Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Masterseminar

Stefan Buschmann

Redundanzkonzepte für RT-Ethernet Backbones im Automobil

Fakultät Technik und Informatik
Studiendepartment Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Inhaltsverzeichnis

1 Einleitung & Motivation

Durch die Entwicklung und Integration immer neuerer Fahrassistenz- und Entertainmentsysteme in modernen Fahrzeugen, steigen auch die Anforderungen an die Kommunikationssysteme. Anwendungen wie z. B. Videostreams oder Rückfahrkameras benötigen eine hohe Bandbreite, während „X-By-Wire“-Lösungen (z. B. Steer-By-Wire) hohe Anforderungen an die Sicherheit stellen. Sollen alle Daten über das gleiche Netzwerk bereitgestellt werden, würde dies bei den derzeitig eingesetzten Technologien wie CAN, FlexRay oder MOST an die Grenzen stoßen.

Eine Möglichkeit das Problem der geringen Bandbreite zu lösen, ist der Einsatz von Standard-Ethernet (vgl. Institute of Electrical and Electronics Engineers (2005)) als Backbone. In modernen Fahrzeugen wird Ethernet schon für bestimmte Aufgaben eingesetzt. Das Problem, dass sicherheitskritischer Verkehr zuverlässig übertragen wird, kann hiermit allerdings noch nicht gelöst werden, da keine Aussagen über die Ankunftszeitpunkte der Nachrichten getroffen werden können.

Um verlässlichen Datenverkehr garantieren zu können, ist das Standard-Ethernet nicht geeignet. Vielversprechende Ansätze, die dieses Problem beheben, sind zum einen Time-Triggered Ethernet (TTE) der Firma TTTech (vgl. TTTech Computertechnik AG) und zum anderen das Audio/Video Bridging (AVB) (vgl. IEEE 802.1 AVB Task Group). Bei beiden Technologien handelt es sich um Erweiterungen des Standard-Ethernets. TTE arbeitet nach dem Time Division Multiple Access (TDMA) Verfahren. Hier werden alle Nachrichten in drei unterschiedlich priorisierte Nachrichtenklassen eingeteilt, wodurch bei hoch priorisierten Nachrichten garantierte Ankunftszeiten vorausgesagt werden können. Bei AVB können die Netzwerkteilnehmer Streams erstellen, die es ermöglichen Bandbreite über die benötigten Pfade zu reservieren. Je nach Klassifizierung können auch hier Ankunftszeiten garantiert werden.

Auch wenn die erforderliche Bandbreite und die garantierten Ankunftszeiten erfüllt werden, kann es in einem Netzwerk zu Problemen kommen. Als Beispiel können hier defekte Hardware oder äußere Einflüsse genannt werden. Grade bei sicherheitskritischen Systemen stellen die Industrie und Normen wie die ISO 26262 (vgl. ISO 26262 (2012)) hohe Anforderungen an die eingesetzten Systeme. Eine Möglichkeit die Zuverlässigkeit in einem Netzwerk zu erhöhen, ist der Einsatz von Redundanzen, wodurch Fehler behoben oder kompensiert werden können.

Mehrfach vorhandene Systeme, wie sie beispielsweise in der Flugzeugindustrie zum Einsatz kommen (vgl. Yeh (1996)), bieten zwar ein hohes Maß an Sicherheit, sind aber für die Autobranche nicht geeignet. Neben dem zusätzlichen Gewicht, das aus der redundanten Hardware ensteht, steigen auch die Kosten. Diesem Umstand ist es geschuldet, dass Alternativen gefunden werden müssen.

Welche Möglichkeiten es gibt ein Netzwerk um Redundanz zu erweitern und welche Konzepte sich für den Einsatz im Automobil anbieten, wird im folgenden Kapitel behandelt. Das darauffolgende Kapitel befasst sich damit, wie unterschiedliche Netzwerke bewertet und deren Fehlerwahrscheinlichkeit ausgerechnet werden können. Der letzte Abschnitt befasst sich mit dem Stand der Arbeit, den anstehenden Aufgaben sowie einer Zusammenfassung.

