



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Projektbericht - SoSe 09

Till Steinbach

Ethernet als Bus für Echtzeitanwendungen im Automobil

Till Steinbach

Thema der Projektbericht - SoSe 09

Ethernet als Bus für Echtzeitanwendungen im Automobil

Kurzzusammenfassung

Die Anforderungen an die Kommunikation von Fahrzeugsystemen steigen stetig, neue Ansätze für Kommunikationsnetzwerke in Fahrzeugen sind gewünscht. Diese Netzwerke müssen strikten real-time Anforderungen genügen und gleichzeitig flexible Unterstützung für Datenverkehr mit schwächeren Anforderungen bieten. Ein neuer Ansatz für Real-time Kommunikation über Ethernet ist TTEthernet. Dieser Bericht beschreibt die Arbeiten, die während der ersten Projektphase des Masterprojektes im Bereich Bewertung und Erstellung einer Simulationsumgebung durchgeführt wurden. Dieser Bericht richtet sich an Leser, die bereits mit der Thematik vertraut sind. Dokumente für einen Einstieg in die Thematik sind an verschiedenen Stellen referenziert.

Betreuende Prüfer

Prof. Dr. Franz Korf, Prof. Dr. Thomas Schmidt

Betreuer

Till Steinbach

Title of the paper

Using Ethernet for realtime applications in automobiles

Abstract

Inhaltsverzeichnis

| | | |
|----------|--|-----------|
| 1 | Einleitung | 1 |
| 1.1 | Motivation | 1 |
| 1.1.1 | Projektplanung und Zielsetzung | 2 |
| 1.1.1.1 | Vergleich von Time-Triggered Ethernet und FlexRay | 2 |
| 1.1.1.2 | Vorbereitung für die Implementierung einer Time-Triggered Ethernet Simulationsumgebung | 2 |
| 1.1.1.3 | Analyse der TTE Toolchain | 3 |
| 2 | Vergleich von Time-Triggered Ethernet und FlexRay | 4 |
| 2.1 | Protokolle | 4 |
| 2.2 | Komponenten | 5 |
| 2.3 | Vergleich der Technologien | 6 |
| 2.3.1 | Latenz | 6 |
| 2.3.2 | Jitter | 7 |
| 2.3.3 | Bandbreite | 7 |
| 2.4 | Fazit | 8 |
| 3 | Omnet++ als Toolchain für die Netzwerksimulation | 9 |
| 3.1 | Omnet++ | 9 |
| 3.2 | INET-Framework | 10 |
| 3.3 | Nutzung der TTEthernet Tools zur Konfiguration | 11 |
| 3.4 | Einbettung von Applikationen | 11 |
| 3.5 | Herausforderung | 11 |
| 4 | TTEthernet-Toolchain | 12 |
| 4.1 | TTE-Protocol Layer Setup | 12 |
| 4.2 | TTE-Diagnose Werkzeug | 12 |
| 4.3 | TTE-Load | 13 |
| 4.4 | TTE-View | 13 |

| | |
|---|-----------|
| 5 Fazit der Projektarbeit | 14 |
| 5.1 Ausblick auf weiteres Vorgehen | 14 |
| 5.2 Risikoabschätzung zum weiteren Vorgehen | 15 |
| Literaturverzeichnis | 16 |

Kapitel 1

Einleitung

Dieser Bericht stellt die Projektplanung und die Fortschritte des Masterprojektes nach dem ersten Semester vor. Das Masterprojekt ist über zwei Semester organisiert.

1.1 Motivation

Mit dem steigenden Einsatz von elektronischen Systemen im Automobil steigen die Anforderungen an breitbandige Kommunikationsverbindungen zwischen den einzelnen Steuergeräten rapide. Neue Architekturentwürfe im Automotive-Bereich sehen ein Backbone-Netzwerk vor, welches die verteilten Steuerungseinheiten — z.B. sternförmig — miteinander verbindet. Ein solches Backbone-Netzwerk muss starker Last gewachsen sein, während es stets einer vorhersagbaren Übertragungsverzögerung unterliegt (vgl. Steinbach, 2008).

Insbesondere die neue AUTOSAR-Architektur (vgl. AUTOSAR Development Partnership), die eine Entwicklung in atomaren Software-Komponenten vorschreibt, welche frei in der Entwicklungsphase zwischen den verschiedenen Steuergeräten verschiebbar sind, erfordert ein Übertragungsmedium, welches flexibel und vorhersagbar Daten mit verschiedenen Anforderungen an Übertragungszeit und Zuverlässigkeit überträgt.

Ein neuer Ansatz für Fahrzeug-Netzwerke ist die Nutzung von Ethernet. Ethernet hat bereits bewiesen, ein flexibles, hoch skalierbares Netzwerkprotokoll zu sein. Dennoch kann Ethernet nicht die erforderliche Vorhersagbarkeit und Zuverlässigkeit in der Paketweiterleitung und dem Medienzugriff garantieren, welche für eine Echtzeit-Kommunikation erforderlich ist.

