(19) Deutsches Patent- und Markenamt

(10) **DE 10 2009 025 495 B4** 2015.08.06

(12) # Patentschrift

(21) Aktenzeichen: **10 2009 025 495.1**
(22) Anmeldetag: **19.06.2009**
(43) Offenlegungstag: **05.01.2011**
(45) Veröffentlichungstag der Patenterteilung: **06.08.2015**

(51) Int Cl.: ***G04G 7/00*** (2006.01)
H04L 7/04 (2006.01)

(73) Patentinhaber:
Universität zu Lübeck, 23562 Lübeck, DE

(74) Vertreter:
BOEHMERT & BOEHMERT Anwaltspartnerschaft mbB - Patentanwälte Rechtsanwälte, 28209 Bremen, DE

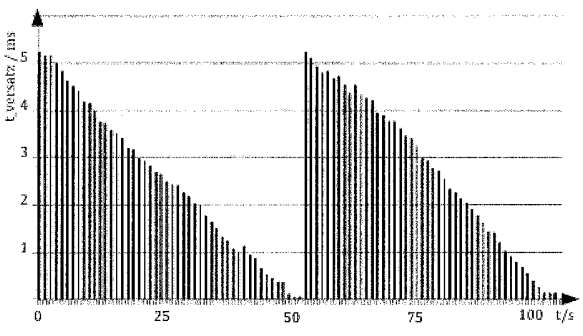
(72) Erfinder:
Carot, Alexander, 23552 Lübeck, DE; Werner, Christian, Prf. Dr., 38259 Salzgitter, DE

(56) Ermittelter Stand der Technik:

DE	**103 11 541**	**A1**
US	**2005 / 0 013 394**	**A1**
US	**5 276 659**	**A**
WO	**2006/ 011 867**	**A1**

(54) Bezeichnung: **Verfahren zur Synchronisation an unterschiedlichen Orten arbeitender Prozessoren über einen asynchronen Kommunikationskanal**

(57) Hauptanspruch: Verfahren zur Synchronisation zweier örtlich beabstandeter Prozessoren durch Übermitteln einer die Taktfrequenz des ersten Prozessors ausdrückenden Information an den zweiten Prozessor und Anpassen der Taktfrequenz des zweiten Prozessors an die Taktfrequenz des ersten Prozessors, gekennzeichnet durch
a) Versenden von mit je einem Zeitstempel versehenen ersten Datenblöcken gleicher Größe mit je K Nachrichten von dem ersten Prozessor an den zweiten Prozessor in gleichen Zeitabständen, wobei K eine festgelegte ganze Zahl ist,
b) Empfangen der ersten Datenblöcke des ersten Prozessors durch den zweiten Prozessor,
c) Erzeugen von zweiten Datenblöcken mit je K Nachrichten auf dem zweiten Prozessor, wobei die zweiten Datenblöcke die gleiche Größe wie die ersten Datenblöcke besitzen und der Prozess der Datenerzeugung auf dem ersten Prozessor simuliert wird,
d) Ermitteln der mittleren Laufzeit der ersten Datenblöcke vom ersten zum zweiten Prozessor aus dem Vergleich der Zeitstempel des ersten Prozessors mit der Uhr des zweiten Prozessors,
e) Messen des zeitlichen Versatzes zwischen dem Fertigstellen eines zweiten Datenblocks auf dem zweiten Prozessor und dem Eintreffen des nächsten ersten Datenblocks vom ersten Prozessor, wobei eine Messung des Versatzes nur gültig ist, wenn die Differenz der Laufzeit des eintreffenden Datenblocks zur vorab ermittelten mittleren Laufzeit einen vorgegebenen Betrag nicht übersteigt,
f) Replizieren des letzten gültigen Versatzmesswerts, wenn dessen Messung einen nicht gültigen Wert liefert,
g) Bilden der Differenz zwischen zwei Versatzmesswerte auf dem zweiten Prozessor, zwischen deren Messung N Datenblöcke empfangen worden sind, wobei N eine festgelegte ganze Zahl ist,
h) Anpassen der Taktfrequenz des zweiten Prozessors derart, dass die auf dem zweiten Prozessor gebildete Differenz minimiert wird und
i) fortlaufendes Wiederholen der Schritte b) bis h).



Beschreibung

[0001] Die Erfindung betrifft ein Verfahren zur Synchronisation technischer Prozessoren, wobei die Synchronisation über einen Kommunikationskanal erfolgen kann, der Informationen lediglich asynchron transportiert, d. h. in dem eine Datenübertragung ohne festen Zeittakt stattfindet.

[0002] Die zunehmende Verbreitung von Kommunikationsnetzen ermöglicht die Kopplung von Prozessen, die an unterschiedlichen Orten ablaufen. Häufig ist die enge zeitliche Kopplung der Prozessoren dabei wünschenswert. Synchronisation im Sinne der vorliegenden Erfindung bedeutet das Angleichen der Ablaufgeschwindigkeiten der Prozesse auf verschiedenen Prozessoren. Es sollen insbesondere die Taktraten auf verschiedenen Rechnern mit unterschiedlichen Architekturen angeglichen werden. Eine Uhrensynchronisation in realen verteilten Systemen – also das Einrichten einer Gleichzeitigkeit des Ablaufs von Prozessen auf verschiedenen Prozessoren – ist hingegen prinzipiell nicht exakt möglich (siehe Druckschrift: Leslie Lamport, Juli 1978, „Time, Clocks and the Ordering of Events in a Distributed System". Communications of the ACM 21 (7): 558–565).

[0003] WO 2006/011867 A1 betrifft ein Empfangsgerät das die Sampling-Rate bei der Wiedergabe anpasst und einen Pufferspeicher überwacht, um Abweichungen in der Rate zwischen einem Sendegerät und einem Empfangsgerät auszugleichen. Das Empfangsgerät beobachtet seinen Pufferspeicher und passt die Sampling-Rate bei der Wiedergabe an, um sich der Rate des Sendegeräts und des Empfangsgeräts anzupassen oder um Jitter zu kompensieren.