2 Redundanz

Aus der DIN 40041 zum Thema Redundanz:

> "Vorhandensein von mehr funktionsfähigen Mitteln in einer Einheit, als für die Erfüllung der geforderten Funktion notwendig sind." (vgl. DIN 40041 (1990))

Dieses Kapitel befasst sich mit den Möglichkeiten wie Redundanz in einem Netzwerk geschaffen werden kann und welche Fehler hier auftreten können. Es stellt zusätzlich konkrete Konzepte vor, die sich mit der Anpassung der Topologie befassen, um die Verlässlichkeit zu erhöhen.

2.1 Allgemein

Um die Zuverlässigkeit eines Systems zu erhöhen, ist die Redundanz ein geeignetes Mittel. Es kann durch unterschiedliche Herangehensweisen erreicht werden. So lässt sich beispielsweise durch Einsatz redundanter Hardware der komplette Ausfall von Komponenten kompensieren, während redundante Software (z. B. zwei Entwicklerteams programmieren unabhängig voneinander die gleiche Funktion) Fehler eines Programms ausgleichen kann. Durch zusätzliche Nachrichten oder mehrere Pfade zu einem Ziel lässt sich in einem Kommunikationsnetzwerk Zuverlässigkeit durch redundante Kommunikation schaffen.

Zu beachten ist aber auch, dass durch jede eingesetzte Redundanz auch neue Fehlerquellen in ein System gebracht werden. Zusätzliche Hardware kann beispielsweise Fehlfunktionen

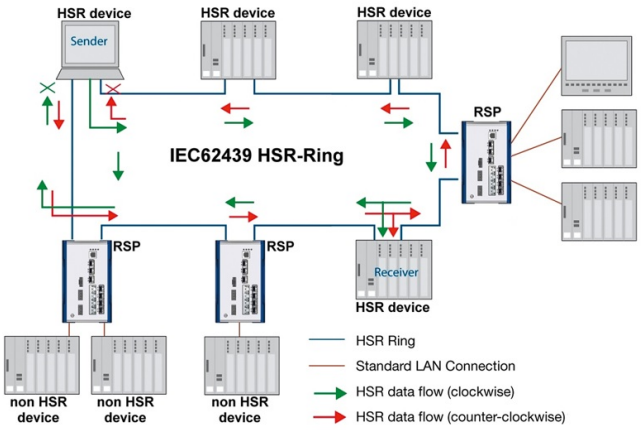


Abbildung 2.1: Aufbau einer HSR Topologie

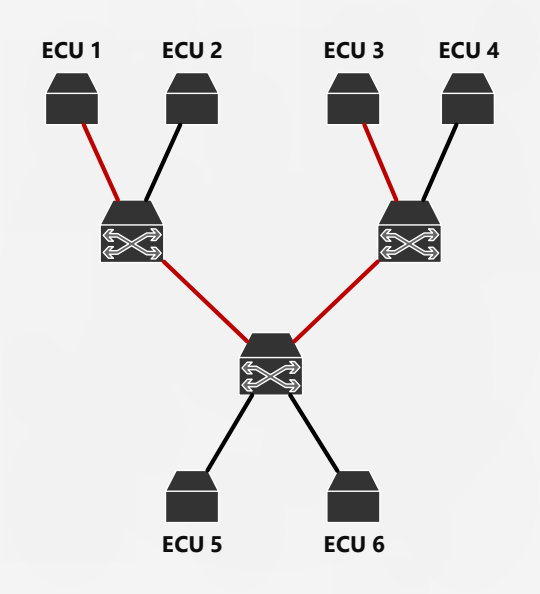
haben und fehlerhafte Nachrichten an das Netzwerk schicken. Hierdurch enstehen wieder neue Anforderungen an die Komponenten eines Netzwerks. Mit den Problemen der Identifikation, Replikation und Eliminierung von redundanten Nachrichten befasst sich der Standard IEEE 802.1CB (vgl. IEEE 802.1 TSN Task Group).

