



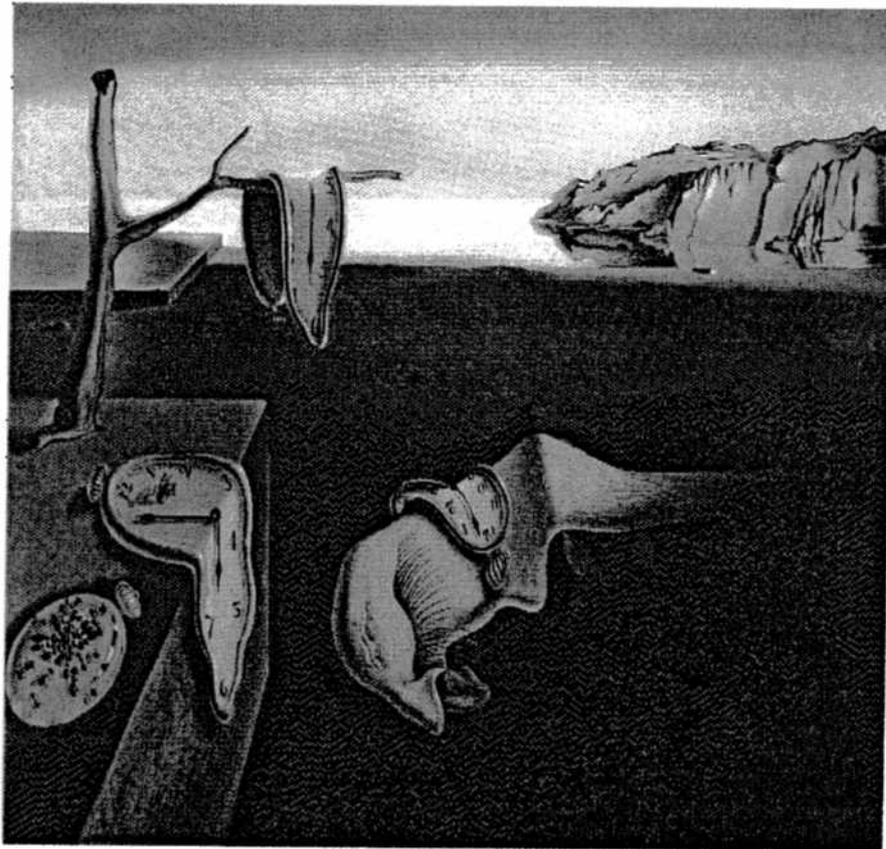
Institut für Automation  
Abt. für Automatisierungssysteme

Technische  
Universität  
Wien

Projektbericht Nr. 183/1-117  
October 2001

# Hochgenaue Uhrensynchronisation in Verteilten Systemen

*Ulrich Schmid, Martin Horauer, Nikolaus Kerö*



Salvador Dali, "Die Beständigkeit der Erinnerung"

# Hochgenaue Uhrensynchronisation in Verteilten Systemen

MARTIN HORAUER

Technische Universität Wien  
Department of Computer Technology  
Gußhausstraße 27–29, A-1040 Vienna  
martin.horauer@tuwien.ac.at  
<http://www.ict.tuwien.ac.at/horauer>

NIKOLAUS KERÖ

Oregano Systems  
Phorusgasse 8, A-1040 Vienna  
support@oregano.at  
<http://www.oregano.at>

ULRICH SCHMID

Technische Universität Wien  
Department of Automation  
Treitlstraße 1, A-1040 Wien  
s@auto.tuwien.ac.at  
<http://www.auto.tuwien.ac.at/Projects/SynUTC>

22. Oktober 2001

## Zusammenfassung

Die im Rahmen unseres Forschungsprojekts SynUTC<sup>1</sup> an der Technischen Universität Wien entwickelte Hard- und Software erlaubt es, die lokalen Uhren der Rechner-Knoten eines netzwerkgekoppelten verteilten Systems hochgenau untereinander bzw. auf Weltzeit unter ausschließlicher Verwendung des vorhandenen Netzwerkes zu synchronisieren. Synchronisationsgenauigkeiten bis in den 10 ns-Bereich, exzellente Fehlertoleranzeigenschaften, weitgehende Flexibilität bezüglich der verwendeten Hardware- und Netzwerktechnologie sowie wohldefinierte theoretische Grundlagen sind Charakteristika der entwickelten “enabling technology”, mit deren Hilfe völlig neue Anwendungsgebiete erschlossen werden können.

## 1 Motivation

“Mangelnde Zusammenarbeit zwischen Universität und Industrie” — eine Schlagzeile, deren Wahrheitsgehalt und Aktualität neben wirtschaftspolitisch interessierten Zeitungslesern auch die unmittelbar Betroffenen allmählich zu realisieren beginnen. Die immer stärker werdende Tendenz, wissenschaftliche Institute zu Entwicklungsabteilungen für Industrieaufträge umzufunktionieren, ist allerdings eher eine besorgniserregende Fehlentwicklung denn eine zukunftssträchtige Lösung, da die Schaffung von Entwicklungskompetenz mittelfristig zu Lasten der Forschungskompetenz gehen muß.

---

<sup>1</sup>Das Projekt SynUTC wurde 1995 als FWF-Projekt am Institut für Automation der TU-Wien gestartet und in Kooperation mit dem Institut für Computertechnik abgewickelt. 1998 wurde im Zuge der Prototyp-Entwicklung eine weitere Kooperation mit dem Institut für Allgemeine Elektrotechnik und Industrielle Elektronik aufgebaut, die letztlich in unserer Spin-off Firma Oregano Systems mündete. Oregano Systems ist nun unter anderem für die Weiterentwicklung und Kommerzialisierung unserer Projektergebnisse zuständig. Das Projekt SynUTC wurde unterstützt vom Österreichischen Fonds zur Förderung der Wissenschaftlichen Forschung (FWF) unter Projektnummer P10244-ÖMA, dem “Jubiläumsfonds-Projekt” 6454 der Österreichischen Nationalbank, der Auftragsforschung ZI.601.577/2-iV/B/9/96 und dem START-Programm Y41-MAT (U. Schmid) des Österreichischen Bundesministeriums für Wissenschaft und Forschung, und der Gesellschaft für Mikroelektronik (GMe). Nähere Informationen sind in Abschnitt 6 bzw. unter <http://www.auto.tuwien.ac.at/Projects/SynUTC/> zu finden.

In der von einem der Autoren geleiteten Arbeitsgruppe für verteilte Systeme am Institut für Automation der TU-Wien wird daher ganz bewußt ein anderer Weg hinaus aus dem “elfenbeinernen Turm” gegangen [Sch00a]: Unbeschadet ihres anspruchsvollen wissenschaftlichen “Innenlebens” zielen die dort abgewickelten—selbstgewählten—Forschungsprojekte primär darauf, eine wohlfundierte Lösung eines anwendungsnahen Problems zu erarbeiten und deren tatsächliche Tauglichkeit mit Hilfe einer Prototyp-Implementierung zu demonstrieren. Erst durch einen evaluierbaren Prototypen wird es für den potentiellen Anwender ja möglich, die Brauchbarkeit einer Lösung und den Aufwand für die konkrete Adaptierung realistisch einzuschätzen; dies allein auf Basis noch so guter wissenschaftlicher Publikationen bewerkstelligen zu wollen, ist erfahrungsgemäß so gut wie aussichtslos.

