



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Ausarbeitung - Anwendungen 1 WS2011-2012

Kai Müller

Zeitsynchronisierung in verteilten Systemen
Ein Peer-to-Peer-basierter Ansatz

Kai Müller

Thema der Ausarbeitung - Anwendungen 1

WS2011-2012

Zeitsynchronisierung in verteilten Systemen

Stichworte

Echtzeit Ethernet, TTEthernet, Bussysteme, Automotive Anwendungen

Kurzzusammenfassung

Der Einsatz von echtzeitfähigen Netzwerken nimmt mehr und mehr zu. Unterschiedliche Industriezweige wie dem Automotive Bereich oder der Medizintechnik erwägen diese Technologie in vielen Bereiche einzusetzen, da die ursprünglichen Bussysteme bereits an ihre Grenzen stossen oder dies in naher Zukunft zu erwarten ist. Zudem können erweiterte Funktionen umgesetzt werden, für die die zur Zeit bereitgestellten Leistungen nicht ausreichen. Bisherige Synchronisationsalgorithmen verfolgen ein standardisiertes Master-Slave-Prinzip und weisen so in Hinblick auf Single Point of Failure und Unterstützung von Energiesparfunktionen erhöhten Planungsbedarf und Kostenaufwand auf. Ein neuer auf Peer-to-Peer basierender Ansatz könnte diese Probleme durch seine Architektur auffangen und dadurch Komplexität reduzieren und Wartungs- und Produktionskosten einsparen.

Title of the paper

Zeitsynchronisierung in verteilten Systemen

Keywords

Real-time Ethernet, TTEthernet, Bussystems, Automotive Applications

Abstract

The amount of real-time capable networks increases fast. Because bus-systems used today come to their limit, several lines of industry, for example automotive or medical engineering, are planning to adopt this technology. Besides additional functions with heavier demands could be build in. Today's synchronisation algorithms uses standard master-slave-concepts, and therefore have high demands in planing with the use of cost excessive solutions for Single point of Failure strategies and energy saving. A new P2P-based architecture could solve these problems with the use of its concept and so, reduces the extra costs in maintenance and production.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Echtzeitfähige Netze im Überblick	2
1.2	Zielsetzung	3
2	Synchronisation im Netzwerk	4
2.1	Offset- und Rate-Correction	4
2.2	Aufbau einer Synchronisation	5
2.2.1	Ereignisermittlung	5
2.2.2	Ansatzselektion	6
2.2.3	Korrekturverfahren	6
2.3	Peer-to-Peer-Synchronisation	7
3	Modellierung des Ansatzes	8
3.1	Abbildung eines Knotens	8
3.2	P2P Übertragungsmodell	8
3.3	Simulationsplattform	9
4	Fazit	10
4.1	Ausblick auf weiteres Vorgehen	10
4.2	Risikoabschätzung	11
	Abbildungsverzeichnis	12
	Bibliography	13

1 Einleitung

Der Einsatz von echtzeitfähigen Netzwerken gelangt in unterschiedlichsten Bereichen mehr und mehr in den Vordergrund. Während zunächst die Umsetzung solcher Technologien auf die Bereiche Flugzeugbau und Automatisierungstechnik spezialisiert waren, so wurde in den letzten Jahren zunehmend mehr und mehr aktiv untersucht, echtzeitfähige Netzwerke auch in weiterführenden Bereichen einzusetzen (vgl. [1]). Als Beispiel seien hier die Automobilindustrie oder auch die Medizintechnik genannt. Somit kann in Zukunft mit einem weit gefächerten Einsatz von echtzeitfähigen Netzwerken gerechnet werden. Hierbei gelangt die Leistungsfähigkeit der Netzwerke in Sachen Bandbreite und Zeitverhalten (geringe Latenzen und Jitter) sowie deren Zuverlässigkeit in der Datenübertragung in den Fokus (vgl. [2]). Diese Faktoren können in den jeweiligen Bereichen unterschiedlich genutzt werden, welche im Folgenden vorgestellt werden sollen:

Produktions und Automatisierungstechnik

In der Produktions und Automatisierungstechnik werden die neuen Technologien bereits in Form von EtherCAT (vgl. [3]), Profinet (vgl. [4]) oder auch Powerlink (vgl. [5]) erfolgreich eingesetzt. Sie haben den CAN-Bus oder das serielle Interface ersetzt und setzen sich aufgrund ihrer Bandbreite und Zuverlässigkeit als neuer Standard durch. Ein weiterer Vorteil liegt in der hohen Skalierbarkeit. So ist eine unkomplizierte Vernetzung von vielen (>100) Einheiten realistisch. Motoren und Sensoren können somit direkt als Knoten ins Netzwerk integriert werden und müssen nicht erst über eine SPS-Steuerung kontrolliert und über mehrere SPS-Systeme hinweg vernetzt werden. Die Kontrolle der Anlage kann daher wahlweise zentral oder verteilt erfolgen.

