FAULT-TOLERANT COMPENSATION OF THE PROPAGATION DELAY FOR HARD REAL-TIME SYSTEMS

Thomas Losert, Martin Schlager, Wilfried Elmenreich
Institute of Computer Engineering
Vienna University of Technology
Vienna, Austria
{thomas, smartin, wilfried}@vmars.tuwien.ac.at

ABSTRACT
In control systems the jitter is a relevant problem since the majority of theoretical results for analysis and design of time-invariant systems do not apply for time-variant systems. Reducing the jitter increases the stability of the closed control-loop thus leading to enhanced reliability.

This paper presents a general model that can be applied to bus topologies as well as to star topologies. Based on this model an algorithm is presented that allows to improve the precision of a set of distributed clocks by measuring the propagation delay of the communication lines in a fault-tolerant way and compensating the jitter introduced by the propagation delay.

Some fault-tolerant architectures already provide means for coping with propagation delays but require manually entering the values in a configuration-tool. With this algorithm the system supports this error-prone task by providing validity checks for the entered values or measuring these values automatically thus rendering this maintenance step obsolete.

KEY WORDS
Hard Real-Time Systems, Communication Model, Compensation of Propagation Delay, Precision of Distributed Clocks, Fault Tolerance.

1 Introduction

In a distributed fault-tolerant application, a scenario that has to be considered is the physical destruction of a limited part of space, e. g., due to a fire on board of an aeroplane. In order to reduce the risk of a correlated malfunction and reach a failure probability on the order of 10^{-9} as required for ultra-dependable systems (see [1, p. 5]) safety-critical components should be spatially separated although this approach will increase the total length of cable. Even for non-safety-critical systems like the in-flight-entertainment system, where each seat requires a connection to the server, signals have to cover significant distances of cabling (see [2]). Substantial propagation delays must also be considered for remote-controlled power plants, railroads, or in satellite communication. Further, submarine communication which often utilizes modulated sound instead of electromagnetic waves, suffer from substantial propagation delays.

A short calculation reveals that 200 m of cable introduce a propagation delay of about 1 μs whereas the propagation delay for acoustic communication over the same distance in water is about five orders of magnitude above. Further, the propagation delay introduced by network equipment (e. g., switches) can be estimated with several microseconds (see also [3]).

If not corrected, the propagation delay has to be considered as jitter in the real-time application, thus degrading precision in an ensemble of distributed clocks (see [4]) or stability in control applications since in a time-varying system the theoretical results for analysis and design of time-invariant systems cannot be used directly (see [5]). There is a tendency to higher bandwidth and shorter execution times which will further increase the negative effects of jitter in future.

Another issue is that often the geometric dimensions of the cabling are not known in advance or are subject to change during the development process. Thus, these correction terms introduce a source of error. Each parameter in a tool that has to be adjusted manually is a potential source of problems. A robust and easy configurable architecture should determine as much parameters as possible automatically and provide validity checks whenever possible. This reduces the mental complexity for the system designer and lowers development and maintenance costs since problems can be detected earlier.

According to [6] time-triggered architectures are ideally suited for the periodic operation in distributed fault-tolerant control systems and the implementation of safety critical systems like X-by-wire probably will fail without the framework of time-triggered architectures. To the best of our knowledge none of the time-triggered architectures that are available today allow the correction of the propagation delay as proposed in this paper.

The algorithm for reconstructing a model for the communication subsystem can be subdivided into several steps of measuring the propagation delay in a star of three nodes. Since the execution time of each step is bounded, this measurement algorithm can be executed as asynchronous communication in a special measurement mode during a single Time Division Multiple Access (TDMA) slot and interleaved with real-time communication. Thus, the execution of this algorithm does not influence real-time communication.

This algorithm can be used in safety critical applications during the startup or during maintenance, when the system is in a safe state, in order to perform a last check of the calibration values that have been entered manually. In uncritical applications maintenance costs can be cut down since the system is able to determine the necessary parameters automatically.

