AnTeS-NeCom

Analyse der Fehlerausbreitung in der Netzwerkkommunikation digitaler Leittechnik-systeme mit Hilfe eines Testsystems

GRS - 689

Gesellschaft für Anlagen-
und Reaktorsicherheit
(GRS) gGmbH

AnTeS-NeCom

Analyse der Fehlerausbreitung in der Netzwerkkommunikation digitaler Leittechnik-systeme mit Hilfe eines Testsystems

Abschlussbericht

Christian Müller
Joachim Herb
Patrick Gebhardt
Jaroslaw Shvab

September 2023

Anmerkung:

Das diesem Bericht zugrunde liegende Forschungsvorhaben wurde mit Mitteln des Bundesministeriums für Umwelt, Naturschutz, nukleare Sicherheit und Verbraucherschutz (BMUV) unter dem Förderkennzeichen RS1590 durchgeführt.

Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei der GRS.

Der Bericht gibt die Auffassung und Meinung der GRS wieder und muss nicht mit der Meinung des BMUV übereinstimmen.

GRS - 689
ISBN 978-3-949088-80-3

Deskriptoren

AnTeS, Cyberangriff, Digitale Leittechnik, Fehlerinjektion, Netzwerkkommunikation, Sensitivitätsanalyse, Simulation, Testsystem

Kurzfassung

Das Forschungs- und Entwicklungsprojekt RS1590, finanziert vom Bundesministerium für Wirtschaft und Energie (BMWi) und später vom Bundesministerium für Umwelt, Naturschutz, nukleare Sicherheit und Verbraucherschutz (BMUV), widmete sich der Untersuchung von Auswirkungen von Kommunikationsfehlern in den Netzwerken digitaler Leittechniksysteme in Kernkraftwerken.

Dieses Projekt baute auf vorherige und teilweise parallellaufende Vorhaben auf, in denen die Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) grundlegende, modellbasierte Methoden zur Analyse des Verhaltens digitaler Leittechnik bei auftretenden Fehlern entwickelte (Förderkennzeichen 3615R01343, 4718R01314, 4722R01215). Ein wesentlicher Bestandteil dieser Forschungsarbeit war die Entwicklung des Analyse- und Testsystems AnTeS, das unter anderem reale sowie simulierte Sicherheitsleittechniksysteme beinhaltet. Die im Rahmen von AnTeS angewendeten Methoden umfassen Failure Mode and Effects Analyses (FMEAs), automatische Auswirkungsanalysen (eine von der GRS entwickelte Automatisierung und Erweiterung der FMEA), Fehlerbaumanalysen und Monte-Carlo-Simulationen. Diese Methoden dienen dazu, potenzielle Fehlerursachen sowie deren Auswirkungen zu identifizieren und zu bewerten.

Eine zentrale Rolle spielen auch moderne Netzwerktechnologien und -topologien, die sowohl für die interne als auch externe Kommunikation in Leittechniksystemen verwendet werden. Der Einfluss dieser Technologien auf die Zuverlässigkeit und Sicherheit der Systeme wurde speziell in diesem Projekt untersucht, um Lücken in den bisherigen Methoden und in der Anwendung von AnTeS zu adressieren.

Das Hauptziel des Projekts bestand darin, ein tiefgreifendes Verständnis der Netzwerkkommunikation innerhalb der Leittechniksysteme zu entwickeln. Dazu gehörte die Entwicklung von Methoden zur Fehlerinjektion in die Netzwerkkommunikation und die anschließende Untersuchung der Auswirkungen solcher Fehler auf die Zuverlässigkeit verschiedener Modellsysteme. Für diese Untersuchungen wurden bestehende Modellsysteme erweitert und neue Systeme konzipiert und analysiert. Mittels Sensitivitätsanalysen wurde der Einfluss unterschiedlicher Parameter auf die Systemzuverlässigkeit evaluiert. Die Ergebnisse des Projekts zeigen, dass digitale Leittechniksysteme in Kernkraftwerken eine hohe Robustheit gegenüber Netzwerkfehlern aufweisen und dass diese

Fehler nur einen marginalen Einfluss auf die Gesamtzuverlässigkeit der Systeme haben. Diese Erkenntnisse tragen maßgeblich zur Weiterentwicklung der GRS-Methodologie bei.

Abstract

The research and development project RS1590, funded by the Federal Ministry for Economic Affairs and Energy (BMWi) and later by the Federal Ministry for the Environment, Nature Conservation, Nuclear Safety and Consumer Protection (BMUV), was dedicated to investigating the impact of communication errors in the networks of digital instrumentation and control (I&C) systems in nuclear power plants. This project builds upon previous and, in part, concurrent initiatives in which the Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) developed fundamental, model-based methods for analyzing the behavior of digital I&C in the event of failures (funding codes 3615R01343, 4718R01314, 4722R01215). A crucial component of this research work was the development of the Analysis and Testing System AnTeS, which includes both real and simulated I&C systems.

The methods applied within the framework of AnTeS include Failure Mode and Effects Analyses (FMEAs), automated impact analyses (an automation and extension of FMEA developed by GRS), Fault Tree Analyses (FTAs), and Monte-Carlo simulations. These methods serve to identify and evaluate potential causes of failures and their impacts.

Modern network technologies and topologies, used for both internal and external communication in I&C systems, also play a central role. The influence of these technologies on the reliability and safety of the systems was specifically examined in this project to address gaps in the existing methods and in the application of AnTeS.

The primary goal of the project was to develop an in-depth understanding of network communication within I&C systems. This included the development of methods for fault injection into network communication and the subsequent examination of the impacts of such failures on the reliability of various model systems. For these investigations, existing model systems were expanded, and new systems were designed and analyzed.

Through sensitivity analyses, the impact of different parameters on system reliability was evaluated. The project's findings indicate that digital I&C systems in nuclear power plants are highly robust against network errors, and that these errors only have a marginal impact on the overall reliability of the systems. These insights contribute significantly to the further development of the GRS methodology.

Inhaltsverzeichnis

1 Einleitung

Kernkraftwerke weltweit verwenden heutzutage häufig Leittechniksysteme mit digitalen Einrichtungen[1]. Diese Systeme sind aufgrund ihrer komplexeren Architekturen, Hardware und des Einsatzes von Software schwerer auf Fehlerfreiheit zu überprüfen als analoge, festverdrahtete Systeme mit ähnlichen Funktionen.

1.1 Stand von Wissenschaft und Technik

Bislang fehlen weitgehend noch immer detaillierte, allgemein anerkannte Nachweisverfahren und Anforderungen für den zuverlässigen Einsatz digitaler Leittechnik in Kernkraftwerken (siehe hierzu auch in /MCH 21/). International ist die Bewertung digitaler Leittechnik daher ein wichtiges Forschungsthema, das auch von der GRS seit mehreren Jahren konsequent verfolgt wird. Bei der GRS werden dabei im Wesentlichen modellbasierte Ansätze verfolgt.

Im Rahmen des BMU-Vorhabens 3615R01343 („Entwicklung und Erprobung eines Werkzeugs zur Sensitivitätsanalyse der Fehlerauswirkungen in der sicherheitsrelevanten digitalen Leittechnik“) /MCH 18/ wurde eine modellbasierte Vorgehensweise entwickelt und erprobt, mit der das dynamische Verhalten digitaler Leittechnik beim Eintreten von systeminternen Fehlern analysiert werden kann. Auf Basis generischer Modelle moderner Systemarchitekturen wurde dabei durch Sensitivitätsanalysen der Einfluss unterschiedlicher Parameter (beispielsweise Reparaturzeiten, Redundanzgrad) auf die Zuverlässigkeit der Systeme untersucht.

Im Rahmen des BMU-Vorhabens 4718R01314 („AnTeS“) /MCH 21/ wurde die entwickelte Methodologie aufgegriffen und vor allem zusätzlich auch erweitert und validiert. Entscheidend hierbei war der Aufbau des Analyse- und Testsystems (AnTeS) der GRS, welches sowohl simulierte als auch reale Leittechniksysteme sowie verfahrenstechnische Simulationen umfasst. Dieses System bietet eine flexible Testumgebung für

[1] Die Sicherheitsanforderungen an Kernkraftwerke (SiAnf) in der Version vom März 2015 (/BMU 15/, /BMU 15a/) unterscheiden zwischen rechnerbasierten und programmierbaren leittechnischen Einrichtungen. Hierbei bestehen programmierbare Geräte definitionsgemäß aus mindestens einem diskreten programmierbaren Bauelement (die Anwendungsfunktion wird durch Verdrahtung oder durch Bauelementfunktionen realisiert), wogegen rechnerbasierte Geräte mindestens einem Prozessor enthalten und die Anwendungsfunktion im Speicher hinterlegt ist. Im Rahmen dieses Vorhabens wird stattdessen das allgemein gängige ‚digital‘ verwendet.

Analysen und Forschungsarbeiten zur Untersuchung, aber auch Verifizierung und Validierung digitaler Leittechnik.

Die vor diesem Projekt bei der GRS entwickelten und angewandten Methoden berücksichtigten insbesondere noch nicht explizit den Einfluss der (digitalen) Netzwerktechnologien, wie sie in modernen Leittechniksystemen zum Einsatz kommen. Die Signalverarbeitung digitaler Leittechniksysteme nutzt sowohl für die interne als auch für die externe Kommunikation unterschiedliche Netzwerktechnologien und -topologien, deren Zuverlässigkeit und Sicherheit im Vordergrund des hier beschriebenen Vorhabens stand.

Hierbei wurde im Rahmen des hier vorgestellten Vorhabens zwischen interner und externer Netzwerkkommunikation wie folgt unterschieden:

- Interne Netzwerkkommunikation:
  - Austausch von Informationen innerhalb des Leittechniksystems mit Hilfe von Netzwerktechnologien, beispielsweise zwischen verschiedenen Redundanzen desselben Leittechniksystems, aber auch zwischen verschiedenen Leittechniksystemen.
- Externe Netzwerkkommunikation:
  - Kommunikation zwischen einem Leittechniksystem und Geräten, die für die Ausführung der leittechnischen Funktion im laufenden Betrieb nicht unmittelbar notwendig sind. Typischerweise z. B. die Kommunikation eines Leittechniksystems mit einem Servicegerät (beispielsweise zur Programmierung des Systems („Engineering“) oder zur Überwachung des leittechnischen Systems).

Wie bereits erwähnt, wurde AnTeS ursprünglich im Rahmen des BMU-Vorhabens 4718R01314 /MCH 21/ entwickelt. Derzeit findet aber im Rahmen des BMUV-Vorhabens 4722R01215 /GRS 23/ auch eine Weiterentwicklung statt (insbesondere hinsichtlich betrieblicher Leittechniksysteme und Prioritätsmodule). AnTeS wird im nachfolgenden Abschnitt etwas genauer beschrieben.

1.2 AnTeS, das Analyse- und Testsystem der GRS

Das Analyse- und Testsystem der GRS (AnTeS) ist eine modulare Plattform unterschiedlicher Werkzeuge und Methoden für Untersuchungen zur Leittechnik. AnTeS verfügt grundsätzlich über vier Module (siehe auch Abb. 1.1):

AnTeS-SILT

- AnTeS-SILT-real: reales Sicherheitsleittechniksystem (SILT)
  - basierend auf Hard- und Software von Teleperm XS von Framatome
- AnTeS-SILT-sim: simulierte Sicherheitsleittechniksysteme
  - basierend auf Matlab/Simulink /MAT 23/

AnTeS-BELT

- AnTeS-BELT-real: reales betriebliches Leittechniksystem (BELT)
  - basierend auf Hard- und Software von Simatic S7 von Siemens
- AnTeS-BELT-sim: simulierte Leittechniksysteme
  - basierend auf Matlab/Simulink

AnTeS-PRIO

- AnTeS-PRIO-real: reale Prioritätsmodule (PRIO)
  - AV42, SPLM1
  - generisches Prioritätsmodul (GRS-Eigenentwicklung für AnTeS)
- AnTeS-PRIO-sim: simulierte Prioritätsmodule
  - basierend auf Matlab/Simulink

AnTeS-FRONT

- AnTeS-FRONT-real: reale verfahrenstechnische Systeme
  - Tanks, Antriebe, Messsensoren, Ventile, Pumpen
- AnTeS-FRONT-sim: simulierte Systeme
  - SimGen, siehe /MCH 21/

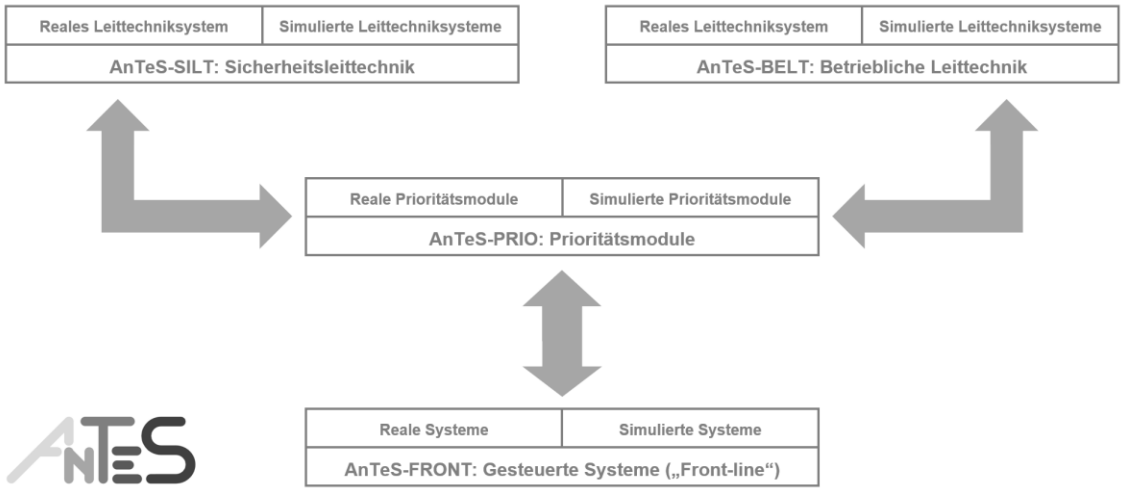

Abb. 1.1 AnTeS in der Übersicht

Darüber hinaus sind verschiedene Analysemethoden verfügbar, die zusammen mit den Modulen von AnTeS für Untersuchungen rund um die Leittechnik genutzt werden können:

- FMEA – Failure Mode and Effects Analysis
  - FMEA ist eine systematische Methode zur Identifizierung, Bewertung und Priorisierung potenzieller Fehler oder Schwachstellen in einem Produkt, Prozess oder System. Durch die Analyse von möglichen Ausfallursachen und den Auswirkungen dieser Fehler unterstützt FMEA dabei, frühzeitig Risiken zu erkennen und geeignete Maßnahmen zur Fehlervermeidung, -minimierung oder -beseitigung zu entwickeln. Die Methode wird in verschiedenen Branchen eingesetzt, darunter die Automobilindustrie, Luftfahrt, Medizintechnik und auch in der Energiewirtschaft, um die Zuverlässigkeit und Sicherheit von Produkten und Prozessen zu erhöhen.
  - Im Zusammenhang mit AnTeS und der bei der GRS angewandten Methodologie wird die FMEA hauptsächlich dazu benutzt, um für die weitere Modellierung die relevanten Ausfallarten (z. B. von Komponenten, Teilsystemen) zu bestimmen. Detailliertere Beschreibungen und weiterführende Referenzen können /MCH 21/ entnommen werden.
- Automatische Auswirkungsanalyse oder Ausfalleffektanalyse
  - Hierbei handelt es sich um eine erweiterte FMEA-Vorgehensweise, die im Rahmen des Vorhabens 4718R01314 /MCH 21/ entwickelt wurde.

Hierbei wird ein simuliertes oder reales System benutzt, in welches mit Hilfe von Fehlerinjektion Ausfälle (z. B. von Komponenten oder Teilsystemen) eingespeist werden können. Durch eine automatische Variation aller denkbaren Zustände aller berücksichtigten Teile des Systems („ist selbstmeldend aufgefallen“, „ist nicht-selbstmeldend ausgefallen“, „funktioniert einwandfrei“) und gleichzeitigem Aufzeichnen des Gesamtzustandes des Systems („Auslösung der Sicherheitsfunktion erfolgt bestimmungsgemäß“, „Auslösung der Sicherheitsfunktion erfolgt nicht bestimmungsgemäß“) können auf diese Weise sämtliche Ausfallkombinationen bestimmt werden, die einen Gesamtausfall des Systems gleichkommen.

  - Im Zusammenhang mit AnTeS ersetzt die umfangreichere und bei der automatischen Durchführung weniger fehleranfällige automatische Auswirkungsanalyse meist eine einfache FMEA. Die Ergebnisse der automatischen Auswirkungsanalyse unterstützen wiederum die Fehlerbaumanalyse. Detailliertere Beschreibungen hierzu können in /MCH 21/ nachgelesen werden.

- Fehlerbaumanalyse

  - Die Fehlerbaumanalyse ist eine systematische Methode zur Untersuchung von potenziellen Fehlerursachen und deren Auswirkungen in komplexen Systemen. Sie visualisiert die möglichen Fehlerpfade in Form eines Baumdiagramms, bei dem die oberste Ebene den unerwünschten Endzustand darstellt. Durch die schrittweise Analyse der Fehlerpfade von der Spitze des Baumes bis zu den Grundursachen können kritische Schwachstellen und potenzielle Kombinationen von Ereignissen identifiziert werden, die zu einem unerwünschten Ereignis führen könnten. Die Fehlerbaumanalyse ist ein leistungsfähiges Instrument, das in verschiedenen Branchen eingesetzt wird, um Risiken zu bewerten, Sicherheitsmaßnahmen zu entwickeln und die Zuverlässigkeit komplexer Systeme zu verbessern. Durch die Einbindung von Wahrscheinlichkeiten und Daten zu Einzelereignissen ermöglicht die Fehlerbaumanalyse auch die quantitative Bewertung von Risiken und die Ableitung von Wahrscheinlichkeiten für das Eintreten unerwünschter Ereignisse, was eine fundierte Entscheidungsgrundlage beispielsweise für präventive Maßnahmen bietet.

- Im Zusammenhang mit AnTeS liefern Fehlerbaumanalysen qualitativ dieselben Ergebnisse wie automatische Auswirkungsanalysen (wodurch sich die beiden Methoden gegenseitig überprüfen). Zusätzlich können mit Fehlerbaumanalysen auch quantitative Ergebnisse zu den untersuchten Systemen gewonnen werden. Detailliertere Beschreibungen und weiterführende Referenzen können /MCH 21/ entnommen werden. Vergleichbare quantitative Ergebnisse können aber auch mit Monte-Carlo-Simulationen gewonnen werden.

- Monte-Carlo-Simulation

  - Monte-Carlo-Simulationen sind eine computergestützte Methode, die in verschiedenen Bereichen angewendet wird, um komplexe Probleme zu analysieren für die analytische Lösungen schwierig oder unmöglich sind. Diese Methode basiert auf zufälligen Stichproben und wiederholt die Analyse eines Modells oder Systems Tausende oder sogar Millionen Male, wobei jedes Mal zufällige Variationen der Eingangsparameter berücksichtigt werden. Die Ergebnisse dieser Simulationen liefern statistische Verteilungen von möglichen Ausgängen und ermöglichen die Schätzung von Wahrscheinlichkeiten, Risiken und anderen quantitativen Informationen. Monte-Carlo-Simulationen finden Anwendung in der Finanzwelt, in Ingenieurswissenschaften, Naturwissenschaften, Risikoanalysen und vielen anderen Disziplinen, um eine bessere Vorstellung von den möglichen Ergebnissen komplexer Systeme oder Modelle zu erhalten.

  - Im Zusammenhang mit AnTeS werden simulierte Leittechniksysteme für Monte-Carlo-Simulationen verwendet, in welche statistische Ausfälle bestimmter Komponenten durch Fehlerinjektion eingespeist werden. Hierbei können zu Fehlerbaumanalysen vergleichbare quantitative Ergebnisse erzielt werden. Somit können Monte-Carlo-Simulationen Fehlerbaumanalyen im Einzelfall ganz ersetzen oder zumindest deren Ergebnisse überprüfen. Detailliertere Beschreibungen und weiterführende Referenzen können /MCH 21/ entnommen werden.

Durch die Kombination realer oder simulierter Module zu einem Gesamtsystem können je nach Anforderung unterschiedliche Konfigurationen und Leittechnikarchitekturen flexibel umgesetzt und mit den verfügbaren Methoden untersucht werden (siehe Abb. 1.1.)

Für dieses Vorhaben wurde ausschließlich das Modul AnTeS-SILT verwendet (reales und simulierte Sicherheitsleittechniksysteme). Abb. 1.2 zeigt im linken Bild das reale Leittechniksystem von AnTeS (TXS). Zu sehen sind die insgesamt drei bei der GRS vorhandenen TXS-Schränke in geschlossenem Zustand. Typischerweise werden für Versuche bei der GRS jedoch nur der linke und mittlere Schrank verwendet (rechtes Bild in Abb. 1.2), der dritte Schrank dient als Reserve und befindet sich noch in dem Zustand, in dem die GRS 2017 alle Schränke vom Kernkraftwerk Krümmel übernommen hatte (Details hierzu: siehe /MCH 21/).

Abb. 1.2 Das Modul AnTeS-SILT-real, ein reales Sicherheitsleittechniksystem basierend auf Teleperm XS

1.3 Anmerkungen zu diesem Bericht

Dieser Bericht wurde so aufgebaut, dass er möglichst übersichtlich und einfach zu lesen ist. Im Haupttext werden daher vorwiegend nur die relevanten Fakten und Ergebnisse präsentiert, während detailliertere Beschreibungen (z. B. zu den durchgeführten Versuchen) im Anhang A zu finden sind. Anhang A enthält darüber hinaus auch zusätzliche Informationen zu Arbeiten innerhalb des Vorhabens, die an keiner anderen Stelle erwähnt werden (Erstellung einer Software zur automatischen Umsetzung von TXS-Funktionspläne in Matlab/Simulink-Simulationsmodelle, Abschnitt A.1). Darüber hinaus werden im Anhang B zusätzlich einige grundlegende Begriffe aus dem Bereich der

digitalen Netzwerke erklärt, die im Haupttext verwendet wurden, aber dort nicht näher erläutert werden. Diese Erläuterungen können dem Leser das Verständnis der Zusammenhänge ggf. erleichtern.