Automotive

Ein weiterer signifikanter Bereich liegt in der Mobilindustrie, welche eine Umstrukturierung ihrer echtzeitfähigen Kommunikation untersucht. Bisher hat sich noch kein echtzeitfähiges Ethernet Protokoll durchsetzen können (vgl. [6]). Die letzte Errungenschaft, der FlexRay-Bus (vgl. [7]), ist aufgrund seines speziellen Konzepts teuer in der Entwicklung und kompliziert in der Planung und Wartung, da er keinem bekannten Prinzip gleicht. Durch neuste Entwicklungen und verbreiteten Anwendung von kamerabasierten Fahrerassistenzsystemen wird ein stetig wachsender Bedarf an die Bandbreite solch eines Netzwerkes gestellt, welche jetzige Technologien an ihre Grenzen stoßen lassen (vgl. [8]). Automobilhersteller (BMW) begannen daher

erste Untersuchungen bezüglich Ethernet basierte Echtzeitkommunikation im Automobil, jedoch ohne Ergebnisse offen zu legen. Ein zusätzlicher Vorteil eines solchen Systems läge in seiner Homogenität. Viele über die Jahre gewachsene Subnetze unterschiedlichster Protokolle liessen sich so unter einer Technologie vereinen um somit die Komplexität, sowie Kosten in Planung und Produktion zu reduzieren.

Medizintechnik

Es ist ebenso in naher Zukunft zu erwarten, dass in der Medizintechnik echtzeitfähige Netzwerke eingesetzt werden. Zur Zeit wird noch von Firmen wie Philips untersucht, welche bisherigen Technologien durch neue Protokolle ersetzt werden können (vgl. [9]). Erste Ansätze der Untersuchungen existieren im Bereich der Radiografie und Kernspintomografie, welche mit Hilfe von echtzeitkritischem Verkehr die Dosierung der eingesetzten Strahlung beeinflussen. Bislang werden proprietäre Bussysteme eingesetzt um Strahlendosis und Sensoreigenschaften auf die Bewegungen der Mechanik abzustimmen. Das Ziel ist diese durch Standards zu ersetzen, sowie multiple Technologien unter einem einheitlichen Protokoll zusammenzufassen.

1.1 Echtzeitfähige Netze im Überblick

Um die Echtzeitfähigkeit eines Netzwerks zu gewährleisten, existieren unterschiedliche Ansätze. Man unterscheidet zwischen drei Verfahren:

Zum einen werden bandbreitenreduzierende Systeme eingesetzt. Diese finden sich z.B. in aktuellen Flugzeugen (A380) als Avionics Full Duplex Switched Ethernet (AFDX [10]) wieder. Innerhalb dieser Netzwerke wird die Echtzeitfähigkeit dahingehend gewährleistet, dass keine Nachrichten verloren gehen können. Dennoch ist es nicht ausgeschlossen, dass mehrere Teilnehmer zur selben Zeit einen Sendevorgang einleiten und es zu Kollisionen von Nachrichten an den Switches kommt. Diese Kollisionen werden durch Switch-interne Buffer abgefangen. Eine Nachricht kann somit in ihrer Übertragungszeit schwanken, wird jedoch garantiert zugestellt. Um zu verhindern, dass es zu einem Bufferüberlauf kommen kann und somit Informationen im Netzwerk verloren gehen, ist die Sendebandbreite eines jeden Teilnehmers beschränkt (Bucketsystem [11]). Hat ein Teilnehmer seine Bandbreitenreserven erschöpft, so ist er gezwungen, seine Senderate einzuschränken. Dies wird von den Switchen kontrolliert und ggf. beschränkt.

Eine weitere Ansatz sind Token-basierte Systeme. Dieses Token stellt den exklusiven Zugriff auf das Netzwerk da und bildet somit zur gleichen Zeit den Synchronisationsmechanismus der Teilnehmer untereinander. Auf diese Weise wird ein Netzwerk, topologieunabhängig auf einen logischen Bus abgebildet, was parallele Nachrichtenverarbeitung innerhalb dieses Netzwerks ausschliesst. Das Problem von Token-basierten Verfahren ist die Verlosterkennung des Tokens.

Dies erfordert spezielle Algorithmen, welche zyklisch ausgeführt werden müssen, und somit einen Overhead an Verarbeitungsaufwand für jeden Knoten bilden.

Die dritte Variante, eine echtzeitfähige Kommunikation zu gewährleisten, ist der Einsatz eines TDMA (Time Division Multiple Access) Verfahrens (z.B. TTEthernet [12], TTP [13]). Hierbei werden dedizierte Zeitschlitze für jede Nachricht definiert, in denen der Link zwischen Sender und Empfänger frei gehalten wird. Die zu übertragende Nachricht kann so mit geringem Jitter und einer Verzögerung, die der Nachrichtenlaufzeit entspricht, übertragen werden. Auch bei diesem Verfahren wird die Einhaltung der Zeitschlitze von den Switches überwacht. Um über alle Teilnehmer des Netzwerkes hinweg eine einheitliche Zeit einhalten zu können, müssen diese die gleiche Zeitbasis aufweisen. Hierzu sind entsprechende Zeitsynchronisationsverfahren notwendig, auf welche in dieser Arbeit eingegangen wird.