In a network with star topology, where each connection line is considered to be in the same fault containment region as the corresponding node, this algorithm can measure the propagation delay of each correct node. Further, periodic execution allows to recover from transient faults.

The remainder of this paper is structured as follows: Section 2 describes the communication model that is flexible enough to cover bus topologies as well as star topologies and is used as the basis for the algorithm presented in section 3 which is used for precise measurement of the propagation delay. Section 4 presents a more general version of this measurement algorithm that supports fault-tolerance also. In section 5 a simplified communication model is described that can be reconstructed based on the measurement values. Section 6 presents an efficient algorithm for compensating the propagation delay that requires just one calibration value per node for a star topology and is flexible enough to cover all communication systems that are possible in the chosen communication model. Section 7 discusses some details of this algorithm whereas section 8 concludes this paper.

2 Communication Model

In this paper a communication model based on the 10 Base-5 ethernet standard ("thick ethernet", see [7, chp. 8]) is assumed that covers, among others, all networks in bus topology and all networks in star topology as special cases (in the latter the propagation delay introduced by the main line is zero). Since a distributed fault-tolerant clock consists of at least three clocks, a degenerated network consisting of a point-to-point link between two nodes is not considered in this model.

As depicted in figure 1, a set of at least three nodes communicates with broadcast messages, i. e., each message that is sent by a correct node is received by each other correct node after an individual propagation delay. For this propagation delay an *a priori* known upper bound δ_{max} is given.

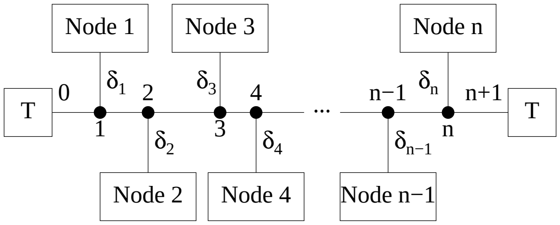

Figure 1. Model for the Propagation Delay

The *main line* is a cable with a terminator T at both ends for preventing reflections of the signal. A set of nodes N (whereas n is the number of nodes) is connected with *branch lines* to the main line. The propagation delay is independent from the direction of propagation of the signal on the medium. We assume that on a correct channel the variation of the propagation delay (e. g., due to variation in temperature) within a measurement cycle can be neglected.

A signal generated by a node propagates along its branch line, splitting up at the intersection point of this branch line with the main line and propagates along the main line towards both terminators. At each intersection point with another branch line it splits up again and is received by the node terminating this branch line.

We assume that a communication according to a collision-free media access strategy (e. g., TDMA) has been established already. Thus, the problem of collisions when two or more nodes send a frame approximately at the same instant will not be considered. Based on the *a priori* knowledge that the end-to-end delay between two arbitrary nodes is bound by δ_{max} the nodes are already synchronized and a global view of time has been established (see [8] for an example of how to reach synchronicity), but the precision of the set of clocks is limited by the propagation delay.

Without loss of generality we call one arbitrary end of the main line the *begin* and the other one the *end*. All positions of interest (i. e., begin and end as well as the intersections with the branch lines to the nodes) are labeled with ascending integral numbers. We refer to the propagation delay from the begin 0 to position k of the main line with $\phi_{0,k}$. This could be imagined as sending a short impulse from the terminator at the beginning and measuring the propagation delay to this position.

The propagation delay on the main line between the intersection with the branch line of node i and node j can be calculated as

$$\phi_{i,j} = \phi_{j,i} = |\phi_{0,i} - \phi_{0,j}|.$$

We refer to the propagation delay introduced by the branch line from node i to its intersection point with the main line as δ_i. Thus, the propagation delay between node i and node j can be calculated as

$$\delta_{i,j} = \delta_{j,i} = \delta_i + \phi_{i,j} + \delta_j$$

and – based on the assumption stated above – is bounded with δ_{max} (i. e., $\delta_{max} \geq \delta_{i,j}\ \forall i, j \in N$).