Time-Triggered Ethernet ist eine Echtzeit-Ethernet-Erweiterung. Über ein ausfallsicheres Synchronisationsprotokoll werden alle sendenden Teilnehmer synchronisiert. Anschließend operieren alle Teilnehmer nach einem vordefinierten gemeinsamen Zeitplan, welcher das simultane Senden zweier Echtzeit-Nachrichten ausschließt. Damit kann Ethernet auch für echtzeitkritische Kommunikation verwendet werden.

Das Ziel der Projektarbeit ist, zu zeigen, dass ein Fahrzeug-Netzwerk auf Basis von Echtzeit-Ethernet realisierbar ist. Weiterhin gibt es bisher wenig Erfahrung mit der Konsolidierung ver-

schiedener Bussysteme mit unterschiedlichen Anforderungen an Zuverlässigkeit und Zeitverhalten im Auto. Vor einer Prototyp-Implementierung steht daher die Analyse und Simulation eines Echtzeit-Ethernet Fahrzeugnetzwerkes.

Eine Simulation bietet viele Vorteile in verschiedenen Stadien der Entwicklung. Bei der Protokollentwicklung kann eine Simulation dafür genutzt werden, frühzeitig, auch ohne Zugriff auf Hardware, Arbeiten an der Protokollschicht zu überprüfen. Testaufbauten mit großen Topologien, welche normalerweise hohe Hardwarekosten verursachen, können in einer Simulation genauso getestet werden wie der Einfluss von einzelnen Konfigurationsparametern (vgl. Steinbach, 2009).

Beim späteren Einsatz der Technologie kann eine Simulation verwendet werden, um frühzeitig verschiedene Topologien zu testen und zu vergleichen.

Sofern die Simulation sehr akkurat die echte Hardware abbildet, kann eine Simulation darüber hinaus dazu eingesetzt werden, Trainings und Übungen durchzuführen.

1.1.1 Projektplanung und Zielsetzung

Das Masterprojekt ist über zwei Semester angelegt. Das Projektziel ist, eine Bewertungsgrundlage für die Time-Triggered Ethernet Technologie zu schaffen, auf deren Basis weitere Evaluationen aufbauen können. Für das erste Semester sind im Wesentlichen die folgenden Tätigkeiten geplant:

1.1.1.1 Vergleich von Time-Triggered Ethernet und FlexRay

Da es bisher kaum veröffentlichte Arbeiten im Bereich Ethernet im Automotive Einsatz gibt, ist ein Ziel der Projektarbeit ein Vergleich zwischen FlexRay und Time-Triggered Ethernet auf Basis eines mathematischen Modells. Das Ziel dieser Arbeit ist zu zeigen, dass es möglich ist, mindestens die gleiche Menge an Daten mit TTEthernet übertragen zu können, die ein voll ausgelasteter FlexRay-Bus bewältigen kann. Dabei sollen auch weitere Metriken wie Verzögerungen oder Laufzeitunterschiede auf Basis eines logischen Modells betrachtet werden.

1.1.1.2 Vorbereitung für die Implementierung einer Time-Triggered Ethernet Simulationsumgebung

Neben einer ersten Evaluation der konkurrierenden Automotive-Systeme FlexRay und TTEthernet ist die Vorbereitung für die Implementierung der Time-Triggered Ethernet Technologie in einer Simulationsumgebung ein bedeutendes Ziel der Projektarbeit. Diese Vorbereitungen sind notwendig, um in der zweiten Projekthälfte die Implementierung der Simulation umsetzen zu können. Die Vorbereitungen sollen soweit abgeschlossen sein, dass ohne weitere Einarbeitung mit der Implementierung begonnen werden kann.

1.1.1.3 Analyse der TTE Toolchain

Neben der Time-Triggered Ethernet Spezifikation sollen als Vorbereitung auf die Implementierung der Simulation die Werkzeuge der Firma TTTech (vgl. TTTech Computertechnik AG) untersucht werden. Die Software sowie ein Evaluierungssystem der TTEthernet Technologie stehen in Hamburg zur Verfügung. Insbesondere soll geprüft werden, ob es möglich ist, die Werkzeuge später auch für die Konfiguration der Simulationsumgebung einzusetzen.

Kapitel 2

Vergleich von Time-Triggered Ethernet und FlexRay

Der erste Teil der Projektarbeit umfasst eine Vergleichsanalyse zwischen Time-Triggered Ethernet und FlexRay. Für diesen Projektabschnitt wurde ein Paper erstellt (vgl. Steinbach u. a., 2009). Dieser Bericht gibt ausschließlich einen Überblick über das Paper und dessen Ergebnisse wieder.

