

# Ausarbeitung Anwendungen 1 WS2010/11

Florian Bartols

Restbussimulation von Time-Triggered Ethernet  
in Automotive-Anwendungen

## **Thema dieser Ausarbeitung**

Restbussimulation von Time-Triggered Ethernet in Automotive-Anwendungen

## **Stichworte**

Restbussimulation, Automotive-Anwendungen, V-Modell, Bussysteme, TTEthernet, RT-Linux

## **Kurzzusammenfassung**

Heutige Automobile sind komplexe, verteilte Systeme die über eine Vielzahl von Steuergeräten für Fahrerassistenz- und Komfortsysteme verfügen. Für frühzeitiges Testen im Entwicklungsprozess dieser Geräte wird die Restbussimulation eingesetzt. Die verwendeten Bussysteme stoßen mit ihren geringen Bandbreiten an die Grenzen neuer kamerabasierter Systeme. Time-Triggered Ethernet ist als möglicher Lösungskandidat im Gespräch, hohe Bandbreiten bei gleichzeitiger Echtzeitfähigkeit anzubieten. Allerdings fehlt bislang ein geeignetes Werkzeug, das die Restbussimulation basierend auf Time-Triggered Ethernet, bietet. In dieser Arbeit werden die Grundlagen der Restbussimulation und die von Time-Triggered Ethernet besprochen. Eine Analyse der Anforderungen, Problemstellungen und Risiken werden dargelegt und es wird ein Ausblick auf kommende Arbeiten gegeben.

## **Topic of this paper**

Clustersimulation of Time-Triggered Ethernet in context of Automotive Engineering

## **Keywords**

Clustersimulation, Automotive Applications, V-Model, Bussystems, TTEthernet, RT-Linux

## **Abstract**

Today's vehicles are complex distributed systems which have a various number of control units for driver assistance and comfort. For early testing in the development process of these units, the utilization of cluster simulation is common. The used bus systems have limits for new camera-based systems due to their low bandwidth. Time-Triggered Ethernet is a possible solution for providing high bandwidth, while keeping real-time behavior. However, there is still a tool missing that provides cluster simulation basing on Time-Triggered Ethernet. In this work, the fundamentals of cluster simulation and Time-Triggered Ethernet are discussed. An analysis of the requirements, problems and risks are explained and finally, an outlook on future work is given.

# Inhaltsverzeichnis

<b>Abbildungsverzeichnis</b>	<b>4</b>
<b>1 Einleitung</b>	<b>5</b>
1.1 Motivation . . . . .	5
1.2 Inhaltlicher Aufbau der Ausarbeitung . . . . .	6
<b>2 TTEthernet-Restbussimulation</b>	<b>7</b>
2.1 Grundlagen einer Restbussimulation . . . . .	7
2.1.1 Anforderungen an eine Restbussimulation . . . . .	7
2.1.2 Anwendungsbeispiel . . . . .	8
2.1.3 Restbussimulation im Entwicklungsstadium . . . . .	9
2.1.4 Vorhandene Lösungen von Restbussimulatoren . . . . .	11
2.2 Grundlagen zu Time-Triggered Ethernet . . . . .	12
2.2.1 Nachrichtenklassen von TTEthernet . . . . .	12
2.2.2 Zeitsynchronisierung in TTEthernet . . . . .	13
<b>3 Anforderungen und Problemstellungen</b>	<b>14</b>
3.1 Beschreibung der Anforderungen . . . . .	14
3.2 Problemstellungen . . . . .	15
<b>4 Abschätzung der Risiken</b>	<b>16</b>
<b>5 Zusammenfassung und Ausblick</b>	<b>17</b>
5.1 Zusammenfassung . . . . .	17
5.2 Ausblick . . . . .	18
<b>Literaturverzeichnis</b>	<b>19</b>

# Abbildungsverzeichnis

2.1	Geschwindigkeitsabhängige Fahrwerksteuerung . . . . .	8
2.2	Restbussimulation aller relevanten Teilnehmer . . . . .	9
2.3	Ansiedlung der Restbussimulation im V-Modell . . . . .	10
2.4	Zweiwegezeitsynchronisierung im TTEthernet . . . . .	13
3.1	Mögliches Szenario einer TTEthernet-basierten Restbussimulation . . . . .	15

# 1 Einleitung

Aktuelle Automobile sind komplexe, verteilte Systeme mit weit mehr als 70 Steuergeräten (vgl. [Navet u. a., 2005](#)), die über verschiedenste Bussysteme miteinander vernetzt sind. CAN (Controller Area Network (vgl. [Robert Bosch GmbH](#))), LIN (Local Interconnect Network (vgl. [LIN-Administration](#))), MOST (Media Oriented Systems Transport (vgl. [MOST Cooperation](#))) und neuerdings auch FlexRay (vgl. [FlexRay Consortium](#)) sind als "state of the art" Bussysteme im Automobil zu finden und unterliegen unterschiedlichen Anforderungen (vgl. [Zimmermann und Schmidgall, 2011](#)). Während CAN und FlexRay für zeitkritische Systeme, wie die Motor- und Fahrwerksteuerung eingesetzt werden, bedient LIN den Bereich mit weniger harten Zeitanforderungen z.B. Tür- oder Schiebedachelektronik. MOST wird ausschließlich für Telematik- und Multimediaanwendungen genutzt. Diese Systeme haben allerdings ein Problem, das für zukünftige Anwendungen, wie z.B. kamerabasierende Fahrerassistenzsysteme (vgl. [Reif, 2009](#)), relevant ist: eine zu geringe Bandbreite oder keine Echtzeitfähigkeit. Time-Triggered Ethernet (vgl. [Steiner, 2008](#)) ist ein möglicher Kandidat das Problem zu lösen, hohe Bandbreiten bei gleichzeitiger Einhaltung von harten Echtzeitanforderungen zu ermöglichen. Basierend auf aktueller standard Switched Ethernet-Technologie steht ein breites Level an Know-How zur Verfügung, sodass eine Einführung dieses Systems erleichtert wird.