Das gegenständliche Projekt SynUTC (*Synchronized Universal Time Coordinated for Fault-Tolerant Distributed Systems*) fällt in diese Kategorie von Forschungsprojekten. Es ist der Fragestellung gewidmet, wie—ohne dedizierte GPS-Empfänger oder Zusatzverkabelung heranzuziehen—die Knoten eines netzwerkgekoppelten, fehlertoleranten Rechnersystems mit hochgenau untereinander und auf Weltzeit synchronisierten lokalen Uhren ausgerüstet werden können. Unbeschadet der komplexen theoretischen und konzeptuellen Grundlagen, die in zahlreichen wissenschaftlichen Arbeiten publiziert wurden, ist das eigentliche Endergebnis des Projekts eine “vorzeigbare” Prototyp-Implementierung von Hard- und Software, die von unserer Spin-off Firma Oregano Systems professionell aufbereitet wird.

Die vorliegende Bewerbung um den Phillip Morris-Preis soll dazu beitragen, die SynUTC-Projektergebnisse einer breiteren Öffentlichkeit vorzustellen und somit weitere Anwendungsmöglichkeiten zu erschließen. Da die Forschungsarbeiten in SynUTC ohne Bindung an einen Industriepartner durchgeführt wurden, sind wir in der Lage, unsere Technologie im Prinzip jedem Interessenten zur Verfügung stellen zu können.

## 2 Anwendungsgebiete

Im Zuge der Projektarbeit an SynUTC wurden bereits eine ganze Reihe möglicher Anwendungen für unsere Technologie identifiziert: Zunächst einmal erlauben synchronisierte Uhren die verteilte Ausführung zeitlich koordinierter Aktionen bzw. die verteilte Erfassung korrelierter Ereignisse. Dies ist eine notwendige Voraussetzung für die dezentrale Automatisierung [Loy97], wo z.B. die (ABS-)Bremsen der vier Räder eines Autos nicht mehr von einem zentralen Computer angesteuert werden, sondern vielmehr von vier netzwerkgekoppelten dedizierten Mikroprozessoren.

Darüberhinaus ist es in der wissenschaftlichen Literatur wohlbekannt, siehe z.B. [Lis93], daß synchronisierte Uhren zur Erhöhung der Leistung verteilter Systeme verwendet werden können. Dies ist deswegen möglich, da das explizite Versenden bestimmter Information in manchen Fällen durch das Nichtsenden einer—zu einem bestimmten Zeitpunkt erwarteten—Nachricht ersetzt werden kann (“communication by time”).

Neben diesen “generischen” Anwendungen gibt es auch sehr spezifische Anwendungsmöglichkeiten von synchronisierter Uhrzeit, wie etwa die Erzeugung zuverlässiger *Nonces* (eindeutiger “Zufallswerte”) in Authentisierungsprotokollen, das Quality-of-Service-Monitoring in Voice-over-IP Netzen, oder die Synchronisation von Video-Datenströmen. Ein besonders vielversprechendes Anwendungsgebiet ist schließlich die genaue Ortsbestimmung, die sowohl in drahtgebundenen als auch in drahtlosen Systemen immer wichtiger wird. So verbessert z.B. die genaue Kenntnis der Position der Knoten in einem Feldbussystem die Möglichkeiten der Konfiguration und Diagnose ganz entscheidend. Zwei weitere, sehr konkrete Anwendungen in diesem Bereich sind:

- *Ortung von Mobilfunk-Endgeräten (Location based services)*

Moderne Mobilfunknetze bieten prinzipiell die Möglichkeit der präzisen Ortung jedes aktiven Teilnehmers. Wenn alle Basis-Stationen eines zellularen Netzwerks über hochgenau zueinander synchronisierte Uhren verfügen würden, so können die 3D-Koordinaten eines Teilnehmers über die Zeitdifferenzen des Empfangs des Signals an drei verschiedenen Basis-Stationen berechnet werden.

Da die Basis-Stationen üblicherweise mittels leistungsfähiger drahtgebundener Netzwerke miteinander verbunden sind, stellt unsere Technologie eine interessante Alternative zum Einsatz dedizierter GPS-Empfänger in allen Basis-Stationen dar. Allerdings erfordert diese Anwendung Maßnahmen, die die Kompatibilität unseres Verfahrens mit Netzwerk-Switches garantieren; für diese haben wir ein internationales Patent angemeldet [KSH00].

Die Bedeutsamkeit derartiger Verfahren zeigt ein Erlass der U.S. Federal Communications Commission, welcher von mobilen Telekommunikationsdiensten ab dem 31. Oktober 2001 die automatische Ortung von Notrufen mit einer Genauigkeit von 125 m in 67% aller Fälle verlangt. Aber auch rein kommerzielle Dienste wie beispielsweise eine interaktive Adressensuche (Wegweiser) im Stadtgebiet werden derzeit von zahlreichen Mobilfunk-Netzbetreibern entwickelt.

- *Kabelfehler-Ortung*

Ein massives Problem, mit dem sich Energieversorgungsunternehmen konfrontiert sehen, ist die genaue Ortung von Fehlerstellen in Hochspannungs-Erdkabeln (und auch Freileitungen). Solche Fehlerstellen entstehen durch Alterung des Isolationsmaterials oder auch mechanische Beschädigungen und verursachen Kurzschlüsse bzw. Unterbrechungen. Da das Kabel zur Reparatur an der Fehlerstelle ausgegraben werden muß, erfordert die rasche Fehlerbehebung zuallererst eine genaue Ortung.

Eine Möglichkeit dazu besteht darin, an allen Kabelenden in Umspannwerken und Transformatorstationen Detektoren anzubringen, die die bei einem Kurzschluß oder einer Unterbrechung von der Fehlerstelle ausgehende Wanderwelle erfassen. Wenn nun die Detektoren über hochgenau synchronisierte Uhren verfügen, so kann über die Zeitdifferenz des Eintreffens der Wanderwelle an beiden Kabelenden der Fehlerort berechnet werden. Angesichts der relativ großen Anzahl von Kabelenden in einem Umspannwerk und der stark störenden Umgebung ist es hier nicht möglich, jeden Detektor mit einem eigenen GPS-Empfänger auszurüsten. Die von SynUTC benötigte Mikroprozessoreinheit mit einem Glasfaser-Netzwerk ist hingegen sehr leicht zu realisieren. Für diese Anwendung existiert bereits eine deutsche Gebrauchsmusteranmeldung [Sch97b].

Die Anwendungsnähe eines Forschungsprojekts impliziert aber natürlich keineswegs die Kenntnis aller möglichen Einsatzgebiete. Wie z.B. die ungeahnte Verbreitung des *Global Positioning Systems* GPS gezeigt hat, stimuliert eine "enabling technology" sehr oft neue Anwendungen — wenn sie einmal verfügbar ist.

### 3 Charakteristiken der Uhrensynchronisation in SynUTC

Anders als bei herkömmlichen Verfahren, die auf einer zentralen Uhr basieren, ist jeder SynUTC-Knoten mit einer eigenen lokalen Uhr ausgerüstet, die allein durch den Austausch von Synchronisationsnachrichten über das verbindende Datennetzwerk mit den anderen Uhren im System hochgenau synchronisiert wird. Der (optionale) Bezug zur Weltzeit wird dadurch hergestellt, daß an mindestens einem der Netzwerkknoten ein GPS Satellitenempfänger mit 1 pps-Zeitausgang angeschlossen

sein muß. Im Gegensatz zu existierenden Verfahren ist es hier jedoch nicht notwendig, jeden Knoten mit einem dedizierten GPS-Empfänger auszurüsten oder Zeitinformation über zusätzliche Verbindungsleitungen zu verteilen.

