Kapitel 1

Einleitung

Eine globale Uhrzeit ermöglicht eine optimale Darstellung von lokalen Ereignissen und Sende-Empfangs-Paaren durch Monitorwerkzeuge zur Analyse paralleler und verteilter Anwendungen.

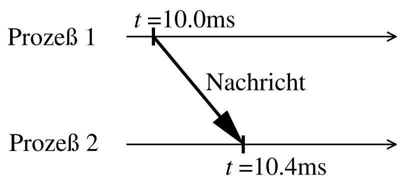


Abb. 1.1: Zeitliniendiagramm basierend auf der physikalischen Zeit

Die Abb. 1.1 zeigt ein Zeitliniendiagramm (timeline diagram) auf der Basis der physikalischen Zeit t (physical time, wall clock time). Die horizontalen Linien sind die Zeitachsen zur Darstellung von Ereignissen in den einzelnen Prozessen eines parallel laufenden Computerprogramms. Die vertikalen Striche symbolisieren Ereignisse (events), hier im Prozeß 1 das Senden einer Nachricht (message) und in Prozeß 2 das Empfangen derselben. Die Nachricht selbst ist mit einem schrägen Vektorpfeil dargestellt. Die horizontale Komponente dieses Vektors repräsentiert die Zeit, die diese Nachricht unterwegs war. Abb. 1.2 zeigt die

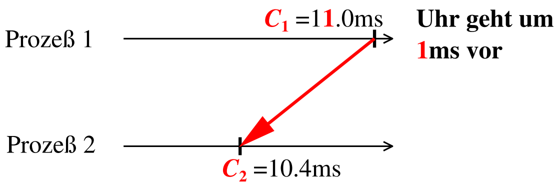


Abb. 1.2: Zeitliniendiagramm mit ungenügend synchronisierten Uhren

gleiche Anwendung, aber basierend auf Zeitstempeln von ungenügend synchronisierten Prozessoruhren C_i. C_1 geht gegenüber der physikalischen Zeit um 1 ms vor, während C_2 korrekt die physikalische Zeit angibt. Das Zeitliniendiagramm zeigt, daß unter solchen Bedingungen bei der Visualisierung des Programmflusses einer parallelen Message-Passing Anwendung die Kausalitäten falsch dargestellt werden, also Nachrichten revers, d.h. mit negativen Nachrichtenlaufzeiten (message delay) gezeichnet werden. Abb. 1.3 enthält die korrekte Darstellung

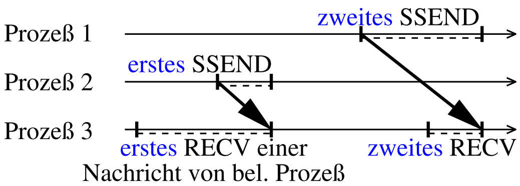


Abb. 1.3: Globales Empfangen mit korrekter Darstellung

eines Programmablaufs, bei dem in den Prozessen 1 und 2 jeweils eine Nachricht mit der Routine MPI_SSEND (synchronous send) des standardisierten Message Passing Interface (MPI) [MPI95] zum dritten Prozeß gesendet wird und diese Nachrichten im dritten Prozeß mit zwei Aufrufen von MPI_RECV(..., MPI_ANY_SOURCE,...) in beliebiger Reihenfolge empfangen werden. Hierbei ist zusätzlich die Dauer der jeweiligen MPI-Routine mit einer horizontalen gestrichelten Linie markiert. Die senkrechten Marken stellen jeweils das Ereignis des Beginns und des Endes der jeweiligen MPI-Routine dar. In Abb. 1.4 wird derselbe Sachverhalt falsch dargestellt, da hier im ersten Prozeß eine nachgehende Uhr zur Zeitmessung benutzt wird. Durch diese nachgehende Uhr werden die Ereignisse im ersten Prozess scheinbar früher dargestellt, und so wird der Anschein erweckt, daß das real zweite Senden nun scheinbar das erste Senden sei. Abb. 1.3 und Abb. 1.4 zeigen also, daß es nicht genügt, nur Rückwärtsbezüge

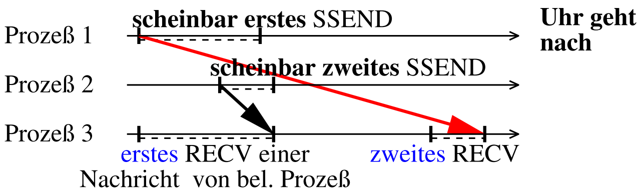


Abb. 1.4: Ereignisse der Abb. 1.3 mit falscher Darstellung

bei der Darstellung von Nachrichten zu verhindern. Durch die nachgehende Uhr in der unteren Darstellung wird der Message-Passing Library unterstellt, daß sie im ersten Aufruf von MPI_RECV die erste Nachricht nicht annimmt, sondern weiter wartet bis das zweite Senden initiiert ist. Beide Beispiele zeigen, daß für das Monitoring paralleler Anwendungen eine für alle Prozesse einheitliche, also globale Zeit, vergleichbar mit der physikalischen Zeit, nötig ist.

Auf Systemen, die keine genügend synchronisierte Prozessoruhren besitzen, d.h. über keine für das Monitoring geeignete globale Uhr verfügen, gibt es verschiedene Strategien, eine solche Uhr zu realisieren. Dieser Bericht stellt hierzu ein neues Verfahren, die geregelte logische Uhr (controlled logical clock), vor. Sie wurde speziell für Workstationcluster oder vergleichbare Systeme entwickelt und kann, wie in Abb. 1.5 auf Seite 11 dargestellt, als Filter für Tracefiles eingesetzt werden. Die geregelte logische Uhr kann als Filterprogramm implementiert werden, das Tracefiles mit logisch fehlerhaften Zeitstempeln einliest, die Zeitstempel korrigiert und die Ereignisinformationen dann mit logisch korrekten Zeitstempeln wieder ausgibt. Hierbei werden die Zeitstempel der eingesetzten Prozessoruhren nachträglich geändert, wenn der Tracefile Rückwärtsbezüge enthält. Rückwärtsbezüge sind hierbei Nachrichten, bei denen das Sendeereignis (send event) einen späteren Zeitstempel als das Empfangsereig-

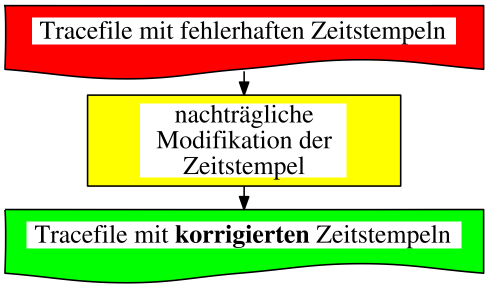


Abb. 1.5: Nachträgliche Korrektur, implementiert als Filter

nis (receive event) besitzt. Die geregelte logische Uhr modifiziert die Zeitstempel, so daß sie Lamports Uhrenbedingung (clock condition) [Lam78] erfüllen, also ein Empfangsereignis dann grundsätzlich einen späteren Zeitstempel als das zugehörige Sendeereignis besitzt.

In Kapitel 2 werden zuerst grundlegende Definitionen bereitgestellt, wie z.B. die Begriffe „synchronisierte Uhren" und „Uhrenbedingung". In Kapitel 3 werden verschiedene Anwendungsklassen vorgestellt, und deren Anforderungsprofile abgeleitet. Anschließend werden existierende Verfahren vorgestellt und ihre Grenzen bezüglich des Anforderungsprofils aufgezeigt. In Kapitel 4 wird der Algorithmus der geregelten logischen Uhr definiert. Kapitel 5 enthält eine detaillierte Fehleranalyse zur geregelten logischen Uhr. Kapitel 6 beschreibt Implementierungsdetails des Algorithmus sowie mit dieser Implementierung durchgeführte Tests. In Kapitel 7 wird die geregelte logische Uhr mit seither existierenden Verfahren verglichen und bezüglich des Anforderungsprofils bewertet. Vor diesem Vergleich werde ich das Verfahren von Hofmann und Hilgers [HH98, Hil96] zuerst detailliert vorstellen und anschließend erweitern, da das Originalverfahren von Hofmann und Hilgers für nicht konstante Uhrengeschwindigkeiten ungeeignet ist, und dies aber eine wesentliche Anforderung ist. Kapitel 8 faßt dann die wesentlichen Aspekte dieser Arbeit zusammen. Im Anhang A findet man eine Liste der verwendeten Symbole.

Es ist noch zu bemerken, daß die meisten Graustufenbilder der gedruckten Version dieser Arbeit auch als Farbabbildungen in der Online-Version verfügbar sind.

Kapitel 2

Definitionen

Problem: Für das Monitoring benötigt man Uhren, die einerseits für Leistungsmessungen möglichst genau gehen sollten, und die andererseits für die Darstellung des Programmablaufs die nachfolgend definierte Uhrenbedingung [Lam78] erfüllen sollten.

Definition 1: Es sei n die Anzahl der Prozesse,
e_i^j das j-te Ereignis (event) im Prozeß i,
$E_i = \{e_i^j | j = 0..j_{\max}(i)\}$ die Menge der Ereignisse im Prozeß i,
$E = \bigcup_{i=1..n} E_i$ die Menge aller Ereignisse und
$M = \{(e_k^l, e_i^j) | e_k^l$ ist ein Sendeereignis und e_i^j ein zugehöriges Empfangsereignis $\}$ ist die Menge der Sende-Empfangs-Paare.
e_i^j ist ein internes Ereignis, wenn es kein Sende- oder Empfangsereignis ist.

Definition 2: t bezeichne die physikalische Zeit (real time).
$t(e_i^j)$ mit $t : E \rightarrow \mathbb{R}$ ist die physikalische Zeit des Ereignisses e_i^j.
Eine *Uhr* ist eine Abbildung der physikalischen Zeit auf eine reelle Zahl:
$\mathcal{C} : \{t | t \in \mathbb{R}\} \rightarrow \mathbb{R}$
$C_i(t)$ mit $C_i : \{t | t \in \mathbb{R}\} \rightarrow \mathbb{R}$ sei die *Prozessoruhr* des Prozesses i.
$C(e_i^j) := C_i(t(e_i^j))$, $C : E \rightarrow \mathbb{R}$ ist die *globale Prozessoruhr.*
Es ist hierbei unerheblich, ob die Prozessoruhren als Hardware mit Quarzuhren realisiert sind oder, ob es sich um eine software-technische Abbildung dieser Quarzuhren handelt, die z.B. schon eine grobe Synchronisation der Quarzuhren enthält.
Zur Synchronisation und Präzision von Uhren gibt es in der Literatur mehrere Begriffsbildungen:
Eine Uhr $\mathcal{C} : \{t | t \in \mathbb{R}\} \rightarrow \mathbb{R}$ ist *korrekt* wenn $\underset{t}{\forall} \mathcal{C}(t) = t$ gilt.
Eine Uhr $\mathcal{C} : \{t | t \in \mathbb{R}\} \rightarrow \mathbb{R}$ *geht genau* wenn $\underset{t}{\forall} \frac{d\mathcal{C}(t)}{dt} = 1$ gilt.
$\frac{d\mathcal{C}(t)}{dt}$ ist die *momentane Uhrengeschwindigkeit.*
$\rho(t) = \frac{d\mathcal{C}(t)}{dt} - 1$ ist die *momentane Gangabweichung* (drift) einer Uhr $\mathcal{C} : \{t | t \in \mathbb{R}\} \rightarrow \mathbb{R}$.
$\rho_{\max} = \max_t |\rho(t)|$ ist die *Ganggenauigkeit*[1] (accuracy) der Uhr $\mathcal{C}$.

[1] Im allgemeinen Sprachgebrauch sind die Begriffe Gangabweichung und Ganggenauigkeit Synonyme. Ich verwende sie hier unterschiedlich, um die momentanen Werte und den maximalen Absolutwert sprachlich besser unterscheiden zu können.

zum Memory-Bedarf, werden in [CB91, MSV91, SK90, SES89, Che92, BDM^{+}94, RS95] aufgezeigt. In [HW88] wird der Begriff der globalen Ereignisse eingeführt. In [PYS92] werden zusätzlich spontane Ereignisse, wie z.B. Kollisionen, berücksichtigt. In der Zusammenfassung [SM94] werden aber auch die Grenzen der diskreten logischen Uhr und der Vektoruhr beleuchtet.

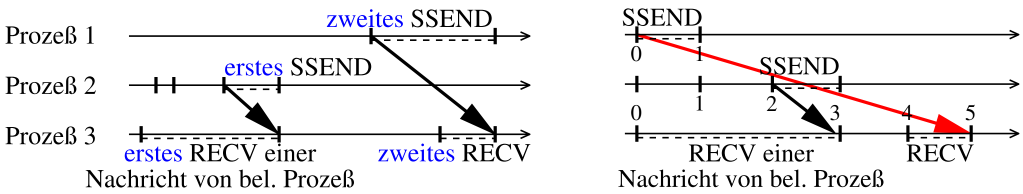


Abb. 3.1: Globales Empfangen dargestellt mittels physikalischer Zeit und diskreter logischer Uhr

Die rechte Darstellung in Abb. 3.1 zeigt eine weitere Beschränkung der Anwendbarkeit von Lamports logischer Uhr auf (s.a. Abb. 1.3 und 1.4 auf S. 10). Hierbei wurden im mittleren Prozeß zu Beginn zwei zusätzliche lokale Ereignisse angenommen. Wie Abb. 1.4 erweckt auch die Darstellung mittels Lamports logischer Uhr den Eindruck, daß das erste globale Empfangen einer Nachricht (d.h. ein Nachrichtenempfang, bei dem nicht festgelegt ist, von welchem sendenden Prozeß die zu empfangende Nachricht kommen soll) die falsche Nachricht empfängt. Dieses Beispiel zeigt damit, daß die logische Uhr für die Darstellung von Monitorinformationen nicht ausreichend ist.

Eine Alternative hierzu ist, durch hinreichend genaue Synchronisierung der einzelnen Uhren eine Darstellung ohne Rückwärtsbezüge zu erreichen [Lam78, Hof93b, MT95]. Eine exakte Synchronisation am Anfang inklusive Bestimmung der Gangabweichungen [Dun91, Dun94], oder besser am Anfang und Ende [MT95] mit einem linearen Ausgleich während der Dauer der Anwendung, sind die in softwarebasierten Monitorwerkzeugen [RTV93] häufig angewandten Methoden. Für die genaue Synchronisation am Anfang und Ende sind deterministische [Sch87] und probabilistische [CF94, CF95] Verfahren geeignet. Eine kontinuierliche Uhrensynchronisation mit geringer Ressourcenbelastung, wie z. B. mit xntp [Mil92], ist normalerweise aufgrund der stark variierenden Nachrichtenlaufzeiten in einem lokalen Netz nicht ausreichend exakt, d.h. der Sychronisationsfehler kann größer sein als die minimale Nachrichtenlaufzeit.

Eine Alternative ist die tracebasierte Synchronisierung. Auf der Basis eines Tracefiles und der Prämisse, daß ein Empfangsereignis frühestens um die minimale Nachrichtenlaufzeit nach dem Sendeereignis ankommen darf, werden die Uhrenabweichungen ermittelt und nach Beendigung des zu monitorenden Programms wird die Traceinformation analysiert, und werden die Zeitstempel modifiziert, damit die modifizierten Zeitstempel die im letzten Kapitel definierten Anforderungen möglichst erfüllen.