[0004] DE 103 11 541 A1 betrifft eine Vorrichtung zum Erfassen der Nullpunktabweichung zwischen zwei Uhren. Die Vorrichtung weist eine Hardwareimplementierung einer Bewertungseinrichtung für die Nullpunktabweichung der Uhr auf. Die Bewertungseinrichtung überwacht empfangene Pakete, die einem Datenstrom zugewiesen sind, und zieht einen Zeitstempel, der durch eine Quellenuhr erzeugt worden ist, aus jedem Paket. Eine Differenz d zwischen dem herausgezogenen Zeitstempel und der lokalen Zeit wird gegen einen Referenzwert d_ref verglichen, um festzustellen, ob das Paket früh oder spät empfangen wurde. Auf einem vorgeschriebenen Zeitplan wird der Grad des späten und frühen Empfangens von Paketen gegen einen Toleranzwert verglichen, um festzustellen, ob eine relative Nullpunktabweichung zwischen der Schrittsteuerung der Quellenuhr und der Schrittsteuerung der lokalen Uhr vorliegt. Die Erfassung der Nullpunktabweichung zwischen den beiden Uhren bietet Unterstützung für Service Level-Garantien beim Bereitstellen von Daten-Streaming-Diensten in paketgeschalteten Umgebungen.

[0005] US 2005/0013394 A1 betrifft die Taktsynchronisation in verteilten Systemen für Echtzeitanwendungen. Dabei werden gleichzeitig in jedem Knoten des Netzes die Taktabweichung und das Taktlesen korrigiert.

[0006] US 5,276,659 betrifft ein taktsynchrones System mit sicherer Taktsynchronisation in einem Netzwerk mit einer Master-Station und mehreren Slave-Stationen. Das System beobachtet ständig die Zeitdifferenz zwischen der Referenzzeit der Master-Station und der lokalen Zeit der Slave-Stationen. Wenn die Abweichung eine Schwelle überschreitet, dividiert das System die Zeitdifferenz durch einen bestimmenden Korrekturwert, um einen Zeitkorrekturkoeffizienten zu bestimmen korrigiert den Zeitunterschied der Slave-Station basierend auf diesem Korrekturwert.

[0007] Es ist nicht ohne weiteres möglich, an getrennten Orten zwei Taktsignale mit exakt gleichen Frequenzen zu generieren, da die hierzu verwendeten Bauteile, insbesondere Schwingquarze, Fertigungstoleranzen unterliegen und ihr Frequenzverhalten durch schwer kontrollierbare Umweltfaktoren (z. B. Temperatur) ändern. Aus diesem Grunde ist die Synchronisation von Prozessoren eine sehr anspruchsvolle technische Aufgabe.

[0008] Nach dem Stand der Technik kann diese Aufgabe durch drei unterschiedliche Verfahren gelöst werden:
Die erste besteht darin, ein gemeinsames Taktsignal aus einem gemeinsamen Taktgenerator als Referenz für die Prozessoren vorzusehen. Dies ist allerdings nur dann möglich, wenn dieses Taktsignal über ein geeignetes Medium an die Prozessoren übertragen werden kann (gemeinsame Taktleitung). Ist dies aufgrund von nicht exakt determiniertem Zeitverhalten im Übertragungskanal (Jitter) – wie z. B. bei der Kommunikation via Internet – nicht möglich, so kann man dieses Verfahren nicht anwenden. (Trischitta, P., und Varma, E.: „Jitter in Digital Transmission Systems". Boston: Artech House, 1989, ISBN 0-890-06248-X).

[0009] Die zweite Möglichkeit ist, das Taktsignal nicht direkt an beide Prozessoren zu senden, sondern es aus einem anderen gemeinsamen Referenzsignal, das beide Prozessoren empfangen können, so genau wie möglich zurück zu gewinnen. Ein bekanntes Beispiel für ein solches Referenzsignal ist das hochgenaue 77,5-kHz-Signal des Langwellensenders DCF77. Durch Frequenzvervielfachung oder Frequenzteilung kann man aus diesem Referenzsignal ein Taktsignal generieren, das für die zu synchronisierenden Prozessoren geeignet ist. Der Empfang eines solchen Referenzsignals ist allerdings aufwendig und auch für das Erzeugen des abgeleiteten Taktsignals müssen vergleichsweise aufwendige Schaltungen vorgehalten werden (Egan, W.:

„Frequency Synthesis by Phase-lock", John Wiley & Sons, 2000, ISBN 0-471-32104-4).

[0010] Die dritte Möglichkeit sieht vor, die Prozessoren mit Taktsignalen aus unabhängigen Taktgeneratoren zu speisen und durch geeignete technische Maßnahmen dafür zu sorgen, dass die jeweils erzeugten Frequenzen möglichst gleich sind. Beispiele hierfür sind hochgenaue, Quarze; durch einen Selektionsprozess kann sichergestellt werden, dass mehrere Quarze möglichst die gleichen physikalischen Eigenschaften aufweisen. Ein genauer Abgleich der Taktgeneratoren kann ihr Schwingverhalten weiter annähern. Auch Umwelteinflüsse, kann man durch geeignete technische Maßnahmen (z. B. Quarzöfen) begrenzen. Allerdings sind diese Maßnahmen sehr aufwendig. (Hewlett Packard: „Fundamentals of Quartz Oscillators, Application Note 200-2", 1997).

[0011] Gegenwärtig wird für praktische Anwendungen zumeist eine gemeinsame Taktleitung realisiert, wenn dies aufgrund der räumlichen Anordnung möglich ist. Weiter entfernt liegende Prozessoren können durch getrennte Taktgeneratoren gespeist werden, wobei eine Phase-Locked-Loop-Schaltung (PLL) dafür sorgt, dass die Frequenz auf Empfängerseite an die Frequenz des Senders angeglichen wird. Allerdings ist auch hier erforderlich, dass es einen allein dem Datenaustausch zwischen Sender und Empfänger vorbehaltenen, von anderen Prozessen ungestörten Kommunikationskanal gibt. Dieses Verfahren kommt häufig bei Rundfunkanwendungen zum Einsatz, um eine Empfängerschaltung exakt auf die Trägerfrequenz eines Senders zu justieren.