Anders als im Falle dedizierter GPS-Empfänger erfordert unser Verfahren daher weder einen "Wald" von GPS-Antennen noch ist es den möglichen Fehlern eines GPS-Empfängers [Höc96, HS97] hilflos ausgeliefert. Ein weiterer Nachteil dedizierter GPS-Empfänger Lösungen ist die lange Dauer von bis zu 15 Minuten (time-to-fix von GPS), bis eine lokale Uhr nach dem Einschalten der Versorgungsspannung synchronisiert ist. Ein detaillierter Vergleich von SynUTC mit alternativen Uhrensynchronisationsverfahren wie etwa [KO87, KKMS95] ist in [SKM<sup>+</sup>00] zu finden.

Die im Rahmen von SynUTC entwickelte Hard- und Software basiert auf theoretisch wohl fundierten Grundlagen und kann durch umfangreiche mathematisch-formale Analysen den Anspruch erheben, die folgenden Leistungsmerkmale —von möglichen Implementierungsfehlern abgesehen— auch garantiert zu erbringen:

- Überbrückung von LAN-Distanzen unter alleiniger Verwendung des vorhandenen Datennetzwerks. Die Eignung einer bestimmten Netzwerktechnologie hängt dabei nicht vom deterministischen Anteil der Nachrichtenverzögerungszeit  $\delta$ , der automatisch gemessen wird, ab, sondern lediglich vom maximalen *Delay-Jitter*  $\varepsilon = \delta_{\max} - \delta_{\min}$ . Unsere spezielle Zeitstempel-Hardware erlaubt es,  $\varepsilon$  bis auf wenige 10 ns zu reduzieren, sogar in Fast-Ethernet-Netzwerken mit Switches.
- Hohe Synchronisationsgenauigkeit der lokalen Uhren zueinander ("Präzision"), die vom Delay-Jitter  $\varepsilon$  des zugrundeliegenden Netzwerks sowie der Stabilität und der Taktfrequenz der lokalen Quarzoszillatoren abhängt.
- Durch die Synchronisation der Ganggeschwindigkeiten der lokalen Uhren ("Rate Synchronisation") können auch billige Quarzoszillatoren mit beschränkter Genauigkeit verwendet werden.
- Im Normalfall hohe Synchronisationsgenauigkeit der lokalen Uhren zur Weltzeit ("Genauigkeit"), auch bei Knoten ohne GPS-Empfänger. Darüberhinaus Bereitstellung eines zuverlässigen worst-case Genauigkeitsintervalles.
- Exzellente Fehlertoleranz-Eigenschaften, v.a.
  - Die Qualität Uhrensynchronisation wird durch fehlerhafte Rechner/Uhren nicht beeinträchtigt.
  - Ein (längerer) Ausfall oder ein Fehlverhalten von GPS-Empfängern verschlechtert lediglich die Genauigkeit in bezug auf die externe Zeit, nicht aber die interne Präzision.

Diese Eigenschaften werden durch die Kombination von fehlertoleranten Uhrensynchronisationsalgorithmen (Software) mit hochgenauer Zeitstempelung von ankommenden und abgehenden Datenpaketen (Hardware) sichergestellt. Das Bild 1 zeigt die wichtigsten Komponenten eines Systems mit zwei Knoten.

Ein für die SynUTC-Technologie geeigneter Knoten muß über folgende Hardware-Komponenten verfügen:

- Standard *CPU* für die regelmäßige Ausführung der Synchronisationsalgorithmen (die jedoch ohne Schwierigkeiten parallel zu anderen Tasks exekutiert werden können).
- Standard *Netzwerk-Controller*.
- *UTCSU*, ein integrierter Schaltkreis, der v.a. die lokale Uhr und diverse Zeitstempelmechanismen bereitstellt (siehe Abschnitt 5).

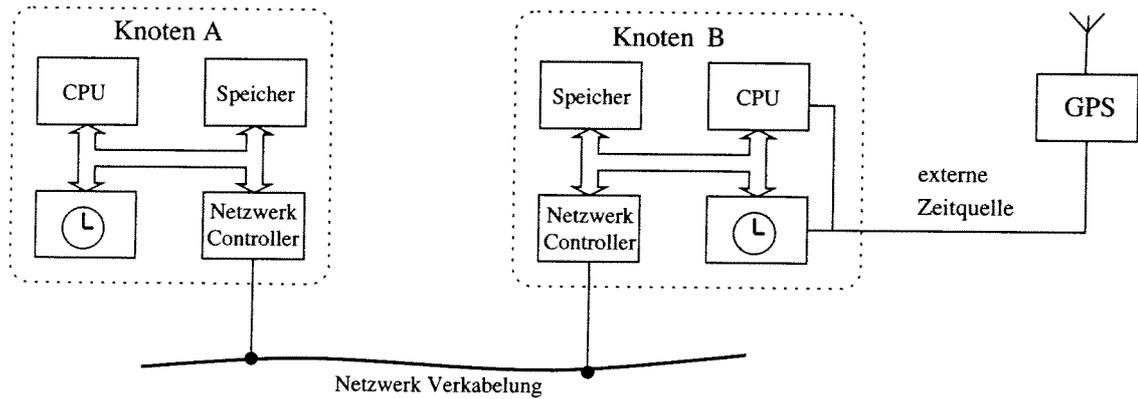


Abbildung 1: Zwei Knoten eines verteilten Systems

- *Paketzeitstempel-Einheit*, die (Speicher- oder Transceiver-basierend) die Zeitstempelung von ankommenden und abgehenden Datenpaketen ermöglicht.

Für die auf SynUTC aufbauenden Anwendungen sind—neben der Möglichkeit, jederzeit die aktuelle Uhrzeit per Software auslesen zu können—vor allem die vielfältigen Zeitstempel-Funktionen in der UTCSU interessant: Zum Zeitpunkt der Zustandsänderung eines digitalen Eingangs wird der Stand der lokalen Uhr in ein dediziertes Register gespeichert, wodurch ein solcher Zeitstempel (zeitlich relativ unkritisch, aber natürlich noch vor der nächsten Zustandsänderung) per Software ausgelesen und weiterverarbeitet werden kann. Weiters stehen Möglichkeiten zur Auslösung von Interrupts bzw. eines Pulses an einem digitalen Ausgang bei Erreichen eines bestimmten Zeitpunkts zur Verfügung.

## 4 SynUTC Software

Die gesamte im Rahmen von SynUTC entwickelte Hard- und Software basiert auf dem *intervallbasierenden Paradigma*, in dem die Zeit  $t$  auf einem Knoten  $p$  nicht wie üblich durch eine punktförmige Uhrzeit  $C_p(t)$ , sondern durch ein *Zeitintervall*  $\mathbf{A}_p(t) = [C_p(t) - \alpha_p^-(t), C_p(t) + \alpha_p^+(t)]$  repräsentiert wird. Zeitintervalle werden von der lokalen *Intervalluhr* eines Knotens geliefert, die mittels einer gewöhnlichen Uhr  $C_p(t)$  und einem (zeitvarianten) Genauigkeitsintervall  $[-\alpha_p^-(t), \alpha_p^+(t)]$  realisiert werden. Abbildung 2 zeigt das Prinzip.

Die Aufgabe intervallbasierender Uhrensynchronisationsalgorithmen ist es, die Intervalluhren so zu steuern, daß für alle fehlerfreien Knoten  $p$  und  $q$  folgendes garantiert werden kann:

- (I) *Interne Synchronisation*:  $|C_p(t) - C_q(t)| \leq \pi_{\max}$  für eine fix vorgegebene *Präzision*  $\pi_{\max} \geq 0$ .
- (E) *Externe Synchronisation*:  $t \in \mathbf{A}_p(t)$ , wobei das Genauigkeitsintervall  $[-\alpha_p^-, \alpha_p^+]$  minimal sein soll.