Duda et al. [DHHB87] entwickelten hierzu zwei Verfahren. Das erste Verfahren enthält eine Regressionsanalyse, die aber nicht ausreicht, um eine korrekte Darstellung von Kommunikationsbeziehungen zu erreichen. Beim zweiten Verfahren werden zwei konvexe Hüllen gebildet. Es ist in Abb. 3.2 auf Seite 18 dargestellt. Das Verfahren bestimmt den Synchronisationsfehler zwischen zwei Prozessoruhren C_i und C_k als eine lineare Funktion der Prozessorzeit. Zur Vereinheitlichung der Darstellung wurde in Abb. 3.2 für die Berechnung dieser linearen Ausgleichsfunktion die physikalische Zeit t als Abszisse gewählt. Dies ist aber unerheblich, da der Maßstab der Abszisse den Unterschied zwischen den Prozessor-Zeiten der

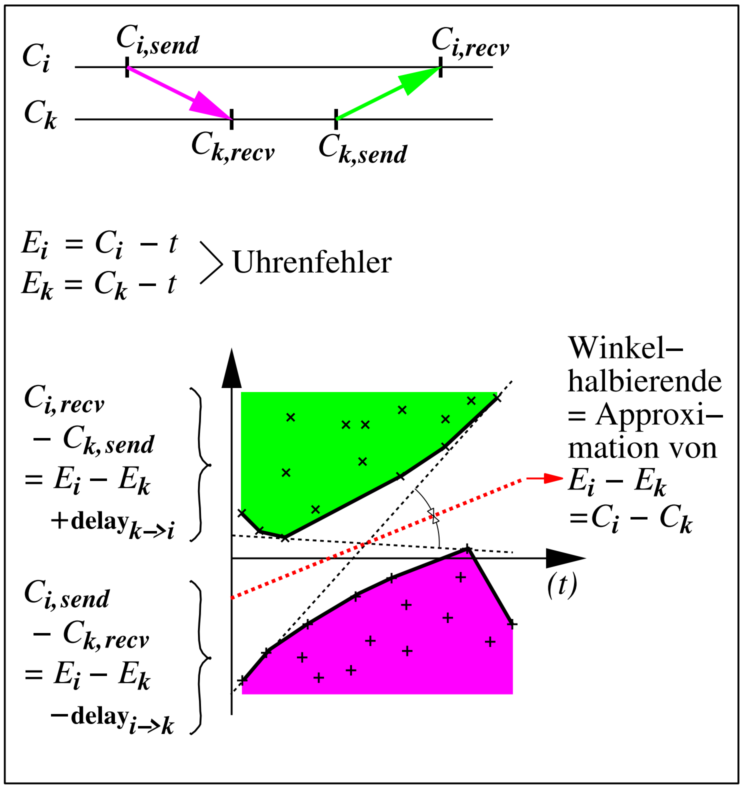


Abb. 3.2: Algorithmus von Duda

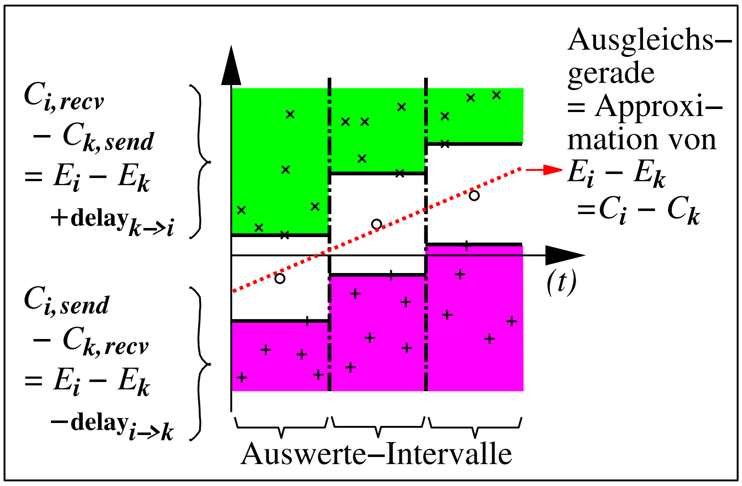


Abb. 3.3: Algorithmus von Hofmann

Kapitel 4

Die geregelte logische Uhr

Da *Lamports logische Uhr* die Basis dieser Arbeit ist, wird sie im folgenden Unterkapitel kurz vorgestellt. Anschließend wird eine *einfache logische Uhr* definiert, die zwar schon einen Bezug zur Prozessorzeit (und damit zur physikalischen Zeit) besitzt, die aber die gestellten Anforderungen noch nicht erfüllt. Aufbauend auf dieser Uhr wird dann in den Unterkapiteln 4.3 und 4.4 die *geregelte logische Uhr* und ihr Regler definiert. In Kap. 4.5 wird der Algorithmus noch durch die *rückwärtige Amortisation* ergänzt.

4.1 Lamports logische Uhr

C_i sei die Prozessoruhr des Prozesses i.

Algorithmus 1: Lamports logische Uhr $\boldsymbol{LC_i^{int}}$ ist definiert mit

$$LC_i^{int}(e_i^j) := \begin{cases} 0 \quad \text{für } j = 0 \\ LC_i^{int}(e_i^{j-1}) + 1 \quad \text{für interne und Sendeereignisse} \\ \max(LC_i^{int}(e_i^{j-1}) + 1, LC_k^{int}(e_k^l) + 1) \quad \text{wenn} \underset{e_k^l}{\exists} (e_k^l, e_i^j) \in M, \\ \qquad \text{d.h. wenn } e_i^j \text{ ein Empfangsereignis ist und } e_k^l \text{ das} \\ \qquad \text{zugehörige Sendeereignis ist.} \end{cases} \tag{4.1}$$

Die globale logische Uhr ist definiert als

$$LC^{int}(e_i^j) := LC_i^{int}(e_i^j).$$

Abb. 4.1 zeigt ein Beispiel. In dem Diagramm rechts oben sind die Ereignisse in den beiden Prozessen 1 und 2 als kleine Kreise dargestellt. Auf der Abszisse ist ihre physikalische Zeit und auf der Ordinate ist jeweils die ihnen zugehörige Prozessorzeit C_1 oder C_2 zugeordnet. Gestrichelt dargestellt sind jeweils die Abbildungen $t \mapsto C_1(t)$ und $t \mapsto C_2(t)$, d.h. die Ereignisse des Prozesses 1 liegen auf der C_1-Linie und die des Prozesses 2 auf der C_2-Linie. Die schrägen Pfeile symbolisieren Nachrichten, z.B. die erste Nachricht geht von dem mit „1“ markierten Sendeereignis in Prozeß 1 zu dem mit „2“ markierten Empfangsereignis in Prozeß 2. Die Zahlen, mit denen die Ereignisse markiert sind, sind die Werte LC_1 und LC_2 von Lamports logischer Uhr gemäß Gleichung (4.1) in Algorithmus 1.

Bei den Ereignissen mit den Werten $LC_2^{int} = 2$ und $LC_1^{int} = 7$ (eingekreiste Werte) war der rechte Term in (4.1) maßgeblich, d.h., daß die logische Uhr zusätzlich vorgestellt werden mußte, damit keine Rückwärtsbezüge auftreten.

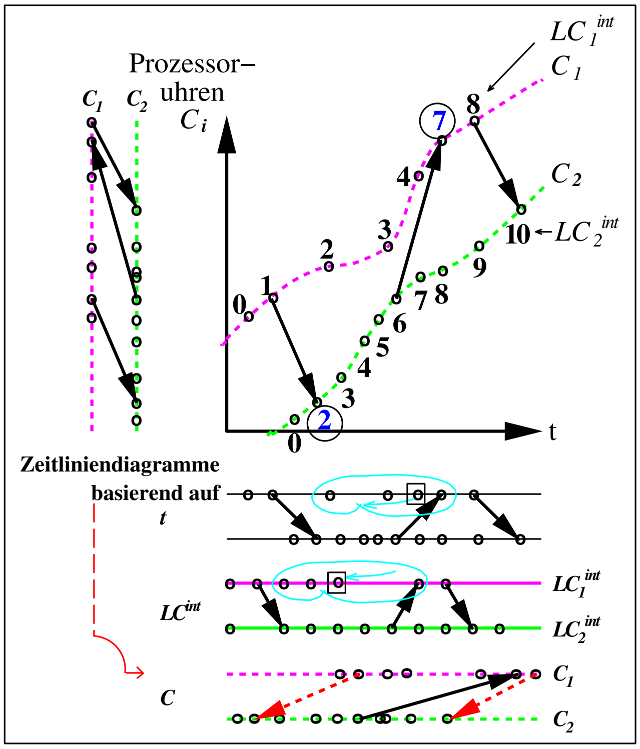


Abb. 4.1: Lamports logische Uhr

Links neben dem beschriebenen t-C_i-Diagramm befindet sich das zugehörige Zeitliniendiagramm auf der Basis der Prozessoruhren. Dies heißt, daß die vertikalen Linien die Zeitachsen C_1 (links) und C_2 (rechts) darstellen. Die Richtung der fortschreitenden Zeit ist nach oben. Es handelt sich hierbei um eine horizontale Parallelprojektion der Ereignisse auf die vertikalen Zeitachsen C_i. Auf den Zeitachsen sind die Ereignisse jeweils zu ihrem Zeitpunkt C_i wieder als kleine Kreise eingezeichnet. Die Pfeile stellen wieder die Nachrichten dar. Man sieht, daß die erste und dritte Nachricht jeweils mit einem rückläufigen Pfeil abgebildet ist. Diese Darstellung ist logisch falsch, da sie den Eindruck erweckt, Nachrichten zu empfangen, bevor sie überhaupt abgeschickt worden sind.

Das unterste horizontale Diagramm stellt nochmals dieses Zeitliniendiagramm auf der Basis der Prozessoruhren dar. Zur Verdeutlichung sind die logisch falsch dargestellten Nachrichten hier gestrichelt dargestellt.

Das oberste horizontale Zeitliniendiagramm stellt den korrekten Verlauf der Ereignisse auf der Basis der physikalischen Zeit t dar. Hier handelt es sich um eine vertikale Parallelprojektion der Ereignisse auf die horizontalen physikalischen Zeitachsen (t) in den beiden Prozessen.

Das mittlere horizontale Zeitliniendiagramm basiert auf Lamports logischer Uhr. Die Ereignisse sind gemäß ihren Werten LC_i^{int} auf den Zeitlinienachsen eingetragen. Lamports logische Uhr erfüllt die Uhrenbedingung, d.h. Rückwärtsbezüge (die gestrichelten Pfeile im untersten, auf den Prozessoruhren basierten Zeitliniendiagramm) werden verhindert, wie das mittlere Zeitliniendiagramm zeigt. Lamports logische Uhr eignet sich aber nicht für Leistungsmessungen, da sie keinen Bezug zur physikalischen Zeit besitzt.

Die folgenden Betrachtungen dienen zur Veranschaulichung der Anforderungen an Uhren, die für Monitorwerkzeuge eingesetzt werden können. Da Lamports logische Uhr nicht für diese Anforderungen entworfen ist, kann diese Uhr als Beispiel dafür dienen, daß eine logische Uhr zwar die Uhrenbedingung erfüllt, aber den nachfolgend genannten, für ein Monitorwerkzeug wichtigen, Anforderungen nicht genügt. Eine der Anforderungen ist, daß Leistungsmessungen möglich sind, also ein Bezug zur physikalischen Zeit besteht. Bei Lamports logischer Uhr fehlt dieser Bezug. Das sieht man deutlich an den vier mit einer punktierten Linie eingekreisten Ereignissen: Das mit einem Rechteck markierte Ereignis hat in Wirklichkeit (d.h. bezüglich der physikalischen Zeit t, dargestellt im obersten Zeitliniendiagramm) einen etwa gleichgroßen zeitlichen Abstand zu seinem vorhergehenden und zu seinem nachfolgenden Ereignis. In dem auf Lamports logischer Uhr basierenden mittleren Diagramm hingegen ist dieses Ereignis dreimal soweit von seinem nachfolgenden Ereignis im Vergleich zu seinem vorhergehenden Ereignis entfernt.

Außerdem können auch globale Wartepunkte (z.B. `MPI_RECV(...MPI_ANY_SOURCE...)`, [MPI95]), wie schon in Abb. 3.1 auf Seite 17 aufgezeigt, mit Lamports logischer Uhr nicht adäquat dargestellt werden. Der Schwerpunkt von Lamports logischer Uhr besteht darin, daß die Uhrenbedingung erfüllt ist.

4.2 Die einfache logische Uhr

Die einfache logische Uhr ist eine Erweiterung von Lamports logischer Uhr. Der Name *einfach* wurde zur Unterscheidung von der darauf aufbauenden *geregelten* logischen Uhr gewählt.

Definition 8: t sei die physikalische Uhrzeit, $T(t)$ sei eine globale Zeit, zu der die Prozessoruhren $C_i(t)(i = 1..n)$ mit einer begrenzten Abweichung synchronisiert sind, d.h. es existieren Konstanten $-e_i^-$ und e_i^+ mit $-e_i^- \leq C_i(t) - T(t) \leq e_i^+$.

Die einfache logische Uhr LC wird folgendermaßen gebildet: Normalerweise läuft sie wie C_i, d.h. $LC_i := C_i$. Beim Empfangen eines Ereignisses wird sie auf $\max(LC_i, \mu + LC_k$ des Senders beim Absenden) vorgestellt, mit $\mu =$ minimale Nachrichtenlaufzeit, die z.B. bei einer internen Synchronisation vor Beginn der Anwendung ermittelt werden kann. Wenn die logische Uhr auf der Basis dieser Regel vorgestellt worden ist, dann bleibt sie anschließend nahezu stehen, solange sie gegenüber der lokalen Prozessoruhr noch vorgeht, d.h. solange $LC_i > C_i$. Genaugenommen wird sie hierbei jeweils bei jedem Ereignis um ein kleines δ vorgestellt. Es handelt sich hierbei um eine Modifikation der von Lamport [Lam78] entwickelten logischen Uhr.

Algorithmus 2: Die exakte Definition der **einfachen logischen Uhr *LC*** ist:

$$LC_i(e_i^j) := \begin{cases} \max(LC_k(e_k^l) + \mu_{k,i},\, LC_i(e_i^{j-1}) + \delta_i,\, C_i(t(e_i^j))) & (4.2) \\ \quad \text{wenn } \underset{e_k^l}{\exists}\, (e_k^l, e_i^j) \in M & \\ \max(LC_i(e_i^{j-1}) + \delta_i,\, C_i(t(e_i^j))) \quad \text{andernfalls,} & (4.3) \\ \text{wobei die Terme } LC_i(e_i^{j-1}) + \delta_i \text{ bei } j = 0 \text{ entfallen} & \end{cases}$$

mit

$\delta_i =$ minimaler Abstand zweier Ereignisse in Prozeß i, d.h. δ_i sind Konstanten für die gelten:

$$\delta_i > 0 \quad \wedge \quad \underset{i,j}{\forall}\, T(t(e_i^j)) - T(t(e_i^{j-1})) \geq \delta_i \tag{4.4}$$

$\mu_{k,i}$ = minimale Laufzeit einer Nachricht von Prozeß k nach Prozeß i, d.h. $\mu_{k,i}$ sind Konstanten für die gelten:

$$\mu_{k,i} > 0 \quad \wedge \quad \underset{(e_k^l, e_i^j) \in M}{\forall} T(t(e_i^j)) - T(t(e_k^l)) \geq \mu_{k,i} \tag{4.5}$$

Die zugehörige globale logische Uhr ist definiert als

$$LC(e_i^j) := LC_i(e_i^j) \tag{4.6}$$

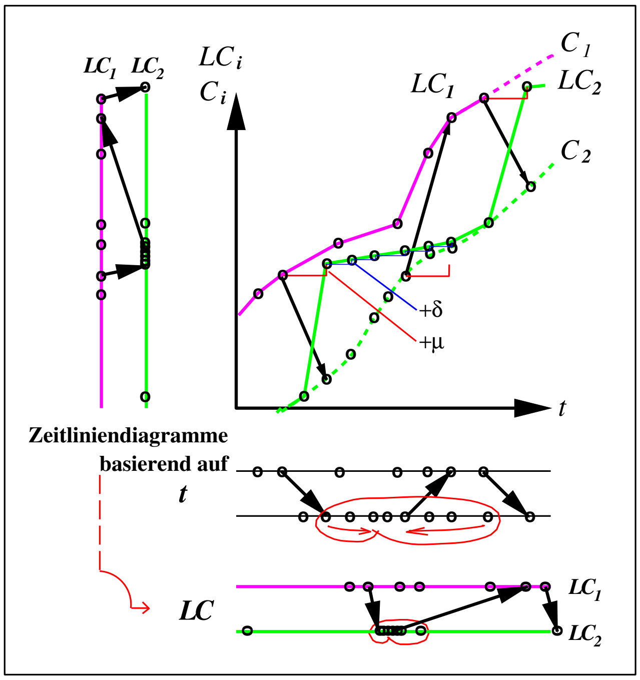


Abb. 4.2: Einfache logische Uhr

In Abb. 4.2 wird der Algorithmus anhand des vorhergehenden Beispiels veranschaulicht. In dem Diagramm rechts oben sind wieder die Ereignisse mit ihrer Prozessorzeit C_i über der physikalischen Zeit t aufgetragen. Zusätzlich wurden die Ereignisse auch mit ihren Werten LC_1 und LC_2 dargestellt und mit durchgezogenen Linien verbunden. Da im Falle des Prozesses 1 die Werte von LC_1 und C_1 identisch sind, wurde auf die gestrichelte Darstellung von C_1 im Bereich der durchgezognenen stückweise linearen Darstellung[1] von LC_1 verzichtet. Im folgenden sei der Verlauf der einfachen logischen Uhr LC_2 erläutert: LC_2 beginnt beim ersten Ereignis in Prozeß 2 mit dem Wert der Prozessoruhr C_2. Das zweite Ereignis ist ein Empfangsereignis. Hier wird der Term $LC_k(e_k^l) + \mu_{k,i}$ in (4.2) relevant, da dieser Wert den der eigenen Prozessoruhr $C_i(t(e_i^j))$ übersteigt. Dies wird im Diagramm durch die mit $+\mu$ markierten Stufe angezeigt. Da nun der Wert der von LC_2 den Wert von C_2 übersteigt, wird bei den folgenden internen Ereignissen der Term $LC_i(e_i^{j-1}) + \delta_i$ in (4.2) relevant. Dies wird

[1] Die logischen Uhren sind nur für Ereignisse im Tracefile definiert. Zur besseren Veranschaulichung wurden die Punktmengen $\{(LC_i(e_i^j), t(e_i^j))\}_{j=1,\ldots}$ für jeden Prozeß P_i mit geraden Linien zu stückweise linearen Kurven verbunden.

