



Ein Beitrag zur Echtzeit-Co-Simulation der Stationsautomatisierung

Dennis Rösch

Dennis Rösch

Ein Beitrag zur Echtzeit-Co-Simulation der Stationsautomatisierung

**Ilmenauer Beiträge zur elektrischen Energiesystem-, Geräte- und
Anlagentechnik (IBEGA)**

Herausgegeben von
Univ.-Prof. Dr.-Ing. Dirk Westermann
(Fachgebiet Elektrische Energieversorgung) und
Univ.-Prof. Dr.-Ing. Frank Berger
(Fachgebiet Elektrische Geräte und Anlagen)
an der Technischen Universität Ilmenau.

Band 41

Dennis Rösch

**Ein Beitrag zur Echtzeit-Co-Simulation der
Stationsautomatisierung**



Universitätsverlag Ilmenau

2026

Impressum

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Angaben sind im Internet über <http://dnb.d-nb.de> abrufbar.

Diese Arbeit hat der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität Ilmenau als Dissertation vorgelegen.

Tag der Einreichung: 02. Juni 2025
1. Gutachter: Univ.-Prof. Dr.-Ing. Peter Bretschneider
(Technische Universität Ilmenau)
2. Gutachter: Prof. Dr. rer. Nat. Jochen Seitz
(Technische Universität Ilmenau)
3. Gutachter: Prof. Dr.-Ing. Christian Rehtanz
(Technische Universität Dortmund)
Tag der Verteidigung: 07. Januar 2026

Technische Universität Ilmenau/Universitätsbibliothek

Universitätsverlag Ilmenau

Postfach 10 05 65

98684 Ilmenau

<https://www.tu-ilmenau.de/universitaetsverlag>

ISSN 2194-2838

DOI 10.22032/dbt.69031

URN urn:nbn:de:gbv:ilm1-202600002



Dieses Werk - mit Ausnahme der anders gekennzeichneten Teile und des Umschlags - ist lizenziert unter einer [Creative Commons Namensnennung - Weitergabe unter gleichen Bedingungen 4.0 International Lizenz](https://creativecommons.org/licenses/by-sa/4.0/) (CC BY-SA 4.0): <https://creativecommons.org/licenses/by-sa/4.0/>

Titelfotos:

© iStockphoto.com : JLGutierre ; timmy ; 3alexnd ; Elxeneize ; tap10

yuyang/Bigstock.com

M. Streck, FG EGA | F. Nothnagel, FG EGA | D. Westermann, FG EEV

Kurzfassung

Der Wandel der elektrischen Energieversorgung hin zu erneuerbaren Energien und zu einem signifikant erhöhten Grad an Digitalisierung erfordert eine Analyse der Abhängigkeiten und Wechselwirkungen zwischen dem Stromnetz und der Kommunikationsinfrastruktur. Dies zeigt sich insbesondere in den Automatisierungsstrukturen von Stromnetzen, wie sie in Umspannwerken vorhanden sind.

Im Zuge der Digitalisierung werden essenzielle Systeme in Umspannwerken durch Netzwerktechnologie miteinander vernetzt und sind auf die Kommunikationsinfrastruktur angewiesen. Für verschiedene Untersuchungen ist es sinnvoll, Simulationen zu nutzen, die Erkenntnisse generieren, ohne Zugriff auf die realen Infrastrukturen zu benötigen. Aufgrund der wechselseitigen Abhängigkeit der beiden Betrachtungsebenen in digitalen Umspannwerken ergeben sich spezifische Anforderungen an eine gemeinschaftliche Simulation.

Ein bewährter Ansatz ist die Co-Simulation, die eine Analyse durch die Koordination mehrerer Simulatoren ermöglicht. Diese können spezifisch für die jeweilige Betrachtungsebene sein und erlauben so die Analyse komplexer Systeme.

Diese Arbeit stellt einen speziellen Ansatz zur Co-Simulation für die Stationsautomatisierung digitaler Umspannwerke vor. Der Ansatz ermöglicht die Untersuchung der beiden Ebenen für Echtzeit-Anwendungen unter Berücksichtigung relevanter, anwendungsspezifischer Randbedingungen. Die Co-Simulation erlaubt unter Nutzung des Standards IEC 61850 eine Kopplung zwischen Stromnetzsimulation und Virtualisierung der Kommunikationsinfrastruktur.

Das im Rahmen der Arbeit entwickelte Co-Simulationssystem zeichnet sich dadurch aus, dass die einzelnen Simulatoren unabhängig voneinander betrieben werden können und spezifische Übertragungsprotokolle für die Kopplung genutzt werden. Der erforderliche Koordinator übernimmt lediglich die Aufgabe der Synchronisierung, wodurch relevante Echtzeit-Untersuchungen für digitale Umspannwerke ermöglicht werden. Diese anwendungsbezogene Erfüllung der Echtzeit-Anforderungen und die effiziente Architektur mit einem generischen Koordinator heben sich als Alleinstellungsmerkmale von anderen Co-Simulationen ab und stellen die wesentlichen Beiträge dieser Arbeit dar. Die Konzeption des Ansatzes sowie die Validierung der Anwendbarkeit nach definierten Anforderungen bilden die Schwerpunkte der Arbeit, wobei das zeitliche Verhalten im Fokus steht.

Abstract

The transformation of power supply systems with renewable energies and a significantly higher degree of digitalization necessitates a thorough analysis of the interdependencies and interactions between the power grid and the communication infrastructure. This is especially valid for the automation systems of power grids, such as those implemented in substations.

As digitalization progresses, the critical systems within substations become interconnected through network technology, relying heavily on the communication infrastructure. For various studies, employing simulations that yield insights without direct access to real-world infrastructure proves beneficial. Given the interrelated nature of the two observation levels in digital substations, specific requirements arise for a joint simulation.

A well-established method is co-simulation, which facilitates a comprehensive analysis by coordinating multiple simulators that can be tailored to their respective levels, thus enabling the examination of complex systems.

This work introduces a distinctive approach to co-simulation tailored for the automation of digital substations. This approach allows for the investigation of both levels of consideration in real-time applications while ensuring compliance with relevant boundary conditions. By leveraging the IEC 61850 standard, this co-simulation effectively integrates power grid simulations with the virtualization of the communication infrastructure.

The co-simulation system developed in this work is notably characterized by the independent execution of individual simulators, allowing for the use of application-specific transmission protocols for coupling. The essential coordinator's role is primarily focused on synchronization, facilitating real-time investigations for digital substations. This application-specific fulfillment of real-time requirements, coupled with a streamlined architecture featuring a generic coordinator for synchronization, distinguishes this work from other co-simulations in the literature and highlights its key contributions.

The design of the approach and the validation of its applicability against defined criteria are central to this work, with a particular emphasis on timing performance.

Inhaltsverzeichnis

Abbildungsverzeichnis	xiii
Tabellenverzeichnis	xv
Abkürzungen und Formelzeichen	xvii
Definitionen	xxv
1 Einleitung	1
1.1 Motivation	1
1.2 Problemstellung	2
1.3 Forschungsfragen	3
1.4 Vorgehensweise und Aufbau der Arbeit	3
2 Theoretische Grundlagen	7
2.1 Allgemeine Grundlagen	7
2.1.1 Grundelemente von Co-Simulationen	9
2.1.2 Modellierungsansätze	13
2.2 Unterscheidung Simulation, Emulation und Virtualisierung	18
2.2.1 Simulation	19
2.2.2 Emulation	19
2.2.3 Virtualisierung	20
2.3 Generelle Anforderungen an Co-Simulationen	20
2.3.1 Synchronität	22
2.3.2 Stabilität	23
2.3.3 Echtzeitfähigkeit	24
2.4 Standards und Frameworks für Co-Simulationen	26
2.4.1 Anwendungsgebiete von Co-Simulationen	26
2.4.2 Co-Simulationsstandards	27
2.4.3 Frameworks und Anwendungen für Co-Simulationen	30
2.4.4 Übersicht der relevanten Standards und Frameworks	33
2.5 Schlussfolgerung	33

3	Anwendungsfall Digitale Station	37
3.1	Grundaufbau eines Umspannwerkes	37
3.2	Beschreibung IEC 61850 basierte Station	38
3.2.1	Grundlagen IEC 61850	40
3.2.2	Aufbau und Teile des Standards	40
3.2.3	Definition IED	40
3.2.4	Datenmodell	41
3.2.5	Kommunikationsarten	43
3.2.6	Kommunikationsstruktur	45
3.3	Kommunikationsanforderungen nach IEC 61850-5	46
3.3.1	Synchronisierung	46
3.3.2	Übertragungszeit	47
3.4	Zusammenfassung der Modellierungsanforderungen Digitaler Stationen	49
3.4.1	Synchronisierungsanforderungen	50
3.4.2	Übertragungszeitanforderungen	50
3.4.3	Echtzeitanforderungen	51
4	Entwurf und Konzeption der Co-Simulation	53
4.1	Modellierung des Stromnetzes	53
4.1.1	Ansatz der Stromnetzsimulation für die vorgestellte Echtzeit-Co-Simulation	54
4.1.2	Anforderungen an die Stromnetzsimulation	55
4.2	Modellierung des Kommunikationsnetzwerkes	56
4.2.1	Ansatz der Modellierung des Kommunikationsnetzwerkes für die vorgestellte Echtzeit-Co-Simulation	57
4.2.2	Zusammenführung der Ansätze zum Kommunikationsnetzwerk-simulator	62
4.2.3	Anforderungen an die Modellierung des Kommunikationsnetzwerkes	64
4.3	Ansatz der Kopplung	65
4.3.1	Konzeption des Datenaustauschs	68
4.3.2	Synchronität	69
4.4	Zusammenfassung des Ansatzes	70

5	Anwendung der Co-Simulation für die Stationsautomatisierung	73
5.1	Grundaufbau der Stromnetzsimulation eines Umspannwerkes	73
5.1.1	Modellierung der Primärtechnik in der Stromnetzsimulation	73
5.1.2	Modellierung der Sekundärtechnik in der Stromnetzsimulation	74
5.2	Modellierung der Kommunikationsinfrastruktur	75
5.2.1	Implementierungen der Komponenten innerhalb der Netzwerkvirtualisierung	75
5.2.2	Automatisierte Modellierung von Umspannwerken	82
5.3	Kopplung der Simulatoren auf dem Prozessbus	89
5.4	Modellierung der Verzögerungszeit des Gesamtansatzes	91
5.4.1	Gesamtverzögerungsmodell	98
5.4.2	Integration des Verzögerungsmodells in das Framework VirtualSubstation	100
6	Validierung der Co-Simulation	101
6.1	Validierungsvorgehen	101
6.1.1	Adäquate Abbildung durch die Nutzung einer Virtualisierung	102
6.1.2	Synchronität der Simulatoren ohne spezifische Koordinator-Umsetzung	105
6.1.3	Anforderungen an die IEC 61850-Übertragungszeit	107
6.1.4	Mathematische Berechnung von auftretenden Verzögerungen	111
6.1.5	Nachbildung eines Netzwerkes mit dem Framework VirtualSubstation	115
6.2	Durchführung der Validierung	117
6.2.1	Validierung der adäquaten Abbildung durch die Nutzung einer Virtualisierung	117
6.2.2	Validierung der Synchronität der Simulatoren ohne spezifische Koordinator-Umsetzung	123
6.2.3	Validierung der Anforderungen an die IEC 61850 Übertragungszeit	123
6.2.4	Validierung der mathematischen Berechnung von auftretenden Verzögerungen	126
6.2.5	Validierung der Nachbildung eines Netzwerkes mit dem Framework VirtualSubstation	143
6.3	Fazit der Validierung	147

7 Zusammenfassung und Ausblick	149
7.1 Zusammenfassendes Fazit	149
7.2 Reflexion der Forschungsfragen	151
7.3 Ausblick, mögliche Erweiterungen und Verbesserungen	152
Literaturverzeichnis	157

Abbildungsverzeichnis

1.1	Flussdiagramm zur Vorgehensweise innerhalb der Promotion	5
2.1	Einordnung von Simulationstypen nach Anzahl der Modelle und verwendeter Solver nach [8], [9]	9
2.2	Struktur und Teile einer Co-Simulation nach [10]	10
2.3	Aufgaben des Koordinators in einer Co-Simulation	11
2.4	Warteschlangenmodell eines einzelnen Bedieners nach [22]	18
3.1	Struktur einer Digitalen Station nach dem Standard IEC 61850	39
3.2	Struktur des Datenmodells nach beispielsweise [64]	42
3.3	Definition der Übertragungszeit nach IEC 61850-5	48
4.1	Schematischer Aufbau des Stromnetzsimulators (SNS)	56
4.2	Grundaufbau eines vIEDs	59
4.3	Grundlegende SDN-Architektur und Grundprinzip der Ebenen-Trennung	61
4.4	Virtualisierungskonzept des KNWS mit der Verschneidung aus vIEDs und einer SDN-Architektur mittels Containernet	64
4.5	Aufgaben und Umsetzung der jeweiligen Instanzen der Co-Simulation	66
4.6	Kopplung der Simulatoren mit Synchronisierung über den Koordinator	67
4.7	Umsetzung der Kopplung beider Simulatoren untereinander und mit dem Koordinator	68
5.1	Single-Controller-Architektur innerhalb der SDN-Infrastruktur	76
5.2	Ablaufpläne (a) der ryu-Implementierung und (b) der beschriebenen Erweiterung	79
5.3	Implementierter Programmablauf des vIED für den Überstromschutz	81
5.4	Workflow des formulierten Frameworks	83
5.5	Workflow des formulierten SCD-Parsers	85
5.6	Workflow der Containernet-Implementierung	87
5.7	Umsetzung der in Abschnitt 4.3 eingeführten Kopplung der Simulatoren	90
5.8	Relevante SDN-Prozesse innerhalb der Co-Simulation	92
5.9	Sequenzdiagramm des Vorwärtspfads von Paketen	96
5.10	Übertragungszeitkomponenten innerhalb der Co-Simulation für das Gesamtverzögerungsmodell	98

6.1	Aufbauten zur Validierung des Netzwerkverhaltens	103
6.2	Minimalbeispiel zur Validierung der Anforderungen zur Übertragungszeit	108
6.3	Evaluationsszenarien mit (i) der Funktionalität direkt laufend auf dem physischen Gerät, (ii) einer virtualisierten Funktion mit Standard Docker-Netzwerk und (iii) dem eingeführten Minimalbeispiel	109
6.4	Schematische Darstellung der Übertragungszeitmessung zwischen dem KNWS und SNS	111
6.5	Komponenten des Verzögerungsmodells im gewählten Versuchsaufbau	112
6.6	Programmablaufplan zur automatisierten Erstellung von Datensätzen unter Variation der Einflussvariablen	114
6.7	Stromverläufe in den drei Evaluationsszenarien mit Grenzwertüberschreitung und erfolgter Schalthandlung	124
6.8	Zeitkomponenten der Übertragungszeit bei fünf Testdurchführungen mit identischen Testparametern (oben: ohne Controller-Aktivität, unten: mit Controller-Aktivität nach Tabelle 6.8)	131
6.9	$\Delta t_{PD,GSE}$ in Abhängigkeit der Anzahl verwendeter Switches und Container in Validierungskategorie 1	134
6.10	Δt_{TT} in Abhängigkeit der Anzahl der Switches in den Fällen mit und ohne Controller-Aktivität unter Nutzung des mathematischen Verzögerungsmodells	135
6.11	$\Delta t_{PD,GSE}$ in Abhängigkeit der Anzahl verwendeter Switches und Sampled Values (SV)-Publisher in Validierungskategorie 2	136
6.12	Δt_{TT} in Abhängigkeit der Anzahl der Switches und SV-Publisher unter Nutzung des mathematischen Verzögerungsmodells	138
6.13	$\Delta t_{PD,GSE}$ in Abhängigkeit der Anzahl verwendeter Switches und Packet-In-Rate in Validierungskategorie 3	140
6.14	Δt_{TT} in Abhängigkeit der Anzahl der Switches und Übergabewahrscheinlichkeit unter Nutzung des mathematischen Verzögerungsmodells	142
6.15	Schematische Darstellung der verwendeten SCD-Datei und des daraus resultierenden Framework-Ergebnisses	144
6.16	Wireshark-Aufzeichnungen der im Rahmen der Validierung erfassten Kommunikationsströme (oben: GOOSE-Kommunikation, unten: MMS-Kommunikation)	146

Tabellenverzeichnis

2.1	Modellierungsarten für ereignisbasierte Systeme mit dem Fokus auf die zeitliche Beschreibung von Netzwerkverkehr	16
2.2	Fünf Ebenen der Beschreibung und Umsetzung von Co-Simulationen nach [29]	21
2.3	Übersicht über Standards und Frameworks für Co-Simulationen	34
3.1	IEC 61850 Normteile nach [32]	41
3.2	Synchronisierungs-Klassen nach IEC 61850-5	47
3.3	Übertragungszeitklassen nach IEC 61850-5	48
5.1	MAC-Adressbereiche für die Multicast-Protokollanwendungen nach IEC 61850	78
6.1	Validierungsgegenstände und -vorgehen	101
6.2	Verwendete Merkmale des Netzwerkverkehrs	106
6.3	Kategorien der Variationen für die Validierung des Verzögerungsmodells	116
6.4	Ausschnitt einer aufgenommenen PCAP-Datei der virtuellen Station	120
6.5	Ergebnisse der Kosinus-Ähnlichkeit der verglichenen Merkmale zwischen dem Aufbau der realen und virtuellen Station	121
6.6	Ergebnisse der Szenarien (i), (ii) und (iii)	126
6.7	Berechnung und Beschreibung der einzelnen Verzögerungskomponenten für die Validierungsdatensätze	128
6.8	Gemessene und berechnete Werte für fünf Durchgänge eines Datensatzes mit einem Switch und einem Container	129
6.9	Statistische Auswertung der Verzögerungskomponenten aus Tabelle 6.8	130

Abkürzungsverzeichnis

API	Application Programming Interface
CHiL	Controller-Hardware-in-the-Loop
CID	Configured IED Description
CIM	Common Information Model
CSV	Comma-Separated Values
DEVS	Discrete Event System Specification
EMT	Electromagnetic Transient
eth	Ethernet Netzwerkadapter
FIFO	First-In-First-Out
FMI	Functional Mockup Interface
FMU	Functional Mockup Unit
GOOSE	Generic Object Oriented Substation Events
GPS	Global Positioning System
HELICS	Hierarchical Engine for Large-scale Infrastructure Co-Simulation
HLA	High Level Architecture
HMI	Human Machine Interface
ICD	IED Capability Description
ICMP	Internet Control Message Protocol
IEC	International Electrotechnical Commission
IED	Intelligent Electronic Device

IGMP	Internet Group Management Protokoll
IID	Instantiated IED Description
IP	Internet Protokoll
KNWS	Kommunikationsnetzwerksimulator
LLDP	Link Layer Discovery Protokoll
MAC	Media-Access-Control
MDNS	Multicast Domain Name System Protokoll
MMS	Manufacturing Message Specification
MoSL	Mosaik Specification Language
MU	Merging Unit
NFV	Network Function Virtualization
NIC	Network Interface Card
NTP	Network Time Protocol
OMT	Object Model Template
OSI	Open Systems Interconnection
PCAP	Packet Capture
PD	Physical Device
PHiL	Power-Hardware-in-the-Loop
PIOC	Protection Instantaneous Overcurrent
PPS	Pakete pro Sekunde
PSM	Plug Setting Multiplier
PTOC	Protection Time Overcurrent

PTP	Precision Time Protocol
RMS	Root Mean Square
RTI	Run-Time-Infrastructure
RTT	Round-Trip-Time
SCD	Substation Configuration Description
SCL	Substation Configuration Language
SDN	Software Defined Network
SNS	Stromnetzsimulator
SSDPU	Substation Data Processing Units
STP	Spanning Tree Protokoll
SV	Sampled Values
TCP	Transmission Control Protocol
TMS	Time Multiplier Setting
UDP	User Datagram Protocol
vIED	Virtuelles Intelligent Electronic Device
VLAN	Virtual Local Area Networks
WAN	Wide Area Network
XML	Extensible Markup Language

Formelzeichenverzeichnis

$E(\Delta T)$	Erwartungswert einer Verzögerung
$E(\Delta T_C)$. . .	Erwartungswert der Verzögerung am Controller
$E(\Delta T_S)$. . .	Erwartungswert der Verzögerung am Switch
$E(\Delta T_i^{tot})$. . .	Erwartungswert der Interaktionszeit vom Switch und Controller
$E(\Delta T_{SDN})$. .	Erwartungswert der Verzögerung durch die SDN-Infrastruktur
$E(\Delta T_{S_i})$. . .	Erwartungswert der Verzögerung am Switch i
$E(\Delta T_{i_{prop}}^C)$. .	Erwartungswert der Verzögerung von Switch i zum Controller
$E(\Delta T_{i_{prop}}^{max})$. .	Erwartungswert der maximalen Flow-Mod-Verzögerung
Δt	Messbare Verzögerung
Δt_{KNWS}	. . .	Verzögerung innerhalb des KNWS
Δt_{Net}	Verzögerung zwischen den Simulatoren auf dem Netzkabel
$\Delta t_{PD,F}$	Verzögerung der Funktionsschicht des physischen Geräts
$\Delta t_{PD,GSE}$. .	Zeitdifferenz zwischen Paketerstellung und Messung am PD
$\Delta t_{PD,N}$	Verzögerung der Netzwerkschicht des physischen Geräts
Δt_{PD}	Verzögerung des physischen Geräts
$\Delta t_{PO,PI}$	Zeitdifferenz zwischen Packet-In- und Packet-Out-Ereignis
$\Delta t_{PO,s2}$	Zeitdifferenz zwischen Loopback und Switch beim Packet-Out
Δt_{RTT}	Zeitkomponente für die RTT
Δt_{SDN}	Verzögerung durch die SDN-Infrastruktur
Δt_{SNS}	Verzögerung innerhalb des SNS
Δt_{TT}	Verzögerung als Übertragungszeit nach IEC 61850

Δt_{acc} Genauigkeit der Synchronisierung
$\Delta t_{d,GSE}$ Zeitdifferenz für das Verpacken der GOOSE-Nachricht
$\Delta t_{d,s}$ Zeitdifferenz zwischen dem Switch und dem Container
Δt_{eth} Verzögerung vom realen Netzwerkadapter
Δt_{f_1} Zeitkomponente zur Lösung der Modellgleichung im SNS
$\Delta t_{in,KNWS}$ Zeitkomponente zum Entpacken eingehender Pakete im KNWS
$\Delta t_{in,sim}$ Zeitkomponente zum Entpacken eingehender Pakete im SNS
$\Delta t_{out,KNWS}$ Zeitkomponente zum Verpacken ausgehender Pakete im KNWS
$\Delta t_{out,sim}$ Zeitkomponente zum Verpacken ausgehender Pakete im SNS
$\Delta t_{prop,CN}$ Verzögerung zwischen Switch und vIED in Containernet
$\Delta t_{prop,SDN}$ feste Ausbreitungsverzögerung zwischen zwei SDN-Komponenten
$\Delta t_{prop,eth}$ Verzögerung zur Weitergabe an die Virtualisierung
$\Delta t_{s,PD}$ Zeitdifferenz zwischen dem Switch und realem Netzwerkadapter
$\Delta t_{s1,PI}$ Zeitdifferenz zwischen Switch und Loopback beim Packet-In
$\Delta t_{s2,s1}$ Zeitdifferenz am OpenFlow Switch des GOOSE-Pakets
Δt_{vIED} Verzögerung vom vIED
Δt_{veth} Verzögerung vom virtuellen Netzwerkadapter
Λ Gesamtankunftsrate
λ Ankunftsrate
μ Servicerate
d Störung
q Wahrscheinlichkeit
t Zeitabhängigkeit

t_a	Zeitanteil vom Kommunikationsstack von PD1 nach IEC 61850-5
t_b	Übertragungszeit zwischen PD1 und PD2 nach IEC 61850-5
t_c	Zeitanteil vom Kommunikationsstack von PD2 nach IEC 61850-5
t_k	Messbarer Zeitstempel
$t_{d,GSE}$	Erzeugter Zeitstempel vor Erstellen des GOOSE-Pakets
$t_{eth,PD}$	Zeitstempel am realen Netzwerkadapter
$t_{eth,d}$	Zeitstempel am Netzwerkadapter des Containers
$t_{eth,s1}$	Zeitstempel am Switch zur Verbindung des Containers
$t_{eth,s2}$	Zeitstempel am Switch für den realen Netzwerkadapter
$t_{lo,FM}$	Zeitstempel des Flow-Mod-Ereignis am Loopback Interface
$t_{lo,PI}$	Zeitstempel des Packet-In-Ereignis am Loopback Interface
$t_{lo,PO}$	Zeitstempel des Packet-Out-Ereignis am Loopback Interface
u	Eingang
x	Allgemein skalarer Zustand
\dot{x}	Ableitung des skalaren Zustandes

Definitionen

Ad-hoc-Kopplung Von einer **Ad-hoc-Kopplung** wird gesprochen, wenn kein bestehender Standards zur Kopplung von Simulatoren genutzt werden kann und eine Co-Simulation ohne die standardisierten Instanzen implementiert wird. Die Kopplung wird speziell für den Anwendungsfall ausgelegt und implementiert. Sie ist nicht verallgemeinerbar für andere Co-Simulationen.

Ankunftsrate Die **Ankunftsrate** beschreibt im Rahmen der Warteschlangenmodellierung die durchschnittliche Anzahl von Einheiten (beispielsweise Anfragen oder Netzwerkpakete), die pro Sekunde in ein System eintreffen und zur Verarbeitung anstehen.

Co-Simulation Eine **Co-Simulation** besteht aus der Theorie und den Techniken, um eine globale Simulation eines gekoppelten Systems durch die Zusammensetzung von unterschiedlichen Simulatoren zu ermöglichen. Jeder Simulator wird allgemein als eine Black Box definiert, der in der Lage ist, Systemverhalten nachzubilden, Inputs zu verarbeiten und Outputs zu generieren. Das gekoppelte System ist nicht limitiert auf die Anzahl und Beschreibungsart der Teilsysteme.

Digitale Station Umspannstationen oder auch Umspannwerke, die hinsichtlich der prozessrelevanten Kommunikation vollständig digitalisiert sind und sich vollständig am IEC 61850-Standard orientieren, werden als **Digitale Stationen** bezeichnet.

