Hardware Support for Clock Synchronization in Distributed Systems

Martin Horauer

Abstract— **This article presents a novel network interface hardware architecture, which enables clock synchronization in fault-tolerant distributed real-time systems with sub-μs-range accuracy. The proposed mechanism, which is applicable for any packet-oriented data network, inserts time information into data packets at the interface between the physical layer transceiver and the network controller upon packet transmission and receipt respectively. Local time is supplied by a high-resolution rate-adjustable adder-based clock, which also contains hardware support easing interval-based external clock synchronization, like maintaining time and accuracy intervals and interfaces to GPS receivers.**

Keywords— **Clock synchronization, distributed systems, network interface, GPS.**

I. Introduction

MANY applications in distributed real-time systems are time-dependent and require a reliable time-service, e.g., several algorithms used in communication systems. In fact, synchronous data acquisition and simultaneous triggering of actuators at several nodes is impossible without such a feature. This service could be implemented with a central time server accessible from all the other nodes or a dispersed time service. The latter can be composed of an ensemble of distributed clocks that can be characterized with the worst case precision π satisfying $|C_i(t) - C_j(t)| \leq \pi \quad \forall t \geq t_0$ for any two fault-free clocks $C_i(t)$ and $C_j(t)$ in the system. The problem of keeping π bounded is addressed by the *internal clock synchronization*, a well-established field in the distributed computing domain. The problem is aggravated when the system time is related to an external reference like UTC, the only legal standard of time. The maximum deviation towards UTC is termed *accuracy* α, formally $|C_i(t) - t| \leq \alpha \quad \forall t \geq t_0$. Maintaining α bounded is the aim of the *external clock synchronization*. Reaching both goals jointly turns out to be a non-trivial problem, since a certain tradeoff seems to be involved [2].

Most internal clock synchronization algorithms are purely software based and usually run on COTS networking hardware, providing a precision in the $10ms$ range. With dedicated hardware support a time service with precision in the $10\mu s$ range can be implemented, cf. [6]. Even smaller precisions can be achieved when a separate fully connected clocking network is used [8]. We do not further consider such pure hardware based solutions because of their limited use and practicality for large scale distributed systems.

Concerning external clock synchronization, the most widely used scheme is the *Network Time Protocol* designed for disseminating UTC throughout the Internet. Worst case accuracies of approximately $20ms$ were observed [15]. Other solutions addressed to the external synchronization problem are related to the LAN domain, e.g., [2] or [16]. The latter "sprays" external time obtained from GPS satellites into broadcast-type LANs with accuracies in the $10\mu s$ range. With moderate hardware support developed along with our SynUTC[1] project, we recently obtained results in the $1\mu s$ range for both precision and accuracy, see [10] and [11] for further details.

The on-going work presented in the remainder of this article exploits some shortcomings we encountered during our experimental evaluation. In fact we are reasonably convinced that the new architecture presented here will enable a worst case precision and accuracy in the $10ns$ range.

II. Related Work

Fetzer and Cristian derived an optimal lower bound on the maximum deviation for convergence function based internal clock synchronization algorithms, see [1]. This bound is given by $4\Lambda + 4\rho r_{max}$, with Λ denoting the maximum clock reading error of the local and remote clocks, ρ the maximum drift rate of correct hardware clocks, and r_{max} the maximum delay between two successive adjustments of a correct clock. The same authors investigated in [2] the integration of internal and external clock synchronization. Their algorithm is able to bound the clocks of two nodes in a distributed system by $4\Lambda + 9\rho r_{max} + 2\rho\beta$, where the additional factor β gives the initial maximum deviation between the clocks of two correct nodes. The maximum external deviation to the external time-reference is given by the sum of $\Delta + \Lambda + \rho r_{max}$, where Δ accounts for the maximum external deviation of correct reference clocks.