Der Einsatz von Elektronik im Automobil ist in den letzten Jahren signifikant gestiegen und spiegelt sich zudem in den Gesamtkosten des Automobils wider. Lag dieser Anteil 1975 mit der Verwendung von einer elektronischen Motorsteuerung bei ca. fünf Prozent, ist dieser im Jahre 2005 auf ca. vierzig Prozent gestiegen. Ein Ende dieser Entwicklung ist durch die Einführung weiterer Assistenzsysteme nicht zu abzusehen und wird einen noch höheren Anteil in den nächsten Jahren ausmachen.

## 1.1 Motivation

Die Elektronik im Automobil wird heutzutage in der Regel nicht durch den Automobilhersteller (Original Equipment Manufacturer - OEM) selber entwickelt, sondern dezentral bei Zulieferern. In einer ca. dreijährigen Entwicklungszeit (vgl. [Schäuffele und Zurawka, 2010](#)) wird das Automobil sowohl vom Design, als auch von der Elektronik durch den Hersteller konzipiert

und als Systemmodell zugrunde gelegt. Anhand diesem Modell entwickelt der Zulieferer die geforderten Steuergeräte.

Das Problem einer dezentralen Entwicklung ist unter Umständen ein zu spätes Entdecken von Fehlern in den Steuergeräten (vgl. [Riegraf u. a., 2007](#)). Je später Fehler entdeckt werden, desto höher sind die Kosten bei der Beseitigung dieser. Im schlimmsten Fall werden Fehler erst entdeckt, wenn das Automobil im Einsatz ist. Die daraus resultierenden Rückrufaktionen bringen nicht zu unterschätzende Kosten mit sich, von einem entstehenden Imageschaden abzusehen. Eventuell verlängerte Entwicklungszeit birgen zusätzliche Kosten.

Ein frühzeitiges Testen der Elektronik im Automobil ist daher unumgänglich. Die Restbussimulation kann als ein Werkzeug eingesetzt werden, um Steuergeräte während der Entwicklung zu testen. Time-Triggered Ethernet verfügt als ein neues Kommunikationsmedium über noch keine geeignete Werkzeugkette in der Entwicklung von Automotive-Komponenten und -Anwendungen. Daher ist dieses Themengebiet sehr attraktiv für zukünftige Arbeiten, da auf bewährte Methoden zurückgegriffen und neue Prinzipien mit einbracht werden können.

## 1.2 Inhaltlicher Aufbau der Ausarbeitung

Im nachfolgenden Abschnitt [2](#) werden die Grundlagen einer Restbussimulation und Prinzipien von Time-Triggered Ethernet erläutert. Im anschließenden Abschnitt [3](#) werden die Anforderungen und die Problemstellung an eine Time-Triggered Ethernet-basierte Restbussimulation nahe gebracht. Die Risiken dieses Themengebietes behandelt der anknüpfende Bereich [4](#). Eine Zusammenfassung und ein Ausblick schließen diese Ausarbeitung in der letzten Passage [5](#) ab.

## 2 TTEthernet-Restbussimulation

Dieser Abschnitt beschreibt zuerst die Grundlagen einer Restbussimulation. Anschließend wird Time-Triggered Ethernet als Basis der Restbussimulation eingeführt und kurz dessen rudimentäre Funktionsweise erläutert.

### 2.1 Grundlagen einer Restbussimulation

Die Restbussimulation ist in der Automobilindustrie eine weit verbreitete Methode in der Entwicklung von elektronischen Steuergeräten (vgl. [Riegraf u. a., 2007](#)). Anders als bei einer reinen Softwaresimulation des Systems werden in einer Restbussimulation nicht das ganze verteilte System simuliert. Es wird ein hybrider Ansatz verwendet, in dem simulierte und reale Teile gemeinsam genutzt werden.

#### 2.1.1 Anforderungen an eine Restbussimulation

Die Verbindung zwischen dem Simulator und dem zu testenden Steuergerät wird über das reale Kommunikationsmedium erstellt. Eine Stimulation erfolgt dabei nicht durch einfache elektronische Signale, sondern es werden Datenpakete übertragen, die über den Protokollstack empfangen werden. Um dieses zu ermöglichen, muss der Restbussimulator das verwendete Kommunikationsprotokoll vollständig unterstützen, indem Zeitsynchronisierungsmechanismen (z.B. bei FlexRay) oder Busarbitrierungsverfahren (bei CAN) durchgeführt werden. Für die Steuergeräte muss die Simulation transparent erfolgen, da sich ansonsten das Verhalten des Gerätes nicht überprüfen lässt. Fehlen protokollspezifische Steuerdaten im System, verlassen die Steuergeräte den Funktionszustand und gehen in einen Fehlerzustand über.

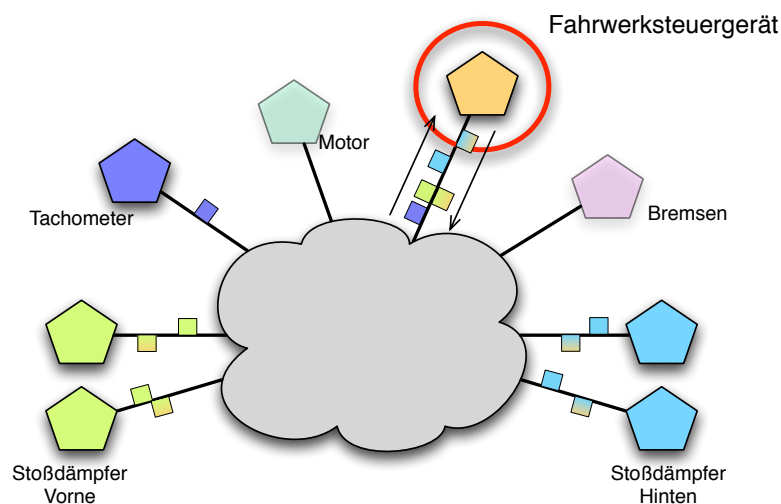
Aktuelle Automobile verfügen über eine Vielzahl von Steuergeräten, die für unterschiedliche Anwendungen verantwortlich sind und über das Bordnetz miteinander kommunizieren. Daraus folgt, dass eine hohe Nachrichtenkomplexität bei der Übertragung entsteht. Während der Testphase in der Entwicklung eines Steuergerätes ist es deshalb wichtig, die hohe Komplexität zu verringern, da ansonsten der Test und die Konfiguration des Simulators zu viel