$\square$

Bemerkung 5 *Die beiden Sätze 5 und 6 sind nicht* symmetrisch *zueinander: Das Vorgehen der einfachen logischen Uhr LC ist nur global durch das Maximum des Vorgehens aller beteiligten Uhren begrenzt, während das Nachgehen von LC innerhalb eines jeden Prozesses durch das maximale Nachgehen der jeweiligen Prozessoruhr begrenzt ist.*

(Beweis durch entsprechende Wahl von $T := C_i$ bzw. $\max_k C_k$ und $e_i^- := 0$ bzw. $e^+ := 0$.)

4.3 Die geregelte logische Uhr

Im Algorithmus 2 bewirkt der Term $LC_k(e_k^l) + \mu_{k,i}$ in (4.2) ggf. ein Vorstellen der logischen Uhr $LC(e_i^j)$ gegenüber der Uhr $C_i(t(e_i^j))$. Bei nachfolgenden Ereignissen im gleichen Prozeß bewirkt der Term $LC_i(e_i^{j-1}) + \delta_i$ in (4.2) und (4.3), daß LC im Prinzip **stehenbleibt** (bzw. – genau gesagt – um eine kleine Zeitspanne δ_i vorgestellt wird), bis sie dann wieder auf den Wert $C_i(t(e_i^j))$ *zurückgefallen* ist (siehe Term $C_i(t(e_i^j))$ in (4.2) und (4.3)). Damit die logische Uhr gleichmäßiger gehen kann, bzw. um das abwechselnde Vorstellen und Stehenbleiben derselben zu verhindern, wird die einfache logische Uhr aus Algorithmus 2 folgendermaßen erweitert:

Algorithmus 3: LC' bildet die Basis für die geregelte logische Uhr:

$$LC'_i(e_i^j) := \begin{cases} \max(LC'_k(e_k^l) + \mu_{k,i},\ LC'_i(e_i^{j-1}) + \delta_i, \\ \qquad LC'_i(e_i^{j-1}) + \gamma_i^j(C_i(t(e_i^j)) - C_i(t(e_i^{j-1}))), \\ \qquad C_i(t(e_i^j))) \quad \text{wenn } \underset{e_k^l}{\exists}\ (e_k^l, e_i^j) \in M & (4.10) \\ \max(LC'_i(e_i^{j-1}) + \delta_i,\ LC'_i(e_i^{j-1}) + \gamma_i^j(C_i(t(e_i^j)) - C_i(t(e_i^{j-1}))), \\ \qquad C_i(t(e_i^j))) \quad \text{andernfalls} & (4.11) \\ \text{wobei die Terme } LC_i(e_i^{j-1}) + \delta_i \text{ bei } j = 0 \text{ entfallen} \end{cases}$$

mit δ_i und $\mu_{k,i}$ wie in Algorithmus 2 und mit frei wählbaren $\gamma_i^j \in [0, 1]$.

Die zugehörige globale logische Uhr wird wie in Algorithmus 2, Gleichung (4.6), ebenfalls aus den logischen Uhren der einzelnen Prozesse gebildet:

$$LC'(e_i^j) := LC'_i(e_i^j) \tag{4.12}$$

Der Algorithmus wird mit dem Regler in Kap. 4.4 später vervollständigt.

Offensichtlich gilt $LC_{\text{Alg.2}} = LC'_{\text{Alg.3}}$ für $\gamma_i^j \equiv 0$. Für $\gamma_i^j \equiv 1$ entspricht LC' der Definition von Lamport [Lam78] für eine stückweise differenzierbare logische Uhr LC mit $d\,LC_i/dt = d\,C_i/dt$ für alle Zeitpunkte zwischen Empfangsereignissen.

Abb. 4.3 auf Seite 30 zeigt die Wirkungsweise dieses Algorithmus für $\gamma \equiv 1$. Durch die erste Nachricht wird die C_2-Kurve parallel nach oben verschoben (d.h. vorgestellt); das ergibt LC'_2 und ist in der Abbildung durch die vertikalen gestrichelten Parallelverschiebungspfeile markiert. Da C_1 die am weitesten vorgehende Uhr ist, wird sie nicht noch weiter vorgestellt, d.h. $LC'_1 := C_1$. Damit verlaufen nun LC'_1 und LC'_2 nahezu gleich, abgesehen von kleinen Schwankungen der Uhrengeschwindigkeiten. Diese Schwankungen können aber dazu führen, daß LC'_2 weiter als die schnellste Prozessoruhr (hier C_1) vorgeht, siehe die mit eingekreisten Ereignisse in Abb. 4.3. Dies zu verhindern, ist die Aufgabe des in den nächsten Kapiteln

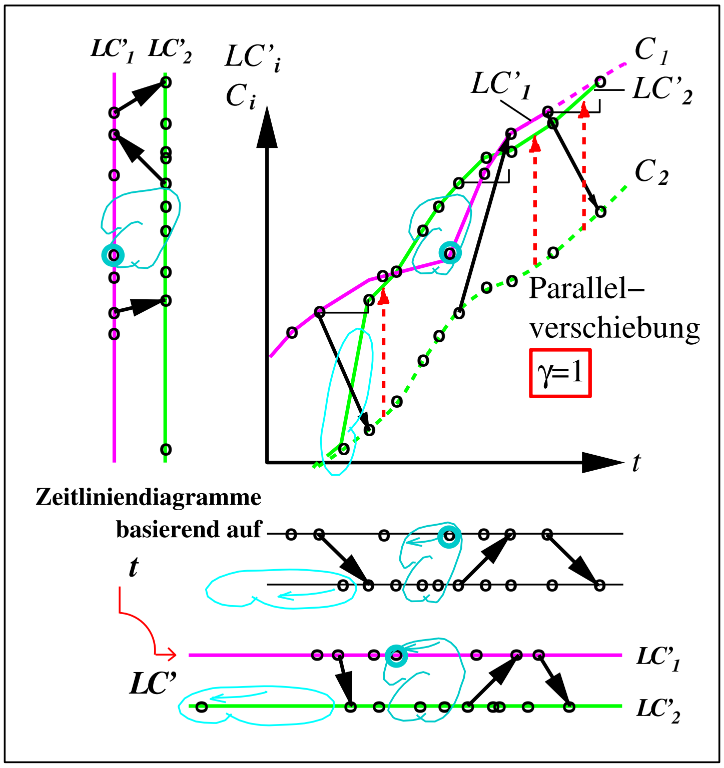


Abb. 4.3: Die geregelte logische Uhr

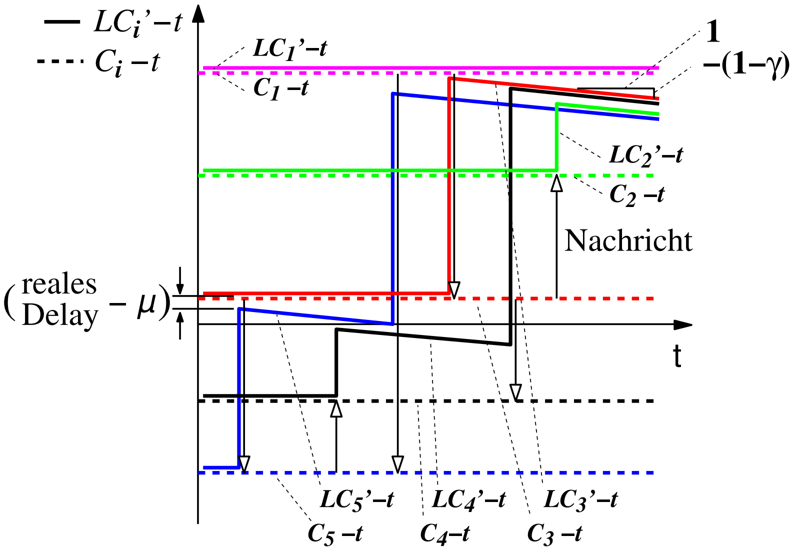


Abb. 4.4: Die geregelte logische Uhr – eine andere Darstellung

beschriebenen Reglers. Außerdem hat man zu Beginn starke Fehler in der Darstellung, solange die Uhrenangleichung mangels entsprechender Nachrichten noch nicht durchgeführt ist. Dies wird deutlich, wenn man das erste Ereignis in Prozeß 2 betrachtet (es ist jeweils mit einer Wolke eingekreist). Während dieses Ereignis in Wirklichkeit nahe an dem zweiten Ereignis im selben Prozeß liegt (siehe das obere, auf der physikalischen Zeit t basierende Zeitliniendiagramm), wird es in dem auf LC' basierenden, unteren Diagramm lange Zeit vor dem zweiten Ereignis dargestellt. Dies kann am einfachsten dadurch verhindert werden, daß vor dem Start der zu beobachtenden Anwendung sämtliche Prozessoruhren synchronisiert werden. Das untere Zeitliniendiagramm basiert auf der geregelten logischen Uhr und zeigt gegenüber der darüberliegenden Darstellung, die auf der physikalischen globalen Zeit basiert, abgesehen von den soeben beschriebenen Problemen, nur noch geringe Abweichungen in der Darstellung.

Abb. 4.4 auf Seite 30 stellt die Funktionsweise der geregelten logischen Uhr in einer anderen Form dar. Auf der Ordinate sind nicht die Uhren (C, LC), sondern ihr Vorgehen ($C - t$, $LC - t$) aufgetragen. Als Beispiel dienen hier Prozessoruhren, die konstant vor- bzw. nachgehen. Man kann die Wirkungsweise der geregelten logischen Uhr und den Einfluß realer Nachrichtenlaufzeiten und des Faktors γ hier sehen:

- LC'_i wird bei einem Empfangsereignis (bis auf die reale Nachrichtenlaufzeit [delay] abzüglich μ) auf das Niveau der LC'_k des Senders angehoben, wenn dessen Uhr entsprechend stärker vorgeht,
- die angehobenen Kurven $LC'_i - t$ fallen um die Steigung $-(1 - \gamma)$ gegenüber einer gedachten Parallele von $C_i - t$.

Außerdem sieht man, wenn es Nachrichtenketten[4] von dem Prozeß mit der am weitesten vorgehenden Prozessoruhr (hier C_1) zu jedem anderen Prozeß gibt, und deren Nachrichtenlaufzeiten in der Nähe von μ liegen, und wenn $\gamma \cong 1$, daß dann für die geregelten logischen Uhren aller Prozesse

$$LC'_i \cong \max_{k=1..n} (C_k)$$

gilt und somit auch global:

$$LC' \cong \max_{k=1..n} (C_k)$$

Hieraus ergibt sich, daß der Regler den Wert von γ nahe 1 belassen sollte. Am Beispiel in Abb. 4.4 sieht man dies auf der rechten Seite: Nachdem es Nachrichtenketten vom Prozeß 1 zu jedem der anderen vier Prozesse gab, liegen die geregelten logischen Uhren LC'_i nun in der Nähe der Prozessoruhr C_1, welche die am weitesten vorgehende ist. Wenn nun γ nahe bei 1 liegt, dann bleiben die geregelten logischen Uhren über einen langen Zeitraum in der Nähe von C_1. Je näher γ bei 1 liegt, desto weiter können die Zeitpunkte auseinanderliegen, an denen Nachrichtenketten wieder zu einer Synchronisierung führen.

[4] Eine Nachrichtenkette von einem Prozeß P_i zu einem Prozeß P_k sei definiert als eine Folge von Nachrichten für die gilt: (a) Die erste Nachricht wird in Prozeß P_i abgesandt, und (b) jede weitere Nachricht wird erst abgesandt, nachdem die vorhergehende im selben (Zwischen-)Prozeß angekommen war, und (c) die letzte Nachricht wird in Prozeß P_k empfangen.

4.4 Der Regler

4.4.1 Reglerbedarf

Der Regler soll verhindern, daß die geregelte logische Uhr weiter vorgeht, als die am weitesten vorgehende Prozessoruhr. Dies läßt sich dadurch erreichen, daß γ_i^j reduziert wird, sobald bei einem Prozessor die geregelte logische Uhr LC_i' weiter vorgeht, als alle Prozessoruhren, d.h. sobald folgende Bedingung erfüllt ist:

$$\exists_i \, LC_i' > \max_{k=1..n} (C_k)$$

Dies kann vorkommen, wenn

$$\rho_{\text{am weitesten vorgehende Uhr}} < \rho_{\text{beliebige Uhr}} - (1-\gamma) \tag{4.13}$$

mit

$$\rho = dC/dt - 1 \quad \text{(drift)} \tag{4.14}$$

Abb. 4.5 auf Seite 32 veranschaulicht dies auf der linken Seite.

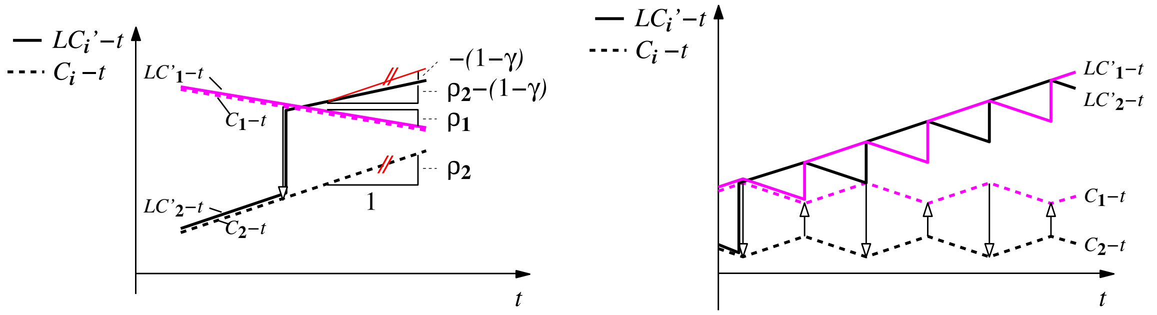


Abb. 4.5: Unbeschränktes Vorgehen der geregelten logischen Uhr bei zu großem γ

C_1 ist die am weitesten vorgehende Prozessoruhr, C_2 eine beliebige andere, deren momentane Gangabweichung so groß (positiv) ist, daß die Ungleichung (4.13) erfüllt ist, d.h. $\rho_2 - \rho_1 > 1 - \gamma$ gilt. Solange noch keine Nachricht das Vorstellen von LC_2' ausgelöst hat, sind die geregelten logischen Uhren LC_1' und LC_2' (durchgezogenen Linien, links von dem vertikalen Nachrichtenpfeil) noch identisch mit ihren Prozesoruhren C_1 und C_2 (gestrichelte Linien). Bei dem Nachrichtenaustausch wird nun LC_2' auf LC_1' vorgestellt. Danach besitzt LC_2' die Gangabweichung $\rho_2 - (1-\gamma)$. Da in diesem Beispiel die Ungleichung (4.13) erfüllt ist, geht nun LC_2' weiter vor als LC_1'.