2.2 Redundanzkonzepte

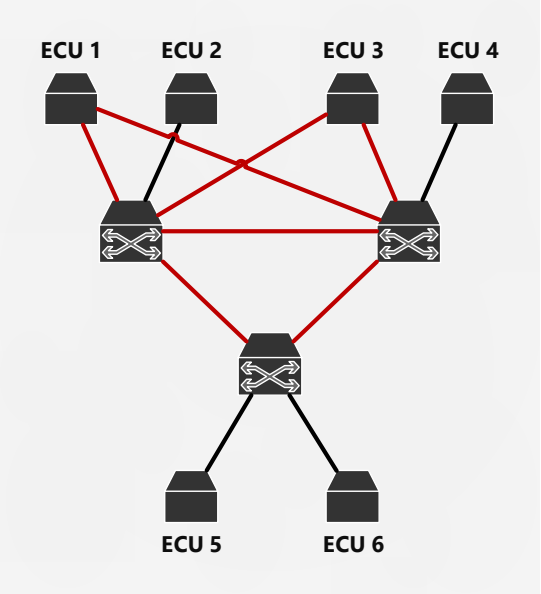
In diesem Abschnitt werden Möglichkeiten gezeigt, wie Redundanz in einem Netzwerk aussehen kann. Im ersten Teil wird die *High Availability Seamless Redundancy* (HSR) vorgestellt (vgl. ISO 62439 (2012)). Anschließend werden weitere Konzepte gezeigt, wie ein Netzwerk erweitert werden kann.

High Availability Seamless Redundancy (HSR) Bei diesem Konzept wird das Netzwerk als Ring aufgebaut. Die Daten werden von den Teilnehmern in beide Richtungen vom Ring verschickt und so lange von Switch zu Switch weitergeleitet, bis das Paket seinen Zielort erreicht hat oder an seinen Ausgangsort zurückkehrt (siehe Abbildung 2.1). Durch die doppelte Übertragung der Nachrichten steht dem Netzwerk allerdings nur die Hälfte der Bandbreite zur Verfügung.

Sollte das Paket an einer Stelle im Netz nicht weitergereicht werden können, kann das andere Paket immernoch das Ziel erreichen. Durch das duplizierte Senden der gleichen Nachricht in beide Richtungen entstehen keine Ausfallzeiten, sollte der Kommunikationsweg an einer Stelle unterbrochen werden.



(a) Kritischer Pfad in einem Netzwerk



(b) Kritischer Pfad in einem Netzwerk, erweitert um redundante Pfade

Abbildung 2.2: Redundanz schaffen durch redundante Pfade

Erweiterung einer Topologie Eine weitere Möglichkeit speziell kritische Bereiche innerhalb eines Netzwerks zu sichern, besteht darin die Topologie um redundante Pfade zu erweitern oder spezielle Komponenten zu duplizieren. Die linke Grafik in Abbildung 2.2 zeigt einen kritischen Pfad zwischen den ECUs 1 und 3. Im rechten Teil wurde das Netzwerk um diverse redundante Verbindungen erweitert.

Das Netzwerk wäre somit gegen Ausfälle einzelner Verbindungen gesichert. Auch nach dem Ausfall eines Routers könnten die Knoten weiterhin uneingeschränkt miteinander kommunizieren. Lediglich die ECUs stellen noch *Single Points of Failure* dar. Sollte hiervon eine Ausfallen, könnte ein redundanter Knoten die Arbeit übernehmen oder ein anderer Knoten müsste den kritischen Aufgabenbereich abdecken.