2 Fehlerinjektion in die Netzwerkkommunikation des Testsystems

Um die Auswirkungen von Fehlern in der Netzwerkkommunikation von Leittechniksystemen analysieren zu können, wurde eine Reihe unterschiedlicher Versuche mit Hilfe eines (realen) Testsystems (konkret: Modul AnTeS-SILT-real) durchgeführt.

Der nachfolgende Abschnitt 2.1 beleuchtet zunächst allgemein die Netzwerkkommunikation innerhalb dieses Testsystems, im nachfolgenden Abschnitt 2.2 werden dann die Möglichkeiten zur Fehlerinjektion in die Netzwerkkommunikation des Testsystems beschrieben.

Obwohl nicht ausdrückliches Ziel dieses Vorhabens, lassen sich die entwickelten Fehlerinjektionsmöglichkeiten auch für die Durchführung (bzw. Nachstellung und Untersuchung) von Cyberangriffen nutzen. Einen kurzen Exkurs hierzu findet man im Abschnitt 2.3.

2.1 Netzwerkkommunikation im Testsystem

Das verwendete Testsystem (AnTeS-SILT-real) basiert auf Hard- und Softwarekomponenten der Leittechnikplattform Teleperm XS (TXS) von Framatome. Für sämtliche Entwicklungen, Tests und Analysen wurde dieses insbesondere so konfiguriert, dass die gesamte Netzwerkkommunikation (extern und intern) des Testsystems ausschließlich über Ethernet (nach IEEE 802.3) erfolgte.[2]

Durch gezielte Versuche unter Verwendung der Software Wireshark /WIS 23/ konnten wichtige grundlegende Erkenntnisse zur Netzwerkwerkkommunikation im Testsystem erlangt werden. So werden bei der Kommunikation über Ethernet im Testsystem Informationen zwischen Kommunikationspartnern in der Regel unidirektional ausgetauscht. D. h. beispielsweise auch, dass der Empfänger einer Nachricht diese nicht in irgendeiner Form quittiert.

[2] Typischerweise findet bei der im Modul AnTeS-SILT-real verwendeten Generation 2 des TXS die interne Kommunikation zwischen Rechnern bzw. Redundanzen des Leittechniksystems über Profibus statt. Viele Leittechniksysteme anderer Hersteller und auch Teleperm XS der Generation 4 (derzeit in Entwicklung) verwenden jedoch standardmäßig eher Ethernet /FRA 23/, das aus diesem Grund als Referenz für die Arbeiten in diesem Vorhaben herangezogen wurde.

Dies vermeidet einerseits evtl. unerwünschte Rückwirkungen vom Empfänger auf den Sender, andererseits allerdings kann in der Regel auf Seite des Senders auch kein Rückschluss über den korrekten Empfang der übermittelten Daten gezogen werden.

Die Netzwerkkommunikation im Testsystem erfolgt hardwarenah und verwendet keine übergeordneten Protokolle (wie z. B. das Internet Protokoll IP). Sämtliche Datenpakete, die innerhalb des Testsystems verschickt werden, können auf OSI-Layer 2 (zur Definition der OSI-Layer siehe Anhang B.5) wie in Abb. 2.1 schematisch dargestellt werden.[3]

Abb. 2.1 Schematische Darstellung von Datenpaketen in der Ethernet-Kommunikation des Testsystems (TXS, Generation 2)

Ein konkretes TXS-Ethernet-Datenpaket sieht somit beispielsweise wie in Abb. 2.2 aus, wenn die übertragenen Bytes durch Hexadezimalzahlen (gekennzeichnet durch vorangestellte „x“) dargestellt werden (zur Darstellung von Bytes als Hexadezimalzahlen siehe auch Anhang B.2).

```
x08 x00 x06 x01 xa0 x01 x08 x00 x06 x01 xa0 x00 x00 x1f
x14 x14 x03 x02 x00 xc0 x88 x05 x00 x44 x81 x0b x00 x0c
x00 x1c x00 x00 x01 x02 x1f x00 x00 x65 x00 x01 x00 x00
x02 x01 x00 x00 x00 x00 x00 x00 x00 x00 x00 x00 x00 x00
x00 x00 x00 x00
```

Abb. 2.2 Beispiel für ein TXS-Datenpaket (innerhalb des Testsystems)

Das Datenpaket in Abb. 2.2 enthält im Einzelnen die folgenden Daten/Informationen:

- MAC-Adresse des Empfängers:
  - x08 x00 x06 x01 xa0 x01 bzw. 08-00-06-01-a0-01
- MAC-Adresse des Senders:
  - x08 x00 x06 x01 xa0 x00 bzw. 08-00-06-01-a0-00

[3] Man beachte, dass das im TXS verwendete Ethernet noch einer frühen Version des Standards IEEE 802.3 genügt. Ethernet-Datenpakete in neueren bzw. aktuellen Computernetzwerken sind in der Regel heute etwas anders aufgebaut.

- Länge der TXS-Daten (also der nachfolgenden „Nutzdaten“):
  - 31 Bytes (x00 x1f = 31)
- Nutzdaten, also die eigentlichen übertragenen TXS-Daten (31 Bytes):
  - `x14 x14 x03 x02 x00 xc0 x88 x05 x00 x44 x81 x0b x00 x0c x00 x1c x00 x00 x01 x02 x1f x00 x00 x65 x00 x01 x00 x00 x02 x01 x00`
  - Diese Nutzdaten enthalten über die übertragenen TXS-Daten (z. B. Werte von Variablen, etc.) hinaus auch:
    - Einen von Datenpaket zu Datenpaket hochzählenden Counter/Zähler
    - Eine Check- bzw. Prüfsumme, die über einen Teil der Nutzdaten (inkl. Counter) berechnet wird
- Angehängte „x00“, damit die gemäß Spezifikation erforderliche Gesamtlänge von 60 Bytes erreicht wird.4

Durch die gezielten Versuche (mit Wireshark /WIS 23/) und insbesondere anhand des Vergleichs unterschiedlicher Datenpakete (zu verschiedenen Zeitpunkten und für verschiedene Modellsysteme) konnte die gesamte Netzwerkkommunikation vollständig analysiert und verstanden werden. Insbesondere ermöglichte dies auch die Erstellung einer Software, mit deren Hilfe z. B. sowohl der in den Nutzdaten vorhandene Counter und die Prüfsumme korrekt berechnet und gesetzt werden können (vgl. auch nachfolgende Abschnitte 2.2 und 2.3).

Die bisher getroffenen Aussagen sind nicht nur für die interne Netzwerkkommunikation gültig. Bis auf spezielle Fälle (siehe Ende dieses Abschnitts) funktioniert die externe Netzwerkkommunikation nämlich exakt genauso. So werden beim Vorhandensein eines Servicegeräts (oder alternativ auch eines sogenannten Gateways für die „Auskopplung“ von Signalen) im Netzwerkplan der TXS-Software (in der TXS-Engineering-Umgebung

[4] In diesem Beispiel wurde nur ein vergleichsweises kleines Datenpaket übertragen. Damit das Datenpaket spezifikationsgemäß mindestens 60 Bytes lang war, wurden weitere 15 Bytes mit Nullen angehängt. Bei längeren Nutzdaten („Payloads“) ist dies nicht notwendig.

SPACE[5]), ausschließlich vom Leittechniksystem Nachrichten (unidirektional) an dieses externe Gerät gesendet (und nicht umgekehrt).

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	08:00:06:01:a0:00	00:00:00:00:00:02	Ethernet	1057	IEEE 802.3 Ethernet
2	0.000045	08:00:06:01:a0:00	08:00:06:01:a0:01	Ethernet	60	IEEE 802.3 Ethernet
3	0.050253	08:00:06:01:a0:00	00:00:00:00:00:02	Ethernet	1057	IEEE 802.3 Ethernet
4	0.050253	08:00:06:01:a0:00	08:00:06:01:a0:01	Ethernet	60	IEEE 802.3 Ethernet
5	0.100034	08:00:06:01:a0:00	00:00:00:00:00:02	Ethernet	1057	IEEE 802.3 Ethernet
6	0.100093	08:00:06:01:a0:00	08:00:06:01:a0:01	Ethernet	60	IEEE 802.3 Ethernet
7	0.150286	08:00:06:01:a0:00	00:00:00:00:00:02	Ethernet	1057	IEEE 802.3 Ethernet
8	0.150286	08:00:06:01:a0:00	08:00:06:01:a0:01	Ethernet	60	IEEE 802.3 Ethernet
9	0.200335	08:00:06:01:a0:00	00:00:00:00:00:02	Ethernet	1057	IEEE 802.3 Ethernet
10	0.200335	08:00:06:01:a0:00	08:00:06:01:a0:01	Ethernet	60	IEEE 802.3 Ethernet
11	0.250091	08:00:06:01:a0:00	00:00:00:00:00:02	Ethernet	1057	IEEE 802.3 Ethernet
12	0.250091	08:00:06:01:a0:00	08:00:06:01:a0:01	Ethernet	60	IEEE 802.3 Ethernet
13	0.300272	08:00:06:01:a0:00	00:00:00:00:00:02	Ethernet	1057	IEEE 802.3 Ethernet
14	0.300272	08:00:06:01:a0:00	08:00:06:01:a0:01	Ethernet	60	IEEE 802.3 Ethernet
15	0.350354	08:00:06:01:a0:00	00:00:00:00:00:02	Ethernet	1057	IEEE 802.3 Ethernet
16	0.350354	08:00:06:01:a0:00	08:00:06:01:a0:01	Ethernet	60	IEEE 802.3 Ethernet
17	0.400356	08:00:06:01:a0:00	00:00:00:00:00:02	Ethernet	1057	IEEE 802.3 Ethernet
18	0.400356	08:00:06:01:a0:00	08:00:06:01:a0:01	Ethernet	60	IEEE 802.3 Ethernet
19	0.450385	08:00:06:01:a0:00	00:00:00:00:00:02	Ethernet	1057	IEEE 802.3 Ethernet
20	0.450385	08:00:06:01:a0:00	08:00:06:01:a0:01	Ethernet	60	IEEE 802.3 Ethernet
21	0.500153	08:00:06:01:a0:00	00:00:00:00:00:02	Ethernet	1057	IEEE 802.3 Ethernet
22	0.500153	08:00:06:01:a0:00	08:00:06:01:a0:01	Ethernet	60	IEEE 802.3 Ethernet
23	0.550315	08:00:06:01:a0:00	00:00:00:00:00:02	Ethernet	1057	IEEE 802.3 Ethernet
24	0.550315	08:00:06:01:a0:00	08:00:06:01:a0:01	Ethernet	60	IEEE 802.3 Ethernet
25	0.600376	08:00:06:01:a0:00	00:00:00:00:00:02	Ethernet	1057	IEEE 802.3 Ethernet

Abb. 2.3 Beispiel für aus einer Redundanz verschickte Nachrichten (von AnTeS-SILT-real)

Konkret sendet hier eine Redundanz von AnTeS-SILT-real (mit der MAC-Adresse 08-00-06-01-a0-00, „Source“) abwechselnd an eine andere Redundanz von AnTeS-SILT-real (mit der MAC-Adresse 08-00-06-01-a0-01, „Destination“) und ans Servicegerät (mit der MAC-Adresse 00-00-00-00-00-02[6], „Destination“). Der Mitschnitt erfolgte mit der Software Wireshark (siehe Anhang B.7).

Repräsentativ zeigt Abb. 2.3 diesen Zusammenhang für ein konkretes Beispiel. Hier wurde von einer Redundanz des Leittechniksystems in jedem Zyklus (also alle 50 ms) eine Nachricht an eine andere Redundanz des Leittechniksystems sowie auch an das Servicegerät verschickt. Der grundsätzliche Aufbau der Datenpakete an das Servicegerät ist dabei identisch mit der weiter oben beschriebenen Struktur der Datenpakete. Es wurden lediglich deutlich mehr Informationen übertragen (hier konkret jeweils 1057 Bytes), da (z. B. für die Visualisierung im Graphischen Servicemonitor GSM) der Zustand sämtlicher Parameter (z. B. der Funktionsblöcke in den Funktionsdiagrammen)

[5] SPACE – **SP**ecification **A**nd **C**oding **E**nvironment (Entwicklungsumgebung des TXS)

[6] Im TXS sind die MAC-Adressen konfigurierbar, innerhalb von AnTeS hat das Service-Gerät standardmäßig genau diese Adresse.

in der entsprechenden Software der sendenden Redundanz an das Servicegerät übermittelt werden müssen.

Wie bereits erwähnt, erfolgt also die Kommunikation (wie auch im Beispiel) mit dem Servicegerät grundsätzlich unidirektional. Es werden insbesondere keine Anfragen oder Rückmeldungen vom Servicegerät an das Leittechniksystem übermittelt. Mit in der Software des Leittechniksystems konfiguriertem Servicegerät findet somit dauerhaft nur eine Übertragung an dieses statt (unabhängig davon ob die Informationen z. B. zur Visualisierung verwendet werden oder nicht) und nicht umgekehrt.

Eine Ausnahme hiervon ist die Übertragung von neuer/veränderter Software vom Servicegerät an das Leittechniksystem. Hierfür müssen allerdings besondere Voraussetzungen gelten, die in einem Sicherheitsleittechniksystem (z. B. Reaktorschutzsystem) sicherheitsgerichtet als nicht gegeben vorausgesetzt werden können.[7]

2.2 Fehlerinjektion in die Netzwerkkommunikation

Auf Basis der Erkenntnisse zur Netzwerkkommunikation im Testsystem (vgl. vorheriger Abschnitt 2.1) wurde eine Software entwickelt (Abb. 2.4), die auf eigens hierfür vorgesehenen Geräten (basierend auf Mikrorechnern des Typs Raspberry Pi 4, Abb. 2.5) läuft und die gezielte Fehlerinjektion in die Netzwerkkommunikation erlaubt. Gemeinsam bilden Software und Mikrorechner somit Geräte zur gezielten Beeinflussung beliebiger Netzwerkkommunikation (nachfolgend Netzwerkmanipulatoren genannt).

[7] Um das Hochladen neuer/veränderter Software über die externe Netzwerkverbindung zu ermöglichen, müssen entsprechende Jumper bzw. Switches auf der Hardware der Verarbeitungseinheiten (Prozessorkarten) des TXS gesetzt sein sowie in der laufenden Software (Funktionsdiagramme) der Verarbeitungseinheiten ein Funktionsbaustein vorhanden sein, der die entsprechenden Rechte explizit erteilt (siehe auch /MCH 21/). Standardmäßig ist das Hochladen von Software über die externe Netzwerkverbindung vollständig ausgeschlossen, stattdessen wird Software ausschließlich über eine direkte serielle Verbindung eines Servicegeräts vor Ort ermöglicht. Analoges gilt genauso auch für die bloße Veränderung von Parametern in der Software des Leittechniksystems durch das Servicegerät.

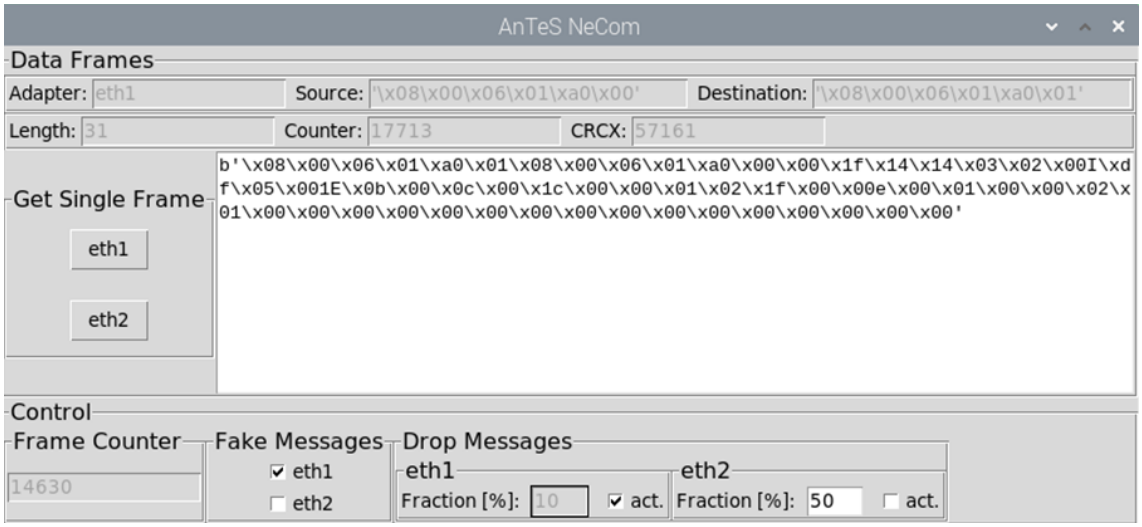

Abb. 2.4 Entwickelte Software zum Mitlesen („Sniffen“) und Beeinflussen der Netzwerkkommunikation des Testsystems

Mehr Details zu dieser Software können im Anhang A.2.2 nachgelesen werden.

Insgesamt verfügt die GRS (inklusive des Prototyps, der sich nur äußerlich von den vier Geräten in Abb. 2.5 unterscheidet) über fünf Netzwerkmanipulatoren. Jedes dieser Geräte verfügt über drei Ethernetanschlüsse, einen internen Ethernetanschluss sowie zwei über USB angeschlossene externe Netzwerkkarten. Die internen Ethernetanschlüsse dienen ausschließlich der Steuerung und Beobachtung der Netzwerkmanipulatoren, während durch die externen Ethernetanschlüsse der gesamte Netzwerkverkehr in beiden Richtungen durchgeleitet, mitgelesen und ggf. verändert werden kann.

Jeder Netzwerkmanipulator kann in jede beliebige Ethernetverbindung eines beliebigen Systems eingebracht werden. Sämtliche Datenpakete, die über das ursprüngliche Kabel verschickt worden wären, werden dann durch den Netzwerkmanipulator (in beide Richtungen) hindurchgeleitet. Zusätzlich kann dann aber auch der gesamte Netzwerkverkehr mitgelesen oder sogar gezielt beeinflusst werden.

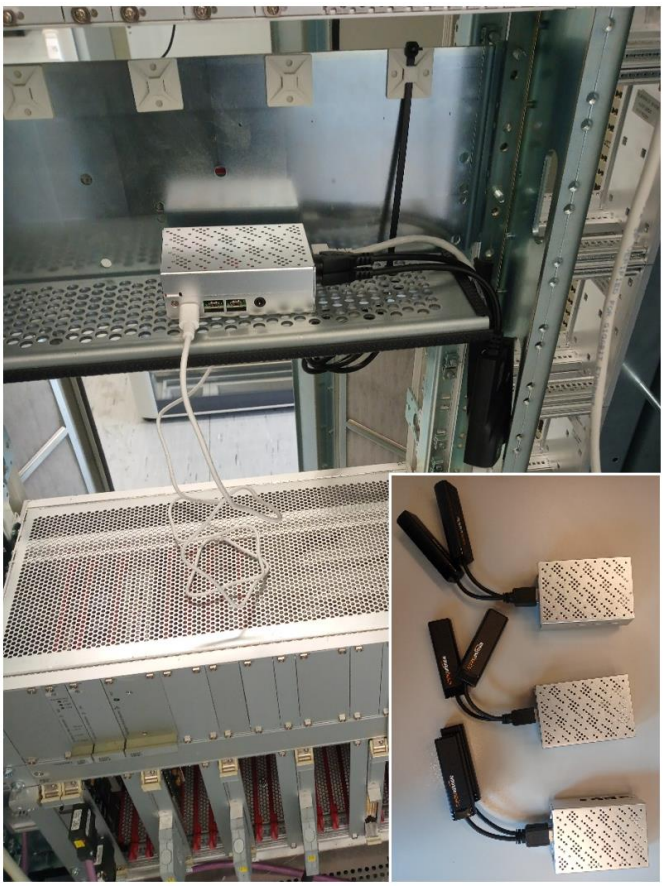

Abb. 2.5 Geräte zur Manipulation der Netzwerkkommunikation (zur Fehlerinjektion), die im Rahmen dieses Vorhabens entwickelt wurden

Neben der ungestörten Weiterleitung von Datenpaketen gibt es die folgenden Möglichkeiten zur Manipulation des Netzwerkverkehrs:

- Weiterleitung nur eines Anteils der Datenpakete (z. B. ein Anteil von x % zufällig ausgewählter Datenpakete oder nur jedes n-ten Datenpakets)
- Veränderung einer einstellbaren Anzahl von Bits jedes Datenpakets
  - Deren Positionen können entweder zufällig ausgewählt oder fest eingestellt werden
- Gezielte Erstellung eigener Datenpakete („Fake Messages“)
  - Mit Hilfe z. B. zuvor aufgezeichneter Datenpakete können so gültige Datenpakete erzeugt werden, die vom Empfänger als echt und valide bewertet werden.

- Hierfür muss insbesondere der in den Nutzdaten von TXS-Datenpaketen vorhandene Counter und die ebenfalls enthaltene Prüfsumme für jedes „Fake“-Datenpaket berechnet und korrekt gesetzt werden.

- Prinzipiell kann somit kommunikationstechnisch aus Sicht des Empfängers jeder Netzwerkmanipulator prinzipiell den „echten“ Sender vollständig ersetzen.

Im Sinne zufälliger Fehler in der Netzwerkkommunikation sind ausschließlich die ersten zwei genannten Manipulationsmöglichkeiten relevant. Die dritte Möglichkeit stellt eine Manipulationsmöglichkeit im Sinne von Cyberangriffen dar, was im nachfolgenden Exkurs kurz beleuchtet wird.