The instant in the TDMA schedule intended for the begin of the transmission of a message m by node i is denoted with t_i^m while $t_{snd,i}^m$ is used for the actual begin of transmission. With $t_{rcv,i,j}^m$ we denote the instant where node j starts receiving the message m that has been sent by node i.

3 Measuring Delays

In a communication system as described in the previous section, that consists of three nodes, all portions of the main line can be considered to be part of the branch lines, thus, having an equivalent model in star topology, i. e., without a main line (see figure 2). We call the intersection of the three communication lines of the nodes r, a, and o the center $C_{r,a,o}$ of the star and the delay from node i to this center $\delta_i^{r,a,o}$.

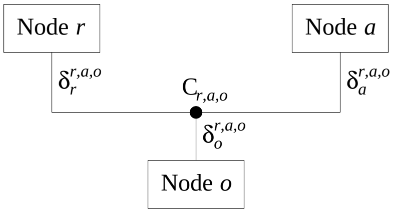

Figure 2. Measuring Delays with three Nodes

The measurement of the propagation delay is based on a calibration request *creq* of *a priori* known length that is sent by a requesting node r to an answering node a in the system. After an *a priori* known delay $\delta_{wait,a}$ (for receiving and processing the packet) node a sends the calibration answer *cans*. The observing node remains passive (i. e., receives only) and calculates the delay $\delta_a^{r,a,o}$.

The measurement can be broken down to the following steps:

1. Node r sends a packet with the calibration request *creq* to the network at the instant $t_{snd,r}^{creq}$.

2. The packet *creq* arrives at node a at the instant $t_{rcv,r,a}^{creq} = t_{snd,r}^{creq} + \delta_r^{r,a,o} + \delta_a^{r,a,o}$ while it arrives on node o at the instant $t_{rcv,r,o}^{creq} = t_{snd,r}^{creq} + \delta_r^{r,a,o} + \delta_o^{r,a,o}$.

3. After an *a priori* known delay $\delta_{wait,a}$ node a sends the calibration answer *cans* at the instant $t_{snd,a}^{cans} = t_{rcv,r,a}^{creq} + \delta_{wait,a}$.

4. The packet *cans* arrives at node r at the instant $t_{rcv,a,r}^{cans} = t_{snd,a}^{cans} + \delta_a^{r,a,o} + \delta_r^{r,a,o}$ while it arrives on node o at the instant $t_{rcv,a,o}^{cans} = t_{snd,a}^{cans} + \delta_a^{r,a,o} + \delta_o^{r,a,o}$.

5. Node o can calculate $\delta_a^{r,a,o}$ since $t_{rcv,a,o}^{cans} - t_{rcv,r,o}^{creq} = \delta_{wait,a} + 2 \cdot \delta_a^{r,a,o}$.

Although $\delta_a^{r,a,o}$ is already known at node o and node r could calculate $\delta_r^{r,a,o}$ since

$$t_{rcv,a,r}^{cans} - t_{snd,r}^{creq} = \delta_{wait,a} + 2 \cdot (\delta_r^{r,a,o} + \delta_a^{r,a,o})$$

this is inadvisable due to measurement errors that could possibly sum up. In order to minimize systematic measurement errors only receiving timestamps of the same node should be compared, but not a sending timestamp with a receiving timestamp. Thus, a sending node cannot be used as observing node in the next round. Instead, the measurement continues as follows:

After the first round the nodes change their roles (node a becomes node r', node r becomes node o', and node o becomes node a') and the calibration answer *cans* is taken as calibration request *creq'* for the second round, thus allowing node o' (the former node r) to calculate $\delta_{a'}^{r',a',o'} = \delta_o^{r,a,o}$. Finally, the nodes change their roles again and the calibration answer *cans'* of round two is taken as calibration request *creq''* for the third round, which allows node o'' (the former node a) to calculate $\delta_{a''}^{r'',a'',o''} = \delta_r^{r,a,o}$ (see table 1).