Der FlexRay-Bus ist aus Sicht der Automobilindustrie der Nachfolger des CAN-Bus (vgl. International Organization for Standardization, 2003). Er soll die Rückgrattechnologie für Fahrzeugnetzwerke bilden. FlexRay ist geeignet für den Einsatz mit harten Echtzeitanforderungen und wird das erste im Automobil eingesetzte Bussystem, welches offline geplanten Echtzeit-Datenverkehr gemeinsam mit zufällig auftretendem event-basierten Datenverkehr erlaubt.

Flexray bietet zwei Kanäle mit jeweils bis zu 10 MBit/s Datenübertragung. Es ist jedoch bereits absehbar, dass zukünftige Automobilsysteme höhere Anforderungen an ihr Kommunikationssystem stellen werden.

2.1 Protokolle

Zunächst wurden die Protokollgrundlagen beider Systeme erarbeitet. Sowohl FlexRay, als auch TTEthernet basieren auf einem Time Division Multiple Access (TDMA) Konzept. Das Grundgerüst der Protokolle basiert auf periodischen Zyklen. Bei FlexRay bestehen diese aus einem statischen und einem dynamischen Segment (Siehe Abbildung 2.1). Das statische und das dynamische Segment haben verschiedene Zugriffskonzepte. Jeder Teilnehmer, der echtzeitkritische Daten versenden muss, erhält gemäß seinen Anforderungen Slots im statischen Segment. Jeder Slot hat eine ID, welche gleichzeitig den Sendezeitpunkt der Nachricht definiert. Dieser Ansatz erfordert eine genaue Synchronisierung zwischen den Teilnehmern. Je genauer die Synchronisation realisiert werden kann, desto mehr Daten können in einem

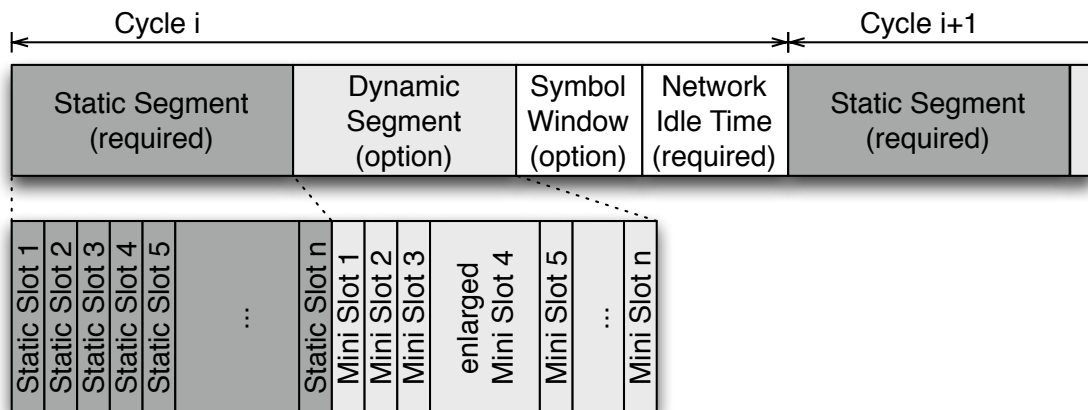


Abbildung 2.1: FlexRay Cycle

Zyklus übertragen werden (vgl. Rausch, 2008). Das Scheduling im statischen Segment ist vergleichbar mit dem Konzept von TT-CAN. Im dynamischen Segment werden Nachrichten prioritätengesteuert übertragen. Das dynamische Segment steht für die Übertragung von event-basierten Daten zur Verfügung, wurde aber für den Vergleich nicht tiefergehend betrachtet.

Das TTEthernet-Protokoll setzt, genau wie FlexRay, bei der Übertragung von Daten mit Echtzeitanforderungen auf ein Verfahren, welches auf Zeitslots aufbaut.

Über ein ausfallsicheres Synchronisationsprotokoll werden alle sendenden Teilnehmer mit einer globalen Zeitbasis versorgt. Anschließend operieren alle Teilnehmer nach einem vordefinierten gemeinsamen Zeitplan, der das simultane Senden zweier Echtzeit-Nachrichten ausschließt. Der Übertragungszeitpunkt wird für jede Nachricht an jeder Stelle zwischen Sender und Empfänger definiert. Auf diese Weise ist die Nachrichtenlaufzeit vorhersagbar, jedoch nicht zu jedem Empfänger gleich.

2.2 Komponenten

Der FlexRay-Standard erlaubt Topologien in Bus, Stern oder gemischter Form. Für Stern-topologien stehen passive und aktive Sterne zur Verfügung. Passive Sterne sind einfache Verbindungen der aufeinandertreffenden Leitungen. Aktive Sterne hingegen empfangen das Signal, bereiten es auf und leiten es auf den Ausgängen des Sterns weiter. Dadurch entsteht eine Verzögerung, die wegen der Synchronisation 250ns nicht übersteigen darf. Wegen dieser Verzögerung schreibt die FlexRay-Spezifikation ein Maximum von zwei aktiven Sternen vor (vgl. Rausch, 2008).