Praktisch wird dies so gelöst, daß alle Knoten periodisch ihr momentanes Zeitintervall an alle anderen Knoten versenden und auf Basis der eingegangenen Informationen einen Korrekturwert für die eigene Uhr berechnen.

Der Hauptvorteil des intervallbasierenden Ansatzes ist das (lokal verfügbare) Genauigkeitsintervall, das die maximale Abweichung der lokalen Uhrzeit  $C_p(t)$  von der wahren Zeit  $t$  angibt:

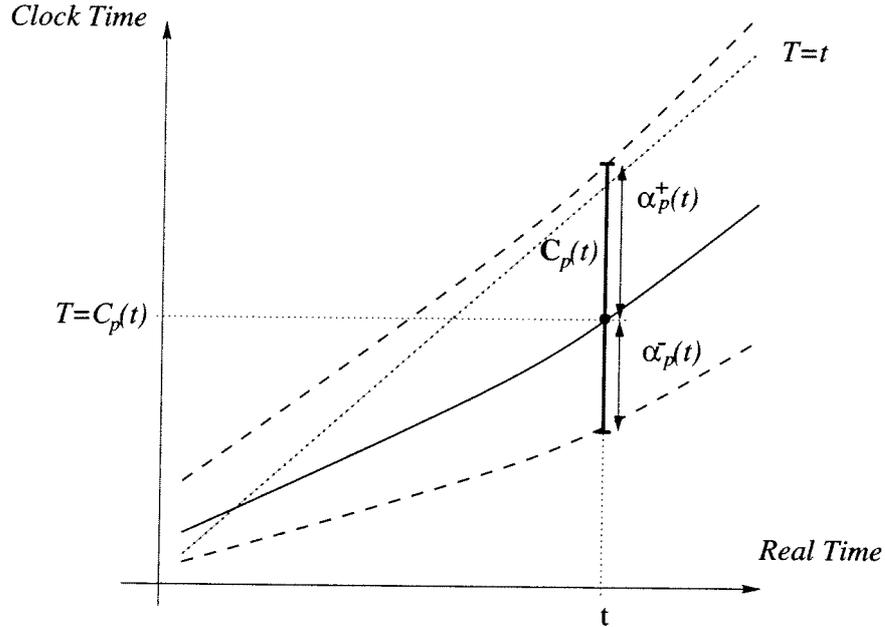


Abbildung 2: Prinzip einer Intervalluhr und der Korrektheitsbedingung  $t \in \mathbf{A}_p(t)$

Aufgrund der externen Synchronisationsbedingung  $t \in [C_p(t) - \alpha_p^-(t), C_p(t) + \alpha_p^+(t)]$  kann eine Anwendung über die Intervalllänge  $|\mathbf{A}_p(t)|$  entscheiden, ob die momentane Genauigkeit für bestimmte Zwecke ausreicht. Dies erleichtert grundsätzlich die externe Synchronisation und ist besonders für Anwendungen in sehr großen und weitverteilten (“clustered”) Systemen interessant, wo die interne Synchronisation aller Knoten wegen des akkumulierten Delay-Jitters  $\varepsilon$  und des hohen Nachrichtenaufkommens nur mehr bedingt möglich ist: Für zwei korrekte Knoten  $x, y$  in verschiedenen Clustern impliziert die externe Synchronisationsbedingung, daß ihre Uhrzeiten  $C_x(t), C_y(t)$  höchstens um  $-(\alpha_x^-(t) + \alpha_y^+(t)) \leq C_y(t) - C_x(t) \leq \alpha_y^-(t) + \alpha_x^+(t)$  voneinander abweichen können, was somit eine wohldefinierte Präzision auch in diesem Fall garantiert.

Der Preis für diese Möglichkeiten ist allerdings die Bereitstellung expliziter Schranken für Clock-Drift  $\rho$  (= Gangungenauigkeit der lokalen Uhr) und Delay-Jitter  $\varepsilon$ , die entweder *a priori* bekannt sein oder aber dynamisch gemessen—im Falle der Clock-Drift sogar auch kontrolliert—werden müssen. Angesichts einer Synchronisationsgenauigkeit im 10 ns-Bereich ist es hier unumgänglich, eine genaue (Round-Trip-basierende) Messung der Nachrichtenverzögerungszeit durchzuführen und darüberhinaus die Clock-Drift durch eine geeignete *Rate Synchronization* zu kontrollieren: Die in [Sch97d, Sch98] vorgestellten intervallbasierenden Algorithmen haben die Aufgabe, die Ganggeschwindigkeiten aller Uhren im System aneinander anzugleichen. Dadurch ist es möglich, die maximale (relative) Clock-Drift klein zu halten, ohne daß dafür extrem stabile Oszillatoren für die lokalen Uhren eingesetzt werden müssen.

Innerhalb von SynUTC wurden zunächst alle formal/mathematischen Grundlagen für intervallbasierende fehlertolerante Uhrensynchronisation geschaffen [Sch95, SS97a, SS97b, Sch97d, Sch98] und geeignete Algorithmen entwickelt und analysiert [Sch00c, Sch97a, SHK99, SW99, SK99, SS01]. Als wichtigste Ergebnisse dieser Arbeiten sind, neben der konsequenten Ausarbeitung des intervallbasierenden Paradigmas und der Rate Synchronization, die Modellierung diskreter Uhren und der daraus resultierenden Granularitätseinflüsse zu nennen. Darüberhinaus konnten wir zeigen, daß einer unserer Algorithmen [SS01] Leslie Lamport’s 13 Jahre alte Frage nach einer fehlertoleranten

Intersection-Funktion mit Lipschitz-Eigenschaft beantwortet, was mit einiger Berechtigung als das wissenschaftlich bedeutendste Ergebnis des Projekte gewertet werden kann.

Die meisten unserer Algorithmen wurden auch implementiert und via Simulation mittels unseres SimUTC-Toolkits [WGSS99, SWGS99] getestet und/oder experimentell evaluiert [SHK99]. Fast alle Realisierungen basieren auf einem generischen Algorithmus [SS97a], in dem jeder Knoten  $p$  periodisch (mit Periode  $P$ ) die folgenden Schritte exekutiert:

1. Wenn  $C_p(t) = kP$ ,  $k \geq 1$ , wird ein Datenpaket mit dem momentanen Zeitintervall  $A_p(t)$  zu allen anderen Knoten im System gesendet.
2. Die hereinkommenden Datenpakete werden vorverarbeitet, um die darin enthaltenen Zeitintervalle "vergleichbar" zu machen.
3. Wenn  $C_p(t) = kP + \Lambda$  für ein hinreichend großes  $\Lambda > 0$ , so wird eine intervallwertige *Konvergenz-Funktion* auf die empfangenen Intervalle angewendet und die lokale Intervalluhr entsprechend korrigiert.

In Schritt 2, in dem die empfangenen Intervalle kompatibel gemacht werden, kommen zwei grundlegende Operation zur Anwendung: *Delay-Kompensation* wird verwendet, um der nicht konstanten Nachrichtenverzögerung Rechnung zu tragen; dies erfolgt durch eine dem maximalen  $\varepsilon$  entsprechende Vergrößerung der empfangenen Intervalle. *Drift-Kompensation* ist notwendig, um die —natürlich nicht zeitinvarianten— Intervalle mit Hilfe der lokalen Empfängeruhr in der Zeit zu verschieben, ohne die Korrektheitsbedingung (Einschluß von  $t$ ) zu verletzen. Die Intervalle müssen dabei kontinuierlich, entsprechend der maximalen Clock-Drift  $\rho$  der Empfängeruhr, vergrößert werden; dies ist natürlich auch während der (langen) Zeit zwischen dem Ende von Schritt 3 und dem nächsten Schritt 1 erforderlich. Bei allen Operationen müssen auch die vielfältigen Granularitätseinflüsse berücksichtigt werden, die insgesamt einen starken Einfluß auf die erreichbare worst-case Synchronisationsgenauigkeit haben, siehe z.B. [Sch00c].

