



Hochschule für Angewandte Wissenschaften Hamburg  
*Hamburg University of Applied Sciences*

# Reusing CAN legacy devices in distributed CAN busses

**Jan Depke**

jan.depke@haw-hamburg.de

30.08.2012

**Seminararbeit: Related Work**

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>2</b>
1.1	Motivation und Problemstellung . . . . .	2
1.2	Struktur . . . . .	3
1.3	Controller Area Network . . . . .	3
1.4	Realtime Ethernet . . . . .	3
<b>2</b>	<b>Related Work</b>	<b>4</b>
2.1	Klassische CAN-Gateways . . . . .	4
2.2	Gateway Design Strategien . . . . .	4
2.2.1	Service Definition . . . . .	4
2.2.2	Concatenation Methods . . . . .	5
2.2.3	Concatenation Invariance . . . . .	5
2.3	CAN Emulation . . . . .	5
2.3.1	Alternating Windows . . . . .	6
2.3.2	ET Services implemented on top of TT Protocol . . . . .	6
2.3.3	TT Services implemented on top of an ET Protocol . . . . .	6
2.4	CAN Scheduling . . . . .	7
2.4.1	Response Time Analysis . . . . .	7
<b>3</b>	<b>Einordnung</b>	<b>8</b>
3.1	Gateway Design . . . . .	8
3.2	Klassische Gateways . . . . .	9
3.3	CAN Emulation . . . . .	10
3.4	Response Time Analyse . . . . .	11
<b>4</b>	<b>Fazit</b>	<b>11</b>
	<b>Literaturverzeichnis</b>	<b>12</b>
	<b>Abkürzungsverzeichnis</b>	<b>14</b>

# 1. Einleitung

In den letzten Jahren nahm die Anzahl der verwendeten Electronic Control Unit (ECU) mit jeder Automobilgeneration deutlich zu. Diese ECU sind häufig mit historisch gewachsenen oder anwendungsspezifischen Bussystemen verbunden. Es ist anzunehmen, daß in den kommenden Jahren der Bedarf an Bussystemen mit a) hoher Bandbreite, b) Echtzeitcharakteristika und c) mitwachsendem physikalischem Bus deutlich steigen wird. Aus diesem Grund erforscht die Core-Gruppe der Hochschule für Angewandte Wissenschaften (HAW) Hamburg ein automotives backbone Bussystem auf Basis von Realtime Ethernet (RTE), welches diesem Bedarf gerecht werden kann.

## 1.1 Motivation und Problemstellung

Der Generationsübergang zwischen zwei Bustechnologien im Automobilbereich wird durch folgende Aspekte erschwert: a) Zertifizierungskosten und -aufwand b) Aufrüstung / Neuentwicklung etablierter ECU Einheiten und c) Implementierungsaufwand der neuen Technologie. Original Equipment Manufacturer (OEM) in der Automobilindustrie stellen einen Großteil der ECU mit einer Controller Area Network (CAN) - Schnittstelle zur Verfügung, da entsprechende ECU preiswert zu produzieren und einfach zu implementieren sind und darüber hinaus als Industriestandard gelten. Es ist wünschenswert, diese CAN-ECU in das backbone Automobilnetzwerk auf Basis von RTE integrieren zu können. Diese Integration soll transparent und flexibel sein, so daß die vorherige Verkabelungsstruktur und Topologie im Automobil nicht beibehalten werden muß und bestehende CAN-Applikationen in den ECU ohne Anpassung weiterverwendet werden können. Dadurch kann ein CAN-Bus verwirklicht werden, der aus physikalisch nicht direkt verbundenen Geräten besteht (siehe Abbildung 1.1). Darüber hinaus ist eine Interbuskommunikation erstrebenswert. Um

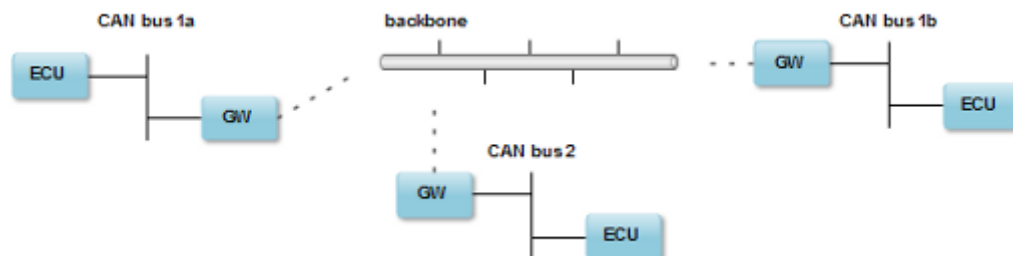


Abbildung 1.1: distributed CAN bus

CAN in RTE transparent einbinden zu können, muß eine CAN-Infrastruktur auf Basis des RTE zur Verfügung gestellt werden. Diese Infrastruktur muß abwärtskompatibel zu CAN sein und darf neue Funktionalität zu CAN hinzufügen. Die Verbindung zwischen RTE und CAN wird durch Gateways realisiert, welche in den angebundenen Bussen als standardkonforme Knoten agieren.