1.2 Zielsetzung

Zur Zeit existieren keine Peer-to-Peer basierenden Ansätze für Synchronisationsalgorithmen. Jeweils ein zentraler Zeitgeber ist für ein Netzwerk bzw. Subnetzwerk zuständig. Ausfallsicherheit wird in neueren Verfahren dadurch gewährleistet, indem diese zentralen Zeitgeber redundant aufgesetzt werden. Ältere Verfahren akzeptieren keine mehrfachen Zeitgeber in dem selben Netzwerk und sind daher ausschliesslich durch Spiegelung des gesamten Netzwerkes und seiner Teilnehmer (Bildung von sog. Schattennetzwerken vgl. [14]) in der Lage Ausfallsicherheit durch Redundanz abzubilden. Dieser Ansatz ist zwar frei skalierbar, dennoch mit hohem Zusatzaufwand in der Herstellung und der Unterhaltung (Energieverbrauch) verbunden. In geschlossenen Systemen, wie z.B. einem Automobil oder Flugzeugs, sind dies ausschlaggebenden Faktoren.

Daher wäre es wünschenswert, einen Ansatz zu finden, welcher keinen dedizierten Zeitgeber mehr fordert, sondern das Netzwerk mit der Gesamtheit aller Teilnehmer diese gemeinsame Zeitbasis bildet. Fällt ein Endsystem aus, so ist die Synchronisation aller übrigen Endsysteme nicht gefährdet. Somit wird der kritische Single Point of Failure unterbunden. Zudem ist es möglich, beliebige Endgeräte des Netzwerkes ohne vorherige Ankündigung oder einem Abmeldevorgang schlafen zu legen um sie zu einem späteren Zeitpunkt wieder dynamisch in das Netz zu integrieren. Dies trägt wiederum zu einer Steigerung der Energieeffizienz des Gesamtsystems bei. Diese Vorteile vereinfachen die Planungs- und Entwicklungsphase solcher Echtzeitnetzwerke erheblich, da keine unterschiedlichen Rollen (Zentraler Zeitgeber, Backupknoten, etc.) für einzelne Knoten eines Netzwerkes definiert werden müssen.

2 Synchronisation im Netzwerk

Ein Synchronisationsalgorithmus ist eine Notwendigkeit für verteilte Anwendungen, die in einer zeitlichen Abhängigkeit zueinander stehen. Z.B. hat die Fahrwerkssteuerung eines Fahrzeugs die Aufgabe die Spurlage so sicher wie möglich zu gestalten, unabhängig zu Untergrund- und Haftungseigenschaften jedes einzelnen Rades. Die Informationen müssen innerhalb kürzester Zeit verarbeitet und den anderen Teilnehmern zur Verfügung gestellt werden um die verteilte Regelung nicht zu gefährden. Da hier jede Verzögerung der Datenpakete eine Verlängerung der Totzeit der Regelung mit sich ziehen würde, ist dies eine Anwendung für eine zeitlich synchronisierte TDMA-Architektur.

Die Synchronisation von verteilten Knoten ist ein kontinuierlicher Prozess. Jede interne Zeiteinheit eines Knotens basiert auf der Taktrate des verwendeten Quarzes und der Multiplikator-Einheit (PPU). Selbst im Falle, dass Taktrate und Multiplikator identisch sind so ist nicht zu verhindern, dass die Geschwindigkeiten der Systeme über die Zeit hinweg divergieren. Fabrikationsunterschiede der Materialien erzeugen einen Fehler, in der Taktung des Quarzes, welcher sich über die Zeit hinweg akkumuliert. Temperatur- und Druckschwankungen erzeugen eine zusätzliche Variation in der Geschwindigkeit. Zudem nimmt der Alterungsprozess des Bauteils negativen Einfluss auf das Quarzelement, wodurch es kontinuierlich an Frequenz verliert. Erschütterungen des Elements tragen zur Beschleunigung des Alterungsprozesses bei.