[0012] Alternativ lassen sich Prozessoren über Telekommunikationsnetze (z. B. ISDN) miteinander koppeln, die synchron arbeiten. Hier wird ein globales Taktsignal im Netz bestmöglich repliziert und als Basis für sämtliche Kommunikationsvorgänge genutzt.

[0013] Durch die zunehmende Verbreitung von asynchronen Kommunikationsnetzen (insbesondere dem Internet) besteht ein zunehmender Bedarf an Synchronisationslösungen, die über asynchronen Datenverkehr realisiert werden können. Die Besonderheit bei asynchronen Netzen ist, dass die Übertragungszeit für Informationen im Netz gewissen Schwankungen (sog. Jitter) unterliegt. Eine präzise und zuverlässige Synchronisation ist unter diesen Bedingungen offenbar besonders schwierig.

[0014] Aufgabe der Erfindung ist daher die Angabe eines Verfahrens zur Synchronisation von Prozessoren über asynchrone Kommunikationskanäle, das keine aufwendigen Schaltungen und teure Spezialbaugruppen erfordert.

[0015] Die Aufgabe wird gelöst durch ein Verfahren mit den Merkmalen des Hauptanspruchs. Die Unteransprüche geben vorteilhafte Ausgestaltungen an.

[0016] Die Erfindung setzt das Vorhandensein eines justierbaren Taktgenerators wenigstens auf der Empfängerseite voraus, der mit kostengünstigen elektronischen Standardschaltungen in an sich bekannter Weise realisiert werden kann. Es wird weiterhin vorausgesetzt, dass die zu synchronisierenden Maschinen technisch dafür ausgelegt sind, mit derselben Taktfrequenz arbeiten zu können, auch wenn die Architekturen möglicherweise nicht identisch sind.

[0017] Das erfindungsgemäße Verfahren beschreibt ein Messverfahren, mit dem die genaue Differenz der Taktfrequenzen zweier Prozessoren bestimmt werden kann. Mittels des Messergebnisses kann hiernach der justierbare Taktgenerator der Empfängerseite auf die Taktfrequenz des Senders eingestellt werden. Das erfindungsgemäße Verfahren erlaubt darüber hinaus die Taktfrequenzangleichung in extrem jitterbehafteten Netzwerken.

[0018] Die Grundzüge des Verfahrens werden zunächst für ein Netzwerk ohne Jitter erläutert:
Ein beliebiger Prozess auf einem ersten Computer (i. F. Sender) generiert in gleichen Zeitabständen Nachrichten. Diese Nachrichten werden zu gleich großen Datenblöcken zusammengefasst, die in gleichen Zeitabständen über einen Kommunikationskanal an einen zweiten Computer (i. F. Empfänger) geschickt werden. Zusätzlich werden die Datennblöcke vom Sender mit einem Zeitstempel versehen, beispielsweise bei Beginn des Erstellens eines Datenblocks oder unmittelbar vor dessen Absenden an den Empfänger.

[0019] Die Datenblöcke benötigen eine Laufzeit durch den Kommunikationskanal zwischen Sender und Empfänger. In einem Kommunikationskanal ohne Jitter ist diese konstant und kann genau gemessen werden.

[0020] Der Empfänger empfängt die Datenblöcke, und es findet eine erste Taktangleichung statt, indem der Empfänger anhand der Zeitstempel und der Anzahl der Nachrichten in den empfangenen Datenblöcken die Senderfrequenz bestimmt und seinen justierbaren Taktgenerator auf diese einstellt. Alternativ kann der Sender auch eine Angabe seiner eigenen Taktfrequenz an den Empfänger senden, die der Empfänger dann bei sich einstellt. Diese erste Anpassung der Taktfrequenz des Empfängers erfolgt also auf der Basis der vom Sender – implizit oder explizit – übermittelten Taktfrequenz des Senders und ist natürlich Stand der Technik.

[0021] Das der Erfindung zugrunde liegende Problem besteht darin, dass Sender und Empfänger

zwar beide behaupten, nunmehr identische Taktfrequenzen aufzuweisen, sich jedoch die physikalisch realen Taktfrequenzen – gemeinhin bei beiden Rechnern – von den nominellen unterscheiden. Entscheidend für eine genaue Synchronisation ist letzten Endes die Frage, wie der Empfänger die Taktfrequenz des Senders in Bezug auf seine eigene bewertet.

[0022] Erfindungsgemäß erzeugt daher der Empfänger genauso Nachrichten wie der Sender und stellt sie zu Datenblöcken derselben Größe zusammen. Hierbei ist der Empfänger auf seinen eigenen Zeittakt angewiesen. Auf den genauen Inhalt der Nachrichten kommt es dabei nicht an. Wesentlich ist vielmehr, dass die auf dem Empfänger erzeugten Datenblöcke genau dieselbe Anzahl an Nachrichten enthalten wie die vom Sender empfangenen. Da der Nachrichteninhalt der auf dem Empfänger erzeugten Daten nicht relevant ist, wird der Prozess der Datenerzeugung auf dem Sender nicht reproduziert, sondern nur „simuliert", denn dem Empfänger stehen normalerweise keine ausreichenden Eingabedaten zur Verfügung, z. B. Mikrofonaufzeichnungen oder dergleichen.

[0023] Ziel ist es, sich Informationen über die Arbeitsgeschwindigkeit des Senders „aus der Sicht" des Empfänger zu verschaffen. Wenn Sender und Empfänger keine identische Taktfrequenz aufweisen, unterscheiden sich die zeitlichen Abstände, in denen Nachrichten auf Sender und Empfänger erzeugt werden.

[0024] Erfindungsgemäß wird nun immer zu dem Zeitpunkt eine Zeitmessung auf dem Empfänger gestartet, an dem der Empfänger das Zusammenstellen bei sich generierter Nachrichten zu einem Datenblock abschließt (Referenzzeitpunkt), Die Zeitmessung wird gestoppt, sobald der nächste Datenblock vom Sender empfangen wird. Das Ergebnis der Zeitmessung wird im folgenden t_versatz genannt.