Zeit beansprucht. Es wird erreicht, in dem nur die für das Steuergerät relevanten Teile des Systems simuliert werden. Bereiche, die weder mit dem Gerät kommunizieren, noch für das Übertragungsprotokoll nötig sind, werden vernachlässigt.

Durch diesen Aufbau ist es möglich, Steuergeräte frühzeitig im Entwicklungsprozess zu testen, ohne das gesamte System einzusetzen, da dieses zum Entwicklungszeitpunkt noch nicht zur Verfügung steht oder aus Kostengründen eingespart wird.

### 2.1.2 Anwendungsbeispiel

In Abbildung 2.1 ist als Beispiel die Entwicklung einer fiktiven geschwindigkeitsabhängigen Fahrwerksteuerung dargestellt, die nach Geschwindigkeit oder Beschleunigung das Fahrverhalten über die Stoßdämpferhärte des Automobils beeinflusst. Die zugehörigen Sensoren und Aktoren sind in diesem Fall die Stoßdämpfer vorn und hinten; außerdem der Tachometer. Weiterhin sind am Bordnetz zusätzliche Steuergeräte für den Motor und Bremsen angefügt. Die Stoßdämpfer senden ihren aktuellen Zustand, z.B. Hublänge des Federbeins,

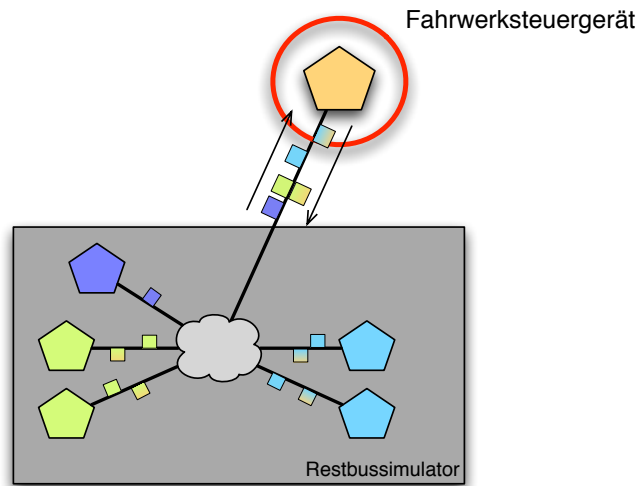


**Abbildung 2.1:** Geschwindigkeitsabhängige Fahrwerksteuerung

an das Fahrwerksteuergerät und dieses berechnet aus den Daten des Tachometers (Geschwindigkeit und Beschleunigung) die neuen Dämpfereinstellungen, die über das Bordnetz an die Aktoren übermittelt werden. Motor- und Bremsendaten werden in dieser fiktiven Beispielanwendung nicht berücksichtigt, sodass diese Geräte keine Auswirkungen auf das Fahrwerksteuergerät haben.



Das Testen des entwickelten Fahrwerksteuergerätes wird mit Hilfe des Restbussimulators in [Abbildung 2.2](#) durchgeführt. Dabei werden die Geräte, die keine Auswirkungen auf das Verhalten der Anwendung haben (Motor und Bremse), nicht simuliert. Alle anderen Geräte (Tachometer und Dämpfer) werden im Simulator so konfiguriert, sodass sich dieser wie ein „echtes“ System verhält, indem die gleichen Nachrichten der simulierten Teilnehmer vom Simulator an das Steuergerät gesendet und empfangen werden.



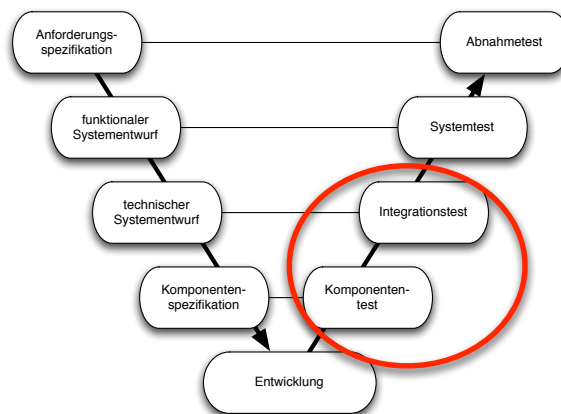
**Abbildung 2.2:** Restbussimulation aller relevanten Teilnehmer

Mit dieser Konfiguration ist es möglich, das Steuergerät intensiv zu testen, ohne auf die Komplexität des Gesamtsystems einzugehen. Die Sensoren und Aktoren dieser Anwendung müssen zudem nicht aufgebaut werden und lassen sich einsparen.

### 2.1.3 Restbussimulation im Entwicklungsstadium

Der Einsatz einer Restbussimulation zum Testen hängt von dem aktuellen Entwicklungsstand des Automobils ab. In der Softwareentwicklung von Automotive Anwendungen wird weitestgehend das V-Modell (vgl. [IABG, 2009](#)) angewendet (vgl. [Schäuffele und Zurawka, 2010](#)), da es gegenüber dem teilweise noch üblichen „wir fangen sofort an und dokumentieren später“ viele Vorteile hat. Die kontrollierten Entwicklungsschritte führen zu einer höheren Endqualität und der aktuelle Stand des Projektes kann leichter bewertet werden, wodurch der Zeitplan realistischer eingeschätzt werden kann (vgl. [Meier, 2008](#)).