Msg.	Node 1	Node 2	Node 3	Data
1	r			$\emptyset$
2		a/r'	o	$\emptyset$
3	o'		a'/r''	δ_2
4	a''	o''		δ_3
5		b'''		δ_1

Table 1. Measurement of the Propagation Delay

The measured delays $\delta_r^{r,a,o}$, $\delta_a^{r,a,o}$, and $\delta_o^{r,a,o}$ are distributed as payload of the respective calibration answer since the observing node becomes the answering node in the next round. Thus, a fourth round is required where the observing node of the previous round broadcasts the last result in a fifth packet.

It is remarkable that the quality of this measurement is independent from the instant of sending the request and thus, a jitter for $t_{snd,r}^{creq}$ can be tolerated. On the other hand the parameter $\delta_{wait,a}$ is crucial since it directly influences the calculation for the propagation delay.

Besides from comparing the calculated propagation delay with the *a priori* known value δ_{max} or checking the measured values by a human operator, transient problems in the answering node that cause a deviation from $\delta_{wait,a}$ can be detected with a given probability by repeated measurements.

Obviously, this algorithm can be generalized to a network of n nodes ($n \geq 3$) in star topology for measuring the propagation delay of all n branch lines by performing n measurement rounds. This requires $n + 1$ packets on the communication medium and in packet $n + 2$ the last result is broadcast to all other nodes. Nevertheless, the worst case execution time is bound by the number of required packets and the *a priori* known values δ_{max} and $\delta_{wait,i}$.

Further, this algorithm is not limited to the model as described above but can be generalized to other network topologies as long as the presented prerequisites are not violated.

4 Fault-Tolerant Measurement

According to [9] a *failure* occurs when the delivered service deviates from the expected or specified service. An *error* is the occurrence in the system that leads to the failure and a *fault* is the cause of the error.

For the fault-tolerant approach we assume a network of n nodes with at most f faulty nodes at a time.

The considered faults of a node cover arbitrary value faults and timing faults within the interval $[0, \delta_{max}]$. Further, we assume all nodes to be fail-silent, i. e., each node i answers within the interval $\delta_{wait,i} + \delta_{max}$ or remains silent. Other faults, such as babbling idiot faults (see [10]) are outside the fault hypothesis of our approach, however, they may be handled using strong fault tolerance mechanisms as assumed in the Time-Triggered Architecture (see [11]).

Each single node together with its branch line is considered a fault containment region of its own. Thus, the fault-tolerant approach requires a network in star topology.

In the following we will extend the measurement algorithm presented in the previous section in order to reach fault-tolerance. We have to distinct between faults in the requesting node, the answering node, or the observing node:

Since the observing node o is passive, according to the fault hypothesis only value faults are possible. This type of fault can be masked by having at least $2 \cdot f + 1$ observing nodes and calculating the median of these results.

Faults in the requesting node r cannot influence the quality of the measurement but it is required as trigger for starting the measurement. Thus, if no valid request packet is received within $\delta_{wait,i} + \delta_{max}$, another node takes over and sends the request. For f faulty nodes at least f spare nodes are necessary and the time required for sending the request is limited with $(f+1) \cdot \delta_{max}$. Since each faulty requesting node reduces the number of possibly faulty observing nodes, we can use up to f observing nodes as spare nodes. A good choice is using the requesting nodes of the previous rounds as spare requesting nodes, beginning with the (spare) requesting node of the round straight before.

A jitter or omission fault in the answering node a corrupts the measurement of the propagation delay of its branch line but this does not matter since they are in the same fault containment region. It is not possible that a faulty answering node degrades the result for correct nodes.