Während der Satz 6 über die Beschränktheit des Nachgehens ebenfalls für LC' gilt[5], kann man zeigen, daß der Satz 5 über die Beschränktheit des Vorgehens für LC' nicht gilt. Man kann sogar zu jeder Wahl eines Wertes e^+ und einer beliebig kleinen positiven unteren Schranke für γ_i^j ein Beispiel konstruieren, indem für ein e_i^j die logische Uhr LC' mehr als e^+ vorgeht. Die rechte Seite von Abb. 4.5 zeigt ein Beispiel. Die gestrichelten Linien geben den Verlauf von C_1 und C_2 an, genauer gesagt von $C_1 - t$ und $C_2 - t$. Die momentane Gangabweichung der Prozessoruhren ist in diesem Beispiel periodisch schwankend. Während die eine Uhr eine

[5]Der Beweis kann direkt übernommen werden

4.4.3 Regler $\mathcal{A}$ – eine obere Schranke

Satz 7 *Es sei $\gamma_{\max}$ eine Konstante $\in (0,1)$. Wenn alle $\gamma_i^j \leq \gamma_{\max}$ und alle $|\rho| \leq \frac{1-\gamma_{\max}}{2}$ dann bleibt $LC_i' \leq \max_{k=1..n}(C_k)$.*

Beweis: Der Satz folgt direkt aus der Wirkungsweise der geregelten logischen Uhr, dargestellt in Abb. 4.5, links, auf Seite 32. Der Index $i = 1$ repräsentiere bei einer Nachricht den Prozeß, dessen Prozessoruhr C_i am weitesten vorgeht. Es ist im wesentlichen zu zeigen, daß nach dem Vorstellen von LC_2' auf LC_1' $(\leq C_1)$ nun LC_2' nicht gegenüber C_1 aufgrund des Termes $LC_2'(e_2^{j-1}) + \gamma_2^j(C_2(t(e_2^j)) - C_2(t(e_2^{j-1}))$ in Gleichung (4.10) vorgehen kann. Aus der Voraussetzung folgt, daß $-\frac{1-\gamma_{\max}}{2} \leq \rho_1$ und $\rho_2 \leq \frac{1-\gamma_{\max}}{2}$ ist. Es ist also zu zeigen, daß $\rho_2 - (1-\gamma) \leq \rho_1$. Beweis: $\rho_2 - 1 + \gamma \leq \frac{1-\gamma_{\max}}{2} - 1 + \gamma_{\max} = -\frac{1-\gamma_{\max}}{2} \leq \rho_1$

□

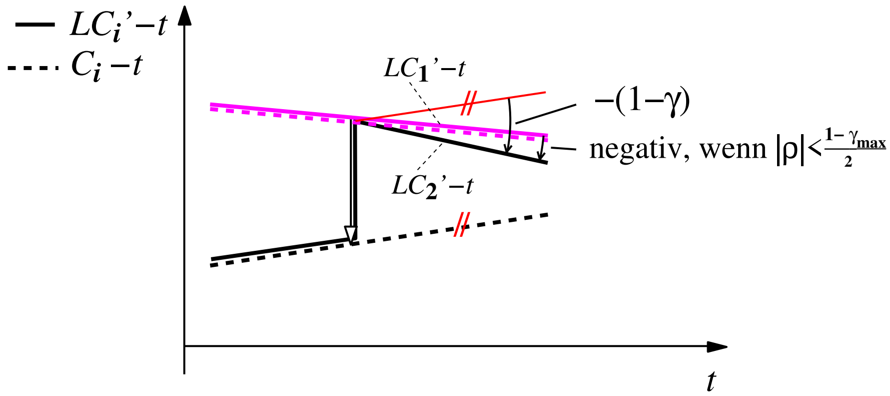


Abb. 4.8: $\gamma_{\max}$, eine Obergrenze für γ

Durch die Wahl einer oberen Grenze $\gamma_{\mathcal{A}} := \gamma_{\max}$ wird das Problem somit gelöst, wenn die Gangabweichungen der Prozessoruhren kleiner als $\frac{1-\gamma_{\max}}{2}$ sind, wie in Abb. 4.8 dargestellt: Es handelt sich hierbei um das gleiche Beispiel wie in Abb. 4.5, nur daß nun $(1-\gamma_{\max})$ so groß gewählt ist, daß LC_2' unter LC_1' bleibt.

In den meisten Fällen ist das Problem des zu starken Vorgehens von LC' damit gelöst. Für z.B. $\gamma_{\max} = 1 - 2 \cdot 10^{-5}$ ist somit das Problem für Gangabweichungen $|\rho| \leq 10^{-5}$ gelöst. Damit sind alle Probleme basierend auf typischen Quarzdrifts im Bereich von 10^{-6} und Gangabweichungen aufgrund schlechter Uhreneinstellungen im Bereich 10^{-5} gelöst. Der hierdurch induzierte Fehler bei Messungen innerhalb eines Prozesses ist auf $1 - \gamma_{\max} = 2 \cdot 10^{-5}$ beschränkt und damit vernachlässigbar. Wenn die Gangabweichung einer Prozessoruhr aber größer ist als $\frac{1-\gamma_{\max}}{2}$, dann ist es möglich, daß die geregelte logische Uhr LC' mehr als die am weitesten vorgehende Prozessoruhr vorgeht. Dieses Problem wird von den beiden folgenden Regelmechanismen gelöst werden.

4.4.4 Regler $\mathcal{B}$ – eine notwendige Regelung

Satz 8 *Es gilt* $\min_k(LC'_k - C_k) > 0 \implies \underset{i}{\exists} LC'_i > \max_{k=1..n}(C_k)$

Beweis: Für i = Index mit $C_i = \max_{k=1..n}(C_k)$ ergibt sich die rechte Seite, da aufgrund der Prämisse $LC'_i > C_i$ gilt.

□

Dieser Satz besagt, daß $\min_k(LC'_k - C_k) > 0$ ein hinreichendes Kriterium dafür ist, daß γ reduziert werden muß, bzw. wenn γ durch dieses Kriterium reduziert wird, dann ist diese Reduktion **notwendig**.

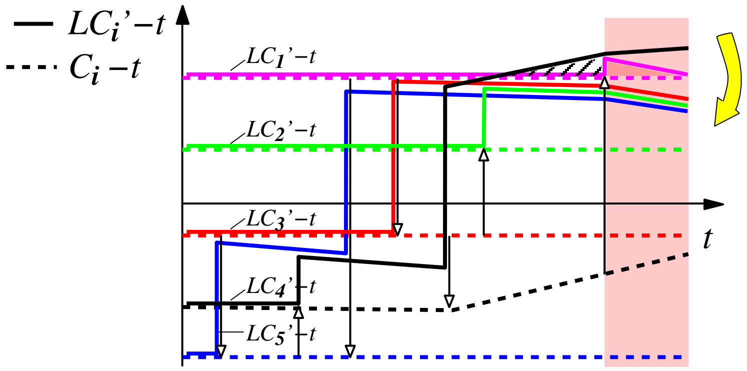


Abb. 4.9: Eine notwendige Regelung

Abb. 4.9 veranschaulicht dieses Kriterium. Das Beispiel ist so konstruiert, daß die Prozessoruhr C_4 eine positive Gangabweichung besitzt. Diese ist damit positiver, als die, der am weitesten vorgehenden Prozessoruhr, hier C_1 mit $\rho = 0$. Durch die aus vierter und fünfter Nachricht gebildeten Nachrichtenkette wird LC'_4 auf den Wert von LC'_1 vorgestellt.[6] Da die Uhrengeschwindigkeit von C_4 größer als die von C_1 ist, geht anschließend LC'_4 gegenüber LC'_1 vor. Dies ist mit dem schräg schraffierten Bereich gekennzeichnet. Die Prämisse von Satz 8 wird nun erfüllt, sobald LC'_1 größer wird als ihre Prozessoruhr. Dies wird durch das Vorstellen von LC'_1 aufgrund der letzten Nachricht von Prozeß 4 nach Prozeß 1 ausgelöst. Nachdem nun auch LC'_1 gegenüber C_1 vorgeht (siehe unterlegtes Dreieck) gehen nun in dem unterlegten Streifen alle logischen Uhren gegenüber ihren Prozessoruhren vor.

Da dieses Kriterium aber nicht aufzeigt, welche geregelten logischen Uhren weiter als erwünscht vorgehen (hier LC'_1 und LC'_4), ist es sinnvoll, global alle γ_i zu reduzieren. Dies führt zu der mit dem breiten, gebogenen Pfeil markierten Absenkung der Uhrengeschwindigkeiten aller LC'_i solange alle weiter als ihre zugehörige Prozessoruhr vorgehen. Für die Reduktion, d.h. für $\gamma_{\mathcal{B}}$ wird die in Abb. 4.10 dargestellte Regler-Charakteristik angewandt.

Die hierbei beim Divisor angewandte Normierung ist von minderer Bedeutung. Daher kann die Maximum-Norm, die bei einer großen Anzahl von Prozessen numerisch nicht skaliert, jederzeit auch durch die Norm mittels der 16. Potenz ersetzt werden. Die Rechenzeit, um die hier vorgesehene Maximum-Norm auszurechnen, ist proportional zur Anzahl n der Prozesse,

[6] Der resultierende Wert kann geringer sein, da die Differenz von realer Nachrichtenlaufzeit minus der minimalen Nachrichtenlaufzeit μ jeweils abgezogen wird, und da der Faktor γ bei einer Nachrichtenkette eine Verlangsamung der dazwischen liegenden Uhren (hier LC'_3) bewirkt.

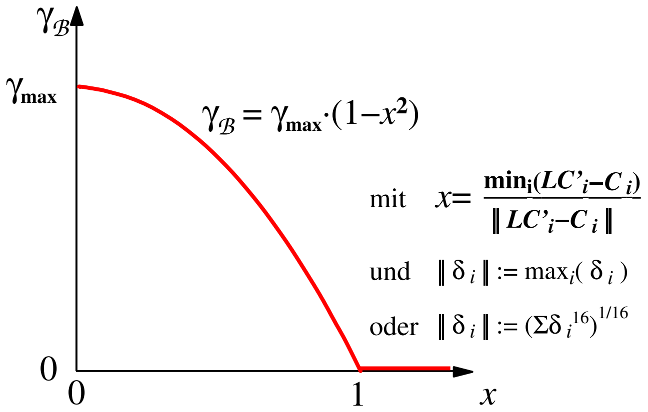


Abb. 4.10: Regler-Charakteristik der notwendigen Regelung

da bei jedem Schritt sich ein $\delta_i := LC'_i - C_i$ ändert, und somit das Maximum über alle n Werte von δ_i wieder neu gebildet werden muß, wenn der alte Wert von δ_i das Maximum lieferte. Ersetzt man hingegen die Maximum-Norm durch die Norm mittels der 16. Potenz, welche numerisch für n Prozesse zwischen der Maximum-Norm und deren $\sqrt[16]{n}$-fachen Wert liegt, d.h. für $n = 1000$ im Bereich des 1 bis 1.54-fachen, dann muß in jedem Schritt der alte Wert von $(\delta_i)^{16}$ von der Summe abgezogen, der neue hinzuaddiert und anschließend die Wurzel berechnet werden. Dieser Aufwand ist zwar pro Iteration groß, aber von der Anzahl der Prozesse unabhängig und für große n geringer als der Aufwand $O(n)$ für die Maximum-Norm.

Das in diesem Absatz angegebene hinreichende Kriterium ist aber nicht notwendig, d.h. es gibt Fälle, in denen eine Reduktion von γ nötig wäre, aber trotzdem $\min_k(LC'_k - C_k) = 0$ ist. Dies ist in dem Beispiel in Abb. 4.9 in dem schräg schraffierten Bereich bei LC'_4 gegeben. Mögliche Szenarien für diesen Fall sind:

- Kurze Phasen bevor durch eine Nachricht der Regler ausgelöst wird, wie in obigem Beispiel.
- Es gibt Gruppen von Prozesse, die zwar innerhalb der Gruppe kommunizieren, aber über längere Zeit nicht zwischen den Gruppen. Dann kann innerhalb einer Gruppe eine logische Uhr übermäßig vorgehen, obwohl das Kriterium nicht anspricht, da in einer anderen Gruppe mindestens eine logische Uhr nicht vorgeht, d.h. $LC'_i = C_i$ ist. Dies kann z.B. in Clustern von Parallelrechnern und darauf abgestimmter Anwendungssoftware passieren, vorausgesetzt die Parallelrechner besitzen intern keine Hardwaresynchronisation.

Nur das letztere Szenarium führt in der Regel zu Abweichungen, die einen weiteren Regelmechanismus erforderlich machen. Dieser wird im folgenden Abschnitt beschrieben.

4.4.5 Regler $\mathcal{C}$ – eine hinreichende Regelung

Der dritte Regler sei anhand der Abb. 4.11 erklärt.

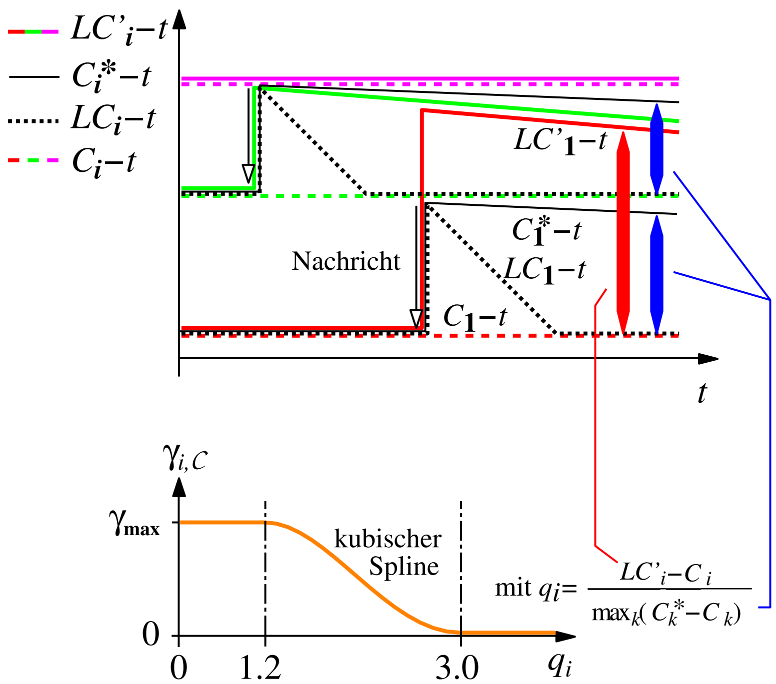


Abb. 4.11: Eine hinreichende Regelung

Zuerst sei die Kurve C_i^* definiert: Sie ist am Anfang identisch mit der einfachen logischen Uhr LC_i. Sie wird immer soweit vorgestellt, daß sie mindestens genauso weit vorgeht wie LC_i, und sie geht mit $[1-\frac{1}{2}(1-\gamma)]dC_i/dt$. Sie ist mit den schwarzen Linien dargestellt. Insgesamt ist aber für den dritten Regler nur $\max_i(C_i^* - C_i)$ relevant. Daher wird folgender Algorithmus angewandt, der im Term $-\frac{1}{2}(1-\gamma)dC_i/dt$ eine geringfügige und vernachlässigbare Abweichung besitzt.