2.1 Offset- und Rate-Correction

Um Ungenauigkeiten entgegen wirken zu können, werden Korrekturverfahren eingesetzt:

Da zur Startup-Phase des Netzwerks alle Knoten zu unterschiedlichen Zeiten ihre Aktivitäten beginnen, werden zu einem definierten Zeitpunkt (ausgelöst durch eine Synchronisationsnachricht) alle Knoten dazu aufgefordert ihre Uhr auf einen definierten Wert zu setzen. Dieser Vorgang wird als *Offset Correction* bezeichnet. Anschliessend sind die Abweichungen der Uhren direkt proportional zu der Genauigkeit ihrer Quarze. Aus dieser Genauigkeit und der gewünschten Präzision innerhalb des Netzwerks ergibt sich die Periode in der ein Synchronisationsvorgang wiederholt werden muss. Algorithmen, welche ausschliesslich Offset Correction unterstützen (vgl. TTEthernet) wiederholen zyklisch das definierte Setzen der internen Uhr des

Netzwerkknotens. Dies hat zur Folge, dass die Ungenauigkeit des Systems innerhalb einer Periode nicht konstant ist, und am Ende jeder Periode ihr Maximum erreicht.

Diese Ungenauigkeit resultiert aus der (nahezu) systematischen Geschwindigkeitsdifferenz der Quarze im Netzwerk und einer Statistischen Abweichung. Wird vom Algorithmus die *Rate-Correction* unterstützt, so kann die Geschwindigkeit der internen Uhr angepasst werden. Hierzu wird nicht der Quarz oder sein Multiplikator in seiner Geschwindigkeit variiert, sondern eine logische Uhr definiert. Diese kann in ihrer Schrittweite innerhalb einer durch Hardware gegebenen Auflösung manipuliert werden. Somit wird für eine Anwendung eine Variation der Zeitbasis emuliert, welche die systematische Geschwindigkeitsdifferenz kompensiert. Durch dieses Verfahren resultiert die Restungenauigkeit in der Statistischen Abweichung, welche innerhalb eines, über die Periode hinweg, konstanten Bereiches definiert werden kann.

2.2 Aufbau einer Synchronisation

Um sich einen Überblick über die jeweiligen Synchronisationskonzepte zu bilden, wurden echtzeitfähige Systeme analysiert und auf ihre Synchronisierung hin untersucht (vgl. [15]). Hierbei handelt es sich immer um Master-Client-basierte Konzepte. Der Master stellt seine Zeit zur Verfügung, an der sich alle Clients des Netzwerks synchronisieren. Die Synchronisation eines Netzwerks lässt sich, wie in Abbildung 2.1 dargestellt, in drei Phasen unterteilen.

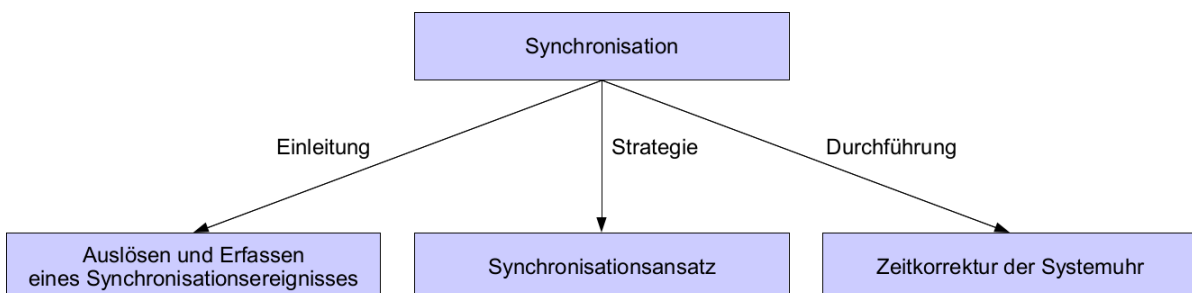


Abbildung 2.1: Übersicht

2.2.1 Ereignisermittlung

Eine Synchronisation kann auf zwei Arten initialisiert werden (2.2 auf der nächsten Seite). Einerseits können festgelegte Zeitpunkte ein Synchronisationsevent auslösen. Ein Client erwartet somit zu besonderen Zeiten eine Synchronisationsnachricht oder fordert diese an. Ebenso sendet der Master zu diesen Zeiten seine Nachrichten. Dieser Ansatz erfordert jedoch eine synchronisierte Ausgangssituation, da sonst Erwartungen und Aktionen der beteiligten Knoten

nicht im Einklang zueinander stehen. Eine andere Methode ist das Auslösen der Synchronisation durch ein Ereignis. Dies ist z.B. das Empfangen einer Nachricht, welche eine spezielle Signatur aufweist und daher als Synchronisationsnachricht klassifiziert werden kann. Hierbei ist zu beachten, dass die Präzision des Algorithmus von der Laufzeit der Nachricht und des Jitters der Übertragung abhängt.