Echtzeitsimulation Eine **Echtzeitsimulation** ist dann möglich, wenn ein Simulator in der Lage ist, die Modellgleichungen in kürzerer Zeit zu lösen und die Ergebnisse bereitzustellen, als die Zeit, die für die Simulationsschrittweite erforderlich ist. Dies gewährleistet, dass die Lösungen der Modellgleichungen zu jedem Abschluss eines Simulationsschrittes verfügbar sind und das reale statische und dynamische Verhalten im entsprechenden Zeitabschnitt nachgebildet wird. Systeme, die eine variable Schrittweite verwenden, sind für diese Art der Simulation jedoch ungeeignet.

Ereignisgleichheit Als ein Resultat der Synchronisierung mehrerer Systeme werden die abgebildeten Vorgänge untereinander abgeglichen. Es wird sichergestellt,

dass ein Ereignis, das aus einem Vorgang resultiert, in jedem abgeglichenen System auftritt und dass auf Basis der Zeitgleichheit, der Ablauf der Ereignisse identisch ist. Folglich ist die **Ereignisgleichheit** zwischen den Systemen hergestellt.

IED Ein **IED** (Intelligent Electronic Device) ist ein Gerät mit einem oder mehreren Prozessoren und der Fähigkeit, Anwendungsfunktionen auszuführen, Daten lokal in einem Speicher zu speichern und Daten mit anderen IEDs (Quellen oder Senken) über eine digitale Verbindung auszutauschen.

Numerische Stabilität Ein Co-Simulationssystem ist **numerisch stabil**, wenn die Fehler, die aufgrund der Kopplung zwischen den Simulatoren und Synchronisierung entstehen, über die gesamte Simulationsdauer begrenzt sind und nicht zu einem exponentiellen Anstieg der Abweichungen führen. Dies bedeutet, dass das System in der Lage ist, die korrekten dynamischen Eigenschaften beizubehalten, selbst bei kleinen Störungen oder Änderungen in den Eingangsbedingungen.

Realzeitbetrieb Der Betrieb eines Rechensystems, bei dem Programme zur Verarbeitung anfallender Daten fortwährend betriebsbereit sind, sodass die Verarbeitungsergebnisse innerhalb einer vorgegebenen Zeitspanne und synchron zum Realsystem verfügbar sind, wird **Realzeitbetrieb** genannt. Die Daten können je nach Anwendungsfall nach einer zeitlich zufälligen Verteilung oder zu vorbestimmten Zeitpunkten anfallen.

Service rate Im Kontext der Warteschlangenmodellierung bezeichnet die **Service rate** die durchschnittliche Anzahl von Einheiten (beispielsweise Anfragen oder Netzwerkpakete), die ein Bediener pro Sekunde bearbeiten kann.

Synchronisierung Unter dem Begriff der **Synchronisierung** wird der Vorgang für die Erfüllung der Zeitgleichheit von Prozessen oder Systemen verstanden.

Zeitgleichheit Als ein Resultat der Synchronisierung sind die betreffenden Systeme zueinander zeitlich abgeglichen und orientieren sich an einer gemeinsamen Referenzzeit. Somit wird die **Zeitgleichheit** zwischen den Systemen hergestellt.

1 Einleitung

1.1 Motivation

In der elektrischen Energieversorgung spielt die Digitalisierung von Prozessen eine immer größere Rolle. Der Automatisierungs- und Autonomiegrad von elektrischen Netzen nimmt stark zu und besonders innerhalb von Forschungs- und innovativen Industrieprojekten sind diese von großer Bedeutung. Es besteht vor allem auch für das zukünftig dekarbonisierte Energiesystem die Notwendigkeit und der Bedarf der Erhöhung der Digitalisierung und Automatisierung der elektrischen Netze. Damit einhergehend ist die digitalisierte Umsetzung sämtlicher Prozesssteuerungen und -regelungen verbunden. Eine Vernetzung unterschiedlichster Systeme und der Datenaustausch zwischen diesen stellt eine notwendige Anforderung in dem Bestreben dar. Diverse Systeme und Komponenten kommunizieren miteinander und tauschen prozessrelevante Daten aus, um die notwendigen Funktionen der Energieversorgung zu realisieren. Im Vergleich zum Status Quo in der Energieversorgung stellen aktuelle und zukünftige technologische Entwicklungen zunehmende Anforderungen an die Kommunikationsnetzwerkinfrastruktur. Diese steht vermehrt im Mittelpunkt, da ihr Vorhandensein eine Voraussetzung für viele neuartige Entwicklungen und Technologien ist, die auf der Vernetzung von Komponenten aufsetzen.

Innerhalb der elektrischen Energieversorgung hat zum einen die reine Funktionsfähigkeit der Kommunikationsinfrastruktur hohe Priorität, aber vor allem erhöht sich ihr Einfluss auf den unterlagerten Energieprozess. Beeinflussungen, Ausfälle und Störungen vom Kommunikationsnetz haben direkte Konsequenzen für den eigentlichen Versorgungsprozess. Für Untersuchungen des Energieversorgungsprozesses inklusive der umliegenden Kommunikationsinfrastruktur und für den stabilen Systembetrieb wird das Verlangen einer gemeinschaftlichen Modellierungs- und Simulationsumgebung der beiden Domänen immer größer [1]–[3]. Die Aussagekraft von Analysen hängt maßgeblich davon ab, alle relevanten Einflussfaktoren mit einbezogen und detailliert abgebildet zu haben. Eine Wechselwirkung zwischen Technologien und Betrachtungsebenen, wie in der beschriebenen Abhängigkeit der elektrischen Energieversorgung von der Netzwerkinfrastruktur, fordert eine gemeinschaftliche Abbildung. Hierfür sind Simulationssysteme ein essenzieller Bestandteil, um unterschiedlichste wissenschaftliche Ansätze in einer Umgebung zu erproben und zu beweisen [4], [5].

Eine solche Umgebung setzt sich für Untersuchungen einer digitalen und vernetzten Energieversorgungsinfrastruktur aus unterschiedlichen Betrachtungsebenen zusammen, die sich hier in zwei Bereiche unterteilen lassen können: Stromnetze und Kommunikationsnetzwerke. Beide Betrachtungsebenen lassen sich vielfältig untersuchen. Dabei stehen unterschiedliche Detaillierungsgrade, zeitliche Betrachtungsrahmen, aber auch ganz unterschiedliche Fragestellungen, wie Planung von Netzen oder Analysen bestehender Netze im Vordergrund. Die eingeleitete Wechselwirkung der Energieversorgungsprozesse mit den Kommunikationsnetzen wird vor allem für die Betrachtungen des dynamischen Verhaltens elektrischer Energiesysteme in Verbindung mit der Kommunikationsinfrastruktur benötigt. Besondere Bedeutung hat die eingeführte Abhängigkeit für Stromnetze mit hohem Automatisierungsgrad. Hierzu zählen insbesondere Umspannwerke, auch Digitale Stationen genannt, die im weiteren Verlauf der Arbeit Gegenstand der Untersuchung sind. Daraus abgeleitet, behandelt die vorliegende Arbeit die Konzeption und Entwicklung einer gemeinschaftlichen Co-Modellierungs- und Simulationsplattform, bestehend aus Stromnetz- und Kommunikationsnetzwerksimulationen, um die Grundlage für komplexe domänenübergreifende Untersuchungsfragestellungen zu schaffen. Die Arbeit liefert demnach einen Beitrag für eine Echtzeit-Co-Simulation der Stationsautomatisierung in digitalisierten Umspannwerken.

1.2 Problemstellung

Allgemein besteht die Herausforderung für domänenübergreifende Betrachtungen darin, die unterschiedlichen Ebenen einzeln korrekt, aber besonders in der einander bedingenden und beeinflussenden Beziehung gemeinschaftlich abzubilden. Im Allgemeinen werden Betrachtungen je Untersuchungsziel voneinander getrennt und separat umgesetzt. Hierfür bildet eine Modellbeschreibung die Basis. Jegliche physikalische Untersuchungen lassen sich auf eine mathematische Beschreibung beziehen. Für Untersuchungen komplexer Modelle werden Hilfsmittel wie Simulationen herangezogen, um das Verhalten, abgeleitet aus dem aufgestellten Modell, in einer definierten Umgebung umzusetzen.

Das Mittel der numerischen Simulation soll innerhalb der Arbeit als Grundlage für Betrachtung des Untersuchungsziels dienen. Die vorliegende Arbeit behandelt die speziellen Anforderungen von digitalisierten Umspannwerken und der sogenannten Stationsautomatisierung als einen elementaren Teil des Stromnetzes. Hierfür grundlegend ist die Beschreibung des Anwendungsfalls und die Ableitung von Anforder-

rungen für ein gemeinschaftliches Simulationssystem. Dabei müssen die Spezifika beachtet werden, welche für eine korrekte und gültige Abbildung von Bedeutung sind und von einem gemeinschaftlichen Untersuchungssystem erfüllt werden müssen. Ein zu entwerfendes Gesamtsystem zur Untersuchung der Abhängigkeiten muss valide Ergebnisse für den Anwendungsfall liefern und dabei eine ausreichende Detailtiefe besitzen. Im Rahmen der Arbeit wird gezeigt, dass klassische, in der Literatur vorhandene oder kommerziell verfügbare Systeme diesen Anforderungen in der Regel nicht genügen. Daraus wird ein Bedarf identifiziert, ein solches System zu entwerfen und seine Gültigkeit nachzuweisen.

1.3 Forschungsfragen

Aus der beschriebenen Problemstellung und dem Ziel der Arbeit leiten sich die folgenden zu untersuchenden Forschungsfragen ab und bilden den roten Faden der Arbeit. Der jeweilige Bedarf zur Beantwortung der formulierten Forschungsfragen wird in den nachfolgenden theoretischen Grundlagen und der Übersicht an vergleichbaren Untersuchungen innerhalb der Literatur motiviert.

1. Wie ist ein Kommunikationsnetzwerk der Stationsautomatisierung für gesamtheitliche Analysen abstrahierbar und kombinierbar mit Stromnetzsimulationen?
2. Welche Herausforderungen treten bei der Kopplung der Simulation und der virtuellen Kommunikationsnetzwerkumgebung auf und wie ist die Einhaltung resultierender Zeitanforderungen sicherzustellen?
3. Wie sind Netzwerk- und Automatisierungskomponenten hinsichtlich des Netzwerkverhaltens in einer isolierten Umgebung zu modellieren und zu implementieren?

1.4 Vorgehensweise und Aufbau der Arbeit

Das in Abbildung 1.1 dargestellte Flussdiagramm zeigt das Vorgehen zur Bearbeitung der Aufgabenstellung mit der Entwicklung eines neuartigen Ansatzes einer Co-Simulation für den Anwendungsfall der Digitalen Station und den Prozessen der inkludierten Stationsautomatisierung. Dabei sind zunächst nach der Problemdefinition in Kapitel 1 die notwendigen Grundlagen zur Problemlösung in Kapitel 2 identifiziert und bearbeitet. Ausgehend von den Grundlagen der Co-Simulation werden die Anforderungen und Einschränkungen auf den hier relevanten Anwendungsfall der

International Electrotechnical Commission (IEC) 61850 basierten Stationsautomatisierung in Kapitel 3 hergeleitet und deren Anforderungen an eine zu entwerfende Co-Simulation definiert. Ausgehend von diesen Grundlagen untersetzten Problemdefinitionen und -anforderungen wird der neuartige Ansatz für die Co-Simulation der Digitalen Station in Kapitel 4 eingeführt und die relevanten Aspekte erarbeitet. Das neuartige Co-Simulationssystem erfordert eine besondere Vorgehensweise, um ein Untersuchungsobjekt adäquat abzubilden und zu modellieren. Die notwendigen Teilaspekte werden in Kapitel 5 ausgeführt. Nachfolgend wird auf die kritischen Faktoren und Gültigkeitseinschränkungen, resultierend aus dem Systementwurf, anhand einer mathematischen Verzögerungs-Beschreibung näher eingegangen. Hieraus wird eine Einschätzung zur Sicherstellung der zeitlichen Restriktionen auf Basis der zu virtualisierenden Infrastruktur ergänzt. Um explizit den entworfenen Ansatz bewerten und beweisen zu können, wird in Kapitel 6 ein Validierungskonzept aufgestellt und die Validierung an den einzelnen Aussagen entlang durchgeführt. Die vorliegende Arbeit endet in Kapitel 7 mit der Zusammenfassung, der Reflexion der Forschungsfragen und dem Ausblick für künftige Arbeiten, die mit dem neu entwickelten Co-Simulationssystem bearbeitet werden können.

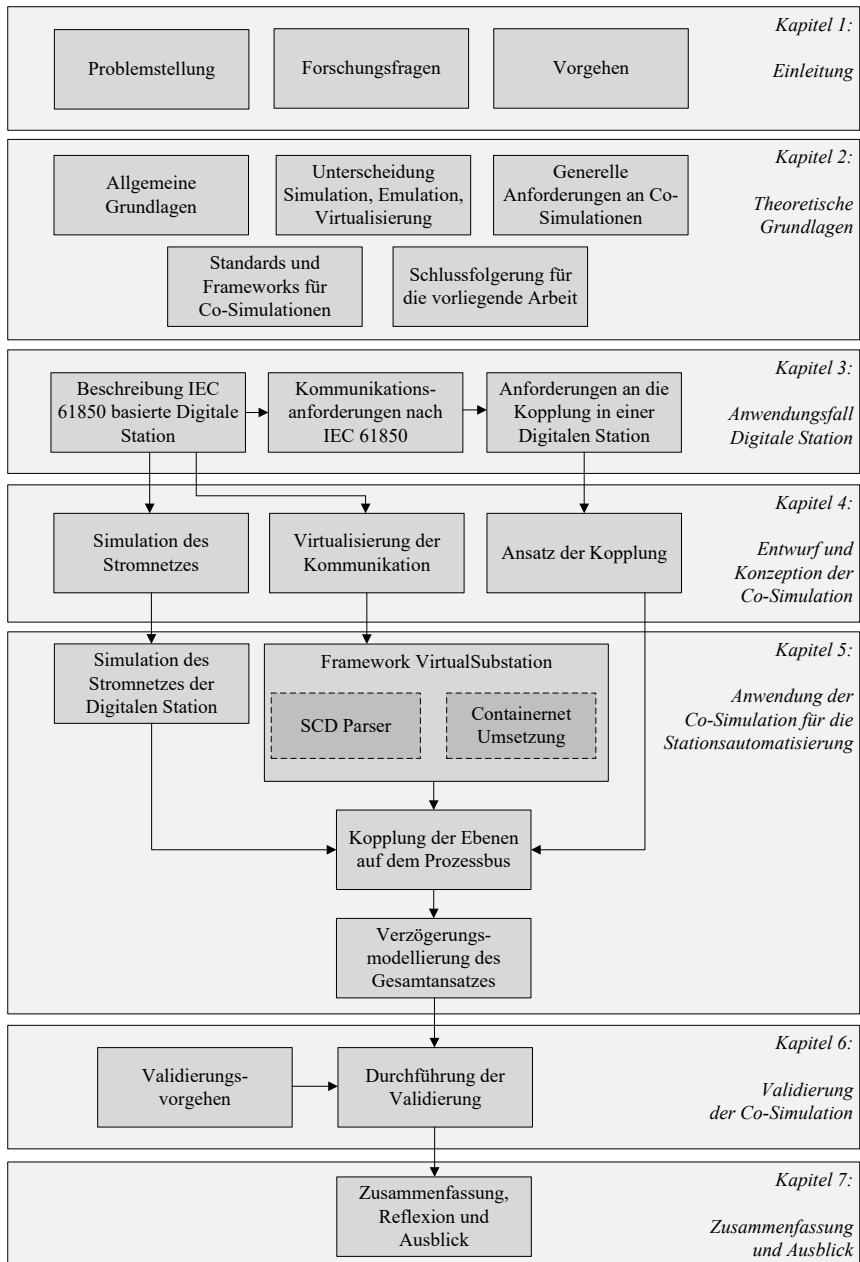


Abbildung 1.1: Flussdiagramm zur Vorgehensweise innerhalb der Promotion

2 Theoretische Grundlagen

Für die Bearbeitung des Themenkomplexes der Co-Simulation, ist zunächst die Definition des Begriffs notwendig, um die nachfolgenden Betrachtungen einordnen und zu anderen Arbeiten abgrenzen zu können. Eine allgemeine Definition wird nach [6] adaptiert und in dieser Arbeit verwendet.

Definition 2.1 *Eine **Co-Simulation** besteht aus der Theorie und den Techniken, um eine globale Simulation eines gekoppelten Systems durch die Zusammensetzung von unterschiedlichen Simulatoren zu ermöglichen. Jeder Simulator wird allgemein als eine Black Box definiert, der in der Lage ist, Systemverhalten nachzubilden, Inputs zu verarbeiten und Outputs zu generieren. Das gekoppelte System ist nicht limitiert auf die Anzahl und Beschreibungsart der Teilsysteme.*

Nachfolgend werden die relevanten theoretischen Grundlagen erörtert und die generellen Anforderungen an Co-Simulationssysteme herausgestellt. Anschließend wird eine Übersicht über unterschiedliche Anwendungsgebiete, sowie Co-Simulationstypen und -standards erarbeitet. Darauf basierend werden die Grundlagen für den Lösungsansatz in Kapitel 4 beschrieben und die Alleinstellungsmerkmale herausgearbeitet. Abschließend werden die Erkenntnisse eingeordnet und in Hinblick auf die beschriebene Problemstellung der Arbeit analysiert.

2.1 Allgemeine Grundlagen

Für die Untersuchung von Prozessen dient die mathematische Modellierung als Grundlage. Ein aufgestelltes mathematisches Modell ermöglicht die Untersuchung des statischen und dynamischen Prozessverhaltens in Abhängigkeit der zu berücksichtigenden Einflussgrößen. Zur Untersuchung des Prozesses auf Basis eines Modells muss der zugrundeliegende mathematische Zusammenhang identifiziert und entsprechend der Aufgabenstellung gelöst werden. Dies kann auf verschiedene Arten geschehen. Oftmals wird neben der Berechnung einer analytischen Lösung eine numerische Simulation verwendet. Es wird folglich eine schrittweise Lösung des in der Regel zeitabhängigen Problems implementiert. Dabei werden Simulationen oftmals für beeinflussbare Systeme mit Ein-Ausgangs-Beziehungen verwendet, indem das System in Abhängigkeit der Zeit und der Eingangsgrößen beschrieben und nachgebildet wird [7]. Ist der

nachzubildende Prozess sehr komplex und vielschichtig, kann es sinnvoll sein, den Prozess in Teilprozesse zu zerlegen, um eine hohe Detailtiefe der unterschiedlichen Betrachtungsbereiche erreichen zu können. Es werden folglich unterschiedliche Simulationen für verschiedene Teilbereiche des komplexen Prozesses definiert. Innerhalb einer Co-Simulation wird das zu betrachtende Gesamtsystem in verschiedene Einzelsysteme zerlegt, getrennt untersucht und untereinander unter Berücksichtigung der Koppelgrößen koordiniert. Die Aufteilung kann innerhalb einer Simulation geschehen, wenn beispielsweise ein komplexer Prozess in mehrere Teilsysteme untergliedert wird und daraus kleinere Teil-Simulationen geformt werden. Andererseits kann die Betrachtung unterschiedlicher Simulationsgegenstände ein Grund zur Unterteilung und Aufteilung einer Gesamtsimulation sein. Werden zudem unterschiedliche Domänen betrachtet, kann es gegebenenfalls sinnvoll sein, auf unterschiedliche Simulationssysteme beziehungsweise Simulatoren, mit den jeweils geeigneten Simulationseigenschaften zur Modellierung und Lösung des zu beschreibenden Sub-Prozesses zurückzugreifen. In beiden Fällen spricht man von einer Co-Simulation. Die unterschiedlichen Konstellationen von Simulationsgegenständen sind als Matrix in Abbildung 2.1 dargestellt. Wie in Abbildung 2.2 eingeführt wird, entspricht ein Simulator der Kombination aus einem Modell und einem Lösungsalgorithmus, dem sogenannten Solver. Der Begriff der Simulation entspricht hingegen der Nutzung eines Simulators zur Untersuchung eines Simulationsgegenstandes. Hierbei entsprechen die Modelle den zu beschreibenden Prozessen und die Solver den Simulationssystemen zur Lösung der Modellgleichungen. Eine genaue Erklärung wird in den folgenden Abschnitten eingeführt.

Innerhalb der Arbeit wird ausschließlich die Co-Simulation betrachtet, also mindestens zwei vollständig voneinander getrennte Simulatoren, die eigenständige Modelle und Solver besitzen. Eine hardwareseitige Trennung der Simulatoren ist laut Definition hingegen nicht notwendig, wird jedoch in vielen Anwendungsfällen durchgeführt. Es werden oftmals spezialisierte Hardwareplattformen für Einzelsimulationen genutzt. Aus der hardwareseitigen Trennung wird die Relevanz der später eingeführten Koordinatoren, oftmals auch Master-Algorithmus genannt, deutlich, da die Einzelsimulatoren häufig koordiniert und überwacht werden müssen [8].

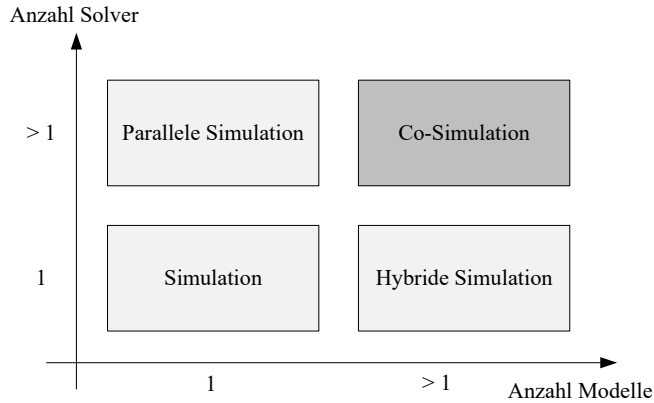


Abbildung 2.1: Einordnung von Simulationstypen nach Anzahl der Modelle und verwendeter Solver nach [8], [9]

2.1.1 Grundelemente von Co-Simulationen

Gemäß der Definition 2.1 setzen sich Co-Simulationen in der Regel aus zwei wesentlichen Komponenten zusammen: den eigenständigen Simulatoren und einem Koordinator. Jeder Simulator umfasst ein beschreibbares Modell sowie die numerische Lösung dieses Modells, die durch einen Solver realisiert wird. Beispielsweise wird ein Solver eingesetzt, der die Modellgleichungen mit numerischen Methoden in fest definierten Zeiten löst und somit Zustände und Ausgangsdaten berechnet. Das Modell wird im Zusammenspiel mit dem Solver als Simulator zusammengefasst. Nach der Definition 2.1 werden mindestens zwei solcher Simulatoren benötigt und im Rahmen der Co-Simulation miteinander gekoppelt. Hierfür wird in der Regel ein Koordinator verwendet, der die beiden Simulatoren überwacht und den Datenaustausch ermöglicht. Der Koordinator hat zudem, falls notwendig, auch die Aufgabe, die beiden Simulatoren zu synchronisieren und zu orchestrieren. Die formulierte Struktur der Co-Simulationsdefinition ist, angelehnt an [10], in Abbildung 2.2 dargestellt. Die einzelnen Bestandteile von Co-Simulationssystemen sind Simulator, Modell, Solver und Koordinator und werden nachfolgend kurz erläutert.

Simulator: Nach Abbildung 2.2 besteht eine Co-Simulation aus mindestens zwei separaten Simulatoren. Ein Simulator hat dabei insgesamt die Aufgabe, ein mathematisches Problem zu lösen, das beispielsweise einen physikalischen Prozess beschreibt.

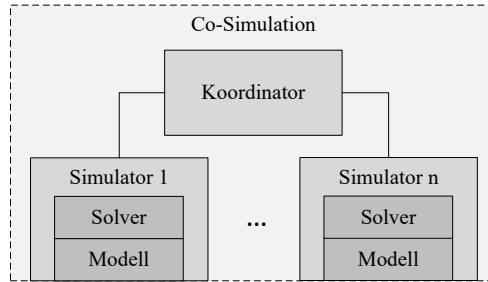


Abbildung 2.2: Struktur und Teile einer Co-Simulation nach [10]

Dabei ist es notwendig, dass mittels Modellbildung der zu beschreibende Sachverhalt mathematisch ausgedrückt wird und mithilfe des nachfolgend beschriebenen Solver berechnet wird. Der Simulator ist zumeist als eigenständiges Tool aufgebaut, in das sowohl das mathematische Modell als auch der Solver eingebettet sind. Über definierte Schnittstellen kann ein Benutzer die hinterlegte Simulation konfigurieren und bedienen. Zudem können Simulatoren auch weitere Schnittstellen für den Datenaustausch zu anderen Systemen besitzen. Für den Bereich der Co-Simulation ist dies ein zwingendes Kriterium, um die Simulationsergebnisse mit einem anderen Simulator teilen zu können.

Abgrenzend zu dem Begriff der Simulation, stellt der Simulator die Kombination aus dem Modell und dem Solver dar, mit dem die Simulation durchgeführt wird. Oftmals wird unter dem Simulator ein physisches Gerät verstanden, jedoch zeigt die eingeführte Definition, dass keine eigenständige Hardware hierfür notwendig ist. Eine Hardware kann mehrere Simulatoren bündeln.

Modell: Das notwendige mathematische Modell als Teilelement eines Simulators kann verschiedene Sachverhalte wie physikalische, elektrische oder chemische Prozesse beschreiben. In der Regel werden die nachzubildenden Prozesse mittels systemtechnischer Modellbildungsverfahren erzeugt. Dabei können die Modellbeschreibungen beliebig komplex sein und auf unterschiedlichen Modellansätzen beruhen. Zumeist, aber nicht darauf begrenzt, wird in statische und dynamische Modelle unterschieden [7]. Dabei werden mittels Simulationen dynamische Problemstellungen und dementsprechend dynamische Modelle mit dem Ziel, zeitliches Verhalten zu beschreiben, abgebildet. Neben den später einzuführenden kontinuierlichen, beziehungsweise diskretisierten Systemen und den Automaten systemen, modelliert als Warteschlangen-

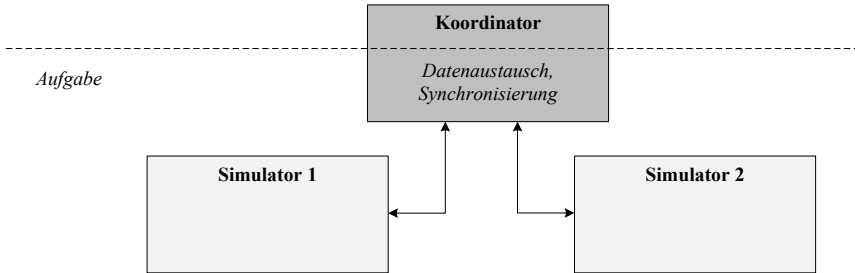


Abbildung 2.3: Aufgaben des Koordinators in einer Co-Simulation

modelle, können auch weitere Beschreibungsformen gewählt werden. In der Regel wird ein Gleichungssystem aufgestellt, das ein Anfangswertproblem darstellt und die Lösung das zeitliche Prozessverhalten beschreibt [8].