3 Fehlermodelle

Jedes Netzwerk hat mit einer Vielzahl an Fehlern und Problemen zu kämpfen. Auch TTE und AVB sind hier keine Ausnahmen. Das Resultat sind fehlerhaft übertragene Nachrichten oder

sogar komplett verlorene Frames. Um für ein Netzwerk ein Szenario kreieren zu können, wird ein Modell benötigt, mit dem dieses dargestellt werden kann. Im Folgenden wird ein solches Fehlermodell erarbeitet und anhand eines Beispiels dargestellt. Im Anschluss werden die Einschränkungen des vorgestellten Modells aufgezeigt und ein detaillierterer Ansatz gezeigt, der es ermöglicht auch mit komplexeren Fehlerszenarien umgehen zu können.

3.1 Fehler

Dieser Abschnitt behandelt mögliche Fehler, die in einem Netzwerk auftreten können. Die erste Frage, die beantwortet wird ist: Wo tritt der Fehler auf und was ist die Ursache dafür? Anschließend wird der mögliche zeitliche Verlauf der Störung behandelt. Zum Ende werden die unterschiedlichen Arten der Fehler aufgeführt.

In dem hier betrachteten Umfang besitzt ein Ethernet-Netzwerk drei potentielle Fehlerquellen bzw. Orte an denen die Fehler auftreten. Das sind die teilnehmenden Knoten, der Switch sowie die einzelnen Verbindungen zwischen den Knoten und Switches. Ein Knoten kann beispielsweise aufgrund von fehlerhaften Sensordaten oder durch Softwarefehler fehlerhafte Nachrichten verschicken. Auch wäre ein kompletter Ausfall durch einen Hardwaredefekt denkbar. Bei einem Switch würde zum Beispiel auch ein Nachrichtenverlust durch einen Überfluss des Empfangsbuffers als mögliche Ursache hinzukommen. Die Verbindung zweier Netzwerkteilnehmer kann durch elektromagnetische Störungen beeinflusst werden oder durch einen mechanischen Defekt des Kabels komplett ausfallen. Je nach Art des Fehlers und dessen schwere wirken diese unterschiedlich lange auf das Netz ein.

Um eine Störung zeitlich zu bestimmen kann zwischen zeitlich begrenzten (z. B. äußere Einflüsse wirken nur 100 ms auf das Medium ein) und permanten (bis zum Austausch der Komponente) Fehlern unterschieden werden. Hieraus ergibt sich dann beispielsweise auch ob eine Nachricht nur einmalig oder mehrere Nachrichten hintereinander falsch übertragen wurden. Des Weiteren können Fehler auch zyklisch auftreten und so in regelmäßigen Abständen zu Störungen führen.

Zum Schluss wird noch die Art des Fehlers bestimmt. Eine Nachricht kann komplett verloren gehen oder fehlerhaft übertragen werden. Bei einer fehlerhaften Übertragung kann zudem bestimmt werden ob es sich etwa durch einen Bitfehler o. ä. um eine ungültige Nachricht handelt (z. B. Checksumme falsch) oder ob die Daten, die von dem Sender mitgeschickt wurden nicht korrekt sind. In Bezug auf TTE und AVB kann auch noch ein falscher Empfangszeitpunkt einen Fehler verursachen.

3.2 Fehlermodell

Ein Fehlermodell dient dazu Fehler, ihre Auswirkungen, ihre Wahrscheinlichkeit und ihre Auswirkungen zu beschreiben. Das in diesem Abschnitt vorgestellte Fehlermodell ($\mathcal{FM}$) basiert auf der wissenschaftlichen Arbeit von Zimmer und Jantsch (2003). Es wird hier in folgender Form beschrieben:

$\mathcal{FM} : f_i \rightarrow (\alpha_i, P_i, DM_i)$

α_i : Wahrscheinlichkeit, dass Fehler f_i auftritt

P_i : Matrix aller möglichen Kombinationen von Fehlern

DM_i : Wie stark werden nebenliegende Verbindungen beeinträchtigt

Die Matrix P_i beschreibt die möglichen Auswirkungen des Fehlers, sollte er eintreten. Er wird wie folgt dargestellt:

$\rho_i(\omega, d, e)$

ω : Anzahl an naheliegenden Komponenten, die betroffen sind

d : Dauer des Effekts e; Einheit kann festgelegt werden

e : Effekt auf dem Kabel (z. B. invertierte Bits)

Da die genaue Kodierung einer Nachricht im Laufe des Masterprojekts nicht genauer betrachtet wird, spielt der genaue Effekt e der sich auf die Nachricht auswirkt keine Rolle. Von daher wird dieser nicht weiter berücksichtigt.