2.3 Manipulationen im Sinne von Cyberangriffen (Exkurs)

Die Netzwerkmanipulatoren erlauben auch sogenannte Man-in-the-Middle-Angriffe (MitM-Angriffe). MitM-Angriffe sind eine Form von Cyberangriffen, bei denen ein Angreifer die Kommunikation zwischen zwei Parteien abfängt, manipuliert oder sogar vollständig kontrolliert, ohne dass die beteiligten Parteien dies bemerken. Der Angreifer platziert sich quasi „in der Mitte" der Kommunikationsverbindung und kann den Datenverkehr abhören, modifizieren oder sogar gefälschte Informationen einschleusen.[8]

In eigens durchgeführten Versuchen, die hier nicht detailliert erläutert werden (da dies kein Ziel des Projektes war), wurden die Fähigkeiten der entwickelten Netzwerkmanipulatoren zur Durchführung solcher Angriffe anhand konkreter Beispiele überprüft und nachgewiesen. Entscheidend hierbei war, dass für veränderte oder komplett gefälschte Datenpakete korrekte Zähler und Prüfsummen in den TXS-Nutzdaten berechnet und gesetzt werden konnten (vgl. Abschnitt 2.1). Zusammenfassend lässt sich feststellen, dass es so beispielsweise gelungen ist, valide Datenpakete aufzuzeichnen und diese dann immer wieder (mit neu berechneten Zählern und Prüfsummen in den TXS-Nutzdaten) an

[8] MitM-Angriffe im allgemeinen Zusammenhang können z. B. in öffentlichen WLAN-Netzwerken, unsicheren Websites oder anderen unsicheren Kommunikationskanälen auftreten. MitM-Angriffe haben das Potenzial, vertrauliche Informationen zu stehlen, Passwörter zu erfassen, Finanztransaktionen zu manipulieren oder sogar die Integrität von Daten zu gefährden. Um solche Angriffe zu verhindern, ist die Verwendung von sicheren Verschlüsselungsprotokollen und das Bewusstsein für verdächtige Aktivitäten in der Kommunikation von entscheidender Bedeutung.

den Empfänger zu senden, so dass der Empfänger dieser Datenpakete unbemerkt vom realen Sender der Datenpakete komplett entkoppelt werden konnte.

3 Fehlerausbreitung in der Netzwerkkommunikation

Dieses Kapitel beschäftigt sich mit der Entwicklung, Anwendung und Validierung der um die Betrachtung von Netzwerkfehlern erweiterten Methodik der GRS zur Untersuchung digitaler Leittechniksysteme. Speziell durch die Anwendung der erweiterten Methodik auf eine Reihe von Modellsystemen (siehe Abschnitt 3.3) konnten einige allgemeingültige Rückschlüsse auf die Bedeutung und Auswirkungen von Netzwerkfehlern auf Leittechniksysteme gezogen werden.

3.1 Relevante Ausfallarten

Mit Hilfe der entwickelten Netzwerkmanipulatoren (siehe Abschnitt 2.2) und dem bei der GRS vorhandenen Leittechniksystem TXS (AnTeS-SILT-real) wurden zunächst allgemein die relevanten Ausfallarten innerhalb der Netzwerkkommunikation des Testsystems bestimmt, die anschließend bei der Methodenentwicklung berücksichtigt werden mussten. Die entsprechenden Versuche sind in Anhang A.2 detaillierter erläutert, hier werden nur die Ergebnisse dieser Versuche zusammengefasst.

Das Testsystem (TXS) erwies sich als äußerst robust hinsichtlich Ausfällen in der Netzwerkkommunikation. So können bis zu etwa 50 % der verschickten Datenpakete komplett verlorengehen oder fehlerhaft sein, ohne dass ein signifikanter Einfluss auf die Funktionalität zu beobachten ist. Dabei ist es unerheblich, ob beispielsweise genau jedes zweite Datenpaket betroffen ist oder ob rein statistisch 50 % der Datenpakete durch die Netzwerkmanipulatoren verändert oder nicht weitergeleitet werden. Erst bei noch größeren Verlustraten von Datenpaketen kann zunächst ein „flimmern“ (wechselnd zwischen scheinbar ungestörtem Verhalten und scheinbar unterbrochener Kommunikation) beobachtet werden, um schließlich bei noch höheren Verlustraten (~ 70 %) dazu zu führen, dass auf Seite des Empfängers die Kommunikation komplett als ausgefallen bewertet wird.[9]

Weitere durchgeführte Versuche zeigten, dass statistische Veränderungen der Datenpakete (z. B. durch zufällige Änderung eines oder mehrerer Bits) äußerst zuverlässig

[9] Die angegebenen Zahlen sind nur Richtwerte. Vermutlich durch den asynchronen Betrieb der beiden Redundanzen variierten die konkreten Zahlen im Einzelfall etwas zwischen den verschiedenen Versuchen.

vom Testsystem entdeckt werden. Theoretisch können zwar zufällig auch valide („gültige“) fehlerhafte Datenpakete entstehen, die dann auch zu einem fehlerhaften, unerwünschten Verhalten führen. Deren unbeabsichtigtes Entstehen ist jedoch derart unwahrscheinlich, dass diese mit hoher Wahrscheinlichkeit ausgeschlossen werden können.[10] Dabei muss nämlich die in den Nutzdaten des Datenpakets enthaltene Prüfsumme des TXS weiterhin korrekt sein, die Empfänger- und Senderadresse im Paket darf nicht verändert worden sein sowie die in den Nutzdaten enthaltenen Kennungen und Variablennamen dürfen sich nicht geändert haben.

Nimmt man beispielweise an, dass in jedem einzelnen übertragenem Datenpaket ein Bit zufällig geändert wird und sich dadurch zufällig ein einzelner übertragener Binärwert (z. B. einer übertragenen binären Variable) so ändern soll, dass zufällig weiterhin ein gültiges Datenpaket entsteht, so kann die Wahrscheinlichkeit hierfür (grob) wie folgt abgeschätzt werden:

Wahrscheinlichkeit für die Änderung eines Bits (eines Datenpakets):

$$P_{bf} = 1 \text{ (per Definition, s.o.)}$$

Wahrscheinlichkeit, dass die Änderung an der Stelle des Binärwerts auftritt:

$$P_{bp} = {}^{1}/_{480} \text{ (Mindestlänge des Pakets = 60 Bytes = 480 Bits)}$$

Wahrscheinlichkeit, dass berechnete Prüfsumme (TXS) noch stimmt[11]:

$$P_{CS} = 2 \cdot 10^{-5}$$

Gesamtwahrscheinlichkeit (für ein einzelnes Datenpaket):

$$P = P_{bf} \cdot P_{bp} \cdot P_{CS} \approx 8{,}5 \cdot 10^{-8}$$

---

[10] Man beachte, dass hier von zufälligen Fehlern die Rede ist. Bewusste Änderungen (z. B. im Sinne eines Cyberangriffs) sind hier nicht gemeint (vgl. Abschnitt 2.3).

[11] Der verwendete Wert beschreibt eigentlich die Wahrscheinlichkeit für den verwendeten Algorithmus zur Berechnung der Prüfsumme, dass zwei unterschiedliche Datenpakete zufällig die gleiche Prüfsumme ergeben (Wert aus /TXS 12/).

Bei einer (typischen) Zykluszeit (zeitlicher Abstand zweier Datenpakete) von 50 ms würde dann nur alle etwa 7 Tage ein einzelnes Paket unerkannt zufällig einen fehlerhaften Wert übertragen.[12] Alle anderen Pakete würden als fehlerhaft erkannt werden. Die Wahrscheinlichkeit, dass dies dann sogar mehrfach hintereinander auftritt, ist natürlich noch einmal wesentlich geringer.

Dies ist nur eine grobe Abschätzung und erhebt nicht den Anspruch, eine allgemeingültige Betrachtung zu sein. Dennoch kann hierdurch verdeutlicht werden, warum solche Fehler praktisch ausgeschlossen werden können.

Insgesamt kann ein einzelne Netzwerkverbindung daher bei der Modellierung nur zwei Zustände einnehmen:

- Kommunikation funktioniert
- Kommunikation ist selbstmeldend ausgefallen (wird vom Leittechniksystem detektiert)

Somit ist bei der Weiterentwicklung der Methoden nur eine einzige relevante Ausfallart für Netzwerkverbindungen zu berücksichtigen.

3.2 Methodenentwicklung und Validierung

In diesem Abschnitt wird beispielhaft anhand eines einfachen Modellsystems die Erweiterung und Validierung der Methoden der GRS zur Untersuchung von digitalen Leittechniksystemen erläutert. Ausgangspunkt ist das Modellsystem A120[13] (Abb. 3.1).

[12] Man beachte, dass hierbei jedes einzelne Datenpaket einen Fehler aufweist. Hier würde etwa alle 7 Tage vom System ein Fehler gemeldet, bevor ein einzelnes fehlerhaftes aber valides Datenpaket übertragen würde.

[13] Die verwendete Nomenklatur sowie das Modellsystem A120 wurden bereits im Vorhaben 4715R01343 /MCH 18/ entwickelt und verwendet (damals aber noch ohne explizite Fehler in den Netzwerkkommunikationswegen). Eine detailliertere Beschreibung aller in diesem Vorhaben verwendeter Modellsysteme befindet sich im Anhang A.3.

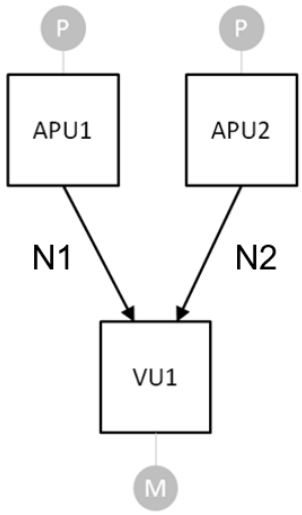

Abb. 3.1 Das Modellsystem A120

Dieses besteht aus zwei APUs (engl. Acquisition and Processing Units) sowie einer VU (engl. Voting Unit) und kann wie folgt beschrieben werden:

- Zwei Messwertgeber („P“ für Druckmessung in Abb. 3.1), bei denen vereinfachend angenommen wird, dass sie stets fehlerfrei funktionieren und ihre Signale fehlerfrei an die oberste Ebene der Leittechnik weitergeben, sind jeweils an eine APU angeschlossen.
- In den APUs werden die Eingangssignale (Messwerte) eingelesen und auf Überschreitung eines MAX-Grenzwerts überwacht. Wird der Grenzwert überschritten, so gibt die jeweilige APU eine logische „1“ aus (Anforderungsfall), andernfalls eine logische „0“.
- Die Übertragung der von den APUs erzeugten Ausgangssignale erfolgt über zwei separate Netzwerkverbindungen zwischen den APUs und der VU1 (N1 und N2 in Abb. 3.1.
- Die Voting Unit VU1 bewertet die Eingangssignale mit einer 1-von-2-Auswahl. Stehen also ein oder zwei Signale mit einer logischen „1“ am Eingang der VU1 an, so gibt sie einen Startbefehl an den angeschlossenen Motor (M) aus.
- Der angeschlossene Motor reagiert stets fehlerfrei auf die Signale der VU1.

Weiterhin wird vereinfachend angenommen, dass sowohl die APUs als auch die VU jeweils nur nichtselbstmeldend ausfallen können.[14] Deren nichtselbstmeldende Fehler (NSF) werden also nicht automatisch detektiert und können daher nur durch gezielte Tests („WKPen“ – wiederkehrenden Prüfungen) entdeckt und anschließend repariert werden.

Dementgegen sind die Fehler in der Netzwerkkommunikation grundsätzlich immer selbstmeldende Fehler (SF) – vgl. Abschnitt 3.1. Sämtliche Parameter für die vollständige Beschreibung des Modellsystems A120 sind in Tab. 3.1 wiedergegeben.[15]

Tab. 3.1 Verwendete Parameter für das Modellsystem A120

Parameter	**Beschreibung**	**Wert**
APU1.NSF	Fehlerrate für nichtselbstmeldender Ausfälle von APU1	$8 \cdot 10^{-8}\ h^{-1}$
APU2.NSF	Fehlerrate für nichtselbstmeldender Ausfälle von APU2	$8 \cdot 10^{-8}\ h^{-1}$
VU1.NSF	Fehlerrate für nichtselbstmeldende Ausfälle von VU1	$8 \cdot 10^{-8}\ h^{-1}$
N1.SF	Fehlerrate für selbstmeldende Ausfälle von N1	$2 \cdot 10^{-5}\ h^{-1}$
N2.SF	Fehlerrate für selbstmeldende Ausfälle von N1	$2 \cdot 10^{-5}\ h^{-1}$
MTTR	Reparaturzeit für sämtliche entdeckte Fehler (MTTR – Mean Time To Repair)	8 h
TI	Testintervall (also der Abstand zwischen zwei WKPen innerhalb einer Redundanz): 6 x 30 Tage = 4320 h	4320 h
TF1[1)]	Zeit bis zum ersten Test von Redundanz 1 (APU1 und VU1)	0 h
TF2[1)]	Zeit bis zum ersten Test von Redundanz 2 (APU2): 3 x 30 Tage = 2160 h	2160 h

[1] Es finden also alle drei Monate WKPen statt, abwechselnd in Redundanz 1 und Redundanz 2.

[14] Diese vereinfachte Betrachtung gilt nur für dieses Beispiel. Bei der späteren Anwendung der Methoden auf komplexere Modellsysteme wurden auch zusätzlich selbstmeldende Ausfälle aller Komponenten berücksichtigt.

[15] Die konkreten Werte sind willkürlich (wenn auch plausibel) angenommen worden. Im betrachteten Beispiel sollen die nur die grundsätzlichen Zusammenhänge erläutert werden.

Als erster Analyseschritt wird gemäß der GRS-Vorgehensweise typischerweise zunächst eine Ausfalleffektanalyse (eine erweiterte FMEA – Failure Mode and Effects Analysis) durchgeführt. In dieser werden sämtliche Kombination aller denkbaren Einzelzustände der berücksichtigten Komponenten tabellarisch erfasst und jede dieser Kombinationen hinsichtlich des Gesamtausfalls des Gesamtsystems bewertet.

Exemplarisch ist dies für das Modellsystem A120 in Tab. 3.2 (ohne Netzwerkfehler) und Tab. 3.3 (mit Netzwerkfehlern) dargestellt. Demnach erhöht sich die Anzahl der zu berücksichtigenden Kombinationen von 8 auf 32, wenn zusätzlich auch die Ausfallmöglichkeiten der Netzwerkkommunikation (N1, N2) berücksichtigt werden.

Tab. 3.2 Ausfalleffektanalyse für A120 ohne Netzwerkfehler

lfd. Nummer	APU1	APU2	VU1	Gesamtausfall
1	OK	OK	OK	nein
2	NSF	OK	OK	nein
3	OK	NSF	OK	nein
4	NSF	NSF	OK	ja
5	OK	OK	NSF	ja
6	NSF	OK	NSF	ja
7	OK	NSF	NSF	ja
8	NSF	NSF	NSF	ja
NSF – Nicht-selbstmeldender Fehler				

Tab. 3.3 Ausfalleffektanalyse für A120 mit Netzwerkfehlern

lfd. Nummer	APU1	APU2	N1	N2	VU1	Gesamtausfall
1	OK	OK	OK	OK	OK	nein
2	NSF	OK	OK	OK	OK	nein
3	OK	NSF	OK	OK	OK	nein
4	NSF	NSF	OK	OK	OK	ja
5	OK	OK	SF	OK	OK	nein
6	NSF	OK	SF	OK	OK	nein
7	OK	NSF	SF	OK	OK	ja
8	NSF	NSF	SF	OK	OK	ja
9	OK	OK	OK	SF	OK	nein
10	NSF	OK	OK	SF	OK	ja
11	OK	NSF	OK	SF	OK	nein
12	NSF	NSF	OK	SF	OK	ja
13	OK	OK	SF	SF	OK	ja
14	NSF	OK	SF	SF	OK	ja
15	OK	NSF	SF	SF	OK	ja
16	NSF	NSF	SF	SF	OK	ja
17	OK	OK	OK	OK	NSF	ja
18	NSF	OK	OK	OK	NSF	ja
19	OK	NSF	OK	OK	NSF	ja
20	NSF	NSF	OK	OK	NSF	ja
21	OK	OK	SF	OK	NSF	ja
22	NSF	OK	SF	OK	NSF	ja
23	OK	NSF	SF	OK	NSF	ja
24	NSF	NSF	SF	OK	NSF	ja
25	OK	OK	OK	SF	NSF	ja
26	NSF	OK	OK	SF	NSF	ja
27	OK	NSF	OK	SF	NSF	ja
28	NSF	NSF	OK	SF	NSF	ja
29	OK	OK	SF	SF	NSF	ja
30	NSF	OK	SF	SF	NSF	ja
31	OK	NSF	SF	SF	NSF	ja
32	NSF	NSF	SF	SF	NSF	Ja

NSF – Nicht-selbstmeldender Fehler
SF – Selbstmeldender Fehler

In komplexeren Systemen kann die Ausfalleffektanalyse einerseits die Fehlerbaumerstellung erleichtern, andererseits zumindest aber immer die korrekte Modellierung während der Fehlerbaumanalyse überprüfbar machen (da die Kombinationen, die gemäß Ausfalleffektanalyse zu einem Gesamtausfall des Systems führen, als sogenannte Minimalschnitte bei der Fehlerbaumanalyse auftauchen müssen).[16]

Wie im Bericht zum Vorhaben 4718R01314 /MCH 21/ ausgeführt, können solche Ausfalleffektanalysen auch vollautomatisch mit Hilfe von simulierten oder realen Leittechniksystemen durchgeführt werden. Hierzu wird entweder ein Simulationsmodell (mit Hilfe von Matlab/Simulink) des zu untersuchenden Leittechniksystems erstellt oder das zu untersuchende System mit Hilfe eines realen Testsystems realisiert. Anschließend werden alle denkbaren Fehlerkombinationen automatisch in den jeweiligen Modellen eingestellt (Fehlerinjektion) und dabei das Verhalten (Auslösung: ja/nein) aufgezeichnet.

Das Modellsystem A120 sieht bei der Umsetzung mit AnTeS-SILT-real (TXS) wie in Abb. 3.2 dargestellt aus (wobei zusätzlich Netzwerkmanipulatoren in die Netzwerkverbindungen zwischen den APUs und der VU eingebracht wurden). Das entsprechende reine Simulationsmodell (Matlab/Simulink) ist in Abb. 3.3 dargestellt (weitere Informationen zu den Modellsystemen sind in Anhang A.3 zu finden).

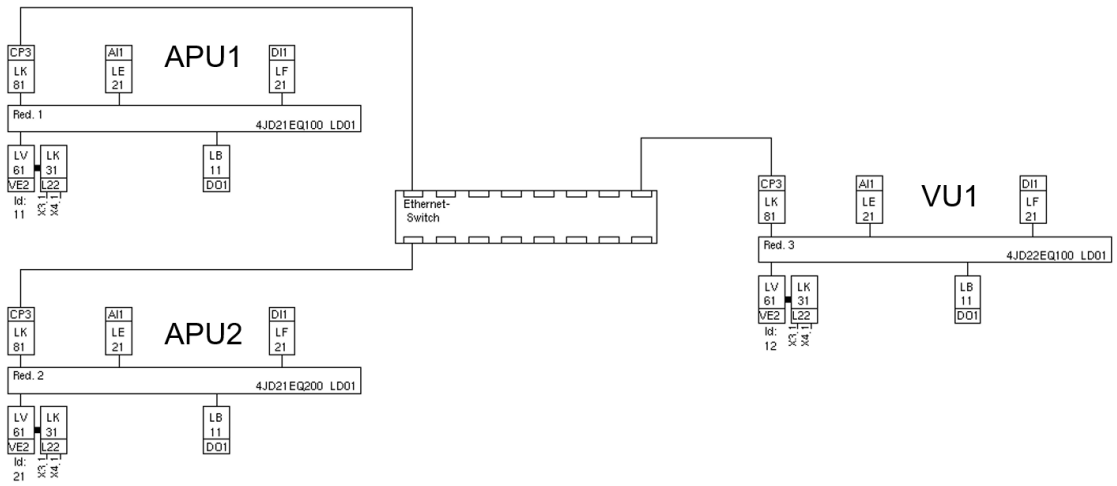

Abb. 3.2 A120 realisiert mit AnTeS-SILT-real (TXS), Netzwerkplan

[16] Man beachte, dass sich in der Regel in der Fehlerbaumanalyse weniger Minimalschnitte als Einträge in der Tabelle zu einer Ausfalleffektanalyse, die zu einem Gesamtausfall führen, ergeben. So gehören beispielsweise die lfd. Nummern 5-8 in Tab. 3.2 alle zum selben Minimalschnitt VU1 hat NSF (nicht-selbstmeldenden Fehler).

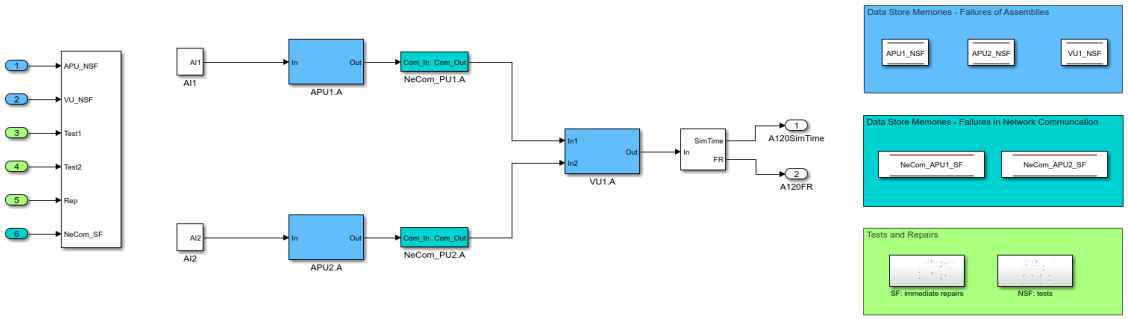

Abb. 3.3 Matlab/Simulink-Modell von A120

Beide Varianten (simuliert und real) liefern bei automatischen Ausfalleffektanalysen exakt dieselben Ergebnisse für das Modellsystem A120, wie sie in Tab. 3.2 und Tab. 3.3 gezeigt werden.