Solver: Der Solver beschreibt das zweite Teilelement des Simulators und dient der Lösung des formulierten mathematischen Problems. Abhängig von dem zu untersuchenden Prozess leitet sich die Art der mathematischen Modellierung ab um anschließend einen passenden Solver zur Problemlösung heranzuziehen. Hierbei werden unterschiedliche Solver-Arten unterschieden, die sich beispielsweise für lineare und nichtlineare Gleichungssysteme, Differentialgleichungen oder stochastische Modelle einsetzen lassen. Dabei werden, wie bereits erwähnt, oftmals numerische Solver eingesetzt, die Lösungsschritte unter Verwendung diskreter Werte und Rechenoperationen berechnen. Bekannte Methoden sind beispielsweise das Euler- oder Runge-Kutta-Verfahren, welche von gängigen Simulatoren eingesetzt werden. Mittels der implementierten Solver lassen sich die Modellgleichungen so in festen Berechnungsschritten lösen und die Ergebnisse für den nächsten Berechnungsschritt der Simulation verwenden [11].

Koordinator: Für die Anwendung der Kopplung von mehreren Simulatoren wird ein Koordinator eingeführt. Diesem werden zwei primäre Aufgaben zugeordnet, der Datenaustausch und die Synchronisierung [8], [10], wie in Abbildung 2.3 dargestellt ist. Durch die Erfüllung dieser beiden Aufgaben wird die Co-Simulation befähigt, einzelne Simulationen parallel und abhängig voneinander auszuführen.

Der Datenaustausch der Simulatoren ist essenziell, um die Kopplung technisch umzusetzen. Dabei wird vorausgesetzt, dass jeder Simulator an sich eine Datenschnitt-

stelle besitzt, um Informationen in einem definierten Datenformat herauszuführen und entgegenzunehmen. Es ist unerheblich, ob ein- und ausgehende Informationen den gleichen physischen Datenkanal nutzen oder eine Unterscheidung zwischen Ein- und Ausgang gemacht wird. Der Simulator muss lediglich technisch sicherstellen, dass er die Fähigkeit besitzt, Daten auszutauschen. Das definierte und dokumentierte Datenformat muss dem Koordinator vorliegen, um es weiter verarbeiten zu können. Es ist keine Voraussetzung, dass beide Simulatoren direkt das gleiche Datenformat nutzen. Wenn dies nicht der Fall ist, hat der Koordinator die Aufgabe, das ausgehende Datenformat eines Simulators in das notwendige Zielformat für den jeweils anderen Simulator zu überführen. Neben dem reinen Format der übertragenen Informationen steht dem Koordinator auch die Aufgabe beim Datenaustausch zu, die Informationen physikalisch weiterzuleiten und gegebenenfalls zwischenzuhalten, falls der Koordinator zusätzlich in einer bestimmten Weise die Simulatoren anstoßen muss.

Das Zwischenhalten und Anstoßen der Simulationen ist der direkte Anknüpfungspunkt zur zweiten grundlegenden Aufgabe des Koordinators – der Synchronisierung. Der Begriff der Synchronisierung, die sich sinngemäß auf die Zeitgleichheit bezieht, wird in der Folge um die Koordination der Ereignisse erweitert, um eine Ereignisgleichheit herzustellen. Die für die Arbeit angenommene Definitionen der Begriffe Synchronisierung, Zeitgleichheit und Ereignisgleichheit lauten folgendermaßen:

Definition 2.2 *Unter dem Begriff der **Synchronisierung** wird der Vorgang für die Erfüllung der Zeitgleichheit von Prozessen oder Systemen verstanden.*

Definition 2.3 *Als ein Resultat der Synchronisierung sind die betreffenden Systeme zueinander zeitlich abgeglichen und orientieren sich an einer gemeinsamen Referenzzeit. Somit wird die **Zeitgleichheit** zwischen den Systemen hergestellt.*

Definition 2.4 *Als ein Resultat der Synchronisierung mehrerer Systeme werden die abgebildeten Vorgänge untereinander abgeglichen. Es wird sichergestellt, dass ein Ereignis, das aus einem Vorgang resultiert, in jedem abgeglichenen System auftritt und dass auf Basis der Zeitgleichheit, der Ablauf der Ereignisse identisch ist. Folglich ist die **Ereignisgleichheit** zwischen den Systemen hergestellt.*

Die Erfüllung der Zeitgleichheit und Ereignisgleichheit wird zusammengenommen fortlaufend unter dem Begriff „Synchronisierung“ oder „synchron“ verstanden. Der Koordinator muss demnach zeitlich und kontextbezogen sicherstellen, dass beide Simulatoren synchron zueinander sind. Dies kann über verschiedene Maßnahmen sichergestellt werden, wie der Orchestrierung der Simulationsschritte der einzelnen Simulatoren. Dabei werden fest definierte Simulationsschritte der einzelnen Simulatoren vom Koordinator ausgelöst. Hier gibt es verschiedene Ansätze, wie der parallelen oder sequenziellen Ausführung der Simulatoren. In beiden Fällen stellt der Koordinator sicher, dass die definierten Anforderungen eingehalten werden und ein Simulationsschritt in der definierten Art initiiert und durchgeführt wird [8], [10]. In der Regel stellt die Orchestrierung der Simulatoren vor allem die Ereignisgleichheit her. Die Zeitgleichheit wird indirekt ebenfalls hergestellt. Wenn jedoch die beteiligten Simulatoren parallel voneinander ausgeführt werden, ohne jeweils vom Koordinator ausgelöst zu werden und die Zeitbasis maßgeblich für die Ausführung einer Simulation ist, so muss der Koordinator sicherstellen, dass beiden Simulatoren die identische Zeitbasis zur Verfügung steht und die Simulatoren auf diese abgeglichen sind. Für diese Aufgabe können beispielsweise klassische Synchronisierungstechniken für Endgeräte genutzt werden, die von den Simulatoren umgesetzt werden und folglich zueinander synchronisiert sind.

2.1.2 Modellierungsansätze

Nachfolgend wird der Begriff der Modellierung und mögliche Ansätze im Kontext der Simulationen ausgeführt, die sich zwischen verschiedenen Anwendungsfällen und Untersuchungsobjekten stark unterscheiden. Es werden grundlegende, für den Untersuchungsfall der hier zu erarbeitenden Co-Simulation, Modellierungsansätze eingeführt, um ein Verständnis für die später verwendeten Simulatoren zu schaffen. Dabei wird innerhalb dieser Arbeit auf die Grundansätze für die zwei verwendeten Untersuchungsebenen – der Stromnetzsimulation und der Netzwerksimulation – fokussiert.

Die Systembeschreibung ist zum einen maßgeblich für die mathematische Beschreibung des Prozesses, aber auch essenziell von der Simulationsart abhängig. Hier wird eine Unterscheidung zwischen zeitkontinuierlichen Systemen und ereignisbasierten Systemen, die diskrete Zustände annehmen können, getroffen. Die Modellierung und Abbildung dieser Beschreibungsformen werden nachfolgend erläutert.

2.1.2.1 Kontinuierliche Systeme

Eine kontinuierliche Systembeschreibung liegt bei zeit- und wertkontinuierlichen Modellierungen vor.

Ein allgemeines kontinuierliches System wird durch ein Zustandsdifferentialgleichungssystem beschrieben:

$$\dot{x} = f(x, u, d, t), \quad x(0) = x_0 \quad (2.1)$$

mit:

x	allgemein skalarer Zustand,
\dot{x}	Ableitung des skalaren Zustandes,
u	Eingang,
d	mögliche Störung,
t	Zeitabhängigkeit.

Im Allgemeinen bildet das Differentialgleichungssystem ein Anfangswertproblem mit der Abbildung von $x : \mathbb{R} \rightarrow \mathbb{R}$.

Diskretisierung: Durch die Fokussierung auf numerische Simulationssysteme wird eine Einschränkung in der nachfolgenden Betrachtung eingeführt. Die kontinuierlich beschriebenen Systeme und Modellansätze werden in der Regel durch eine numerische Simulation mit einer zeitlichen Diskretisierung gelöst, was eine feste Simulationsschrittweite nach sich zieht. Für die korrekte Abbildung von dynamischen Vorgängen kontinuierlicher Systeme ist unter Beachtung des Shannon-Abtasttheorems [12] die Schrittweite so klein wie möglich zu wählen. Die maximale Auflösung von dynamischen Phänomenen dementsprechend diesem zu berücksichtigen. Folglich wird eine definierte Zeitpunktfolge angelegt:

$$t_k < t_{k+1} < t_{k+2} \text{ für } k = 0, 1, 2, \dots \quad (2.2)$$

Zwischen zwei Abtastzeitpunkten liegt die Schrittweite $T_s = t_{k+1} - t_k$. Gleichung (2.1) wird adaptiert zu:

$$x(k+1) = f(x(k), u, d, k), \quad x(0) = x_0 \quad (2.3)$$

Ein abgetastetes System nach Gleichung (2.3) liefert somit die Zustandswerte für die Zeitpunkte t_k . Im Sinne der Simulation muss ein Solver für jeden Zeitpunkt t_k ein Ergebnis des zugrundeliegenden mathematischen Modells bereitstellen. Die Berech-

nungsschrittweite der numerischen Simulation gibt demnach die Abtastschrittweite der numerischen Simulation des kontinuierlichen Systems vor.

Im Rahmen der Co-Simulation stellt der Simulator für die Stromnetzsimulation eine Kombination aus einem kontinuierlichen Modell und einem numerischen Solver dar. Es resultiert eine schrittweise Lösung des Modells und diskrete Berechnungszeitpunkte mit einer festen Berechnungsschrittweite.

2.1.2.2 Ereignisbasierte Systeme

Unter ereignisbasierten Systemen werden Systeme verstanden, die von Ereignissen gesteuert werden und diskrete Zustände annehmen können. Verschiedene Modellierungsansätze können zur Beschreibung solcher Systeme herangezogen werden. Dabei stehen unterschiedliche Gesichtspunkte im Vordergrund. Oftmals werden Zustandsübergänge durch Automatenysteme beschrieben und modelliert [13], [14]. Hier steht vor allem im Fokus, unter welchen Umständen ein System einen definierten Zustand einnehmen kann und welche Auswirkungen daraus resultieren [15].

Für die nachfolgenden Betrachtungen innerhalb der Arbeit steht jedoch vor allem das zeitliche Verhalten von ereignisbasierten, diskreten Systemen im Vordergrund. Hier liegt der Betrachtungsschwerpunkt primär auf den Verarbeitungszeiten innerhalb eines Netzwerkes und den daraus resultierenden Zeitverläufen von einzelnen Netzwerkpaketen in einem Computernetzwerk. Die Netzwerkpakete können von mehreren Teilnehmern bearbeitet werden, wodurch eine Verkettung von Verarbeitungszeiten resultiert. Für die zeitliche Modellierung werden vor allem Warteschlangensysteme [16], [17] verwendet, die nachfolgend eingeführt und näher beschrieben werden. Alternative Modellierungsmethoden sind in Tabelle 2.1 mittels einer kurzen Beschreibung gegenübergestellt. Die dargestellten Methoden sind bereits eingeschränkt auf die mögliche Anwendung der Simulation von Computer- und Netzwerksystemen. Die Warteschlangenmodellierung eignet sich besonders, wenn die Teilnehmer und die Übertragungstrecken bekannt sind und demnach das Verhalten verlässlich abgeschätzt werden kann. Der vorliegende Anwendungsfall im Rahmen der Arbeit erfüllt diese Anforderung, wie in der nachfolgenden Beschreibung der Warteschlangensysteme erläutert wird. Folglich werden die weiteren Modellierungsarten aus Tabelle 2.1 nicht detaillierter eingeführt.

Warteschlangensysteme: In den späteren Betrachtungen wird ein Warteschlangenmodell genutzt, um die Verzögerung von Netzwerkpaketen innerhalb der Co-Simulation zu beschreiben. Die Pakete werden in einem Kommunikationsnetzwerk

Tabelle 2.1: Modellierungsarten für ereignisbasierte Systeme mit dem Fokus auf die zeitliche Beschreibung von Netzwerkverkehr

Modellierung	Referenz	Beschreibung
Stochastische Modelle	z. B. Iqbal [18]	Ende-zu-Ende Verzögerung bei Weitverkehrsnetzen
Messungsbasierte Modelle	z. B. Jarschel [19], [20]	Messungen der OpenFlow Controller und Vergleich anhand von Simulationen
Netzwerktheorie	z. B. Azodolmoky [21]	Nutzung eines mathematischen Frameworks der Netzwerktheorie (Network Calculus) für Verzögerungsberechnungen von SDN-Infrastrukturen

verschickt, empfangen und verarbeitet. Dafür dient die Automatenmodellierung als Grundlage, wonach jeder Bearbeitungsschritt durch einen Bediener mit festen Bearbeitungszeiten verstanden wird, wie beispielsweise in [22]–[24] dargelegt wurde. In der nachfolgenden Beschreibung wird bereits von eingehenden Netzwerkpaketen in das System ausgegangen. In der grundlegenden Definition von Warteschlangensystemen können auch beispielsweise Kunden oder Produkte betrachtet werden, die Eingangsobjekte des Warteschlangensystems sind. Im Bereich der Netzwerktechnik wird ein Netzwerkpaket von einem Teilnehmer in einer bestimmten und zumeist festen Zeit verarbeitet, nachdem es die Warteschlange passiert hat. Dabei kann ein Netzwerkteilnehmer neben den klassischen Sendern und Empfängern auch ein netzwerkformender Teilnehmer wie beispielsweise ein Router oder ein Switch sein. Ein Netzwerkpaket wird von diesen bearbeitet und weitergeleitet oder ausschließlich intern verarbeitet. Die genaue Tätigkeit eines jeden Teilnehmers, nachfolgend analog zur Theorie der Warteschlangenmodelle Bediener genannt, ist dabei nicht von Interesse. Es wird lediglich definiert, dass die Aufgabe in einer definierten Zeit erledigt wird.

Bevor der Verarbeitungsschritt erfolgen kann, wird davon ausgegangen, dass Netzwerkpakete sequenziell verarbeitet werden. Ist der Bediener noch mit einer Verarbeitung eines anderen Netzwerkpakets beschäftigt, muss sich das neue Paket in die Warteschlange stellen, bevor es selbst verarbeitet werden kann. Es folgt eine Wartezeit, die sich über die Logik einer Warteschlange modellieren lässt unter Berücksichtigung der Bearbeitungszeit, beziehungsweise der daraus abgeleiteten Service rate μ und der Ankunftsrate λ der jeweiligen Bediener. Es gelten für die Begriffe der Service- und Ankunftsrate folgende Definitionen:

Definition 2.5 *Im Kontext der Warteschlangenmodellierung bezeichnet die **Service-rate** die durchschnittliche Anzahl von Einheiten (beispielsweise Anfragen oder Netzwerkpakete), die ein Bediener pro Sekunde bearbeiten kann.*

Definition 2.6 *Die **Ankunftsrate** beschreibt im Rahmen der Warteschlangenmodellierung die durchschnittliche Anzahl von Einheiten (beispielsweise Anfragen oder Netzwerkpakete), die pro Sekunde in ein System eintreffen und zur Verarbeitung anstehen.*

In der Regel wird das First-In-First-Out (FIFO) Prinzip für Warteschlangen verwendet. Die Ankunftsrate der Netzwerkpakete wird durch Verteilungsfunktionen beschrieben, die meist auf stochastischen Prozessen basieren, etwa aufgrund des zufälligen Eintreffens der Pakete oder anderer Eingangsgrößen. Als Grundmodell wird für Warteschlangen oftmals ein Markov-System formuliert, das für die Ankunfts- und Bearbeitungsprozesse gilt. Folglich kann ein Erwartungswert des Zeitverhaltens definiert werden, der von dem stochastischen Eingangsverhalten abhängig ist [16].

Die Länge der Warteschlange hat direkte Auswirkung auf den Zeitbedarf des Prozesses. In der Regel ist die Warteschlange eines Bedieners von endlicher Kapazität. Es können nur bestimmte viele Pakete in der Warteschlange vorgehalten werden. Ist die maximale Kapazität erreicht, muss das neu ankommende Paket fallen gelassen werden, da keine technische Möglichkeit der Zwischenspeicherung mehr besteht. Ab diesem Zeitpunkt kann also keine endliche Zeit der kompletten Bearbeitung inklusive der Wartezeit beschrieben werden. Das Überlaufen der Warteschlange folgt, wenn die Ankunftsrate größer als die Servicerate des Systems ist.

Insgesamt lässt sich ein beispielhafter Bedienerprozess auf Basis eines Warteschlangenmodells nach folgender Gleichung beschreiben und ist in Abbildung 2.4 dargestellt:

$$E(\Delta T) = \frac{1}{\mu - \lambda} \quad (2.4)$$

mit:

$E(\Delta T)$	erwartete Verarbeitungszeit, inklusive der Warteschlange,
μ	Servicerate,
λ	Ankunftsrate.

Für die spätere Notation in dieser Arbeit wird abweichend von der verwendeten Literatur die Verarbeitungszeit, wie auch alle anderen Zeitelemente als Verzögerung, also als Zeitdifferenz verstanden. Folglich werden diese mit Δt , beziehungsweise für

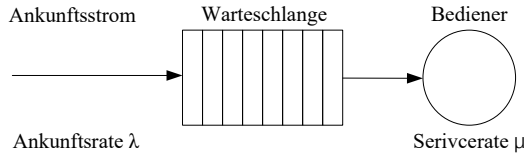


Abbildung 2.4: Warteschlangenmodell eines einzelnen Bedieners nach [22]

den Erwartungswert $E(\Delta T)$, beschrieben, um den Unterschied zu den Zeitpunkten t_k zu festigen.

Dieses grundlegende Modell kann um die genaue Ankunftsrate beziehungsweise deren Wahrscheinlichkeitsverteilung erweitert werden und für eine Verkettung von Bedienern mit individuellen Ankunfts- und Serviceraten herangezogen werden. Bei komplexen Netzwerkprozessen durchläuft ein Netzwerkpaket beispielsweise mehrere Instanzen und wird von unterschiedlichen Bedienern, die jeweils einen eigenen Warteschlangenprozess beschreiben, bearbeitet. Die vollständige Übertragung des Pakets von Sender zum Empfänger kann folglich mehrere dieser Prozessschritte beinhalten und die Übertragungszeit von weiteren Übergangswahrscheinlichkeiten abhängig sein.

Das eingeführte Warteschlangenmodell wird im Rahmen der Arbeit für die Simulation und Beschreibung des Verhaltens der Kommunikationsnetzwerkebene verwendet. Die genauere Beschreibung, Verwendung und Adaption des Warteschlangenmodells werden in den nachfolgenden Kapiteln adressiert.

2.2 Unterscheidung Simulation, Emulation und Virtualisierung

In den folgenden Abschnitten werden die Umsetzungen der ereignisbasierten Systeme für die Betrachtung der Kommunikationsnetzwerkebene für die nachfolgende Co-Simulation weiter beschrieben. Besonders für die Untersuchung und Abbildung von ereignisbasierten Netzwerksystemen gibt es unterschiedliche Konzepte, die sich in drei Kategorien – Simulation, Emulation und Virtualisierung – einordnen lassen. Die Kategorien werden nachfolgend kurz eingeführt und ihre Unterscheidungsmerkmale herausgearbeitet.

2.2.1 Simulation

Die Methode der Simulation ermöglicht im Rahmen einer definierten Umgebung das statische und dynamische Verhalten realer Systeme mathematisch in Form von Modellen zu analysieren. Dabei werden Simulationen herangezogen, um komplexe Problemstellungen zu untersuchen und mittels definierter, oftmals numerischer, Verfahren zu lösen. Im Vergleich zur rein analytischen Betrachtung resultieren einige Einschränkungen, aber vor allem viele Vorteile, wenn die mathematische Basis und die korrekte Beschreibung der physikalischen Zusammenhänge bereits nachgewiesen und korrekt erfolgt ist. Simulationen bilden in erster Linie Verhaltensweisen von Systemen ab, die sich über Systemantworten modellieren lassen [15]. In der Regel wird ein System innerhalb der Simulation untersucht, wenn Eingangsgrößen eine Veränderung der abgebildeten Zustände und Ausgänge nach sich zieht. Eine Simulation reagiert dementsprechend auf Eingangsgrößen und produziert berechnete Ausgangsgrößen. Dabei können Eingangssignale unter Nutzung von definierten Schnittstellen außerhalb des Simulators erzeugt und als externe Größe übergeben werden. Aber auch intern erzeugte Eingangssignale können von dem Simulator verarbeitet werden. Hierüber ist es in der Regel möglich, fest bestimmte und definierte Szenarien zu simulieren und deren Ein-Ausgangsverhalten abzubilden. Der Begriff der Simulation ist demnach unabhängig von der Art der Systemmodellierung. Um ein System simulieren zu können, ist eine mathematische Modellierung des Prozesses zu verwenden, die vom Simulator interpretiert werden kann und über eine zeitliche Abfolge von Eingangswerten die Ausgangs- und gegebenenfalls Zustandsgrößen berechnet.

2.2.2 Emulation

Eine Emulation stellt nach [25] eine Technik dar, die den Anspruch hat, eine Lücke zwischen der Simulation und den Tests in realen Experimentierumgebungen zu schließen. Hierfür wird die Funktion der abzubildenden Prozesse möglichst detailgetreu nachgebildet und in einer Emulationsumgebung implementiert. Es lässt sich die Verhaltensweise in einer kontrollierten Umgebung in Echtzeit reproduzieren und kann an externe Systeme angebunden werden. Dementsprechend hat die Emulation nicht die Aufgabe, eine Modellgleichung, wie es bei der Simulation der Fall ist, zu implementieren und das Verhalten auf Basis der Lösung dieser zu erzeugen, sondern die einzelnen Komponenten werden individuell nachgebildet und das Ergebnis der Emulation mittels einer Ausführung erzeugt. Eine Emulation folgt entsprechend der verwendeten Implementierungen und Nachbildung des Prozesses und ist maßgeblich

abhängig von dieser. Eine modellbasierte Vorhersage von Verhaltensweisen ist nicht angestrebt. Die Netzwerkemulation stellt nach [25] ein Beispiel der Emulation dar, in dem über dedizierte Emulatoren das Netzwerkverhalten nachgebildet werden kann.

2.2.3 Virtualisierung

Die Virtualisierung ist eine spezielle Umsetzung der Emulation, die beschreibt, wie virtuelle Instanzen eines Systems erstellt und ressourcentechnisch voneinander isoliert werden, um eine effiziente Nutzung der Hardware zu ermöglichen. Somit ist die Virtualisierung ein Teilbereich der Emulation. Eine Definition kann grundlegend von [26] und [27] verwendet werden, wonach die Virtualisierung ein Hardware-Software-Duplikat einer real existierenden Maschine darstellt. Dabei kann eine „nicht-triviale Teilmenge der Befehle der virtuellen Maschine direkt auf dem Host-Rechner ausgeführt werden“. Die virtuelle Maschine als Instanz einer Virtualisierung stellt also für eine Emulation die Möglichkeit bereit, die Befehle und Funktionen in einer isolierten Umgebung auf einem Host-Rechner auszuführen und nachzubilden.

Neben den klassischen virtuellen Maschinen existiert das Konzept der Container als virtuelle Instanzen. Dabei nutzen die Container gemeinschaftlich das Betriebssystem des Host-Rechners, wohingegen klassische virtuelle Maschinen ein eigenes virtuelles Betriebssystem aufbauen. Folglich wird bei einem Container unter Nutzung eines Linux-basierten Host-Rechners eine Umgebung des Namespaces isoliert und nutzt das Host-Betriebssystem. Die einzelnen Prozesse der Container werden in der Ausführungslogik des Host-Rechners integriert. Die Isolation führt jedoch dazu, dass ein Container als eine abgeschlossene virtuelle Instanz betrachtet werden kann, in der wiederum Funktionen und Verhaltensweisen implementiert werden können [28]. Da in der Folge der Arbeit Container als Virtualisierungslösungen verwendet werden, wird auf die nähere Beschreibung der virtuellen Maschinen nicht weiter eingegangen.

2.3 Generelle Anforderungen an Co-Simulationen

Neben den eingeführten Grundlagen für eine Co-Simulation und Erklärung der möglichen Simulatoren und deren Eigenschaften geht es in der Folge um generelle Anforderungen an Co-Simulationen.

Nach [29] werden unterschiedliche Abstraktionsebenen für Co-Simulationen in Form eines 5-Ebenen-Modells definiert und verschiedene Aspekte der Co-Simulation entsprechend Tabelle 2.2 dargestellt.