In Schritt 3 wird schließlich eine geeignete *Konvergenz-Funktion* auf die Menge der empfangenen und vorverarbeiteten Zeitintervalle angewendet. Ihre Aufgabe ist es, daraus ein neues (kleineres) Intervall für die eigene lokale Uhr zu berechnen, das —systemweit gesehen— die Bedingungen (I) und (E) garantiert. Man beachte, daß die Fehlertoleranzeigenschaften des obigen Algorithmus ausschließlich von der Konvergenz-Funktion bestimmt werden, siehe [Sch00c, Sch97a, SS01].

## 5 SynUTC Hardware

Die im vorigen Abschnitt vorgestellten Uhrensynchronisationsalgorithmen operieren mit Intervallzeitgestempelten Datenpaketen und einer justierbaren Intervalluhr. Die hohe Synchronisationsgenauigkeit von SynUTC wird dadurch erreicht, daß sowohl die Justierung der Uhr als auch die Paketzeitstempelung nicht per Software, sondern per Hardware erfolgt.

### UTCSU-ASIC

Fast die gesamte Hardware-Unterstützung für die intervallbasierende Uhrensynchronisation wurde in Form eines eigenen integrierten Schaltkreises — *UTCSU-ASICs* (*Universal Time Coordinated Synchronization Unit*) [SSHL97, Loy96, HL97, HL98] — realisiert. Die UTCSU wurde von Atmel-ES2 im Rahmen von *Europractice* in  $0.7 \mu\text{m}$  CMOS gefertigt und enthält etwa 80,000 Gatter in einem 208-pin PQFP Gehäuse<sup>2</sup>. Abbildung 3 zeigt eine Übersicht über die wichtigsten Komponenten der UTCSU.

---

<sup>2</sup>Als Prototyp zur Verwendung in unserem Forschungsprojekt wurde versucht, möglichst viel an Funktionalität in das ASIC hineinzupacken; der von Oregano Systems gegenwärtig zu entwickelnde Nachfolger der UTCSU wird natürlich stark abgespeckt.

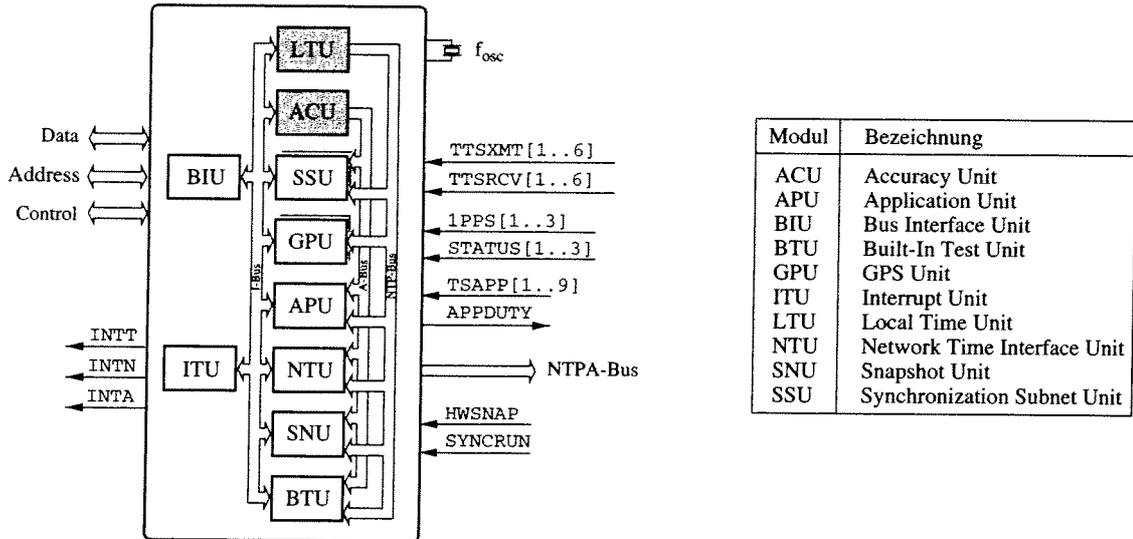


Abbildung 3: Funktionale Blöcke des UTCSU-ASICs

Die zentrale Komponente der UTCSU ist die *Local Time Unit* (LTU), die eine hochauflösende Uhr im 56-bit NTP-Zeitformat bereitstellt. Die Uhrzeit kann atomar als 32 bit *Timestamp* mit einer Auflösung von  $2^{-24} \approx 60 \text{ ns}$  und als 32 bit *Macrostamp* mit den restlichen 24 höherwertigen Bits und einer 8-bit Prüfsumme über die gesamte Zeitinformation gelesen werden. Die LTU basiert auf einem neuartigen *Adder-Based Clock* (ABC), der intern statt eines einfachen Zählers einen schnellen 91-bit Addierer für das Aufsummieren der Zeit zwischen je zwei Oszillator-Ticks verwendet.

Durch die Verwendung eines ABC kann die UTCSU mit beliebiger Oszillatorfrequenz  $f_{osc} \in 1 \dots 20 \text{ MHz}$  betrieben werden; insbesondere kann auch der 10 MHz-Frequenzausgang eines GPS-Referenzempfängers verwendet werden. Die so realisierte Uhr kann in Stufen von etwa  $10 \text{ ns/s}$  in ihrer Ganggeschwindigkeit justiert werden und erlaubt die Korrektur der Uhrzeit mittels automatischer *Continuous Amortization*<sup>3</sup> genauso wie das Einfügen von Schaltsekunden. Das Konzept des ABCs kann auf Grund dieser Eigenschaften somit als wissenschaftlich wichtigstes Ergebnis des "praktischen" Teils von SynUTC gewertet werden.

Zur Unterstützung der intervallbasierenden Uhrensynchronisation (siehe Abschnitt 4) enthält die UTCSU zwei weitere *Adder-Based Clocks* (ACU), die ebenfalls mit der Oszillatorfrequenz  $f_{osc}$  betrieben werden. Diese sind für die automatische Aufweitung der 16+16-bit Genauigkeitsintervalle  $[\alpha_p^-(t), \alpha_p^+(t)]$  zuständig, die die maximale Abweichung der momentanen lokalen Uhrzeit  $C_p(t)$  von der wahren Zeit  $t$  angeben.

Eine Vielzahl von externen Ereignissen, die der UTCSU über dedizierte Eingangsleitungen mit programmierbarer Flanke gemeldet werden, können atomar mit der momentanen Uhrzeit + dem Genauigkeitsintervall zeitgestempelt werden; zusätzlich kann dabei ein Interrupt ausgelöst werden. Drei verschiedene Funktionsblöcke in der UTCSU stellen derartige Mechanismen bereit: Insgesamt 6 SSUs stehen für die Paketzeitstempelung beim Senden und Empfangen von Datenpaketen zur Verfügung. Drei unabhängige GPUs können für das Zeitstempeln des *1 Pulse-per-Second* Signals (das den exakten Beginn einer GPS-Sekunde<sup>4</sup> meldet) von bis zu 3 GPS-Empfängern verwendet

<sup>3</sup>Die Korrektur einer Uhr darf in vielen Anwendungen nicht sprunghaft durch Setzen auf einen neuen Wert erfolgen. Die Uhr sollte vielmehr für eine kurze Zeit langsamer/schneller gestellt werden.