Für quantitative Analysen können für das zu betrachtende Modellsystem einerseits Fehlerbäume oder alternativ Simulationsmodelle (in Monte-Carlo-Simulationen) verwendet werden. Abb. 3.4 zeigt den (mit der Software RiskSpectrum) erstellten Fehlerbaum für das Modellsystem A120, allerdings noch ohne Basisereignisse für Ausfälle in der Netzwerkkommunikation. Werden diese zusätzlich berücksichtigt, so ergibt sich der Fehlerbaum in Abb. 3.5.

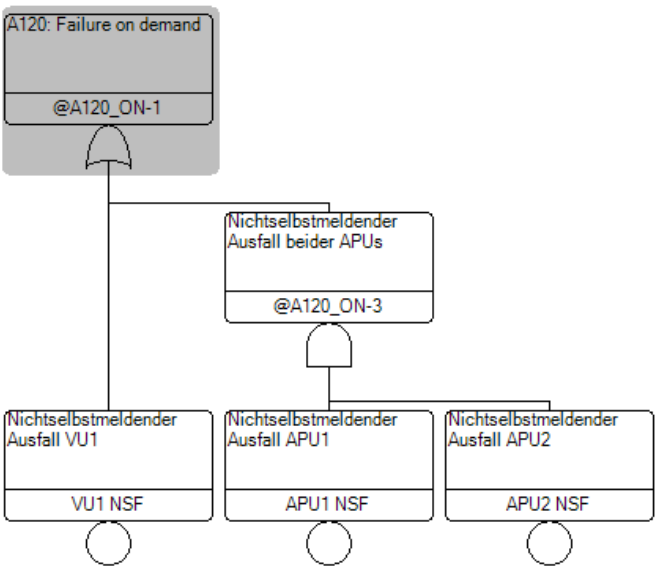

Abb. 3.4 Fehlerbaum für A120 ohne Netzwerkfehler

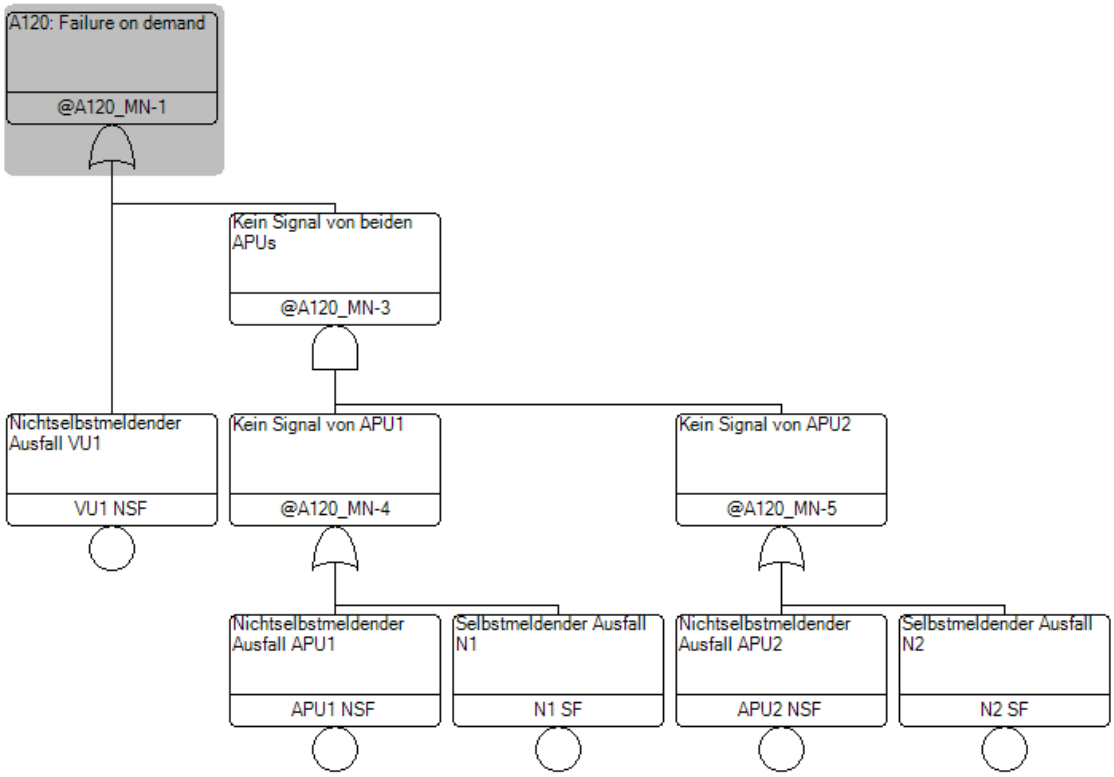

Abb. 3.5 Fehlerbaum für A120 mit Netzwerkfehlern (N1, N2)

Die Minimalschnitte[17] (engl. Minimal Cut Sets – MCS) zu den beiden Fehlerbäumen für das Modellsystem A120 (mit und ohne Berücksichtigung von Netzwerkfehlern) sind in Tab. 3.4 bzw. Tab. 3.5 wiedergegeben.

Tab. 3.4 Minimalschnitte für das Modellsystem A120 (ohne Netzwerkfehler)

No	Probability	%	Event 1	Event 2
1	1,73E-04	99,98	VU1 NSF	
2	3,01E-08	0,02	APU1 NSF	APU2 NSF

[17] Minimal Cut Sets (MCS) oder Minimalschnitte sind Kombinationen von Fehlerursachen, bei denen jede einzelne Ursache notwendig ist, um das unerwünschte Ereignis zu verursachen. Wenn auch nur eine dieser Ursachen entfernt wird, kann das Top-Ereignis nicht mehr eintreten.

Tab. 3.5 Minimalschnitte für das Modellsystem A120 (mit Netzwerkfehlern)

No	Probability	%	Event 1	Event 2
1	1,73E-04	99,98	VU1 NSF	
2	3,01E-08	0,02	APU1 NSF	APU2 NSF
3	2,77E-08	0,02	N1 SF	APU2 NSF
4	2,77E-08	0,02	APU1 NSF	N2 SF
5	2,56E-08	0,01	N1 SF	N2 SF

Die mit Hilfe der Fehlerbäume bestimmten Ausfallkombinationen, die zu einem Totalausfall des Systems führen („Minimalschnitte“, Tab. 3.4 und Tab. 3.5), bestätigen sowohl die händisch durchgeführten als auch die mit Hilfe der automatischen Auswirkungsanalyse gewonnenen Ergebnisse (Tab. 3.2 und Tab. 3.3).[18]

Der repräsentative Vergleich der quantitativen Ergebnisse der Fehlerbaumanalyse mit der Monte-Carlo-Simulation für das Modellsystem A120 (mit Netzwerkfehlern) in Abb. 3.6 zeigt, dass beide Methoden übereinstimmende Werte für die Wahrscheinlichkeit eines Totalausfalls („failure on demand“) liefern. Nach etwa 100.000 simulierten Jahren (~ 900 Mio. „Wiederholungen“ im Sinne einer Monte-Carlo-Simulation) pendelt sich die berechnete mittlere Wahrscheinlichkeit gut beim durch eine Fehlerbaumanalyse erhaltenen Wert ein.

[18] Man beachte hierbei auch die Fußnote 16.

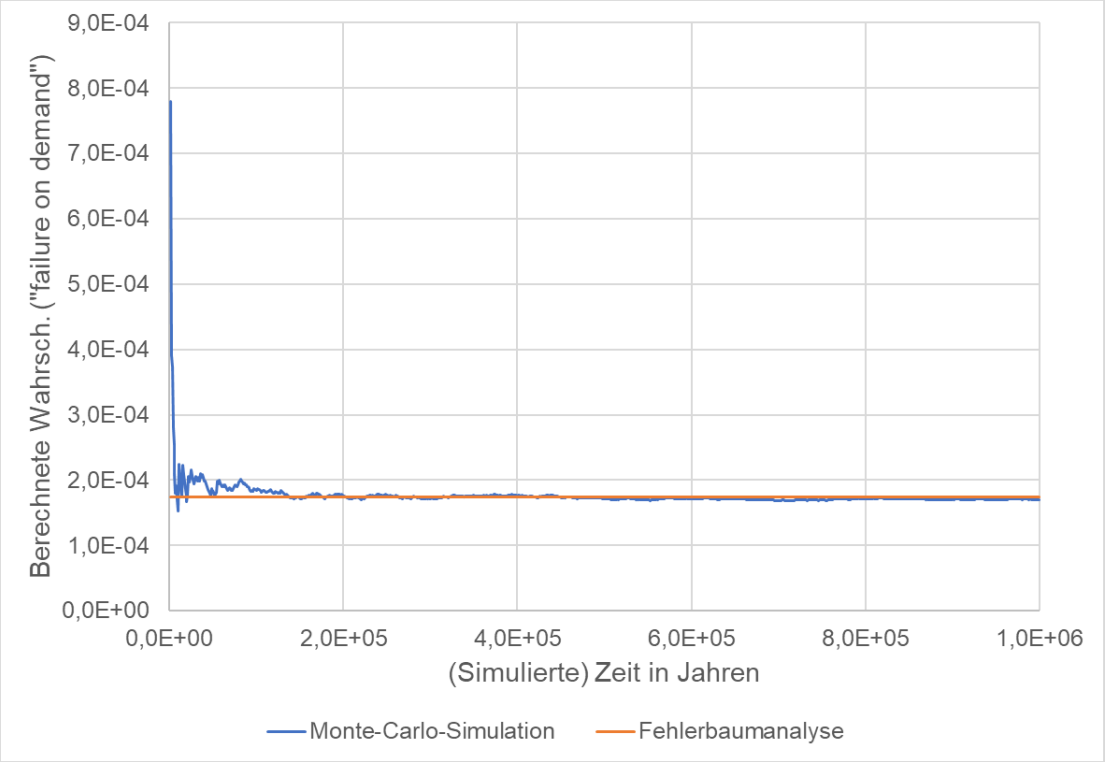

Abb. 3.6 A120, Vergleich Fehlerbaumanalyse mit Monte-Carlo-Simulation

Details zu diesen Arten von Simulationen können auch in /MCH 21/ nachgelesen werden.

3.3 Analysen von Modellsystemen

Wie im vorausgegangenen Abschnitt für ein einfaches Modellsystem (A120) demonstriert, wurden ausführliche Analysen für eine Reihe repräsentativer, zunehmend komplexere Modellsysteme durchgeführt:

- A122
  - 1 Voting Unit (VU), 2 Processing Units (PU), 2 Acquisition Units (AU) der Leittechnikplattform A
- A122mod
  - 1 Voting Unit (VU), 2 Processing Units (PU), 2 Acquisition Units (AU) der Leittechnikplattform A
  - Aus A122 gewonnen durch die Veränderung eines einzigen Funktionsbausteins. Details hierzu können /MCH 21/ entnommen werden.

- A222
  - 2 Voting Units (VU), 2 Processing Units (PU), 2 Acquisition Units (AU) der Leittechnikplattform A
- A222mod
  - 1 Voting Unit (VU), 2 Processing Units (PU), 2 Acquisition Units (AU) der Leittechnikplattform A
  - Aus A222 gewonnen durch die Veränderung zweier Funktionsbausteine. Details hierzu können /MCH 21/ entnommen werden.
- A133
  - 1 Voting Unit (VU), 3 Processing Units (PU), 3 Acquisition Units (AU) der Leittechnikplattform A
- A333
  - Voting Units (VU), 3 Processing Units (PU), 3 Acquisition Units (AU) der Leittechnikplattform A
- A133B133
  - 1 Voting Unit (VU), 3 Processing Units (PU), 3 Acquisition Units (AU) der Leittechnikplattform A und 1 Voting Unit (VU), 3 Processing Units (PU), 3 Acquisition Units (AU) der Leittechnikplattform B
  - Die Leittechnikplattformen A und B sind diversitär zueinander
- A333B333
  - Voting Units (VU), 3 Processing Units (PU), 3 Acquisition Units (AU) der Leittechnikplattform A und 3 Voting Units (VU), 3 Processing Units (PU), 3 Acquisition Units (AU) der Leittechnikplattform B
  - Die Leittechnikplattformen A und B sind diversitär zueinander

Genauere Beschreibungen der Modellsysteme befinden sich im Anhang A.3. Für die nachfolgenden Betrachtungen ist jedoch nur wesentlich, dass die aufgelisteten Modellsysteme von oben nach unten tendenziell komplexer werden (und auch mehr Redundanzen haben) sowie dass die letzten beiden Modellsysteme darüber hinaus auch Diversitäten (Teilsysteme A und B) aufweisen.

Als Zuverlässigkeitskenndaten wurden Parameter in Anlehnung an /MCH 18/ und /MCH 21/ verwendet (siehe Tab. 3.6).

Tab. 3.6 Verwendete Parameter für selbstmeldende und nicht-selbstmeldend Ausfälle sowie GVAs der Module der Modellsysteme (vgl. auch /MCH 21/)

Parameter	**FR in Anlehnung an GRS-494**	**Anmerkungen**
FR AL NSF	1E-10 h^{-1}	Analoger Voter
FR AU SF	2,10E-05 h^{-1}	
FR AU NSF	8,26E-08 h^{-1}	
FR PU SF	1,57E-05 h^{-1}	
FR PU NSF	8,26E-08 h^{-1}	
FR VU SF	6,97E-06 h^{-1}	
FR VU NSF	8,26E-08 h^{-1}	
FR NeCom SF	1,00E-04 h^{-1}	Einzelfehler, kein GVA[19]
FR AU CCF	4,35E-09 h^{-1}	GVA aller AU in einem Teilsystem
FR PU CCF	4,35E-09 h^{-1}	GVA aller PU in einem Teilsystem
FR VU CCF	4,35E-09 h^{-1}	GVA aller VU in einem Teilsystem

Da zusätzlich jetzt jedoch auch explizite Ausfälle in den Netzwerkverbindungen berücksichtigt wurden, musste für deren Fehlerrate ein Wert angenommen werden. Als erste Näherung wurde ein Wert von (willkürlich, aber dafür vergleichsweise groß) 1,0E-04 h^{-1} festgelegt. Da sich diese Annahme jedoch nicht ohne weiteres zweifelsfrei begründen lässt, wurden stattdessen Sensitivitätsanalysen genau zu diesem Parameter durchgeführt.

Variiert man die Fehlerrate für Netzwerkausfälle bei ansonsten gleichbleibenden Parametern, ergibt sich beispielsweise der in Abb. 3.7 für das Modellsystem A122mod dargestellte Zusammenhang.

[19] Da sämtliche Ausfälle der Netzwerkverbindungen grundsätzlich selbstmeldend sind, müssen hier GVAs nicht explizit betrachtet werden. Jeder auftretende Fehler wird unmittelbar entdeckt und innerhalb der Reparaturzeit behoben. Typischerweise sind GVA aber nur unentdeckt relevant (also für NSF).

Zusätzlich zeigt diese Abbildung auch, wie gut die mit Monte-Carlo-Simulationen gewonnenen Ergebnisse mit den entsprechenden Fehlerbaumanalysen übereinstimmen.

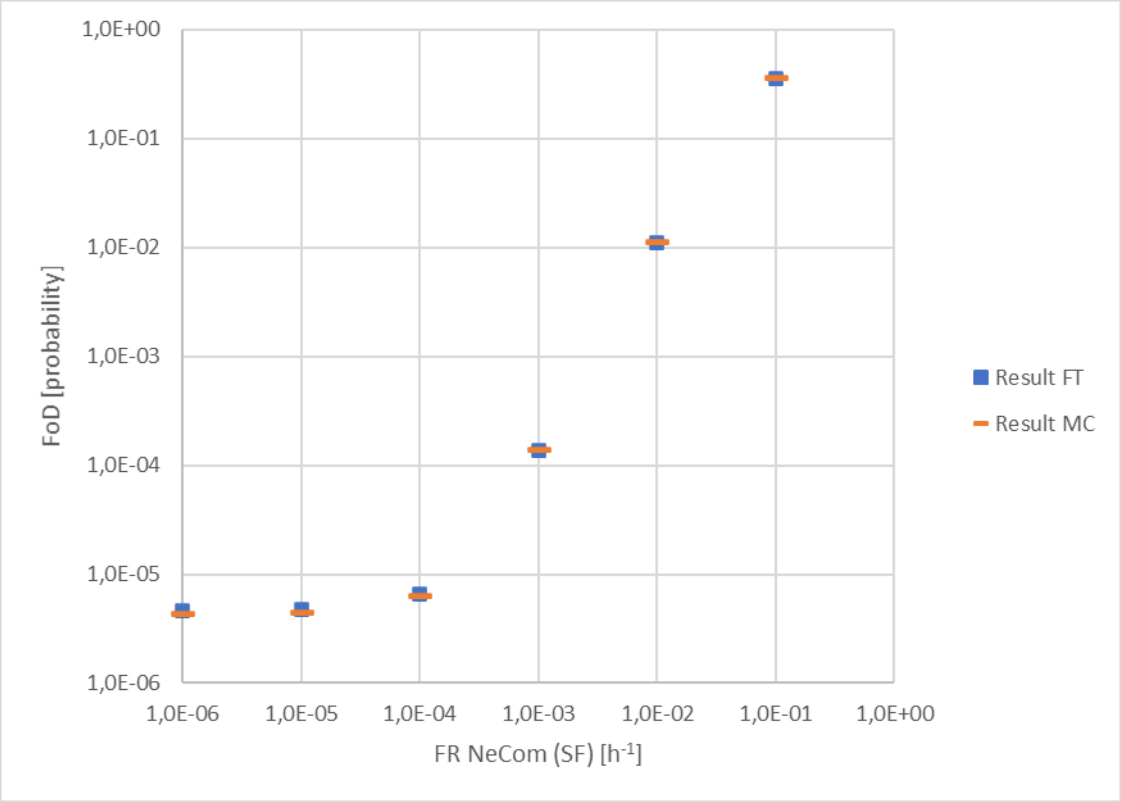

Abb. 3.7 Sensitivitätsanalyse: Einfluss der Fehlerrate in der Netzwerkkommunikation (FR NeCom) auf „failures on demand“ (FoD) für A222mod

Fehlerbaumanalysen (FT) und Monte-Carlo-Simulationen (MC)

Entsprechende Analysen wurden für alle Modellsysteme durchgeführt. Die Ergebnisse der Sensitivitätsanalysen hinsichtlich der Fehlerrate der Netzwerkkommunikation für alle Modellsysteme ist gemeinsam in Abb. 3.8 dargestellt. In dieser Darstellung ist einerseits unmittelbar die höhere Zuverlässigkeit von Systemen mit mehr Redundanzen und insbesondere mit Diversitäten ersichtlich. So ist beispielsweise das Modellsystem A333B333 um mehr als sechs Größenordnungen zuverlässiger als das Modellsystem A222.

Andererseits vermittelt das Diagramm in Abb. 3.8 auch einen Eindruck davon, wie sich Ausfälle der Netzwerkkommunikation bei unterschiedlich angenommenen Fehlerraten auf die Modellsysteme auswirken.

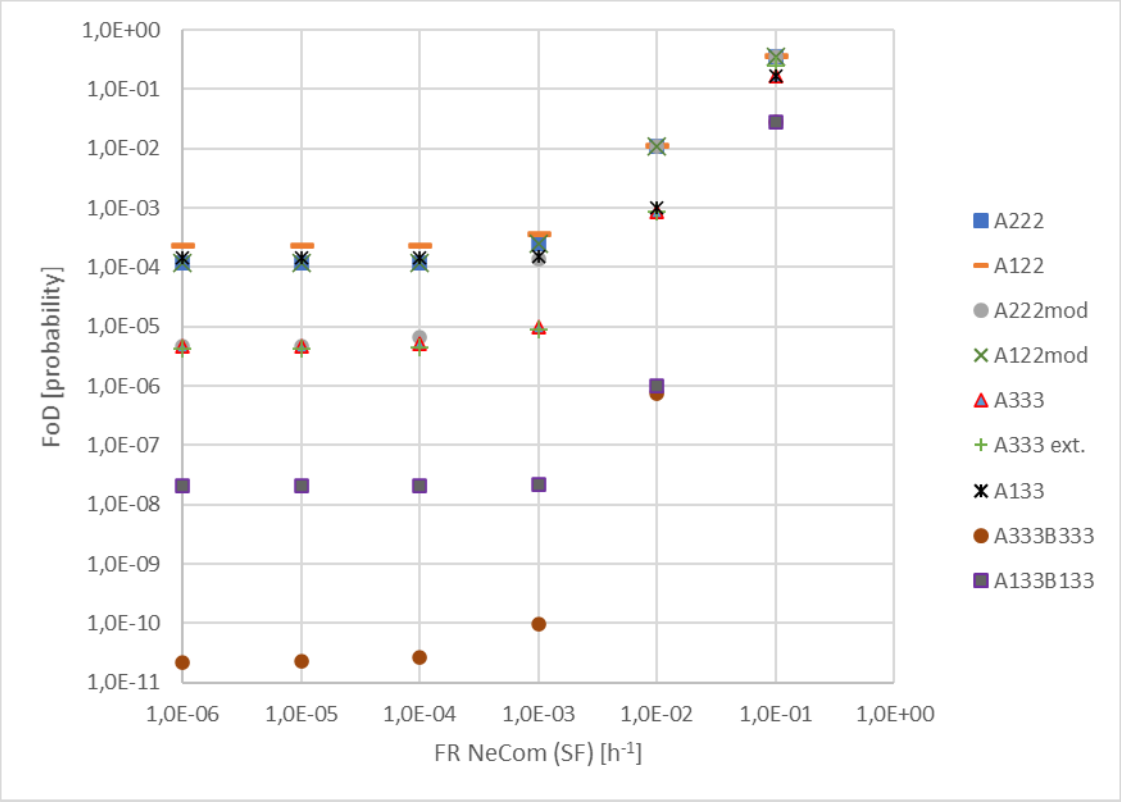

Abb. 3.8 Sensitivitätsanalysen: Einfluss der Fehlerrate in der Netzwerkkommunikation (FR NeCom) auf „failures on demand“ (FoD) für alle Modellsysteme (I)

Die entsprechenden Zusammenhänge lassen sich noch klarer durch Normierung der Ergebnisse visualisieren. Wird die Wahrscheinlichkeit für einen Ausfall bei Anforderung („failure on demand“) für alle Modellsysteme und eine Fehlerrate der Netzwerkkommunikation FR NeCom = $1 \cdot 10^{-6}$ h^{-1} auf 1 normiert, so ergibt sich das in Abb. 3.9 gezeigte Diagramm.