## 1.2 Struktur

Für die Implementierung eines derartigen Gateways sind diverse konzeptionelle Ansätze denkbar, im Folgenden werden die Ansätze aus sachverwandter Fachliteratur untersucht und ihre Anwendbarkeit anschließend bewertet. In Abschnitt 1.4 wird unter die Technologie RTE mit den für diese Ausarbeitung relevanten Eigenschaften vorgestellt. Der Abschnitt 2.1 zeigt auf, welche Ansätze zur Nutzung eines Tunnels zwischen CAN-Bussen über ein Drittmedium in der aktuellen Literatur diskutiert werden. Abschnitt 2.2 analysiert eine strukturierte Herangehensweise an die Entwicklung von Netzwerkgateways, Abschnitt 2.3 vertieft die Möglichkeiten der CAN-Busverbindung anhand eines Ansatzes zur Emulation von CAN auf Drittmedien. Abschließend wird in Abschnitt 2.4 das Scheduling und die Berechnung der Worst Case Response Time (WCRT) untersucht.

## 1.3 Controller Area Network

CAN ist ein serielles, halbduplex Multi-Master Bussystem, welches von 1983 an von der Robert Bosch GmbH entwickelt und 1993 im ISO-Standard 11898:1993 festgeschrieben wurde. CAN-Busse sind nachrichteorientiert, ordnen übertragene Nachrichten also genau einem Sender zu. CAN-Nachrichten haben eine ID, welche gleichzeitig die Priorität der Nachricht angibt. Die Buszugriffskontrolle erfolgt per Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA), die Kollisionsvermeidung und Priorisierung übernimmt das CAN-eigene Arbitrierungsverfahren. Während der Arbitrierung werden die IDs der sendefähigen Nachrichten auf den Bus geschrieben wobei eine logische „0“ der IDs aufgrund der open collector Implementierung des Busses eine logische „1“ überschreibt. Dadurch erkennt jeder Busteilnehmer, ob eine Nachricht mit kleinerer ID und somit höherer Priorität zur Übertragung ansteht und seine eigene Übertragung nötigenfalls einstellen.

## 1.4 Realtime Ethernet

Im Kontext der Core-Gruppe der HAW Hamburg wird eine switchbasierte Realtime Ethernet Implementierung der Firma TTTech verwendet (im Folgenden: Realtime Ethernet, RTE). Realtime Ethernet stellt drei Trafficklassen zur Verfügung: Time Triggered (TT), Rate Constraint (RT) und Best Effort (BE). TT Nachrichten erfüllen harte Echtzeitanforderungen und werden zu vordefinierten Zeitpunkten übertragen, RT-Übertragungskanäle garantieren eine wählbare Bandbreite und erfüllen weniger harte Echtzeitanforderungen. BE-Nachrichten werden entsprechend der bekannten Ethernetstandardgruppe IEEE 802 zeitlich-nichtdeterministisch übertragen.

## 2. Related Work

### 2.1 Klassische CAN-Gateways

Die etablierten Ansätze, CAN-Busse und Ethernetnetzwerke zu verbinden, erschöpfen sich schwach unterschiedlich implementierten CAN-over-IP Tunneln [JKR09]. Ausarbeitungen zur Umkehrrichtung IP-over-CAN existieren [DBKA03], sind aber von beschränkter Relevanz im Sinne der Problemstellung und können nicht ohne Anpassung der CAN-Busteilnehmer umgesetzt werden.

Exemplarisch für die Klasse der CAN-over-IP Tunnel steht die Ausarbeitung [JKR09], welche das Ausschleusen von Diagnoseinformationen aus Automobil-CAN-Bussen über das Internet untersucht: Die Entwicklung eines CAN-over-IP Tunnel muß hinsichtlich Designentscheidungen wie die Wahl der des Protokolls (UDP, TCP, RTP), dem Anwendungszweck (monitoring vs message injection) und dem Hardwareumfeld (CAN vs Echtzeit-CAN) durchdacht werden. Für CAN-over-IP ohne Echtzeitanforderungen existieren diverse Gerätetreiber für Linux-Betriebssysteme sowie Hardwaregateways mit diversen Schnittstellen (USB, Ethernet, RS232). Grundlegend werden CAN-Nachrichten per Encapsulation in IP-Paketen übertragen. Hierfür werden die CAN-Nachrichten in anwendungsspezifische Headerinformationen (Message-ID, Bus-ID, Nutzlastlänge u.Ä.) eingehüllt, welche unter Umständen noch um Sequenznummern erweitert werden müssen, wenn das Übertragungsprotokoll verbindungslos und ohne Fehlererkennung / -korrektur implementiert ist (z.B. User Datagram Protocol (UDP)). Derartige Implementierungen sind nicht für Echtzeitanforderungen geeignet, da der physikalische Übertragungskanal teilweise drastische Latenzen und Jitter aufweist (z.B. General Packet Radio Service (GRPS)).