Tabelle 2.2: Fünf Ebenen der Beschreibung und Umsetzung von Co-Simulationen nach [29]

Ebene	Erklärung	Herausforderungen
Konzeption	Höchste Ebene, in der Modelle als „Black-Boxes“ angenommen werden und ein Framework entwickelt wird	Generische Struktur des Frameworks; Meta-Modelling der Komponenten
Semantik	Beschreibt die Bedeutung und die Rolle des Co-Simulation-Frameworks hinsichtlich Fragen des untersuchten Systems und des untersuchten Phänomens.	Bedeutung der einzelnen Modelle; Interaktion zwischen den Modellen; Bedeutung jeder Interaktion.
Syntax	Beinhaltet Formalismus des Co-Simulation-Frameworks	Formalisierung der einzelnen Modelle unterschiedlicher Domänen; Spezifikation des Umgangs mit Domänen-Unterschieden
Dynamik	Umsetzung des Frameworks durch Beschreibung der Synchronisierung und Harmonisierung unterschiedlicher Modelle und Simulationen	Detailtiefe der Berechnungen und Kausalität der Modelle; Harmonisierung unterschiedlicher Modellierungsarten; Synchronisierung und potenzielle Konflikte gleichzeitiger Ereignisse
Technologisch	Beschreibung von Details der Umsetzung und Evaluierung der Simulation	Verteilte oder zentrale Implementierung; Robustheit der Simulation; Vertrauenswürdigkeit und Effektivität der Simulation

Es lässt sich hieraus ableiten, dass eine Co-Simulation nicht ausschließlich die reine Kopplung von zwei Simulatoren beschreibt, sondern die Entwicklung eines gemeinsamen Frameworks gegebenenfalls auch Standards oftmals im Vordergrund steht. Vor allem in der Semantik- und Syntax-Ebene ist die Beschreibung des eigentlichen Simulationsmodells von großer Bedeutung. Über gemeinsame und gegebenenfalls standardisierte Datenmodelle soll ein gemeinsames Verständnis des Simulationsgegenstandes und die spätere Etablierung vom Datenaustausch sichergestellt werden. Hierbei kommen vor allem Datenmodelle wie das Common Information Model (CIM) [30], [31] oder IEC 61850 [32] zum Tragen, um die Parameter der abgebildeten Modelle zu beschreiben und unter den Simulationssystemen auszutauschen. Auch die automatisierte Konfiguration der Simulationssysteme auf Basis eines Datenmodells spielt hier oft eine Rolle [33], [34].

Fragestellungen der reinen Implementierung und Bewertung der gemeinschaftlichen Co-Simulation finden sich in den Ebenen der Dynamik und Technologie nach Tabelle 2.2 wieder. Hier steht vor allem die korrekte und robuste Kopplung im Vordergrund, sodass die gemeinschaftliche Simulation vertrauenswürdige Ergebnisse liefert und weiterführende Analysen ermöglicht.

Diese robuste Kopplung soll fortan im Vordergrund stehen und ist maßgeblich davon abhängig, dass die einzelnen Simulationssysteme zueinander synchron sind und der Datenaustausch geregelt wird, sodass eine Stabilität nicht durch die Zusammenführung der Simulatoren gefährdet wird. Diese Annahme verringert die Allgemeingültigkeit der folgenden Betrachtungen, wird jedoch in den nächsten Abschnitten als notwendig für die vorliegende Arbeit begründet. Zusätzlich spielt in verschiedenen Anwendungen die Echtzeitfähigkeit der einzelnen Simulatoren und der gemeinschaftlichen Co-Simulation eine wesentliche Rolle – besonders bei dynamischen Prozessen und Anwendungsfällen. Im besonderen Maße für hybride Systeme, also der Vereinigung aus zwei unterschiedlich modellierten Systemen, sind diese Anforderungen von großer Bedeutung und wurden beispielsweise anhand des Functional Mockup Interface (FMI) Standards in [35] näher untersucht. Nachfolgend werden die drei genannten Anforderungen näher eingeführt und beschrieben.

2.3.1 Synchronität

Die Synchronität bezieht sich nach der eingeführten Definition 2.2 auf die beiden Aspekte der Zeit- und Ereignisgleichheit nach den Definitionen 2.3 und 2.4. Eine wichtige Voraussetzung für die Zusammenführung und Kopplung zweier oder mehrere Simulationssysteme zum Verbund der Co-Simulation stellt die Zeitgleichheit dar, die sicherstellt, dass Simulationen zeitlich gleich ablaufen. Es wird eine gemeinsame Zeitbasis benötigt, damit die Simulationen in der gleichen Geschwindigkeit ablaufen oder zumindest die auftretenden Ereignisse und Verläufe einer gemeinsamen Zeitbasis zuzuordnen sind. Die Ereignisgleichheit besagt, dass ein auftretendes Ereignis in allen Teilsystemen der Co-Simulation eintreten muss und potenzielle Auswirkungen in allen Systemen geschehen können. Generell lassen sich drei unterschiedliche Synchronisierungsmethoden nach [36] feststellen.

- Master-Slave: Die zu koppelnden Simulationen werden in Master und Slave unterteilt, wobei die Master-Simulation höhere Priorität besitzt und die einzelnen Co-Simulationsschritte steuert. Ein Master-Simulationssystem kann sowohl ein einzelner Simulator als auch zeitgleich der Koordinator sein. Die Slave-
-

Simulation wird aus der Master-Simulation heraus angestoßen. Es lassen sich verschiedene Arten der Ansteuerung (zeit- oder ereignisgetrieben) umsetzen.

- **Zeitgesteuert:** Die zu koppelnden Simulationssysteme arbeiten unabhängig voneinander und tauschen Daten zu festen Synchronisierungszeitpunkten aus. Falls Ereignisse, die für den Datenaustausch oder das jeweils andere System relevant sind, zwischen den festen Zeitpunkten stattfinden, muss ein solches Ereignis in gewisser Art zwischengespeichert werden, um die Informationen beim nächsten Synchronisierungszeitpunkt zu übermitteln. Um dies möglichst fehlerfrei und vollständig umzusetzen, ist ein Koordinator zwischen den Systemen notwendig.
- **Ereignisgesteuert:** Eine Auflistung globaler Ereignisse wird vordefiniert, um Synchronisierungspunkte zu erstellen. Das Eintreffen solcher Ereignisse folgert eine Synchronisierung zwischen den Systemen mit verbundenen Datenaustausch. Ein globaler Scheduler wird als Funktion des Koordinators benötigt, von dem die Geschwindigkeit der Co-Simulation abhängig ist.

Die Synchronität von hybriden Co-Simulationen wird besondere Bedeutung beigemessen. Grund ist die unterschiedliche Beschreibungsart der Teilsysteme. Während kontinuierliche Simulationen zeitlich feste Berechnungsschritte haben, ist eine Synchronisierung zu diesen fest definierten Zeiten vergleichsweise einfach umzusetzen und eine zweite Simulation auf diese Zeitpunkte abzugleichen. Die ereignisbasierte Simulation hingegen besitzt diese feste Abtastung nicht, was zu einer Herausforderung wird, um die oben genannten Synchronisierungsmethoden umzusetzen.

Zusammenfassend hat der Koordinator einer entsprechenden Co-Simulation die Aufgabe, sowohl die Zeit- als auch die Ereignisgleichheit unter genannten Herausforderungen sicherzustellen.

2.3.2 Stabilität

Die Stabilität von Untersuchungen unter Nutzung von Co-Simulationsansätzen spielt eine wichtige Rolle für die Bewertung der Simulationsergebnisse und deren Aussagekraft. Besonders hybride oder schaltende Systeme stehen hier im Fokus, da Instabilitäten entstehen können, selbst wenn die einzelnen Modelle innerhalb der Teilsimulationen stabil scheinen [4]. In der Literatur wird vornehmlich die numerische Stabilität von Co-Simulationen in Betracht gezogen, in der die Auswirkungen der numerischen Solver betrachtet werden. Folglich wird in Anlehnung an beispielsweise Gomes [4] folgende Definition der Stabilität von Co-Simulationssystemen angenommen:

Definition 2.7 Ein Co-Simulationssystem ist **numerisch stabil**, wenn die Fehler, die aufgrund der Kopplung zwischen den Simulatoren und Synchronisierung entstehen, über die gesamte Simulationsdauer begrenzt sind und nicht zu einem exponentiellen Anstieg der Abweichungen führen. Dies bedeutet, dass das System in der Lage ist, die korrekten dynamischen Eigenschaften beizubehalten, selbst bei kleinen Störungen oder Änderungen in den Eingangsbedingungen.

Die Stationsautomatisierung als mögliches Untersuchungsobjekt beinhaltet solche schaltenden Vorgänge beispielsweise durch die Integration von Reglern auf Basis der messbaren Momentanwerte. Insbesondere spielt die Einhaltung der Stabilität eine besondere Rolle für Echtzeit-Systeme, die ein dynamisches Verhalten und einen hybriden oder schaltenden Charakter aufweisen. Ein geeigneter Koordinator hat die Aufgabe, die Stabilität für den Verbund der Simulatoren zum Beispiel durch eine Anpassung der Kommunikationsschrittweite sicherzustellen [4]. Der Koordinator der Co-Simulation muss demnach in der Lage sein, geeignete Strategien zu implementieren. Für die Einhaltung und Sicherstellung der Stabilität solcher gekoppelten Systeme lassen sich angepasste Stabilitätskriterien, wie das Nyquist-Kriterium auf Basis von numerischen Berechnungen für Echtzeit-Co-Simulationen nach [37], heranziehen.

2.3.3 Echtzeitfähigkeit

Die Anforderung der Echtzeitfähigkeit lässt sich sowohl auf die einzelnen Simulatoren als auch auf den Verbund im Rahmen einer Co-Simulation formulieren.

Der Begriff Realzeitbetrieb ist in DIN 44300 [38], wie folgt, definiert:

Definition 2.8 Der Betrieb eines Rechensystems, bei dem Programme zur Verarbeitung anfallender Daten fortwährend betriebsbereit sind, sodass die Verarbeitungsergebnisse innerhalb einer vorgegebenen Zeitspanne und synchron zum Realsystem verfügbar sind, wird **Realzeitbetrieb** genannt. Die Daten können je nach Anwendungsfall nach einer zeitlich zufälligen Verteilung oder zu vorbestimmten Zeitpunkten anfallen.

In der Folge wird die Definition des Echtzeit-Begriffs für Simulationen nach [39] und [40] adaptiert. Die Autoren definieren besonders Systeme mit diskreter Zeit und konstanter Simulationsschrittweite als geeignete Echtzeitsimulatoren. Die Zeitschritte zur Berechnung der Modellgleichungen sollen sequenziell nacheinander und synchronisiert mit einer realen Uhr durchgeführt werden. Feste Schrittweiten der Simulation werden als Voraussetzung genannt, um für jeden Simulationsschritt die Einhaltung

der Echtzeit garantieren zu können. Nach [39] und [40] wird Echtzeit so definiert, dass der Simulator in der Lage ist, die Modellgleichungen in einer geringeren Zeit zu lösen und das Ergebnis bereitzustellen, als die Simulationsschrittweite an Zeit benötigt. Damit wird garantiert, dass die Lösung der Modellgleichung zum Abschluss eines jeden Simulationsschrittes zur Verfügung steht. Für Simulationsverfahren mit variabler Schrittweite ist die Einhaltung der Echtzeitbedingung nicht ohne Weiteres möglich.

Die Echtzeit beschreibt also keine fest vorgegebene Simulationsschrittweite, sondern ist abhängig vom Untersuchungsgegenstand. Dementsprechend wird in [39] neben der reinen Schrittweite der Simulation auch auf die gesamte Zeitspanne der Simulation verwiesen. Die Echtzeitsimulation muss in der identischen Zeitspanne wie das nachzubildende physische System die notwendigen Zustände und Ausgaben berechnen können. Demnach kann Echtzeit für langsame Systeme eine andere zeitliche Bedeutung haben als für sehr dynamische Systeme. Maßgeblich ist für die Echtzeitsimulation, dass die realen Zustandsänderungen inklusive der Ein- und Ausgaben in einer Zeit berechnet werden können, um zu einer ausreichenden Genauigkeit das reale Verhalten nachzubilden. Maßgeblich für die Fähigkeit eines Simulators, die Modellgleichungen in der für Echtzeit notwendigen Zeit berechnen zu können, sind die mathematische Komplexität der Modellgleichungen und die verfügbare Berechnungsleistung der Simulator-Plattform.

Zusammengefasst lautet die hier angepasste Definition der Echtzeit von Simulationssystemen:

Definition 2.9 *Eine **Echtzeitsimulation** ist dann möglich, wenn ein Simulator in der Lage ist, die Modellgleichungen in kürzerer Zeit zu lösen und die Ergebnisse bereitzustellen, als die Zeit, die für die Simulationsschrittweite erforderlich ist. Dies gewährleistet, dass die Lösungen der Modellgleichungen zu jedem Abschluss eines Simulationsschrittes verfügbar sind und das reale statische und dynamische Verhalten im entsprechenden Zeitabschnitt nachgebildet wird. Systeme, die eine variable Schrittweite verwenden, sind für diese Art der Simulation jedoch ungeeignet.*

Das Kriterium der Echtzeitfähigkeit ist nachfolgend auf zur Laufzeit ausgeführte Simulationen, sogenannte Online-Simulationen, anzuwenden und zu bewerten. Diese sind zu einer realen Uhr synchronisiert, um einen realen physischen Prozess abzubilden. Für die Emulation, wie beispielsweise das Netzwerkverhalten, hat die Echtzeit die Bedeutung, dass die verwendete Ausführungsumgebung für die Emulation eine ausreichend kleine Berechnungszeit sicherstellt, um die Prozessschritte nachzu-

bilden. Die Emulation eines Prozesses darf nicht mehr Zeit in Anspruch nehmen als der äquivalente Prozess in der realen Umgebung [41]. Bei der Emulation ist für die Schrittweite der Ausführung der interne Scheduler der ausführenden Rechenplattform entscheidend. Dedizierte Echtzeitbetriebssysteme sind hierbei für zeitkritische Anwendungsfälle heranzuziehen [42]–[44].

2.4 Standards und Frameworks für Co-Simulationen

In der Folge werden verschiedene Anwendungsgebiete, Standards und Frameworks mit dem Ziel eingeführt, den Status Quo von Co-Simulationen zu beschreiben. Die späteren Ausführungen zu dem neuartigen Ansatz der Co-Simulation für die Stationsautomatisierung können so mithilfe der bereits veröffentlichten Ansätzen eingeordnet werden. Am Ende des Abschnitts findet sich eine Zusammenfassung der im Folgenden beschriebenen Aspekte in Tabelle 2.3.

2.4.1 Anwendungsgebiete von Co-Simulationen

Co-Simulationen spielen für ganz unterschiedliche Bereiche der Technik eine große Rolle und stehen daher schon seit einiger Zeit im Fokus der Untersuchung, um komplexe Systeme gesamtheitlich abzubilden. Eine Übertragbarkeit von Kopplungsansätzen auf verschiedene Problemstellungen unterschiedlicher Domänen rückt vermehrt in den Fokus. Hierfür ist ein umfassender Überblick über die unterschiedlichen Projekte mit der Zielstellung der Co-Simulation notwendig, um potenzielle Synergien zu entdecken, aber auch um gemeinsame offene Fragestellungen und Probleme zu identifizieren. Ein solcher Überblick wurde 2017 von Gomes *et al.* veröffentlicht [4] und betrifft die Bereiche Automotive, Energieproduktion und -verteilung, Klimatechnik, Embedded Systems, maritime Systeme und Robotik. Die Auswertung ergibt, dass die Mehrzahl der Projekte die Verschneidung von nur zwei Simulatoren behandelt, die aus unterschiedlichen Domänen stammen. Nach der reinen Definition der Co-Simulation ist die Anzahl der zu verschneidenden Systeme nicht begrenzt. Gomes zufolge wird geschlussfolgert, dass die Anpassungen, die erforderlich sind, um Simulatoren aus verschiedenen Formalismen und Domänen zu kombinieren, sehr schwierig auf jedes Co-Simulationsszenario zu verallgemeinern ist – selbst, wenn sie demselben Standard und derselben Domäne entsprechen. Eine der Hauptschlussfolgerungen aus [4] ist, dass es an Forschungsarbeiten zur modularen, stabilen und genauen Kopplung von Simulatoren für dynamische Strukturszenarien mangelt.

Besonders in dem Anwendungsfall der Energietechnik mit der Digitalisierung des Energiesystems stehen vermehrt auch Ansätze der Co-Simulation im Vordergrund. Allgemein verschmelzen innerhalb der Energieversorgung verschiedene Bereiche miteinander, was dazu führt, dass simulative Untersuchungen oft unterschiedliche Anforderungen abdecken müssen. Unter anderem spielen unterschiedliche physikalische Domänen eine Rolle, wie beispielsweise bei der Betrachtung von crosssektoralen Systemen. Aber vor allem die Fragestellung der Integration von Prozessen der Energieautomatisierung in Informations- und Kommunikationsnetzwerken stellt eine besondere Herausforderung dar. Hieraus resultieren diverse Ansätze der Co-Simulation, um eine Netzwerksimulation für Untersuchungen der Kommunikationsebene mit einer Stromnetzsimulation zu koppeln. Es bestehen bereits viele Vorarbeiten und unterschiedliche Ansätze der Co-Simulation, vor allem mit dem Anwendungsfokus der Energietechnik und Kopplung mit Informationstechnik, wie beispielhaft in [3], [4], [45], [46] erörtert und verglichen.

2.4.2 Co-Simulationsstandards

Für die Entwicklung von domänenübergreifenden Lösungen für die Herausforderung der Co-Simulation haben sich maßgeblich zwei Standards gebildet. Ziel dieser Standards ist die Vereinheitlichung von Schnittstellen, dem Datenaustausch und der Synchronisierung mehrerer Simulatoren. Für die zwei Simulationsarten der kontinuierlichen Simulation und der ereignisgesteuerten Simulation sind die Standards FMI und High Level Architecture (HLA) entstanden. Neben den Standards wird auch eine Ad-hoc-Kopplung als nicht-verallgemeinerbarer Ansatz der Kopplung zweier dedizierter Simulatoren verwendet. An dieser Stelle werden die Standards vorgestellt und das jeweilige Grundprinzip beschrieben.

2.4.2.1 FMI

Das FMI beschreibt einen anwendungs- und systemunabhängigen Standard für den Austausch von dynamischen Modellen mit dem Ziel der Errichtung einer Co-Simulation und wurde in der ersten Version bereits im Jahre 2010 veröffentlicht. Über die Zeit hinweg stand dieser Standard oft im Fokus verschiedener Anwendungen der Co-Simulation und wurde 2012 erweitert zur Version 2.0 [47]. Die Voraussetzung für FMI ist das Vorhandensein von zwei oder mehreren unabhängigen Simulatoren mit eigenen Solvern mit der Grundidee, diese Simulationen zu festen diskreten Zeitpunkten zu einem Datenaustausch zu führen. Zwischen den Zeitpunkten erfolgt die unabhän-

gige Berechnung der Subsysteme, die als Functional Mockup Unit (FMU) bezeichnet werden. Dabei wird eine Master-Slave-Beziehung der FMUs definiert. Der Datenaustausch wird über einen Koordinator umgesetzt und dieser bewerkstelligt ebenfalls die Synchronisierung. Generell besteht FMI aus zwei Teilen. Das Co-Simulations-Interface wird aus einem Satz C-Funktionen umgesetzt, die die Slave-Simulationen steuern lassen und den Datenaustausch der Ein- und Ausgänge der einzelnen Simulationssysteme umsetzen. Weiter wird ein Co-Simulations-Beschreibungsschema definiert, das im Extensible Markup Language (XML) Format eine standardisierte Beschreibung der statischen Informationen der einzelnen Systeme ermöglicht. Ein übergeordneter Koordinator wird über das FMI nicht direkt vorgegeben. Dieser ist stark anwendungsabhängig. Mithilfe der beschriebenen Teile des FMI ist der Koordinator anwendungsspezifisch zu entwickeln und setzt den Datenfluss sowie die Art der Kopplung um. Die Art der Kopplung kann als Jacobi- oder Gauss-Seidel-Verfahren ausgeprägt sein.

Der Standard hat inzwischen eine hohe Durchdringung in verschiedenen Anwendungen, wodurch im Dezember 2016 bereits 15 unterschiedliche Simulationstools diesen Standard unterstützen beziehungsweise in diesem implementiert sind [4]. Gomes *et al.* beschreiben steigende Herausforderungen bei der Kopplung von hybriden Systemen und vor allem für Ansätze nach diskret ereignisbasierten Systemen. Die standardisierte Schnittstelle über FMI leistet hierzu einen großen Beitrag, bestehende Offline-Systeme zu koppeln.

Offene Punkte in unterschiedlichen Projekten und Untersuchungen ist die Echtzeitfähigkeit unter Nutzung der FMI-Kopplung. Zumeist werden Tools mit Offline-Berechnung gekoppelt, da der Koordinator jeweils einen neuen Simulationsschritt in beiden unabhängigen Simulationssystemen anstoßen muss. Insgesamt liefert FMI einen Standard zur Beschreibung von vorrangig zeitkontinuierlichen Simulationssystemen und Ableitung einer vereinheitlichten Schnittstelle für eine Kopplung mit anderen FMUs.

2.4.2.2 HLA

Bereits im Jahr 2000 wurde erstmals der Standard IEEE 1516 zur HLA veröffentlicht, der eine Architektur zur integrierten und verteilten Simulation definiert [48]. Über eine zentrale Komponente, der sogenannten Run-Time-Infrastructure (RTI), wird der Datenaustausch und die Kommunikation von Einzelsimulationen realisiert. Innerhalb des HLA-Standards werden die Einzelsimulatoren als Föderaten bezeich-

net, die sich in der Gesamtheit als Föderation ausbilden. Zwischen den Föderaten wird eine Schnittstelle benötigt, die objektorientiert umzusetzen ist und verschiedene Objekte und Funktionen unterstützen, die allgemein beschreiben, wie Daten ausgetauscht werden, wie die einzelnen Föderaten koordiniert werden und wie eine zeitliche Synchronisierung umgesetzt wird. Innerhalb der HLA werden drei Definitionsbereiche festgelegt, welche die Schnittstellenspezifikation, Objekte des Datenaustauschs und sogenannte HLA-Regeln, also die Anforderungen an die Simulation, umfassen. Vor allem die Objekte des Datenaustauschs stehen im Vordergrund und können neben reinen Objekten auch Interaktionen, also Ereignisse mit bestimmt definierten Parametern, sein. Hierzu wird ein sogenanntes Object Model Template (OMT) genutzt, das von jeder Föderate genutzt wird, um einen standardisierten Datenaustausch zwischen zwei Simulatoren umzusetzen, wobei das jeweilige Datenmodell auf die Anwendung angepasst und erstellt werden kann. Innerhalb der HLA-Regeln werden dann die Anforderungen definiert, die ein Föderate erfüllen muss, um kompatibel zum OMT zu sein oder damit die Kommunikation umgesetzt werden kann.

Der HLA-Ansatz wird vorrangig für die Beschreibung und spätere Kopplung von ereignisbasierten Simulationssystemen nach Zeiglers Discrete Event System Specification (DEVS) Ansatz [49] verwendet und stellt hier analog zu FMI die Beschreibungsform und den speziellen Lösungsansatz dar. Eine Integration von zeitkontinuierlichen Systemen in die Beschreibungsform der DEVS wird unter anderem in [50] vorgenommen. Allgemein arbeitet das zeitkontinuierliche Simulationssystem autonom, bis er einen fest definierten, quantisierten Synchronisierungszeitpunkt erreicht hat, der als Ereignis nach DEVS definiert ist. An diesem Synchronisierungszeitpunkt werden die definierten Kopplungsvariablen unter den Simulationssystemen ausgetauscht.

2.4.2.3 Ad-hoc-Kopplung

Die Ad-hoc-Kopplung stellt an sich kein Standard dar, ist aber eine verbreitete Art der Co-Simulation ohne eigenes fest definiertes Framework oder einer standardisierten Beschreibung. Bei dem Ansatz wird davon ausgegangen, dass die Schnittstellen bereits in den einzelnen Simulatoren implementiert sind und genutzt werden. Dabei müssen potenziell umfangreiche Anpassungen in der Simulation an sich vorgenommen werden, um eine äußere Ansteuerung zu ermöglichen. Insgesamt kann der Ansatz der Ad-hoc-Kopplung effizient und kompakt sein, da keine weiteren Module ergänzt werden müssen. Voraussetzung für die zuvor eingeführten Co-Simulation-Standards sind definierte Schnittstellen für externe Ansteuerungen und Orchestrierung der Sys-

teme wie dem Koordinator. Viele proprietäre Simulationssysteme unterstützen diese Schnittstellen jedoch nicht, wodurch der Ansatz der Ad-hoc-Kopplung populär und weit verbreitet ist, da keine eigene Schnittstelle für die Co-Simulations-Koordination benötigt wird. Beispielhaft besteht bei vielen Simulationssystemen nicht die Möglichkeit, die Integration von Synchronisierung oder Datenformatanpassungen über externe Instanzen zu ermöglichen. Bei einer Ad-hoc-Kopplung müssen die genannten Anforderungen an Co-Simulation ohne standardisierte Vereinheitlichung anderweitig erfüllt werden, beispielsweise individuell von jedem verwendeten Simulator. Neben der Ermöglichung der gemeinsamen Synchronisierung und des Datenaustauschs stellt ein Koordinator vor allem eine Kontrollinstanz dar, um Informationen über die Synchronität, Stabilität oder weiteren Merkmalen während der Simulation zu erhalten [29], [45].

Folglich wird folgende Definition des Begriffs der Ad-hoc-Kopplung verwendet:

Definition 2.10 *Von einer **Ad-hoc-Kopplung** wird gesprochen, wenn kein bestehender Standards zur Kopplung von Simulatoren genutzt werden kann und eine Co-Simulation ohne die standardisierten Instanzen implementiert wird. Die Kopplung wird speziell für den Anwendungsfall ausgelegt und implementiert. Sie ist nicht verallgemeinerbar für andere Co-Simulationen.*

2.4.3 Frameworks und Anwendungen für Co-Simulationen

In der folgenden Übersicht werden bereits veröffentlichte Co-Simulationssysteme als generalisierte Frameworks oder als spezifische Anwendungen kurz eingeführt. Aus der Übersicht soll eine Grundlage zur Abgrenzung der später einzuführenden Co-Simulation für die Stationsautomatisierung geschaffen werden. Es handelt sich dabei nicht um einen allumfassenden Überblick, sondern um eine Auswahl, die bereits auf für die Arbeit relevante Ansätze eingegrenzt wurde. Neben den nachfolgend beschriebenen Frameworks liefert beispielsweise [51] einen aktuellen Überblick über weitere Umsetzungen von Co-Simulationen für die Anwendung im Strom- und Kommunikationsnetz.