Similar results are given by algorithms developed in the SynUTC project, see [9] and [13]. The results obtained therein are suitable for high-accuracy clock synchronization due to the very detailed system model, which incorporated several non-standard issues like non-zero clock granularity and broadcast latencies. The results revealed that — apart from clock reading error, clock drift, re-synchronization period and external deviation — the clock granularity and rate adjustment uncertainty have an impact on the achievable precision and accuracy.

M. Horauer, currently working on his PhD thesis, is with the Institute of Computer Technology, Technische Universität Wien, Gußhausstraße 25-27, A-1040 Vienna, Austria. E-mail: horauer@ict.tuwien.ac.at WEB: http://www.ict.tuwien.ac.at/horauer

[1]The SynUTC-project received support from the Austrian Science Foundation (FWF) grant P10244-ÖMA, the OeNB "Jubiläumsfonds-Projekt" 6454, the BMfMV research contract Zl.601.577/2-iV/B/9/96, the Gesellschaft für Mikroelektronik (GMe), and the START programme Y41-MAT. See http://www.auto.tuwien.ac.at/Projects/SynUTC/ for further information.

Pure software-based clock synchronization approaches incorporate the medium access uncertainty at the sending node, any variable network delay and the reception interrupt latency. The first one can be quite large for any network utilizing a shared medium, and the last one is seriously impaired by code segments with interrupts disabled. Typical values for the clock reading error Λ reside in the range of several $100\mu s - ms$. The maximum drift rate of ordinary oscillators used in most systems is in the order of 10^{-5} or 10^{-6}, with a given small enough re-synchronization period r_{max}, this parameter has a more or less significant influence. The initial maximum deviation β between clocks requires extra consideration at system startup and during node join. Finally, Δ depends on the the given access facilities and the provision of an external time source. For the given GPS technology this may be in the range of several $10 - 100ns$.

In our research project we developed a dedicated hardware module termed Network Time Interface (NTI) for COTS-CPUs that provides precision in the range of $1\mu s$. This is achieved with a tight coupling of the local clock maintained within our *Universal Time Coordinated Clock Synchronization Unit* (UTCSU) — a dedicated application specific integrated circuit — and an Ethernet controller. The used architecture allows to reduce the remote clock reading error [4]. This is done by inserting time-stamps in hardware when the network controller grabs a packet from the memory for transmission or when a received packet is written to the memory respectively. With the help of this mechanism we can avoid given medium access uncertainties and reception interrupt latencies. Furthermore with timestamping packets at both the sending and receiving node we can even account for variable network delays thus reducing reading errors to a minimum. A rigorous evaluation of our system verified the bounds on precision and accuracy in the range of $1\mu s$. A careful data analysis complementing our experiments revealed the internal packet FIFO's of the network controller to be a limiting factor for even tighter synchronization, see [12].

III. Network Interface Hardware Architecture

The results given in [1], [2], [9] and [13] as well as our previous made experimental results led to the identification of the following items that need to be addressed when trying to minimize the precision and accuracy for both external and internal clock synchronization in a distributed application:

- The local clock and the mechanisms required for time-stamping messages should be coupled very tight to the network medium, in particular the physical network layer. This in turn allows to reduce the variability of the remote clock readings.
- The local oscillator that paces the hardware clock should exhibit only a very small oscillator drift and good stability, see [5]. Therefore an ovenized oscillator (OCXO) with suitable characteristics is preferable.
- The local hardware clock should be fine grained and provide mechanisms for both clock state and rate corrections.

The prototype node architecture tailored for the hardware support of clock synchronization in distributed systems, that addresses these requirements, is shown in Figure 1. This implementation employs a *Media Independent Interface*-based time-stamping method that can be used in conjunction with almost every modern 10/100 Mb/s Ethernet chipset. It exploits the fact that almost all Fast Ethernet controllers support the standard IEEE 802.3-compliant *Media Independent Interface* (MII) to physical layer devices.