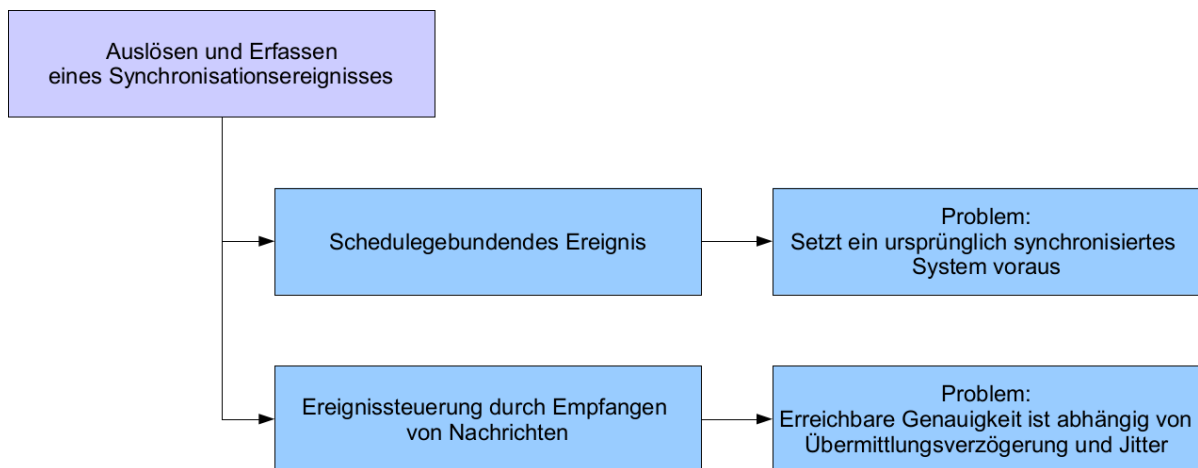


Abbildung 2.2: Ereignisermittlung

2.2.2 Ansatzselektion

Während die Ereignisermittlung ausschliesslich den Zeitpunkt der Einleitung des Algorithmus festlegt, so wird in der Ansatzselektion festgelegt mit welcher Methode der Austausch der Zeitinformationen erfolgt (2.3 auf der nächsten Seite). Hier wird zwischen dem *Time Transmission* Ansatz (TT) und dem *Remote Clock Reading* Ansatzes (RCR) entschieden. Ein Master mit TT-Ansatz sendet aktiv Synchronisationsnachrichten, welche seine lokale Zeit enthalten, per Broadcast an das gesamte Netzwerk. Im RCR-Ansatz ist der Master ein passiver Knoten, der auf Anfragen der Clients reagiert. Jeder Client sendet eine Anfrage an den Master, welcher wiederum mit einer Synchronisationsnachricht antwortet. Diese Methode ermöglicht die dynamische Ermittlung von Nachrichtenlaufzeit und Jitter, benötigt jedoch im Gegenzug mehr Bandbreite und Verarbeitungsaufwand.

2.2.3 Korrekturverfahren

Liegt die globale Zeit des Masters einem Client vor, so wird die Korrektur der internen Uhr des Clients eingeleitet. Hierbei werden Absolute Zeitkorrekturverfahren oder konvergenzbasierte Annäherungsverfahren eingesetzt. Absolute Zeitkorrekturverfahren nutzen Algorithmen

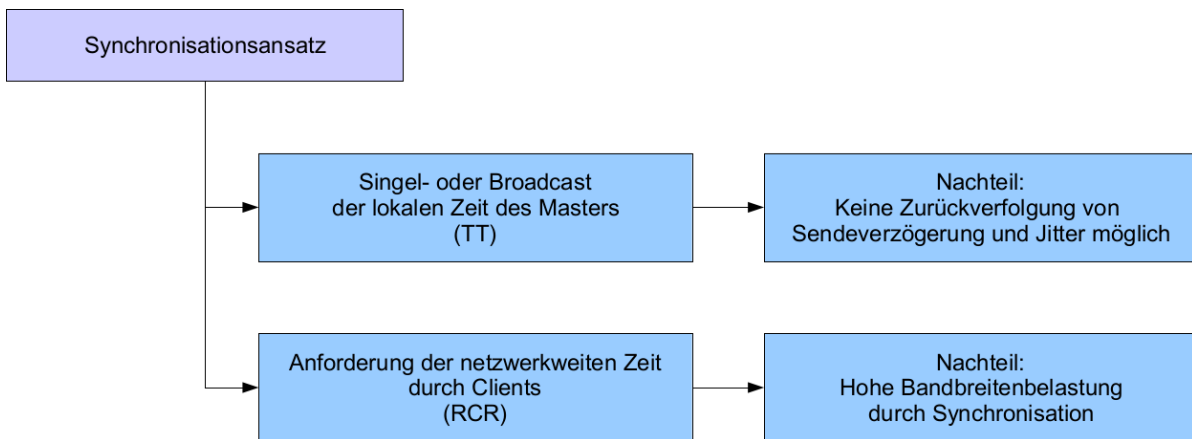


Abbildung 2.3: Ansatzselektion

zur Offset-Correction um ihre Zeitbasis zu beeinflussen. Dies führt zu Zeiten eines Netzwerkstarts zu Zeitsprüngen und Zeitdrifts in Perioden. Die Verarbeitung ist jedoch nicht komplex und daher mit wenig Berechnungsaufwand verbunden. Annäherungsverfahren setzen Rate-Correction-Algorithmen ein, welche wiederum in deterministische oder probabilistische Ansätze unterschieden werden und so indirekt Einfluss auf die Zeit des Knotens nehmen. Hierbei wird die Geschwindigkeit, je nach Abweichung zur Zeit des Synchronisationsmasters, angepasst, ohne ihren Wert direkt zu manipulieren. Somit wird eine variable Trägheit des Netzwerks erreicht, welche kurzzeitige Fehlsynchronisation durch fehlerhafte Pakete kompensieren kann. Jedoch benötigt ein solcher Ansatz im Falle eines Kaltstarts einige Zeit um ein echtzeitfähiges System zu bilden.