[0025] Kerngedanke der Erfindung ist, dass die Zeitdifferenz t_versatz eine allein auf der Empfängerseite bestimmbare Observable ist, die sich eignet, um die Taktfrequenzen schnell und genau anzugleichen. Dies ist überraschend selbst dann möglich, wenn ein Netzwerkjitter die Observable t_versatz zwischen den Einzelmessungen stark streuen lässt.

[0026] Die Messung der Observable t_versatz erfolgt fortlaufend für alle auf dem Empfänger erzeugten und vom Sender eintreffenden Datenblöcke. Dabei zeigt sich:

a) Wird die Zeitdifferenz t_versatz mit der Zeit größer, ist der Prozess auf dem Empfänger schneller als auf dem Sender (der Sender kommt mit dem Nachliefern der Datenblöcke nicht hinterher);

b) wird die Zeitdifferenz t_versatz mit der Zeit kleiner, ist der Prozess auf dem Empfänger langsamer als auf dem Sender (der Sender liefert Datenblöcke schneller nach, als der Empfänger deren Erzeugung simulieren kann);

c) bleibt die Zeitdifferenz t_versatz gleich, stimmen Empfänger und Sender in der Frequenz überein.

[0027] Die gemessene Zeitdifferenz t_versatz lässt sich unmittelbar zur Berechnung der Frequenzdifferenz heranziehen: Die Differenz zweier aufeinander folgender t_versatz Messwerte, t_diff, ergibt addiert zur Blocklänge des Empfängers (die Zeit, die der Empfänger zum Erstellen eines Datenblocks benötigt) sofort die Blocklänge des Senders. Wird dieser Wert durch die Anzahl der mit einem Block empfangenen Nachrichten geteilt, erhält man die Periodendauer des Senderprozesses, deren Kehrwert die Frequenz des Senderprozesses, f_sender, angibt.

f_sender = 1/((Blocklänge_empfänger + t_diff)/ Anzahl_nachrichten)

[0028] Offenbar gilt ebenso

f_sender = 1/((N·Blocklänge_empfänger + N·t_diff)/ N·Anzahl_nachrichten)

mit irgendeiner ganzen Zahl N. Es ist daher genauso möglich, den Wert N·t_diff als Differenz zweier t_versatz Werte zu bestimmen, zwischen deren Messung N Datenblöcke am Empfänger eingetroffen sind. Bei einer typischen Blocklänge um 5 ms und beispielsweise N = 200 reicht es aus, etwa einmal pro Sekunde die Senderfrequenz zu berechnen.

[0029] Die momentane Frequenz des Empfängers ist dem Empfänger selbst stets bekannt. Folglich ist hiernach auch die Frequenzdifferenz zwischen Empfänger und Sender bestimmt. Entsprechend der berechneten Frequenzdifferenz wird die Frequenz des empfängerseitigen Taktgenerators angepasst.

[0030] Die Observable t_versatz wird fortlaufend gemessen, so dass auch die Anpassung der Traktfrequenz auf der Empfängerseite fortlaufend erfolgen kann. Die Schrittweite der Anpassungsschritte nimmt dabei mit der Zeit ab, d. h. die Synchronisation wird immer besser. Der experimentell bestimmte Restfehler beträgt bei typischen Anwendungen weniger als 1 ppm, d. h. der verbleibende Taktunterschied beträgt weniger als einen millionsten Teil des Prozesstaktes.

[0031] Die Erfindung wird nachfolgend noch näher erläutert und auch im Zusammenhang mit Netzwerkjitter anhand der Figuren diskutiert. Dabei zeigt:

[0032] Fig. 1 Messwerte der Observable t_versatz bei der Kommunikation zweier nicht-synchroner Soundkarten in einem Local Area Network (LAN, praktisch kein Netzwerkjitter);

[0033] **Fig.** 2 Messwerte von t_versatz in der Konfiguration wie **Fig.** 1 während des Ablaufs des Synchronisationsverfahrens;

[0034] **Fig.** 3 Messwerte von t_versatz bei der Kommunikation zweier nicht-synchroner Soundkarten in einem Wide Area Network (WAN, starker Netzwerkjitter);

[0035] **Fig.** 4 Messwerte von t_versatz und ihre Replikate zur Jitter-Kompensation für die Konfiguration aus **Fig.** 3;

[0036] **Fig.** 5 Messwerte von t_versatz und ihre Replikate wie in **Fig.** 4 während des Ablaufs des Synchronisationsverfahrens.

[0037] In **Fig.** 1 sind exemplarisch Messwerte der Observable t_versatz gezeigt, die beim Datenaustausch zweier nicht-synchroner Soundkarten über ein – praktisch jitterfreies – lokales Netzwerk (LAN) entstehen. Die Observable nimmt dabei kontinuierlich ab, weil der Sender schneller Datenblöcke erzeugt und nachliefert, als der Empfänger in seinem Simulationsprozess. Erreicht t_versatz den Wert Null, so ist der Empfänger gerade mit dem Erstellen eines Datenblocks fertig als auch ein Datenblock vom Sender eintrifft. Der nachfolgend erzeugte Datenblock auf der Empfängerseite wird erst kurz nachdem der nächste vom Sender bereits eingetroffen ist, fertig, so dass die Messung von t_versatz fast eine ganze Blocklänge dauert, bis der übernächste Datenblock des Senders ankommt. Hieraus erklärt sich das Sägezahnmuster in den Daten. Offenbar würde bei fortgesetztem Datentransfer zwischen den beiden Soundkarten der Bufferspeicher des Empfängers irgendwann überlaufen bzw. Datenblöcke des Senders gingen verloren.

[0038] Das erfindungsgemäße Synchronisationsverfahren führt durch Anpassung der empfängerseitigen Taktfrequenz dazu, dass t_versatz mit der Zeit einen konstanten Wert annimmt, wie aus **Fig.** 2 zu ersehen ist. Zwar bleiben noch geringe Taktunterschiede erkennbar, aber diese bleiben durch die fortlaufende Anwendung der Synchronisation auf sehr kleine Werte beschränkt. Der Mittelwert von t_versatz in **Fig.** 2 entspricht der Laufzeit der Datenblöcke im Netz.