Im V-Modell wird eine Top-Down Entwicklung betrieben, in der zuerst Anforderungen definiert und anschließend abstrakt das System entworfen wird. Nachdem die einzelnen Komponenten entworfen wurden, wird in der Entwicklungsphase die Software für das Steuergerät implementiert. Abbildung 2.3 illustriert die einzelnen Entwicklungsschritte auf der linken Seite.



**Abbildung 2.3:** Ansiedlung der Restbussimulation im V-Modell

Wie auf der rechten Seite in Abbildung 2.3 zu sehen ist, werden zu den einzelnen Stadien im Entwicklungsprozess Tests ausgeführt, die die Entwicklungsschritte validieren. Zuerst wird dabei auf Komponentenebene getestet. Ein anschließender Integrationstest überprüft das Zusammenspiel verschiedener Komponenten. Der Systemtest kontrolliert das System auf korrekte Funktionsweise. Abschließend in der Entwicklung wird im Abnahmetest überprüft, ob das Produkt den geforderten Anforderungen entspricht.

Die Restbussimulation befindet sich in Hinsicht der Testebenen im V-Modell auf Komponenten- und Integrationsteststufe, da es mit diesem Werkzeug möglich ist, das Verhalten und das Zusammenspiel mit anderen Komponenten des entwickelten Steuergerätes zu überprüfen.

### Testmöglichkeiten

Die Testmöglichkeiten der Restbussimulation richten sich nach dem aktuellen Entwicklungsstatus des Gerätes. Während in den frühen Stadien grundlegende Funktionen zur Kommunikation getestet werden, muss im weiteren Verlauf auf eine korrekte Ausführung der Anwendung geachtet werden. Eine Überprüfung auf Einhaltung der Protokollspezifikation oder

Balastungs- und Stresstest des Protokollstacks kann dabei frühzeitig helfen, Fehler auf unterer Ebene zu erkennen. Außerdem kann dadurch sichergestellt werden, dass die gesendeten Nachrichten systemkonform übertragen werden, indem der Inhalt analysiert wird.

Wird das Verhalten eines Gerätes überprüft, kann dieses über zwei Wege erfolgen (vgl. [Galla, 1999](#); [Galla und Pallierer, 1999](#); [Schlager, 2008](#)). Zum einen wird das Gerät durch den Restbussimulator nur stimuliert. Es werden Befehle an das Gerät gesendet, die es daraufhin ausführt, ohne dabei auf die Reaktion des Gerätes einzugehen. Man spricht bei dieser Variante von „Open Loop“-Testing.

Zum anderen wird auf die Reaktion des Gerätes eingegangen, sodass Reglerstrukturen getestet werden können. Bei dieser Variante spricht man deshalb von „Closed Loop“-Testing. Erreicht wird dieses dadurch, dass die Nachrichten, die das Gerät sendet, analysiert werden. Anschließend wird eine passende Antwortnachricht gesendet. Gerade bei Fahrerassistenzsystemen, wie dem Antiblockier System (ABS) oder dem Elektronischen Stabilitätsprogramm (ESP), ist diese Form des Testens wichtig. Diese Systeme basieren auf komplexen Reglerstrukturen und können so auf korrektes Verhalten überprüft werden.

#### 2.1.4 Vorhandene Lösungen von Restbussimulatoren

In einer kurzen Marktübersicht wird auf einige der momentan vorhandenen Tools zur Verwendung einer Restbussimulation eingegangen.

**CANoe** wird vom Hersteller Vektor Informatik GmbH entwickelt (vgl. [Vector Informatik](#)). Es unterstützt die derzeit im Automobil verwendeten Bussysteme CAN, FlexRay, LIN und MOST. Zusätzlich wird Ethernet unterstützt.

**FlexXCon** unterstützt FlexRay und LIN als Bussystem in der Entwicklung von Steuergeräten und wird von Eberspächer GmbH (vgl. [Eberspächer](#)) entwickelt.

**CanEasy** ist eine Entwicklung der Firma Schleißheimer GmbH (vgl. [Schleißheimer GmbH](#)). Mit diesem Tool können LIN- und CAN-basierte Netzwerke getestet und simuliert werden.

Diese vorhandenen Tools unterstützen derzeit alle im Automobil verwendeten Kommunikationsprotokolle. Allerdings wird derzeit Ethernet nur durch **CANoe** unterstützt. Für Time-Triggered Ethernet sind aktuell noch keine Werkzeuge am Markt zu finden, die eine Restbussimulation ermöglichen. Für eine zukünftige Einführung dieses Systems ein entscheidender Faktor.

## 2.2 Grundlagen zu Time-Triggered Ethernet

Nachdem die Grundlagen einer Restbussimulation erklärt wurden, wird jetzt auf das echtzeitfähige Netzwerkprotokoll Time-Triggered Ethernet eingegangen und dessen grundlegenden Eigenschaften beschrieben.

Time-Triggered Ethernet (TTEthernet) (vgl. [Steiner, 2008](#)) wurde von TTTech (vgl. [TTTech Computertechnik AG](#)) in Zusammenarbeit mit Honeywell (vgl. [Honeywell International](#)) entwickelt, basiert auf Forschungsergebnissen der TU Wien und ist echtzeitfähiges Netzwerkprotokoll, das auf dem Ethernet-Standard basiert. Die Herausforderung besteht darin, eine gleichzeitige Übertragung von Echtzeitnachrichten im Switch zu verhindern, da ansonsten kein deterministisches Verhalten erzielt werden kann.