TTEthernet basiert wegen den eingesetzten Switchen auf der Sterntopologie. Dabei können

theoretisch beliebig viele Switche kaskadiert werden. Im Standard-Ethernet gibt es zwei verschiedene Arten von Switchen. *Store-and-forward-Switches* empfangen zunächst den kompletten Frame, bevor sie eine Nachricht weiterleiten. Damit kann bereits im Switch überprüft werden, ob ein Paket vollständig und korrekt ist. Korrupte Pakete werden bereits vor der Weiterleitung im Switch verworfen. Im Gegensatz dazu leiten *Cut-through-Switches* Nachrichten direkt vom Eingang auf den Ausgang weiter, sobald sie den kompletten Ethernet-Header mit der Zieladresse empfangen haben. Damit sind Cut-through-Switches in der Weiterleitung deutlich schneller.

Da TTEthernet-Switches einen eigenen Weiterleitungszeitplan besitzen, werden dort derzeit Store-and-forward Switches eingesetzt. Die Nachrichten werden bis zu Ihrer Weiterleitung im Switch gespeichert. Eine Umsetzung als Cut-through-Switch ist möglich, jedoch komplexer.

2.3 Vergleich der Technologien

Für den Vergleich der Technologien wird ein FlexRay Netzwerk definiert und einem TTEthernet Netzwerk gegenüber gestellt, welches dieselbe Topologie und Datenmenge verwendet. Die Berechnungen basieren auf einem Netz aus zwei aktiven Sternen.

2.3.1 Latenz

Abgesehen von aktiven Sternen ist FlexRay ein durchverkabelter Bus. Damit hängt die Übertragungszeit im Wesentlichen von der Signallaufzeit ab. Für das Beispielsystem wurden je nach Framegröße Übertragungszeiten von $12,2\mu s$ bis $265,2\mu s$ berechnet.

Da TTEthernet eine Technologie mit Switches ist, hängt die Laufzeit hier im Wesentlichen von der Verzögerung im Switch ab. Im Gegensatz zu FlexRay ist hier die nutzbare Bandbreite jedoch nicht abhängig von der Signallaufzeit. Im FlexRay-Bus kann die nächste Nachricht erst gesendet werden, wenn der Bus wieder frei ist. Im TTEthernet kann dagegen die nächste Nachricht bereits gesendet werden, wenn der Link zwischen Sender und Switch frei ist. Damit ist für TTEthernet die Latenz weniger bedeutend als im FlexRay-Bus.

Da die Latenz im TTEthernet nicht gleichförmig ist, muss sie für jeden Pfad zwischen Sender und Empfänger gesondert berechnet werden. Im wesentlichen hängt sie von der Verzögerung im Switch ab, welche wiederum von der Switch-Implementierung und Konfiguration abhängig ist. Hinzu kommt jeweils die Übertragungszeit der Nachricht.

Die Firma TTTech gibt den TTEthernet-Switch mit $1\mu s$ bis $2.4\mu s$ Verzögerung und einer Präzision im niedrigen Mikrosekunden Bereich an. Damit lässt sich für die Beispieltopologie mit zwei Switches eine Nachrichtenlaufzeit von $26\mu s$ bis $266\mu s$, je nach Nachrichtengröße, berechnen.

2.3.2 Jitter

In diesem Bericht wird der Jitter als die Differenz zwischen maximaler und minimaler Übertragungszeit definiert. Dies ist eine relevante Metrik für Echtzeit-Kommunikation.

Bei FlexRay ist die Übertragungszeit im statischen Segment annähernd konstant, da sie ausschließlich von der Signallaufzeit abhängt. Im dynamischen Segment gibt es dagegen Schwankungen, welche sich jedoch nur mit Kenntnis der darüber liegenden Anwendung berechnen lassen.

Bei TTEthernet resultiert der Jitter aus dem Auseinanderdriften des synchronisierten Taktes zwischen Sender, Empfänger und Switch. Die längste Laufzeit zwischen Sender und Empfänger ist gegeben, wenn die Uhr des Senders vorgeht und die des Empfängers nachgeht. Sie kürzeste Zeit ist gegeben, wenn die Zeit des Senders nachgeht und die Zeit des Empfängers vorgeht. TTTech gibt die maximale Taktabweichung als im niedrigen einstelligen Mikrosekundenbereich an. Damit kann der Jitter für TTEthernet als kleiner $10\mu s$ angenommen werden.

2.3.3 Bandbreite

Im Zentrum der Analyse steht die Berechnung der Nettobandbreite für FlexRay und TTEthernet. Es soll dabei gezeigt werden, dass es möglich ist, den kompletten Echtzeitverkehr des FlexRay-Busses über ein TTEthernet-Netzwerk zu übertragen.