[0039] In Weitverkehrsnetzwerken (Wide Area Network, WAN) ist der Netzwerkjitter ein inhärentes Problem. Aufgrund des Netzwerkjitters können die eintreffenden Pakete um eine unbestimmte Zeitspanne verzögert sein und dadurch jeden einzelnen Messwert t_versatz verfälschen oder – abhängig vom Ausmaß des Fehlers – nutzlos machen. **Fig.** 3 zeigt die Observable t_versatz, bei der Kommunikation der Soundkarten über das Internet. Es ist hier aber anzumerken, dass man das bereits erwähnte Sägezahnmuster auch in **Fig.** 3 noch grob erkennen kann.

[0040] Um das beschriebene Synchronisationsverfahren trotz Netzwerkjitter einsetzen zu können, dürfen zu spät eintreffende Datenblöcke nicht mit in die Messung von t_versatz aufgenommen werden. Aus diesem Grunde werden die vom Sender generierten und an den Empfänger übermittelten Zeitstempel auf der Empfängerseite zusätzlich mit der Zeit des Empfängers verglichen. Zwar sind die internen Uhren von Sender und Empfänger selbst nicht synchronisiert, aber der Zeitvergleich von Datenblock zu Datenblock gibt dennoch Aufschluss darüber, welche Verzögerung vom Netzwerkjitter herrührt.

[0041] Die Differenz zwischen der Empfangszeit (gemessen am Empfänger) und der Sendezeit (gemessen am Sender und festgehalten im übertragenen Zeitstempel) ist eine im WAN möglicherweise stark streuende Messgröße, die nachfolgend Paketlaufzeit genannt werden soll und deren Mittelwert über einen längeren Zeitraum (z. B. 1 Sekunde) leicht bestimmt werden kann. Dabei ist anzumerken, dass die Messung der Paketlaufzeit natürlich auch durch die fehlende Taktsynchronisation von Sender und Empfänger fehlerbehaftet ist. Sie kann zudem durch unkontrollierbare Einflüsse auf die Netzwerkverbindung (etwa die Last) kurzfristig stark schwanken.

[0042] Erfindungsgemäß wird deshalb der Mittelwert der Paketlaufzeit wiederkehrend bestimmt, beispielsweise einmal pro Sekunde. Sobald dieser Mittelwert bestimmt ist, werden nur noch Datenblöcke zur Messung von t_versatz benutzt, deren tatsächliche Laufzeit im asynchronen Netz dem Mittelwert der Paketlaufzeit im Wesentlichen entspricht. Alle übrigen Datenblöcke werden für die Messung von t_versatz als ungültig verworfen.

[0043] Anstelle der wahren Werte von t_versatz, die infolge der Ungültigkeit von empfangenen Datenblöcken nicht fortlaufend neu bestimmt werden können, wird der letzte gültige Wert für t_versatz schlicht wiederholt (repliziert) bis das nächste gültige Datenpaket eintrifft. Durch dieses Vorgehen ergibt sich die Stufenfunktion aus **Fig.** 4 anstelle der stark streuenden Werte aus **Fig.** 3, wobei Lücken in den gemessenen Daten durch Replikate aufgefüllt werden. Die Verwendung der Replikate für den zuvor beschriebenen, kontinuierlich laufenden Synchronisationsvorgang (fortlaufendes Anpassen der empfängerseitigen Taktfrequenz) ist unkritisch und führt zu einem befriedigenden Ergebnis, wie **Fig.** 5 verdeutlicht. Zwar kommt es nur noch dann zu einer neuen Berechnung der Senderfrequenz und folglich zur Angleichung der Empfängerfrequenz, wenn ein neuer gültiger Datenblock am Empfänger eintrifft, doch ist dies häufig genug der Fall, um die gewünschte Taktsynchronisation zu erreichen. Die durch Selektion vom Einfluss des Jitters bereinigten Messwerte t_versatz nehmen schnell einen konstanten Wert an.

[0044] Das beschriebene Verfahren kann beispielsweise verwendet werden, um die Aufnahme- und Wiedergabeprozesse zweier Soundkarten an unterschiedlichen Standorten zu synchronisieren, wobei der Datenaustausch über das Internet erfolgt.

[0045] Bei herkömmlichen Soundkarten arbeitet man mit einer typischen Samplefrequenz von 48 kHz. Da die Taktgeneratoren herkömmlicher Soundkarten jedoch nicht exakt abgestimmt sind und auch einer gewissen Temperaturabhängigkeit unterliegen, hat man in der Praxis das Problem, dass eine aufnehmende Soundkarte an Standort A in einem festen Zeitintervall geringfügig mehr bzw. weniger Abtastwerte generiert als die wiedergebende Soundkarte an Standort B abspielt. Je nach Soundkarte ergeben sich Abweichungen von bis zu 100 ppm, so dass bei 48 kHz Samplefrequenz innerhalb einer Minute eine Differenz von 60 s × 48.000 1/s × 100/1.000.000 = 288 Samples entsteht. Neben einer geringfügig veränderten Tonhöhe macht sich in der Praxis vor allem negativ bemerkbar, dass in den empfängerseitigen Puffer durch dieses Ungleichgewicht von Zeit zu Zeit ein Unter- bzw. Überlauf stattfindet. Nach dem gegenwärtigen Stand der Technik wird dies (z. B. bei Voice-over-IP-Anwendungen) entweder in Kauf genommen oder durch Resampling bzw. gezieltes Weglassen oder Verdoppeln einzelner Samples softwareseitig ausgeglichen. Diese Ansätze führen jedoch zu einer Verfälschung des Signals.

[0046] Das erfindungsgemäße Verfahren ermöglicht nun die unverfälschte Wiedergabe des Signals. Jedes vom Sender gesendete Datenpaket repräsentiert eine feste Anzahl von Samples, z. B. Blöcke mit jeweils 128 Samples. Der Sender schickt das Datenpaket sofort ab, nachdem alle Samples für einen kompletten Block vorliegen. Der Empfänger empfängt die Blöcke und benutzt die übertragenen Daten, um den Prozess der Senderseite nachzuvollziehen; im Beispiel ist die Wiedergabe der Samples über die lokale Soundkarte eine adäquate Form der Simulation. Der Wiedergabeprozess läuft zunächst typisch schneller oder langsamer ab als der Aufnahmeprozess des Senders.