We conclude that in a star with up to f faulty nodes one requesting node plus one answering node plus at least $2 \cdot f + 1$ observing nodes (in total $2 \cdot f + 3$ nodes) are required for a fault-tolerant measurement. Since the request frame is also used for distributing the measured data, each packet contains $2 \cdot f + 1$ values (at the beginning and in the presence of faults some values are undefined but the packet length should be kept constant in order to avoid systematic measurement errors).

In the absence of faults a full measurement in a network of n nodes requires $n + 2 \cdot f + 2$ packets to be sent on the network, whereas up to f further packets are required with faults. The last $2 \cdot f + 1$ packets of each measurement are used for broadcasting the last results. Since the length of each message is known, the maximum execution time can be determined as $(\delta_{max} + \delta_{wait,i}) \cdot (n + 3 \cdot f + 2)$.

Table 2 shows an example for $n = 5$ nodes that can tolerate up to $f = 1$ faulty nodes. In this example node 3 is considered to be faulty (e. g., its branch line is broken). Thus, node 3 cannot answer and the observing nodes detect a timeout in the second measurement round. Then node 2 issues the request for the third round and cannot be used as observing node in the next round due to the limitations stated in the previous section. At least, we have two valid observations for each node (except for the faulty node 3) which is sufficient.

Compared with the non fault-tolerant measurement we notice that – independent from the number of nodes n in the network – just $2 \cdot f$ packets more have to be sent whereas each packet contains $2 \cdot f$ values more as payload.

The redundancy of having several observing nodes does also improve the quality of the measurement by reducing the influence of nodes with a high drift at the local clock. Further, it is possible to extend this algorithm to any network according to the presented model as long as it can be grouped into subnets in star topology with at least $2 \cdot f + 3$ nodes.

5 Reconstruction of a Simplified Model

In a fault-tolerant application – depending on the number of tolerated faults – the minimum number of nodes per subnet in star topology is five or even more. Thus,

Msg.	Node 1	Node 2	Node 3	Node 4	Node 5	Data
1	r					$\emptyset$, $\emptyset$, $\emptyset$
2		a/r'	~~o_1~~	o_2	o_3	$\emptyset$, $\emptyset$, $\emptyset$
3	o'_3		~~a'/r''~~	o'_1	o'_2	~~$\emptyset$, $\emptyset$, δ_2~~
4		r''				$\emptyset$, $\emptyset$, $\emptyset$
5	o''_2	~~o''_3~~		a''/r'''	o''_1	$\emptyset$, δ_2, ~~δ_3~~
6	o'''_1	o'''_2	~~o'''_3~~		a'''/r''''	δ_2, ~~δ_3~~, δ_4
7	a''''	o''''_1	~~o''''_2~~	o''''_3		~~δ_3~~, δ_4, δ_5
8		b'''''				~~δ_4~~, δ_5, δ_1
9			~~b''''''~~			~~δ_5, δ_1, δ_2~~
10				b'''''''		δ_1, δ_2, δ_3

Table 2. Fault-Tolerant Measurement of the Propagation Delay with a Fault in Node 3 ($n = 5, f = 1$)

only a subset of all possible models is supported and some *a priori* knowledge about the topology of the network is required. Reconstructing the model based on this knowledge is straight forward with the fault-tolerant measurement algorithm as described in the previous section.

However, in the absence of faults a star consisting of three nodes is sufficient. Since any three nodes according to the presented model can be grouped to a network in star topology, for reconstructing a valid model of the propagation delays there is no need for any further *a priori* knowledge besides from the fact, that the network is valid according to the full model as described in section 2.

Due to the fact that only the propagation delay from one node to a particular reference point (the center $C_{i,j,k}$ of the star established by three arbitrary nodes i, j, and k of the network) can be measured, it is not possible to reconstruct the model unambiguously. Instead, we use the measured delays to reconstruct a simplified model. The set of simplified models is a subset of the set of full models.