### 2.2 Gateway Design Strategien

Der strukturierte Entwurf von Gateways in inhomogenen Netzwerken anhand des Open Systems Interconnection Reference Model (OSI-RM) der International Organization for Standardization (ISO) wird in [vBMM90] diskutiert. Bochmann und Mondain-Mondal gliedern die Methodik zur Netzwerkverbindung in die Bereiche a) „Service Definition“, b) „Concatenation Methods“ und c) „Concatenation Invariance“. Dabei liegt Hauptaugenmerk auf den Service Primitives (SP) der einzelnen Schichten des OSI-RM, welche jeweils mindestens Provider einer Request-API für Usern übergeordneter Schichten und Provider einer Indication-API für Usern tiefergelegener Schichten sind.

#### 2.2.1 Service Definition

Prinzipielle Grundüberlegung ist dabei, ob die zu verbindenden Netzwerke über eine kompatible Menge an Service Primitives (SP) auf den korrespondierenden OSI-RM-Schichten

verfügen: Im einfachsten Falle ist ein Gateway nur eine Zuordnungsinstanz, die SP eines Netzwerkes auf SP des anderen Netzwerkes abbildet. Bei fehlender Kompatibilität muß im Gateway eine transparente Erweiterung der Services stattfinden, so daß Funktionsumfangsunterschiede zwischen den SP der Netzwerke ausgeglichen werden können.

Um die Kompatibilität einzelner SP zu bewerten, muß ein Service definiert werden. Eine Definition besteht mindestens aus den Bestandteilen „User-Provider-Interface“, „User-Provider-Interactions“ und „Temporal Ordering“. Im Falle des CAN können diese Service Definitionen der ISO 11898 Spezifikation entnommen werden.

## 2.2.2 Concatenation Methods

Bochmann schlägt drei unterschiedliche Verbindungsmethoden für Services vor, nämlich a) „Direct Concatenation“, b) „Interface Adaption“ und c) „Service Adaption“. Die direkte Verbindung („Direct Concatenation“) von Services folgt der Grundannahme, daß innerhalb einer OSI-RM-Schicht ohne Anpassungenkorrespondierende Request- und Indication-Services existieren.

Die „Interface Adaption“ erfolgt immer dann, wenn eine direkte Verbindung nicht möglich ist, aber beide Schnittstellen semantisch identische Services anbieten, somit nur eine syntaktische Transformation vorgenommen werden muß. In diesem Fall würde das Gateway eine Vermittlungsinstanz darstellen, welche die Interfaces angeschlossener Schnittstellen nutzt und zwischen den Schnittstellen die transportierten Daten gemäß der API der SP in Übertragungsrichtung anpaßt.

„Service Adaption“ gewinnt dann an Bedeutung, wenn die zu verbindenden Schnittstellen eine gemeinsame Teilmenge von SP haben und die Schnittstelle mit dem schwächer ausgeprägtem Funktionsumfang so erweitert werden kann, daß die fehlende Funktionalität durch eine Zwischenschicht in der Schnittstelle hinzugefügt werden kann. Dadurch läßt sich wieder eine Kongruenz der Merkmale der SP erreichen, allerdings sind hierfür nicht nur Modifikation im Gateway notwendig, auch die Geräteschnittstellen des mit schwächer ausgeprägten SP versehenen Busses müssen mit Hilfe der Zwischenschichten erweitert werden.

## 2.2.3 Concatenation Invariance

Als wünschenswerte Verbindungsinvarianz im Sinne von v.Bochmann ist die semantische Unveränderlichkeit einer globalen Eigenschaft eines Kommunikationsdienstes anzusehen, sobald dieser Kommunikationsdienst sich über mehrere, miteinander verbundene Netzwerke erstreckt. Bezogen auf die Problemstellung dieser Ausarbeitung wäre es wünschenswert, wenn CAN-SP auch bei einer Übertragung über ein RTE-Zwischenmedium keine semantische Neudeutung erfahren würden.

## 2.3 CAN Emulation

Sobald eventbasierte CAN-Kommunikation über ein TT-Kommunikationsmedium übertragen werden soll, stellt sich die Frage, wie sich Event Triggered (ET) CAN-Nachrichten in ein Netzwerk mit definierten Übertragungszeitpunkten und -dauern (TT) integrieren läßt. Der Symposiumbeitrag „CAN emulation in a time-triggered environment“ von Roman Obermaisser [Obe02] untersucht die Möglichkeiten, diese Fragestellung zu lösen und

dabei die CAN-Anwendungen von den technischen Mechanismen des TT-Kommunikationsmediums profitieren zu lassen.

Obermaisser sieht drei unterschiedliche Lösungskonzepte, nämlich a) „Alternating Windows“, b) „ET Services implemented on top of TT Protocol“ und c) „TT Services implemented on top of an ET Protocol“:

### **2.3.1 Alternating Windows**

Der „Alternating Windows“-Ansatz ist durch Implementierungen wie „Flexray“ [Con10] hinlänglich erprobt, hierbei wird neben den Zeitfenstern für TT-Kommunikation ein weiteres Zeitfenster für ET-Kommunikation bereitgestellt. Die TT-Zeitfenster unterliegen einer Time Division Multiple Access (TDMA) Zugriffskontrolle, innerhalb des ET-Zeitfensters dürfen Busteilnehmer auf Basis eines zweiten, dynamischen Zugriffsverfahrens an der Kommunikation teilnehmen.