Um die mögliche Bandbreite für Echtzeitdaten so gut wie möglich auszunutzen, wurden zunächst das optionale dynamische Segment und das optionale Symbolwindow aus dem Zyklus gestrichen. Die maximale Nettobandbreite wird nur erreicht, wenn jeder Frame mit den maximalen Nutzdaten von 254 Byte ausgenutzt wird. Auf dieser Basis konnte berechnet werden, dass maximal 58 Slots pro Zyklus möglich sind. Dies führte zu einer nutzbaren Bandbreite von 7,366 MBit/s.

Es konnte gezeigt werden, dass ein TTEthernet mit derselben Zykluszeit und Nutzdatengröße nur zu 11% ausgelastet ist, was viel Bandbreite für die Übertragung von event-basierten Nachrichten lässt.

Bei den Berechnungen wurde deutlich, wie groß der Einfluss der Nutzdatengröße auf die nutzbare Bandbreite ist. Grafik 2.2 zeigt die Anzahl möglicher Slots bei verschiedenen Nutzdatengrößen und dazu die Nettobandbreite. Dabei ist deutlich zu sehen, dass die nutzbare Bandbreite von TTEthernet stets höher ist als die von FlexRay. Deutlich höhere Bandbreite bietet TTEthernet jedoch erst bei höherer Nutzdatengröße. Im Bereich unter der minimalen Framegröße von Ethernet (84Byte) werden die Nutzdaten aufgefüllt (padding). Dies zeigt sich für TTEthernet in der Grafik als konstanter Bereich, für Nutzdatengrößen kleiner 46 Byte.

Im Gegensatz zu FlexRay, welches stets als Broadcast-Kommunikation arbeitet, ermöglicht TTEthernet auch Unicast- oder Multicast-Kommunikation. Da dabei Nachrichten, die über disjunkte Links gesendet werden, parallelisiert werden können, sind die dargestellten Band-

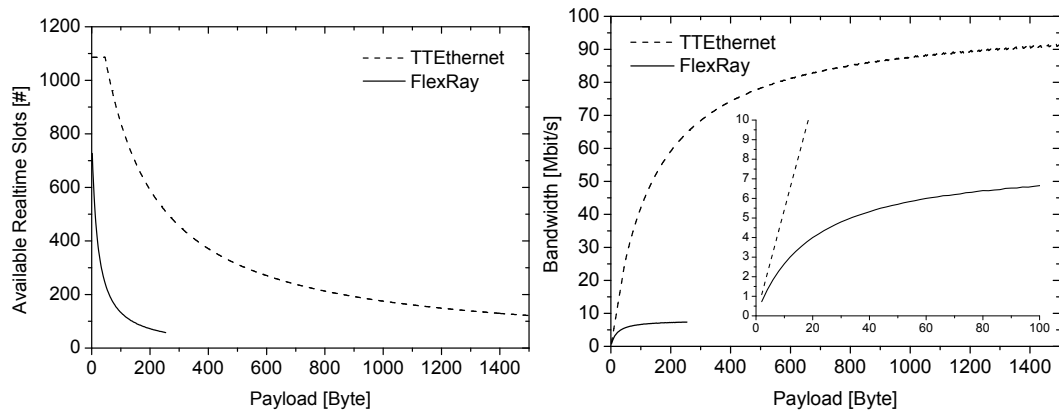


Abbildung 2.2: FlexRay und TTEthernet Slotzahl und netto-Bandwidth bei verschiedenen Nutzdaten-Größen (16ms Zyklus)

breiten als worst-case zu betrachten. Da TTEthernet im Full-Duplex-Modus arbeitet, können also im Idealfall alle Teilnehmer durchgehend kollisionsfrei Senden und Empfangen. Dies ist mit FlexRay nicht möglich.

2.4 Fazit

Die Analyse zeigt, dass es möglich ist, die Daten eines voll ausgelasteten FlexRay-Busses über ein TTEthernet-Netzwerk zu übertragen. Darüber hinaus können weitere event-basierte Nachrichten über denselben Kanal versandt werden, was eine Zusammenfassung und Ersetzung verschiedener Busse ermöglicht.

Es wurde gezeigt, dass der Jitter und die Latenz für eine fahrzeug-spezifische Anwendung dezidiert berechnet werden müssen, dass diese jedoch verglichen mit der Zykluszeit sehr gering sind.

Durch Parallelisierung bei Unicast- und Multicast-Kommunikation kann die Bandbreite des Systems noch effizienter ausgenutzt werden.

Dieser Bericht gibt einen Überblick über die Projektarbeit. Detaillierte Ergebnisse können dem Papier *Comparing Time-Triggered Ethernet with FlexRay: An Evaluation of Competing Approaches to Real-time for In-Vehicle Networks* (vgl. Steinbach u. a., 2009) entnommen werden.