Abb. 3.9 zeigt eindrucksvoll, dass sich Fehler in der Netzwerkkommunikation überhaupt erst bei Fehlerraten über $1 \cdot 10^{-4}$ h^{-1} signifikant auswirken. Unterhalb dieses Wertes ist so gut wie kein Einfluss beobachtbar. Oberhalb dieses Wertes verhalten sich die Modellsysteme ähnlich, wobei der relative Einfluss für tendenziell zuverlässigere Modellsysteme etwas höher liegt (durch die größere Anzahl von Netzwerkverbindungen innerhalb dieser Systeme).

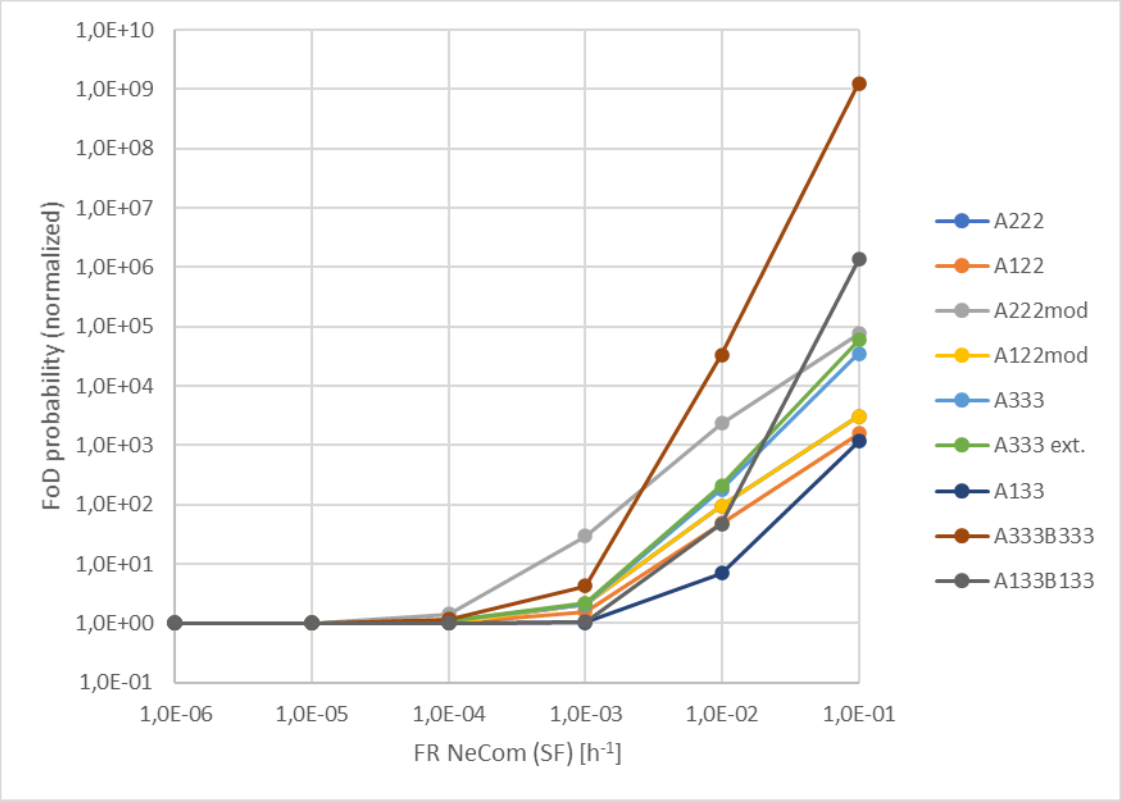

Abb. 3.9 Sensitivitätsanalysen: Einfluss der Fehlerrate in der Netzwerkkommunikation (FR NeCom) auf „failures on demand“ (FoD) für alle Modellsysteme (II)

Wie Abb. 3.8, nur wurden die Wahrscheinlichkeiten für einen „failure on demand“ für FR NeCom = $1{,}0 \cdot 10^{-06}$ h^{-1} auf 1 normiert

Um einen signifikanten Einfluss auf die Zuverlässigkeit der Modellsysteme zu haben, müssen also vergleichsweise große Fehlerraten für die Netzwerkkommunikation angenommen werden. Diese liegen dann mehr als eine Größenordnung höher als die Fehlerraten anderer selbstmeldender Ausfälle (vgl. Tab. 3.1). Bei noch größeren angenommenen Fehlerraten werden die Ausfälle in der Netzwerkkommunikation zwar schließlich dominant (siehe Abb. 3.8), entsprechende Beobachtungen in realen digitalen Systemen wurden aber nicht gemacht. Insgesamt ist also davon auszugehen, dass die ursprünglich angenommene Fehlerrate bereits mehr als ausreichend konservativ ist.

Insgesamt kann dementsprechend geschlussfolgert werden, dass Ausfälle in der Netzwerkkommunikation nur einen äußerst geringen Einfluss auf die Gesamtzuverlässigkeit der Leittechniksysteme haben.

4 Zusammenfassung und Gesamtergebnis

Ziel des Vorhabens war die Untersuchung von Fehlern in der Netzwerkkommunikation digitaler Leittechniksysteme, um diese künftig in den von der GRS verwendeten Methoden explizit und somit genauer berücksichtigen zu können.

Hierzu wurden zunächst mit Hilfe eines Testsystems, einem realen Leittechniksystem basierend auf Komponenten und Software der Plattform Teleperm XS von Framatome, diverse Versuche durchgeführt. Dadurch gelang es die Netzwerkkommunikation innerhalb des Testsystems so gut zu verstehen, dass im Rahmen des Projekts sogenannte Netzwerkmanipulatoren entwickelt werden konnten. Diese erlauben nicht nur das Mitlesen des gesamten Netzwerkverkehrs jeder beliebigen Netzwerkverbindung, sondern es können im Sinne von Fehlerinjektionen auch vielfältige Manipulationen am Netzwerkverkehr vorgenommen werden.

Es zeigte sich, dass das repräsentative Testsystem äußerst robust gegen zufällig auftretende Fehler in den Netzwerkverbindungen ist. So ist nicht davon auszugehen, dass Fehler unerkannt in Netzwerken auftreten, da sämtliche Fehler grundsätzlich selbstmeldend sind.[20] Somit ist grundlegend auch nur von einer relevanten Ausfallart in Netzwerkverbindungen auszugehen, die bei der Modellierung von Systemen berücksichtigt werden muss. Entsprechende Ausfälle wurden daher als zusätzliche Ausfallarten (neben den schon in der Vergangenheit berücksichtigen Ausfällen, z. B. von Hardwarekomponenten) in bereits vorhandene und neu erstellte Modellsysteme integriert.

Anschließend wurden Fehler, die in der Netzwerkkommunikation auftreten können, sowie deren Ausbreitung und Auswirkungen in einer Reihe komplexer werdender Modellsysteme genauer untersucht. Hierzu wurden neben Fehlerbaumanalysen und Monte-Carlo-Simulationen auch konkrete Umsetzungen einzelner Modellsysteme mit realen Leittechnikkomponenten verwendet.

[20] Dies schließt allerdings nicht aus, dass z. B. durch fehlerhafte Planung dennoch Fehler unentdeckt bleiben. Zwar ist jeder Fehler prinzipiell selbstmeldend, er muss aber auch entsprechend verarbeitet bzw. registriert werden. Eine Analogie in der Welt festverdrahteter Leittechniksysteme wäre es, wenn ein Fehler zwar prinzipiell immer detektiert werden könnte, hierfür aber keine Meldung (z. B. ein Meldeschlitz) auf der Warte vorhanden ist. Solche Fehler wurden in diesem Projekt nicht zusätzlich betrachtet.

Da die konkrete Wahrscheinlichkeit des Auftretens von Fehlern in der Netzwerkkommunikation nicht bekannt ist, wurden stattdessen entsprechende Sensitivitätsanalysen durchgeführt. Es zeigte sich, dass ein signifikanter Einfluss auf die Zuverlässigkeit sämtlicher Modellsysteme erst für recht große angenommene Fehlerraten[21] zu beobachten ist. Insgesamt sind daher die (zusätzlichen) Auswirkungen bei expliziter Berücksichtigung von Netzwerkfehlern als gering einzustufen, was vor allem auch daran liegt, dass diese grundsätzlich selbstmeldend sind.[22] Dennoch können und werden diese nach Abschluss dieses Projekts ein fester Bestandteil der GRS-Methodologie.

Daneben konnte im Rahmen dieses Projekts, auch wenn dies kein ursprüngliches Ziel war, aufgezeigt werden, wie die entwickelten Netzwerkmanipulatoren zur Durchführung (bzw. Nachstellung) von Cyberangriffen auf das Leittechniksystem verwendet werden können. Hierfür ist zwar ein physischer Zugriff auf die anzugreifende Netzwerkverbindung notwendig, dennoch erscheinen weitere Untersuchungen in dieser Richtung in nachfolgenden Projekten lohnend.

Eine weitere zukünftige Ausweitung der in diesem Projekt durchgeführten Analysen, könnten noch Hardware-nähere Untersuchungen sein. So wurden die Netzwerknachrichten sowohl bei Verwendung der Software Wireshark als auch in den Netzwerkmanipulatoren auf der Ethernet-Ebene betrachtet. Dies ist die unterste Ebene, die innerhalb der Rechner (also auch der Netzwerkmanipulatoren) erreichbar ist. Darunter befindet sich nur noch der physikalische Layer (also wirklich Spannungen/Ströme im digitalen Netzwerk). Z. B. durch Verwendung von Logic Analyzers wären noch tiefere Analysen auf dieser Ebene denkbar.

---

[21] Signifikante Auswirkungen sind erst zu beobachten, wenn für die Fehlerraten in der Netzwerkkommunikation Werte angenommen werden, die mindesten eine Größenordnung über den bisher größten angenommenen Fehlerraten anderer Komponenten liegen.

[22] Auch andere selbstmeldende Fehler haben tendenziell geringere Auswirkungen auf das Gesamtsystem als nicht-selbstmeldende Fehler (siehe z. B. /MCH 21/).

Referenzen

/BMU 15/ Bundesministerium für Umwelt, Naturschutz, Bau und Reaktorsicherheit: *Änderungen und Neufassung der Bekanntmachung zu den „Sicherheitsanforderungen an Kernkraftwerke“*, BAnz AT 30.03.2015 B2, März 2015

/BMU 15a/ Bundesministerium für Umwelt, Naturschutz, Bau und Reaktorsicherheit: *Bekanntmachung der Interpretationen zu den Sicherheitsanforderungen an Kernkraftwerke*, BAnz AT 30.03.2015 B3, März 2015

/CRC 23/ https://crccalc.com/, zuletzt abgerufen am 30.08.2023

/FRA 23/ Gespräche mit Mitarbeiten von Framatome während deren Besuchs bei der GRS am 22.03.2023

/GEE 22/ Geeks for Geeks: *OSI Model Full Form in Computer Networking*, https://www.geeksforgeeks.org/osi-model-full-form-in-computer-networking/, zuletzt abgerufen am 17.04.2023

/GRS 23/ Bei der GRS laufendes BMUV-Vorhaben 4722R01215 („*AnTeS-PRIO*“)

/ISO 94/ ISO/IEC 7498-1:1994: *Information technology - Open Systems Interconnection - Basic Reference Model: The Basic Model*, siehe hierzu z. B. https://www.iso.org/standard/20269.html, zuletzt abgerufen am 17.04.2023

/MAT 23/ https://de.mathworks.com/products/simulink.html, zuletzt abgerufen am 31.08.2023

/MCH 18/ C. Müller, J. Peschke, E. Piljugin: *Entwicklung und Erprobung eines Werkzeugs zur Sensitivitätsanalyse der Fehlerauswirkungen in der sicherheits-relevanten digitalen Leittechnik*, GRS-494, März 2018

/MCH 21/ C. Müller, E. Piljugin, P. Gebhardt, J. Shvab: *AnTeS – Entwicklung und Anwendung des Analyse- und Testsystem der GRS*, GRS-648, März 2021

/TXS 12/ Areva (Anm.: heute Framatome): *Teleperm XS User Manual, TXS Core-SW Release 3.6.5*, PTLSD-G/2012/en/0217 A (Restricted), 2012

/WIK 23/ https://de.wikipedia.org/wiki/OSI-Modell, zuletzt abgerufen am 17.04.2023

/WIK 23a/ https://de.wikipedia.org/wiki/Single_Event_Upset, zuletzt abgerufen am 30.08.2023

/WIK 23b/ https://de.wikipedia.org/wiki/Latch-up-Effekt, zuletzt abgerufen am 30.08.2023

/WIS 23/ https://www.wireshark.org/, zuletzt abgerufen am 17.08.2023

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

AnTeS	Analyse- und Testsystem der GRS für (digitale) Leittechnik
AnTeS-BELT	AnTeS-Modul BELT (betriebliche Leittechnik)
AnTeS-FRONT	AnTeS-Modul FRONT (Front-Line-Systeme)
AnTeS-PRIO	AnTeS-Modul PRIO (Prioritätsmodule)
AnTeS-SILT	AnTeS-Modul SILT (Sicherheitsleittechnik)
APU	Acquisition and Processing Unit (kombinierte AU und PU) innerhalb eines Leittechniksystems
AU	Acquisition Unit (Erfassungseinheit/-rechner) innerhalb eines Leittechniksystems
AV42	Konkretes Priority Actuation and Control Module
BELT	Betriebliche Leittechnik
BMU	Bundesministerium für Umwelt, Naturschutz und nukleare Sicherheit (bis 2021)
BMUV	Bundesministerium für Umwelt, Naturschutz, nukleare Sicherheit und Verbraucherschutz (ab 2021)
CCF	Common Cause Failure (engl.) – auch GVA genannt
(S)CP3	Kommunikationsprozessor (Communication Processor) des TXS
CRC	Cyclic Redundancy Check (engl.) – auch zyklische Redundanzprüfung genannt: Verfahren zur Bestimmung eines Prüfwerts für Daten, um Fehler bei der Übertragung oder Speicherung erkennen zu können. Wird auch für die Bezeichnung des Prüfwertes verwendet.
DLL	Dynamic Link Library
FMEA	Failure Mode and Effects Analysis
GRS	Gesellschaft für Anlagen- und Reaktorsicherheit gGmbH
GSM	Graphischer Servicemonitor (Software) des TXS
GVA	Gemeinsam verursachter Ausfall (bzw. Ausfall aufgrund gemeinsamer Ursache)
IEEE	Institute of Electrical and Electronics Engineers
ISO	International Organization for Standardization

MitM	Man-in-the-Middle
MTTR	Mean Time To Repair (mittlere Reparaturzeit in Fehlerbaumanalysen)
NSF	Nicht-Selbstmeldender Fehler
OSI	Open Systems Interconnection
PU	Processing Unit (Prozesseinheit/-rechner) innerhalb eines Leittechniksystems
SEL	Single Event Latchup
SEU	Single Event Upset
SF	Selbstmeldender Fehler
SILT	Sicherheitsleittechnik
SimGen	Simulation Generator, eine von der GRS im Rahmen des Vorhabens 4718R01314 entwickelte Software zur flexiblen Simulation von verfahrenstechnischen Systemen
SPACE	Specification And Coding Environment (Entwicklungsumgebung des TXS)
SPLM1	Konkretes Priority Actuation and Control Module (festverdrahtet) von TXS
TF	Time to First test (Zeit bis zur ersten WKP) in Fehlerbaumanalysen
TI	Testintervall (von WKPen) in Fehlerbaumanalysen
TXS	Leittechnikplattform Teleperm XS der Firma Framatome
USB	Universal Serial Bus
(S)VE2	Verarbeitungseinheit (Prozessorkarte) des TXS
VU	Voting Unit (Bewertungseinheit/-rechner) innerhalb eines Leittechniksystems
WKP(en)	Wiederkehrende Prüfung(en)

A Detailliertere und zusätzliche Beschreibungen

A.1 TXS2Simulink

Häufig müssen im Rahmen von Untersuchungen und Analysen Modellsysteme sowohl mit AnTeS-SILT-real (reales Leittechniksystem TXS) als auch mit AnTeS-SILT-sim (simuliertes Leittechniksystem) gleichermaßen betrachtet werden. In diesen Fällen müssen normalerweise die leittechnischen Funktionen in Form von Funktionsdiagrammen sowohl für das reale als auch das simulierte System händisch erstellt werden. Im Rahmen des Vorhabens /MCH 21/ konnte in einem Proof-of-Concept jedoch nachgewiesen werden, dass prinzipiell auch die Möglichkeit besteht, Funktionspläne des realen leittechnischen Systems (TXS) automatisch in Matlab/Simulink-Dateien (simuliertes leittechnisches System) zu übersetzen.

Beim Engineering-Prozess („Programmierung“) des TXS werden leittechnische Funktionen mit Hilfe von SPACE („Specification And Coding Environment“) graphisch als Funktionspläne erstellt. Bevor diese in das eigentliche Leittechniksystem hochgeladen werden können, findet ein zweistufiger Prozess[23] statt, bei dem die erstellten Funktionen zunächst in C-Files[24] übersetzt und anschließend kompiliert werden. Diese automatisch erstellen C-Files verfügen über eine gute automatisch erstellte Kommentierung, so dass deren Analyse vergleichsweise einfach möglich war.

Auf dieser Basis wurde im Rahmen dieses Projekts die Software TXS2Simulink (in der Programmiersprache Python) erstellt, welches TXS-Funktionspläne (beschrieben durch C-Files) automatisch in Simulink-Modelle übersetzt. Diese Software wird nachfolgend kurz vorgestellt.

---

[23] In der der GRS zur Verfügung stehenden Version des TXS werden beide Schritte (Übersetzung in C-Files und Kompilierung) einzeln durch den Programmierer angestoßen. In neueren Versionen sind beide Schritte gemeinsam durch einen einzelnen Programmaufruf erreichbar, die prinzipielle Vorgehensweise im TXS hat sich hierbei jedoch nicht geändert.

[24] Programmcode in der Programmiersprache C.