2.4.3.1 MOSAIK

Ein Standard der Co-Simulation, der gleichzeitig als Framework dient, stellt MOSAIK vom OFFIS Oldenburg dar. MOSAIK hat das Ziel, eine vereinfachte Kopplung mehrerer Simulatoren im Kontext von Smart Grids herzustellen. Gegenstand

des Ansatzes ist die Kopplung von Stromnetzsimulationen und Kommunikations-Netzwerksimulationen als Basis für die Erstellung eines vereinheitlichten Standards der Co-Simulation. Der Ansatz ist im Vergleich zu FMI und HLA noch recht unbekannt. Innerhalb von MOSAIK wird von verschiedenen Simulatoren gesprochen, die analog zu den Föderaten aus HLA zu verstehen sind. Diese einzelnen Simulatoren, die zumeist domänenspezifisch ausgeprägt sind, haben die Aufgabe des Datenaustauschs und müssen hierzu synchronisiert sein. Zu diesem Zweck wird eine eigene Spezifikations-sprache, die Mosaik Specification Language (MoSL) eingeführt, die die Synthese unterschiedlicher Simulatoren und Komponenten herstellen soll [52]. MOSAIK wirkt als Middleware zwischen den Simulatoren und benötigt einen direkten Zugriff auf diese. Es werden vier Funktionen umgesetzt und implementiert, um den Simulator zu initialisieren, das Simulationsmodell zu parametrieren, Eingangsdaten zur Verfügung zu stellen und die Ausgangsdaten des Simulators abzufragen. Zudem wurde in [52] MOSAIK mit HLA verglichen. Beide Ansätze ähneln sich vom Grundprinzip sehr. Nur unter HLA sind die einzelnen Simulatoren beziehungsweise Föderaten autonomer und die Schnittstelle zur Synchronisierung und zum Datenaustausch ist umfangreicher gestaltet, was einen Vorteil bei großen und zeitkritischen Systemen darstellt, wohingegen der Aufwand als Negativpunkt von HLA dargestellt wird. Insgesamt soll MOSAIK als generalisiertes Framework verstanden werden, um die Anwendung von Co-Simulationen durch eine geführte Application Programming Interface (API) zu erleichtern [53], [54].

Die Anwendung von MOSAIK ist abgeleitet nur möglich, wenn die zu verwendenen Simulatoren den externen Zugriff von MOSAIK gewährleisten. Die Besonderheit, dass die Simulatoren über MoSL weitgehend initialisiert und parametrieren werden, stellt eine hohe Anforderung an die Simulatoren dar, sodass nur eine geringe Anzahl an verwendbaren Simulationssystemen nutzbar sind.

2.4.3.2 HELICS

Hierarchical Engine for Large-scale Infrastructure Co-Simulation (HELICS) stellt ein weit verbreitetes Framework, basierend auf dem HLA-Standard, dar und lässt die Integration verschiedener Simulatoren und Programmiersprachen zu [55]. Dabei werden Simulatoren als ausführbare Instanzen verstanden, die mehrere Föderaten instanzieren können, die über ein Publish-Subscriber-Mechanismus miteinander kommunizieren. Für die Kommunikation werden dedizierte, sogenannte Cores in den einzelnen Simulatoren vorgesehen. Ein zusätzlicher Broker verwaltet den Datenaustausch

zwischen den Förderaten und kann ebenfalls in die Kommunikation eingreifen. Orchestriert wird in diesem Fall die gesamte Co-Simulation von einer RTI, die an HLA angelehnt ist. In einem Vergleich zwischen den Anwendungen von MOSAIK und HELICS [56] zeigen die experimentellen Untersuchungen unter anderem das Zeitverhalten und die benötigte Zeit für die Simulationsschritte in Abhängigkeit der abgebildeten Strukturen.

HELICS ist ein Nachfolger des aus 2014 stammenden Frameworks FNCS, das konkret für die Verbindung der Stromnetz- und Kommunikationsebene definiert wurde [57]. FNCS erlaubt die Integration von drei Open-Source Simulatoren: a) für Betrachtungen im elektrischen Verteilnetz GridLAB-D, b) für Betrachtungen im Übertragungsnetz PowerFlow und c) für Betrachtungen der Kommunikationsebene den Simulator NS-3. HELICS stellt ein flexibleres Framework, mit der Unterstützung einer größeren Anzahl von Simulatoren im Vergleich zu FNCS dar, ist jedoch auf die unterstützten Simulationssysteme limitiert [58].

2.4.3.3 INSPIRE

Ein weiteres Framework für die Realisierung einer Co-Simulation im Bereich der Energieversorgung und Kommunikationsebene stellt INSPIRE [59] dar und nutzt HLA. Im Rahmen von INSPIRE werden die Simulatoren DIgSILENT PowerFactory und OPNET Modeler miteinander gekoppelt. Es können demnach die Ebenen der Stromnetze und Kommunikation gemeinschaftlich simuliert werden. Als Hauptschwerpunkt der Anwendung wird die Weitbereichskommunikation (Wide Area Network (WAN)) im Smart Grid definiert, wonach dynamische elektrische Größen in Form von aggregierten Mittelwerten (Root Mean Square (RMS)) verwendet werden. Der Fokus liegt in der gesamtheitlichen Simulation eines Stromnetzes mit Leitsystemankopplung, wobei auf eine Generalisierbarkeit auf verschiedene Anwendungen innerhalb des Smart Grids inklusive der stationsinternen Funktionen wie Steuerung und Regelung verwiesen wird. Über eine eigens definierte Stationsverwaltung kann, abgeleitet aus einer CIM-basierten Szenariobeschreibung, das Simulationsmodell extrahiert und instanziiert werden. Über sogenannte Substation Data Processing Units (SSDPU) werden die beiden Simulationsebenen einer Station zusammengefasst. Untersuchungen zu Schutz- und Regelungsfunktionen innerhalb des Stromnetzes sind in [60] mittels INSPIRE dargestellt. Das Framework lehnt sich an die Nutzung des IEC 61850 Datenmodells an, ohne dabei das Protokoll direkt als Übertragungsform zu nutzen. Die

Nachbildung der relevanten Kommunikationsstrukturen muss je nach Anwendungsfall innerhalb der Netzwerksimulation integriert werden.

2.4.3.4 Ad-hoc Co-Simulationen in der Energietechnik

Neben den bisher vorgestellten Frameworks, die für die Umsetzung der angestrebten Co-Simulation verwendbar wären, stehen darüber hinaus in der Literatur weitere spezialisierte Ansätze unter der Nutzung von Ad-hoc-Kopplungen zur Verfügung. Eine entsprechende Übersicht mit den verschiedenen Umsetzungen inklusive der verwendeten Simulatoren wurde von den Autoren von [51] auf Basis von [61] mit dem Fokus der Emulation von Cyber-Angriffen innerhalb einer gekoppelten Strom- und Kommunikationsnetz-Co-Simulation veröffentlicht. Die herausgearbeiteten Ansätze beschreiben grundsätzlich eine Integration von realen Automatisierungskomponenten, beziehungsweise auch von Intelligent Electronic Device (IED)s in Form eines cyber-physischen Testbeds. Die größten Schwächen und Herausforderungen der untersuchten Systeme werden nach [61] in der Echtzeitfähigkeit der verwendeten Simulationen gesehen. Besonders die Abbildung einer reproduzierbaren, detaillierten Kommunikationsebene wird als Nadelöhr in den verschiedenen Systemen und Testbeds gesehen. Außerdem wird ein verstärktes Aufkommen der Integration von SDN zur Umsetzung von Netzwerkfunktionen innerhalb eines Simulators erwähnt.

2.4.4 Übersicht der relevanten Standards und Frameworks

In Tabelle 2.3 sind die als relevant identifizierten und beschriebenen Ansätze zur Kopplung von Simulatoren aufgeführt. Zudem erfolgt eine Bewertung der Echtzeitfähigkeit und Nachbildbarkeit der wichtigen Prozesse und Protokolle.

2.5 Schlussfolgerung

Der Überblick über Co-Simulationen für elektrische Netze und Kommunikationsnetze zeigt, dass ein erheblicher Aufwand notwendig ist, um verschiedene Simulationssysteme zu koppeln und eine ganzheitliche Betrachtung zu ermöglichen. Die Herausforderung in der Stationsautomatisierung liegt in der Notwendigkeit hybrider, echtzeitfähiger Co-Simulationsansätze. Die Kopplung der zeitkontinuierlichen Stromnetzsimulation mit der ereignisbasierten Netzwerksimulation deckt unterschiedliche Fragestellungen und Forschungsschwerpunkte auf.

Tabelle 2.3: Übersicht über Standards und Frameworks für Co-Simulationen

Art	Name	Beschreibung
Standards	FMI [47]	<ul style="list-style-type: none"> • Kein übergeordneter Koordinator, anwendungsabhängig • Master-Slave-Beziehung zwischen den Simulatoren • Datenaustausch zu festen, diskreten Zeitpunkten • Standardisiertes Beschreibungsschema
	HLA [48]	<ul style="list-style-type: none"> • Zentrale Komponente für Datenaustausch • Objektorientierte Schnittstelle • Entwickelt für ereignisbasierte Systeme
	Ad-Hoc	<ul style="list-style-type: none"> • Kein Standard • Schnittstelle bereits implementiert • Koordinator benötigt Zugriff auf Simulatoren
Frameworks	MOSAIK [52]	<ul style="list-style-type: none"> • Angelehnt an HLA • Kopplung mehrerer Systeme im Kontext von Smart Grids • Eigene Spezifikationssprache • Bedarf an direktem Zugriff auf Simulatoren
	HELICS [55]	<ul style="list-style-type: none"> • Basiert auf HLA • Publish-Subscriber-Mechanismus • Broker verwaltet Datenaustausch
	INSPIRE [59]	<ul style="list-style-type: none"> • Basiert auf HLA • Kopplung von zwei expliziten Systemen • Zusammenfassung über definierte Berechnungsinstanz

Besonders wichtig ist die Koordination hinsichtlich Synchronität, Stabilität und Echtzeit-Bedingungen. Im Bereich der Ad-hoc-Co-Simulationen zeigt sich die große Bandbreite möglicher Kombinationen von Simulatoren und Untersuchungsgegenständen. Eine Übertragbarkeit der veröffentlichten Ergebnisse ist oft eingeschränkt, da sie

an spezielle Anwendungsfälle gebunden sind. Diese Diversität führt zu Bestrebungen nach Vereinheitlichung durch Standards und Frameworks, um verschiedene Anwendungsfälle und Simulatoren zu integrieren. Drei relevante Frameworks und zwei wichtige Standards wurden vorgestellt und analysiert.

Gemeinsam haben die unterschiedlichen Umsetzungen, dass die Echtzeitfähigkeit und die korrekte Integration der Rahmenbedingungen der Stationsautomatisierung kritische Punkte darstellen. Aktuelle Veröffentlichungen zeigen zunehmend Ansätze, die Netzwerkumgebungen emulieren, anstatt zu simulieren, wodurch einige Problemstellungen der verschiedenen Simulationstypen entfallen. Die Umsetzung relevanter Energie-Übertragungsprotokolle wie IEC 61850 wird hingegen selten und oberflächlich behandelt. Es besteht eine Lücke in der Literatur bezüglich dynamischer, echtzeitfähiger Co-Simulationssysteme für die IEC 61850-basierte Stationsautomatisierung. Die Nutzung der Frameworks ist auf die unterstützten Simulatoren limitiert, da die definierten Koordinatoren umfangreichen Zugriff auf diese benötigen.

Der Überblick über den Stand der Technik bekräftigt die Relevanz der in Abschnitt 1.3 eingeführten Forschungsfragen, insbesondere hinsichtlich der Abbildungsart der Kommunikationsebene, die zwischen Simulation und Emulation variiert. Die Eigenschaften der Stationsautomatisierung sind in den Implementierungen oft unzureichend berücksichtigt, um die Automatisierungstechnik von Umspannwerken sowie relevante Übertragungsprotokolle und Netzwerkeigenschaften abzudecken. Die Fragestellung der Synchronität und Echtzeitfähigkeit muss für jeden Ansatz separat bewertet werden.

3 Anwendungsfall Digitale Station

Der in der Problemstellung und Motivation diskutierte Anwendungsfall der digitalisierten Umspannwerke und deren Prozesse der Stationsautomatisierung werden in diesem Kapitel eingeführt und für die Untersuchung mithilfe der Co-Simulation erörtert. Hierfür wird zunächst der Begriff der Digitalen Station erläutert und deren Eigenschaften beschrieben. Anschließend werden die essenziellen Anforderungen an die Kommunikation innerhalb der Digitalen Stationen herausgearbeitet und die Anforderungen für den Entwurf der Co-Simulation daraus zusammenfassend abgeleitet.

Das Untersuchungsobjekt ist die Stationsautomatisierung von durchgehend digitalisierten Stationen mit der Nutzung des Kommunikationsstandards IEC 61850. Im Rahmen der vorliegenden Arbeit ist somit der Begriff der Digitalen Station folgendermaßen definiert:

Definition 3.1 *Umspannstationen oder auch Umspannwerke, die hinsichtlich der prozessrelevanten Kommunikation vollständig digitalisiert sind und sich vollständig am IEC 61850-Standard orientieren, werden als **Digitale Stationen** bezeichnet.*

3.1 Grundaufbau eines Umspannwerkes

Das Umspannwerk spielt eine essenzielle Rolle im Stromversorgungssystem und bündelt Prozesse der Spannungsumformung sowie die Integration von Schutz- und Regelungssystemen. Die Umformung des Spannungsniveaus im elektrischen Netz erfolgt auf verschiedenen Ebenen, beispielsweise zwischen Hoch- und Mittelspannung. Umspannwerke sind unterschiedlich komplex, abhängig von den Spannungsebenen und der Bedeutung der Spannungsumformung. Umspannwerke zwischen höheren Netzebenen sind in der Regel komplexer und enthalten mehr Schutzsysteme als die Umspannung von Mittel- auf Niederspannung.

Bei der verbauten Prozesstechnik unterscheidet man in die Primär-, Sekundär- und Netzwerktechnik. Dabei sind Systeme der Primär- und Sekundärtechnik bezogen auf die elektrische Funktion des Umspannwerkes, die Kommunikationstechnik hingegen fasst Geräte zusammen, die für das Kommunikationsnetzwerk verantwortlich sind.

Als Primärtechnik werden Systeme und Komponenten verstanden, die direkt am elektrischen Verhalten beteiligt und spannungsführend sind. Hierzu zählen vor allem Leitungen, Schalter, Stufensteller, Sammelschienen und Transformatoren.

Neben den beschriebenen Komponenten der Primärtechnik ist die Sekundärtechnik für ein Umspannwerk essenziell. Hierunter fallen Komponenten und Geräte, die nur indirekten Einfluss auf das elektrische Verhalten haben, beispielsweise Schutzgeräte [62]. Nach der Terminologie von IEC 61850 gilt die später eingeführte Definition 3.2 der IEDs für die Sekundärsysteme. Das Schutzgerät steuert die Primärtechnik, also den Leistungsschalter an, um einen Eingriff im elektrischen Netz vorzunehmen. Das Schutzgerät bewirkt somit Änderungen im elektrischen Verhalten, jedoch wirkt die Primärtechnik in Form des Leistungsschalters als Akteur auf der elektrischen Leitung und unterbricht diese.

Die Kommunikationstechnik fasst sämtliche Systeme zusammen, die für den Aufbau eines Kommunikationsnetzwerkes im Umspannwerk zuständig sind. Hierunter zählen Netzwerkschalter, Firewalls, Router oder Netzkabel. Die Komponenten haben keinen direkten oder indirekten Einfluss auf das elektrische Verhalten im Umspannwerk, können aber essenziell für die Funktion des Datenaustauschs von Sekundärtechnik sein. Es entsteht eine Abhängigkeit der Sekundär- von der Kommunikationstechnik.

3.2 Beschreibung IEC 61850 basierte Station

Es wird ein Ansatz eines digitalisierten Umspannwerks zwischen der Hoch- und Mittelspannungsebene mit einer einheitlichen Kommunikation nach IEC 61850 vorgestellt. Das Ziel ist die realitätsnahe Nachbildung des Aufbaus, einschließlich der Abbildung der notwendigen Stationskomponenten und Kommunikationsstrecken. Der prinzipielle Aufbau ist in Abbildung 3.1 dargestellt. Gezeigt wird ein Ausschnitt eines Umspannwerks aus einem Verteilnetz, dessen kommunikative Anbindung über Merging Unit (MU)-Komponenten und nicht näher spezifizierte Input-Output-Geräte (I/O) an den Prozessbus erfolgt. Die IEDs sind direkt an den Prozessbus angebunden und werden über den Stationsbus den Stations- und Netzleitsystemen zugänglich gemacht [5].

Die schematische Darstellung untersetzt den Begriff der Digitalen Station im Nachfolgenden und kann für ein beispielhaftes Umspannwerk oder auch für einen Abgang eines solchen stehen. In der realen Implementierung eines digitalen Umspannwerkes wird in der Regel ein eigener Prozessbus und zumeist ein eigener Stationsbus für jeden Mittelspannungsabgang aufgebaut. Auch einzelne Hochspannungszugänge können mit separaten Kommunikationsstrukturen aufgebaut werden. Der Standard IEC 61850 versucht, einen gemeinsamen Rahmen für unterschiedlichste Größen und Kon-

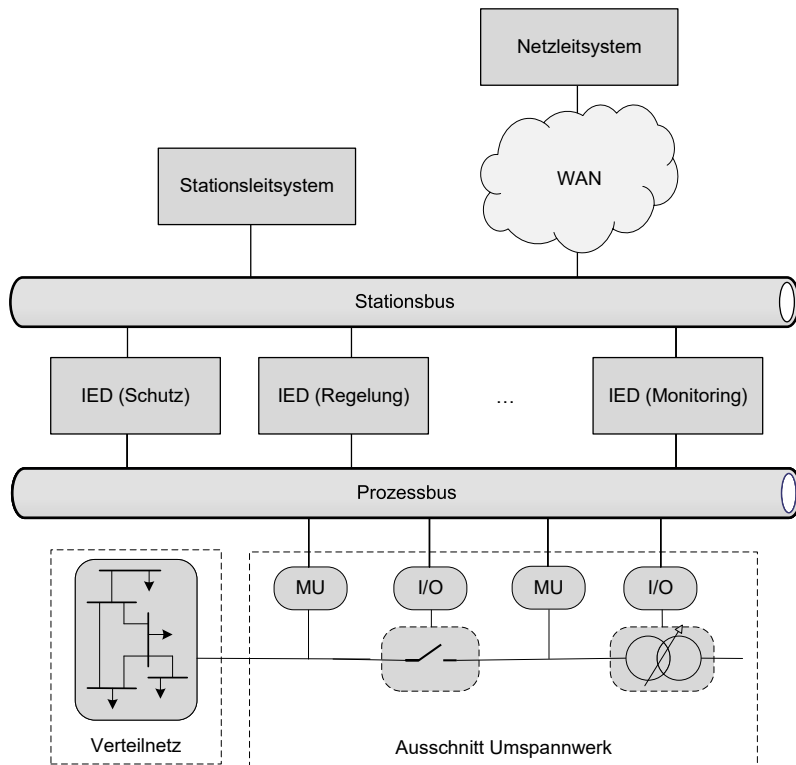


Abbildung 3.1: Struktur einer Digitalen Station nach dem Standard IEC 61850

figurationen von Umspannwerken zu definieren. Die exakte Umsetzung kann in der Realität von der dargestellten Referenzarchitektur aus Abbildung 3.1 abweichen, jedoch fordert der Standard mindestens die Separation der Kommunikationsbusse und die einheitlichen Kommunikationsschnittstellen der Netzwerkteilnehmer. Die weiteren Betrachtungen stützen sich auf den schematischen Charakter der abgebildeten Architektur. Genaue Details der verschiedenen Umsetzungen und abgeleiteten Anforderungen für den Entwurf der Co-Simulation werden nachfolgend näher erläutert.

3.2.1 Grundlagen IEC 61850

Der Standard IEC 61850, der seit 2004 herausgegeben wird, umfasst die Einführung einer Kommunikationsarchitektur für Schaltanlagen und die Definition eines Übertragungsprotokolls mit unterlagertem Datenmodell. Das Datenmodell beschreibt, welche Informationen von den einzelnen Instanzen zur Verfügung gestellt werden und für die Kommunikation genutzt werden können. Die Kommunikationsarchitektur mit der Trennung von Prozess- und Stationsbus ist in Abbildung 3.1 dargestellt. In den folgenden Abschnitten werden die für die vorliegende Arbeit relevanten Aspekte behandelt, die für eine simulative Beschreibung beachtet werden müssen. Aspekte, die nicht im Fokus stehen, werden nicht näher eingeführt, da kein vollständiger Überblick über alle Teile des Standards angestrebt wird. Einen umfassenden Überblick bieten beispielsweise die Autoren von [32].

3.2.2 Aufbau und Teile des Standards

Die IEC 61850 Reihe ist in unterschiedliche Teile untergliedert, die als Überblick über die einzelnen Normteile und deren Titel in Tabelle 3.1 dargestellt sind [32]. Die Norm umfasst neben den grundlegenden Begriffsdefinitionen und generellen Anforderungen die Einführung und Betrachtung unterschiedlicher Konzepte. Die Kommunikationsstruktur aus IEC 61850-7 bildet die Grundlage für die vorgestellte Architektur in Abbildung 3.1 unter Nutzung des einheitlichen Datenmodells aus IEC 61850-7 und der Kommunikationsprotokolle in IEC 61850-8 und IEC 61850-9.

3.2.3 Definition IED

Essenziell für die vorliegende Arbeit ist der im Standard IEC 61850-2 definierte Begriff IED, der sämtliche Sekundärsysteme bündelt, die Daten entgegennehmen, verarbeiten und weiter versenden. Die entsprechende Definition nach IEC 61850-2 lautet:

Definition 3.2 *Ein IED (Intelligent Electronic Device) ist ein Gerät mit einem oder mehreren Prozessoren und der Fähigkeit, Anwendungsfunktionen auszuführen, Daten lokal in einem Speicher zu speichern und Daten mit anderen IEDs (Quellen oder Senken) über eine digitale Verbindung auszutauschen.*

Demnach werden im Rahmen der Arbeit, wie in Abbildung 3.1 eingeführt, sämtliche Sekundärgeräte als IEDs bezeichnet. Dabei ist ein Unterschied zur Primärtechnik weiterhin gegeben. Im Anwendungsfall des Umspannwerks betrifft die Kategorie der

Tabelle 3.1: IEC 61850 Normteile nach [32]

Teil	Titel
1	Introduction
2	Glossary of term
3	General Requirements
4	System and Project Management
5	Communication Requirements for Functions and Device Models
6	Configuration Description Language for Communication in Electrical Substations related to IEDs
7	Basic Communication Structure for Substations and Feeder Equipment
7.1	Principles and Models
7.2	Abstract Communication Service Interface (ACSI)
7.3	Common Data Classes
7.4	Compatible logical node classes and data classes
8	Specific Communication Service Mapping (SCSM)
8.1	Mappings to MMS (ISO/IEC 9506 - Part 1 and Part 2) and to ISO/IEC 8802-3
9	Specific Communication Service Mapping (SCSM)
9.1	Sampled Values over Serial Unidirectional Multidrop Point-to-Point Link
9.2	Sampled Values over ISO/IEC 8802-3
10	Conformance Testing

Primärtechnik vor allem die Messinstrumente (MU) und Schaltelemente (Leistungs- oder Trennschalter). In einigen Definitionen werden diese Komponenten auch als IED bezeichnet [63]. In der vorliegenden Arbeit werden sie aber getrennt von den IEDs betrachtet, da sie als Prozesskomponenten verstanden werden und demnach nicht in die Definition der IEDs integriert werden. Auf die detaillierte Beschreibung der Komponenten des zu modellierenden Umspannwerkes wird in Kapitel 5.1.1 eingegangen.

3.2.4 Datenmodell

Der Standard IEC 61850 beinhaltet als Kernelement ein einheitliches Datenmodell, um eine herstellerunabhängige Beschreibung und Kommunikation zu gewährleisten. Hierzu wird jedes IED in einem Schichtenmodell, bestehend aus den Schichten der Logischen Devices, Logischen Knoten, Objekten und Attributen, beschrieben. Sämtliche Informationen der Geräte werden strukturiert den Funktionen und Geräten zugewiesen. Die einheitliche Datenmodellierung ermöglicht eine herstellerübergreifende Kommunikation und definiert den Rahmen für die verwendbaren Informationen. Die

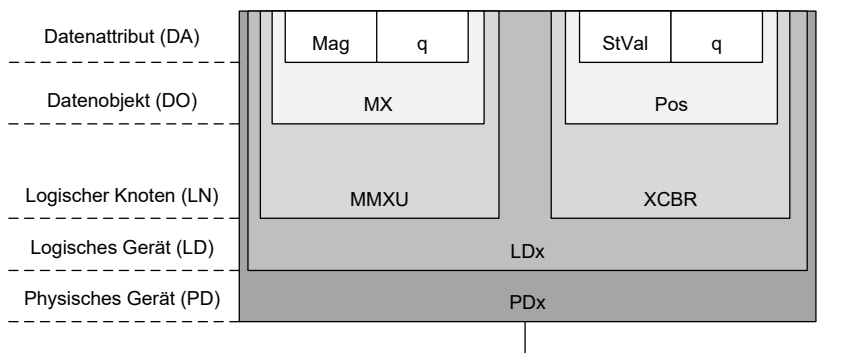


Abbildung 3.2: Struktur des Datenmodells nach beispielsweise [64]

Struktur eines IED mit zwei beispielhaften Logischen Knoten ist in Abbildung 3.2 angelehnt an [64] eingeführt.

Neben der Verallgemeinerung des Datenmodells wird dieses mittels einer eigenen Beschreibungssprache (Substation Configuration Language (SCL)) in unterschiedlichen Dateien entlang des Engineering-Prozesses dokumentiert. Für SCL wird eine XML-basierte Formatierung genutzt und beschreibt durchgängig die verwendeten Daten aller Komponenten innerhalb einer Station. Durch die Beschreibungsdateien wird eine Interoperabilität zwischen verschiedenen Herstellern hergestellt. Als mögliche Dateiformate stehen beispielsweise die Substation Configuration Description (SCD), IED Capability Description (ICD), Instantiated IED Description (IID) oder Configured IED Description (CID) zur Verfügung.