Kapitel 3

Omnet++ als Toolchain für die Netzwerksimulation

Für eine erfolgreiche Umsetzung des Time-Triggered Ethernet-Konzeptes in einem realen automotive System sind verlässliche Analysen und Simulationen unumgänglich. Mit einem leistungsfähigen Simulationswerkzeug kann bereits in einem frühen Stadium eine fahrzeugspezifische Anwendung unter realitätsnahen Bedingungen getestet werden. In Zeiten paralleler Soft- und Hardwareentwicklung ist die Simulation oft die einzige Möglichkeit, erste Integrationstests durchzuführen. Änderungen am Protokoll lassen sich objektiv aufzeichnen und mit früheren Versionen vergleichen. Auch Tests mit umfangreichen Topologien sind ohne große finanzielle Aufwände möglich.

3.1 Omnet++

Omnet++ (vgl. OMNeT++ Community, b) ist ein Simulationsframework für Simulationen auf Basis von diskreten Events. Omnet++ hat sich in der Praxis als eine Standard-Toolchain für die event-basierte Simulation von Netzwerken herausgestellt. Dies hängt damit zusammen, dass die Paketweiterleitung sehr gut in ein event-basiertes System überführt werden kann. Nimmt man den Paketfluss zwischen zwei Teilnehmern als Beispiel, interessiert man sich nur für einige wenige Zeitpunkte zwischen Versand und Empfang der Pakete, wie z.B. das Eintreffen in den Instanzen zwischen den Teilnehmern. Alle weiteren Zeitpunkte dazwischen können in der Simulation vernachlässigt werden. Im Gegensatz zu physikalischen Modellen, in denen ein bedeutender Rückfluss des Zustandes in das System stattfindet und eine große zeitliche Auflösung zu Ungenauigkeiten führt, hat das System in der event-basierten Simulation bedeutend weniger Zustände.

Omnet++ und seine Komponenten sind in C++ entwickelt und wurden darauf ausgerichtet, möglichst effizient und performant zu arbeiten. Damit können auch umfangreiche Topologien auf aktueller Computerhardware in akzeptabler Zeit simuliert werden.

Da die Simulation in einer eigenen Simulationszeit verläuft, müssen die Algorithmen in der Simulation nicht in Echtzeit ausgeführt werden. Somit können die Ergebnisse als zuverlässig angesehen werden, auch wenn das Simulationssystem mit einer Verzögerung arbeitet.

3.2 INET-Framework

Das INET-Framework (vgl. OMNeT++ Community, a) ist ein API für Omnet++, welches die Entwicklung von Simulationen im Bereich der Netzwerkprotokolle vereinfacht. Es besteht aus einer Reihe von Modulen, die die Funktionalität von verschiedenen Elementen vom physikalischen Layer bis zur Applikationsschicht des OSI-Modells zur Verfügung stellt. Für die Simulation von Echtzeit-Ethernet Technologien sind die folgenden Ethernet Module von Bedeutung:

- Ethernet Client App
- Ethernet Server App
- Ethernet MAC
- Ethernet Switch

Das Inet Framework ist modular aufgebaut und verwendet intensiv Vererbung. Daher ist es besonders gut für die Verwendung in einer Simulation von Echtzeit-Ethernet geeignet. Da TTEthernet auf Standard-Ethernet basiert, können große Teile übernommen und um die TTEthernet-Logik erweitert werden. Für die Implementierung von TTEthernet muss sowohl der Ethernet Host, als auch der Ethernet Switch angepasst werden. Weiterhin muss das INET-Framework um das TTE-Protokoll erweitert werden. Damit ist es möglich, Echtzeitverkehr direkt aus einem Trafficgenerator zu speisen.

Das TTE-Protokoll ist ein Ethernet-Protokoll und kann daher als eigener Stack parallel z.B., zu dem IPv4- oder dem IPv6-Stack implementiert werden. Diese können auch als Vorlage verwendet werden. Weiterhin bietet sich als Vorlage zur Implementierung der quelloffene TTEthernet-Treiber der Firma TTTech an.

Der Ethernet Host muss um die Synchronisierung und das Scheduling erweitert werden. Es bietet sich an, die Funktionalität der event-triggered Nachrichten auszulagern und als gesonderte Interface zu behandeln. Damit besteht der logische TTEthernet-Host aus einem Interface für die Time-triggered Kommunikation und einem Interface für die event-basierte Kommunikation.

Der Hauptaufwand der Implementierung liegt in der Umsetzung des TTEthernet-Switches. Hier muss der Ethernet Switch des Inet Frameworks um die TTEthernet-Funktionalität erweitert werden. Leider steht für diese Arbeiten keine Referenzimplementierung als Vorlage zur Verfügung. Somit muss die erforderliche Funktionalität aus der TTEthernet-Spezifikation abgeleitet werden.

3.3 Nutzung der TTEthernet Tools zur Konfiguration

Die Konfiguration des Switches liegt in einem Binärformat vor. Da die Struktur dieses Binärformates bekannt ist, ist es möglich, die TTE-Toolchain auch für die Konfiguration der Simulation zu verwenden. Damit könnte eine Konfiguration mit den TTEthernet Werkzeugen erstellt und in der Simulation getestet werden, bevor sie auf einem echten Switch zum Einsatz kommt.