<sup>4</sup>Dieses einfache Interface genügt zur Ankopplung von GPS-Zeitempfängern, da die üblicherweise über eine serielle RS-232 Schnittstelle gelieferte Zusatzinformation nicht in der UTCSU verarbeitet werden muß.

werden. Schließlich bietet die APU neun unabhängige Intervallzeitstempel-Eingänge für Anwendungen.

Diese Zeitstempelfunktionen werden durch mehrere 48-bit programmierbare *Duty-Timer* ergänzt: Wenn die lokale Uhr die programmierte Zeit eines aufgesetzten Duty-Timers erreicht, wird dies durch den Zustandswechsel eines externen oder internen Signals gemeldet und optional ein Interrupt ausgelöst. Duty-Timer werden für das Triggern von Aktivitäten des Uhrensynchronisationsalgorithmus, für die Kontrolle der Continuous Amortization und dem Einfügen von Schaltsekunden, und schließlich für die Anwendungen benötigt.

## Paketzeitstempelung

Das wichtigste Problem bei der hochgenauen Uhrensynchronisation ist die exakte Zeitstempelung von Datenpaketen (*Clock Synchronization Packets, CSPs*) beim Sender und beim Empfänger. Dies ist eine Konsequenz des wohlbekannten Ergebnisses von [LWL84], das besagt, daß  $n$  Uhren nicht mit einer kleineren Präzision als  $\varepsilon(1 - 1/n)$  synchronisiert werden können, wenn  $\varepsilon$  die zeitliche Unsicherheit bei der Datenpaketübertragung (also den Delay-Jitter) angibt. Zur Abschätzung der Faktoren, die  $\varepsilon$  beeinflussen, kann folgender Ablauf der Übertragung von Datenpaketen in einem Netzwerk herangezogen werden:

1. Sender-CPU erstellt das Datenpaket
2. Sender-CPU beauftragt den Netzwerk-Controller (NC) mit dem Versenden des Pakets
3. Sender-NC versucht, das Netzwerk-Medium zu akquirieren
4. Sender-NC liest das Datenpaket aus dem Speicher und legt die resultierende Bitfolge auf das Medium
5. Empfänger-NC liest die Bitfolge vom Medium und schreibt das Datenpaket in den Speicher
6. Empfänger-NC teilt der CPU den Empfang eines Datenpaketes durch einen Interrupt mit
7. Empfänger-CPU verarbeitet das Datenpaket

Rein Software-basierende Ansätze der Uhrensynchronisation generieren die CSP-Zeitstempel beim Senden bzw. Empfangen in den Schritten 1 bzw. 7. Das bedeutet allerdings, daß  $\varepsilon$  voll die zeitliche Unsicherheit durch eventuelle Senderwarteschlangen (1→2), den Mediumszugriff (3→4), die physikalische Datenübertragung (4→5), und schließlich die Latenzzeit des Empfangsinterrupts (6→7) beinhaltet. Selbst ohne Senderwarteschlange kann nun die Unsicherheit bei Netzwerken mit einem "shared channel" (z.B. Ethernet) fast beliebig groß werden. Auch die Interrupt-Latenzzeit kann z.B. durch Code-Segmente mit abgeschalteten Interrupts fast beliebig lange verzögert werden.

Um den Delay-Jitter  $\varepsilon$  klein zu machen, muß die Paketzeitstempelung also so eng wie möglich an das Netzwerk gekoppelt werden, also im Schritt 4 bzw. 5 erfolgen. Hier verbleibt für  $\varepsilon$  dann nurmehr der Jitter bei der physikalischen Datenübertragung (und eventuelle Latenzzeiten beim Speicherzugriff des NC). Da bei unserem LAN-basierenden Ansatz davon ausgegangen werden kann, daß die Übertragungszeit am Medium praktisch konstant ist,<sup>5</sup> kann ein sehr kleines  $\varepsilon$  erreicht werden.

Eine Möglichkeit der Realisierung ist ein "Speicher-basierender" Ansatz [KO87], bei dem die Zeitstempelung dann erfolgt, wenn der Netzwerk-Controller das Datenpaket aus dem Speicher holt bzw. dort ablegt. Dies erfordert lediglich eine modifizierte Adreßdekodierungslogik für den Speicher, der bei bestimmten Zugriffen des Netzwerk-Controllers Triggersignale für die UTCSU-Zeitstempel erzeugt und dann die UTCSU-Register mit den abgezogenen Zeitstempeln transparent in den gerade vom NC zu lesenden Speicherbereich einblendet. Dadurch werden die Zeitstempel transparent in das

---

<sup>5</sup>Sofern keine Gateway-Knoten oder store-and-forward Switches vorhanden sind, oder wenn diese ebenfalls mit einem speziellen Zeitstempel-Mechanismus [KSH00] ausgestattet sind.

Datenpaket eingefügt. In der im Rahmen von SynUTC entwickelten Prototyp-Hardware [HSS97, HSS98, Man99, SN99, SKM<sup>+</sup>00] sind UTCSU-ASIC, Speicher und diverse Interface-Logik in Form eines *Network Time Interface* (NTI) M-Moduls [MUM96] realisiert.

Die Ergebnisse unserer experimentellen Evaluation [SN99, SKM<sup>+</sup>00] haben gezeigt, daß mit diesem—für praktisch alle existierenden Netzwerktechnologien verwendbaren—Ansatz ein  $\varepsilon$  im  $\mu\text{s}$ -Bereich erzielt werden kann. Allerdings machen die immer größer werdenden internen Speicher moderner Netzwerk-Controller diese Methode zunehmend problematisch.

Besser sollten die Zeitstempel am Sender bzw. Empfänger genau dann abgezogen werden, wenn beispielsweise das erste Datenbit auf das Medium gelegt bzw. vom Medium geholt wird. Dieser “Transceiver-basierende” Ansatz liegt unserem PCI-basierenden *MII Network Time Interface* (MII-NTI) für Fast-Ethernet [SHK99, HKS00, KSH00, Hor01] zugrunde, das ein  $\varepsilon$  im 10 ns-Bereich erreicht. Abbildung 4 zeigt ein Blockschaltbild der MII-NTI.

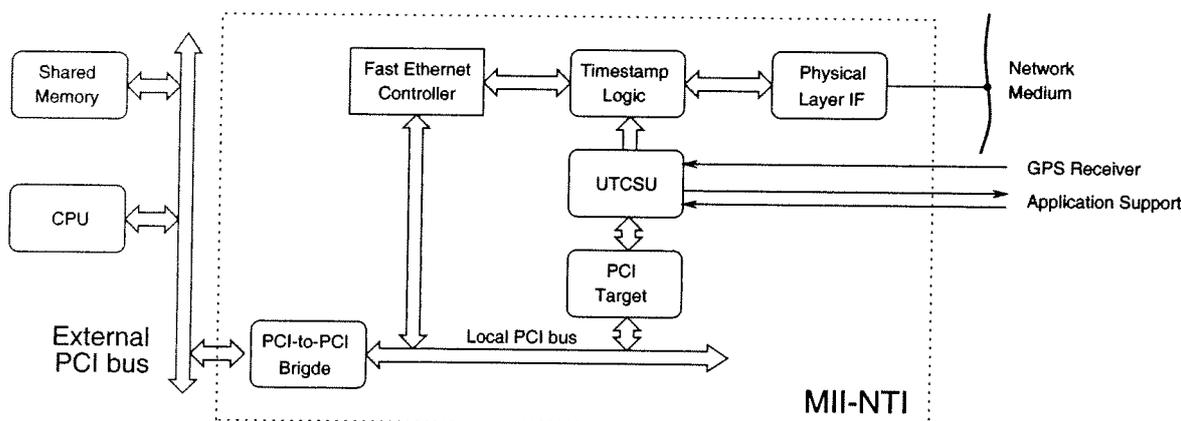


Abbildung 4: Blockschaltbild der MII-NTI PCI-Karte

Dieser Ansatz ist nicht nur für gängige Ethernet und Fast-Ethernet Lösungen geeignet sondern kann in einfacher Weise auch auf Gigabit und 10-Gigabit Netzwerk Technologien erweitert werden. Diese Technologien haben mit dem leistungsfähigen Switching in letzter Zeit im Bereich der Kommunikations-Netze immer stärker an Bedeutung gewonnen. Typische Switches haben leider keine deterministische Packet-Durchlaufzeit, wodurch hochgenaue Uhrensynchronisation unmöglich wird. Um diese Technologie trotzdem einsetzen zu können, entwickeln wir gegenwärtig eine Lösung [KSH00], die hochgenaue Uhrensynchronisation auch in “gesitwchten” Netzwerken ermöglicht.