The terminators do not participate in the communication and thus have been removed from the simplified model. Further, in order to reduce ambiguity we mandate that the simplified model of a network is either in star topology or the two intersections at the begin and at the end of the main line are shared by at least two nodes, i. e., $\phi_{1,2} = \phi_{n-1,n} = 0$.

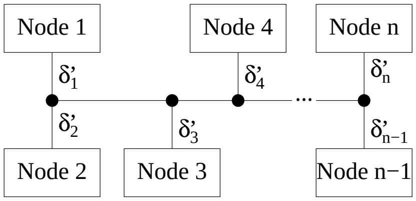

Figure 3. Simplified Model for the Network in Figure 1

In fact the network in figure 3 is equivalent to the network in figure 1 if $\delta'_1 = \delta_1 + \phi_{1,2}$, $\delta'_n = \delta_n + \phi_{n-1,n}$, and $\delta'_i = \delta_i$ ($\forall i \in [3, n-2]$).

Without loss of generality the nodes are numbered from 1 to n in an arbitrary order. The simplified model is generated as follows (this algorithm is a generalization of the algorithm presented in [12]):

1. The nodes 1, 2, and 3 of the set of nodes are used to reconstruct a model for these three nodes in star topology with $C_{1,2,3}$ as center as described in the previous section.

2. If the current model is a network in star topology consisting of i nodes, node $i+1$ is integrated by performing $\lceil \frac{i}{2} \rceil$ measurements as follows: If i is even, all i nodes that are already in the model are grouped to $\frac{i}{2}$ disjunct pairs $(1, 2) \ldots (i-1, i)$ and for each pair (a, b) the measurement for the star consisting of node a, b, and $i+1$ is performed. If i is odd, one node is used twice for another measurement, e. g., node 1, i, and $i+1$.

3. If the current model consists of i nodes that are not in star topology (i. e., $\exists k, l$ that $\phi_{k,l} > 0$) the nodes are renumbered that the nodes 1 to b ($b \geq 2$) share the intersection point at the begin and the nodes e to i ($i - e \geq 2$) share the intersection point at the end (all the nodes between node b and node e are at other intersection points).

4. Node $i+1$ is integrated into the existing model by performing the measurement as described in the previous section for the star consisting of node k, l, and $i+1$ whereas node k is a node from the begin and node l is a node from the end. This requires $\max(b, i-e)$ measurements and solving a system of equations for determining the proper place of node $i+1$ in the current model.

5. Each remaining node is integrated by performing the last two steps.

This allows calculation of a valid simplified model as well as adapting the model if further nodes have to be integrated later. Since the measurement can be subdivided into simple steps with an upper bound for the execution time, this measurement algorithm can be executed as asynchronous communication in a special measurement mode during a dedicated TDMA slot. This slot for reconstructing the model or integrating new nodes can be interleaved with slots for hard real-time communication.

6 Compensation of Propagation Delay

Based on the simplified communication model as described above, this approach allows compensation of the propagation delay on the communication lines. Each node i in the network requires to know the following parameters:

$$\delta_i \quad \text{and} \quad \phi_{i,j}\ \forall j \in N \backslash \{i\}.$$

Each sending node i compensates the delay of message m introduced due to the branch line of node i by sending it δ_i before the intended point in time t_i^m. Thus, message m arrives at the intersection of its branch line with the main line exactly at t_i^m. Each receiving node j corrects the timestamp for the perception of the message $t_{rcv,i,j}^m$ by subtracting $\phi_{j,i} + \delta_j$ from this timestamp. This compensates the propagation delay introduced by the main line and the branch line of the receiving node (cf. figure 4).