Algorithmus 4:

Variablen:

MAX für $\max_i(C_i^* - C_i)$ für das letzte berechnete Ereignis e_i^j

LC_{MAX} für den den Wert von LC_i bei der letzten Erhöhung von MAX

Initialisierung:

$MAX = 0$ und $LC_{MAX} = LC_i(e_i^0)$ beim ersten Ereignis

Vorschrift bei jedem weiteren Ereignis e_i^j:

$MAX := MAX - \frac{1}{2}(1-\gamma_{\max})max(0, LC_i(e_i^j) - LC_{MAX})$

$MAX := \max(0, MAX)$

if $(LC_i(e_i^j) > C_i(e_i^j))$ then

$MAX = LC_i(e_i^j) - C_i(e_i^j)$

$LC_{MAX} = LC_i(e_i^j)$

endif

Für die Kurven C_i^* gilt, daß ihre Abstände zu C_i genauso beschränkt sind, wie bei den einfachen logischen Uhren LC_i. Aber C_i^* erfüllt nicht die Uhrenbedingung. Im Prinzip handelt

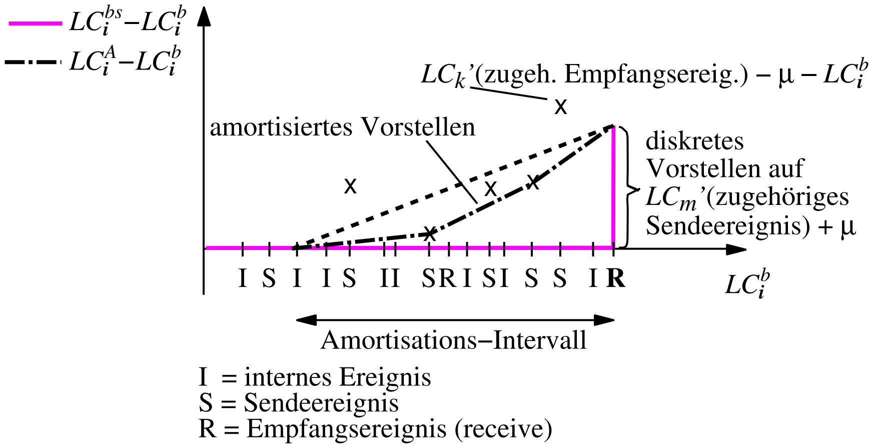


Abb. 4.12: Nachträgliche stückweise lineare Amortisation

Zu Beginn ist vom Anwender eine zu erwartende maximale Uhrendifferenz vorzugeben. Diese wird während der Berechnung der Amortisation erhöht, wenn eine Sprunghöhe über diesem Wert liegt.

Die durchgezogene Linie zeigt den Verlauf der geregelten logischen Uhr LC_i' eines Prozesses i mit einem Empfangsereignis **R**, bei dem die logische Uhr aufgrund des zugehörigen Sendeereignisses vorgestellt werden muß. Hierbei ist auf der Ordinate nicht LC_i' sondern $LC_i' - LC_i^b$ abgetragen, d.h. sie hat vor **R** den Wert Null und an R den Sprung, der durch die rückwärtige Amortisation ausgeglichen werden soll.

Die Kreuze (x) markieren innerhalb des Amortisationsintervalls zu jedem Sendeereignis des Prozesses i den Wert $LC_k'(\text{zugehöriges Empfangsereignis}) - \mu - LC_i^b$. Die Uhrenbedingung garantiert, daß diese Werte immer positiv sind.

Die gestrichelte Gerade verbindet den Wert Null am Anfang des Amortisationsintervalls mit der Sprunghöhe $= LC_i^{bs} - LC_i^b$ des Empfangsereignisses **R**. Falls es kein Ereignis vor dem Amortisationsintervall innerhalb des Prozesses i gibt, beginnt die gestrichelte Gerade beim ersten Ereignis mit dem Wert des untersten Kreuzes, und falls zudem das Amortisationsintervall keine Sendeereignisse enthält, ist die gestrichelte Gerade konstant auf der Sprunghöhe.

Wenn sich kein Kreuz unterhalb der gestrichelten Linie befindet, dann ergibt die gestrichelte Linie den rückwärtig amortisierten Wert LC_i^A der geregelten logischen Uhr in dem Amortisationsintervall. Andernfalls ist der Wert durch die untere konvexe Hüllkurve unterhalb der Kreuze zwischen den Endpunkten der gestrichelten Linie gegeben. In der Skizze ist sie strich-punktiert dargestellt.

Der Amortisationsgradient A bezeichnet die Steigung der konvexen Hülle im jeweiligen Teilintervall.

Im Gegensatz zu der von F. Schmuck und F. Cristian in [SC90] untersuchten Amortisation wird hier die diskrete Änderung nicht in einem an dem Synchronisationszeitpunkt beginnenden Zeitintervall ausgeglichen, sondern der Ausgleich wird rückwirkend in einem Intervall vor dem Synchronisationszeitpunkt durchgeführt. Die zeitliche Länge dieses Amortisationsintervalls ist durch die maximale Uhrendifferenz und die gewünschte Mindestgenauigkeit beschränkt, typischerweise in einem Workstation-Cluster auf z.B. $5\,\mathrm{ms}\,/\,1\,\% = 0.5\,\mathrm{sec}$. Eine rückwirkende Amortisation ist nur möglich, wenn der Einsatz der geregelten logischen Uhr eine derartige nachträgliche Änderung der Zeitstempel zuläßt. Bei der in Kap. 6 vorgestellten Filterimplementierung ist das möglich, bewirkt aber eine Vergrößerung der zu sortierenden Ereignismenge auf (*die Länge des Amortisationsintervalls*) $\cdot\,(\Delta E/\Delta t)$. Bei der in Kap. 7 skizzierten online Implementierung im Rahmen eines Debuggingwerkzeuges muß berücksichtigt werden, daß sich die Darstellung durch die Amortisation nachträglich ändert oder erst verspätet zur Verfügung steht. Die Verspätung ist einerseits durch die Länge des Amortisationsintervalls gegeben und andererseits dadurch, daß die Amortisation erst berechnet werden kann, wenn zu jedem Sendeereignis innerhalb des Amortisationsintervalls das zugehörige Empfangsereignis schon stattgefunden hat. Eine nachträgliche Änderung der Darstellung kann, z.B. bei einem Zeitliniendiagramm sukzessive im Rahmen des beim zeitlichen Fortschreiten notwendigen Scrollens realisiert werden, wobei dies für den Anwender in den meisten Fällen verborgen bleibt.

Es bleibt nun zu zeigen, daß die rückwärtig amortisierte geregelte logische Uhr LC^A die Uhrenbedingung (Def. 7) erfüllt:

Satz 9 *Die durch Algorithmus 5 definierte rückwärtig amortisierte geregelte logische Uhr LC^A erfüllt die Uhrenbedingung.*

Beweis: Lamport zeigte in [Lam78], daß es genügt, die Gültigkeit der Regeln IR1 und IR2 – siehe Fußnoten (2) und (3) auf Seite 27 – zu zeigen.

Für zwei aufeinanderfolgende Ereignisse innerhalb des selben Prozesses i gilt: Die Regel IR1 ist für LC_i^b erfüllt, da sie auch schon ohne den Term $LC_k'(e_k^l)+\mu_{k,i}$ in (4.10) erfüllt ist. Da die für die rückwärtige Amortisation addierten Werte mit der Ereignisfolge monoton steigend sind, ist auch für LC_i^A die Regel IR1 erfüllt. Daß die zu addierenden Werte monoton steigend sind, ergibt sich daraus,
(i) daß kein Kreuz unterhalb des Anfangswertes liegt und, daß der Anfangswert nicht kleiner oder gleich dem Endwert (Sprunghöhe) ist, d.h. die Steigung des ersten linearen Abschnitts ist immer größer oder gleich Null und,
(ii) daß die untere **konvexe** Hülle gebildet wurde, d.h. die Steigungen der linearen Abschnitte sind von Abschnitt zu Abschnitt zunehmend.

Für jede beliebige Nachricht ist die Regel IR2 zu zeigen: Die Bildung der **unteren** konvexen Hülle garantiert, daß der beim Sendeereignis addierte Wert immer kleiner oder gleich dem Wert des zugehörigen Kreuzes ist. Dies ergibt sich aus

$$\begin{aligned}
LC_i^A(\text{Sendeereignis}) &\overset{(1)}{\leq} LC_i^b + (LC_k'(\text{zugehöriges Empfangsereignis}) - \mu - LC_i^b) \\
&= LC_k'(\text{zugehöriges Empfangsereignis}) - \mu \\
&\overset{(2)}{\leq} LC_k^A(\text{zugehöriges Empfangsereignis}) - \mu \\
&\overset{(3)}{<} LC_k^A(\text{zugehöriges Empfangsereignis})
\end{aligned}$$

Die Abschätzungen gelten
(1) wegen der Definition der Kurve in Abb. 4.12,
(2) da die Amortisation die Zeitstempel nur vergrößert, und
(3) da $\mu > 0$.

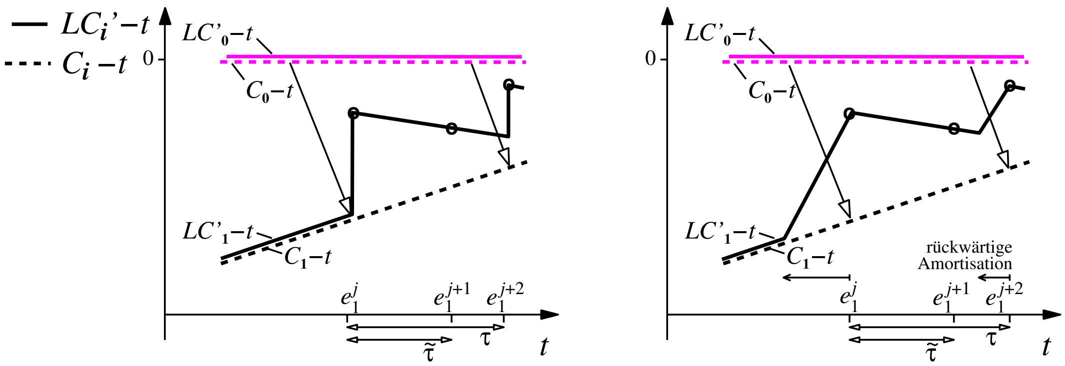


Abb. 5.2: Szenarium zur Fehlerabschätzung des Reglers $\mathcal{A}$, links *ohne* und rechts *mit* rückwärtiger Amortisation

Abb. 5.2 stellt den Verlauf von LC'_1 dar. Hierbei gilt

$$\begin{aligned}
LC'_1(e_1^j) - t(e_1^j) &= LC'_0(e_0^j) - t(e_0^j) + \mu_{0,1} - m(e_1^j) \\
&= \mu_{0,1} - m(e_1^j) && (5.3) \\
\text{mit} \quad m(e_1^j) &= {}_{\substack{\text{Nachrichtenlaufzeit} \\ \text{der empfangenen Nachricht}}} = t(e_1^j) - t(e_0^j) \\
LC'_1(e_1^{j+2}) - t(e_1^{j+2}) &= \mu_{0,1} - m(e_1^{j+2}) && (5.4) \\
\text{mit} \quad m(e_1^{j+2}) &= {}_{\substack{\text{Nachrichtenlaufzeit} \\ \text{der empfangenen Nachricht}}} = t(e_1^{j+2}) - t(e_0^{j+2}) \\
\text{Steigung von } LC'_1 - t \text{ nach } e_1^j &= \gamma_{\max}(1+\rho) \; - \; 1 && (5.5) \\
\text{Steigung von } LC'_1 - t \text{ mit Amortisation vor } e_1^{j+2} &= \gamma_{\max}(1+\rho)(1+A) \; - \; 1 && (5.6)
\end{aligned}$$

Im folgenden soll der relative Fehler der geregelten logischen Uhr LC'_1 im Vergleich zur exakten Zeit abgeschätzt werden, d.h. für

$$\epsilon^{LC'_1} = \left| \frac{LC'_1(e_1^l) - LC'_1(e_1^j)}{t(e_1^l) - t(e_1^j)} - 1 \right|$$

soll eine obere Schranke ermittelt werden.

Ohne rückwärtige Amortisation

ist der maximale Fehler durch zwei Sachverhalte bestimmt:

1. die Sprünge bei den Empfangsereignissen und
2. der durch $\gamma_{\max}$ und ρ bestimmte Verlauf zwischen zwei Sprüngen.

Im folgenden wird nur der Fehler bei zwei direkt aufeinanderfolgenden Ereignissen untersucht, da der maximale Fehler der Zeitbestimmung zwischen zwei beliebigen Ereignissen immer kleinergleich dem maximalen Fehler bei direkt aufeinanderfolgenden Ereignissen sein muß.

Der Fehler durch die Sprünge wirkt sich dann nur auf das Intervall aus, das vor dem Sprung liegt. Da sich innerhalb dieses Intervalls auch der Fehler durch $\gamma_{\max}$ und ρ entsteht, sind beide Fehlerarten zu addieren. Aus (5.3), (5.4) und (5.5) ergibt sich damit:

$$
\begin{aligned}
&\max_j \left| \frac{LC_1'(e_1^{j+1}) - LC_1'(e_1^j)}{t(e_1^{j+1}) - t(e_1^j)} - 1 \right| \\
&\leq \frac{\max_{\substack{j<l,\ e_1^j, e_1^l \text{ aufeinanderfolgende} \\ \text{Empfangsereignisse},\ \rho\in[-\rho_{\max},\rho_{\max}]}} \left|\mu_{0,1} - m(e_1^l) - (\mu_{0,1} - m(e_1^j) + (\gamma_{\max}(1+\rho) - 1)[t(e_1^l) - t(e_1^j)])\right|}{\min_j (t(e_1^{j+1}) - t(e_1^j))} \\
&\quad + \max_{\rho\in[-\rho_{\max},\rho_{\max}]} |\gamma_{\max}(1+\rho) - 1| \\
&\leq \frac{\max_{j<l,\ e_1^j, e_1^l \ \substack{\text{aufeinanderfolgende} \\ \text{Empfangsereignisse}}} \left|m(e_1^j) - m(e_1^l)\right|}{\min_j (t(e_1^{j+1}) - t(e_1^j))} \\
&\quad + \frac{\max_{j<l,\ e_1^j, e_1^l \ \substack{\text{aufeinanderfolgende} \\ \text{Empfangsereignisse}}} (t(e_1^l) - t(e_1^j))}{\min_j (t(e_1^{j+1}) - t(e_1^j))} [(1-\gamma_{\max}) + \rho_{\max}\gamma_{\max}] \\
&\quad + [(1-\gamma_{\max}) + \rho_{\max}\gamma_{\max}]
\end{aligned} \tag{5.7}
$$

Diese Abschätzung gilt nicht für den ersten Sprung, da hierbei die Sprunghöhe durch die Uhrendifferenz und nicht durch die Gangabweichung seit dem letzten Sprung bestimmt wird. Der erste Summand in (5.7) ist durch den Quotient der Varianz der Nachrichtenlaufzeiten geteilt durch den minimalen Abstand zweier Ereignisse bestimmt. Dieser Wert liegt normalerweise im Bereich mehrerer Zehnerpotenzen und impliziert damit die Notwendigkeit, die rückwärtige Amortisation einzusetzen. Der zweite Summand in (5.7) besteht aus einem Produkt. Der erste Multiplikator, der den maximalen Abstand zweier aufeinanderfolgender Empfangsereignisse in Relation zum minimalen Abstand zweier beliebiger Ereignisse setzt, kann ebenfalls mehrere Zehnerpotenzen betragen. Er wird aber durch den zweiten Multiplikator relativiert, der durch $(1 - \gamma_{\max})$ und $\rho_{\max}$ bestimmt ist und damit typischerweise 10^{-4} nicht übersteigt. Das Produkt kann natürlich immer noch wesentlich über der maximale tolerierbaren Fehlerrate ϵ (z.B. 10 %) liegen. Auch dieser Summand wird durch die rückwärtige Amortisation auf ein akzeptables Maß begrenzt werden. Der dritte Summand ist im Vergleich zu den ersten beiden vernachlässigbar.

Mit rückwärtiger Amortisation

ist der Fehler durch das Maximum der Beträge der Steigungen der beiden Kurvenstücke gegeben, also

$$
\begin{aligned}
\max_j \left| \frac{LC_1^A(e_1^{j+1}) - LC_1^A(e_1^j)}{t(e_1^{j+1}) - t(e_1^j)} - 1 \right| &\leq \max\{ \max_{\rho\in[-\rho_{\max},\rho_{\max}]} (|\gamma_{\max}(1+\rho) - 1|)\ , \\
&\qquad \max_{\rho\in[-\rho_{\max},\rho_{\max}]} (|\gamma_{\max}(1+\rho)(1+A) - 1|)\} \qquad (5.8) \\
&\leq A(1+\rho_{\max}) + (1-\gamma_{\max}) + \rho_{\max} \qquad (5.9)
\end{aligned}
$$

$\langle$[1] Herleitung:
Da $\gamma_{\max} < 1$ ist gilt für den ersten Term, daß das Maximum für $\rho = -\rho_{\max}$ eingenommen wird, d.h.:

$$
\begin{aligned}
\max_{\rho\in[-\rho_{\max},\rho_{\max}]} (|\gamma_{\max}(1+\rho) - 1|) &= -\left(\gamma_{\max}(1 + (-\rho_{\max})) - 1\right) \\
&= (1-\gamma_{\max}) + \gamma_{\max}\rho_{\max} \\
&\leq (1-\gamma_{\max}) + \rho_{\max}
\end{aligned}
$$

[1] Nebenrechnungen sind in Winkelklammern $\langle\rangle$ gesetzt. Sie sind zwar für das Verständnis des Beweises irrelevant, sind aber hilfreich bei der Verifikation des Beweises.