Echtzeitfähige Ethernet-Protokolle lassen sich in drei Klassen aufteilen, die voneinander abweichende Strategien verfolgen die gleichzeitige Übertragung von Echtzeitdaten zu verhindern. In **Token-basierten Netzwerken** ist es nur demjenigen Knoten erlaubt eine Übertragung zu starten, der momentan den Token besitzt. Dieser wird an jeden Teilnehmer im Netz weitergereicht, sodass eine faire Übertragung für jeden Teilnehmer erreicht wird. Diese Verfahren wird bereits erfolgreich in der Prozessautomatisierung eingesetzt (vgl. [EtherCAT Technology Group](#)). **Bandbreiten limitierende Protokolle** ermöglichen eine echtzeitfähige Übertragung, in dem die Bandbreite für jeden Knoten festgelegt wird und diese nicht übersteigen darf. AFDX (Avionics Full Duplex Switched Ethernet (vgl. [Aeronautical Radio Incorporated, 2002](#))) nutzt dieses Prinzip und wird derzeit im Airbus A380 verwendet. **Zeitgesteuerte (Time-Triggered) Protokolle** verwenden einen zeitlich koordinierten Zugriff auf das Übertragungsmedium (Time Division Multiple Access), das durch eine gemeinsame synchronisierte Zeitbasis aller Teilnehmer erreicht wird. Jedem Teilnehmer ist es nur an einem bestimmten Zeitpunkt erlaubt eine Übertragung zu starten. Als Beispiel in der Prozessautomatisierung wird im Profinet (vgl. [PROFIBUS & PROFINET International](#)) dieses Verfahren angewendet.

TTEthernet basiert auf dem zuletzt beschriebenen Verfahren und dessen Verhalten wurde in vorherigen Arbeiten durch Messungen (vgl. [Bartols, 2010](#); [Bartols u. a., 2011](#)), Simulationen (vgl. [Steinbach, 2011](#); [Steinbach u. a., 2011](#)) und einer mathematischen Analyse (vgl. [Steinbach u. a., 2010](#)) nachgewiesen.

### 2.2.1 Nachrichtenklassen von TTEthernet

TTEthernet ermöglicht es, Echtzeitnachrichten und normale Nachrichten über das gleiche physikalische Medium zu übertragen. Dazu werden drei Nachrichtenklassen definiert, die unterschiedliche Prioritäten, Übertragungs- und Zeitanforderungen haben.

**TT-Nachrichten** besitzen die höchste Nachrichtenpriorität, die genauesten Zeit- und Übertragungsanforderungen. Deshalb werden die Echtzeitdaten zeitgesteuert versendet und empfangen.

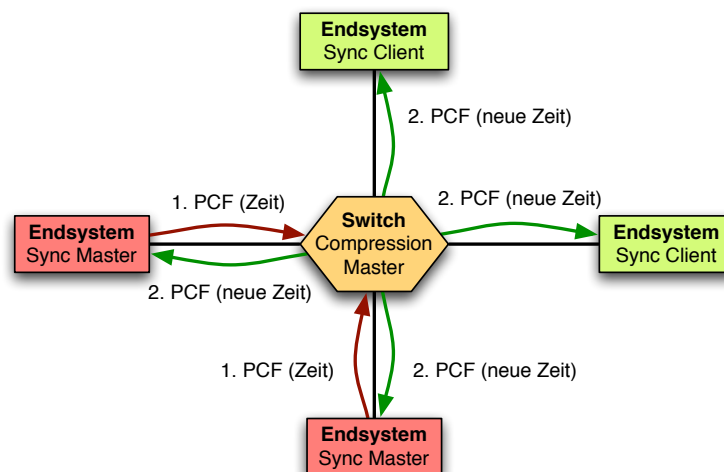
**RC-Nachrichten** werden über eine Bandbreitenlimitierung gesteuert und entsprechen der AFDX-Protokollspezifikation. Nachrichten werden garantiert übertragen, allerdings mit niedriger Priorität als **TT-Nachrichten**.

**BE-Nachrichten** entsprechen standard Ethernet und sind am niedrigsten priorisiert. Eine Garantie der Nachrichtenübertragung ist nicht vorgesehen.

## 2.2.2 Zeitsynchronisierung in TTEthernet

Damit eine zuverlässige Übertragung bei allen Teilnehmern möglich ist, muss es eine gemeinsame Zeitbasis geben. TTEthernet definiert daher einen Zweiwegezeitsynchronisierungsdienst, der transparent zum Übertragungsprotokoll arbeitet und keine zusätzliche Hardware erfordert. Es werden Synchronisationsnachrichten (PCF - Protocol Control Frames) versendet, die auf Endsystemen und Switches verarbeitet werden.

In Abbildung 2.4 ist dieser Prozess dargestellt. Im ersten Schritt senden die **Synchronisations Master** ihre aktuelle Zeit zum **Compression Master**. Dieser berechnet eine Durchschnittszeit aus allen Werten und sendet diese im zweiten Schritt an die **Synchronisations Clients** und alle weiteren beteiligten Teilnehmer im Netz. Diese synchronisieren sich anschließend auf die neue Zeit.



**Abbildung 2.4:** Zweiwegezeitsynchronisierung im TTEthernet

# 3 Anforderungen und Problemstellungen

Nachdem in den vorherigen Abschnitten die Restbussimulation und Time-Triggered Ethernet vorgestellt wurde, wird in diesem Abschnitt auf die Anforderungen und anschließend Problemstellungen an eine TTEthernet-basierte Restbussimulation eingegangen.

## 3.1 Beschreibung der Anforderungen

Die Anforderungen an eine Restbussimulation wurden zum Teil im Abschnitt [2.1](#) beschrieben und bestehen in einer vollständigen Unterstützung des verwendeten Protokolls bei gleichzeitiger dynamischer Konfigurierbarkeit in Abhängigkeit zur Testumgebung. Abweichende Funktionalitäten durch nicht Einhaltung des Protokolls beeinflussen die Testergebnisse negativ. Weiterhin sollte möglichst immer der gleiche Simulator zum Einsatz kommen, sodass ein aufwendiges Neuaufsetzen des Simulators entfällt. Dadurch wird Zeit und gleichzeitig Kosten eingespart. Wichtig bei der dynamischen Konfigurierbarkeit ist die Möglichkeit, die vom Hersteller vergebenen Systemmodelle zu laden. Dieses erleichtert die Konfiguration erheblich, da ein händisches modellieren des Systems ausbleibt und Fehler, die dabei auftreten können, vermieden werden.