[0047] Zu jedem Referenzzeitpunkt der empfängerseitigen Soundkarte wird eine Zeitmessung gestartet. Diese Messung wird gestoppt, sobald ein Datenpaket über das Netzwerk eintrifft. Allerdings sind die so durchgeführten Messungen nicht immer brauchbar: Netzwerkjitter sorgt in der Regel dafür, dass die Abstände zwischen den Empfangszeitpunkten der Datenpakete nicht konstant sind. Um dadurch entstehende Messfehler zu vermeiden, werden die empfangenen Datenpakete einem Selektionsprozess unterzogen. Verzögerte Datenpakete führen dann zu ungültigen Zeitmessungen, die nicht weiter verwendet werden.

[0048] Laufen Sender und Empfängerprozess mit derselben Geschwindigkeit, so ist das gemessene Zeitintervall t_versatz im Mittel konstant. Anderenfalls wird die Taktfrequenz der Empfängersoundkarte verringert, sofern t_versatz zunimmt (Soundkarte des Empfängers arbeitet schneller als die des Senders), und erhöht, sofern t_versatz abnimmt (Soundkarte des Empfängers arbeitet langsamer als die des Senders). Dieser Anpassungsvorgang wird fortlaufend wiederholt.

[0049] Durch die dargestellte indirekte Form der Zeitmessung wird erreicht, dass empfängerseitig keine absolut exakte Uhr benötigt wird, um hieraus eine absolute Frequenz bzw. einen absoluten Frequenzfehler für die Justage des Taktgenerators abzuleiten. Stattdessen wird lediglich die Veränderung des Zeitintervalls betrachtet, das zwischen dem Referenzpunkt und dem Eintreffen des nächsten Pakets liegt. Hierfür ist keine absolute Zeitmessung notwendig, da lediglich die Veränderung einer Zeitspanne betrachtet wird – und zwar in Bezug auf eine beliebige Uhr, an die keine besonderen Genauigkeitsanforderungen gestellt werden.

[0050] Die Erfindung ist jedoch nicht auf Anwendungen in Verbindung mit Soundkarten beschränkt, sondern kann zur Synchronisation beliebiger technischer Prozesse verwendet werden, die im Empfänger nachvollzogen, d. h. simuliert werden können. Weitere Beispiele sind die Synchronisation von Produktionsprozessen, die durch Roboter ausgeführt werden, oder die Synchronisation von Software-Prozessen, die auf getrennten Rechnern ablaufen.

Patentansprüche

1. Verfahren zur Synchronisation zweier örtlich beabstandeter Prozessoren durch Übermitteln einer die Taktfrequenz des ersten Prozessors ausdrückenden Information an den zweiten Prozessor und Anpassen der Taktfrequenz des zweiten Prozessors an die Taktfrequenz des ersten Prozessors, gekennzeichnet durch

a) Versenden von mit je einem Zeitstempel versehenen ersten Datenblöcken gleicher Größe mit je K Nachrichten von dem ersten Prozessor an den zweiten Prozessor in gleichen Zeitabständen, wobei K eine festgelegte ganze Zahl ist,

b) Empfangen der ersten Datenblöcke des ersten Prozessors durch den zweiten Prozessor,

c) Erzeugen von zweiten Datenblöcken mit je K Nachrichten auf dem zweiten Prozessor, wobei die zweiten Datenblöcke die gleiche Größe wie die ersten Datenblöcke besitzen und der Prozess der Datenerzeugung auf dem ersten Prozessor simuliert wird,

d) Ermitteln der mittleren Laufzeit der ersten Datenblöcke vom ersten zum zweiten Prozessor aus dem Vergleich der Zeitstempel des ersten Prozessors mit der Uhr des zweiten Prozessors,

e) Messen des zeitlichen Versatzes zwischen dem Fertigstellen eines zweiten Datenblocks auf dem zweiten Prozessor und dem Eintreffen des nächsten ersten Datenblocks vom ersten Prozessor, wobei eine Messung des Versatzes nur gültig ist, wenn die Differenz der Laufzeit des eintreffenden Datenblocks zur vorab ermittelten mittleren Laufzeit einen vorgegebenen Betrag nicht übersteigt,
f) Replizieren des letzten gültigen Versatzmesswerts, wenn dessen Messung einen nicht gültigen Wert liefert,
g) Bilden der Differenz zwischen zwei Versatzmesswerte auf dem zweiten Prozessor, zwischen deren Messung N Datenblöcke empfangen worden sind, wobei N eine festgelegte ganze Zahl ist,
h) Anpassen der Taktfrequenz des zweiten Prozessors derart, dass die auf dem zweiten Prozessor gebildete Differenz minimiert wird und
i) fortlaufendes Wiederholen der Schritte b) bis h).

2. Verfahren nach Anspruch 1, **dadurch gekennzeichnet,** dass das Anpassen der Taktfrequenz des zweiten Prozessors durch Angleichen an die zuvor aus der Differenz der Versatzwerte berechnete Taktfrequenz des ersten Prozessors erfolgt.

3. Verfahren nach einem der vorangehenden Ansprüche, **dadurch gekennzeichnet,** dass das Ermitteln der mittleren Laufzeit der ersten Datenblöcke durch Mittelwertbildung der Laufzeiten einzelner Datenblöcke über etwa 1 Sekunde erfolgt.

4. Verfahren nach einem der vorangehenden Ansprüche, **dadurch gekennzeichnet,** dass die ganze Zahl N derart vorgegeben wird, dass das Produkt N·Blocklänge etwa 1 Sekunde beträgt.