## 6 Kurzzusammenfassung und Projektdaten

Projektgegenstand von SynUTC war die Entwicklung der theoretischen Grundlagen sowie Hardware+Software zur Bereitstellung einer hochgenauen globalen Uhrzeit in fehlertoleranten verteilten Systemen. Eingesetzt wurde ein intervallbasierendes Verfahren, beruhend auf der Integration von Uhrensynchronisation mit GPS-Satellitenempfängern. Zentrale Ergebnisse sind u.a. elaborater Hardware-Support (UTCSU-ASIC), neuartige Uhrensynchronisationsalgorithmen, leistungsfähige Simulationsumgebung und eine ausführliche mathematisch/formale sowie experimentelle Analyse.

Projektdaten SynUTC	
<i>Homepage:</i>	<a href="http://www.auto.tuwien.ac.at/Projects/SynUTC">http://www.auto.tuwien.ac.at/Projects/SynUTC</a>
<i>Kooperationspartner:</i>	Institut für Automation, TU-Wien Institut für Computertechnik, TU-Wien Institut für Industrielle Elektronik, TU-Wien Spin-off-Firma Oregano Systems
<i>Laufzeit (ohne Spin-off):</i>	Jänner 1995 – Dezember 1999
<i>Förderungsmittel:</i>	ca. 7.5 Mio.ATS
<i># Mitarbeiter:</i>	ca. 20
<i># Personenmonate:</i>	ca. 300–400 (inkl. Studierende)

Die entstandenen Arbeiten wurden primär in erstklassigen internationalen Fachzeitschriften (z.B. Distributed Computing, Chicago Journal of Theoretical Computer Science, J. Real-Time Systems) und Konferenzen (z.B. PTTI, WRTP, MASCOTS, SRDS, PODC) publiziert.

Publikationen SynUTC					
<i>Typ</i>	<i>#</i>	<i># best</i>	<i># invited</i>	$\sum p.$	<i>Avg.p.</i>
Journalartikel	8		2	273	34
Konferenzartikel	20	3	3	197	10
Editorials	1		1	360	360
Diplomarbeiten	5			795	159
Dissertationen	2			419	210
Technical Reports	18			1253	70
<b>Summe:</b>	<b>54</b>	<b>3</b>	<b>6</b>	<b>3.297</b>	

Best papers am IFAC WRTP'94 [Sch94] und WRTP'99 [SN99], best student paper DSN'01 [Hor01]; Einladung als Gasteditor für Sonderausgabe J. Real-Time Systems [Sch97c]; eingeladene Arbeiten und Vorträge in CEP [Sch95], J. Real-Time Systems [SKM<sup>+</sup>00], IASTED AMS'99 [SWGS99], 37th IFIP WG10.4 (<http://www.dependability.org>) Annual Meeting 2000 (Workshop on "Time and Dependability") [Sch00b], WRTP'00 [Sch00a]; Einladung zu Dagstuhl-Seminar "Time Services" [Sch96a, Sch96b].

Wissenschaftliche Highlights: Beantwortung von Leslie Lamport's 13 Jahre alter Frage nach einer fehlertoleranten Intersection-Funktion mit Lipschitz-Eigenschaft [SS01]; Konzept des Adder-based Clocks für die UTCSU.

Präsentation der Prototyp-Implementierung auf der "Hannover Messe'98". Deutsches Gebrauchsmuster für Kabel-Fehlerortung [Sch97b], internationale Patentanmeldung [KSH00] für switched Ethernet-Technologie. Spin-off Firma Oregano Systems betreibt die Weiterentwicklung und Kommerzialisierung der Projektergebnisse.

## Literatur

- [HKS00] Martin Horauer, Nikolaus Kerö, and Ulrich Schmid. A network interface for highly accurate clock synchronization. In W. Maierwieser and K.G. Posch, editors, *Proceedings AUSTROCHIP'00*, pages 93–101, Graz, Austria, October 2000. ISBN 3-9501349-0-5.
- [HL97] Martin Horauer and Dietmar Loy. UTCLIENT – an ASIC supporting clock synchronization in distributed real-time systems. In *Proceedings AUSTROCHIP'97*, pages 290–296, Linz, Austria, April 9, 1997.

- [HL98] Martin Horauer and Dietmar Loy. Hardware-unterstützte Uhrensynchronisation in Verteilten Systemen. In *Proceedings AUSTROCHIP'98*, pages 67–72, Wiener Neustadt, Austria, October 1998. (ISBN 3-901578-03-X, in German).
- [Höc96] Dieter Höchtel. Verfügbarkeitsuntersuchung von GPS-Satellitenempfängern. Diplomarbeit, Vienna University of Technology, Department of Automation, September 1996.
- [Hor01] Martin Horauer. Hardware support for clock synchronization on distributed systems. **best** In *Proceedings of the International Conference on Dependable Systems and Networks (DSN'01)*, pages A-10–A-12, Göteborg, Sweden, July 1–4, 2001.
- [HS97] Dieter Höchtel and Ulrich Schmid. Long-term evaluation of GPS timing receiver failures. In *Proceedings of the 29th IEEE Precise Time and Time Interval Systems and Application Meeting (PTTI'97)*, pages 165–180, Long Beach, California, December 1997.
- [HSS97] Martin Horauer, Ulrich Schmid, and Klaus Schossmaier. NTI: A Network Time Interface M-Module for high-accuracy clock synchronization. Technical Report 183/1-76, Vienna University of Technology, Department of Automation, January 1997.
- [HSS98] Martin Horauer, Ulrich Schmid, and Klaus Schossmaier. NTI: A Network Time Interface M-Module for high-accuracy clock synchronization. In *Proceedings 6th International Workshop on Parallel and Distributed Real-Time Systems (WPDRTS'98)*, pages 1067–1076, Orlando, Florida, March 30 – April 3 1998.
- [KKMS95] Hermann Kopetz, Andreas Krüger, Dietmar Millinger, and Anton Schedl. A synchronization strategy for a time-triggered multicluster real-time system. In *Proceedings Reliable Distributed Systems (RDS'95)*, Bad Neuenahr, Germany, September 1995.
- [KO87] Hermann Kopetz and Wilhelm Ochsenreiter. Clock synchronization in distributed real-time systems. *IEEE Transactions on Computers*, C-36(8):933–939, 1987.
- [KSH00] Nikolaus Kerö, Ulrich Schmid, and Martin Horauer. Verfahren für die Synchronisation von Computeruhren in Netzwerken. **Patent** Technical Report 183/1-105, Department of Automation, TU Vienna, March 2000. Patent Europe, US and Japan.
- [Lis93] Barbara Liskov. Practical uses of synchronized clocks in distributed systems. *Distributed Computing*, 6:211–219, 1993.
- [Loy96] Dietmar Loy. *GPS-Linked High Accuracy NTP Time Processor for Distributed Fault-Tolerant Real-Time Systems*. Dissertation, Vienna University of Technology, Faculty of Electrical Engineering, April 1996. (Appeared in *Dissertationen der Technischen Universität Wien*, Band 78. Österreichischer Kunst- und Kulturverlag, Wien 1997, ISBN 3-85437-056-3).
- [Loy97] Dietmar Loy. Time-services hardware support in fault-tolerant real-time systems. In *Proceedings of the 5th International Conference on VLSI and CAD (ICVC'97)*, Seoul, Korea, October 13–15, 1997.
- [LWL84] Jennifer Lundelius-Welch and Nancy A. Lynch. An upper and lower bound for clock synchronization. *Information and Control*, 62:190–204, 1984.
- [Man99] Thomas Mandl. Network Time Interface Benutzerhandbuch. Diplomarbeit, Vienna University of Technology, Department of Automation, February 1999. (in German).