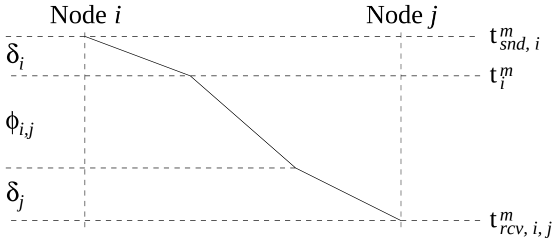

Figure 4. Compensation of Propagation Delay for message m

For star topologies, where the term $\phi_{i,j}$ equals to zero for all pairs of nodes i, j it is sufficient for each node i to know the propagation delay δ_i only. This is a highly scalable approach since the number of correction values per node does not depend on the number of nodes in the network. Furthermore, the fault-tolerant measurement could be used.

For masking certain types of faults a central control device is placed at the center of a network in star topology. The temporal design of this device is simplified, since all frames arrive at the intended instant and thus no correction terms are required in this device.

7 Discussion

In distributed control systems two parameters have major influence on the performance of the closed control loop: the latency and the jitter. Depending on the application, the latency can be compensated by applying state estimation algorithms in order to gain a similar performance as for a locally controlled device (see [13] for an example).

The jitter is a major problem since in a time-varying system the theoretical results for analysis and design of time-invariant systems cannot be used directly. Reducing the jitter is correlated with improving the performance and stability in a distributed control application. In [14] and [15] the negative effects that jitter can introduce in a control-loop are demonstrated as well as the importance of compensation.

The presented algorithm for measuring the propagation delay of a cable for reaching higher precision in a set of distributed clocks requires an accuracy in the submicrosecond range. Although the algorithm for measuring the round-trip time as proposed in [16] might look similar to the non fault-tolerant algorithm presented in section 3 there are differences in details that allow to increase the accuracy by reduction of systematic measurement errors:

Neither $\delta_r^{r,a,o}$ (plus any additional delays in the sending path of node r) nor $\delta_o^{r,a,o}$ (plus any additional delays in the receiving path of node o) influence the measurement of $\delta_a^{r,a,o}$ as long as these delays remain constant. Since both packets, packet *creq* as well as packet *cans*, are timestamped by the local clock of observing node(s) the current precision of the distributed clock does not influence the quality of this measurement but only the quality of the local clock. Further, the influence of high drift at a few nodes is lowered by calculating the median of all available results.

By allowing nodes to be gateways to other networks with synchronized TDMA-cycles this model can be generalized to hierarchical networks of high complexity. If the delay of the gateway is deterministic and constant it can be modeled as an additional propagation delay.

8 Conclusion

This paper demonstrates how the propagation delay in an arbitrary network of nodes according to the presented model can be measured. The proposed measurement method eliminates several sources of systematic measurement error by design thus allowing a better quality of the measurement result.

By using the measurement values a communication model of the system is reconstructed and the propagation delay of the communication channel compensated, thus, eliminating the delay that has to be considered as jitter if unknown. This allows e. g., a distributed clock with better precision or a control loop that is more stable thus improving the reliability of the whole system.

For achieving fault tolerance not all networks possible with the simplified model are supported but just networks that can be grouped into subnets in star topology with a sufficient number of nodes each. The required number of nodes depends on the number of tolerated faults.

Acknowledgements

This work has been supported by the CRTC project (contract No. 808039) which is funded by the FIT-IT programme of the Austrian Federal Ministry of Transport, Innovation and Technology and by the DOC programme of the Austrian Academy of Sciences.

Further, the authors would like to thank the anonymous reviewers for valuable comments regarding this paper.

References

[1] Neeraj Suri, Chris J. Walter, and Michelle M. Hugue, editors. *Advances in Ultra-Dependable Systems*. IEEE Computer Society Press, Los Alamitos, CA, U. S. A., 1995. ISBN 0-8186-6285-9.

[2] Gerald D. Lui-Kwan. In-Flight Entertainment: The Sky's the Limit. *IEEE Computer*, 33(10):98–101, October 2000.