3.3 Beispiele

Um die Erstellung und die Nutzung des Fehlermodells zu erläutern, folgen in diesem Abschnitt Beispiele, die zeigen wie ein konkretes Fehlermodell aussieht, wie diese in einem Netzwerk eingesetzt werden können und wie ein ganzes Fehlerszenario aussehen kann.

$$P_1 = \begin{pmatrix} p_1(1,0) & p_1(1,1) & p_1(1,2) \\ p_1(2,0) & p_1(2,1) & p_1(2,2) \\ p_1(3,0) & p_1(3,1) & p_1(3,2) \end{pmatrix} \tag{3.1}$$

Fehlermodell: defektes Kabel Die Matrix 3.1 zeigt die Matrix P_1 dieses konkreten Fehlermodells. Für dieses Beispiel wurde angenommen, dass höchstens 3 angrenzende Verbindungen

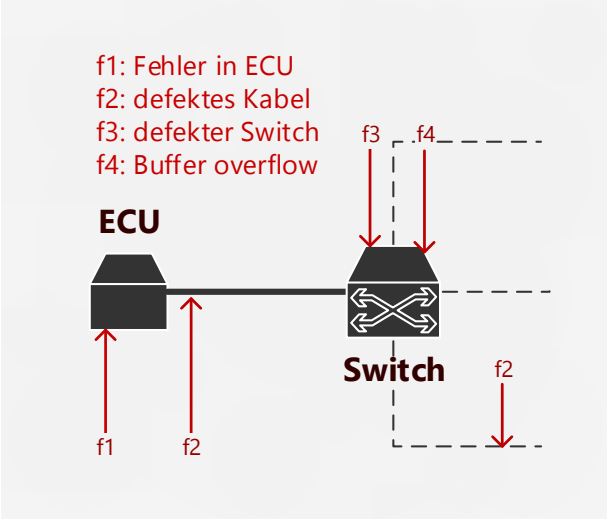


Abbildung 3.1: Ausschnitt eines Fehlerszenarios

von einer Störung auf diesem Kabel betroffen sind. Als maximale Dauer d_{max} wird $d = 2$ angenommen. Die Kombinationen von p_1 mit einer Dauer von 0 stehen für permanente Fehler.

$$P_1 = \begin{pmatrix} 0,01 & 0,60 & 0,10 \\ 0,00 & 0,15 & 0,03 \\ 0,00 & 0,10 & 0,01 \end{pmatrix} \tag{3.2}$$

Tritt ein Fehler in einer Komponente auf, muss noch ermittelt werden, welcher konkrete Fehlerfall eintritt. Um eine Auswahl zu ermöglichen, wird eine zweite Matrix erstellt, welche die Wahrscheinlichkeiten angibt welcher Fehler explizit auftritt. Matrix 3.2 zeigt die Verteilung für das gegebene Beispiel. So ist es mit einem Wert von 0,6 am wahrscheinlichsten, dass ein kurzzeitiger Fehler, der keine weiteren Komponenten beeinflusst, eintritt. Die Wahrscheinlichkeit, dass es sich um einen permanenten Fehler handelt, liegt lediglich bei einem Prozent.