- [MUM96] MUMM. *ANSI/VITA 12-1996, M-Module Specification*. Manufacturers and Users of M-Modules e.V., 1996.
- [Sch94] Ulrich Schmid. Synchronized UTC for distributed real-time systems. In *Proceedings 19th IFAC/IFIP Workshop on Real-Time Programming (WRTP'94)*, pages 101–107, Lake Reichenau, Germany, 1994. **best**
- [Sch95] Ulrich Schmid. Synchronized Universal Time Coordinated for distributed real-time systems. *Control Engineering Practice*, 3(6):877–884, 1995. (Reprint from Proceedings 19th IFAC/IFIP Workshop on Real-Time Programming (WRTP'94), Lake Reichenau/Germany, 1994, p. 101–107.). **invite**
- [Sch96a] Ulrich Schmid. Interval-based clock synchronization. In *Seminar-Report 138 of Dagstuhl-Seminar 9611 on "Time Services"*, page 7, Schloß Dagstuhl, Germany, March 1996. (Reprinted in *J. Real-Time Systems* 12(3), 1997, p. 334).
- [Sch96b] Klaus Schossmaier. UTC SU — an ASIC to support clock synchronization for distributed real-time systems. In *Seminar-Report 138 of Dagstuhl-Seminar 9611 on "Time Services"*, page 8, Schloß Dagstuhl, Germany, March 1996. (Reprinted in *J. Real-Time Systems* 12(3), 1997, p. 334).
- [Sch97a] Ulrich Schmid. Interval-based clock synchronization with optimal precision. Technical Report 183/1-78, Vienna University of Technology, Department of Automation, July 1997. (submitted to *Information and Computation*).
- [Sch97b] Ulrich Schmid. Kabel-Fehlerortung mit SynUTC. Technical Report 183/1-81, Vienna University of Technology, Department of Automation, July 1997. (Patent: German Gebrauchsmuster 296 23086.3). **Patent**
- [Sch97c] Ulrich Schmid, editor. *Special Issue on The Challenge of Global Time in Large-Scale Distributed Real-Time Systems*, *J. Real-Time Systems* 12(1–3), 1997. **invited**
- [Sch97d] Klaus Schossmaier. An interval-based framework for clock rate synchronization algorithms. In *Proceedings 16th ACM Symposium on Principles of Distributed Computing*, pages 169–178, St. Barbara, USA, August 21–24, 1997.
- [Sch98] Klaus Schossmaier. *Interval-based Clock State and Rate Synchronization*. Dissertation, Vienna University of Technology, Faculty of Technical and Natural Sciences, 1998. (Appeared in *Dissertationen der Technischen Universität Wien*, Band 87. Österreichischer Kunst- und Kulturverlag, Wien, December 1997, ISBN 3-85437-201-9.).
- [Sch00a] Ulrich Schmid. Applied research: A scientist's perspective. In *Proceedings joint 25th IFAC/IFIP Workshop on Real-Time Programming (WRTP'00) and Workshop on Algorithms and Architectures for Real-Time Control (AARTC'00)*, pages 5–12, Palma de Mallorca, Spain, May 2000. (Reprinted in *Annual Reviews in Control* 25 (2001): 183–190.). **invited**
- [Sch00b] Ulrich Schmid. High-accuracy time services and fault-tolerant clock synchronization. In *37th IFIP WG10.4 Annual Meeting (Workshop on "Time and Dependability"*, pages 39–57, Martinique, France, January 21–25, 2000. **invited**
- [Sch00c] Ulrich Schmid. Orthogonal accuracy clock synchronization. *Chicago Journal of Theoretical Computer Science*, 2000(3):3–77, 2000.

- [SHK99] Ulrich Schmid, Martin Horauer, and Nikolaus Kerö. How to distribute GPS-time over COTS-based LANs. In *Proceedings of the 31th IEEE Precise Time and Time Interval Systems and Application Meeting (PTTI'99)*, pages 545–560, Dana Point, California, December 1999.
- [SK99] Klaus Schossmaier and Johann Klasek. Implementing the optimal precision algorithm for clock state & rate synchronization. Technical Report 183/1-91, Vienna University of Technology, Department of Automation, February 1999.
- [SKM<sup>+</sup>00] Ulrich Schmid, Johann Klasek, Thomas Mandl, Herbert Nachtnebel, Gerhard R. Cadek, and Nikolaus Kerö. A Network Time Interface M-Module for distributing GPS-time over LANs. *J. Real-Time Systems*, 18(1):24–57, January 2000. **invite**
- [SN99] Ulrich Schmid and Herbert Nachtnebel. Experimental evaluation of high-accuracy time distribution in a COTS-based Ethernet LAN. In *Proceedings 24th IFAC/IFIP Workshop on Real-Time Programming (WRTP'99)*, pages 59–68, Schloß Dagstuhl, Germany, May/June 1999. **best**
- [SS97a] Ulrich Schmid and Klaus Schossmaier. Interval-based clock synchronization. *J. Real-Time Systems*, 12(2):173–228, March 1997.
- [SS97b] Ulrich Schmid and Klaus Schossmaier. Interval-based clock synchronization revisited. Technical Report 183/1-80, Vienna University of Technology, Department of Automation, July 1997.
- [SS01] Ulrich Schmid and Klaus Schossmaier. How to reconcile fault-tolerant interval intersection with the Lipschitz condition. *Distributed Computing*, 14(2):101 – 111, May 2001.
- [SSHL97] Klaus Schossmaier, Ulrich Schmid, Martin Horauer, and Dietmar Loy. Specification and implementation of the Universal Time Coordinated Synchronization Unit (UTCUSU). *J. Real-Time Systems*, 12(3):295–327, May 1997.
- [SW99] Klaus Schossmaier and Bettina Weiss. An algorithm for fault-tolerant clock state & rate synchronization. In *Proceedings 18th IEEE Symposium on Reliable Distributed Systems (SRDS'99)*, pages 36–47, Lausanne, Switzerland, October 19–22, 1999.
- [SWGS99] Ulrich Schmid, Bettina Weiss, Günther Gridling, and Klaus Schossmaier. A unified approach for simulation and experimental evaluation of fault-tolerant distributed systems. In *Proceedings of the IASTED International Conference on Applied Modelling and Simulation (AMS'99)*, pages 43–48, Cairns, Queensland, Australia, September 1–3, 1999. **invited**
- [WGSS99] Bettina Weiss, Günther Gridling, Ulrich Schmid, and Klaus Schossmaier. The SimUTC fault-tolerant distributed systems simulation toolkit. In *Proceedings 7th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS'99)*, pages 68–75, College Park, MD, USA, October 24–28, 1999.