Da dieses Arbeitspaket keine direkten Vorteile für die Simulation bringt, wird es in der kommenden Implementierung vorerst nicht umgesetzt.

3.4 Einbettung von Applikationen

Ein Langzeitziel kann weiterhin die Integration von Applikationen in die Simulation sein. Hier wäre z.B. denkbar, eine AUTOSAR-Laufzeitumgebung in die Simulation zu integrieren. Damit ließe sich die Kommunikation einer kompletten Fahrzeugfunktionalität, welche in AUTOSAR implementiert und auf verschiedene Steuergeräte verteilt ist, simulieren.

Zu beachten ist dabei, dass in diesem Fall die Simulation nicht mehr in Simulationszeit, sondern ausschließlich in Echtzeit durchgeführt werden kann.

Da AUTOSAR bisher keine Unterstützung für Echtzeit-Ethernet enthält, ist die Realisierung momentan nicht sinnvoll.

3.5 Herausforderung

Um ein genaues Ergebnis zu erhalten, müssen die Zeitanforderungen des späteren Zielsystems möglichst exakt im Simulationsmodell berücksichtigt werden. Nur so ist es möglich, aus der Simulation direkt Rückschlüsse auf die zukünftige reale Situation zu ziehen. Wegen der beschränkten Zeit für das Masterprojekt wird modular implementiert. Da vorerst nicht alle Funktionen des TTEthernet für die Simulation notwendig sind, kann somit Zeit gespart werden, indem nicht alle Funktionen parallel umgesetzt werden.

Kapitel 4

TTEthernet-Toolchain

Die TTEthernet-Toolchain gliedert sich in vier Produkte. *TTE-Protocol Layer Setup* ist ein Werkzeug, um den Zeitplan eines Endsystems zu konfigurieren. *TTE-Diagnose* ist ein Diagnose- und Netzwerkmanagement-Werkzeug. *TTE-Load* lädt die Netzwerkkonfiguration in den TTEthernet-Switch. *TTE-View* ist ein Überwachungs- und Trafficanalysewerkzeug.

4.1 TTE-Protocol Layer Setup

Mit TTE-Protocol Layer Setup werden Konfigurationen für Endsysteme eines Kommunikationsclusters erstellt. Diese Konfigurationen liegen später als C-Files (für jeden Node) vor. Die Applikation ist in Python geschrieben und läuft Plattformunabhängig. Neben globalen Einstellungen wie der Anzahl der Teilnehmer und der Zykluszeit, werden hier alle späteren Time-Triggered-Nachrichten mit Sender und Empfängerliste definiert. Weiterhin kann für jeden Teilnehmer ein genauer Zeitplan festgelegt werden, nach dem z.B. Nachrichten versandt werden.

4.2 TTE-Diagnose Werkzeug

Das TTE-Diagnose Werkzeug hat die Aufgabe, die Diagnoseframes, welche vom Switch gesendet werden, zu analysieren und kann gleichzeitig mit einem Traffic-Generator Standard-Ethernet-Frames erzeugen.

Die Diagnosenachrichten vom Switch enthalten globale Werte wie den Zustand, die Version der Konfiguration oder die Summe der verworfenen Pakete genauso wie Diagnoseinformationen für jeden Port.

Der interne Traffic-Generator erzeugt Frames mit einer frei wählbaren Quell- und Zieladresse, einer wählbaren Größe und einem definierten Abstand zwischen den Nachrichten.

4.3 TTE-Load

TTE-Load lädt die Switchkonfiguration in den TTEthernet-Switch. Die Konfiguration besteht aus einem Binärfile, welches mit Hilfe der *TTE-Build* Software aus einem XML-Dokument gewonnen wird. Die *TTE-Verify* Software kann diese Konfiguration auf Korrektheit gegenüber der Bitlevel-Spezifikation überprüfen.

4.4 TTE-View

TTE-View ist eine Erweiterung für das Wireshark Protokoll-Analyseprogramm. Mit diesem Plugin ist es möglich, aufgezeichneten TTEthernet-Traffic aufzuschlüsseln und in einer übersichtlichen Weise für einen menschlichen Leser aufzubereiten.

Kapitel 5

Fazit der Projektarbeit

In diesem Bericht wurden die Fortschritte der Projektarbeit dokumentiert. In der Projektzeit konnte erfolgreich eine Vergleichsanalyse zwischen dem FlexRay-Bussystem und der Time-Triggered Ethernet-Technologie erstellt werden. Es wurde über mathematische Methodik gezeigt, dass es möglich ist, ein System mit einem vollständig ausgelasteten FlexRay-Bus in ein System, welches auf Time-Triggered Ethernet basiert, zu überführen. Darüber hinaus wurde gezeigt, dass es dabei möglich ist, zusätzlichen Traffic ohne Echtzeitanforderungen zu übertragen und damit mehrere Bussysteme zu kombinieren.