Erstellen eines Fehlerszenarios Um ein ganzes Netzwerk abzudecken, wird eine Vielzahl an unterschiedlichen Fehlermodellen benötigt. Ein solches Fehlerszenario wird wie folgt definiert: $S = \{\mathcal{FM}(f_1), \mathcal{FM}(f_2), \mathcal{FM}(f_3), \mathcal{FM}(f_4)\}$

Abbildung 3.1 zeigt, wie die Fehlermodelle den einzelnen Komponenten zugeordnet werden können. Das Modell $\mathcal{FM}(f_1)$ betrifft den Knoten, $\mathcal{FM}(f_2)$ das Kabel, während $\mathcal{FM}(f_3)$ und $\mathcal{FM}(f_4)$ beide unterschiedliche Fehler im Switch darstellen. Einer Komponente können also auch mehrere Fehlermodelle zugeordnet werden. Es ist auch möglich mehreren Knoten

das gleiche Modell zuzuordnen ($\mathcal{FM}(f_2)$ in 3.1), wenn es sich beispielsweise um baugleiche Komponenten handelt.

Mit diesen Mitteln lassen sich Komponenten mit unterschiedlichen Eigenschaften modellieren. So können besonders fehleranfällige Komponenten in einem Netzwerk verbaut werden, um anschließend die Auswirkungen auf das Netzwerk zu errechnen. Ebenso kann gezeigt werden, an welchen Stellen sich das Netz besonders unzuverlässig verhält und wie sich etwaige Änderungen am Netz auf das ganze Systemverhalten auswirken.

3.4 Erweiterung der Fehlermodelle

Bei der im vorherigen Abschnitt vorgestellten Methode tritt ein Fehler ausschließlich bei einer Komponente auf und kann gegebenenfalls andere beeinflussen. Außerdem wird nur dazwischen unterschieden ob eine Nachricht übertragen werden kann oder nicht. In der Realität sind die Ausfallmöglichkeiten und die Art der auftretenden Fehler viel komplexer. So können Nachrichten beispielsweise erst verspätet beim Empfänger ankommen oder es bestehen Abhängigkeiten bei gewissen Ereignissen (z. B. mehrere Komponenten hängen an der gleichen Stromversorgung und diese fällt aus). Auch zyklisch auftretende Fehler lassen sich so nicht abbilden.

Eine Möglichkeit ein detailreicheres Szenario darzustellen ist der Einsatz der Fehlerbaumanalyse (engl. Fault Tree Analysis, FTA). Es wird ein Fehlerbaum aufgebaut, der die Identifizierung aller möglichen Ausfallkombinationen innerhalb eines technischen Systems ermöglicht. Der folgende Abschnitt gibt einen Einblick über die Funktionsweise und die Möglichkeiten der FTA.

Fehlerbaumanalyse Die DIN 25424 (vgl. DIN 25424-1 (1981) und DIN 25424-2 (1990)) befasst sich mit den Zielen, dem Aufbau und der Verwendung von Fehlerbäumen. Außerdem werden Berechnungen vorgestellt, mit deren Hilfe etwa Zuverlässigkeitskenngrößen wie z. B. die Ausfallhäufigkeit, die Eintrittshäufigkeit eines unerwünschten Ereignisses oder die Wahrscheinlichkeit der Nichtverfügbarkeit des Systems zu einem bestimmten Zeitpunkt berechnet werden.

Um ein System anhand eines Fehlerbaums zu analysieren, müssen zuerst unerwünschte Ereignisse und Ausfallkriterien definiert werden. Es wird also beispielsweise festgelegt, welche Komponenten in einem Netzwerk nicht gleichzeitig ausfallen dürfen, damit Nachrichten weiterhin übertragen werden können. Anschließend werden Zuverlässigkeitskenngrößen und

Zeitintervalle bestimmt, die das Verhalten der einzelnen Komponenten darstellen. Im nächsten Schritt wird festgelegt auf welche Arten eine Komponente ausfallen kann.

Mit Hilfe der gegebenen Informationen kann im Anschluss der Fehlerbaum aufgestellt werden. Verknüpfungen zwischen den Komponenten, die zu einem unerwünschten Ereignis führen, können durch logische Zusammenhänge wie UND, ODER oder NICHT beschrieben werden. Anhand des so konstruierten Fehlerbaums können Auswertungen vorgenommen und das System schließlich bewertet werden.