2.3 Peer-to-Peer-Synchronisation

Das Prinzip des P2P-Ansatzes ist, die Auflösung aller Synchronisationsrollen. In einer zukünftigen P2P-basierenden Synchronisation würde jeder Teilnehmer, und somit auch seine eigene interne Uhr, das gleiche Gewicht haben. Es gäbe daher keine Masterzeit, an der sich alle Clients zu orientieren haben. Statt dessen sendet jeder Teilnehmer des Netzwerkes seine interne Uhr an seine Nachbarn (P2P). Diese ermitteln aus der Menge aller empfangenen Zeiten, und ihrer eigenen, eine neue Zeit, welche anschliessend über Offset- und Phase-Correction die Basis für die eigene interne Uhr bildet. Hierbei lässt sich in der Anzahl und Definition der Nachbarn die Stärke des Einflusses beschreiben, wodurch ein hochskalierbares Gesamtsystem entsteht. Sollte die Verbindung, sei es durch einen Komplettausfall (System Reliability) oder durch Stromsparfunktionen (Sleepmode) zum Verlust eines Knotens kommen, so ist die Echtzeitfähigkeit des Netzwerkes zu keiner Zeit gefährdet.

3 Modellierung des Ansatzes

Um das Verhalten eines Peer-to-Peer basierten Systems zu studieren wird ein physikalisches Modell aufgestellt und beschrieben. Die Basis bildet ein Physikalisches Pendel. Dieses Beschreibt das Verhalten der internen Uhr eines Knotens (3.1).

3.1 Abbildung eines Knotens

Aus dem Aufbau des Pendels lassen sich Funktionsweise und Netzwerkanalogie herleiten. Die Ruheposition entspricht einer Übereinstimmung der internen Uhr zum Synchronisationsollpunkt. Dieser wird aus den aktuellen Pendelstellungen der Nachbarknoten gebildet und variiert somit über die Zeit. Jede Abweichung der Uhr wird in einem entsprechenden Winkel über die Auslenkung in einer Dimension interpretiert. Der maximale Schwingungsbereich des Pendels ist durch die maximale Korrekturfähigkeit des internen Zeitgebers definiert und wird nur durch die Plattform begrenzt, welche den Synchronisationsalgorithmus in Zukunft ausführen soll.

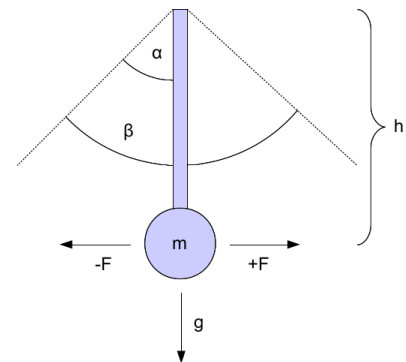


Abbildung 3.1: Pendel

Die Rückstellkräfte des Pendels summieren sich aus dem Grad der Abweichung bezogen auf den Synchronisationsruhepunkt und der Ungenauigkeit des simulierten Knotens. Die Synchronisationskräfte treiben das Pendel somit zurück in die dynamische Ruheposition, während die Ungenauigkeiten dem entgegen wirken.

3.2 P2P Übertragungsmodell

Der Einfluss der Nachbarknoten geschieht durch direkte Interpretation der Nachbarpositionen auf die Sollposition des aktuellen Pendels. Dies kann in ersten Ansätzen als direkt angenommen werden, muss allerdings bei näherer Betrachtung durch ein Federmodell dargestellt werden. Ein direkter Einfluss würde für ein Netzwerkmodell bedeuten, dass weder Nachrichtenlaufzeiten, noch Verarbeitungsaufwand auftreten. Da dieses Modell als realistische Basis für

einen P2P-Netzwerksynchronisationsalgorithmus dienen soll, sind diese Zeiten in Form eines Federmodells zur Kraftübertragung zwischen benachbarten Knoten einzubeziehen.

3.3 Simulationsplattform

Zur Darstellung, Simulation und Validierung dieses Modells ist MATLAB (vgl. [16]) eine geeignete Plattform. Die Numerische Lösungsmethoden der Problemstellungen ermöglicht eine Analyse eines einzelnen Knotens, sowie die Untersuchung des Gesamtsystems, da die gesamte Problemstellung auf Differentialgleichungen zurückzuführen ist. Da MATLAB die Simulation objektorientierte Systeme unterstützt, ist ein skalierbares Modell zu erstellen, welches zudem eine parallelisierbare Ausführung erlaubt.