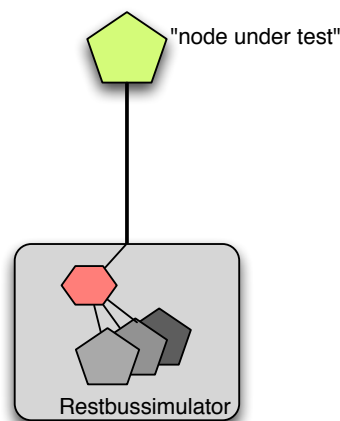
Speziell für eine Time-Triggered Ethernet Restbussimulation gelten zusätzliche Anforderungen, die es so bei gängigen Simulatoren aktueller Bussysteme nicht gibt. TTEthernet unterscheidet zwischen TT-, RC- und BE-Nachrichten, wobei TT- und RC-Nachrichten jeweils Echtzeitanforderungen besitzen (siehe Abschnitt [2.2.1](#)). Es muss dabei sichergestellt werden, dass diese Nachrichten unter keinen Umständen verzögert werden, dass eine Deadline in einer Anwendung verpasst wird. Zusätzlich ist darauf zu achten, dass die Priorität der Nachrichten eingehalten wird und niederprioritäre Daten zurückgehalten werden. Weiterhin müssen Zeitsynchronisierungsfunktionalitäten unterstützt werden. Ist das zu testende Gerät ein Client im Synchronisierungsprozess, so muss der Simulator die fehlenden Aufgaben der Zeitsynchronisierung (Synchronisations Master, Compression Master, siehe Abschnitt [2.2.2](#)) übernehmen, damit ein zeitgesteuertes Übertragen weiterhin möglich ist und das Gerät nicht in den Fehlerzustand übergeht.

## 3.2 Problemstellungen

Die Problemstellungen an eine TTEthernet-basierte Restbussimulation kann wiederum eingeteilt werden in allgemeine Problemstellungen und eigens für TTEthernet.

Es müssen komplexe Reglerstrukturen in einer Restbussimulation verwendet werden können, die das Verhalten der simulierten Teilnehmer wiedergeben. Dafür müssen die simulierten Knoten im Schedule des Restbussimulators abgebildet werden (vgl. [Galla, 1999](#)). Es muss sichergestellt werden, dass genug Rechenleistung für die simulierten Knoten zur Verfügung steht und die Berechnungen keine Auswirkungen auf das Senden von TT- und RC-Nachrichten haben. Ist die Funktionalität zu umfangreich für eine Abbildung im Systemschedule, muss der Regler vereinfacht werden, sodass es trotzdem möglich ist, das Verhalten zu simulieren.

TTEthernet bietet weiterhin die Möglichkeit, das Netzwerk mit einem separaten Tool zu designen und als XML-Schema zu speichern, ähnlich einer FIBEX-Datei (Field Bus Exchange Format (vgl. [FIBEX Expert Group, 2009](#); [Zimmermann und Schmidgall, 2011](#))) die bereits bei FlexRay-basierten Bord-Netzen eingesetzt wird. In dieser Datei werden sämtliche Teilnehmer und deren Verbindungen zueinander beschrieben. Es muss daher möglich sein, diese Datei einzulesen, um die Netzwerkkonfiguration des Simulators zu vereinfachen. Hierbei gilt es herauszufinden, welche Nachrichten den Restbussimulator verlassen und welche Nachrichten intern ausgetauscht werden. Anders als bei den momentan eingesetzten Bussen handelt es sich bei TTEthernet um keine gemeinsame Verbindung, sondern um ein geschichtetes Netz, das für jeden Teilnehmer eine eigene Kollisionsdomäne bietet, zu sehen in [Abbildung 3.1](#). Die Kommunikation der simulierten Knoten muss nicht konfiguriert werden, sodass dieses die Verhaltensimplementierung bei verteilten Reglerstrukturen erleichtert.



**Abbildung 3.1:** Mögliches Szenario einer TTEthernet-basierten Restbussimulation

## 4 Abschätzung der Risiken

Die aktuellen Risiken dieser Arbeit sind unter anderem fehlende Geräte, die TTEthernet als Übertragungsprotokoll unterstützen, sodass derzeit nur Zugriff auf das TTEthernet-Development-System möglich ist. Das System besteht aus zwei Intel Atom PCs mit einer Beispielanwendung, die das Protokollverhalten anschaulich verdeutlicht. Der Protokollstack für diese PC-basierten TTEthernetsysteme liegt allerdings nur für zwei Typen von Netzwerkkarten vor, sodass der zu entwickelnde Restbussimulator zwingend auf dieser Hardware ausgerichtet sein muss.

Ein Testen des Restbussimulators mit der wenigen momentan vorhandenen Hardware ist ein weiteres Problem, da ein eventuelles Fehlverhalten so schwer entdeckt werden kann. Neue Geräte befinden sich derzeit als Bachelorarbeiten in der Entwicklung, in dem der TTEthernet-Protokollstack auf ARM-basierte Microcontroller implementiert wird. Diese Systeme sollen im Laufe des Sommersemesters in Betrieb genommen werden.

Ein zusätzliches Risiko besteht darin, dass der Anwendungsbereich von TTEthernet noch nicht klar definiert ist. Im Gespräch ist eine Lösung als „Backbone“-Netzwerk, um z.B. den vorderen mit dem hinteren Teil des Automobils zu verbinden oder als zusätzliches Bussystem, das für neue Aufgaben (z.B. hochauflösende Kameraassistenzsysteme) eingesetzt werden soll. Das heißt, ein realistisches Datenmodell der zu übertragenden Daten ist noch nicht vorhanden.