Wesentliche Herausforderungen hierbei sind die Kollisionsvermeidung im ET-Zeitfenster, welche CAN über das eigene Arbitrierungsverfahren (CSMA/CA) lösen würde. Darüber hinaus werden hier zwei Zugriffskontrollprotokolle verwendet, was neben erhöhtem Implementierungsaufwand ebenso zur Notwendigkeit eines Kontrollmechanismus führt, der die Einhaltung der zeitlichen Grenzen der TT- und ET-Zeitfenster garantiert. Nach Obermaisser ist obendrein gen Ende des ET-Zeitfensters eine Sendepause der Dauer der maximalen Übertragungszeit der spezifizierten CAN-Nachrichten notwendig, um eine Kollision mit dem Start des folgenden TT-Zeitfensters zu vermeiden.

### **2.3.2 ET Services implemented on top of TT Protocol**

Um ET-Kommunikation über ein TT-Medium zu ermöglichen, kann ein Busteilnehmer dem ihm zugeordneten TT-Zeitschlitz ebenfalls für ET-Kommunikation nutzen. Hierbei darf die ET-Kommunikation die zeitlichen Grenzen des TT-Zeitschlitz nicht verletzen, weshalb eine Softwareschicht zwischen ET-Applikation und TT-Medium ggf. eine Fragmentierung ausgehender bzw. Defragmentierung eingehender ET-Nachrichten auf dem TT-Kanal vornehmen muß. Diese Softwareschicht ist ebenso verantwortlich für die Wahl der Nachrichtensendeordnung und würde somit im Falle von CAN das Zugriffsverfahren CSMA/CA ersetzen. Dieser Ansatz löst mehrere Probleme des „Alternating Windows“-Ansatzes, indem z.B. die De-/Fragmentierung der ET-Nachrichten eine volle Auslastung des TT-Zeitschlitzes ohne Sendepause ermöglichen würde, nur ein Zugriffsprotokoll auf dem Übertragungsmedium nötig ist und die Softwareschicht weitere Verwaltungsfunktionen zur Verfügung stellen kann.

### **2.3.3 TT Services implemented on top of an ET Protocol**

Die Implementierung von TT-Diensten auf Basis eines zugrundeliegenden ET-Protokolls wird z.B. von Time Triggered CAN (TTCAN) genutzt. Die ET-Applikation erlauben in diesem Szenario das Senden oder Empfangen von TT-Nachrichten über das ET-Medium nur in vordefinierten Zeitfenstern, die restliche Bandbreite wird für ET-Kommunikation genutzt. Hierbei muß besondere Rücksicht auf die nötige Kollisionskontrolle und -auflösung zwischen ET- und TT-Nachrichten genommen werden. Innerhalb der TT-Zeitfenster erweisen sich im Falle wohldefinierten Busteilnehmerverhaltens die Mechanismen zur Arbitrierung und Kollisionsauflösung als überflüssig - im Fehlerfall hingegen kann ein einzelner Busteilnehmer durch zyklischen, ungewollten Arbitrierungsgewinn das gesamte Kommunikationsmedium blockieren.

## 2.4 CAN Scheduling

In automotive-CAN-Bussen wird ein großer Anteil der Nachrichten unter Echtzeitrestriktionen versendet. Somit ist es notwendig, das Zusammenspiel der Nachrichten der unterschiedlichen Busteilnehmer hinsichtlich der Einhaltung der Zeitgrenzen zu untersuchen. Für Nachrichten muß daher der Wiederholungszyklus (Periode), die Priorität, Nachrichtenlänge und der späteste, noch tolerierte Empfangszeitpunkt (Deadline) bekannt sein. Zusammen bilden derartige Daten aller definierten Nachrichten einen CAN-Schedule. Ein CAN-Schedule ermöglicht die Durchführung einer Worst Case Response Time (WCRT)-Analyse für jede definierte Nachricht. Sobald jede Instanz einer Nachricht eine WCRT aufweist, die kleiner ist als die definierte Deadline und gleichzeitig die Busauslastung kleiner 1 ist, werden die Echtzeitanforderungen nicht verletzt und der CAN-Schedule ist validiert. Das 1994 und 1995 von Tindell et al in [TB94], [TBW94], [THW94] vorgestellte Modell zur Response Time Analyse per Berechnung der WCRT fand schnell Akzeptanz bei Automobilherstellern (Volvo Car Corporation) und Entwicklern kommerzieller Scheduling-Analysesoftware (Volcano Communications Technologies) und wurde ab 1995 produktiv eingesetzt.