3.2.5 Kommunikationsarten

Die Kommunikation zwischen den IEDs in der definierten Kommunikationsstruktur wird in den Normen IEC 61850-8-1 und 61850-9 näher beschrieben und ordnet sich in verschiedene Kategorien. Dabei wird unterschieden in die Prozess- und Stationsbusprotokolle, analog zur Netzwerktrennung in der Strukturbeschreibung aus Abbildung 3.1. Es werden unterschiedliche Kommunikationsprinzipien genutzt. Zwei Kommunikationsarten lassen sich dabei grundsätzlich voneinander abgrenzen, die sich in drei Protokollanwendungen niederschlagen. Es wird unterteilt in feste Client-Server-Kommunikationen und in Multicast-basierte Publisher-Subscriber-Konstrukte. Die Arten der Kommunikation und ihre nach IEC 61850 definierten Protokolle werden nachfolgend vorgestellt.

3.2.5.1 Client-Server-Kommunikation

Das Client-Server-Prinzip stellt ein klassisches Kommunikationsverfahren dar, das auf einer Kommunikationsinitialisierung zwischen zwei Kommunikationspartnern basiert. Dabei wird die Verbindung zwischen den Partnern initial durch einen Handshake-Mechanismus etabliert. Nachfolgend können verschiedene Daten zwischen den zwei Teilnehmern ausgetauscht oder auch abgefragt werden [65]. In der Norm IEC 61850 ist hierfür der sogenannte Manufacturing Message Specification (MMS)-Service im Teil IEC 61850-8-1 festgelegt. Der MMS-Service kann auf das gesamte Datenmodell des definierten Servers zurückgreifen und über dedizierte Funktionen kommunizieren. Dabei erfolgt zu Beginn jeder Kommunikation ein sogenannter „Initialization-Request“ des Clients, der sich mit dem Server verbinden möchte. Der Server antwortet, wenn der Client im Datenmodell als valider Client hinterlegt und nicht explizit ausgeschlossen ist, mit einer positiven „Initiate-Response“. Dieser Mechanismus ist mit einem klassischen Client-Server-Handshake vergleichbar. MMS setzt auf eine Transmission Control Protocol (TCP)-Verbindung auf, die zusätzlich im Vorhinein zu etablieren ist. Auf die ausführliche Beschreibung der überlagerten Schichten nach dem Open Systems Interconnection (OSI)-Modells wird hier verzichtet und lediglich die MMS-Anwendungsschicht als wichtige Ausprägung der IEC 61850 Kommunikation erläutert.

Nach der Initialisierungsphase werden Daten vor allem durch das Anfrageprinzip ausgetauscht. Hiernach startet der Client jeweils mit Anfragen an den verbundenen Server. Dies kann beispielsweise eine Lese- oder Schreibe-Anfrage für definierte Datenobjekte sein. Hier spricht man in der Regel auch von dem Polling-Prinzip. Neben den

Polling-basierten Funktionen steht vor allem der Information-Report nach MMS für die geregelte Kommunikation im Vordergrund. Der Information-Report stellt dabei ein zyklisches Versenden von definierten Datenobjekten dar, sodass kein wiederholtes Abfragen notwendig ist. Nach Aktivierung eines solchen Information-Reports werden Daten zyklisch vom Server an den Client gesendet als ein sogenannter Unconfirmed-Service, der keine Anfrage- und Antwort-Abfolge darstellt. Der MMS-Service wird vor allem auf der Stationsebene verwendet. Er unterliegt keinen signifikanten Zeitrestriktionen, wie in der später eingeführten Tabelle 3.3 dargestellt ist [32].

3.2.5.2 Publish-Subscriber

Im Gegensatz zum Client-Server-Prinzip wird beim Publish-Subscriber, nachfolgend Publisher-Prinzip genannt, keine feste Kommunikationsverbindung zwischen zwei Teilnehmern aufgebaut. Dieses Prinzip basiert auf einem Sender, der seine Nachrichten unterschiedlichen Kommunikationsteilnehmern zur Verfügung stellt und die Empfänger die öffentlichen Nachrichten abonnieren (englisch subscribe). Das Prinzip setzt zumeist auf das Multicast-Prinzip auf, das die Nachrichten allen angeschlossenen Teilnehmern verfügbar macht. Es wird kein spezifizierter Empfänger im Netzwerkpaket definiert. Hierfür wird ein Layer-2-Multicast-Mechanismus verwendet. Das Datenpaket enthält zwar ein Datenfeld für den Empfänger, jedoch wird hier ein definierter Adressbereich für Multicast-Anwendungen hinterlegt, der von den Empfängern zur Erkennung der Netzwerkpakete verwendet werden kann. Der Empfänger in diesem Konstrukt lauscht folglich im Netzwerk auf Nachrichten, die er abonniert hat und verarbeitet diese nach ihrem Eintreffen. Durch dieses Prinzip ist es möglich, dass mehrere Empfänger die gleichen Nachrichten eines einzelnen Senders abonnieren und je nach Aufgabenstellung und Funktion verarbeiten. Eine grundlegende Anwendung für solche Publish-Nachrichten ist das zyklische Versenden von Status- oder Messdaten, die von unterschiedlichen Empfängern verwendet werden.

Innerhalb des IEC 61850 Standards werden zwei unterschiedliche Kommunikationsservices definiert, die auf dem Publisher-Prinzip aufsetzen. Zum einen wird in der Norm IEC 61850-8-1 der Generic Object Oriented Substation Events (GOOSE)-Service beschrieben, der unter Nutzung des Datenmodells das zyklische Versenden von Status- und Steuergößen für feste Datenobjekte erlaubt. Zum anderen wird in IEC 61850-9-2 der SV-Service eingeführt, der das zyklische Versenden von Messdaten in einer fest definierten Datenstruktur beschreibt. Für die praktische Umsetzung hat sich hierbei die spezifizierte Leitlinie IEC 61850-9-2LE als industrieweit etablierte Varian-

te durchgesetzt. Sie definiert unter anderem eine feste Abtastrate von 80 Samples pro Periode bei einem 50-Hz-System sowie einen standardisierten Datenumfang. Die beiden Publisher-Services sind im Vergleich zu den Client-Server-Services zeitkritischer und ein Publisher-Subscriber-Nachrichten-Austausch erfolgt in höherer Frequenz, da kein Anfrage-Antwort-Mechanismus genutzt werden muss. Die Publisher-Services finden demnach vor allem auf der Prozessbusebene Anwendung und stehen maßgeblich für die Verbindung von Kommunikationskomponenten, die prozessnah sind und Mess- und Steuerinformationen austauschen. Demgegenüber kann der GOOSE-Service als zusätzlicher Service auf dem Stationsbus zur Verbindung zwischen zwei IEDs genutzt werden, wobei der Fokus auf den Anwendungen des Prozessbusses liegt.

3.2.6 Kommunikationsstruktur

Die beiden beschriebenen Kommunikationsarten bilden eine Anforderung für den unterlagerten Netzwerkaufbau inklusive der IEDs. Dabei werden die essenziellen Geräte als IEDs zusammengefasst und setzen die Automatisierungs- und Kommunikationsfunktionen in einer Digitalen Station um. Abhängig von der Funktion werden die Geräte in dem Netzwerk platziert. Wie bereits dargelegt, unterscheidet der Standard IEC 61850 beim Netzwerkaufbau die zwei lokale Bussysteme (Prozess- und Stationsbus). Dabei liegt die Unterscheidung darin, dass die IEDs des Prozessbusses direkt die Primärtechnik ansteuern können. Folglich werden die klassischen Automatisierungsgeräte, die Messwerte aufnehmen, verarbeiten und Steuerbefehle abgeben, mit dem Prozessbus verbunden. Sämtliche IEDs, die zur Überwachung oder zu anderen Zwecken dienen, die nicht direkt auf die Primärtechnik des unterlagerten Stromnetzes zugreifen, werden miteinander über den Stationsbus verbunden. Innerhalb der Station werden die beiden Kommunikationsbusse, die in der Regel durch Netzwerkwitches implementiert sind, möglichst strikt voneinander getrennt. Ein IED kann auf beide Bussysteme zugreifen und stellt somit eine implizite Verbindung der beiden Kommunikationsbusse dar. Für den Prozessbus wird aufgrund der genutzten Layer-2-Multicast-Kommunikation ein Layer-2-Switch benötigt. Der Standard IEC 61850 lässt es dem Anwender offen, in welcher Netzwerktopologie die beiden Bussysteme aufgebaut werden und fordert nur eine Verbindung der relevanten Komponenten.

3.3 Kommunikationsanforderungen nach IEC 61850-5

Für die auf IEC 61850 basierte Digitale Station sind insbesondere die Kommunikationsinfrastruktur, die Übertragungsprotokolle und die verbindlichen Vorgaben zum Datenaustausch aus IEC 61850-5 von Bedeutung. Dieser Normteil beschreibt die Anforderungen für die Performanz und Kommunikationsarten sowie die verschiedenen Logischen Knoten und deren Spezifikationen. Das Hauptziel von Teil fünf ist die Sicherstellung der Interoperabilität der Komponenten unterschiedlicher Hersteller durch ein einheitliches Datenmodell. Anschließend werden die relevanten Aspekte des Netzwerkverhaltens für die zu formulierende Co-Simulation kurz eingeführt.

Grundlegend werden in IEC 61850-5 folgende Anforderungen an die Kommunikation näher definiert:

1. Generelles,
2. Zeit und Synchronisierung,
3. Übertragungszeit,
4. Übertragungszeit für Kommunikation zwischen Stationen,
5. Übertragung von analogen Signalen,
6. Verbindungen zu Betreibern.

Daraus ergeben sich für die Co-Simulation einer Digitalen Station die Anforderungen zwei und drei als die relevantesten und werden näher beleuchtet.

3.3.1 Synchronisierung

In Teil fünf der IEC 61850 wird die Zeitbasis in der Art beschrieben, dass sie zwischen den verschiedenen Geräten und IEDs innerhalb einer Station mit einer angegebenen Genauigkeit voneinander abweichen darf. Dementsprechend ist eine Voraussetzung, dass alle Uhren der IEDs für eine kohärente Zeitzone mit angemessener Genauigkeit synchronisiert sein müssen. Neben den Uhren der einzelnen Endgeräte wird weiter eine Anforderung beschrieben, dass die Daten, die zwischen den Geräten für verschiedene Anwendungen ausgetauscht werden, ebenfalls durch synchronisierte Abtastungen zeitlich kohärent sein müssen.

Folglich wird eine Genauigkeit definiert, welche als die Abweichung der jeweiligen Uhren der Geräte in Bezug auf eine Referenz-Uhr, in der Regel einer Global Positioning System (GPS) Uhr, beschreibt:

$$\Delta t_{acc} = |t_{master} - t_{slave}| \quad (3.1)$$

Die Genauigkeit Δt_{acc} wird für verschiedene Anwendungen wiederum, abhängig von der zeitlichen Kritikalität (hier Synchronisierungs-Klassen genannt) unterschiedlich definiert. Dabei beläuft sich die maximale Zeitdifferenz zwischen zwei IEDs, die jeweils zum Master mit der gleichen Genauigkeit synchronisiert sind, zu $2 \Delta t_{acc}$. Die Zeitsynchronisierungs-Klassen nach IEC 61850-5 sind in Tabelle 3.2 dargestellt.

Tabelle 3.2: Synchronisierungs-Klassen nach IEC 61850-5

Synchronisierungs-Klasse	Genauigkeit Δt_{acc} in [μ s]	Anwendung
TL	> 10000	Geringe Genauigkeit der Synchronisierung – Verschiedenes
T0	10000	Zeitstempel von Ereignissen mit einer Genauigkeit von 10 ms
T1	1000	Zeitstempel von Ereignissen mit einer Genauigkeit von 1 ms
T2	100	Zeitstempel von Nulldurchgängen und von Daten für den verteilten Synchronisationscheck. Zeitstempel zur Unterstützung der Punkt-auf-Welle-Umschaltung
T3	25	Verschiedenes
T4	4	Zeitstempel von Abtastungen bzw. synchronisierte Abtastungen
T5	1	Hochpräzise Zeitmarkierung der Abtastungen bzw. hochsynchronisierte Abtastungen

3.3.2 Übertragungszeit

Der Standard IEC 61850 definiert verschiedene Anforderungen an die Kommunikationsarten zur Sicherstellung einer zeitnahen Übertragung prozessrelevanter Daten wie Messgrößen oder Steuersignale. Nach IEC 61850-5 ist die Übertragungszeit definiert, die sich aus der Übertragung eines Netzwerkpakets zwischen zwei physischen Geräten (Physical Device (PD)) ergibt. Die Übertragungszeit aus dem Standard ist in Abbildung 3.3 dargestellt.

Die Übertragungszeit umfasst demnach die Protokollstapel der beiden Geräte und die Verbindung zwischen ihnen. Für das verwendete Netzwerk bedeutet dies, dass alle zeitlichen Einflüsse, wie beispielsweise Verzögerungen, in der Übertragungskomponente t_b enthalten sind. Unterschiede innerhalb der Netzwerkarchitektur wirken sich auf diese Komponente der Übertragungszeit aus.

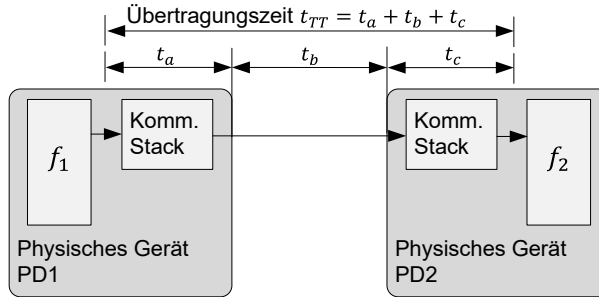


Abbildung 3.3: Definition der Übertragungszeit nach IEC 61850-5

Tabelle 3.3: Übertragungszeitklassen nach IEC 61850-5

Übertragungszeitklasse	Übertragungszeit [ms]	Anwendungsbeispiel: Übertragung von
TT0	> 1000	Dateien, Events, Log-Nachrichten
TT1	1000	Events, Alarme
TT2	500	Steuerbefehle
TT3	100	Langsame automatische Interaktionen
TT4	20	Schnelle automatische Interaktionen
TT5	10	Freischaltungen, Statusänderungen
TT6	3	Schalthandlungen, Sperr-Befehle

Innerhalb der IEC 61850-5 sind für verschiedene Nachrichtentypen, Schnittstellen und Anwendungsfälle unterschiedliche Leistungsklassen definiert. Für reale Anwendungen bedeutet das, dass verschiedene Teile der Netzwerkinfrastruktur unterschiedlichen Anforderungen genügen müssen. Die höchste Zeitanforderung, die nach IEC 61850-5 definiert wird, betreffen Schalthandlungen und Sperr-Befehle innerhalb von Stationen, sie entsprechen der Übertragungszeitklasse TT6, wie es in Tabelle 3.3 dargestellt ist.

3.4 Zusammenfassung der Modellierungsanforderungen Digitaler Stationen

Das Ziel dieser Arbeit ist die Abbildung einer Digitalen Station unter Berücksichtigung der Anforderungen aus IEC 61850. Die Co-Simulation soll das zu beschreibende Energie- und Kommunikationssystem jeweils separat abbilden und übergeordnet zusammenführen. Somit werden die Berechnungen und Simulationen der elektrischen Größen getrennt von den Funktionen der Kommunikationsgeräte betrachtet. Zugleich muss die Co-Simulation für die Kombination der zwei Betrachtungen, analog zum realen Aufbau einer Digitalen Station die formulierten Anforderungen nach IEC 61850 erfüllen. Der Co-Simulations-Ansatz basiert folglich auf den Prinzipien der IEC 61850 Norm und setzt eine Instanziierung von Kommunikationsgeräten mitsamt der verschiedenen Arten der Datenübertragung um. Demnach muss die Kopplung der beiden Ebenen, bei der ein Datenaustausch auf Basis von Netzwerkkommunikation elementar ist, die Einhaltung der maßgeblichen Anforderungen und Restriktionen gewährleisten. Grundvoraussetzung ist die Möglichkeit, die Kommunikationsstruktur nach IEC 61850 aufbauen zu können, in der das Netzwerk in den Stations- und Prozessbus getrennt ist. Die Kopplung der Simulatoren erfolgt innerhalb der Co-Simulation auf dem Prozessbus. Dabei wird definiert, dass auch die höchsten Anforderungen, die sich aus den Übertragungszeit- und Synchronisierungsklassen ergeben, eingehalten werden. Besonders die Übertragungszeit wird maßgeblich im Vordergrund stehen. Sie begründet sich durch die gemeinsame Abbildung des Prozessbusses, wo die Mess- und Steuerdaten ausgetauscht werden. Dementsprechend muss sichergestellt werden, dass besonders Daten, die von der Kopplung betroffen sind, also zwischen den Simulatoren ausgetauscht werden, die formulierten Restriktionen nicht verletzen. Der Anspruch der Co-Simulation ist die Abbildung der Stationsautomatisierung mit den eingesetzten IEDs, welche die Funktionen des Schutzes oder der Regelung umsetzen. Somit besteht die Notwendigkeit, die Restriktionen der vorgegebenen Übertragungszeiten einzuhalten.

Die Synchronisierung hingegen bezieht sich vornehmlich auf die Endgeräte. Da die Endgeräte jeweils in nur einer Simulation abgebildet werden, muss bei der Kopplung die Synchronisierung zwischen den Simulatoren beachtet werden. Nachfolgend werden die Anforderungen an die Co-Simulation und die Kopplung der beiden Simulatoren in Bezug auf die Synchronisierung und Übertragungszeit eingeordnet. Zusätzlich wird das Kriterium der Echtzeit ausgeführt, welches nicht unmittelbar aus IEC 61850 hervorgeht, jedoch das reale Verhalten der IEDs bestimmt.

3.4.1 Synchronisierungsanforderungen

Die Synchronisierungsanforderungen aus dem Normteil IEC 61850-5 müssen von der Co-Simulation eingehalten werden. Es gelten vor allem die unterschiedlichen Synchronisierungsklassen aus Tabelle 3.2. Bei der Co-Simulation und der Kopplung der beiden Simulatoren müssen demnach alle Instanzen untereinander synchronisiert sein und eine gemeinsame Zeitbasis verwenden. Es ist davon auszugehen, dass instanziierte Objekte innerhalb eines Simulators stets die gleichen Eigenschaften der Synchronisierung aufweisen, wie der Simulator selbst, da die Zeitbasis an die inneren Instanzen weitergegeben wird. Hier kann es jedoch zu einer Verzögerung der Zeitweitergabe kommen, die sich auf die Synchronisierungsqualität der implementierten Instanzen auswirken könnte. Demnach gilt zu beachten, dass die Co-Simulation bezüglich der Synchronisierung der Instanzen potenziell eine zusätzliche Ungenauigkeit $\Delta t_{sync,Sim}$ enthält. Abweichend zu der Berechnung der einzuhaltenden Synchronisierungsgenauigkeit aus Abschnitt 3.3.1 gilt folglich für die Co-Simulation:

$$\Delta t_{acc,Co-Sim} = |t_{master} - t_{slave,Instanz}| = |t_{master} - t_{slave,Simulator}| + \Delta t_{sync,Sim} \quad (3.2)$$

Die Restriktionen aus Tabelle 3.2 bleiben bestehen, wonach die maximale Abweichung des Zeitstempels zwischen zwei ausgeführten Instanzen maximal $2\Delta t_{acc}$ betragen darf. Die Synchronisierungsmethode der Co-Simulation muss dieses Kriterium erfüllen.

3.4.2 Übertragungszeitanforderungen

Analog zur Synchronisierung müssen die eingeführten Restriktionen bei der Übertragungszeit zwischen den Instanzen mit der Umsetzung der Co-Simulation eingehalten werden. Das betrifft vor allem den Fall, wenn zwei Instanzen die Informationen über IEC 61850 Netzwerkpakete austauschen, über die beiden Simulatoren verteilt sind. Hier hat die Kopplung der Simulatoren maßgeblichen Einfluss und führt zu Verzögerungen in der Paketübertragung. Nach Abbildung 3.3 ist die Definition der Übertragungszeit eingeführt. Die Geräte werden über ein beliebiges Medium verbunden und es resultiert eine Beschreibung von drei Verzögerungskomponenten. Die Zeitkomponenten t_a und t_c beschreiben die Verzögerung, die vom Erfassen der Daten bis zur Erstellung des IEC 61850 Pakets und umgekehrt entstehen. Verantwortlich sind die jeweiligen Kommunikationsstapel. Eine Kopplung von zwei Simulatoren hat hier keinen Einfluss, da sich diese Verzögerungskomponenten auf ein einzelnes Gerät beziehen

und Geräte nicht über mehrere Simulatoren hinweg modelliert werden. Dementsprechend haben die Konzeption und Umsetzung der Co-Simulation einen vorrangigen Einfluss auf die Größe der Verzögerungskomponente t_b .

Es gilt generell, dass t_a und t_c innerhalb der beiden Simulatoren anfallen und t_b zwischen den Simulatoren. Mit der Kopplung der Simulatoren und einem Simulatorübergreifenden Datenaustausch kommen zwei Verzögerungs-Terme (je Simulator einer) hinzu. In der Folge müssen diese bei der Konzeption und Umsetzung der Co-Simulation betrachtet werden. Der Term $t_{b,Trans}$ beschreibt dabei die Übertragungszeit, die zwischen den beiden Simulatoren liegt, $t_{b,Sim1}$ und $t_{b,Sim2}$ hingegen Terme, die innerhalb des jeweiligen Simulators auftreten können. Sie entstehen bei der Übergabe der Information des Simulationssystems an die reale Netzwerkschnittstelle. Es gilt folglich:

$$t_{TT,Co-Sim} = t_a + t_{b,Sim1} + t_{b,Trans} + t_{b,Sim2} + t_c \quad (3.3)$$

Die Restriktionen aus Tabelle 3.3 müssen beim Entwurf der Co-Simulation unter den Berücksichtigungen aus Gleichung (3.3) eingehalten werden.

3.4.3 Echtzeitanforderungen

Die eingeführten Anforderungen an die Co-Simulation zeigen, dass vor allem zeitliche Aspekte im Vordergrund stehen. Daher wird zusätzlich gefordert, dass die Simulatoren der Co-Simulation die definierten Echtzeitbedingungen erfüllen, um in der Lage zu sein, das reale Verhalten der IEC 61850-basierten Geräte in Echtzeit nachbilden zu können. Die eingeführten Definitionen 2.8 und 2.9 werden hierzu herangezogen.

Somit leitet sich ab, dass beide Simulatoren die formulierte Echtzeit-Eigenschaft einhalten müssen, mit einer zu definierenden Datenverarbeitungszeit. Aus den beschriebenen Vorgaben nach der Norm IEC 61850 werden keine konkreten Anforderungen an die Endgeräte oder ähnliches gestellt. Es wird lediglich auf die Einhaltung der Übertragungszeiten „für alle Betriebsbedingungen“ verwiesen. Dementsprechend muss die Co-Simulation die Übertragungszeiten aus Abschnitt 3.4.2 und Tabelle 3.3 zwischen den abgebildeten Instanzen der kommunizierenden Endgeräte einhalten. Zusätzlich gilt für die Nachbildung der Stationsautomatisierung, dass die in der Netzwerksimulation integrierten Funktionen in möglichst identischer Zeit zur Realität ohne Informationsverlust auszuführen sind.

4 Entwurf und Konzeption der Co-Simulation

Ableitet aus der Beschreibung des Anwendungsfalls der Stationsautomatisierung in Digitalen Stationen wird im vorliegenden Abschnitt ein Ansatz vorgestellt, das elektrische Energiesystem und das Netzwerkkommunikationssystem über separate Simulatoren übergreifend zu betrachten. Die essenziellen Bestandteile der eingeführten Co-Simulation werden nachfolgend näher beschrieben und auf die Ansätze und Ideen für die anschließende Umsetzung im Rahmen dieser Arbeit eingegangen. Beginnend mit der Beschreibung der beiden Betrachtungsebenen als separate Simulatoren, werden anschließend die relevanten Aspekte der Kopplung der zwei Simulatoren mitsamt der Alleinstellungsmerkmale des vorgestellten Ansatzes ausgeführt.

4.1 Modellierung des Stromnetzes

Die Grundlage vieler Co-Simulations-Ansätze für elektrische Versorgungsnetze bildet die Stromnetzsimulation. Die Stromnetzsimulation dient in der Regel dazu, verschiedene Situationen in einem modellierten Ausschnitt eines Stromnetzes simulativ zu untersuchen. Dabei wird grundlegend in zwei verschiedene Simulationsarten unterschieden, die sich in der Umsetzung voneinander unterscheiden. Man trennt zwischen statischen und dynamischen Simulationen. Mit einer statischen Simulation werden beispielsweise Lastflussberechnungen zur Bestimmung der Maximalbelastungen durchgeführt, die einen planerischen Charakter für den Netzausbau haben. Die dynamischen Simulationen hingegen haben das Ziel, die zeitabhängigen Größen im Stromnetz numerisch, nach festen Simulationszeitschritten zu berechnen. Innerhalb der dynamischen Simulationen kann zusätzlich noch in verschiedene Zeitskalen unterschieden werden. Vor allem die Unterscheidung zwischen Root Mean Square (RMS) und Electromagnetic Transient (EMT) ist hier relevant. Bei RMS spricht man auch oft von quasidynamisch, da die Simulation darauf basiert, Mittelwerte von den Grundgrößen Strom und Spannung zu bilden. Die EMT-Simulation basiert darauf, die Momentanwerte der Größen Strom- und Spannung zu jedem Abtastzeitpunkt, also in jedem Simulationsschritt, zu berechnen. Hierfür werden vom Simulator feste Simulationsschritte definiert, die Auswirkung auf die Genauigkeit der Simulation

haben. Hoch-abgetastete, dynamische Simulationen haben eine um ein Vielfaches höhere Berechnungs-Frequenz als die Grundfrequenz des Systems, also beispielsweise 50 Hz für das europäische Stromnetz [66].

4.1.1 Ansatz der Stromnetzsimulation für die vorgestellte Echtzeit-Co-Simulation

Für den Ansatz der Co-Simulation wird eine EMT-Simulation verwendet, die Momentanwerte der Größen Strom und Spannung mit einer Abtastung von $50 \mu\text{s}$ berechnet. Es handelt sich um eine dedizierte Simulationshardware von OPAL-RT mit der Simulationssoftware HYPERSIM. Die Momentanwerte werden innerhalb der vordefinierten Schrittweite numerisch berechnet und erlauben eine hoch-abgetastete Abbildung der zeitabhängigen Größen im Vergleich zu deren grundlegenden Eigendynamik in einem Stromnetz mit der Grundfrequenz von 50 Hz. Die hohe Abtastung ermöglicht es, beispielsweise Oberwellen und Übergangsvorgänge abzubilden und zu untersuchen. Hieraus können unter anderem Schaltvorgänge und deren dynamische Auswirkung auf die Strom- und Spannungsverläufe betrachtet werden. Für die Arbeit wird angenommen, dass der Simulator die formulierte Echtzeiteigenschaft nach Definition 2.9 erfüllt.