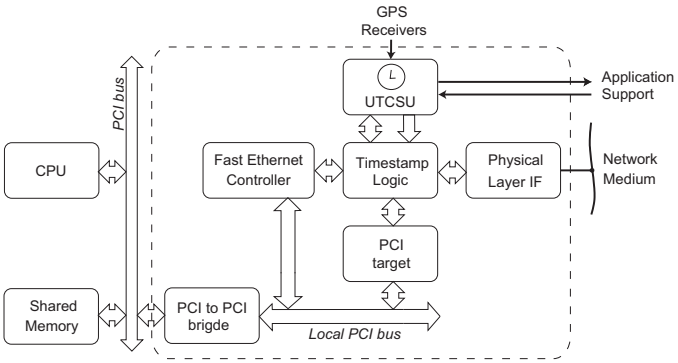

Fig. 1. Prototype Node Architecture

Before a packet can be transmitted it must be assembled by a local host CPU and stored in some Shared Memory. Then the CPU notifies the network controller that a packet should be transmitted. The latter in turn acquires the network channel and succinctly following grabs the packet from the Shared Memory and pushes the data stream via a Physical Layer Interface serialized onto the network medium.

The *Timestamp Logic*, placed into the data path between the network controller and the physical layer device, recognizes packets used solely for clock synchronization by means of a special type field and inserts local time+accuracystamps into the outgoing and incoming data respectively. There are of course some additional technical additions involved since the modification of the data stream requires to readjust the included checksum.

Local clock+accuracy information is maintained in the UTCSU Asic described in [14], which is paced by a suitable ovenized oscillator. The UTCSU hosts an unconventional *adder-based clock* design instead of the usual employed counter approach for summing up the elapsed time between succeeding oscillator ticks. This local clock is fine-grained rate adjustable and supports state adjustments via continuous amortization as well as (optional) leap second insertion in hardware. Two more adder-based clocks are used to support interval-based clock synchronization, see [13]. They are responsible for holding and automatically deteriorating the accuracy information, thereby maintaining a bound on the local clock's instantaneous deviation w.r.t. real-time. The UTCSU provides next to some application specific modules several interfaces that facilitate coupling to external time-sources, e.g., GPS-receivers. All

these mechanisms are complemented with several built-in test features that rely on signatures, block-sums and checksums.

The CPU that executes the clock synchronization software needs access to both the Fast Ethernet controller for configuration purposes and the UTCSU for maintaining and accessing the local clock+accuracy. Since most Fast Ethernet controllers are usually equipped with an integrated PCI interface, we chose PCI as our local bus and use a *PCI Target* chip for mediating accesses to the UTCSU. The additional *PCI-to-PCI Bridge* is required due to the PCI bus specification, since our prototype is implemented as PCI card and hosts more than one PCI device.

The current architecture enables us to access and programm the UTCSU registers either via the PCI Target device or via network data packets. The latter will eliminate the need for a PCI target chip and the PCI-to-PCI bridge, will allow re-use of existing network controller device drivers, and will finally open up many interesting possibilities for remote clock synchronization.

This hardware support mechanism allows us to reduce the remote clock reading error Λ to some ns by even avoiding uncertainties impaired by the inherent FIFOs embedded in network controllers. The influence of drift rate ρ and granularity on the precision+accuracy can be reduced by employing a suitable ovenized oscillator. Finally our UTCSU Asic allows for very fine grained clock adjustment.

IV. Conclusions and Future Work

This article presented an overview of the hardware support used for the distributed system in our SynUTC project. The architecture of our nodes facilitate fault-tolerant external clock synchronization even in Ethernet-based distributed systems. The new Media Independent Interface-based timestamping method will allow to improve the achievable precision+accuracy of shared-media based clock synchronization mechanisms by the order of several magnitudes drastically reducing the remote clock reading error.