4 Ausblick & Risiken

In diesem Kapitel wird ein Ausblick darauf gegeben, welche Arbeiten im Laufe des Projekts noch anfallen werden. Es wird auf mögliche Risiken hingewiesen und abschließend eine Zusammenfassung über den derzeitigen Stand der Arbeit geliefert.

Ausblick Sobald die vorbereitenden Schritte im Hinblick auf die Analyse und Bewertung unterschiedlicher Redundanzkonzepte abgeschlossen sind, wird in der eventbasierten Netzwerksimulation OMNeT++ (vgl. OMNeT++ Community) das bereits vorhandenen Simulationsmodell für TTE und AVB, *CoRE4INET* (vgl. CoRE-Arbeitsgruppe), erweitert. In der Simulation ist es möglich, Netzwerke mit mehreren Teilnehmern, Switches und Verbindungen zu erstellen. Außerdem kann der Datenverkehr zwischen den Knoten konfiguriert und analysiert werden.

Das derzeitige Simulationsmodell besitzt weder Funktionen um mit Redundanz umzugehen, noch um Fehlerszenarien zu erstellen. Die Implementierung der nötigen Funktionen stellt somit einen wichtigen Arbeitsabschnitt dar. Es wird ein Konzept entwickelt, in dem die nötigen Anpassungen erfasst werden.

Wurden die Funktionen soweit integriert, muss die Möglichkeit geschaffen werden verschiedene Redundanzkonzepte miteinander zu vergleichen und die Ergebnisse zu bewerten. Hierzu werden Statisiken geführt, die während der Simulation erfasst werden. Mögliche Inhalte wären hier z. B. wie viele Nachrichten verloren gegangen sind, ob die zeitlichen Anforderungen jederzeit eingehalten wurden oder wie sich gewisse Fehlersituationen auf das gesamte Netzwerk ausgewirkt haben.

Wurde das Simulationsmodell um alle benötigten Funktionen erweitert, können unterschiedliche Konzepte entwickelt, simuliert und analysiert werden. Aufgrund der durch die Simulation gelieferten Ergebnisse können unterschiedliche Lösungswege geschaffen werden um bestimm-

te Bereiche des Netzwerks abzusichern. Die Frage, an welcher Stelle sich welche Redundanz (z. B. redundante Hardware oder redundante Verbindungen) am besten eignet, wird somit beantwortet werden können. Auch ein gesamtes Autonetzwerk, basierend auf einem realen Fahrzeug, kann anschließend auf diese Weise angepasst werden.

Risiken Durch die Forderungen, die an das Kommunikationssystem gestellt werden, nach möglichst geringen Kosten (Industrie) aber auch an ein nötiges Maß an Sicherheit (z. B. ISO 26262) kann es zu Problemen kommen. Wenn die Analysen der Redundanzkonzepte ergeben, dass die erforderliche Redundanz zu kostenintensiv ist, kann dies dazu führen, dass die Autohersteller die Technologie nicht in Betracht ziehen.

Ein weiterer Punkt ist, dass sich die derzeitigen Standards teilweise noch in der Entwicklung befinden. Ändern sich hier maßgebliche Eigenschaften, kann dies Auswirkungen auf die bisherigen Implementationen haben. Da sich die auszuführenden Arbeiten des Masterprojekts aber hauptsächlich auf die Simulation beziehen werden, besteht hier die Möglichkeit Änderungen leicht anzupassen.

Zusammenfassung Um die Zuverlässigkeit eines Kommunikationssystems zu erhöhen, bietet die Redundanz die Möglichkeit viele unterschiedliche Probleme zu lösen. So können Ausfälle von Komponenten kompensiert, verloren gegangene oder verfälschte Nachrichten ersetzt und Engpässe an zentralen Stellen umgangen werden.

