposium on Reliable Distributed Systems (SRDS'00)*, pages 118–124, Nürnberg, Germany, October 2000.

[Dec05] Jean-Dominique Decotignie. Ethernet-Based Real-Time and Industrial Communications. *Proceedings of the IEEE*, 93(6):1102–1117, June 2005.

[Eth05] Ethernet. EtherType Field Public Assignments. http://standards.ieee.org/regauth/ethertype/eth.txt, 2005.

[Fel04] Joachim Feld. PROFINET – Scalable Factory Communication for all Applications. In *IEEE International Workshop on Factory Communication Systems*, pages 33–38, September 2004. ISBN 0-7803-8734-1.

[Fel05] Max Felser. Real-Time Ethernet-Industry Prospective. *Proceedings of the IEEE*, 93(6):1118–1129, June 2005.

[FFMT04] P. Ferrari, A. Flammini, D. Marioli, and A. Taroni. Experimental Evaluation of PROFINET performance. In *IEEE International Workshop on Factory Communication Systems*, pages 331–334, September 2004. ISBN 0-7803-8734-1.

[IEE] IEEE Standard 802.3, 2000 Edition. Carrier Sense Multiple Access with Collision Detect on (CSMA/CD) Access Method and Physical Layer Specifications.

[JB04] D. Jansen and H. Buttner. Real-Time Ethernet the EtherCAT Solution. *Computing & Control Engineering Journal*, 15(1):16–21, February 2004. ISSN 0956-3385.

[KAGS05] Hermann Kopetz, Astrit Ademaj, Petr Grillinger, and Klaus Steinhammer. The Time-Triggered Ethernet (TTE) Design. In *8th IEEE International Symposium on Object-oriented Real-time distributed Computing (ISORC)*, Seattle, Washington, May 2005.

[KAH04] Hermann Kopetz, Astrit Ademaj, and Alexander Hanzlik. Clock-State and Clock-Rate Correction in Fault-Tolerant Distributed Systems. In *The 25th IEEE International Real-Time Systems Symposium, Lisbon, Portugal, Dec. 2004*, pages 415–425, December 2004.

[KBJ00] L. M. Kaufman, S. Bhide, and B. W. Johnson. Modeling of Common-Mode Failures in Digital Embedded Systems. In *Proceedings of the Reliability and Maintainability Symposium 2000. IEEE Press*, pages 350–357, Los Angeles, USA, 2000.

[Kop03] H. Kopetz. Fault Containment and Error Detection in TTP/C and FlexRay. In *Proceedings of The Sixth International Symposium on Autonomous Decentralized Systems (ISADS '03)*, pages 139–148, Pisa, Italy, April 2003.

[KS00] S.-K. Kweon and K. G. Shin. Achieving Real-Time Communication over Ethernet with Adaptive Traffic Smoothing. In *Sixth IEEE Real-Time Technology and Applications Symposium*, pages 90–100, May 2000.

[LL84] J. Lundelius and N. Lynch. An Upper and Lower Bound for Clock Synchronization. *Information and Control*, 62:190–204, 1984.

[LSP82] L. Lamport, R. Shostak, and M. Pease. The Byzantine Generals Problem. *ACM Transactions on Programming Languages and Systems*, 4(3):382–401, 1982.

[OMG02] OMG. Smart Transducer Specification TTP/A. ftp://ftp.omg.org/pub/docs/orbos/01-10-02.pdf Object Management group, 2002.

[PAG02] P. Pedreiras, L. Almeida, and P. Gai. The FTT-Ethernet Protocol: Merging Flexibility, Timeliness and Efficiency. In *14th Euromicro Conference on Real-Time Systems*, pages 134–142, June 2002.

[Ste04] Wilfried Steiner. *Startup and Recovery of Fault-Tolerant Time-Triggered Communication*. PhD thesis, Vienna University of Technology, Real-Time Systems Group, Vienna, Austria, 2004.

[VC94] C. Venkatramani and T. Chiueh. Supporting Real-Time Traffic on Ethernet. In *Real-Time Systems Symposium (RTSS)*, pages 282–286, December 1994.