### 2.4.1 Response Time Analysis

In [DBBL07] stellen Davis et al eine Weiterentwicklung des Modells von Tindell et al vor, die eine Unzulänglichkeit behebt, welche optimistische WCRT produzieren würde. Die Unzulänglichkeit bestand darin, daß in Sequenzen von direkt aufeinanderfolgenden Nachrichten die Auswirkungen von Verzögerungen auf nachfolgende Nachrichten nicht hinreichend Einfluß auf die Berechnung der WCRT nahmen. Davis et al stellen vollumfängliche Auswertungsvorschriften zur Verfügung, anhand derer eine Menge von Nachrichten auf Einhaltung der Zeitgarantien hin überprüft werden kann und beachten dabei drei wesentliche Einflußgrößen auf die WCRT:

#### Queuing Jitter

Das die CAN-Kommunikation auslösende Ereignis tritt üblicherweise in einem Subsystem des Busteilnehmers auf. Dieses Ereignis kann dann via Interrupt, Polling oder vergleichbarer Mechanismen ausgewertet werden und eine CAN-Nachricht wird in die Queue des CAN-Buscontrollers übergeben. Die Standardabweichung der Verzögerung zwischen dem Auftreten des Ereignisses und dem Zeitpunkt, an dem die Nachricht dem Buscontroller zur Verfügung steht, ist der Queuing Jitter.

#### Transmission Time

Die Transmission Time ist die maximale Übertragungsdauer einer CAN-Nachricht und basiert auf der maximalen Nachrichtenlänge. Hierauf hat der Mechanismus „Bit Stuffing“ des CAN Wirkung: Der CAN ISO 11898 Standard definiert sechs aufeinander folgende, identische Bits als Fehlernachricht und fügt daher in fehlerfreier Kommunikation nach fünf identischen Bits ein invertiertes Bit ein. Unter Annahme einer Nachrichtennutzlast, welche maximal viele eingefügte, invertierte Bits aufgrund Bit Stuffing erfordert wird die Transmission Time für jede Nachricht errechnet.



## Queuing Delay

**Blocking / Interference** Sobald eine Nachricht im Ausgabepuffer des CAN-Buscontrollers zur Verfügung steht können drei unterschiedlichen Situationen vorliegen: 1.) Der Bus ist durch eine niedriger priorisierte Nachricht, welche sich bereits in der Übermittlung befindet, blockiert (Blocking), oder 2.) die Nachricht wird durch andere, höher priorisierte Nachrichten verdrängt (Interference). Die Buscontroller kann 3.) ebenso die Arbitrierung der Nachricht erfolgreich abschließen, da Situation 1.) und 2.) nicht vorliegen.

**Priority Level-m Busy Period** Davis et al definieren die Priority Level-m Busy Period basierend auf Arbeiten von Lehoczy [GMKL91], [Leh90] neu: Die Periode beginnt zu einem Zeitpunkt, an eine Nachricht der Priorität  $m$  oder höher zur Arbitrierung bereit steht und gleichzeitig keine weitere, vorher ausgelöste Nachricht selber oder höherer Priorität zur Arbitrierung bereitsteht. Die Periode endet, sobald der CAN-Bus für eine weitere Arbitrierung frei ist und gleichzeitig keine keine weitere, vorher ausgelöste Nachricht der Priorität  $m$  oder höher zur Arbitrierung bereitsteht. Innerhalb der zeitlich zusammenhängenden Periode können keine Nachrichten einer Priorität kleiner  $m$  in die Arbitrierung übergehen.

Somit ist die Priority Level-m Busy Period die Menge von Nachrichten, die - ausgehend von einem wählbaren Zeitpunkt - durch aus den Mengenelementen resultierenden Interferenzen beeinflusst werden kann. Zur Untersuchung des Interferenzanteils der WCRT muß dadurch optimalerweise nur eine endliche Sequenz an Nachrichten betrachtet werden. Bei mehrfacher Übermittlung einer Nachricht innerhalb einer solchen Sequenz gilt das Maximum der einzelnen Übertragungszeiten als WCRT der Nachricht.

# 3. Einordnung

## 3.1 Gateway Design

Die von Bochmann und Mondain-Mondal ausgearbeiteten Designstrategien eignen sich nur bedingt zur Implementierung eines RTE-CAN-Gateways. Während die Echtzeitdatenübertragung in RTE auf Schicht 4 des OSI-RM stattfinden, spezifiziert die CAN ISO 11898 nur die Schichten 1 und 2 des OSI-RM. Es würde nach Bochmann und Mondain-Mondal also naheliegen, die SP der beiden CAN-Schichten auf SP der entsprechenden Schichten des RTE abzubilden.

Eine „Interface Adaption“ kann nicht durchgeführt werden, da RTE und CAN keine semantisch identischen Schnittstellen zwischen SP-User und SP-Provider aufweisen.

Die ebenfalls vorgestellte „Service Adaption“ wäre rein konzeptionell eine ebenso naheliegende Entwicklungsstrategie für den Entwurf eines RTE-CAN-Gateways. In Abhängigkeit der Entwurfsichtweise müßten hierfür die vom RTE bzw vom CAN bereitgestellten SP erweitert werden und an die entsprechende Kommunikationsgegenstelle angepaßt werden, was zumindest einen Eingriff in die Firmware der Hardware erfordert. Derartige Eingriffe

sind aufgrund fehlender Entwicklungshoheit im Kontext RTE nicht möglich, im Falle des CAN widersprechen derartige Eingriffe der Problemstellung, nämlich der Wiederverwendbarkeit existierender CAN-ECU.