Ein Werkzeug zur Erstellung von TTEthernetnetzwerken fehlt ebenfalls, sodass die Erstellung der XML-Datei aktuell händisch erfolgen muss. Es kann dabei auf eine Beispielformatierung zurückgegriffen werden. Allerdings ist dieses Verfahren eine weitere mögliche Fehlerquelle, da das XML-Schema sehr umfangreich ist. Till Steinbach hat in [Steinbach \(2011\)](#) eine Netzwerkkonfiguration mit sechs Teilnehmern und einem TT-Switch erstellt, um diese zu simulieren. Die dazugehörige Netzwerkkonfiguration umfasst mehr als 35000 Zeilen Code. Das händische Erstellen stellte sich als sehr fehleranfällig heraus.

Das Scheduling aller simulierten Knoten stellt das größte Risiko dieses Themengebietes dar. Es muss ausreichend Rechenleistung vorhanden sein, um auch bei TTEthernet-Schedules im niedrigen Mikrosekundenbereich allen simulierten Knoten die Ausführung zu garantieren. In der Dissertation von Thomas Galla (vgl. [Galla, 1999](#)) ist ein Verfahren beschrieben, dieses zu ermöglichen, dessen Beschreibung allerdings den Rahmen dieser Ausarbeitung übersteigt.



# 5 Zusammenfassung und Ausblick

Abschließend wird in diesem Abschnitt eine Zusammenfassung dieser Ausarbeitung und ein Ausblick auf die anstehende Projekt- und Masterarbeit wiedergegeben.

## 5.1 Zusammenfassung

Die Restbussimulation ist eine kostengünstige Variante zum frühzeitigen Testen im Entwicklungsprozess eines Steuergerätes für Automotiveanwendungen. Der Einsatz dieses Verfahrens ermöglicht es, die Funktionen einer verteilten Anwendung zu Testen, ohne das Gesamtsystem aufzubauen. Durch die hohe Komplexität der Nachrichtensignale in aktuellen Automobilen empfiehlt es sich, die Konfiguration des Simulators zu vereinfachen. Es werden nur die Nachrichten konfiguriert, die tatsächlich vom zu testenden Gerät verarbeitet werden. In Abhängigkeit vom Entwicklungsstatus lassen sich so verschiedene Tests durchführen.

TTEthernet ist eine Echtzeiterweiterung des Netzwerkprotokolls Ethernet, das die Übertragung von zeitkritischen und standard Daten über das gleiche physikalische Medium erlaubt. Zeitkritische Daten werden dabei entweder in einer zeitgesteuerten oder in bandbreiteregulierenden Weise übertragen. Die Verwendung eines zeitlich koordinierten Medienzugriffs für die zeitgesteuerten Daten bringt zusätzliche Funktionalität in Form einer Synchronisierung mit sich. Dieses wird durch ein zwei Wege Verfahren bewerkstelligt, in dem jeder Teilnehmer eine separate Rolle übernimmt.

Für eine Restbussimulation von TTEthernet muss daher das Protokoll vollständig unterstützt werden. Es ist darauf zu achten, dass die Echtzeitfähigkeit von TT- und RC-Nachrichten gewährleistet ist und eine Zeitsynchronisierung ermöglicht wird. Die simulierten Knoten müssen im Schedule der Restbussimulation abgebildet werden und dürfen keine Auswirkungen auf das Senden von zeitkritischem Verkehr haben.

## 5.2 Ausblick

Für das Projekt ist geplant Grundfunktionalitäten einer Restbussimulation unter der Verwendung eines RT-Linux zu implementieren. Es wird dabei auf den vorhandenen Protokollstack zurückgegriffen werden, sodass eine Restbussimulation nur mit spezieller Netzwerkhardware erfolgen kann. Darüber hinaus ist eine Minimierung der aufgelisteten Risiken durch weitere Literaturrecherchen geplant.

Interessante Fragestellungen sind dabei, ob die Verwendung von mehreren Netzwerkkarten sinnvoll oder wie hoch die Skalierbarkeit der simulierten Knoten unter RT-Linux ist; das heißt wie viele Knoten können abgebildet werden.

Als Idee der Masterthesis würde sich deshalb *„Skalierbare Restbussimulation von Time-Triggered Ethernet durch Verwendung des RT-Linux Kernel im Automotive Kontext“* als Thematik anbieten.

# Literaturverzeichnis

- [Aeronautical Radio Incorporated 2002] AERONAUTICAL RADIO INCORPORATED: Aircraft Data Network / ARINC. Annapolis, Maryland, 2002 (664). – Standard
- [Bartols 2010] BARTOLS, Florian: *Leistungsmessung von Time-Triggered Ethernet Komponenten unter harten Echtzeitbedingungen mithilfe modifizierter Linux-Treiber*. Hamburg, HAW Hamburg, Bachelorthesis, Juli 2010. – Bachelorthesis
- [Bartols u. a. 2011] BARTOLS, Florian ; STEINBACH, Till ; KORF, Franz ; SCHMIDT, Thomas C.: Performance Analysis of Time-Triggered Ether-Networks Using Off-The-Shelf-Components. In: *14th IEEE International Symposium on Object/Component/Service-oriented Real-time Distributed Computing*, 2011. – to appear
- [Eberspächer ] EBERSPÄCHER: *FlexXCon - Restbussimulation*. Eberspächer. – URL [http://www.eberspaecher.com/servlet/PB/menu/1064600\\_11/index.html](http://www.eberspaecher.com/servlet/PB/menu/1064600_11/index.html). – Zugriffsdatum: 2010-12-28
- [EtherCAT Technology Group ] ETHERCAT TECHNOLOGY GROUP: *EtherCAT*. – URL <http://www.ethercat.org>
- [FIBEX Expert Group 2009] FIBEX EXPERT GROUP: Data Model for ECU Network Systems (Field Bus Data Exchange Format) / ASAM. Januar 2009 (3.1). – Specification
- [FlexRay Consortium ] FLEXRAY CONSORTIUM: *FlexRay*. – URL <http://flexray.com/>. – Zugriffsdatum: 2011-02-07
- [Galla 1999] GALLA, Thomas M.: *Cluster Simulation in Time-Triggered Real-Time Systems*. Wien, TU Wien, Dissertation, Dezember 1999
- [Galla und Pallierer 1999] GALLA, Thomas M. ; PALLIERER, Roman: Cluster simulation-support for distributed development of hard real-time systems using TDMA-based communication. In: *Proceedings of the 11th Euromicro Conference on Real-Time Systems, 1999.*, Juni 1999, S. 150–157
- [Honeywell International ] HONEYWELL INTERNATIONAL: . – URL <http://www.honeywell.com>. – Zugriffsdatum: 2011-01-23