Octave, eine Alternative zu MATLAB

MATLAB unterstützt mit Simulink eine zeitgesteuerte Simulation von Prozessen. Zudem können ereignisorientierte Simulationsumgebungen erstellt werden, welche jedoch für das bestehende Problem nicht von Relevanz wären. Ebenso wird versucht auf eine Umgebung für zeitgesteuerte Simulationen, und somit auf Simulink zu verzichten, da eine rein numerische MATLAB-Simulation unter freier Lizenz (mit der Hilfe von Octave) steht und dadurch Plattformunabhängig ist.

OMNeT++

Für die fortgeschrittene Simulation von netzwerkbasieren Modellen kann OMNeT++ (vgl. [17]) genutzt werden. Es handelt sich hierbei um eine C++-basierte diskrete und ereignisorientierte Simulationsumgebung. Sie dient der Modellierung und Simulation von Kommunikationsnetzen, Multiprozessorsystemen und anderen verteilten oder parallelen Systemen. Sie stellt ausserdem eine Bibliothek und Framework dar, mit der weitere Simulationsmodelle und Frameworks entwickelt werden können.

Ein Beispiel hierfür ist das INET Framework (vgl. [15]), welches aus mehreren Modellen (TCP, IP, Ethernet, etc.) besteht. OMNeT++ ist frei nutzbar für akademische und nicht kommerzielle Zwecke. Bei einer diskreten ereignisorientierten Simulation treten Ereignisse an diskrete Punkte auf der Zeitachse auf. Dabei wird angenommen, dass nichts zwischen zwei aufeinanderfolgenden Ereignissen passiert. Somit wird die interne Uhr des Modells nach Verarbeitung eines Ereignisses jeweils auf den nächsten Ereigniszeitpunkt vorgerückt. Die Simulation erfolgt also nicht in Echtzeit.

4 Fazit

In dieser Arbeit wurde ein neuartiger Peer-to-Peer-basierter Ansatz präsentiert, welcher eine Alternative zu den Master-Slave-Ansätzen bildet, die zur Zeit in echtzeitfähigen time-triggered Ethernet Netzwerken Verwendung finden. Das Prinzip wurde anhand eines geeigneten physikalischen Modells vorgestellt, welches sich über seine Parametren auf ein Netzwerk portieren lässt. Es wurde gezeigt, dass dieser Ansatz Problemstellungen der heute eingesetzten Synchronisationsalgorithmen angeht und Lösungsvorschläge bietet. Im folgenden müssen die Eigenschaften, das Verhalten und die Anforderungen analysiert, spezifiziert und bestimmt werden.

4.1 Ausblick auf weiteres Vorgehen

Um das Ziel, eines leistungsfähigen Synchronisationsalgorithmuses für den Einsatz in echtzeitfähigen Netzwerken, zu erreichen, muss im nächsten Schritt eine Simulation des physikalischen Prinzips erfolgen. Dies kann über die vorgeschlagene Plattform MATLAB erfolgen, woraus sich die Funktion, Rahmenbedingungen und das Verhalten analysieren lassen. Zudem können Metriken bestimmt und bewertet werden, um einen späteren Vergleich mit anderen Synchronisationsverfahren herstellen zu können. Hierauf aufbauend erfolgt die Portierung des Ansatzes und seiner Parameter auf ein netzwerkfähiges Modell, da das Ziel ist diese Algorithmen in echtzeitfähigen Ethernet-Netzwerken einzusetzen. Die Simulation des netzwerkfähigen Modells mit Hilfe der Modellierungsumgebung OMNeT++ lässt eine Evaluation und Validierung zu. Anschliessend kann ein Vergleich zwischen dem entwickelten Synchronisationsansatz mit eingesetzten Synchronisationsverfahren gezogen werden. Alle Schritte sind iterativ zu sehen, so dass neu gewonnene Erkenntnisse in die Entwicklung des Peer2Peer-basierten Ansatzes fließen und umgesetzt werden können.

Ist die Spezifikation des Synchronisationsalgorithmus abgeschlossen, so ist ein Einsatz geplant. Hierzu würden sich die eigens entwickelten Time-Triggered-Ethernet Steuergeräte der CoRE (Communication over Real-time Ethernet) Gruppe eignen, welches zur Zeit ein durch TTEthernet spezifiziertes Master-Slave-Protokoll nutzt. Somit würde diese Arbeit ihren Abschluss unter einer realen Umgebung finden.