Es folgen 4 Seiten Zeichnungen

Anhängende Zeichnungen

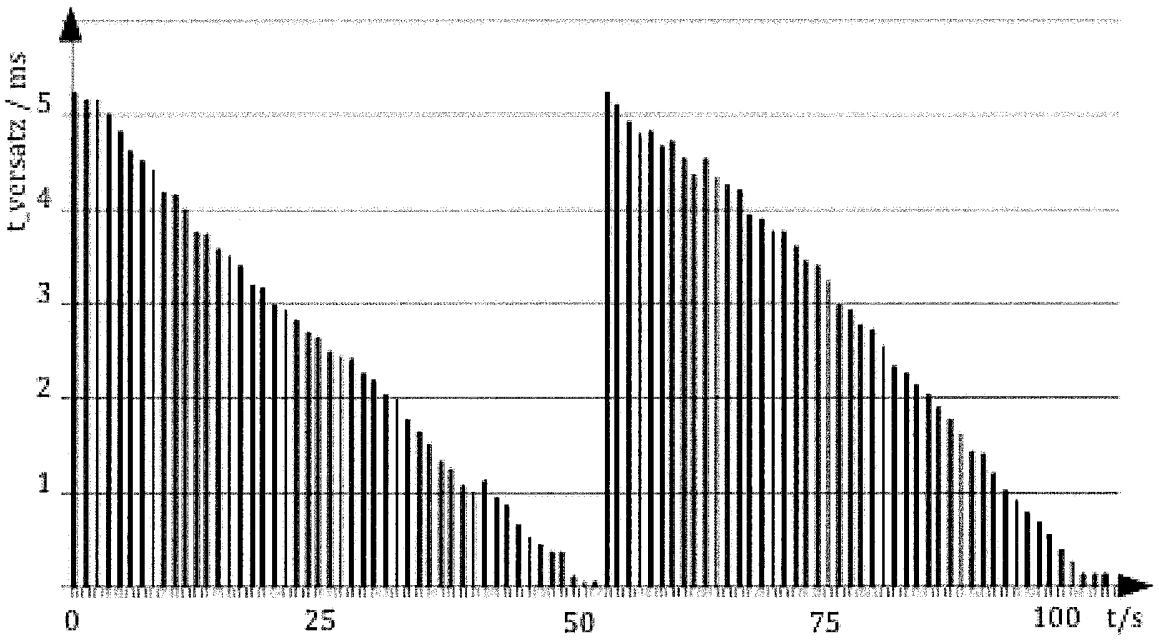


Fig. 1

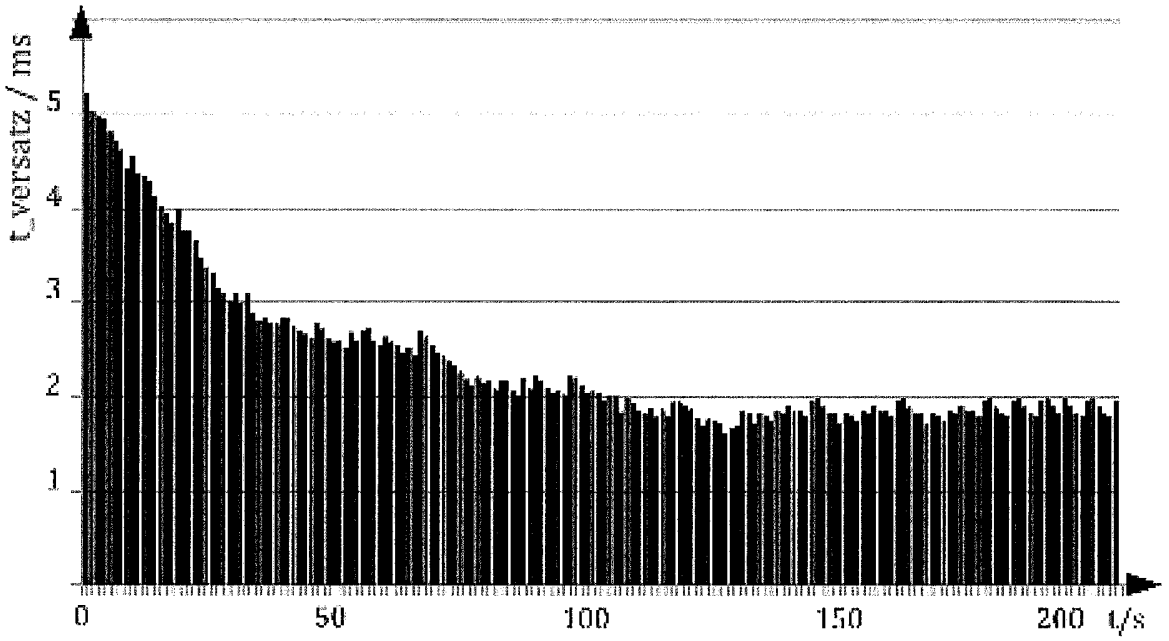

t_versatz / ms
5
4
3
2
1
0
50
100
150
200
t/s

Fig. 2