- [IABG 2009] IABG: *Das V-Modell XT Version 1.3*. IABG. Februar 2009. – URL <http://v-modell.iabg.de/dmdocuments/V-Modell-XT-Gesamt-Deutsch-V1.3.pdf>. – Zugriffsdatum: 2011-02-07
- [LIN-Administration ] LIN-ADMINISTRATION: *Local Interconnect Network*. – URL <http://www.lin-subbus.org/>. – Zugriffsdatum: 2011-01-06
- [Meier 2008] MEIER, Erich: *V-Modelle in Automotive-Projekten*. Automobil Elektronik. Februar 2008. – URL <http://www.methodpark.de/ressourcen/download/v-modelle-in-automotive-projekten/open-download/>. – Zugriffsdatum: 2011-02-09
- [MOST Cooperation ] MOST COOPERATION: *Media Oriented Systems Transport*. – URL <http://www.mostcooperation.com/>. – Zugriffsdatum: 2011-01-06
- [Navet u. a. 2005] NAVET, Nicolas ; SONG, Yeqiong ; SIMONOT-LION, Françoise ; WILWERT, Cédric: Trends in Automotive Communication Systems. In: *Proceedings of the IEEE* 93 (2005), Juni, Nr. 6, S. 1204–1223. – ISSN 0018-9219
- [PROFIBUS & PROFINET International ] PROFIBUS & PROFINET INTERNATIONAL: *Profinet*. – URL <http://www.profibus.com/technology/profinet>. – Zugriffsdatum: 2011-02-07
- [Reif 2009] REIF, Konrad: *Automobilelektronik*. Wiesbaden : Vieweg und Teubner, 2009. – ISBN 978-3-8348-0446-4
- [Riegraf u. a. 2007] RIEGRAF, Thomas ; BEHH, Siegfried ; KRAUS, Stefan: *Effizientes Testen in der Automobilelektronik - Von der Simulation bis zur Diagnose*. Vector Informatik. August 2007. – URL [http://www.vector.com/portal/medien/cmc/press/PND/Testen\\_ATZ\\_200708\\_PressArticle\\_DE.pdf](http://www.vector.com/portal/medien/cmc/press/PND/Testen_ATZ_200708_PressArticle_DE.pdf)
- [Robert Bosch GmbH ] ROBERT BOSCH GMBH: *Controller Area Network*. – URL <http://www.semiconductors.bosch.de/>. – Zugriffsdatum: 2011-02-03
- [Schäuffele und Zurawka 2010] SCHÄUFFELE, Jörg ; ZURAWKA, Thomas: *Automotive Software Engineering*. Wiesbaden : Vieweg und Teubner, 2010. – ISBN 978-3-8348-0364-1
- [Schlager 2008] SCHLAGER, Martin: *Hardware-in-the-Loop Simulation*. Vdm Verlag Dr. Müller, April 2008. – ISBN 978-3-8364-6216-7
- [Schleißheimer GmbH ] SCHLEISSHEIMER GMBH: *CanEasy*. Schleißheimer GmbH. – URL <http://www.schleissheimer.de/produkte/caneasy/>. – Zugriffsdatum: 2010-12-28

- [Steinbach 2011] STEINBACH, Till: *Echtzeit-Ethernet für Anwendungen im Automobil: Metriken und deren simulationsbasierte Evaluierung am Beispiel von TTEthernet*. Hamburg, Hochschule für Angewandte Wissenschaften Hamburg, Masterthesis, Februar 2011. – Eingereicht zur Erlangung des Master-Grades
- [Steinbach u. a. 2011] STEINBACH, Till ; DIEUMO KENFACK, Hermand ; KORF, Franz ; SCHMIDT, Thomas: An Extension of the OMNeT++ INET Framework for Simulating Real-time Ethernet with High Accuracy. In: *Proceedings of the 4th International ICST Conference on Simulation Tools and Techniques*, 2011. – to appear
- [Steinbach u. a. 2010] STEINBACH, Till ; KORF, Franz ; SCHMIDT, Thomas C.: Comparing Time-Triggered Ethernet with FlexRay: 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, Mai 2010, S. 199–202
- [Steiner 2008] STEINER, Wilfried: *TTEthernet Specification*. TTTech Computertechnik AG. November 2008. – URL <http://www.tttech.com>
- [TTTech Computertechnik AG ] TTTech COMPUTERTECHNIK AG: . – URL <http://www.tttech.com>. – Zugriffsdatum: 2011-01-17
- [Vector Informatik ] VECTOR INFORMATIK: *CANoe - Restbussimulation*. Vektor Informatik. – URL [http://www.vector.com/vi\\_canoe\\_de.html](http://www.vector.com/vi_canoe_de.html). – Zugriffsdatum: 2010-12-28
- [Zimmermann und Schmidgall 2011] ZIMMERMANN, Werner ; SCHMIDGALL, Ralf: *Bussysteme in der Fahrzeugtechnik - 4. aktualisierte Auflage*. Vieweg + Teubner, 2011. – ISBN 978-3-8348-0907-0