4.2 Risikoabschätzung

Das Risiko liegt in der Funktionsweise des Algorithmus. Da es sich um einen vollständig neuen Ansatz handelt, welcher auf Parallelität basiert, ist es schwer abzuschätzen, wie sein Verhalten unter grenzwertigen Bedingungen ausfallen wird. Eine Gefahr besteht darin, dass die Stabilität beeinträchtigt ist und ein System basierend auf dem Peer-to-Peer-Konzeptes unangebrachte Laufzeiten aufweist, um einen stabilen Zustand zu erreichen. Jedoch ist damit nicht zu rechnen, da durch Parametervariation die Eigenschaften des Systems beeinflusst werden können. Allerdings ist es denkbar, dass es sich bei dem Algorithmus um ein eskalierendes System handelt, welches nicht oder nur mit viel Aufwand zu kontrollieren wäre.

Da zur Zeit keine Berichte über ähnliche Vorhaben oder deren Entwicklung bekannt sind, ist die Gefahr Ergebnisse in dieser Arbeit zu produzieren, welche hierdurch obsolet werden würden, zu vernachlässigen.

Insgesamt kann das Risiko des Projekts als gering eingestuft werden. Die in dieser Arbeit zusammengefassten Prinzipien und Möglichkeiten lassen eine positive Prognose im Hinblick auf das Ziel der Arbeit zu.

Abbildungsverzeichnis

2.1	Übersicht	5
2.2	Ereignisermittlung	6
2.3	Ansatzselektion	7
3.1	Pendel	8

Literaturverzeichnis

- [1] Bob Pickles, "Avionics Full Duplex Switched Ethernet (AFDX)," SBS Technologies, Tech. Rep., 2006. [Online]. Available: http://www.sierrasales.com/pdfs/AFDX_Overview.pdf
- [2] K.-W. Gaede, *Zuverlässigkeit, mathematische Modelle*. München: Hanser, 1977.
- [3] P. Van Mieghem, *Performance Analysis of Communications Networks and Systems*. Cambridge: Cambridge University Press, 2006.
- [4] Aeronautical Radio Incorporated, "Aircraft Data Network," ARINC, Annapolis, Maryland, Standard 664, 2002.
- [5] M. Felser, *Real-Time Ethernet: Standards and PROFINET*. International Federation of Automatic Control, 2005, vol. IFAC Summer School in Prague 2005.
- [6] J. Loeser and H. Haertig, "Low-latency hard real-time communication over switched Ethernet," in *Real-Time Systems, 2004. ECRTS 2004. Proceedings. 16th Euromicro Conference on*, Jun. 2004, pp. 13–22.
- [7] G. Schnell and B. Wiedemann, *Bussysteme in der Automatisierungs- und Prozesstechnik - 7. Auflage*. Vieweg + Teubner, 2008.
- [8] C. Hammerschmidt, "BMW brings Internet protocol under the hood," *EE Times Europe*, 2007. [Online]. Available: <http://www.eetimes.com/showArticle.jhtml?articleID=204300325>
- [9] K. Müller, T. Steinbach, F. Korf, and T. C. Schmidt, "A Real-time Ethernet Prototype Platform for Automotive Applications," in *Proceedings of the 1st IEEE International Conference on Consumer Electronics - Berlin (ICCE-Berlin)*, 2011, to appear.
- [10] T. Steinbach, F. Korf, and T. C. Schmidt, "Comparing Time-Triggered Ethernet with Flex-Ray: An Evaluation of Competing Approaches to Real-time for In-Vehicle Networks," in *8th IEEE Intern. Workshop on Factory Communication Systems*. Piscataway, New Jersey: IEEE Press, May 2010, pp. 199–202.

-
- [11] —, “Simulationsbasierte Evaluierung von Metriken in Echtzeit-Ethernet basierten Fahrzeugnetzen,” in *6ter GI/ITG-Workshop Leistungs-, Zuverlaessigkeits- und Verlaesslichkeitsbewertung von Kommunikationsnetzen und verteilten Systeme (MMBnet 2011)*, 2011, to appear.
- [12] FlexRay Consortium, “Protocol Specification,” FlexRay Consortium, Stuttgart, Specification 2.1, Dec. 2005.
- [13] T. Pop, P. Pop, P. Eles, Z. Peng, and A. Andrei, “Timing analysis of the FlexRay communication protocol,” *Real-Time Systems*, vol. 39, no. 1-3, pp. 205–235, 2007.
- [14] P. Ferrari, A. Flammini, S. Rinaldi, and G. Gaderer, “Evaluation of clock synchronization accuracy of coexistent Real-Time Ethernet protocols,” in *IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication*, Sep. 2008, pp. 87–91.
- [15] OMNeT++ Community, “INET Framework for OMNeT++ 4.0.” [Online]. Available: <http://inet.omnetpp.org/>
- [16] A. M. Law and W. Kelton, *Simulation modeling & analysis*, 2nd ed., ser. McGraw-Hill series in industrial engineering and management science. New York: McGraw-Hill, 1991.
- [17] OMNeT++ Community, “OMNeT++ 4.0.” [Online]. Available: <http://www.omnetpp.org>