Die berechneten Momentanwerte können über physische Schnittstellen und Netzwerkschnittstellen aus dem Simulator ausgekoppelt und an externe Systeme gegeben werden. Für die Verwendung der Netzwerkschnittstellen sind bereits in dem verwendeten Simulator Protokollimplementierungen vorhanden, die Informationen aus der Simulation in die Netzwerkpakete über definierte Übertragungsprotokolle einbetten, wie dem IEC 61850, IEC 60870-5-104, MODBUS oder andere. In der Beschreibung des Anwendungsfalls wurde der Standard IEC 61850 eingeführt und seine Wichtigkeit für die Stationsautomatisierung herausgestellt. Folglich wird für den Aufbau der Co-Simulation dieser Kommunikationsstandard mit den unterschiedlichen Übertragungsprotokollen innerhalb des Simulators für die Stromnetzsimulation verwendet, um Informationen über ausgewählte Ströme und Spannungen an externe Systeme weiterzugeben und Stellbefehle entgegenzunehmen. Für die Co-Simulation ist vor allem der Prozessbus von Interesse, worauf in den späteren Kapiteln näher eingegangen wird.

4.1.2 Anforderungen an die Stromnetzsimulation

Ein Schwerpunkt des vorliegenden Co-Simulation-Ansatzes für die Stationsautomatisierung liegt auf der nachfolgenden Beschreibung der Virtualisierung des Kommunikationsnetzwerkes und der Kopplung zur Stromnetzsimulation. Die Stromnetzsimulation selbst steht weniger im Vordergrund. Aus diesem Grund wird die Stromnetzsimulation als Black-Box unter der Voraussetzung betrachtet, dass folgende Anforderungen stets erfüllt sein müssen:

1. Innerhalb der Stromnetzsimulation können die relevanten Zusammenhänge für Umspannwerke zwischen zwei Spannungsebenen modelliert und simuliert werden. Dabei können relevante Momentanwerte berechnet und Stellbefehle für die schaltbaren Betriebsmittel entgegengenommen werden.
2. Die Stromnetzsimulation ist in der Lage, in der geforderten Schrittweite nach IEC 61850-9-2LE Momentanwerte für Ströme und Spannungen an den Netzknoten zu berechnen. Dabei stellt der Simulator sicher, dass die interne Abtastung klein genug ist, um beispielsweise das Abtasttheorem auf Basis der Abtastung für den Versand von Netzwerkpaketen und die Anforderungen an Echtzeit zu erfüllen.
3. Die Stromnetzsimulation ist in der Lage, die berechneten Momentanwerte in Netzwerkpakete nach dem IEC 61850-9-2LE zu verpacken und über eine definierte Netzwerkschnittstelle zu schicken.
4. Die Stromnetzsimulation ist in der Lage, Stellbefehle über IEC 61850-8-1 GOOSE zu empfangen und deren Informationen an die modellierten Betriebsmittel wie Leistungsschaltern zu geben. Die Betriebsmittel reagieren zeitnah auf die Stellbefehle – idealerweise im nächsten Berechnungsschritt.
5. Die Stromnetzsimulation ist in der Lage, sich über ein Synchronisierungsprotokoll zu einem Zeitgeber über eine Netzwerkschnittstelle zu synchronisieren. Der Stromnetzsimulator bleibt stets synchron zu der vorgegebenen Zeitquelle mit der vorgegebenen Genauigkeit Δt_{acc} .

Es wird folglich nicht weiter auf die möglichen Modellierungen innerhalb der Stromnetzsimulation eingegangen und von einer Black-Box ausgegangen, die eine Simulation von zwei Spannungsebenen und einem Umspannwerk umsetzt. Die vorhandenen Netzwerk- und Kommunikationsschnittstellen werden verwendet und in den Ansatz der Co-Simulation einbezogen. Dabei wird der Prozessbus des modellierten Umspannwerkes mit den Messinformationen als SV mit der Möglichkeit der externen Einspei-

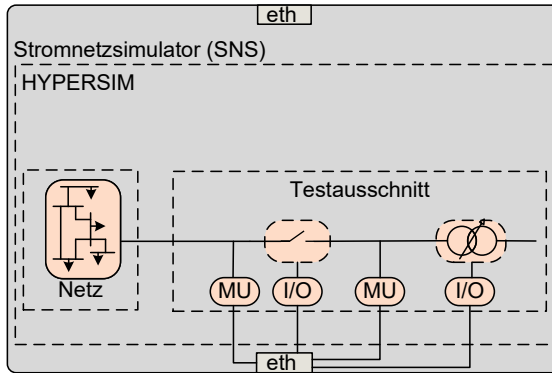


Abbildung 4.1: Schematischer Aufbau des Stromnetzsimulators (SNS)

sung von GOOSE-Stellsignalen zugänglich gemacht. Das Zeitverhalten beim Übertragen von Stellbefehlen wird nur exemplarisch betrachtet und es werden im Rahmen der Arbeit die formulierten Annahmen für die Echtzeit-Stromnetzsimulation stets als erfüllt angenommen. Die Annahme wird gestützt durch die Verwendung eines kommerziellen Echtzeit-Simulators, dessen Systemeigenschaften den geforderten Anforderungen entsprechen. Ein schematischer Aufbau des Stromnetzsimulator (SNS) ist in Abbildung 4.1 dargestellt. Der Ausschnitt entspricht einer Digitalen Station, die für die Kopplung innerhalb der Co-Simulation genutzt wird, inklusive der Grundelemente und der Anbindung an ein externes Netzwerkinterface.

4.2 Modellierung des Kommunikationsnetzwerkes

Neben dem SNS ist der zweite essenzielle Simulator für die Co-Simulation ein System zur Modellierung und Simulation des Kommunikationsnetzwerkes von Umspannwerken. Für Untersuchungen des Netzwerkverkehrs können sowohl Simulationssysteme als auch Emulationen verwendet werden. Die Simulationssysteme bilden das ereignisdiskrete Kommunikationsnetz durch Automatenysteme ab, wodurch sich Zustände und Zustandsübergänge untersuchen lassen [11]. Für solche Simulationssysteme stehen Untersuchungen des Netzwerkverhaltens im Vordergrund, wohingegen korrekte Protokollimplementierungen zur Erzeugung von realitätsnahem Netzwerkverkehr und die Implementierung von Systemfunktionalitäten nicht im Fokus stehen. Funkiona-

litäten sind hier in Bezug auf implementierte Endgeräte zu verstehen. Für die Co-Simulation sind die Erzeugung des Netzwerkverkehrs und die Implementierung von Funktionen, vor allem zur Abbildung von IEDs, von essenzieller Bedeutung. Das Ziel ist es, ein System zur Abbildung des Kommunikationsnetzwerkes zu definieren, das über die Implementierung von Übertragungsprotokollen des Anwendungsbereiches gekoppelt werden kann. Um die Funktion der Stationsautomatisierung nachbilden zu können, muss innerhalb des Kommunikationsnetzwerkes die Implementierung der IEC 61850 Protokolle und der Stationsfunktionalitäten möglich sein. Hier sollen die Funktionen der unterschiedlichen IED wie Schutz- oder Regelungssystemen nachgebildet werden.

Die beschriebenen Anforderungen werden von Simulationssystemen, wie NS-2, OM-NeT++ oder OPNET nicht ausreichend berücksichtigt und können somit nicht zur Lösung der Aufgabenstellung genutzt werden [61]. Weitere Anforderungen für die spätere Kopplung des Systems zur Co-Simulation sind die Echtzeitfähigkeit und Schnittstellenfähigkeit. Dementsprechend muss das Simulationssystem in der Lage sein, das Netzwerkverhalten in Echtzeit zu simulieren und den infolgedessen erzeugten Netzwerkverkehr über dedizierte Schnittstellen an externe Systeme übermitteln zu können.

Neben der klassischen Simulation kann auch eine Emulation mittels Virtualisierung, wie in Abschnitt 2.2.3 eingeführt, zur Nachbildung einer Netzwerkinfrastruktur genutzt werden. Um den Anforderungen der Co-Simulation zu entsprechen, müssen Komponenten implementiert werden, die auf Ereignisse automatisiert reagieren. Die Emulation sollte über eine Konfigurationsschnittstelle und eine isolierte Umgebung verfügen. Setzt man hier eine Virtualisierung innerhalb der Emulation ein, sollte die virtuelle Infrastruktur und integrierten Tools bedienbar sein.

Die Formulierung eines eigenen Ansatzes der Modellierung der Netzwerkinfrastruktur von Umspannwerken stellt einen essenziellen Baustein der Co-Simulation dar.

4.2.1 Ansatz der Modellierung des Kommunikationsnetzwerkes für die vorgestellte Echtzeit-Co-Simulation

Die Abbildung der Netzwerkinfrastruktur der Digitalen Stationen soll in der Folge über eine Virtualisierung umgesetzt werden. Dabei werden zwei Grundkonzepte herangezogen und zusammengeführt. Durch die Verwendung von Containern (beispielsweise Docker-Container) und SDN können virtuelle Abbildungen von IEDs der Stationsfunktionen über virtuelle OpenFlow-Switches miteinander verbunden werden, um die Kommunikation über das aufgebaute virtuelle Netzwerk umzusetzen.

Die verwendeten Grundkonzepte und deren Anwendung sowie das daraus entstehende Konzept der virtuellen IEDs werden in den folgenden Abschnitten beschrieben.

4.2.1.1 Docker-Container

In Abschnitt 2.2.3 wurde bereits die Nutzung von Containern, zur Umsetzung einer Virtualisierung von Funktionen, Systemen oder Geräten eingeführt. Dabei wird in einem Container ein eigener Namespace aufgebaut, in dem ein Linux-Betriebssystem als Basis-Image verwendet wird. Auf dieser Basis können Konfigurationen des Containers wie auch Netzwerkschnittstellen oder Softwarepakete vorgenommen werden. Damit bildet ein Container eine definierbare Umgebung für den Benutzer, um Funktionen darin zu implementieren. Für die Ausführung der Prozesse eines Containers wird auf das Basis-System des Virtualisierungshosts und dem zugrundeliegenden Linux-Scheduler zurückgegriffen. Es wird explizit kein eigenes Betriebssystem mit eigener Berechnungslogik für einen Container bereitgestellt. In der Regel haben Container das Ziel autonom ohne Benutzereingriff zu agieren. Es werden vornehmlich automatisierte Systeme oder Funktionen durch Container implementiert, wobei auch beispielsweise die Möglichkeit besteht, Webserver als Benutzerschnittstellen zu implementieren.

Die Möglichkeit, automatisierte Funktionen über Container auszuführen, entspricht der beschriebenen Anforderung, die Stationsfunktionen, die zumeist Automatisierungssysteme beschreiben, umzusetzen. Folglich eignen sich Docker-Container für den Ansatz der Co-Simulation zur Implementierung der Stationsfunktionen in gekapselten Containern mit dedizierten Netzwerkschnittstellen. Ein Container stellt dementsprechend im Rahmen der Arbeit eine Abbildung eines kompletten physischen Geräts in der virtuellen Umgebung dar [67]. Die nachfolgend beschriebenen Container der Stationsfunktionen (entspricht den IEDs) werden als virtuelle IEDs (Virtuelles Intelligent Electronic Device (vIED)s) definiert und für die Co-Simulation herangezogen.

4.2.1.2 Virtuelle IEDs

Im Grunde besteht ein reales IED aus mehreren Bestandteilen, um analog zu der eingeführten Definition 3.2 von IEDs eine Funktion innerhalb der Stationsautomatisierung zu erfüllen. Das Gerät muss über eine Netzwerkschnittstelle Informationen verarbeiten, die über Netzwerkprotokolle übermittelt werden. Nachdem die Informationen aus den Netzwerkpaketen extrahiert wurden, müssen sie intern bereitgestellt werden. Die umzusetzenden Funktionen für die Stationsautomatisierung sind vor allem Schutz- und Regelungsfunktionen. Es werden primär Messdaten von prozessnahen

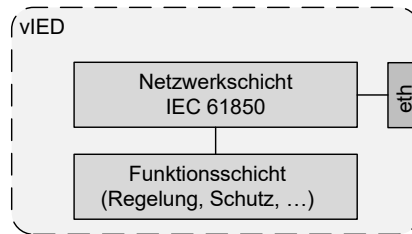


Abbildung 4.2: Grundaufbau eines vIEDs

MUs entgegengenommen und Schutz- oder Regelungsfunktionen gespeist. Das Beispiel eines Schutzsystems wertet eingehende Messinformationen aus. Bei Verletzungen von Schutzkriterien wie Überstrom oder Überspannung generiert das Schutzsystem Schaltbefehle, um der Verletzung entgegenzuwirken. Als ein Beispiel wird ein angeschlossener Leistungsschalter mittels eines Stellbefehls angesteuert. Also muss das IED die erzeugten Ausgangsinformationen wiederum in ein Übertragungsprotokoll überführen und anschließend über die Netzwerkschnittstelle bereitstellen.

Aus dieser grundlegenden Beschreibung von IEDs wird abgeleitet, dass zwei wichtige Bestandteile identifiziert werden können. Zum einen eine Netzwerkschicht, die die Extraktion der Informationen aus empfangenen Daten und die Überführung der internen Informationen in das Übertragungsprotokoll zum Versand von Netzwerkpaketen realisiert. Weiterhin wird eine Funktionsschicht definiert, die die Informationen von der Netzwerkschicht entgegennimmt, weiterverarbeitet und Ausgangsinformationen zurück an die Netzwerkschicht übergibt. Neben diesen beiden Grundelementen können noch weitere Funktionen innerhalb eines IEDs implementiert sein, die nur indirekt der Stationsfunktionalität dienlich sind. Hierunter können beispielsweise interne Datenhaltungssysteme wie Datenbanken oder Log-Funktionalitäten verstanden werden. Zur weiteren Vereinfachung werden diese zusätzlich möglichen Bestandteile vernachlässigt und die zwei direkten Bestandteile für die Erfüllung der Stationsfunktion weiter betrachtet.

Die eingeführten vIEDs müssen die beiden wesentlichen Bestandteile der Netzwerk- und Funktionsschicht in einem Container umsetzen. Der grundlegende Aufbau eines vIEDs ist Abbildung 4.2 zu entnehmen. Wie in Kapitel 3.2 eingeführt, wird der Übertragungsstandard IEC 61850 nachfolgend verwendet, weswegen dieser durchgehend für die Netzwerkschicht verwendet wird.

Auf die expliziten Umsetzungen und Implementierungen wird in den folgenden Kapiteln näher eingegangen. Es sind verschiedene, frei verfügbare Protokollstacks in unterschiedlichen Programmiersprachen für die Netzwerkschicht vorhanden, die die Anforderungen der Norm IEC 61850 erfüllen. Auch die Implementierung der Stationsfunktion kann mittels unterschiedlicher Tools und Programmiersprachen erfolgen. Die Umsetzungen im Rahmen der vorliegenden Arbeit erfolgen sehr rudimentär zur Erfüllung exemplarischer Stationsfunktionen. Der Aufbau und das Design lässt es jedoch auch zu, sehr detaillierte Implementierungen bis hin zu original getreuen Implementierungen von kommerziellen Systemen, die als digitales Abbild der realen Komponenten verstanden werden kann, zu integrieren.

Neben den essenziellen zwei Schichten des vIEDs wird mindestens ein Ethernet Netzwerkadapter (eth), oftmals auch als Network Interface Card (NIC) bezeichnet, benötigt, um die Netzwerkpakete analog zum realen IED entgegennehmen zu können. Über die Docker-Container-Implementierung können virtuelle Netzwerkadapter implementiert werden, die im Namespace des Containers angelegt werden. In der Grundimplementierung wird ein Netzwerkadapter mit dem Namen „docker0“ angelegt, der über ein Standardnetzwerk mit dem Virtualisierungshost und weiteren instanziierten Containern kommunizieren kann. Zusätzlich können weitere Netzwerkadapter im Namespace definiert und über eine dedizierte Netzwerkadresse angesprochen werden [68].

4.2.1.3 Software-Defined-Network

Das Konzept von SDN zielt auf die Abstraktion des Netzwerkes und funktionale Trennung in drei Ebenen ab. Zentrales Konzept von SDN ist die Trennung zwischen der Daten- und Kontrollschicht. Oftmals werden die englischen Begriffe Data- und Control-Plane verwendet. Dabei werden die Daten, wie beispielsweise Netzwerkpakete, auf der untersten Ebene nach einfachen Regeln verarbeitet. Es können beispielsweise unterschiedliche Implementierungen von Switches für das Routing gewählt werden. Die Logik des Routings wird von der Data-Plane, auf der sich die Routing-Komponenten befinden, getrennt und in der Control-Plane implementiert. Hier können sogenannte SDN-Controller umgesetzt sein, die die Regeln für die Data-Plane auf Basis von Software-Implementierungen bestimmen. Dementsprechend erfolgt durch das SDN eine Zentralisierung der Logik im Netzwerk, indem zentrale Controller eingesetzt werden und bestimmend für das Netzwerkverhalten sind. Die beiden Ebenen sind über eine Standardschnittstelle, das Southbound Interface, verbunden. Die

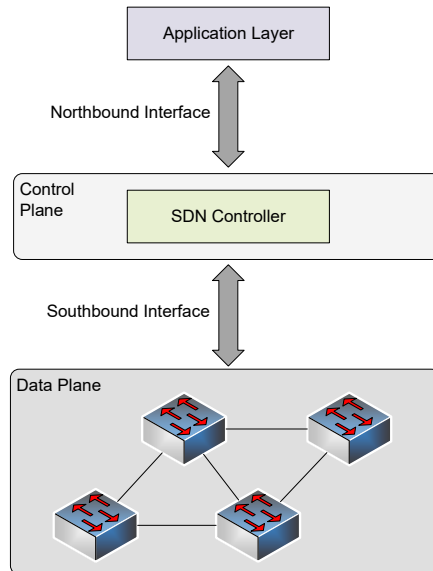


Abbildung 4.3: Grundlegende SDN-Architektur und Grundprinzip der Ebenen-Trennung

Standardimplementierung für das Southbound Interface wird durch OpenFlow [69] realisiert und ermöglicht eine offene und gesicherte Kommunikation zwischen den Switches und dem Controller [70]. Eine dritte Ebene dient als Benutzerschnittstelle, oftmals Application- oder Management-Layer genannt. Hier kann eine Einbindung der Controller in unterschiedliche Tools, Umgebungen oder Applikationen über das sogenannte Northbound Interface erfolgen. Der Grundaufbau von SDN ist in Abbildung 4.3 dargestellt.

Es existieren diverse softwaretechnische Umsetzungen von SDN-Architekturen mit unterschiedlichen Schwerpunkten. Das Grundprinzip basiert, wie beschrieben, auf einer Zentralisierung der Netzwerkfunktion, die von den SDN-Controllern übernommen wird. Generell kann jedoch zwischen zentralisierten und verteilten SDN-Controllern unterschieden werden.

- Zentralisiert: Der zentralisierte Controller ermöglicht die zentrale Verwaltung des Netzwerks, wobei ein einziger Controller das gesamte Netzwerk verwaltet. Der Controller kann sich mit mehreren Switches verbinden, die mit mehreren Hosts kommunizieren. Der zentralisierte Controller fungiert als einziger An-

sprechpartner. Beispielhafte Implementierungen stellen NOX [71] und Ryu [72] dar.

- Verteilt: Die verteilten Controller werden hauptsächlich in WAN und Rechenzentren eingesetzt, wo Skalierbarkeit erforderlich ist. Die Controller sind dezentral verteilt, wobei die Switches mit verschiedenen Controllern verbunden werden können. Die verteilten Controller ermöglichen unter anderem ein Clustering. Beispielhafte Implementierungen verteilter SDN Controller sind ONOS [73] und OpenDayLight [74].

In Verbindung mit der Implementierung von OpenFlow-Switches kann der SDN-Controller die Funktion von konventionellen Switches mit Routing Vorgaben oder sogar Firewalls mit dedizierten Filtern in Form von adaptiven Flow-Tabellen übernehmen. Die OpenFlow-Funktionen können sowohl in Hardware-Plattformen implementiert als auch rein virtuell instanziiert werden. Das SDN ermöglicht es so, das Netzwerkverhalten rein softwarebasiert zu steuern. Aus diesem Grund wird das SDN oftmals für Anwendungen der Virtualisierung verwendet. Die virtuellen Instanzen, wie virtuelle Maschinen oder Container, können über OpenFlow-Switches miteinander verbunden werden und über das so geformte Netzwerk über einen SDN-Controller gesteuert werden. Es entsteht ein komplett virtuelles Netzwerk, das über gegebene Schnittstellen konfigurierbar ist.

Darüber hinaus kann ein Netzwerk über die beschriebene Architektur virtualisiert und über dedizierte Tools instanziiert und verwaltet werden. Ein Tool, das virtuelle Testbeds auf Basis der SDN-Architektur aufbaut und weit verbreitet für das Zusammenspiel von SDN und OpenFlow für die Prototypen-Entwicklung ist, stellt Mininet dar [75]. Mininet erlaubt es, ein virtuelles Netzwerk durch virtuelle Hosts und Switches zu beschreiben und mit einem SDN-Controller zu verwalten. Eine Erweiterung von Mininet stellt das Tool Containernet [76] dar, das die Einbindung von Docker-Containern als virtuelle Hosts erlaubt.

4.2.2 Zusammenführung der Ansätze zum Kommunikationsnetzwerksimulator

Die zentralen Komponenten für den Virtualisierungsansatz der Kommunikationsebene der Stationsautomatisierung werden zusammengeführt, um die Umsetzung des Kommunikationsnetzwerksimulator (KNWS) zu beschreiben. Mittels Containernet wird ein Testbed aufgebaut, in dem Docker-Container als virtuelle Hosts instanzii-

iert werden. Diese Container sind mit virtuellen Netzwerkadaptoren ausgestattet, die nicht mit dem Docker-Standard-Netzwerk (docker0), sondern über dedizierte Verbindungen kommunizieren. Für die Kommunikation werden über Containernet virtuelle OpenFlow-Switches definiert und entsprechend mit den Containern verbunden. Zusätzlich wird ein SDN-Controller instanziiert, der über das OpenFlow-Protokoll mit den Switches kommuniziert und das Netzwerkverhalten steuert. Der Controller kann somit eine logische Verbindung von Containern über den OpenFlow-Switch implementieren, indem sogenannte Flow-Regeln definiert und an die zugehörigen Switches übergeben werden.

Insgesamt kommunizieren die Container über das SDN-Netzwerk miteinander und tauschen Daten untereinander aus. Mit der Implementierung nach dem Konzept der vIEDs werden die geforderten Stationsfunktionen und die Schnittstellen des IEC 61850 Übertragungsprotokoll integriert. Folglich können die einzelnen Container nun IEC 61850 Netzwerkpakete wie beispielsweise Messdaten oder Stellbefehle untereinander austauschen. Bei korrekter Anordnung der Container und Ausstattung mit Implementierungen von Stationsfunktionen laufen die automatisierten Funktionen innerhalb eines Umspannwerkes wie Regelungen und Schutzalgorithmen autonom auf den vIEDs.

Das Virtualisierungskonzept für die Modellierung der Kommunikations- und Sekundärtechnik beziehungsweise Stationsfunktionen innerhalb von Digitalen Stationen ist in Abbildung 4.4 dargestellt, in der drei beispielhafte vIEDs an einen OpenFlow-Switch angeschlossen sind, der den Prozessbus des Umspannwerkes definiert. Die vIEDs und der OpenFlow-Switch werden über Containernet gestartet und orchestriert. Der Switch ist mit einem separat gestarteten SDN-Controller verbunden. Containernet realisiert die OpenFlow-basierte Verbindung zwischen dem SDN-Controller und dem Switch. Sämtliche Instanzen werden gebündelt auf einem Virtualisierungs-Host gestartet, dessen Netzwerkschnittstellen ebenfalls mit dem virtuellen Prozessbus-Switch verbunden werden können. Demnach wurde die Möglichkeit geschaffen, die Virtualisierung der Kommunikation auf einem Host, dem KNWS, zu platzieren und mit einer Schnittstelle zu verbinden. Zudem kann die Schnittstelle von der Co-Simulation genutzt werden, um eine Verbindung mit einem zweiten Simulator herzustellen.

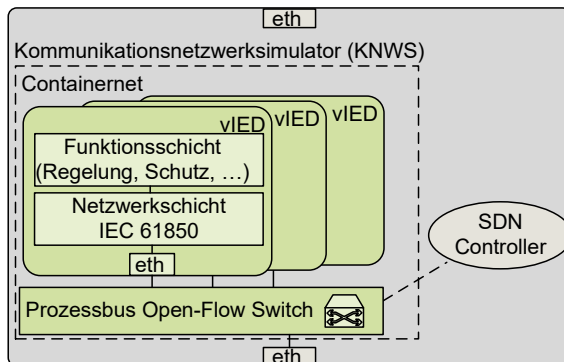


Abbildung 4.4: Virtualisierungskonzept des KNWS mit der Verschneidung aus vIEDs und einer SDN-Architektur mittels Containernet

4.2.3 Anforderungen an die Modellierung des Kommunikationsnetzwerkes

Zusammenfassend wurde ein Ansatz eingeführt, der die Abbildung eines Netzwerkes und deren Funktionen durch eine Emulation mittels Virtualisierung auf der Basis von Docker-Containern und einer SDN-Architektur erlaubt. Die Ausgestaltung der Emulation durch Definition der SDN-Netzwerkinfrastruktur und Stationsfunktionen durch vIEDs entspricht einer Modellierung innerhalb des Simulators. Im Gegensatz zu den meisten in Abschnitt 2.4.3 beschriebenen veröffentlichten Co-Simulationen mit klassischen Netzwerksimulations-Tools kann die notwendige Umsetzung der Stationsfunktionen für die IEC 61850 basierten Umspannwerken in Echtzeit erfolgen. Es handelt sich somit um eine Echtzeit-Ausführung von (vereinfachten) Modellen von Stationsfunktionen in einer konfigurierbaren virtuellen Netzwerkumgebung. Die Implementierung der Netzwerk- und Automatisierungsfunktionen kann als Emulation verstanden werden. Dabei werden im Laufe der Arbeit auf die Einflussgrößen des Ansatzes und auf die Anforderungen eingegangen, die bestehen, um eine gültige Co-Simulation für den IEC 61850 Anwendungsfall zu genügen. Nachfolgend werden die einzuhaltenden Anforderungen für die vorgestellte Co-Simulation aufgezählt. Im Vergleich zu der Stromnetzsimulation steht die Umsetzung und genaue Betrachtung der Einflüsse und Randbedingungen der Netzwerkebene im Vordergrund und wird

in den folgenden Abschnitten näher betrachtet. Der entwickelte Ansatz stellt den wesentlichen Kern der vorliegenden Arbeit dar.