5.3 Charakteristik der Regler $\mathcal{B}$ und $\mathcal{D}$

Der Regler $\mathcal{B}$ soll anhand des folgenden Szenariums analysiert werden. Zuerst wird der Verlauf der geregelten logischen Uhr LC' bestimmt, um anschließend eine Fehlerabschätzung durchführen zu können.

Szenarium 2 *Die Ereignisse e_0^j und e_1^j seien jeweils das Senden und Empfangen zweier gegenläufiger Nachrichten zwischen den Prozessen 0 und 1. Die Nachrichtenlaufzeiten seien Null. Der zeitliche Abstand der Nachrichten sei τ. Die Uhr von Prozeß 0 geht exakt, d.h. $C_0(j) = j\tau$. Die Uhr von Prozeß 1 geht zum Zeitpunkt Null um D nach und sie geht um ρ schneller als C_0, d.h. $C_1(j) = j\tau(1+\rho) - D$. Das Szenarium ist in Abb. 5.3 dargestellt und entspricht dem linken Diagramm in Abb. 4.5.*

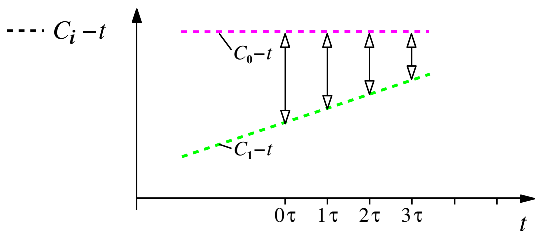


Abb. 5.3: Szenarium zur Fehlerabschätzung

Der gegenseitige Nachrichtaustausch ergibt, daß die geregelten logischen Uhren der beiden Prozesse and den Zeitpunkten $j\tau$ identisch sind, d.h. $LC'_0(e_0^j) = LC'_1(e_1^j)$. $LC'(j)$ sei daher eine Abkürzung für $LC'_0(e_0^j)$ und $LC'_1(e_1^j)$. Für LC' gilt:

$$\begin{aligned} LC'(0) &= 0 \\ LC'(j) &= \max(LC'(j-1) + \gamma_{j-1}(C_1(j) - C_1(j-1)), C_0(j)) \\ &= \max(LC'(j-1) + \gamma_{j-1}\tau(1+\rho), j\tau) \end{aligned} \tag{5.18}$$

mit

$$\begin{aligned} \gamma_0 &= \gamma_{\max} \\ \gamma_{j-1} &= \gamma_{\max}\left[1 - \left(\frac{LC'(j-1) - C_0(j-1)}{D}\right)^2\right] \\ &= \gamma_{\max}\left[1 - \left(\frac{LC'(j-1) - (j-1)\tau}{D}\right)^2\right] \quad \text{für } \gamma_{\max}(1+\rho) > 1 \end{aligned} \tag{5.19}$$

Die ersten beiden Formeln definieren die Strecke des Regelkreises, die letzten beiden Formeln stellen den Regler dar. Damit der Regler $\mathcal{B}$ zum Einsatz kommt, muß $\gamma_{\max}(1+\rho) > 1$ sein, d.h.

$$\rho \quad > \quad \frac{1 - \gamma_{\max}}{\gamma_{\max}} \tag{5.20}$$

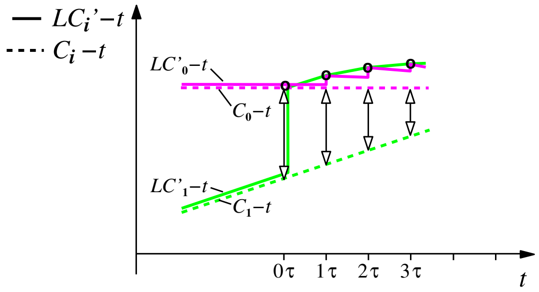


Abb. 5.4: Die geregelte logische Uhr in obigem Szenarium

Abb. 5.4 stellt den Verlauf der geregelten logischen Uhr schematisch dar. Im Gegensatz zu Abb. 4.8 ist hier nun die Regelung $\mathcal{B}$ und nicht $\mathcal{A}$ für die Begrenzung des Überschwingens verantwortlich. Um den Verlauf von LC' in Abhängigkeit der Größen τ, ρ, $\gamma_{\max}$ und D zu analysieren, wird zuerst eine transformierte Kurve analysiert, und anschließend die Erkenntnisse wieder auf LC' übertragen. Die Transformation ist definiert durch:

$$O_j = LC'(j) - C_0(j) \qquad \text{Offset} \tag{5.21}$$

$$\Omega_j = \frac{O_j}{D} \qquad \text{Relation Offset zu } D \tag{5.22}$$

$$\theta = \frac{\tau}{D} \qquad \text{Relation } \tau \text{ zu } D \tag{5.23}$$

$$R = \gamma_{\max}(1+\rho) \qquad \text{relativierte Uhrenabweichung} \tag{5.24}$$

Die Bedingung (5.20) für den Einsatz des Reglers $\mathcal{B}$ ist äquivalent zu $R > 1$. Das Bildungsgesetz für LC' ergibt dann folgende transformierte Darstellung:

$$\begin{aligned} \Omega_0 &= 0 \\ \Omega_j &= \max(0, \Omega_{j-1} - \Omega_{j-1}^2 \theta R + \theta(R-1)) \end{aligned}$$

⟨[4] Herleitung:

$$\begin{aligned}
LC'(j) &= \max\left(LC'(j-1) + \gamma_{\max}\left[1 - \left(\frac{LC'(j-1)-(j-1)\tau}{D}\right)^2\right]\tau(1+\rho), j\tau\right) \\
O_j &= LC'(j) - C_0(j) = LC'(j) - j\tau \\
&= LC'(j) - (j-1)\tau - \tau \\
&= \max\left(LC'(j-1) - (j-1)\tau - \tau + \gamma_{\max}\left[1 - \left(\frac{LC'(j-1)-(j-1)\tau}{D}\right)^2\right]\tau(1+\rho), j\tau - j\tau\right) \\
&= \max\left(O_{j-1} - \tau + R\tau\left[1 - \left(\frac{O_{j-1}}{D}\right)^2\right], 0\right) \\
&= \max\left(O_{j-1} - \tau + R\tau - R\tau\left(\frac{O_{j-1}}{D}\right)^2, 0\right) \\
&= \max\left(O_{j-1} + (R-1)\tau - R\tau\left(\frac{O_{j-1}}{D}\right)^2, 0\right) \\
\Omega_j &= \max(\Omega_{j-1} + (R-1)\theta - R\theta\Omega_{j-1}^2, 0)
\end{aligned}$$

⟩[4]

[4] siehe Fußnote 1 auf Seite 48

Aus (5.53) folgt, daß dann $LC'(j-1) - C_0(j-1) > O_\infty$ sein muß.
Aus (5.54) folgt dann, daß $LC'(j-1) - C_0(j-1) - O_\infty \le 0.08867\, O_\infty$ ist.
Aus (5.53) folgt dann, daß $|LC'(j) - C_0(j) - O_\infty| \le 0.08867\, O_\infty$ ist. Zusammen ergibt dies, daß $|(LC'(j) - C_0(j)) - (LC'(j-1) - C_0(j-1))| \le 2 \cdot 0.08867 \cdot O_\infty$ ist und

$$\begin{aligned}
\left|\frac{LC'(j) - LC'(j-1)}{\tau} - 1\right| &= \left|\frac{(LC'(j) - C_0(j)) - (LC'(j-1) - C_0(j-1))}{\tau}\right| \\
&\le (2 \cdot 0.08867)\frac{O_\infty}{\tau} \quad = \quad 0.17734\frac{fO_\infty}{\tau}\frac{1}{f} \\
&\overset{f > \frac{1}{2}}{\le} 0.17734\frac{fO_\infty}{\tau}\frac{1}{1/2} \\
&= 0.35468(\rho - (1-\gamma_{\max}) - (1-\gamma_{\max})\rho) \\
&\le 0.35468(\rho - (1-\gamma_{\max}))
\end{aligned}$$

Für ein beliebiges Intervall ist somit der Fehler $\le \max(1, 0.35468)(\rho - (1-\gamma_{\max})) = \rho - (1-\gamma_{\max})$. Mit der obigen Überlegung gilt diese Abschätzung dann auch für beliebiges n. Somit ist die linke Seite des Minimums in (5.59) gezeigt.
Die rechte Seite ergibt sich aus (5.54) mit folgender Abschätzung:

$$\begin{aligned}
\left|\frac{LC'(j+n) - LC'(j)}{n\tau} - 1\right| &= \left|\frac{(LC'(j+n) - C_0(j+n)) - (LC'(j) - C_0(j))}{n\tau}\right| \\
&\overset{(5.54)}{\le} \frac{(1.08867 - 0) \cdot O_\infty}{n\tau} \quad = \quad 1.08867\frac{fO_\infty}{\tau}\frac{1}{f} \\
&\overset{f \ge \frac{1}{2}}{\le} \frac{1.08867}{n}\frac{fO_\infty}{\tau}\frac{1}{1/2} \quad \le \quad \frac{2.17734}{n}(\rho - (1-\gamma_{\max}))
\end{aligned}$$

(5.60) Analog zum letzten Beweis folgt aus (5.55) und $f \ge 1$

$$\begin{aligned}
\left|\frac{LC'(j+n) - LC'(j)}{n\tau} - 1\right| &\le \frac{1.41276 - 0}{1 \cdot n}(\rho - (1-\gamma_{\max})) \\
&= \frac{1.41276}{n}(\rho - (1-\gamma_{\max}))
\end{aligned}$$

(5.60) Ebenso analog zum letzten Beweis folgt aus (5.56) und $f > 1.224635$

$$\begin{aligned}
\left|\frac{LC'(j+n) - LC'(j)}{n\tau} - 1\right| &\le \frac{1.51338 - 0}{1.224635 \cdot n}(\rho - (1-\gamma_{\max})) \\
&\le \frac{1.2358}{n}(\rho - (1-\gamma_{\max}))
\end{aligned}$$

(5.62) Für jedes Intervall ist der Fehler:

$$\begin{aligned}
\left|\frac{LC'(j) - LC'(j-1)}{\tau} - 1\right| &= \left|\frac{(LC'(j) - C_0(j)) - (LC'(j-1) - C_0(j-1))}{\tau}\right| \\
&\overset{(5.57)}{=} \frac{fO_\infty}{\tau} \\
&= \rho - (1-\gamma_{\max}) - (1-\gamma_{\max})\rho \le \rho - (1-\gamma_{\max})
\end{aligned}$$

Mit der Überlegung zu (5.58) gilt diese Abschätzung dann auch für beliebiges n.

□

Satz 13 läßt sich folgendermaßen zusammenfassen:

Satz 14 *Im Szenarium 2 auf S. 51 gilt für $\gamma_{\max}(1+\rho) > 1$ folgende Fehlerabschätzung für die lokale Bestimmung des Zeitintervalls zwischen zwei Ereignissen e^j und e^l mit $j \neq l$*

$$\left|\frac{LC'(j)-LC'(l)}{C_0(j)-C_0(l)} - 1\right| \leq 1.41276(\rho - (1-\gamma_{\max})). \tag{5.63}$$

Die folgenden Szenarien relativieren Szenarium 2 in zweierlei Hinsicht:

- Szenarium 3 verzichtet auf die symmetrische Nachrichtenübertragung, und
- Szenarium 4 verzichtet auf den konstanten zeitlichen Abstand zwischen zwei Nachrichtenübertragungen. Das Szenarium idealisiert diesen Fall in der Form, daß abwechselnd ein kurzer und ein langer Nachrichtenabstand vorhanden ist.

Szenarium 3 *Dieses Szenarium ist wie Szenarium 2 auf S. 51, aber zum Zeitpunkt $j = 0$ wird nur eine Nachricht von Prozeß 0 nach Prozeß 1 gesandt und bei allen weiteren Zeitpunkten ($j > 0$) nur in der Gegenrichtung.*

Das Verhalten von LC' ist im Szenarium 3 in den Fällen $f \in (0, 1.272858]$ des Satzes 10 identisch mit dem Verhalten im Szenarium 2, d.h. gemäß (5.34) bis (5.41). Für die Fehleranalyse sei nur der erste Zyklus betrachtet, d.h. bis LC'_1 erstmals unter C_0 fällt. Alle weiteren Zyklen sind ähnlich wie der erste Zyklus, sobald durch das schneller Gehen von C_1 auch LC'_1 wieder gegenüber C_0 vorgeht. Für die Fälle $f \in [1.272859, \infty)$ muß zwischen LC'_0 und LC'_1 unterschieden werden, da hier LC'_0 auf C_0 absinkt, während LC'_1 unter C_0, aber höchstens auf C_1 absinkt.
Es sei j der Zeitpunkt, an den LC'_1 erstmals unter C_0 fällt.

Für $f \in (1, \sqrt{2})$ gilt gemäß (5.55) und (5.56), daß

$$0 \leq LC'_1(j-1) - C_0(j-1) \leq 1.51338\, O_\infty$$

Der Fall ist in Abb. 5.8 dargestellt.

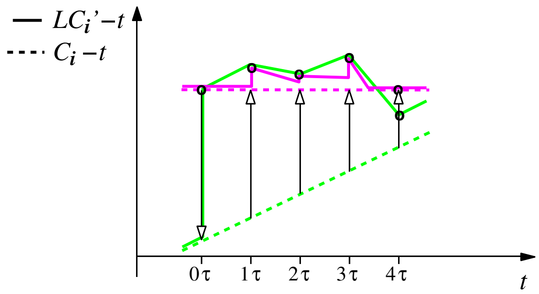


Abb. 5.8: LC'_i im Szenarium 3 für $f \in (1, \sqrt{2})$

Damit kann man den Fehler folgendermaßen abschätzen:

$$
\begin{aligned}
\left|\frac{LC'(j)-LC'(j-1)}{\tau}-1\right| &= \left|\frac{\underbrace{(LC'(j)-C_0(j))}_{<0}-\underbrace{(LC'(j-1)-C_0(j-1))}_{\geq 0}}{\tau}\right| \\
&= -\frac{(LC'(j)-C_0(j))-(LC'(j-1)-C_0(j-1))}{\tau} \\
&\overset{(5.18)}{=} -\frac{(LC'(j-1)+\gamma_{j-1}\tau(1+\rho)-(C_0(j-1)+\tau))-(LC'(j-1)-C_0(j-1))}{\tau} \\
&= 1-\gamma_{j-1}(1+\rho) \\
&\overset{(5.19)}{=} 1-\gamma_{\max}\left[1-\left(\frac{\underbrace{LC'(j-1)-C_0(j-1)}_{0\leq\ldots\leq 1.51338\,O_\infty}}{D}\right)^2\right](1+\rho) \\
&\leq 1-\gamma_{\max}\left[1-\left(\frac{1.51338\,O_\infty}{D}\right)^2\right](1+\rho) \\
&= 1-\gamma_{\max}(1+\rho)+1.51338^2\,(\rho-(1-\gamma_{\max})(1+\rho)) \\
&= -(\rho-(1-\gamma_{\max})(1+\rho))+1.51338^2\,(\rho-(1-\gamma_{\max})(1+\rho)) \\
&= 2.02676(\rho-(1-\gamma_{\max})(1+\rho)) \\
&\leq 2.03(\rho-(1-\gamma_{\max}))
\end{aligned}
$$

Für den Fall $f \geq \sqrt{2}$ besteht der erste Zyklus nur aus den beiden ersten Intervallen. Es gilt:

$$
\begin{aligned}
LC_1'(1)-C_0(1) &\overset{(5.57)}{=} fO_\infty \overset{(5.50)}{=} \rho-(1-\gamma_{\max})(1+\rho) \\
\gamma_1 &\overset{(5.19)}{=} \gamma_{\max}\left[1-\left(\frac{fO_\infty}{D}\right)^2\right] \\
LC_1'(2)-C_0(2) &= \max(LC'(1)+\gamma_1\tau(1+\rho), 2\tau(1+\rho)-D)-2\tau \\
&= \max(fO_\infty+\tau-\gamma_{\max}\left[1-\left(\frac{fO_\infty}{D}\right)^2\right]\tau(1+\rho), \\
&= \max(fO_\infty-\tau(\gamma_{\max}(1+\rho)-1)-\gamma_{\max}\tau(1+\rho)\left(\frac{fO_\infty}{D}\right)^2, 2\tau\rho-D) \\
&\overset{(5.50)}{=} \max(-\gamma_{\max}\tau(1+\rho)\left(\frac{fO_\infty}{D}\right)^2, 2\tau\rho-D) \\
&\overset{(5.50)}{=} \max(-\tau^3\frac{\gamma_{\max}(1+\rho)[\rho-(1-\gamma_{\max})(1+\rho)]^2}{D^2}, 2\tau\rho-D)
\end{aligned}
$$

Für die folgende Fallunterscheidung sei vorausgesetzt, daß $\sqrt[3]{\rho} \ll 1$. Die Fälle sind in Abb. 5.9 dargestellt.