Abb. A 1 Startfenster von TXS2Simulink, Screenshot

Abb. A 1 zeigt einen Screenshot von TXS2Simulink unmittelbar nach dem Start der Software. Öffnet man über das Menü („File“ → „Open TXS File“) das C-File eines TXS-Funktionsplans, wird dieser im Fenster graphisch dargestellt (Abb. A 2). Durch die anschließende Auswahl von „Make Simulink File“ im Menü kann anschließend der geöffnete TXS-Funktionsplan automatisch in ein Matlab/Simulink-Modell übersetzt werden. Hierbei werden zusätzliche Informationen zum umgewandelten TXS-Funktionsplan auf der Konsole des Rechners ausgegeben (Abb. A 3):

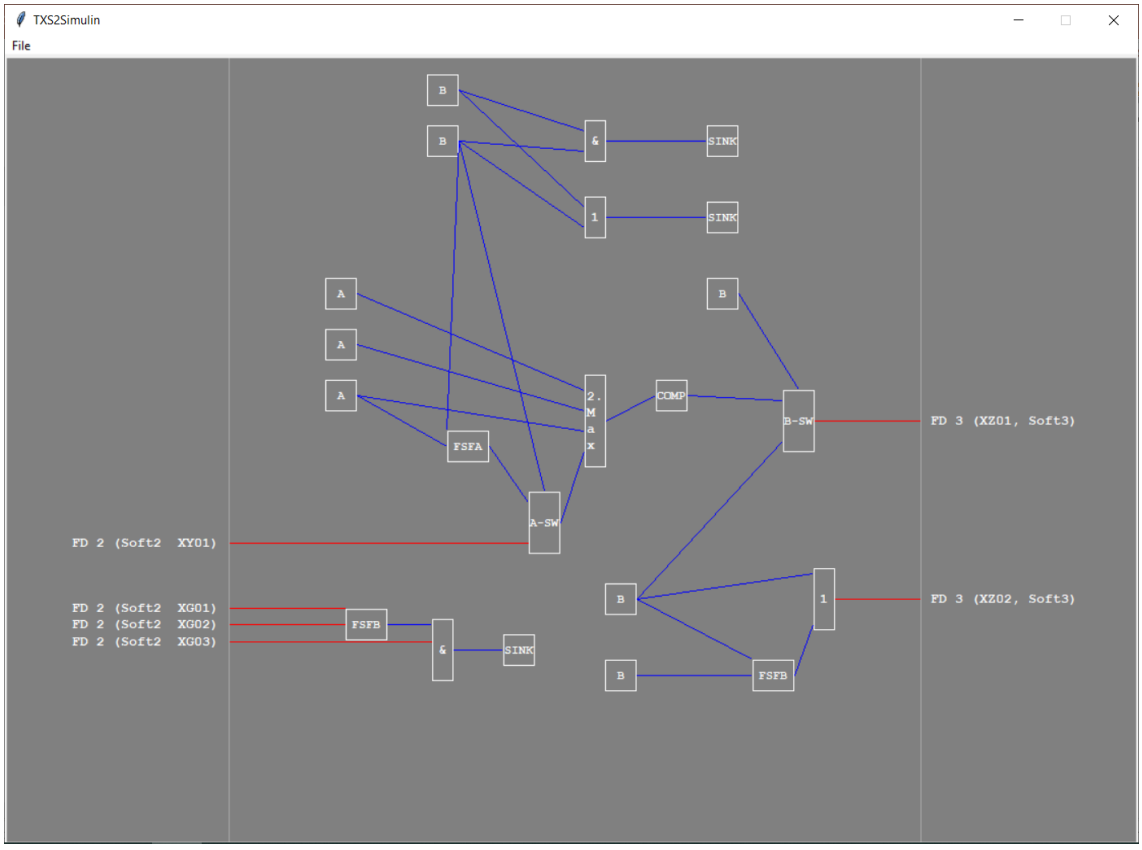

Abb. A 2 Darstellung eines TXS-Funktionsplans in TXS2Simulink

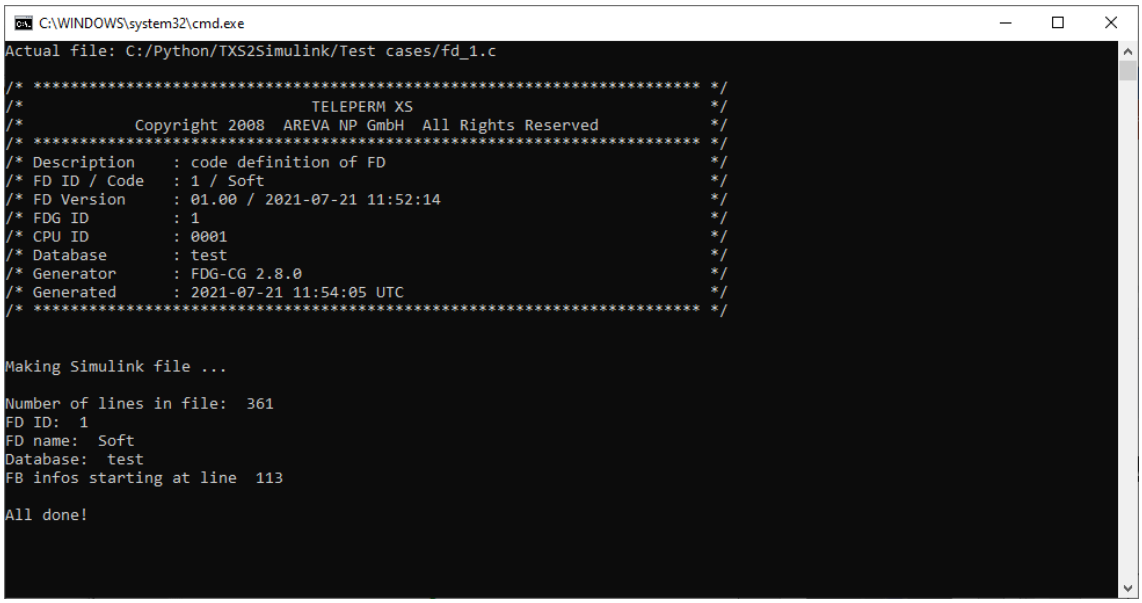

Abb. A 3 Konsolenausgabe von TXS2Simulink bei der Umwandlung eines TXS-Funktionsplans in ein Matlab/Simulink-Modell

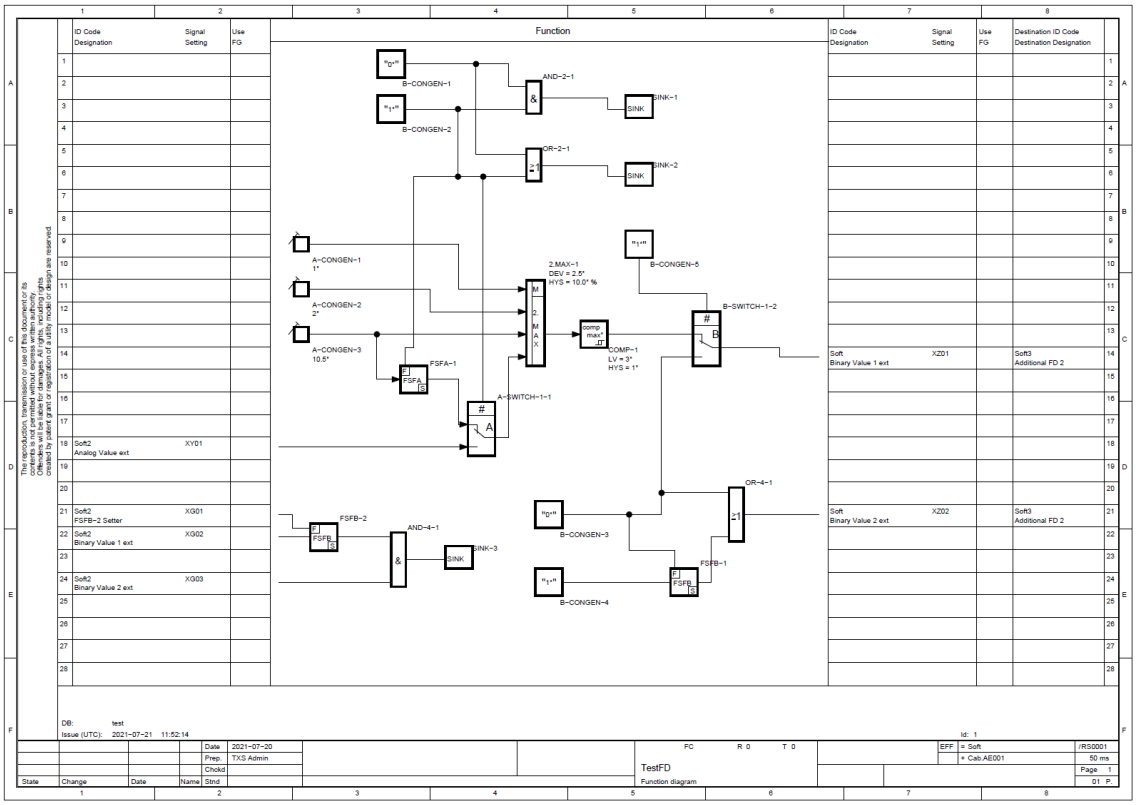

Abb. A 4 TXS-Beispielfunktionsplan

Wandelt man beispielsweise den in Abb. A 4 gezeigten TXS-Funktionsplan auf diese Weise in ein Matlab/Simulink-Modell um, so ergibt sich der in Abb. A 5 dargestellte Funktionsplan für die Simulation. Die in diesem Funktionsplan enthaltenen Funktionsblöcke (FB_1, FB_2, …) enthalten selbst weitere Logiken, die das Verhalten der entsprechenden TXS-Funktionsbausteine nachbilden (siehe hierzu auch in /MCH 21/).

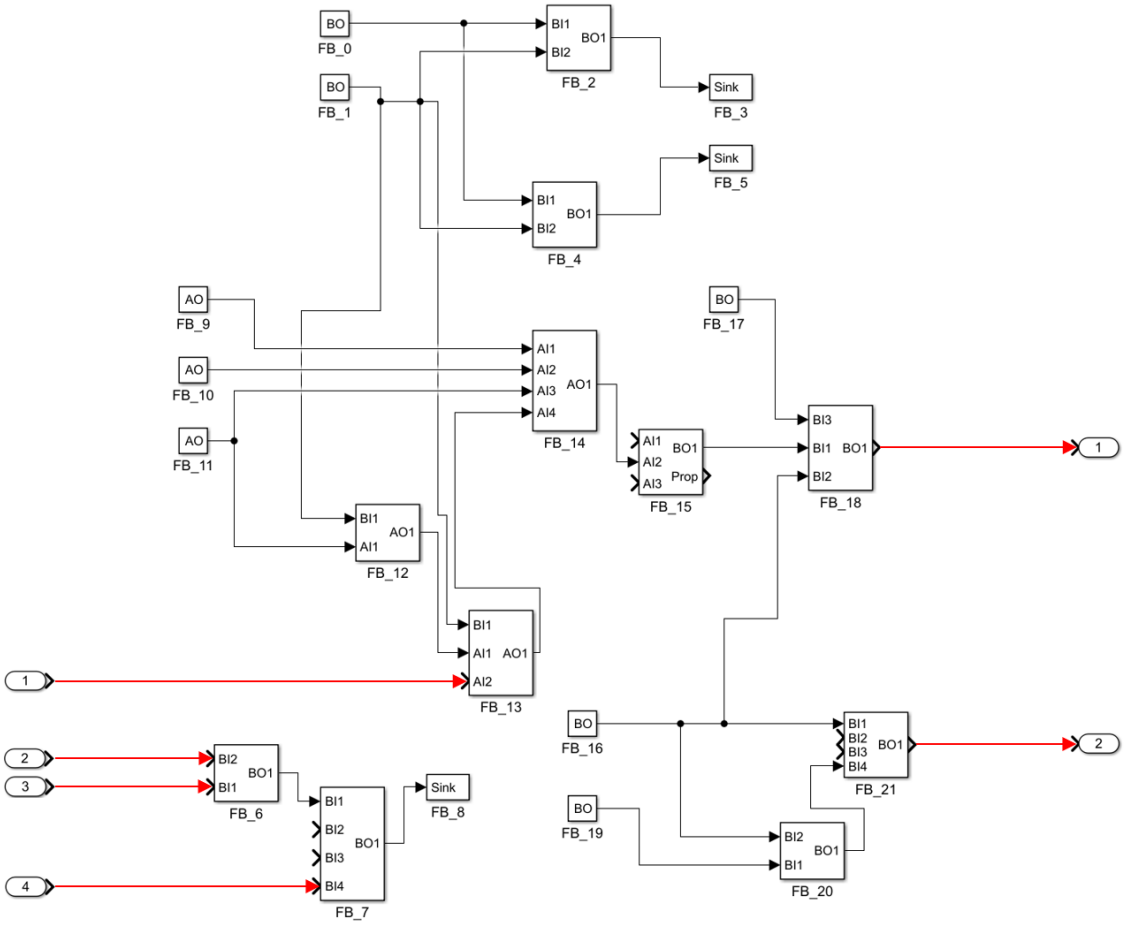

Abb. A 5 Von TXS2Simulink generiertes Simulink-Modell für den Beispielfunktionsplan in Abb. A 4, die rot markierten Pfeile sind Ein- bzw. Ausgangssignale

Einschränkend ist zu beachten, dass zwar zu jedem TXS-Funktionsplan auch ein eigenes C-File gehört, dieses stellt aber für sich allein noch keine vollwertige Software dar. Bei der Kompilierung der C-Files werden nicht nur sämtliche Funktionspläne, sondern auch viele weitere C-Files (z. B. zum Hardwareaufbau, etc.) gemeinsam zu einer ausführbaren Software kompiliert. Dieser Umstand führt u.a. dazu, dass für die Anbindung der Ein- und Ausgangssignale (rote Pfeile in Abb. A 5) an Funktionsblöcke im jeweiligen C-File allein nicht ersichtlich ist, an welchen Port („Anschluss“) des Funktionsblocks diese angebunden sind. Häufig ist die Zuordnung trotzdem eindeutig, wenn z. B. nur ein einziger Port in Frage kommt (beispielsweise steht dem Ausgangssignal „2“ in Abb. A 5 am Funktionsblock FB_21 überhaupt nur ein Ausgangsport zur Verfügung), dennoch ist auch bei der Anwendung von TXS2Simulink weiterhin eine manuelle Nachkontrolle und ggf. Nachbearbeitung notwendig.

A.2 Analysen mit den Netzwerkmanipulatoren

Auf Basis der Erkenntnisse der Versuche mit Wireshark wurde eine Software entwickelt, die das Mitlesen und, im Sinne einer Fehlerinjektion, das Verändern von Datenpaketen erlaubt (Abb. A 7).[25] Diese Software läuft auf eigens hierfür vorgesehenen Netzwerkmanipulatoren (basierend auf Mikrorechnern des Typs Raspberry Pi 4, Abb. A 6). Eine etwas ausführlichere Beschreibung der Software ist im Abschnitt A.2.2 zu finden.

Jeder Netzwerkmanipulator verfügt über drei Ethernetanschlüsse. Einer dient ausschließlich der Steuerung der Netzwerkmanipulatoren von außen, durch die anderen beiden wird der gesamte Netzwerkverkehr in beide Richtungen entweder unbeeinflusst oder manipuliert weitergeleitet.

Einen typischen Aufbau eines Systems bei Verwendung eines Netzwerkmanipulators zeigt Abb. A 8. In diesem Beispielaufbau kann der gesamte Datenverkehr von und zur Red. 1 mitgelesen und ggf. verändert werden.

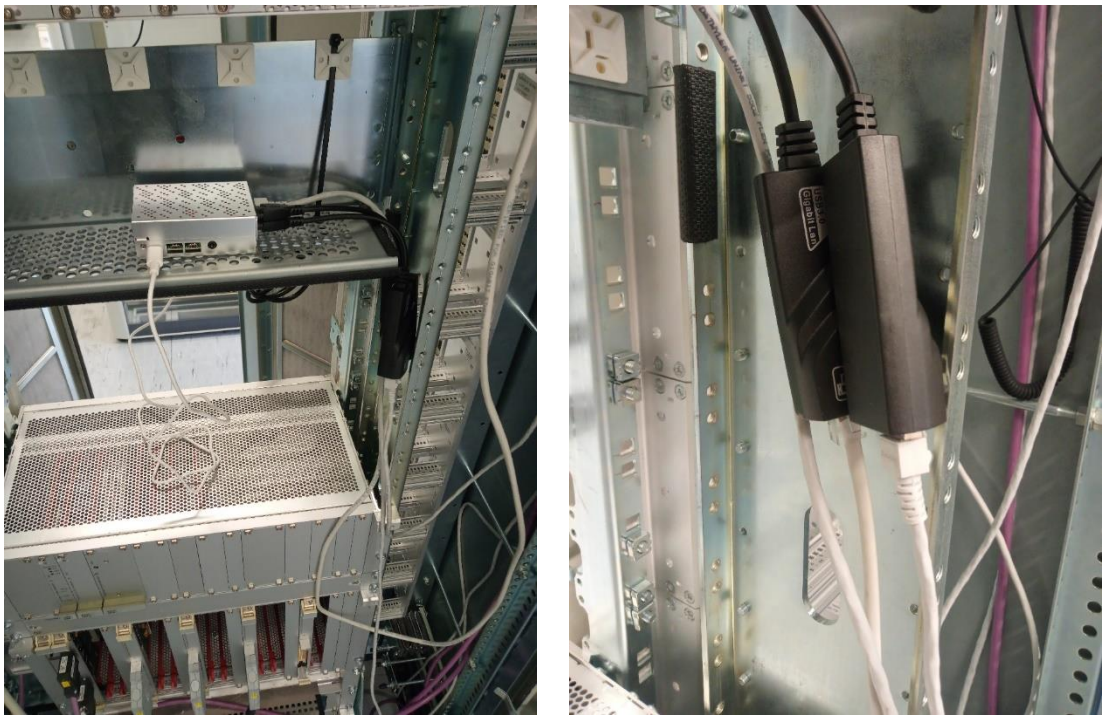

Abb. A 6 Ein Netzwerkmanipulator im Einsatz

[25] Zusätzlich kann die Software auch im Sinne von Cyberangriffen gezielte Manipulationen der Datenpakete vornehmen.

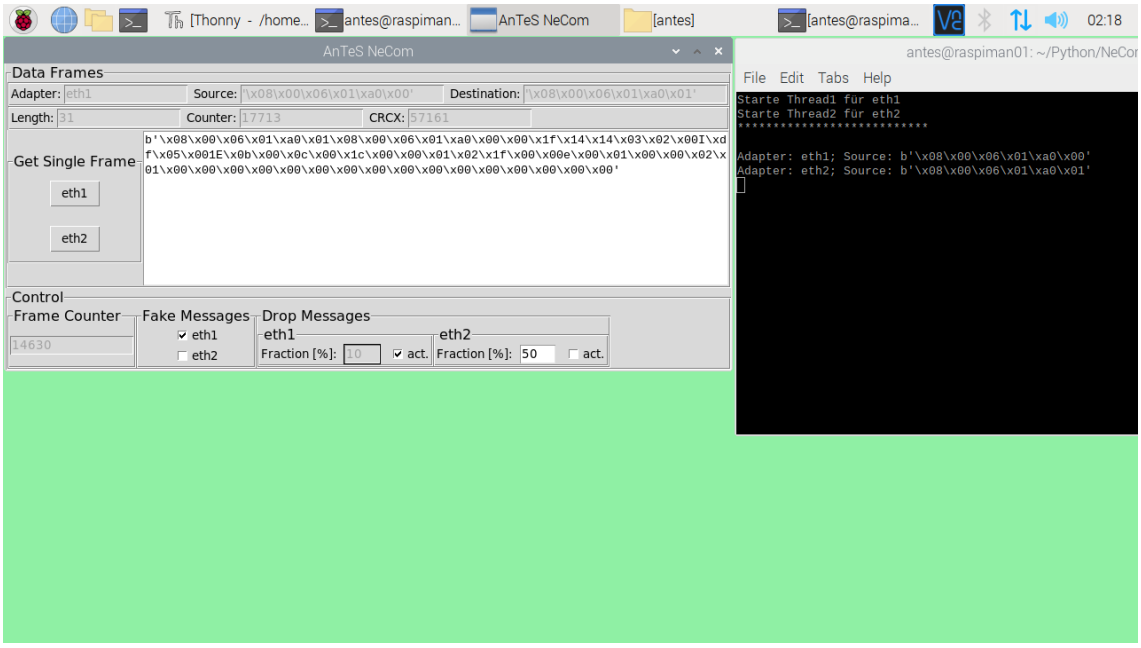

Abb. A 7 Software der Netzwerkmanipulatoren, Screenshot

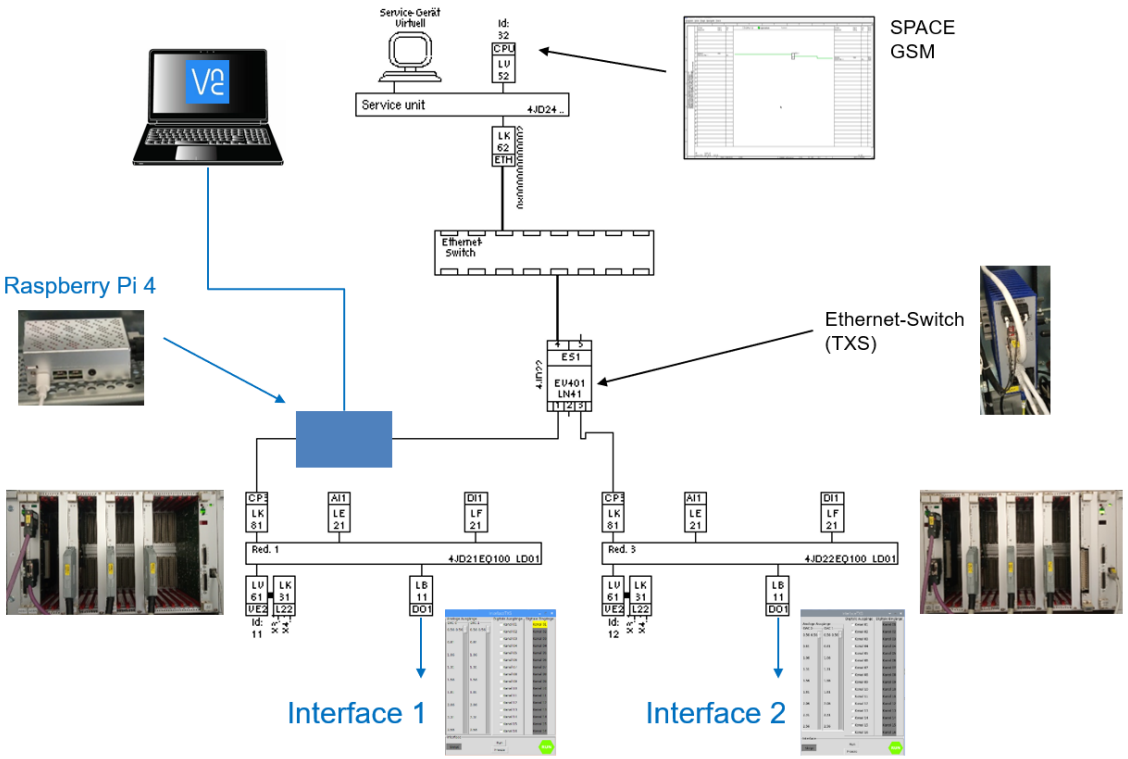

Abb. A 8 Beispiel für die Integration und Verwendung eines Netzwerkmanipulators („Raspberry Pi 4“ im Bild)

Mit den Netzwerkmanipulatoren konnte einerseits das grundsätzliche Verhalten der Netzwerkkommunikation bei Fehlern (also deren relevante Ausfallarten) bestimmt

werden. Andererseits konnte mit deren Hilfe das Verhalten bestimmter Modellsysteme (durch Fehlerinjektion) auch mit real aufgebauten Systemen (AnTeS-SILT-real) untersucht werden (siehe z. B. Abschnitt 3.2).

Komplexere Modellsysteme mit mehr als fünf Netzwerkverbindungen[26] wurden hingegen ausschließlich auf Basis der zuvor bestimmten relevanten Ausfallarten ausschließlich mit Hilfe simulierter Systeme untersucht (siehe ebenfalls z. B. Abschnitt 3.2).

Da die Analysen der (Gesamt-)Modellsysteme ausführlich in den Abschnitten 3.2 und 3.3 beschrieben werden, folgt hier nur eine detailliertere Beschreibung der Bestimmung der relevanten Ausfallarten der Netzwerkkommunikation.