Für die Simulation wurde Omnet++ als Simulationsumgebung ausgewählt. Um im nächsten Schritt direkt mit der Implementierung beginnen zu können, wurde die Simulationsumgebung vorbereitet. Als Framework wurde das INET-Framework für OMNet++ ausgewählt und analysiert. Das INET-Framework bietet bereits weite Teile der Implementierung des OSI-Modells. Relevante Module für die Simulation von Echtzeit-Ethernet sind im Abschnitt 3.2 dokumentiert.

Aus Zeitgründen konnte die TTEthernet-Toolchain bisher nur oberflächlich analysiert werden. Dies wird in Kürze im Zuge des Bring-Ups des Evaluierungssystems nachgeholt.

5.1 Ausblick auf weiteres Vorgehen

In der zweiten Projekthälfte soll TTEthernet für Omnet++ implementiert werden. Zunächst soll das Evaluierungssystem in Betrieb genommen werden, da es als Referenz für die Simulation dienen soll. Es ist geplant, die Implementierung in Module aufzuteilen, welche voneinander unabhängig getestet werden können. Dabei sollen vorerst nur die Teile implementiert werden, welche sich auf TTEthernet beziehen. Der TTEthernet-Switch bietet über das TTEthernet-Protokoll hinaus weitere Funktionalität, welche vorerst nicht benötigt wird. Es ist geplant, zunächst das TTEthernet-Protokoll selbst zu implementieren, und erst in einem späteren Schritt das Synchronisierungskonzept. Da in der Simulation eine globale Zeit zur Verfügung steht, kann die Synchronisierung vorerst emuliert werden.

5.2 Risikoabschätzung zum weiteren Vorgehen

Für den kommenden Projektabschnitt — der Implementierung des TTEthernet-Protokolls in Omnet++ — ist über die Spezifikation hinaus möglicherweise weiteres Material notwendig, um protokollkonform zu arbeiten. Zusammen mit dem Evaluierungssystem liegt das Interface Control Document für den TTEthernet-Switch vor. Bisher konnte dieses Dokument aus Zeitgründen noch nicht vollständig durchgearbeitet werden, so dass eventuelle Unklarheiten erst in der Implementierung auftauchen können. Wegen des guten Kontaktes zu der Firma TTEch ist dieses Risiko als eher gering einzustufen.

Ein weiteres Risiko ist die Ausrichtung der Arbeitsgruppe. Da sich die Arbeitsgruppe momentan vor Gesprächen mit einem potentiellen Partner befindet, besteht aus meiner Sicht die Möglichkeit, dass das Projekt kurzfristig anders ausgerichtet wird und sich somit die Aufgabenstellung für die zweite Projekthälfte ändert. Da die in diesem Projekt beschriebenen Arbeiten in sich komplett sind und auch ohne die Implementierung im zweiten Projektteil für sich stehen, ist diese Situation hinnehmbar. Im Abschnitt 5.1 wurde bereits auf ein weiteres modulares Vorgehen hingewiesen, so dass auch eine Teilimplementierung aussagekräftige Ergebnisse liefern kann.

Literaturverzeichnis

- [AUTOSAR Development Partnership] AUTOSAR DEVELOPMENT PARTNERSHIP: *AUTomotive Open System Architecture*. – URL <http://www.autosar.org/>
- [International Organization for Standardization 2003] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION: Road vehicles - Controller area network (CAN) / ISO. Geneva, Switzerland, 2003 (11898). – ISO
- [OMNeT++ Community a] OMNET++ COMMUNITY: *INET Framework for OMNeT++ 4.0*. – URL <http://inet.omnetpp.org/>
- [OMNeT++ Community b] OMNET++ COMMUNITY: *OMNeT++ 4.0*. – URL <http://www.omnetpp.org>
- [Rausch 2008] RAUSCH, Mathias: *FlexRay: Grundlagen, Funktionsweise, Anwendung*. München : Carl Hanser Verlag, 2008. – ISBN 978-3-446-41249-1
- [Steinbach 2008] STEINBACH, Till: *Ethernet als Bus für Echtzeitanwendungen im Automobil*. Dec 2008. – URL <http://papers.till-steinbach.de/s-ebea-08a.pdf>. – Bericht
- [Steinbach 2009] STEINBACH, Till: *Time-Triggered Ethernet in Fahrzeugnetzwerken - Related Work*. Jun 2009. – URL <http://papers.till-steinbach.de/s-ttefr-09a.pdf>. – Bericht
- [Steinbach u. a. 2009] STEINBACH, Till ; KORF, Franz ; SCHMIDT, Thomas: *Comparing Time-Triggered Ethernet with FlexRay: An Evaluation of Competing Approaches to Real-time for In-Vehicle Networks*. May 2009. – Unveröffentlicht
- [TTTech Computertechnik AG] TTTECH COMPUTERTECHNIK AG: *TTEthernet*. – URL <http://www.tttech.com>