1. Wenn $\tau \lesssim \frac{D}{\sqrt[3]{\gamma_{\max}(1+\rho)[\rho-(1-\gamma_{\max})(1+\rho)]^2}} \simeq D\,\rho^{-2/3}$, d.h. wenn $f \lesssim \sqrt[6]{1/\rho}$, dann ist die linke Seite der Maximumbildung maßgebend und

$$\begin{aligned}
\left|\frac{LC'(j)-LC'(j-1)}{\tau}-1\right| &= -\frac{(LC_1'(2)-C_0(2))-(LC_1'(1)-C_0(1))}{\tau} \\
&= \rho-(1-\gamma_{\max})(1+\rho)+\tau^2\frac{\gamma_{\max}(1+\rho)[\rho-(1-\gamma_{\max})(1+\rho)]^2}{D^2} \\
&\leq \rho-(1-\gamma_{\max})(1+\rho)+\sqrt[3]{\gamma_{\max}(1+\rho)[\rho-(1-\gamma_{\max})(1+\rho)]^2} \\
&\simeq [\rho-(1-\gamma_{\max})]+\sqrt[3]{[\rho-(1-\gamma_{\max})]^2}
\end{aligned}$$

2. Wenn $\tau > \frac{D}{\sqrt[3]{\gamma_{\max}(1+\rho)[\rho-(1-\gamma_{\max})(1+\rho)]^2}} \simeq D\,\rho^{-2/3}$, d.h. wenn $f \gtrapprox \sqrt[6]{1/\rho}$, dann ist die rechte Seite der Maximumbildung maßgebend und

$$\begin{aligned}
\left|\frac{LC'(j)-LC'(j-1)}{\tau}-1\right| &= -\frac{(LC_1'(2)-C_0(2))-(LC_1'(1)-C_0(1))}{\tau} \\
&\leq \rho-(1-\gamma_{\max})(1+\rho)\frac{D}{\tau} \\
&\leq \rho-(1-\gamma_{\max})(1+\rho)+\sqrt[3]{\gamma_{\max}(1+\rho)[\rho-(1-\gamma_{\max})(1+\rho)]^2} \\
&\simeq [\rho-(1-\gamma_{\max})]+\sqrt[3]{[\rho-(1-\gamma_{\max})]^2}
\end{aligned}$$

Zusammenfassend gilt dann:

$$\left|\frac{LC'(j)-LC'(l)}{C_0(j)-C_0(l)}-1\right| \leq \begin{cases} 2.03[\rho-(1-\gamma_{\max})] & \text{für } f<\sqrt{2} \\ [\rho-(1-\gamma_{\max})]+\sqrt[3]{[\rho-(1-\gamma_{\max})]^2}) & \text{für } f\geq\sqrt{2} \end{cases}$$

für f definiert gemäß (5.28) und (5.29).

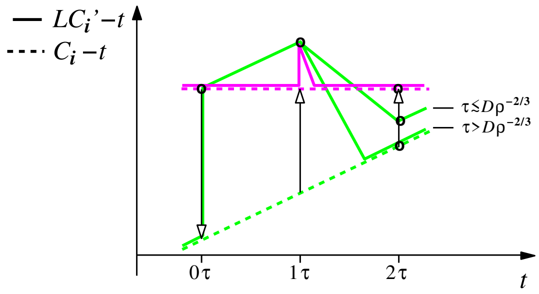


Abb. 5.9: LC_i' im Szenarium 3 für $f \geq \sqrt{2}$

Szenarium 4 *Dieses Szenarium ist wie Szenarium 3 auf S. 62, die Intervalle sind jedoch unterschiedlich lang:* $C_0(j)-C_0(j-1)=\tau_j$ *und* $C_1(j)-C_1(j-1)=\tau_j(1+\rho)$. *Die Intervallänge variiert um höchstens* $\kappa=\tau_{\max}/\tau_{\min}$ *mit* $\tau_{\max}=\max_{j=1,2,\ldots}\tau_j$ *und* $\tau_{\min}=\min_{j=1,2,\ldots}\tau_j$.

Die Fehlerabschätzung zu diesem Szenarium ist identisch bis auf die Fälle, in denen LC' auf C absinkt, d.h. für $f \geq 1.272859$. Da bei der Fehlerabschätzung zum Szenarium 3 für $f<\sqrt{2}$ die Maximum-Bildung mit C_1 nicht berücksichtigt werden mußte, ändert sich in diesem Fall die Fehlerabschätzung auch nicht. Das gleiche gilt für $f \geq \sqrt{2}$ und $f \leq \sqrt[6]{1/\rho}$. Nur für $f \geq \sqrt{2}$ und $f \geq \sqrt[6]{1/\rho}$ kann durch ein kurz nach dem zweiten Ereignis folgendes drittes Ereignis ein um κ größerer Fehler auftreten, da nun die Kurve um den Faktor κ steiler sein kann als im Szenarium 2, bevor der rechte Teil in der Maximumbildung in (5.18) relevant wird. Zusammenfassend gilt dann:

sortierten Ergebnisse der einzelnen Prozesse in einen global geordneten Datenstrom mischen. Bei Implementierung mittels eines vollständigen Binärbaums ist der Aufwand proportional zu $\log_2(\textit{Anzahl der Prozesse})$.

Es ist nicht zu empfehlen, den im vorigen Kapitel angegebenen Sortieralgorithmus zu verwenden, da die im Vergleich zur maximalen Uhrendifferenz große Länge des Amortisationsintervalls die Längen der auftretenden Sortierintervalle stark vergrößern. Daher ist in der Regel der zu $\log_2(\textit{Anzahl der Prozesse})$ proportionale Mischaufwand geringer als der Sortieraufwand, wenn man den Algorithmus aus dem vorherigen Kapitel übernehmen würde.

Tabelle 6.1 zeigt die Ausführungszeiten für einen 1.9 MB großen binären VAMPIR-Tracefile [NA95, NAW+96] mit von 16 Prozessen erzeugten 56622 Ereignissen. Für die zweite Spalte wurde eine Version mit entferntem Berechnungsalgorithmus eingesetzt. Die dritte Spalte wurde nicht direkt gemessen, sondern ergibt sich aus der Differenz der ersten beiden. Die ersten 4 Zeilen wurden mit lokalen Platten ermittelt. Die Tabelle gibt den Hinweis, daß langfristig die überproportionale Steigerung der Rechnergeschwindigkeiten gegenüber den Plattengeschwindigkeiten dazu führt, daß der Hauptaufwand im Lesen und Schreiben der Tracefiles liegen wird. Die letzte Zeile zeigt diesen Effekt schon heute, wenn man einen langsameren Plattenzugriff zugrunde legen muß, wie z.B., wenn auf die Daten über ein lokales Netz zugegriffen werden muß.

Da es sich bei der geregelten logischen Uhr um einen Filteralgorithmus handelt, kann er direkt in das Visualisierungswerkzeug eines Monitors integriert werden. Dann entfällt der Lese- und Schreibaufwand vollständig. Dies ist ein wesentlicher Vorteil gegenüber allen Algorithmen, die zwei Durchläufe benötigen, wie z.B. die in Kap. 3.3 dargestellten Verfahren von Duda et al. und Hofmann.

6.3 Test des Algorithmus

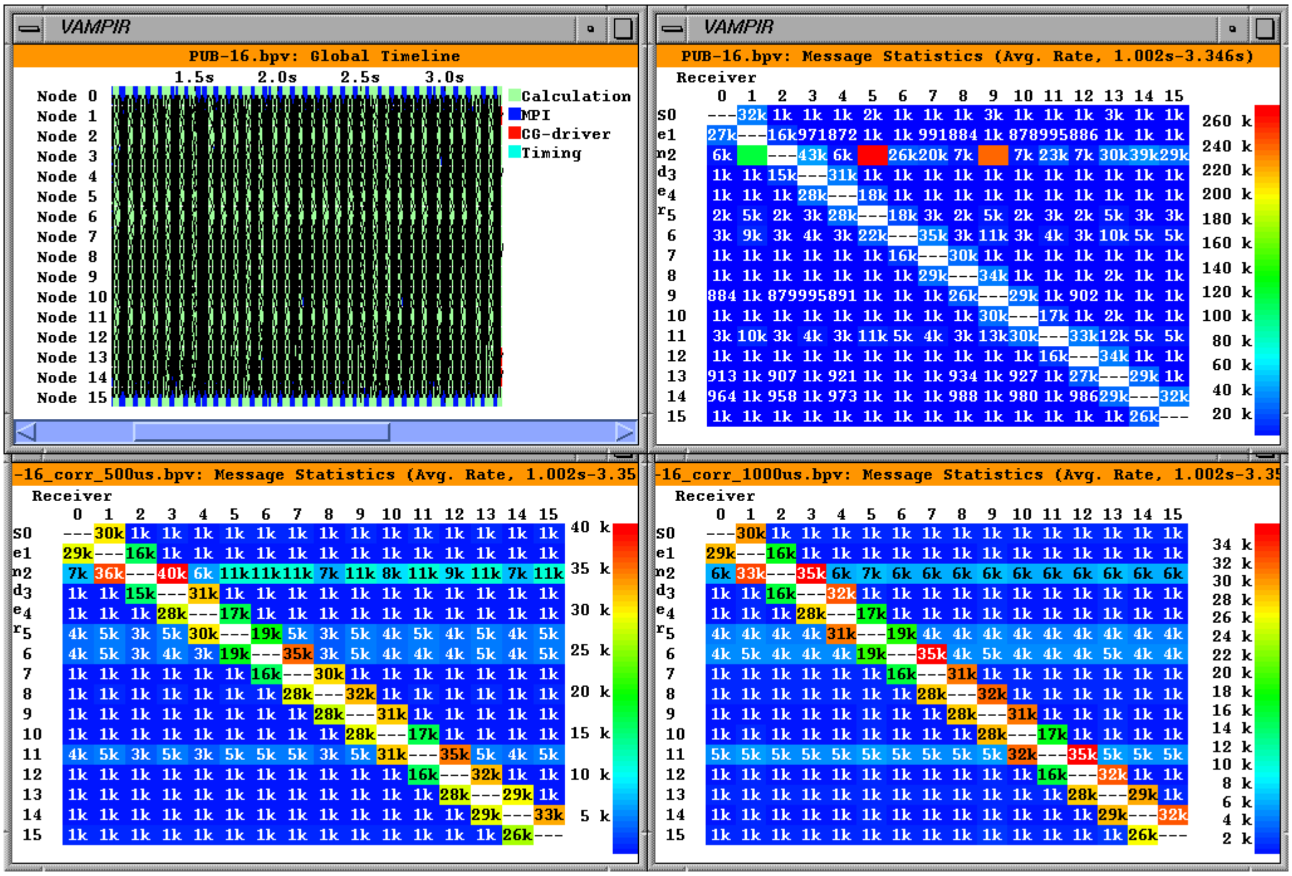
Der Algorithmus wurde mit drei Beispielen getestet:

1. Mittels eines Tracefiles einer realen Anwendung mit 16 Prozessen auf einem Parallelrechner mit ungenügender Uhrensynchronisation.
2. Ein Beispiel mit großem Uhrentick. Es handelt sich hierbei um das Beispiel 1 in [Rab96]. Ohne die Regelung, d.h. bei $\gamma \equiv 1$, würde bei diesem Beispiel LC' unbegrenzt wachsen.
3. Mittels einer simulierten FE-Rechnung.

Diese sind in den folgenden Kapiteln beschrieben.

6.3.1 Tracefile einer realen Anwendung

Abb. 6.3 auf S. 72 zeigt links oben einen Ausschnitt des Zeitliniendiagramms einer realen Anwendung und rechts oben das zugehörige Diagramm der durchschnittlichen Transferraten. Beide Diagramme sind auf der Basis der nicht korrigierten Zeitstempel der Prozessoruhren erstellt. Man sieht bei von Knoten 2 ausgehenden Nachrichten extrem unrealistische Transferraten im Bereich 120 (nach Knoten 1), 270 (nach Knoten 5) und 240 (nach Knoten 9) kB/sec. Nach einer Korrektur mittels der geregelten logischen Uhr mit $\mu = 500\mu$s ergibt sich die wesentlich realistischere Darstellung links unten. Hierbei war nur der Regler $\mathcal{A}$ mit




```
> pvcorrect -maxerr 0.1 -mmd 500 PUB-16.bpv

Differences new timestamps - old timestamps at the last event in [ms]
Node   i=      0      1      2      3      4      5      6      7
    0+i   0.0000 1.1534 0.1186 0.3070 0.1886 1.1617 0.4447 0.4085
    8+i   0.2006 1.2684 0.0000 0.4334 0.0000 1.2413 0.8874 0.4823

Maximal clock differences = 1302.92 us,  used start-value = 1000.00 us

ADVICE -- use for correcting similar tracefiles:  -cldiff 1303

Minimal difference between the original timestamps of two events
in the same process = 5.000 us

Minimal message delay between 2 processes =
   (min=620.200us,  avg=2093.456us,  max=3157.700us)
with min/avg/max over all paires of processes with messages in both
directions.   These values include an error <= 2*(max. clock drifts
over the whole time)

The correction of the timestamps causes additional errors on the length
of time intervals between two successive events in the same process:
 13689 intervals with        error  =   0%
 42617 intervals with   0% < error <= 0.1% : avg.= 0.002%, max= 0.003%
   300 intervals with 0.1% < error          : avg.= 0.468%, max= 1.137%
Error summary:
 56606 intervals with 0.0% <=error <= 1.137% : avg.= 0.004%
```

```
> pvcorrect -maxerr 0.1 -mmd 1000 PUB-16.bpv

Differences new timestamps - old timestamps at the last event in [ms]
Node   i=      0      1      2      3      4      5      6      7
    0+i   0.0152 1.4885 0.0000 0.6455 0.1850 1.5002 0.7830 0.7470
    8+i   0.2067 1.6069 0.0766 0.7774 0.1430 1.5825 0.9634 0.8395

Maximal clock differences = 1858.71 us,  used start-value = 1000.00 us

ADVICE -- use for correcting similar tracefiles:  -cldiff 1859

Minimal difference between the original timestamps of two events
in the same process = 5.000 us

Minimal message delay between 2 processes =
   (min=620.200us,  avg=2093.456us,  max=3157.700us)
with min/avg/max over all paires of processes with messages in both
directions.   These values include an error <= 2*(max. clock drifts
over the whole time)

ADVICE -- use for correcting similar tracefiles:  -mmd 496.160
but this value can be meaningless if the clock drifts are to large.

The correction of the timestamps causes additional errors on the length
of time intervals between two successive events in the same process:
  5394 intervals with        error  =   0%
 49484 intervals with   0% < error <= 0.1% : avg.= 0.002%, max= 0.077%
  1728 intervals with 0.1% < error          : avg.= 0.982%, max= 8.522%
Error summary:
 56606 intervals with 0.0% <=error <= 8.522% : avg.= 0.032%
```

Abb. 6.3: Zeitliniendiagramm und durchschnittliche Transferraten bei einer realen Anwendung,
Transferraten rechts oben basierend auf C,
links unten basierend auf LC', berechnet mit $\mu = 500\mu$s,
rechts unten basierend auf LC', berechnet mit $\mu = 1000\mu$s,
jeweils darunter mit Fehleranalyseprotokoll des geregelten logischen Uhrenprogramms

laufzeit (halbe Roundtriptime) lag bei ca. 1150 ns. Die Abbildung zeigt, daß die Uhren im Sekundentakt mittels abruptem Vor- und Zurücksetzen synchronisiert werden und hierbei bezüglich der Uhrendifferenzen Sprünge im Bereich von 1000 ns auftreten können. Da außerdem die Uhrendifferenzen im Bereich bis 2300 ns liegen und das Betriebssystem jederzeit Partitionen verschieben kann, sind zudem Sprünge in diesem Bereich beim Monitoring von parallelen Anwendungen zu berücksichtigen. Dies heißt, daß auf der T3E bei `shmem`-basierten Programmen die Nachrichtenlaufzeiten geringer sein können als die Uhrenfehler. Die geregelte logische Uhr kann derartige Fehler dann ausgleichen. Die geregelte logische Uhr kann aber auch eingesetzt werden, wenn die Uhren mittels eines linearen Ausgleichs synchronisiert werden, d.h. das Vor- und Zurücksetzen über einen Zeitraum linear verteilt wird.