Es darf aber nicht außer Acht gelassen werden, dass durch Redundanz auch neue Anforderungen enstehen. Das Netzwerk wird durch die zusätzlichen Teilnehmer komplexer und es kommen neue Fehlerquellen hinzu. Zusätzlich muss der Umgang mit redundanten Nachrichten geklärt werden.

Damit möglichst viele dieser Faktoren bei der Kommunikation berücksichtigt werden können, wird ein Verfahren benötigt, das diese erfasst. Hier bieten Fehlermodelle wie die in Kapitel 3 erläuterten Konzepte eine gute Basis um Netzwerke miteinander zu vergleichen und sie zu bewerten.

Als erste Testumgebung, die sowohl die Redundanzkonzepte als auch die Fehlermodelle umsetzt, wird die Simulationsumgebung OMNeT++ mit dem Simulationsmodell *CoRE4INET* dienen. Somit wird eine gute Grundlage geschaffen um verschiedene Redundanzkonzepte zu simulieren, zu analysieren, zu bewerten und zu vergleichen. Mit diesen Ergebnissen kann anschließend abgeschätzt werden, ob sich Real-Time-Ethernet Backbones im Automobil als praktikabel erweisen.

Literaturverzeichnis

[CoRE-Arbeitsgruppe] CoRE-Arbeitsgruppe: *CoRE for INET*. – URL http://core.informatik.haw-hamburg.de/en/projects/simulation/core4inet.html. – Zugriffsdatum: 2015-02-27

[DIN 25424-1 1981] Deutsches Institut für Normung: Fehlerbaumanalyse; Methode und Bildzeichen / DIN. Berlin, 1981 (25424-1). – DIN

[DIN 25424-2 1990] Deutsches Institut für Normung: Fehlerbaumanalyse; Handrechenverfahren zur Auswertung eines Fehlerbaumes / DIN. Berlin, 1990 (25424-2). – DIN

[DIN 40041 1990] Deutsches Institut für Normung: Zuverlässigkeit / DIN. Berlin, 1990 (40041). – DIN

[IEEE 802.1 AVB Task Group] IEEE 802.1 AVB Task Group: *IEEE 802.1 Audio/Video Bridging (AVB)*. – URL http://www.ieee802.org/1/pages/avbridges.html

[IEEE 802.1 TSN Task Group] IEEE 802.1 TSN Task Group: *IEEE 802.1CB - Frame Replication and Elimination for Reliability*. – URL http://www.ieee802.org/1/pages/802.1cb.html

[Institute of Electrical and Electronics Engineers 2005] Institute of Electrical and Electronics Engineers: IEEE 802.3: LAN/MAN CSMA/CD Access Method / IEEE. 2005 (IEEE 802.3-2005). – Standard

[ISO 26262 2012] International Organization for Standardization: Road vehicles – Functional safety / ISO. Genf, 2012 (26262). – ISO

[ISO 62439 2012] International Electrotechnical Commission: Industrial communication networks / IEC. Genf, 2012 (62439). – IEC

[OMNeT++ Community] OMNeT++ Community: *OMNeT++ 4.6*. – URL http://www.omnetpp.org. – Zugriffsdatum: 2015-02-27

[TTTech Computertechnik AG] TTTech Computertechnik AG: . – URL http://www.tttech.com. – Zugriffsdatum: 2015-02-27

[Yeh 1996] Yeh, Y.C.: Triple-triple redundant 777 primary flight computer. In: *Aerospace Applications Conference, 1996. Proceedings., 1996 IEEE* Bd. 1, 1996, S. 293–307 vol.1

[Zimmer und Jantsch 2003] Zimmer, H. ; Jantsch, A.: A fault model notation and error-control scheme for switch-to-switch buses in a network-on-chip. In: *Hardware/Software Codesign and System Synthesis, 2003. First IEEE/ACM/IFIP International Conference on*, Oct 2003, S. 188–193