Sollten zukünftige Anforderungen ebenso eine bidirektionale auslösbare Kommunikation zwischen CAN-Knoten und RTE-Knoten durch Modifikation von CAN-ECU vorsehen, wäre eine „Service Adaption“ auf Basis von IP-over-CAN ([DBKA03]) naheliegend, würde aber durch die parallel zum OSI-RM implementierten Zeitsynchronisationsmethoden des RTE erschwert.

Die von Bochmann und Mondain-Mondal angestrebte „Concatenation Invariance“ könnte nicht gänzlich erreicht werden. Nach CAN ISO 11898 ist die Aussagegehalt eines gesetzten Acknowledge (ACK)-Bit in einer CAN-Nachricht so definiert, daß ein gesetztes ACK-Bit bestätigt, daß der Empfänger die Nachricht erhalten und an den entsprechenden SP-User weitergereicht hat. Das ACK-Bit hat somit einen „remote acknowledge“-Charakter. Sobald eine CAN-Nachricht gatewaygestützt über ein Zwischenmedium übertragen wird (siehe Abbildung 1.1, Busse 1a und 1b), quittiert das ACK-Bit den Empfang durch das erste Gateway hin zum Zwischenmedium und nicht den Empfang durch den Endempfänger. Dieser semantische Unterschied läßt sich technisch nicht beheben, da das ACK-Bit bereits in die gesendete Nachricht selbst hineingeschrieben wird - eine z.B. verzögerte Bestätigung auf Applikationsebene des tatsächlichen Empfangs der Nachricht durch die Gegenstelle ist somit im Sinne der „Concatenation Invariance“ nicht zweckdienlich und würde zum Bruch der Wiederverwendbarkeitsanforderung der CAN-ECU führen.

## 3.2 Klassische Gateways

Die verfügbaren, kommerziellen CAN-Ethernet Gateways würden ohne Modifikationen im Kontext RTE allerdings nur Daten mittels der Trafficklasse BE übertragen, da diese Gateways den IEEE-Standard 802.3 realisieren und somit kein Echtzeitverhalten implementieren. Somit ist die Entwicklung eines spezialisierten Gateways notwendig, welches je nach Bedarf hinsichtlich der Trafficklasse der Nachrichten konfiguriert werden kann. Ein neu zu entwickelndes Gateway im Sinne der Problemstellung kann die in den kommerziellen IEEE 802.3 Gateways verwendete Encapsulation weiterverwenden.

Eine schlichte Encapsulation und Weiterleitung von CAN-Nachrichten im Sinne der kommerziellen IEEE 802.3 Gateways ist im Sinne der Problemstellung allerdings nicht ausreichend:

Die Inter-CAN-Buskommunikation (siehe Abbildung 1.1, z.B. Kommunikation von Bus 2 zu den Bussen 1a oder 1b) erfordert, daß die Empfänger einer Inter-CAN-Busnachricht bereits zur Busentwurfszeit eine Reaktion auf die Nachricht vorsehen. Die Vermittlung solcher Inter-CAN-Busnachricht wird in aktuellen Automobil-CAN-Bussystemen meist durch ein zentrales, dediziertes Gateway vorgenommen [DBBL07]. Dabei findet bedarfsweise ein Mapping der CAN-Nachrichten des Quellbusses auf andere Nachrichten im Zielbus statt, was einer Änderung der Nachrichtenpriorität gleichkommt. Ein derartiges Mapping wird durch automotiv Netzwerkbeschreibungsstandards wie Fibex (ASAM MCD-2 NET) [ASA11] unterstützt und muß durch ein neuentwickeltes Gateway ebenso umgesetzt werden.

### 3.3 CAN Emulation

Die Ausarbeitung [Obe02] von R. Obermaisser bietet Ansätze zur Implementierung einer CAN-Emulation, die unterschiedliche Anforderungen an ein zu realisierendes Gateway stellen und teilweise mit den in RTE bereitgestellten Trafficklassen (BE, RT, TT) konkurrieren.

**TT Services implemented on top of an ET Protocol** Der Ansatz, einen TT-Service auf Basis eines ET-Protokolls einzurichten, entspräche im RTE-Kontext der Nutzung der BE-Trafficklasse als Übertragungsprotokoll. Das würde eine Softwareschicht in den Gateways erfordern, welche Zeitsynchronisation der Gateways, Pufferung der ET-Nachrichten und den TT-Service selbst implementiert. Dies erscheint nicht zweckmäßig, da RTE bereits die TT-Trafficklasse anbietet, welche diese Aufgaben vollständig übernimmt. Eine Nutzung des ET-Protokolls der BE-Trafficklasse würde somit nur die wesentlichen, nativen Funktionsmerkmale des RTE nachvollziehen.