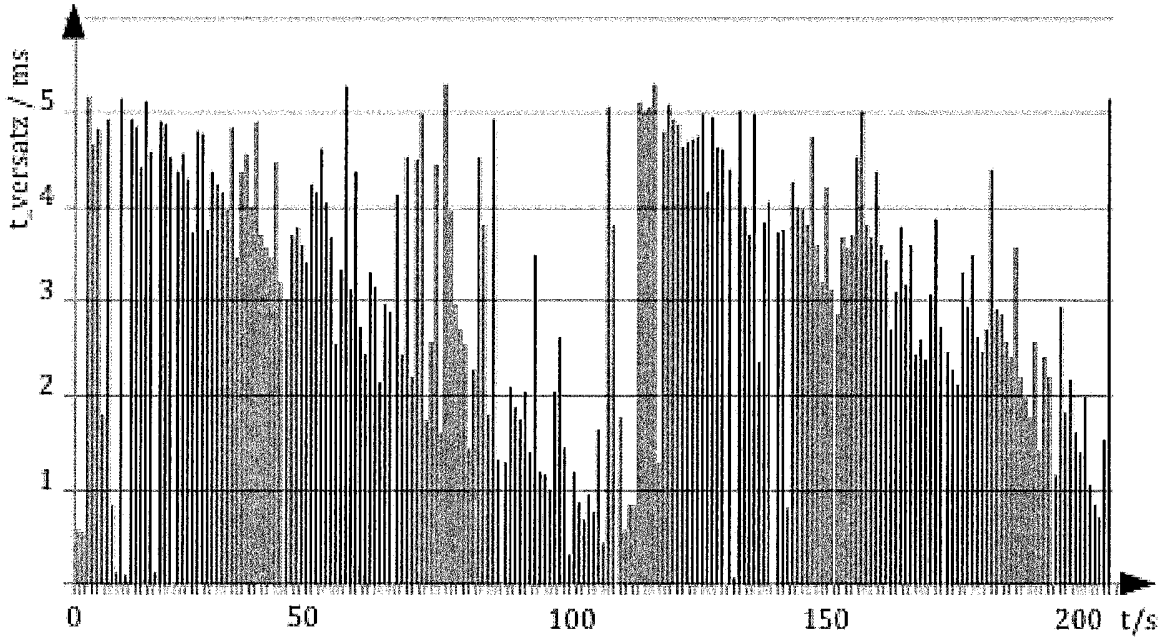

t_versatz / ms
5
4
3
2
1
0
50
100
150
200
t/s

Fig. 3

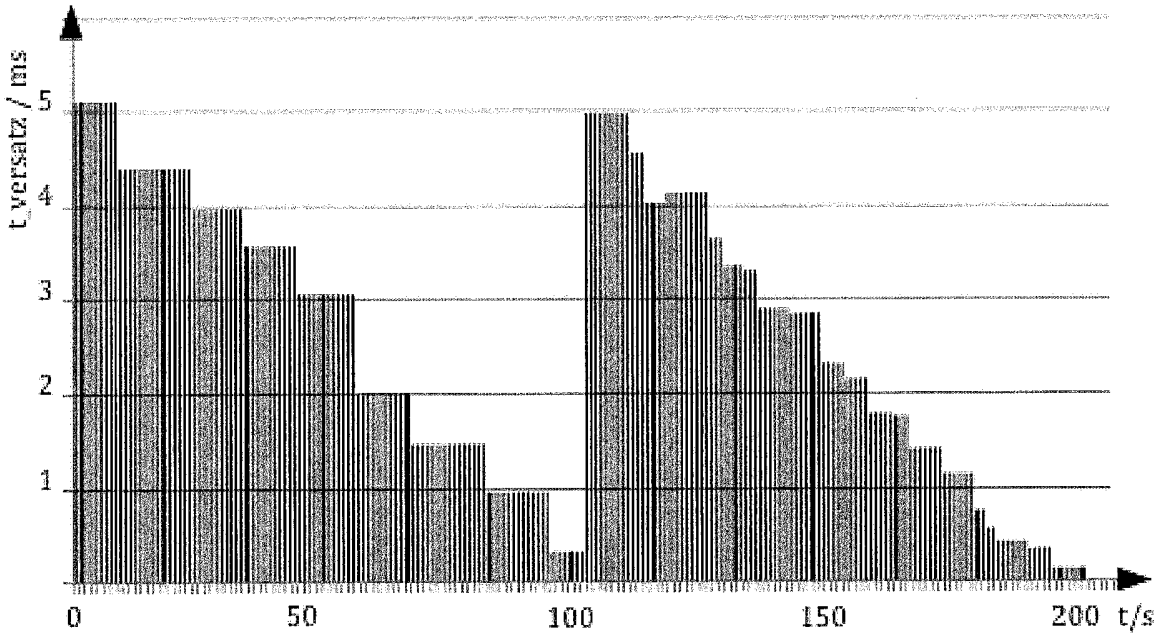


Fig. 4

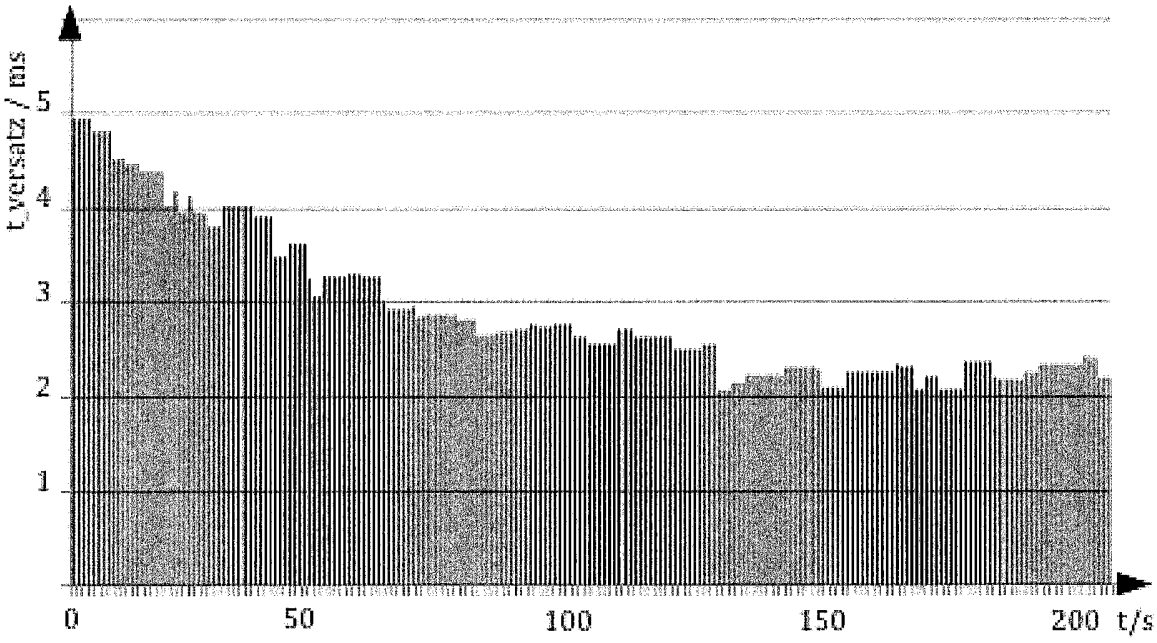


Fig. 5