Beim Einsatz der geregelten logischen Uhr ist hierbei zu beachten, daß derartige Synchronisationsverfahren die maximale Gangabweichung der Uhren C_i ändert. Dies ist bei der Wahl der Werte für $\gamma_{\mathcal{A}}$, $\gamma_{\mathcal{D}}$ und A_{def} zu berücksichtigen.

7.5 Existierende Verfahren zur nachträglichen Uhrenkorrektur

Wie schon in Kap. 3.3 aufgezeigt, ist die nachträgliche Bestimmung der Uhrenabweichungen eine Alternative zur geregelten logischen Uhr. Im wesentlichen werden die in Kap. 3.3 angegebenen Verfahren von Duda et al. [DHHB87], Jézéquel [Jez89a], Hofmann [Hof93a, Hof93b], Hofmann und Hilgers [HH98, Hil96] und Babaoğlu und Drummond [BD87, DB93] mit der geregelten logischen Uhr verglichen. Diese Verfahren unterscheiden sich bezüglich der Anwendbarkeit in folgenden Aspekten:

- Werden bestimmte Kommunikationstopologien vorausgesetzt?

  Babaoğlu und Drummond setzen regelmäßig wiederkehrende vollständige Kommunikationsbeziehungen voraus, während Duda et al., Jézéquel, sowie Hofmann und Hilgers keine Einschränkungen machen.

- Wird die Uhrenbedingung nach Anwendung der Uhrenkorrektur erfüllt?

  Der Algorithmus von Babaoğlu und Drummond hat dies im Gegensatz zu den anderen nicht zum Ziel. Er wird daher im folgenden nicht mehr erörtert.

- Müssen die Uhrengeschwindigkeiten konstant sein?
  – und –
  Werden Uhrenkorrekturen für eine beliebige Anzahl von Prozessoren ermittelt?

  Das Verfahren mit Bildung einer konvexen Hülle von Duda et al. beherrscht ein beliebiges Verhalten der Uhrengeschwindigkeiten beim 2-Prozeß-Problem, d.h. daß Traces von parallelen Anwendungen mit jeweils 2 Prozessen korrigiert werden können. Die übrigen Algorithmen setzen hinreichend konstante Uhrengeschwindigkeiten voraus, da bei ihnen eine lineare Approximation durchgeführt wird. Sie lösen aber dann das N-Prozeß-Problem.

Intern unterscheiden sich die Verfahren folgendermaßen:

- Das Verfahren mit der konvexen Hülle von Duda et al. (s. Abb. 3.2 auf S. 18) ermittelt stückweise lineare obere (und untere) Grenzen für die **Uhrendifferenzen zwischen**

je zwei Prozessen für das Gesamtintervall, in dem Ereignisse erzeugt wurden, wenn zwischen den Prozessen in der einen (und in der anderen) Richtung Nachrichten ausgetauscht worden sind.

Anschließend wird eine **über das Gesamtintervall lineare** obere (und untere) Grenze für die Uhrendifferenzen gebildet.

Mit einem **Ausgleichsalgorithmus** werden nun für die einzelnen Uhren lineare Korrekturfunktionen gebildet. Der Algorithmus garantiert, daß die Differenz aus den Korrekturfunktionen zu zwei Uhren innerhalb der oben ermittelten Grenzen für die Uhrendifferenzen liegen.

- Jézéquel gibt hierzu einen neuen Ausgleichsalgorithmus an.
- Das Verfahren von Hofmann und Hilgers vereinfacht sowohl den Algorithmus zur Ermittlung der Uhrendifferenzen, als auch den Ausgleichsalgorithmus.

Damit im nachfolgenden Kapitel die Grenzen des Verfahrens aufgezeigt werden können, sei hier das Verfahren von Hofmann und Hilgers gemäß [HH98] kurz zusammengefaßt:

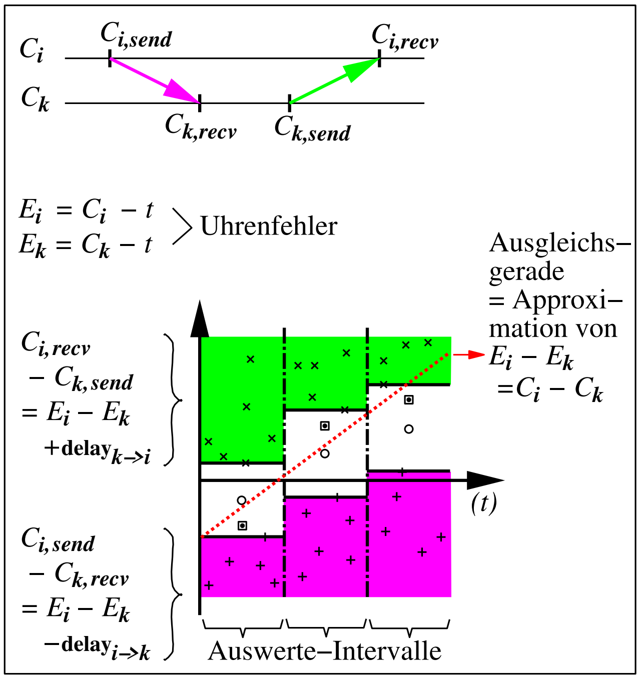


Abb. 7.2: Algorithmus von Hofmann und Hilgers

In Abb. 7.2 ist das Verfahren skizziert. Links oben sind die Bezeichnung der Zeitstempel von Sende- und Empfangsereignissen in den Prozessen i und k mit den Prozessoruhren C_i und C_k dargestellt. Das Diagramm in der Abbildung wird für jedes Prozeßpaar aufgestellt. Zuerst wird das zu untersuchende Gesamtintervall in gleichlange Auswerte-Intervalle aufgeteilt. In jedem Auswerte-Intervall werden die Werte

$$O^+_{ik} = C_i(e_i^{send}) - C_k(e_k^{recv}) \tag{7.2}$$

$$O^-_{ik} = C_i(e_i^{recv}) - C_k(e_k^{send}) \tag{7.3}$$

daher neben dem dort angegebenen Kriterium `while (change in any` $\underline{O_{ik}} + \overline{O_{ik}}$`)` das zusätzliche Kriterium `and (all elements` $E_{ik} \geq$ `0)`. Die resultierende Matrix kann nur verwendet werden, wenn alle Elemente der Fehlermatrix nicht negativ bleiben.

Prinzipiell ist Hofmanns Algorithmus zur Berechnung der Uhrendifferenzen Dudas konvexer Hülle vorzuziehen, da die Zeit für die Berechnung der Parallelstreifen proportional der Anzahl der Ereignisse ist, d.h. in der Ordnung $O(\sharp E)$. Auch ist Hofmann und Hilgers Algorithmus zur Berechnung des Ausgleichs wesentlich einfacher, da hierbei nur konstante Uhrenabweichungen und keine linearen Uhrenabweichungen (wie bei Duda und Jézéquel) ausgeglichen werden müssen. Dieser Ausgleich ist dafür für jedes Teilintervall durchzuführen. Da normalerweise die Rechenzeit für den Ausgleichsalgorithmus gegenüber der Rechenzeit für die Berechnung der Uhrendifferenzen vernachlässigbar ist, und da normalerweise die Anzahl der Teilintervalle gering ist, ist es kein Nachteil, daß der Ausgleich mehrfach durchzuführen ist.

Hofmanns Algorithmus benötigt aber zur Beherrschung konstanter, aber unterschiedlicher Uhrengeschwindigkeiten eine Unterteilung des Gesamtintervalls in Teilintervalle und am Ende eine Regressionsanalyse zur Berechnung der Uhrenkorrektur und Gangabweichung über das Gesamtintervall.

Das Beispiel in Abb. 7.4 zeigt, daß diese Regressionsanalyse im Extremfall zu falschen Ergebnissen kommen kann. Daher ist auch hier, wie schon oben erwähnt, eine nachträgliche Kontrolle des Ergebnisses unumgänglich. Rechnet man das Beispiel mit Hofmanns Algorithmus für das Gesamtintervall, dann erhält man für den ersten Prozeß eine Korrektur von 0 ms und für den zweiten Prozeß -0.5 ms, wobei die Laufzeit nicht kleiner als 1.5 ms sein darf, damit die Werte in der Fehlermatrix größer oder gleich Null bleiben. Teilt man dagegen das Intervall in drei gleiche Teile, dann erhält man für den ersten Prozeß die drei Korrekturwerte (0 ms, 0 ms, 0 ms) und für den zweiten Prozeß (1 ms, -2 ms, 1 ms). Die lineare Regression ergibt für beide Prozesse eine konstante Uhrenkorrektur, und zwar 0 ms für beide Prozesse. Für Laufzeitwerte innerhalb von [1.5, 2) werden Nachrichten von Prozeß 2 nach Prozeß 1 in der Mitte des Intervalls kausal falsch dargestellt.

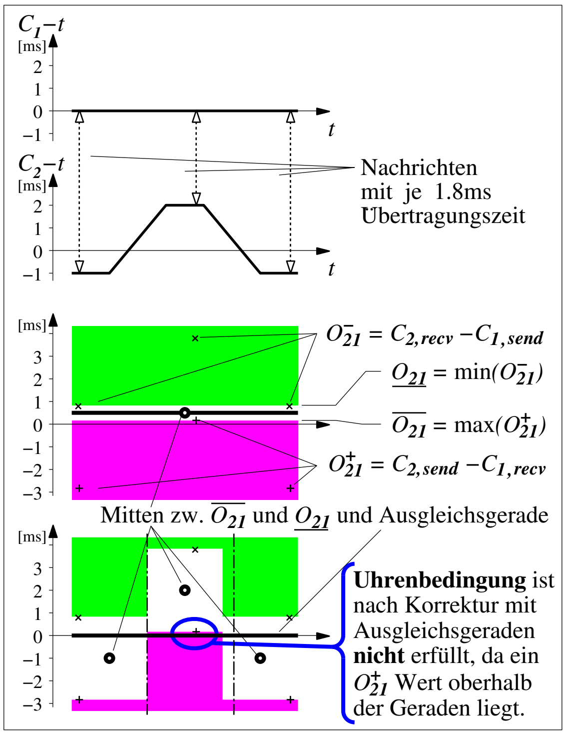


Abb. 7.4: Uhrenkorrektur bei beliebiger Gangabweichung

Beide Beispiele stellen prinzipielle Probleme dar, die auftreten können, sobald die Uhrengeschwindigkeiten der Prozessoruhren nicht hinreichend konstant sind. In der Praxis können Dudas und Hofmanns Algorithmen daher nicht eingesetzt werden, wenn sich die Uhrengeschwindigkeiten der Prozessoruhren im Meßzeitraum hinreichend stark ändern und dadurch bei mindestens einem Prozessorpaar die weiße Fläche in Abb. 3.2 und Abb. 3.3 auf S. 18f so krümmt, daß keine Gerade mehr hineingelegt werden kann. Dies kann z.B. schon bei Zeiträumen im

Sekundenbereich gegeben sein, wenn die Zeitmessung eine XNTP-Synchronisation beinhaltet.

Im folgenden Kapitel werde ich daher ein Verfahren vorstellen, das auf Hofmann und Hilgers Algorithmus aufbaut, aber weder konstanter Uhrengeschwindigkeiten bedarf, noch das soeben dargestellte Problem mit der Regressionsanalyse besitzt.

7.7 Neues Verfahren zur nachträglichen Synchronisation für variable Uhrengeschwindigkeiten

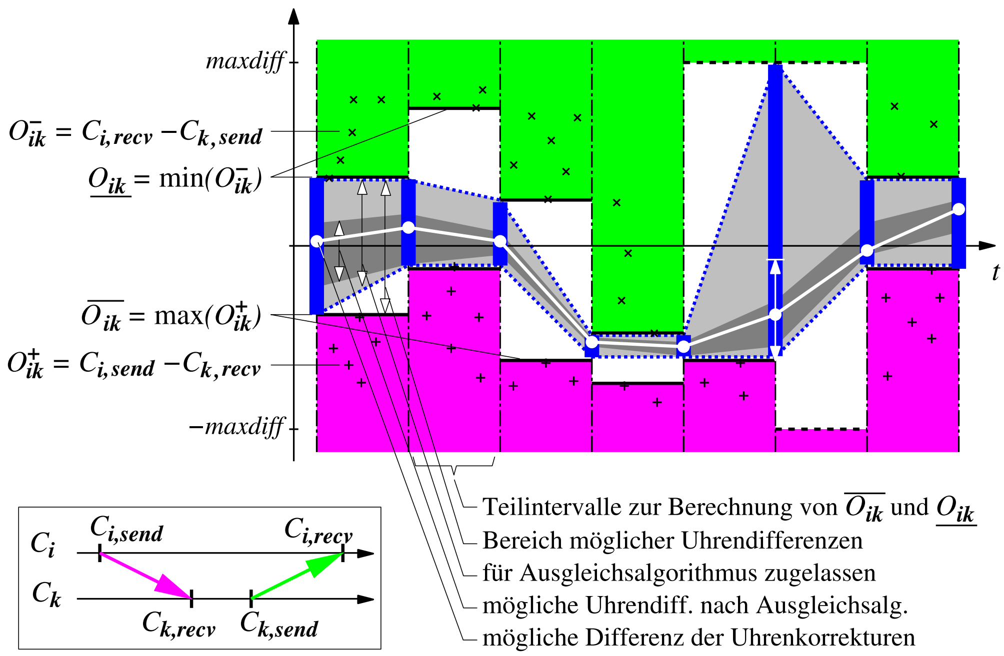


Abb. 7.5: Uhrenkorrektur für variable Uhrengeschwindigkeiten

In Abb. 7.5 auf S. 86 stelle ich ein neues Verfahren zur nachträglichen Synchronisation vor, das die im vorherigen Kapitel geschilderten Grenzen des Verfahrens von Hofmann und Hilgers teilweise überwindet. Im folgenden wird zuerst wie bei Hofmann eine Intervallaufteilung mit anschließender Anwendung des Min/Max-Algorithmus durchgeführt. Anschließend stelle ich die gegenüber Hofmann und Hilgers neue Ausgleichsvorbereitung vor und gebe dann einen Überblick über den restlichen Algorithmus. Nach einer Beschreibung der hierzu auch im Ausgleichsalgorithmus angebrachten Änderungen wird das vollständige Programmschema dieses neuen Verfahrens angegeben. Das Kapitel wird mit einer kurzen Bewertung abgeschlossen.

Intervallaufteilung und Min/Max-Algorithmus

Zuerst wird, wie bei Hofmann, das Gesamtintervall in gleichlange Abschnitte geteilt. Diese Aufteilung geschieht lokal für jeden Prozeß auf der Basis der Prozessoruhren. Anschließend wird für jedes Teilintervall und jedes Prozeßpaar (P_i, P_k) Hofmanns Min/Max-Algorithmus