We are currently in the run to develop the entire software that will be embedded in an *add-on* of the industrial multiprocessing/multitasking real-time kernel OSE (ENEA OSE Systems, Inc.). In addition we are planning an implementation targeting RTLinux as well. After the successful test of the first prototypes we will perform a long-term system evaluation by coupling a set of nodes with an atomic clock.

Acknowledgments

The author would like to acknowledge the suggestions and vital support from Ulrich Schmid.

References

[1] C.Fetzer and F.Cristian, *Lower bounds for convergence function based clock synchronization*, In Proceedings of the 14^{th} ACM Symposium on Principles of Distributed Computing, Ottawa - California, Aug. 1995, pp. 137–143.

[2] C.Fetzer and F.Cristian, *Integrating External and Internal Clock Synchronization*, Journal of Real-Time Systems, May 1997, No. 3, Vol. 12 (2), pp. 123–172.

[3] M.Horauer, N.Kerö and U.Schmid, *A network interface for highly accurate clock synchronization*, Proceedings Austrochip 2000, Graz - Austria, October 2000, pp. 93-101.

[4] M.Horauer, U.Schmid and K.Schossmaier, *NTI: A Network Time Interface M-Module for High-Accuracy Clock Synchronization*, Proceedings of the 6th International Workshop on Parallel and Distributed Real-Time Systems (WPDRTS), Orlando Florida USA, March 30 – April 3 1998.

[5] K.Schossmaier, *Interval-based Clock State and Rate Synchronization*, PhD Thesis, Technische Universität Wien, Faculty of Technical and Natural Sciences, Dept. of Automation, September 1998.

[6] H.Kopetz and W.Ochsenreiter, *Clock Synchronization in Distributed Real-Time Systems*, IEEE Transactions on Computers, C-36 (8), Aug. 1987, pp. 933–939.

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

[8] P.Ramanathan, D.D.Kandlus and R.W:Butler, *Fault-Tolerant Clock Synchronization in Distributed Systems*,

[9] U.Schmid, *Orthogonal Accuracy Clock Synchronization*, Chicago Journal of Theoretical Computer Science 2000(3), 2000, pp. 3–77. IEEE Computer, 1990, Vol. 23 (10), pp. 33–42.

[10] U.Schmid, M.Horauer and N.Kerö, *How to Distribute GPS-Time over COTS-based LANs*, 31st Annual Precise Time and Time Interval (PTTI) Systems and Applications Meeting, Dana Point - California, December 7-9 1999.

[11] U.Schmid, J.Klasek, Th.Mandl, H.Nachtnebel, G.R.Cadek and N.Kerö, *A Network Time Interface M-Module for Distributing GPS-time over LANs*, Journal of Real-Time Systems, Jan. 2000, No. 1, Vol. 18, pp. 24–57.

[12] U.Schmid and H.Nachtnebel, *Experimental Evaluation of High-Accuracy Time Distribution in a COTS-based Ethernet LAN*, Proceedings of the 24th IFAC/IFIP Workshop on Real-Time Programming (WRTP'99), Schloß Dagstuhl, Germany, May/June, 1999, pp. 59-69.

[13] U.Schmid and K.Schossmaier, *Interval-based clock synchronization* Journal of Real-Time Systems, May 1997, No. 3, Vol. 12 (2), pp. 173–228.

[14] K.Schossmaier, U.Schmid, M.Horauer and D.Loy, *Specification and Implementation of the Universal Time Coordinated Synchronization Unit (UTCSU)*, Journal of Real-Time Systems, May 1997, No. 3, Vol. 12 (1), pp. 295–327.

[15] G.D.Troxel, *Time Surveying: Clock Synchronization over Packet Networks*, PhD thesis, Massachusetts Insitute of Technology, Departement of Electrical Engineering and Computer Sciene, 1994.

[16] P. Veríssimo, L. Rodrigues, and A. Casimiro, *Cesiumspray: a precise and accurate global clock service for large-scale systems*, Journal of Real-Time Systems, May 1997, No. 3, Vol. 12 (1), pp. 241–294.