A.2.1 Bestimmung der relevanten Ausfallarten

Mit Hilfe der bisher erlangten Erkenntnisse und insbesondere den darauf aufbauend entwickelten Netzwerkmanipulatoren, wurden die relevanten Ausfallarten der Netzwerkkommunikation ermittelt. Das entsprechende hierfür verwendete Testsystem entspricht dem in Abb. A 8 gezeigtem Beispiel.

Erste Versuche konzentrierten sich zunächst auf die grundlegenden Auswirkungen der Veränderung von einzelnen Bits eines Datenpakets darauf, wie und ob eine solche Nachricht vom Empfänger noch verarbeitet wird. Hierzu wurden systematisch Änderungen von Bits in jeder einzelnen Position eines Datenpakete durchgeführt.

Solche Änderungen eines einzelnen Bits entsprechen entweder sogenannten Single Event Upsets (SEU) /WIK 23a/, wie sie beispielsweise in Halbleiterbauelementen beim Durchgang hochenergetischer ionisierender Teilchen (z. B. Schwerionen, Protonen) auftreten können, oder sogenannten Single Event Latchups (SEL) /WIK 23b/, wie sie beispielsweise durch lokale Kurzschlüsse auftreten können, wodurch der Zustand einzelner Bits („bitflip“) entweder einmalig (d.h. ohne dauerhaften Schaden - SEU) oder in einem Bauteil dauerhaft (SEL) verändert wird.

---

[26] Inklusive des Prototyps (der nur äußerlich etwas anders Aussieht), verfügt die GRS insgesamt über fünf Netzwerkmanipulatoren. Systeme mit mehr als fünf Netzwerkverbindungen können daher im Sinne automatischer Ausfalleffektanalysen mit den Netzwerkmanipulatoren nicht untersucht werden. Mit Hilfe von TXS-Ersatzschaltungen (siehe Abschnitt A.2.3) kann diese Einschränkung zwar umgangen werden, aufgrund der begrenzten Geschwindigkeit der realen Systeme wächst der Zeitaufwand dennoch für komplexere Modellsysteme dann sehr stark an.

Es zeigte sich, dass in allen Versuchen Datenpakete zuverlässig auf Seite des Empfängers als fehlerhaft erkannt werden, sobald auch nur ein einziges Bit an einer beliebigen Stelle verändert wird.[27] Insbesondere wirkt sich zusätzlich die Änderung eines einzelnen Bits nur in einem einzelnen Datenpaket praktisch überhaupt nicht aus (siehe auch Ausführungen zu verschiedenen Verlustraten von Datenpaketen weiter unten).[28] Werden einzelne Bits in einer Reihe von Datenpaketen verändert, so wird die entsprechende Netzwerkkommunikation vom Empfänger als fehlerhaft erkannt (und beispielsweise im Graphischen Servicemonitor GSM mit roten Kreuzen gekennzeichnet (siehe beispielsweise Abb. A 10) – diese Ausfälle sind demnach selbstmeldend bzw. zumindest (einfach) detektierbar).

Damit also überhaupt Fehler in der Netzwerkkommunikation mit signifikanten Auswirkungen auftreten, müssen mehrere Datenpakete betroffen sein. Daher wurde in den nachfolgenden Versuchen sukzessive der Anteil fehlerhafter Datenpakete (durch Fehlerinjektion mit den Netzwerkmanipulatoren) immer weiter erhöht (0 %, 10 %, 20 %, …) und die jeweiligen Auswirkungen auf das Gesamtsystem aufgezeichnet und dokumentiert.

Hier erwies sich die Netzwerkkommunikation im Testsystem als äußerst robust. Bis zu etwa 50 % der Datenpakete können fehlerhaft sein oder komplett verlorengehen, ohne dass sich signifikante Auswirkungen auf Empfängerseite ergeben (Abb. A 9).

---

[27] Theoretisch könnten in einzelnen Fällen durch Änderung eines einzelnen Bits auch Datenpakete entstehen, die nicht als fehlerhaft erkannt werden. Deren zufälliges Auftreten ist jedoch sehr unwahrscheinlich (siehe hierzu Ausführungen im Abschnitt A.2.1 ab Seite 50).

[28] Es kann aber bei einem einzelnen „defekten“ Datenpaket unter Umständen zu einer etwa um die in der Leittechnik eingestellten Zykluszeit verzögerten Auslösung kommen (typischerweise ~ 50 ms). Dies kann und sollte bei der Auswahl der verwendeten Zykluszeit des Leittechniksystems in Betracht gezogen werden.

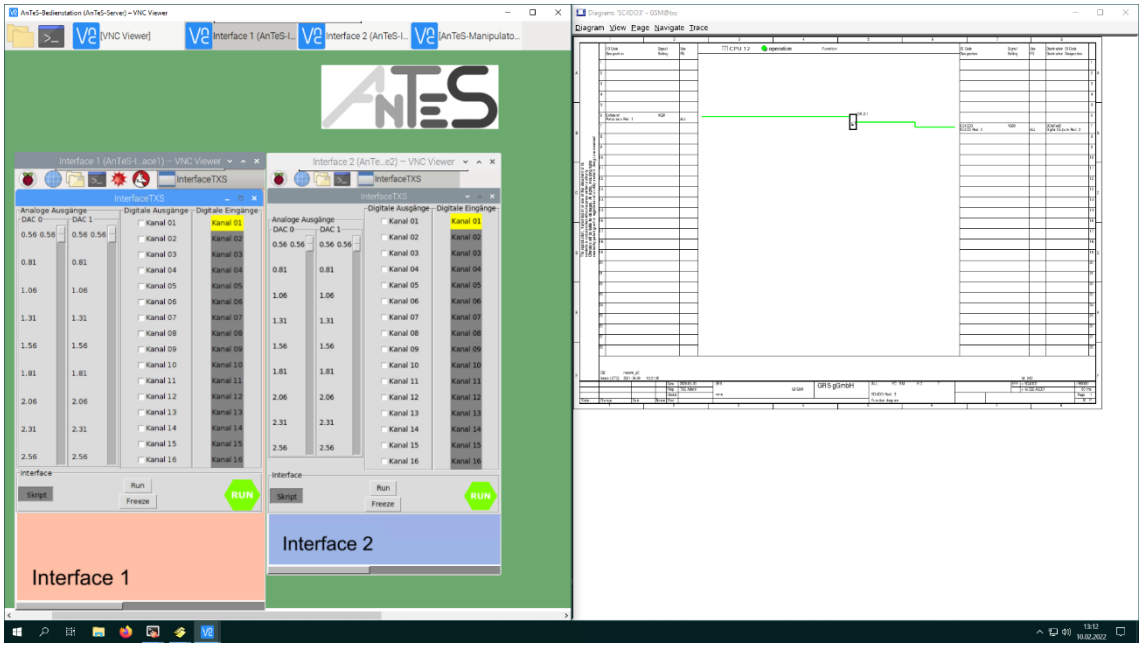

Abb. A 9 Verhalten des Systems in Abb. A 8 bei Beeinflussung von weniger als 50 % der Datenpakete (Momentaufnahme)

Auf der linken Seite sind die Ausgaben der beiden Interfaces zu sehen, auf der rechten Seite das von der Red. 3 empfangene Signal im GSM.

Dabei spielte es auch keine Rolle, ob z. B. 1/3 (~ 33 %) der Datenpakete zufällig verändert wurden oder tatsächlich genau jedes dritte Datenpaket betroffen war.

Sind mehr als 70 % der Datenpakete fehlerhaft oder gehen verloren, so wird die Netzwerkkommunikation auf Empfängerseite komplett als ausgefallen bewertet und die entsprechenden Signale im GSM mit roten Kreuzen gekennzeichnet (Abb. A 10).[29]

[29] Anmerkung: Durch Auswahl geeigneter Defaultwerte von Funktionsbausteinen oder Auswertung dieser Fehler in der Software, können solche Ausfälle fail-safe gestaltet werden.

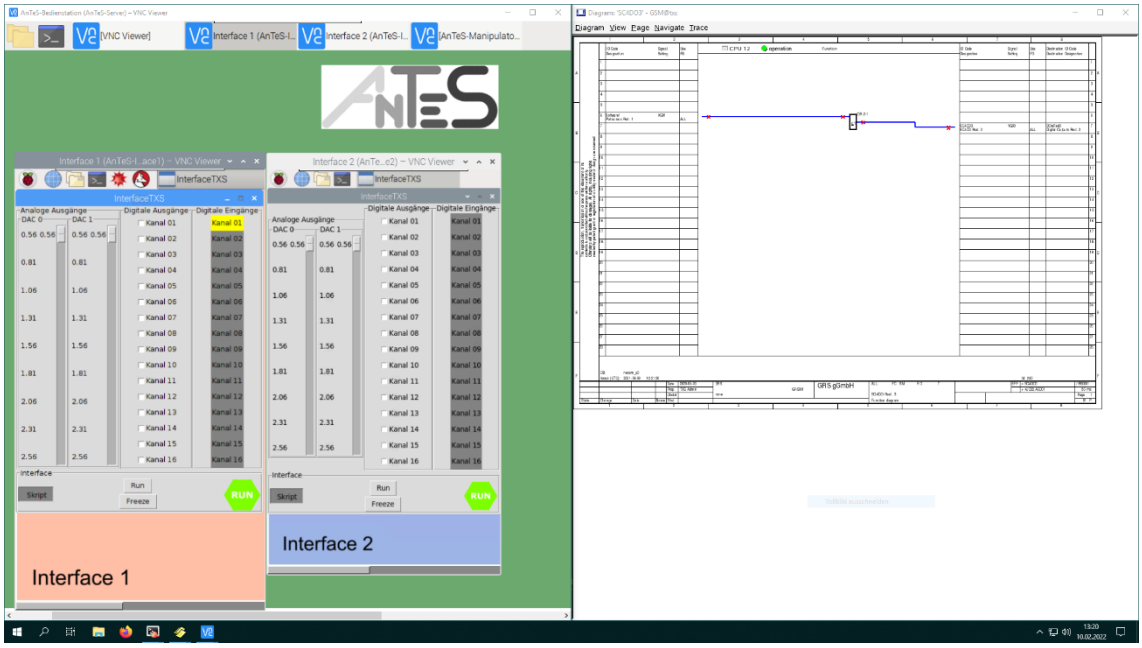

Abb. A 10 Verhalten des Systems in Abb. A 8 bei Beeinflussung von mehr als 70 % der Datenpakete (Momentaufnahme)

Im Interface 2 (des Empfängers) keine Signalausgabe mehr zu sehen. Im GSM wird die Kommunikation durch rote Kreuze als fehlerhaft/ausgefallen gekennzeichnet.

Zwischen typischerweise 50 % und 70 % fehlerhafter oder verlorener Datenpakete, kommt es zu einem „flimmern“. In diesem Bereich wechselt das beobachtete Verhalten (mehrfach pro Sekunde) zwischen den beiden dargestellten Zuständen in Abb. A 9 und Abb. A 10. In diesem Bereich würden Auslösungen (wenn auch evtl. etwas verzögert) immer noch erfolgen, aber gleichzeitig auch das fehlerhafte Verhalten detektiert.

Anmerkung: Die angegebenen Zahlenwerte sind nur als ungefähre Richtgrößen zu verstehen. Bei wiederholten Versuchen waren die konkreten Zahlenwerte (im Prozentpunktebereich) häufig etwas verschieden. Dies basiert vermutlich auf einem in jedem Versuch etwas variierendem Timing – die beiden Redundanzen des Testsystems und der verwendete Netzwerkmanipulator arbeiten alle zyklisch, aber vollkommen asynchron zueinander.

Insgesamt ergaben die Versuche das in Tab. A 1 zusammengefasste Verhalten bei unterschiedlichen Anteilen veränderter Datenpakete.

Tab. A 1 Einfluss des Anteils „gestörter“ Datenpakete auf das beobachtete Verhalten in durchgeführten Versuchen

Versuch	Beobachtetes Verhalten
Ungestörte Kommunikation	ok
10 % der Datenpakete verändert	ok
20 % der Datenpakete verändert	ok
30 % der Datenpakete verändert	ok
40 % der Datenpakete verändert	ok
50 % der Datenpakete verändert	flimmern*)
60 % der Datenpakete verändert	flimmern*)
70 % der Datenpakete verändert	SF
80 % der Datenpakete verändert	SF
90 % der Datenpakete verändert	SF
100 % der Datenpakete verändert	SF
*) das System wechselt mehrfach pro Sekunde zwischen den beiden in Abb. A 9 und Abb. A 10 gezeigten Zuständen hin und her SF – selbstmeldender Fehler	

A.2.2 Software zur Netzwerkmanipulation

In diesem Abschnitt wird die in den Netzwerkmanipulatoren laufende Software (Abb. A 11) etwas näher vorgestellt.

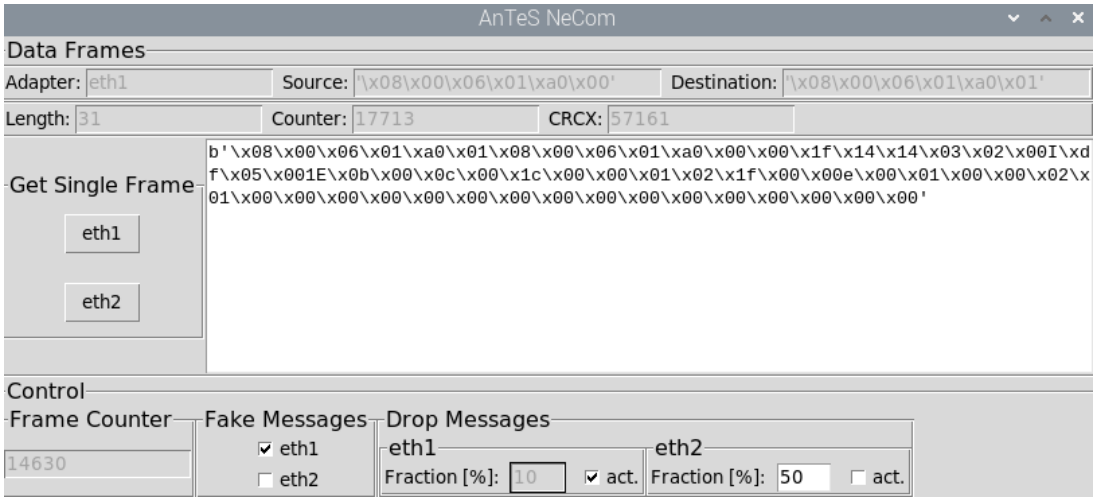

Abb. A 11 Software zur Netzwerkmanipulation, Screenshot

Nach dem Start der Software werden im Bereich „Data Frames“ (siehe Screenshot) zunächst keine Informationen angezeigt.

Im unteren Bereich („Control“) läuft lediglich ein kontinuierlicher Zähler („Frame Counter“) hoch, der die Anzahl der seit dem Start der Software (in beide Richtungen) durchgeleiteten (und ggf. veränderten) Datenpakete anzeigt.

Durch Betätigen der Knöpfe „eth1“ und „eth2“ (unter „Get Single Frame“) kann jederzeit ein einzelnes Datenpaket angezeigt werden. Durch „eth1“ wird das zum Zeitpunkt des Drückens gerade aktuelle Datenpaket angezeigt, dass vom Netzwerkmanipulator am Ethernetanschluss 1 empfangen und über den Ethernetanschluss 2 weiterverschickt wurde. Entsprechendes gilt für „eth2“.

Durch Auswahl von „eth1“ oder „eth2“ unter „Fake Messages“ kann anstatt der aktuell empfangenen Datenpakete, die aktuell aufgezeichnete und im Datenbereich angezeigte Nachricht immer wieder versendet werden (wobei aber der Counter in den TXS-Nutzdaten sowie die TXS-Prüfsumme beim Versand angepasst wird – siehe auch Abschnitt 2.3 zu potenziellen Cyberangriffen).[30]

Unter „Drop Messages“ können für beide Ethernetanschlüsse die Anteile der Datenpakete eingestellt werden, die nicht an den anderen Ethernetanschluss weitergeleitet werden (die konkrete Auswahl erfolgt statistisch durch Zufallsauswahlen).

Neben dieser Software mit (einfacher) graphischer Oberfläche wurden eine Reihe weiterer Python-Skripte erstellt, die für weitere Versuche notwendig waren (beispielsweise das Nicht-Weiterleiten jedes n-ten Datenpakets).

Da die Versuche zeigten, dass die Netzwerkkommunikation nur zwei relevante Zustände kennt (entweder funktioniert die Kommunikation noch oder sie ist selbstmeldend ausgefallen), besteht neben der Nutzung der Netzwerkmanipulatoren auch die Möglichkeit Fehler der Netzwerkkommunikation auch durch TXS-Ersatzschaltungen (im Sinne einer Fehlerinjektion) umzusetzen. Dies wird im nachfolgenden Abschnitt erläutert. Trotzdem ist die zukünftige Weiterentwicklung der Netzwerkmanipulatoren, insbesondere für die Untersuchung von Cyberangriffszenarien, sicher angezeigt.

---

[30] Diese Manipulationen sind für die Nachstellung zufälliger Fehler irrelevant und spielen nur bei (nachgestellten) Cyberangriffen eine Rolle. Da Cybersicherheit nicht Thema dieses Vorhabens war, ist die Funktionalität hier noch recht einfach gehalten. Es wäre aber z. B. problemlos möglich die Software so zu erweitern, dass eine ganze Serie von Datenpaketen aufgezeichnet wird, die dann später (z. B. in einem Loop) an die Empfänger geschickt werden. Dies würde Manipulation jenseits der Netzwerkmanipulatoren dann komplett vor den Empfängern verbergen.

A.2.3 TXS-Ersatzschaltung zur Netzwerkmanipulation

Die in diesem Abschnitt vorgestellte Vorgehensweise, anstatt der Netzwerkmanipulatoren TXS-Ersatzschaltungen zu nutzen, wurde für die in diesem Vorhaben durchgeführten Versuche noch nicht genutzt. Sie ist vielmehr ein Ergebnis ebendieser Versuche und der dabei gewonnenen Erkenntnisse. Da die Netzwerkkommunikation nur zwei Zustände[31] einnehmen kann, können mit Hilfe weniger TXS-Funktionsbausteine einfache Schaltungen erstellt werden, die exakt zu demselben Verhalten wie die Netzwerkmanipulatoren führen. Dies wird (repräsentativ) durch Abb. A 12 verdeutlicht.

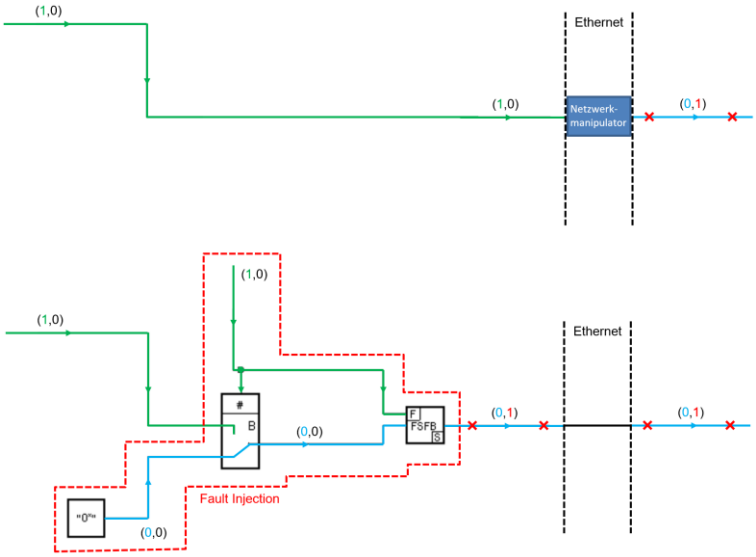

Abb. A 12 TXS-Ersatzschaltung für einen Netzwerkmanipulator („Fault Injection"), repräsentatives Beispiel

Im oberen Teil wird eine fehlerfreie binäre 1 über Ethernet übertragen. D. h. das zu übertragende Signal hat den Wert 1, das Fehlerattribut (s. /MCH 21/) des Signals ist 0. Wird diese vom Netzwerkmanipulator verändert oder nicht weitergeleitet, so wird dies beim Empfänger auf der rechten Seite als (Defaultwert) 0 mit dem Fehlerattribut 1 interpretiert. Dasselbe lässt sich durch die zusätzlichen TXS-Funktionsbausteine innerhalb von „Fault Injection" im unteren Bild erreichen. Dort kann durch ein zusätzliches Signal (das z. B. über das entsprechende Interface vorgegeben wird), mittels eines Umschalters und eines sogenannten FSFB-Bausteins[32], derselbe Effekt erreicht werden, ohne dass noch ein Netzwerkmanipulator benötigt wird.

[31] Anmerkung: Der dritte Zustand („flimmern"), z. B. in Tab. A 1, kann problemlos als abwechselnd „ok" und „SF" (selbstmeldender Fehler) interpretiert werden.

[32] Der Name des Funktionsbausteins wird in der verfügbaren Dokumentation (u.a. /TXS 12/) nicht erläutert. Dieser steht vermutlich für FehlerSetzer-FunktionsBaustein (o.ä.). Bei einer logischen 1 am Eingang „F" wird jedenfalls das Fehlerattribut des eingehenden Signals am Ausgang gesetzt.

A.3 Verwendete Modellsysteme

Ausführliche Beschreibungen der meisten Modellsysteme (und der entsprechenden Nomenklatur) können in /MCH 18/ und /MCH 21/ nachgelesen werden. An dieser Stelle werden die in diesem Vorhaben verwendeten Modellsysteme nur kompakt anhand Ihrer Darstellung in Matlab/Simulink vorgestellt. Vergleichsweise einfache Modellsysteme wurden aber auch mit dem realen Sicherheitsleittechniksystem von AnTeS (AnTeS-SILT-real) für Analysen umgesetzt (siehe z. B. Abb. 3.2 auf Seite 25).