**Alternating Windows** Da RTE keine native „Alternating Windows“-Lösung anbietet, wäre eine abwechselnde Nutzung der TT- und BE-Trafficklasse denkbar, um dieses Konzept nachzuempfinden. Hier würde das TT-Zeitfenster des „Alternating Windows“-Ansatzes einem Zeitslot der TT-Trafficklasse entsprechen und die Kommunikation des ET-Zeitfensters würde über die BE-Trafficklasse übertragen. Das Gateway würde somit zwischen Echtzeit-CAN-Kommunikation und CAN-Kommunikation ohne Echtzeitanforderungen unterscheiden und diese entsprechend unterschiedlich behandeln.

Darüber hinaus könnte der „Alternating Windows“-Ansatz unter exklusiver Nutzung der BE-Trafficklasse realisiert werden, wobei für das TT-Fenster eine globale Zeitordnung parallel zur bereits im RTE existierenden globalen Zeit der TT-Trafficklasse implementiert werden müßte.

Die Implementierung des „Alternating Windows“-Ansatzes auf Basis der TT-Trafficklasse könnte hingegen auf die globale RTE-Zeit zurückgreifen können - würde allerdings den Kunstgriff vollziehen, CAN-Kommunikation ohne Echtzeitanforderung im ET-Fenster dennoch über einen TT-Kommunikationskanal des RTE zu übermitteln.

Diese drei Implementierungsmöglichkeiten beinhalten entweder einen Bruch mit dem durch RTE vorgegebenen Konzept der Trafficklassentrennung, eine unnötige Neuimplementierung vorhandener Funktionalität oder vollziehen unnötige Kunstgriffe alleine um die Kommunikation in das Korsett des „Alternating Windows“ Ansatzes zu zwingen und erscheinen somit nicht opportun.

**ET Services implemented on top of TT Protocol** Unter der Prämisse, daß in automotive-CAN-Bussen der Anteil der ET-Kommunikation mit Echtzeitanforderungen überwiegt, bietet der Ansatz einen ET-Service auf Basis eines TT-Protokolls zu etablieren vielversprechende Möglichkeiten: Echtzeit-CAN-Kommunikation würde von den Zeitgarantien der TT-Trafficklasse des RTE gewährleistet und die von Obermaisser vorgeschlagene De-/Fragmentierung innerhalb einer Softwareschicht des Gateways kann zur optimalen Ausnutzung eines RTE-TT-Frames durch die deutlich kleineren CAN-Nachrichten genutzt werden. Die vorgenannte Softwareschicht kann ebenso Sicherheits- und Verwaltungsfunktionen implementieren, um z.B. fehlerhafte CAN-Busteilnehmer von der Kommunikation auszuschließen oder Priorisierungen unterschiedlicher Übertragungskanäle zu regeln. Da in diesem Ansatz jedes Gateway als ET-CAN-Busteilnehmer einen dedizierten Zeitschlitz auf dem RTE-Medium zum Versenden von Nachrichten hält, kann somit die Bandbreite

für die Gateways garantiert werden und zusätzlich noch das Sendegateway einer eingehenden ET-Nachricht anhand des Empfangszeitpunktes identifiziert werden.

### 3.4 Response Time Analyse

Um die von einem CAN-Schedule geforderten Zeitgarantien in einem per RTE-Medium verteilten Bussystem validieren zu können, ist die Response Time Analyse nach Davis et al [DBBL07] ein gut geeigneter Ausgangspunkt. In dem durch Abbildung 1.1 beschriebenen Kontext können CAN-businterne Schedules ohne Anpassungen des Modells validiert werden, für inter-CAN-Buskommunikation muß das Modell erweitert werden.

Um die inter-CAN-Buskommunikation zu untersuchen, muß zwischen eingehenden und ausgehenden Nachrichten unterschieden werden. Ein Gateway ist ein vollqualifizierter CAN-Busteilnehmer im Ziel-CAN-Bus und überträgt via RTE empfangene Nachrichten ggf. mit einer geänderten Priorität in den Ziel-CAN-Bus. Der Übertragungsabschnitt im Ziel-CAN-Bus kann somit ohne Modellerweiterung validiert werden. Ebenso verhält es sich mit dem Übertragungsabschnitt im Quell-CAN-Bus vor Erreichen des Gateway hin zum RTE. Somit kann die Sicht bei Planung und Validierung der Übertragung von CAN-Nachrichten über ein RTE-Netzwerk im erweiterten Modell auf die Anpassung der minimalen Deadlines von tatsächlich über ein Gateway gerouteten Nachrichten beschränkt werden. Diese minimale Deadline ist im erweiterten Modell die Summe aus drei Einzelverzögerungen, nämlich der im Quell-CAN-Bus, der durch das Gateway und RTE verursachten Verzögerung und der Verzögerung im Ziel-CAN-Bus. Die durch Gateway und RTE verursachte Verzögerung ist bei Verwendung der TT-Traffic-Klasse des RTE nur dann erschwert zu ermitteln, wenn die unter 2.3 erwähnte Softwareschicht eine De-/Fragmentierung der CAN-Nachrichten vornimmt. Eine Untersuchung der Nachrichtenverzögerung in diesem Fall steht noch aus.