[3] Klaus Steinhammer. A TT-Ethernet Switch based on COTS-Components. Research Report 54/2004, Department of Computer Engineering, Vienna University of Technology, Vienna, Austria, 2004. Available at `http://www.vmars.tuwien.ac.at`.

[4] Hermann Kopetz. *Real-Time Systems: Design Principles for Distributed Embedded Applications*. Kluwer Academic Publishers, Boston, Dordrecht, London, January 1997. ISBN 0-7923-9894-7.

[5] Johan Nilsson. *Real-Time Control Systems with Delay*. Dissertation, Department of Automatic Control, Lund Institute of Technology, 1998. ISRN LUTFD2/TFRT–1049–SE.

[6] Albert Amos. Comparison of Event-Triggered and Time-Triggered Concepts with Regard to Distributed Control Systems. In *Proceedings of the Embedded World 2004*, pages 235–252, Nuremberg, Germany, February 17–19, 2004. Available at `http://www.can.bosch.com/docu/embedded_world_04_albert.pdf`.

[7] Institute of Electrical and Electronics Engineers, New York, NY, U. S. A. *Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications*, 2000 edition, 2000. ISBN 0-7381-2674-8.

[8] Wilfried Steiner, John Rushby, Maria Sorea, and Holger Pfeifer. Model Checking a Fault-Tolerant Startup Algorithm: From Design Exploration To Exhaustive Fault Simulation. In *Proceedings of the International Conference on Dependable Systems and Networks (DSN 2004)*, pages 171–180, Florence, Italy, June 28, – July 1, 2004.

[9] Victor P. Nelson. Fault-Tolerant Computing: Fundamental Concepts. *IEEE Computer*, 23(7):19–25, July 1990.

[10] Günther Bauer, Thomas Frenning, Anna-Karin Jonsson, Hermann Kopetz, and Christopher Temple. A Centralized Approach for Avoiding the Babbling-Idiot Failure in the Time-Triggered Architecture. In *Workshops and Abstracts of the International Conference on Dependable Systems and Networks (DSN 2000)*, pages B-6–B-7, New York, NY, U. S. A., June 25–28, 2000.

[11] Günther Bauer, Hermann Kopetz, and Wilfried Steiner. The Central Guardian Approach to Enforce Fault Isolation in the Time-Triggered Architecture. In *Proceedings of the Sixth International Symposium on Autonomous Decentralized Systems (ISADS 2003)*, pages 37–44, Pisa, Italy, April 9–11, 2003.

[12] Thomas Losert, Wilfried Elmenreich, and Martin Schlager. Semi-Automatic Compensation of the Propagation-Delay in Fault-Tolerant Systems. In *Proceedings of the Third IASTED International Conference on Communications, Internet, and Information Technology (CIIT 2004)*, pages 455–460, St. Thomas, VI, U. S. A., November 22–24, 2004. ISBN 0-88986-445-4.

[13] Gustavo Hommerding Alt and Walter Fetter Lages. Networked Robot Control with Delay Compensation. Technical report, Federal University of Rio Grande do Sul, Porto Alegre, RS, Brazil, November 9–11, 2003. Available at `http://www.eletro.ufrgs.br/~fetter/rtlws03.pdf`.

[14] Pau Marti, Josep M$^{\text{a}}$ Fuertes, Gerhard Fohler, and Krithi Ramamritham. Jitter Compensation for Real-Time Control Systems. In *Proceedings of the 22nd IEEE Real-Time Systems Symposium (RTSS 2001)*, pages 39–48, London, U. K., December 3–6, 2001. Available at `http://www.mrtc.mdh.se/publications/0321.pdf`.

[15] Wei Zhang, Michael S. Branicky, and Stephen M. Phillips. Stability of Networked Control Systems. *IEEE Control Systems Magazine*, 21(1):84–99, February 2001.

[16] David L. Mills. Internet Time Synchronization: the Network Time Protocol. RFC 1129, University of Delaware, Newark, DE, U. S. A., October 1989.