Die ersten beiden Modellsysteme (A122 und A222) als Matlab/Simulink-Modelle (AnTeS-SILT-sim) sind gemeinsam in Abb. A 13 dargestellt. Das hier tatsächlich zwei unterschiedliche Modellsysteme gemeinsam simuliert werden können, soll durch die beiden nachfolgenden beiden Abbildungen (Abb. A 14 und Abb. A 15) verdeutlicht werden.

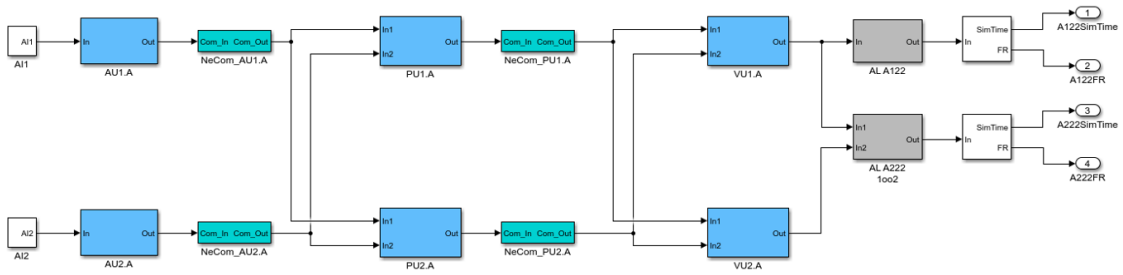

Abb. A 13 Modellsysteme A122 und A222 in Matlab/Simulink

Abb. A 14 ist mit Abb. A 13 identisch, es wurde lediglich der das Modellsystem A122 repräsentierende Teil des Gesamtmodells rot-umrandet. Bei der Auswertung (Bestimmung der in der bisher erreichten Ausfallrate) im grün-umrandete Teils des Bildes wird ausschließlich das Ausgangssignal der VU1.A (über eine aus Vergleichbarkeitsgründen immer vorhandenen zusätzlichem analogen Logik (AL)) verwendet. Der entsprechende berechnete Wert gehört also zu einem System mit 2 AUs, 2 PUs und 1 VU (des „Teilsystems“ A) – A122.

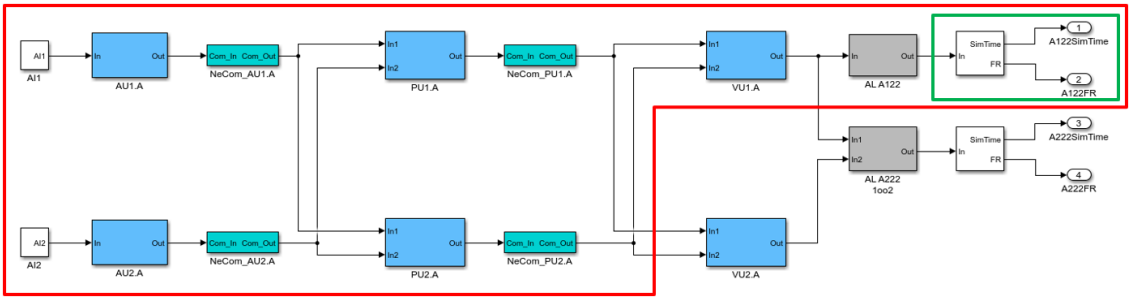

Abb. A 14 Modellsystem A122 (rot umrandet)

Entsprechend werden im rot-umrandeten Teilsystem in Abb. A 15 die Ausgänge beider VUs berücksichtigt – A222.

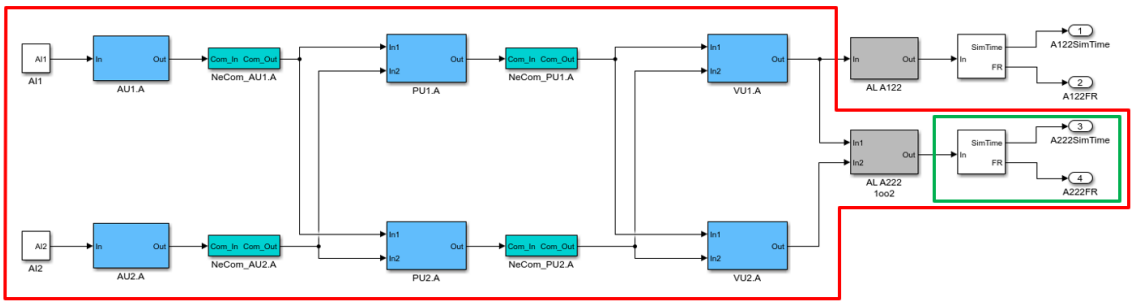

Abb. A 15 Modellsystem A222 (rot umrandet)

Weitere verwendeten Modellsysteme sind in den nachfolgenden Abbildungen dargestellt.

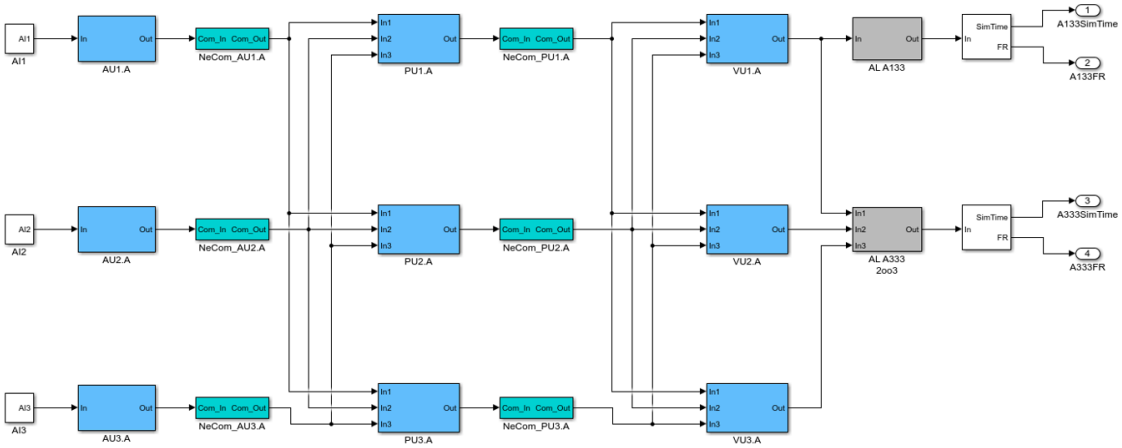

Abb. A 16 Modellsysteme A133 und A333 in Matlab/Simulink

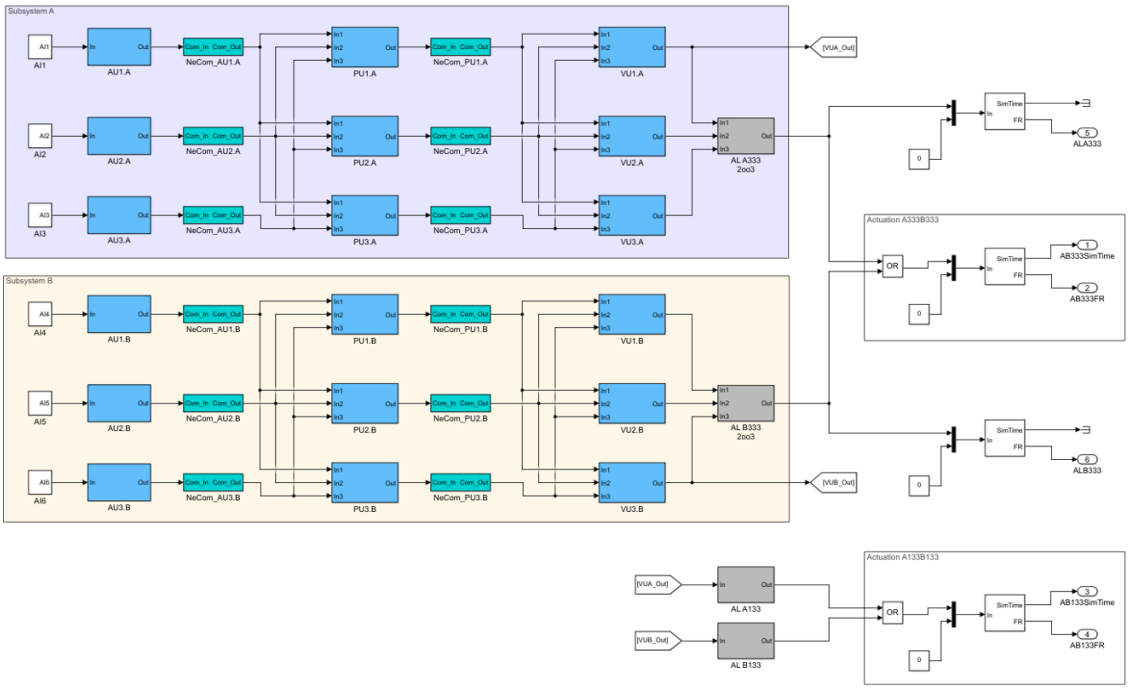

Abb. A 17 Modellsysteme A133B133, A333B333 sowie A333 und B333 in Matlab/Simulink

Die Ergebnisse in Abb. 3.9 auf Seite 34 weisen zusätzlich drei weitere Modellsysteme aus (A222mod, A122mod und A333ext):

- A122mod:
  - Ist mit A122 weitgehend identisch, es wurde lediglich ein Funktionsbaustein (konkret eine 2.Max-Auswahl) in beiden PUs ausgetauscht (durch eine Max-Auswahl).
  - Auf der in Abb. A 13 Darstellungsebene[33] sieht A122mod aus wie A122.
  - Gründe und weitere Erläuterungen hierzu können /MCH 21/ entnommen werden.

[33] Die in den Abbildungen dieses Abschnitts gezeigten Modellsysteme zeigen diese auf der obersten Darstellungsebene. In jedem „Kasten“ verbergen sich sogenannten Simulink-Subsysteme, so u.a. z. B. die gesamten Funktionspläne der verwendeten leittechnischen Funktionen.

- A222mod:
  - Ist mit A222 weitgehend identisch, es wurde lediglich ein Funktionsbaustein (konkret eine 2.Max-Auswahl) in beiden PUs ausgetauscht (durch eine Max-Auswahl).
  - Auf der in Abb. A 13 Darstellungsebene[34] sieht A222mod aus wie A222.
  - Gründe und weitere Erläuterungen hierzu können /MCH 21/ entnommen werden.
- A333ext:
  - Ist grundsätzlich mit A333 identisch.
  - Anstatt interner Zufallszahlen, wurden testweise extern bestimmte Zufallszahlen in jedem Berechnungsschritt in das Modell eingespeist.
  - Dies war ein Versuch die Berechnungen insgesamt zu beschleunigen. Da die Simulink-Modelle jedoch in der Sprache C vorliegen (kompiliert als recht „schnelle“ DLL-Dateien), konnte hier kein Geschwindigkeitszuwachs gewonnen werden.
  - Bis auf leicht unterschiedliche Genauigkeiten der verwendeten Zufallszahlen sind daher A333 und A333ext als identisch zu betrachten.

[34] Die in den Abbildungen dieses Abschnitts gezeigten Modellsysteme zeigen diese auf der obersten Darstellungsebene. In jedem „Kasten“ verbergen sich sogenannten Simulink-Subsysteme, so u. a. z. B. die gesamten Funktionspläne der verwendeten leittechnischen Funktionen.

B Begriffserläuterungen

B.1 Broadcast in Netzwerken

Broadcast in Netzwerken ist ein Mechanismus, der es ermöglicht, Datenpakete an alle Geräte in einem Netzwerk zu senden. Der Sinn und Zweck von Broadcast ist es, die Kommunikation zwischen Geräten zu erleichtern, ohne dass der Absender die genaue Adresse jedes Empfängers kennen muss. Durch Broadcast können beispielsweise ARP-Anfragen gesendet werden, um die MAC-Adresse eines bestimmten Geräts zu ermitteln. ARP-Anfragen (Address Resolution Protocol) sind Netzwerkpakete, die verwendet werden, um die physikalische MAC-Adresse eines Geräts in einem Netzwerk zu ermitteln.

B.2 Bytes und ihre Darstellung

Ein Byte ist eine grundlegende Einheit der digitalen Information und besteht aus 8 Bits. Jedes Bit kann den Wert 0 oder 1 annehmen und repräsentiert somit eine Binärzahl (0 oder 1). Die Darstellung eines Bytes als Binärzahl erfolgt durch die Aneinanderreihung von 8 Bits (z. B.: 00101101).

Die Darstellung eines Bytes als Dezimalzahl erfolgt durch die Umrechnung der Binärzahl in das Dezimalsystem. Jedes Bit hat dabei eine Wertigkeit von 2 hoch n, wobei n von 0 bis 7 läuft. Die Dezimalzahl wird berechnet, indem die Produkte der Bits mit ihren Wertigkeiten addiert werden. Zum Beispiel:

00101101 (Binär) = $2^5 + 2^3 + 2^2 + 2^0 = 32 + 8 + 4 + 1 = 45$ (Dezimal).

Die Darstellung eines Bytes als Hexadezimalzahl erfolgt durch die Gruppierung von jeweils 4 Bits und ihre Umrechnung in eine entsprechende Hexadezimalziffer. Jedes Nibble (4 Bits) hat eine Wertigkeit von 2 hoch n, wobei n von 0 bis 3 läuft. Die Hexadezimalzahlen werden durch die Symbole 0-9 und A-F dargestellt. Zum Beispiel:

00101101 (Binär) = 0010 (Binär) 1101 (Binär) = 2 (Hexadezimal) D (Hexadezimal) = 2D (Hexadezimal).

Insgesamt sind diese Darstellungen (binär, dezimal und hexadezimal) wichtige Werkzeuge zur Analyse und Verarbeitung von digitalen Daten und bieten verschiedene Perspektiven auf die Informationen, die in einem Byte enthalten sind.

Häufig (wie in diesem Bericht) werden Bytes durch Hexadezimalzahlen mit vorangestellten „x“ dargestellt, beispielsweise x2D.

B.3 Ethernet

Ethernet ist ein Standard für die kabelgebundene Datenübertragung in lokalen Netzwerken (LANs). Es ist ein gemeinsamer Standard für kabelgebundene Netzwerktechnologie und wird am häufigsten in Computer-Netzwerken verwendet. IEEE 802.3 beschreibt das physikalische und das MAC-Protokoll für die Übertragung von Datenpaketen zwischen Netzwerkgeräten. Es definiert die physische Verkabelung, die Übertragungsrate, die Signalisierung, die Fehlererkennung und -korrektur sowie die Kollisionsvermeidung im Netzwerk.

Der entsprechende Standard unterstützt verschiedene Übertragungsmedien wie Koaxialkabel, Twisted Pair-Kabel und Glasfaser. Es definiert auch verschiedene Geschwindigkeiten, von 10 Mbit/s bis (derzeit) 400 Gbit/s.

IEEE 802.3 Ethernet ist ein weit verbreiteter und etablierter Standard in der Netzwerktechnologie und wird in einer Vielzahl von Anwendungen wie Büros, Rechenzentren und industriellen Netzwerken eingesetzt.

B.3.1 Ethernet-Hubs und -Switches

Ethernet-Hubs senden eingehende Datenpakete an alle angeschlossenen Geräte weiter, unabhängig davon, ob sie das Paket empfangen müssen oder nicht. Ethernet-Switchs hingegen untersuchen eingehende Datenpakete und senden sie nur an das Zielgerät, für das sie bestimmt sind. Dadurch reduziert ein Switch die Netzwerkbelastung und erhöht die Sicherheit, da keine unnötigen Datenpakete an unautorisierte Geräte gesendet werden. Insgesamt bieten Ethernet-Switches eine höhere Leistung und Funktionalität als Ethernet-Hubs und werden daher in den meisten Netzwerken bevorzugt.

In den durchgeführten Versuchen innerhalb dieses Vorhabens war das ansonsten ungünstige Verhalten von Ethernet-Hubs, dass alle Datenpakete an alle angeschlossenen Geräte weitergeleitet werden, für das „Sniffen“ („Mitlesen“) jedoch erwünscht, da dies (z. B. mittels der Software Wireshark) eine einfache Möglichkeit zum Mitlesen des Datenverkehrs erlaubt.

B.4 MAC-Adresse

Eine MAC-Adresse (Media Access Control Address) ist eine eindeutige Kennung, die Netzwerkgeräten auf der Ebene der Datenverbindungsschicht (Layer 2 im OSI-Modell) zugewiesen ist. Jede Netzwerkschnittstelle, z. B. Ethernet- oder WLAN-Karte, hat eine eindeutige MAC-Adresse, die aus 48 Bits (6 Bytes) besteht und normalerweise in Hexadezimalzahlen dargestellt wird. Die ersten 24 Bits (3 Bytes) der MAC-Adresse identifizieren den Hersteller des Netzwerkadapters, während die letzten 24 Bits (3 Bytes) eine eindeutige Kennung des Adapters darstellen. Die MAC-Adresse wird verwendet, um Datenpakete innerhalb eines lokalen Netzwerks (LAN) direkt zwischen zwei Netzwerkgeräten zu übertragen, indem sie als Zieladresse in den Paketen eingebettet wird. Die MAC-Adresse ist daher ein wichtiges Element bei der Kommunikation in lokalen Netzwerken. Im Zusammenhang mit TXS sind MAC-Adressen der beteiligten Komponenten konfigurierbar. Hierbei ist darauf zu achten, dass die Eindeutigkeit (zumindest innerhalb eines Systems) nicht verletzt wird.

B.5 OSI-Modell/Layer

Das OSI (OSI – Open Systems Interconnection) Layer-Modell ist ein Referenzmodell für Netzwerkprotokolle als Schichtenarchitektur /WIK 23/ /GEE 22/. Es wird seit 1983 von der International Telecommunication Union (ITU) und seit 1984 auch von der International Organization for Standardization (ISO) als Standard veröffentlicht /ISO 94/. Jede Schicht bietet eine spezifische Funktion und arbeitet zusammen mit den anderen Schichten, um die Kommunikation zwischen Geräten in einem Netzwerk zu ermöglichen.

Die sieben Schichten/Layer des OSI-Modells sind:

1. Physikalische Schicht: Diese Schicht beschreibt die physikalische Übertragung von Daten über das Netzwerk, einschließlich der Art der Verkabelung, der Signalstärke und der Übertragungsrate.
2. Sicherungsschicht: Diese Schicht stellt sicher, dass die Daten korrekt über das Netzwerk übertragen werden, indem sie Fehlererkennung und -korrektur sowie die Adressierung von Netzwerkgeräten bereitstellt.
3. Netzwerkschicht: Diese Schicht ist für die Adressierung und das Routing von Datenpaketen im Netzwerk verantwortlich. Sie verwendet IP-Adressen, um Datenpakete an das richtige Zielgerät weiterzuleiten.
4. Transportschicht: Diese Schicht ist für die Übertragung von Daten zwischen Anwendungen auf verschiedenen Geräten verantwortlich und stellt sicher, dass die Daten in der richtigen Reihenfolge und ohne Fehler übertragen werden.
5. Sitzungsschicht: Diese Schicht ist für die Verwaltung und Synchronisierung von Sitzungen zwischen Anwendungen auf verschiedenen Geräten verantwortlich.
6. Darstellungsschicht: Diese Schicht ist für die Darstellung von Daten verantwortlich, indem sie Datenformate und Kodierung von Datenpaketen übersetzt, damit sie von verschiedenen Geräten gelesen und verstanden werden können.
7. Anwendungsschicht: Diese Schicht stellt Anwendungen zur Verfügung, die Netzwerkressourcen verwenden, wie z. B. E-Mail, Web-Browser oder Filesharing-Anwendungen.

Das OSI-Modell ist ein wichtiges Konzept für die Netzwerkkommunikation, da es ein gemeinsames Verständnis für die Funktionsweise von Netzwerken bietet und es ermöglicht, Netzwerkprobleme auf verschiedenen Ebenen zu identifizieren und zu lösen.

B.6 „Sniffen“

"Sniffen" ist ein Begriff aus der Informatik und bezieht sich auf das Abfangen und Überwachen von Datenpaketen, die über ein Netzwerk übertragen werden. Dabei werden die Datenpakete von einem speziellen Programm oder einem sogenannten "Sniffer" erfasst und analysiert, ohne dass die Übertragung beeinträchtigt wird. Sniffen wird oft zu Sicherheits- und Diagnosezwecken eingesetzt, kann aber auch zu unerlaubtem Abhören und Datenmissbrauch verwendet werden. In englischer Sprache wird "sniffing" als entsprechender Begriff verwendet.

B.7 Wireshark

Wireshark /WIS 23/ ist eine kostenlose, quelloffene Software zur Netzwerkanalyse. Sie wird von Netzwerkadministratoren, Sicherheitsanalysten und Entwicklern verwendet, um Netzwerkverkehr zu überwachen, zu analysieren und zu diagnostizieren. Wireshark ermöglicht es, den Datenverkehr von verschiedenen Netzwerkprotokollen aufzuzeichnen und anzuzeigen, wie z. B. TCP, UDP, HTTP, DNS und viele andere. Es bietet eine grafische Benutzeroberfläche, um die erfassten Daten zu visualisieren und zu filtern, um die Analyse von bestimmten Netzwerkereignissen zu erleichtern. Wireshark ist eine leistungsstarke Software, die beispielsweise dazu beitragen kann, Netzwerkprobleme zu erkennen und zu beheben und die Netzwerksicherheit zu erhöhen.

In diesem Vorhaben wurde Wireshark für erste Voruntersuchungen zur Analyse des Datenverkehrs in den Netzwerken von TXS verwendet.

Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) gGmbH

Schwertnergasse 1
50667 Köln
Telefon +49 221 2068-0
Telefax +49 221 2068-888

Boltzmannstraße 14
85748 Garching b. München
Telefon +49 89 32004-0
Telefax +49 89 32004-300

Kurfürstendamm 200
10719 Berlin
Telefon +49 30 88589-0
Telefax +49 30 88589-111

Theodor-Heuss-Straße 4
38122 Braunschweig
Telefon +49 531 8012-0
Telefax +49 531 8012-200

www.grs.de

ISBN 978-3-949088-80-3