## 4. Fazit

In dieser Ausarbeitung sind die für die Problemstellung relevanten Ausarbeitungen Dritter vorgestellt. Die Untersuchung der vorgestellten Ausarbeitungen hat gezeigt, daß für die Lösung relevanter Teilbereiche der Problemstellung gute und vielversprechende Methodiken bekannt sind, auf denen sich aufbauen ließe. Ebenso wurde ersichtlich, daß die etablierten Lösungsansätze in den Bereichen Gatewaydesign, CAN-Protokolltunnel, Koexistenz von ET- und TT-Kommunikation sowie Response Time Analyse nicht ausreichen um der Problemstellung vollumfänglich zu begegnen. Hauptsächliche Gründe hierfür sind die Aspekte „Weiterverwendung existierender CAN-Applikationen“ und „Verteilung eines existierenden CAN-Bus auf mehrere Teilbusse“ der Problemstellung, welche die anstehende Arbeit von den Arbeiten Dritter abgrenzt.

# Literaturverzeichnis

- [ASA11] ASAM. Asam mcd-2 net 4.0.0, fibex. [http://www.asam.net/nc/home/standards/standard-detail.html?tx\\_rbwmmasamstandards\\_pi1\[showUid\]=1246&start=](http://www.asam.net/nc/home/standards/standard-detail.html?tx_rbwmmasamstandards_pi1[showUid]=1246&start=), 2011.
- [Con10] FlexRay Consortium. Flexray communications system protocol specification version 3.0.1. <http://www.flexray.com>, 2010.
- [DBBL07] Robert Davis, Alan Burns, Reinder Bril, and Johan Lukkien. Controller area network (can) schedulability analysis: Refuted, revisited and revised. *Real-Time Systems*, 35:239–272, 2007. 10.1007/s11241-007-9012-7.
- [DBKA03] Michael Ditze, Reinhard Bernhardt, Guido Kämper, and Peter Altenbernd. Porting the internet protocol to the controller area network. 2nd International Workshop On Real-Time LANs In The Internet Age, July 2003.
- [GMKL91] Michael Gonzalez, Harbour Mark, H. Klein, and John P. Lehoczky. Fixed priority scheduling of periodic tasks with varying execution priority. In *In Proceedings, IEEE Real-Time Systems Symposium*, pages 116–128. IEEE Computer Society Press, 1991.
- [JKR09] Mathias Johanson, Lennart Karlsson, and Tore Risch. Relaying controller area network frames over wireless internetworks for automotive testing applications. *Systems and Networks Communication, International Conference on*, 0:1–5, 2009.
- [Ken12] Hermand Dieumo Kenfack. Designmigrationsstrategien von flexray nach time-triggered ethernet. Technical report, Hochschule für Angewandte Wissenschaften Hamburg, 2012.
- [Leh90] John P. Lehoczky. Fixed priority scheduling of periodic task sets with arbitrary deadlines. In *IEEE Real-Time Systems Symposium '90*, pages 201–213, 1990.
- [Obe02] Roman Obermaisser. Can emulation in a time-triggered environment. In *Proceedings of the IEEE International Symposium on Industrial Electronics (ISIE'02)*, volume 1, pages 270–275, L'Aquila, Italy, 2002. TU Wien, IEEE.
- [TB94] Ken Tindell and Alan Burns. Guaranteeing message latencies on control network (can). In *In Proceedings of the 1st International CAN Conference*, pages 1–2. CiA, 1994.

- [TBW94] K. W. Tindell, A. Burns, and A. J. Wellings. An extendible approach for analyzing fixed priority hard real-time tasks. *Real-Time Systems*, 6:133–151, 1994. 10.1007/BF01088593.
- [THW94] K.W. Tindell, H. Hansson, and A. J. Wellings. Analysing real-time communications: Controller area network (can). In *Real-Time Systems Symposium, 1994., Proceedings*, 1994.
- [vBMM90] Gregor von Bochmann and Pierre Mondain-Monval. Design principles for communication gateways. *IEEE Journal on Selected Areas in Communications*, 8(1):12–21, 1990.

# Abkürzungsverzeichnis

**ACK** Acknowledge

**BE** Best Effort

**CAN** Controller Area Network

**CSMA/CA** Carrier Sense Multiple Access/Collision Avoidance

**ECU** Electronic Control Unit

**ET** Event Triggered

**GRPS** General Packet Radio Service

**HAW** Hochschule für Angewandte Wissenschaften

**ISO** International Organization for Standardization

**OEM** Original Equipment Manufacturer

**OSI-RM** Open Systems Interconnection Reference Model

**RS232** Serielle Schnittstelle nach EIA/TIA-232-F:1997-10

**RT** Rate Constraint

**RTE** Realtime Ethernet

**RTP** Real-Time Transport Protocol

**SP** Service Primitives

**TCP** Transmission Control Protocol

**TDMA** Time Division Multiple Access

**TT** Time Triggered

**TTCAN** Time Triggered CAN

**UDP** User Datagram Protocol

**USB** Universal Serial Bus

**WCRT** Worst Case Response Time