1. Die Emulation des Netzwerks ist in der Lage, in der geforderten Schrittweite nach IEC 61850-9-2LE Momentanwerte für Ströme und Spannungen an definierten Netzknoten entgegenzunehmen.
2. Die Emulation ist in der Lage, die Strom- und Spannungsinformationen aus den Netzwerkpaketen nach IEC 61850-9-2LE zu entpacken und weiterzuverarbeiten.
3. Die Emulation ist in der Lage, Stellbefehle über IEC 61850-8-1 GOOSE zu generieren und zu senden. Dies wird realisiert auf Basis einer autonom arbeitenden Automatisierungsfunktion, wie beispielsweise einer Schutzfunktion, die innerhalb eines VIEDs implementiert ist.
4. Die Emulation ist in der Lage, sich über ein Synchronisierungsprotokoll zu einem Zeitgeber über eine Netzwerkschnittstelle zu synchronisieren. Die Emulation bleibt stets synchron zu der vorgegebenen Zeitquelle mit der vorgegebenen Genauigkeit Δt_{acc} .
5. Der Einfluss der Emulation auf die Übertragungszeit von Netzwerkpaketen kann abgeschätzt werden und verletzt nachweisbar nicht die definierten Vorgaben nach IEC 61850-5.
6. Die Emulation ist nach der eingeführten Definition 2.8 echtzeitfähig. Der Host stellt demnach sicher, dass die Prozessschrittlänge so kurz ist, dass das Verhalten der zu modellierenden Systeme ausreichend genau nachgebildet werden kann.

4.3 Ansatz der Kopplung

Um die beiden Simulatoren zu verschneiden und eine Co-Simulation zu entwerfen, ist es notwendig, dass die Stromnetzsimulation und die Virtualisierung der Kommunikationsebene miteinander verbunden werden. Wie in Abschnitt 2.1.1 eingeführt, existieren verschiedene Ansätze und Möglichkeiten der Kopplung von Simulatoren. Es können flexible Ansätze gewählt werden, die auf unterschiedliche Simulatoren und Anwendungsfälle anwendbar sind. Dabei muss der Kopplungsansatz stets die eingeführten Aufgaben der Koordination (Datenaustausch und Synchronisierung) erfüllen, was in der Regel über einen expliziten Koordinator erfolgt. Die Abbildung 4.5 zeigt die Erweiterung von Abbildung 2.3 um die Umsetzungen im Rahmen des vorgestellten Ansatzes.

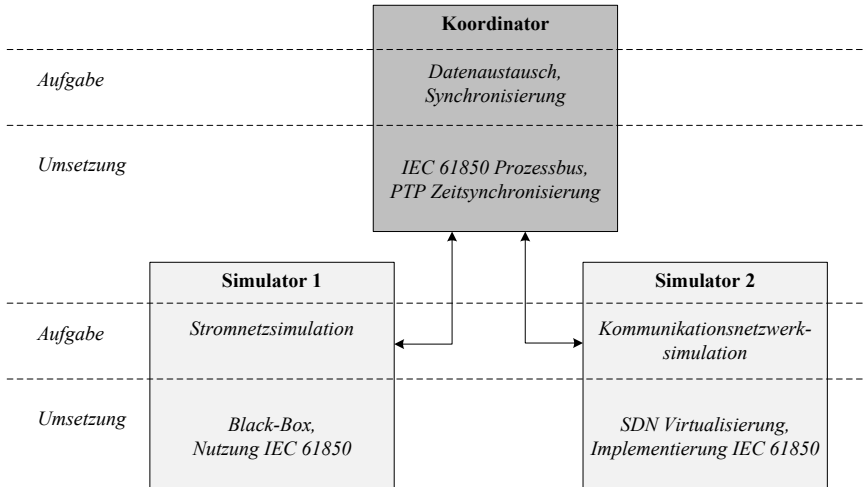


Abbildung 4.5: Aufgaben und Umsetzung der jeweiligen Instanzen der Co-Simulation

Für den beschriebenen Anwendungsfall der Stationsautomatisierung und den beiden eingeführten Simulatoren, die in Echtzeit ausgeführt und miteinander gekoppelt werden sollen, sind die verfügbaren Koordinationsansätze nicht anwendbar. Begründet ist dies aus dem identifizierten Fehlen von Echtzeit-Unterstützung und der Herausforderung einer externen Ansteuerung. Es wird nachfolgend ein neuer Ansatz entworfen, um die beiden Ebenen direkt miteinander zu verbinden und den Datenfluss sowie die Synchronität zu gewährleisten. Dabei werden die, für den vorliegenden Fall speziellen Anforderungen berücksichtigt und es kann von einer Ad-hoc-Kopplung, also einer spezifisch für den Anwendungsfall definierten und nicht universell anwendbaren Kopplung, aus der eingeführten Beschreibung in Abschnitt 2.4.2.3 gesprochen werden.

Aus den Beschreibungen der vorgestellten Simulatoren geht hervor, dass beide externe Schnittstellen über das IEC 61850 Protokoll verfügen und für eine Echtzeit-Ausführung ausgelegt sind. Infolgedessen wird eine Kopplung zur Laufzeit entworfen, in der beide Simulatoren direkt miteinander verbunden sind, separat gestartet werden und Daten über eine etablierte Verbindung untereinander austauschen. Beide Simulatoren werden von den eingehenden Daten und Informationen getriggert, um ihre internen Prozesse nachfolgend anzustoßen. Zusätzlich sorgt die Verbindung der

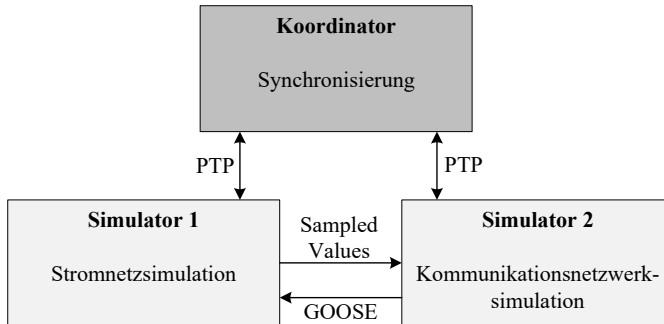


Abbildung 4.6: Kopplung der Simulatoren mit Synchronisierung über den Koordinator

Simulatoren mit einer Synchronisierungsinstanz (Koordinator) für die Zeitgleichheit der Systeme.

Der Datenaustausch der Simulatoren erfolgt nicht über den Koordinator. Vielmehr werden beide Simulatoren direkt miteinander verbunden. Die direkte Kopplung erlaubt eine Sicherstellung des Datenaustauschs zwischen den beiden Simulatoren aufgrund der Verwendung des gleichen Übertragungsprotokolls. Die zweite essenzielle Kopplungsaufgabe nach Abbildung 4.5 ist jedoch hinsichtlich der Synchronisierung nicht direkt erfüllt. Für die Synchronisierung wird eine zusätzliche Instanz eingeführt, die diese Teilaufgabe des Koordinators übernimmt und in Abbildung 4.6 dargestellt wird.

Der Aufbau der Kopplung ist Abbildung 4.7 zu entnehmen, wobei zunächst nur eine Netzwerkschnittstelle jedes Simulators genutzt und eine einzelne direkte Kopplung aufgebaut wird. Es besteht jedoch die Möglichkeit, zusätzliche Verbindungen zwischen den Systemen zu etablieren, wenn beispielsweise ein weiterer Abgang des Umspannerkes untersucht werden soll. Zur anfänglichen Vereinfachung wird zunächst von einer physischen Verbindung ausgegangen. Abbildung 4.7 zeigt die direkte Kopplung der beiden Simulatoren und die schematische Umsetzung der Stromnetzsimulation im SNS und Emulation des Kommunikationsnetzwerkes im KNWS.

Zusätzlich sind beide Simulatoren mit einem Precision Time Protocol (PTP)-Zeitgeber verbunden, der als Koordinator fungiert. Hierfür wird jeweils ein dedizierter Netzwerkadapter verwendet.

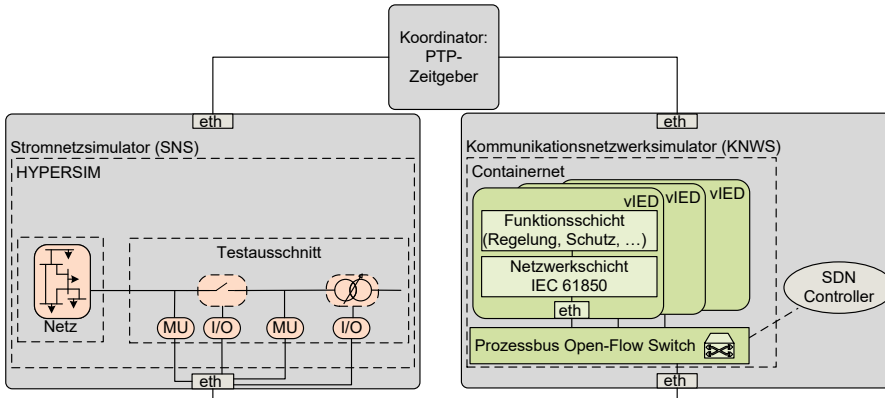


Abbildung 4.7: Umsetzung der Kopplung beider Simulatoren untereinander und mit dem Koordinator

Die Ausgestaltung der beiden Koordinator-Aufgaben Datenaustausch und Synchronisierung werden in den folgenden beiden Abschnitten erläutert. Die Implementierung wird in Kapitel 5 dargelegt.

4.3.1 Konzeption des Datenaustauschs

Die ausgetauschten Daten können durch die beschriebenen Schnittstellen direkt vom jeweiligen System interpretiert werden und bedürfen keiner zusätzlichen Umwandlung. Innerhalb der Stromnetzsimulation, die im Bezug auf die Co-Simulation vorrangig als Black-Box angesehen wird, ist unmittelbar eine IEC 61850 Schnittstelle integriert und garantiert das korrekte Verpacken und Extrahieren der Informationen in und aus IEC 61850 Paketen nach den Protokolltypen SV und GOOSE. Andererseits beinhalten die eingeführten vIEDs angepasste IEC 61850 Schnittstellen, um die Protokollinformationen ebenso korrekt zu verarbeiten. Die beiden Simulatoren bilden mit ihren erzeugtem IEC 61850 Netzwerkverkehr einen gemeinsamen Prozessbus, auf dem sämtlicher Datenverkehr zwischen IEDs auf der Prozessebene erfolgt. Der gesamte Netzwerkverkehr läuft auf den virtuellen Switches innerhalb der Netzwerkvirtualisierung auf. Aufgrund der parallelen Ausführung der beiden Simulatoren ist keine Zwischenspeicherung der Daten von einem Koordinator notwendig. Die erzeugten Daten werden innerhalb der beiden Simulatoren direkt und ohne zusätzliche Verzögerung an den jeweils anderen Simulator gesendet. Dabei wird sichergestellt, dass keine In-

stanz die Laufzeit oder den Inhalt der Netzwerkpakete verändert oder beeinflusst. Die unterschiedlichen Charakteristiken der zeitdiskreten dynamischen Stromnetzsimulation und der ereignisdiskreten Betrachtung des Kommunikationsnetzwerkes bilden in der vorliegenden Co-Simulation keine zusätzliche Aufgabe der Synchronisierung für den Koordinator. Diese Annahme setzt voraus, dass die Abtastung der zeitdiskreten Simulation deutlich höher ist als die minimale Zeit zwischen zwei übermittelten Ereignissen.

Unter diesen Voraussetzungen ist für den Datenaustausch keine Koordinator-Funktion in dem untersuchten Anwendungsfall notwendig. Beide Simulatoren gewährleisten die gleichartige Nutzung von IEC 61850 Schnittstellen, wonach der gemeinsame Prozessbus zum Datenaustausch gebildet werden kann. Ein anderer Datenaustausch zwischen beiden Simulatoren wird ausgeschlossen und wird nicht in die Co-Simulation einbezogen. Sollten für zukünftige Anwendungen noch weitere Informationen zwischen den Simulatoren ausgetauscht werden müssen, kann eine separate Koordinator-Funktion für den Datenaustausch notwendig werden. In der vorliegenden Arbeit wurde jedoch der Fokus auf den Datenaustausch über den gemeinsamen Prozessbus gelegt.

4.3.2 Synchronität

Für die Synchronität der Co-Simulation kommen die in Abschnitt 2.3.1 beschriebenen Merkmale sowie die Unterscheidung zwischen Ereignis- und Zeitgleichheit zum Einsatz. Ereignisgleichheit bedeutet, dass Statusänderungen im Kommunikationsnetz, die das elektrische Netz beeinflussen, auch in der Stromnetzsimulation korrekt und konsistent abgebildet werden. Diese Ereignisse müssen in beiden Simulatoren gleichzeitig eintreten, um Abhängigkeiten und Wechselwirkungen realistisch zu erfassen [10].

Voraussetzung dafür ist die Zeitgleichheit beider Simulatoren, insbesondere im Hinblick auf ihre parallele Ausführung. Beide müssen auf einer identischen Zeitbasis arbeiten, um zeitlich konsistente Abläufe sicherzustellen. Während die Stromnetzsimulation auf einen internen Referenzzeitstempel zugreift, orientiert sich die Virtualisierung an der Systemzeit des Ausführungsrechners. Abweichungen in der Zeitbasis würden zu Beschleunigungen oder Verzögerungen einzelner Simulationsprozesse führen und das Gesamtergebnis verfälschen. Daher müssen die Zeiteinheiten und deren Länge in beiden Systemen exakt übereinstimmen.

Zur Gewährleistung dieser Zeitgleichheit werden gängige Synchronisationsprotokolle wie Network Time Protocol (NTP) oder PTP eingesetzt, die eine zentrale Referenz-Zeitbasis an alle Netzwerkteilnehmer verteilen. Es wird angenommen, dass beide Simulatoren die über das Netzwerk empfangene Zeitbasis unverändert für ihre Ausführung nutzen. Damit sind auch alle abgeleiteten Abläufe innerhalb der Simulationen synchron zum jeweiligen Zeitsystem. Die Genauigkeit der Synchronisation hängt vom eingesetzten Protokoll ab: PTP erlaubt durch Offset-Korrekturen in der Regel eine deutlich präzisere Abstimmung als NTP, wie in Gleichung (3.1) dargestellt. In der Co-Simulation übernimmt die Zeitgeber-Instanz somit die Rolle eines zentralen Koordinators, der die zeitliche Konsistenz aller beteiligten Systeme sicherstellt.

Für die Umsetzung der Co-Simulation wird PTP verwendet, das einen PTP-Master im Netzwerk erfordert, der in festen Abständen Referenz-Zeitsignale mit Latenzkompensation versendet. Eine PTP-Implementierung kann eine Genauigkeit im Sub-Microsekunden-Bereich erzielen und wird daher auch für verschiedene reale Anwendungen in einer Digitalen Station gefordert. In der Folge wird für die Co-Simulation abgeleitet, dass die gemeinsame Genauigkeit der Synchronisierung nach der Gleichung (3.2) eingehalten wird. Zusätzlich wird für die Etablierung der Synchronisierung gefordert, dass beide Simulatoren sich über einen dedizierten Kanal mit dem Koordinator unter Nutzung von PTP synchronisieren. Der dedizierte Kanal ist nicht identisch mit dem Datenaustausch zwischen den Simulatoren. Die Synchronisierung darf folglich den Datenaustausch der Co-Simulation nicht beeinflussen und keinen zusätzlichen Verkehr auf dem Prozessbus erzeugen. Der PTP-Master entspricht in dem vorliegenden Ansatz dem Koordinator der Co-Simulation und ist ausschließlich für die Synchronisierung der Simulatoren verantwortlich.

4.4 Zusammenfassung des Ansatzes

Der vorgestellte Ansatz der Co-Simulation zeichnet sich durch die parallele Ausführung von zwei separaten Simulatoren aus, die jeweils eine Betrachtungsebene der Stationsautomatisierung beinhalten. Eine Echtzeit-Stromnetzsimulation mit gegebenen Schnittstellen wird mit einem neuartigen Ansatz einer Emulation durch die Virtualisierung der Kommunikationsebene von Strukturen der Stationsautomatisierung direkt gekoppelt. Mittels einer PTP-Synchronisierung können die Anforderungen an eine Co-Simulation und den formalen Koordinator erfüllt werden. Die Aufgabe der Realisierung des Datenaustauschs des Koordinators kann durch die direkte Kopplung und Nutzung der IEC 61850 Protokolle entfallen.

Die beschriebenen Abbildungen der beiden Betrachtungsebenen sollen eine vollständige Simulation der Stationsautomatisierung von Digitalen Stationen im Rahmen der Co-Simulation erlauben. Der vorgestellte Ansatz ermöglicht die gemeinschaftliche Untersuchung inklusive der korrekten Abbildung der Abhängigkeiten zwischen den Betrachtungsebenen. Die Umsetzung im Rahmen der Arbeit wird im nächsten Kapitel vorgestellt und die Gültigkeit des Ansatzes bewiesen. Die notwendigen Nachweise der aufgestellten Thesen werden in den nachfolgenden Abschnitten erbracht.

5 Anwendung der Co-Simulation für die Stationsautomatisierung

In diesem Kapitel werden die modellierten Umgebungen für die formulierte Co-Simulation der Stationsautomatisierung eingeführt. Beginnend mit den Eigenschaften und Beschreibungen der Stromnetzsimulation, werden die Untersuchungsgegenstände und deren Spezifika beschrieben. Anschließend folgt die Umsetzung der Virtualisierung des Kommunikationsnetzwerkes. Zudem werden die Einschränkungen für die Co-Simulation der zu untersuchenden Digitalen Station vorgestellt.

5.1 Grundaufbau der Stromnetzsimulation eines Umspannwerkes

Aus den Beschreibungen des Ansatzes der Co-Simulation ist hervorgegangen, dass der Schwerpunkt der vorliegenden Arbeit auf der Erstellung der Virtualisierung der Kommunikationsebene in Form des KNWS liegt. Um aber die vollständige Kopplung zu einer Co-Simulation umzusetzen, werden die notwendigen Anforderungen an die Modellierung innerhalb der Stromnetzsimulation erläutert. Die Grundannahmen gelten dabei aus Kapitel 4.1. Dementsprechend wird davon ausgegangen, dass dynamische Verläufe innerhalb des Stromnetzes simuliert, Messwerte als Momentanwert über eine Kommunikationsschnittstelle an externe Systeme geleitet und externe Signale in die Stromnetzsimulation integriert werden können. Die Formulierung und Definition der Co-Simulation beschränkt sich auf die Stationsautomatisierung mit deren prozessrelevanten Komponenten und Strukturen. Demnach werden die Modellierungen innerhalb eines Umspannwerkes als Hauptuntersuchungsobjekt für die Co-Simulation eingeführt und nachfolgend kurz beschrieben. Dabei werden die in Kapitel 3.1 beschriebenen Komponenten des Umspannwerkes betrachtet.

5.1.1 Modellierung der Primärtechnik in der Stromnetzsimulation

Für die grundlegende Funktion eines Umspannwerkes muss ein Transformator innerhalb der Stromnetzsimulation modelliert werden, der mit zwei verschiedenen Spannungsebenen über seine Primär- und Sekundärseite verbunden ist. Stufenschalter oder

Stufensteller bilden die Möglichkeit, die Ausgangsspannung durch Umschaltung des Abgriffs auf der Unterspannungsseite zu regeln. Je nach Komplexität des Umspannwerkes können mehrerer Einspeisestränge und Abgänge gebündelt modelliert und in einem Umspannwerk zusammengefasst werden. Die Verteilung ist dabei über eine Sammelschiene abbildbar. Dabei können die einzelnen Stränge und Abgänge über dedizierte Leistungsschalter zu- und abgeschaltet werden. Dementsprechend kann ein Schaltfeld nachgebildet werden. Die Simulation des elektrischen Verhaltens eines Umspannwerkes orientiert sich somit an den abgebildeten Komponenten und Zusammenhängen des elektrischen Stromlaufplans, auch Einliniendiagramm genannt, und ist ein Bestandteil der in Abbildung 4.1 bereits dargestellten Co-Simulation. Für die elektrische Simulation eines Umspannwerkes ist die Modellierung folgender Primärsysteme notwendig: Transformatoren, Stufensteller, Sammelschienen und Leistungsschalter. Die Möglichkeiten der Komponentenmodellierung und Umsetzungen werden hierbei nicht weiter betrachtet.

5.1.2 Modellierung der Sekundärtechnik in der Stromnetzsimulation

In der Stromnetzsimulation wird lediglich die Primärtechnik abgebildet. Im Rahmen der Co-Simulation muss diese über Schnittstellen der zu emulierenden, externen Sekundärtechnik zugänglich gemacht werden. Grundlegend bieten Simulatoren der Stromnetzsimulationen die Abbildung der Sekundärtechnik an, die sich aber ebenso mit externen Geräten zur Umsetzung der Funktionalitäten koppeln lassen. Beispielsweise wird bei einem Power-Hardware-in-the-Loop (PHiL) oder Controller-Hardware-in-the-Loop (CHiL) ein realer Regler oder auch ein Schutzsystem, also ein Sekundärsystem, mit der Stromnetzsimulation gekoppelt, um deren Wechselwirkung zu analysieren [77].

Über die Einführung der Stationsautomatisierung in Digitalen Stationen wurde zudem im Rahmen der Arbeit bereits definiert, dass Übertragungsprotokolle nach dem Standard IEC 61850 genutzt werden sollen. Dementsprechend sind für die Stromnetzsimulation die Prozessbusprotokolle SV und GOOSE relevant. Dies bedeutet, dass in der Stromnetzsimulation Komponenten zur Erfüllung der Schnittstellen integriert werden müssen. Diese sind demnach Messeinrichtungen, auch MU genannt, die an dedizierten Messpunkten Momentanwerte über das Protokoll IEC 61850-9-2LE SV zu den verarbeitenden Sekundärsystemen weiterleiten und Leistungsschalter, um die IEC 61850-8-1 GOOSE Stellbefehle entgegenzunehmen. Die genannten Kompo-

nenten werden entsprechend dem Zweck der Sekundärsysteme nach im Modell des Umspannwerkes platziert. Die Schutzsysteme benötigen dabei Messinformationen der zugehörigen MUs und besitzen hierzu GOOSE-Schnittstellen zu Leistungsschaltern.

In Umspannwerken werden klassischerweise eine Reihe unterschiedlicher Sekundärsysteme eingesetzt. Neben den bereits genannten Schutzsystemen sind vor allem Regelungs- und Steuerungssysteme verbaut. Die Systeme weisen eine hohe Vielfalt auf und sind in unterschiedlichen Anwendungen einsetzbar. Für den Schutz der Umspannwerke und des elektrotechnischen Prozesses kommen Systeme wie beispielsweise

- Überstrom-,
- Transformator-Differential-,
- Leitungs-Differential- oder
- Sammelschienen-Schutzsysteme

zum Einsatz. Durch neue Anwendungen und zunehmender Digitalisierung nimmt die Anzahl und Diversität der Systeme stetig zu [78].

In der Folge wird die Sekundärtechnik der Umspannwerke als Bindeglied der Co-Simulation dienen und mit dem gemeinsamen Prozessbus verbunden.

5.2 Modellierung der Kommunikationsinfrastruktur

Nachfolgend wird die Modellierung der Kommunikationsinfrastruktur beschrieben. Dabei wird zunächst auf die grundlegenden Umsetzungen zur Virtualisierung eingegangen und anschließend ein Framework zur automatisierten Erstellung der Netzwerkinfrastruktur auf Basis einer Beschreibungsdatei vorgestellt. Die zuvor beschriebene Trennung zwischen Primär- und Sekundärtechnik von Umspannwerken wird hierbei aufgegriffen.

5.2.1 Implementierungen der Komponenten innerhalb der Netzwerkvirtualisierung

Der gewählte Ansatz der Virtualisierung als Emulation des Kommunikationsnetzwerkverhaltens ist auf die Umsetzung der SDN-Infrastruktur inklusive Controller und Switch sowie der Modellierungsumgebung Containernet und den implementierten Sekundärgeräten (vIEDs) abgestimmt. Auf die jeweiligen Details der Implementierung wird nachfolgend eingegangen. [79], [80]

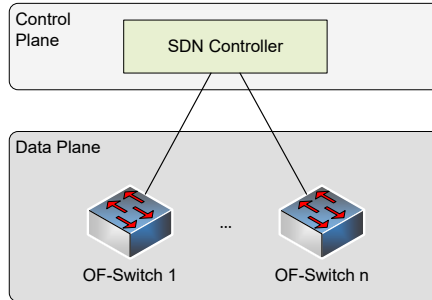


Abbildung 5.1: Single-Controller-Architektur innerhalb der SDN-Infrastruktur

5.2.1.1 SDN-Controller-Implementierung

Maßgeblich für die Implementierungen der Netzwerkvirtualisierung ist die Verwendung des SDN-Controllers, der die Logik der Network Function Virtualization (NFV) realisiert. Wie in Abschnitt 4.2.1.3 eingeführt, wird eine Single-Controller-Architektur verwendet. Ein zentral gehosteter Controller wird auf dem Virtualisierungsrechner gestartet und kommuniziert mit den instanziierten OpenFlow-Switches (OF-Switch) innerhalb von Containernet, wie in Abbildung 5.1 dargestellt. Hierzu sind verschiedene Implementierungen von SDN-Controllern verfügbar, die sich vor allem in der Anwendung unterscheiden. In der Literatur existieren einige Vergleiche der unterschiedlichen Controller, bei denen auf Performanz und Anwendung geachtet wird [81]. Für die Umsetzungen wird der Python-basierte ryu-Controller verwendet, der primär als zentrales System eingesetzt wird. Der ryu-Controller ist OpenSource verfügbar, wodurch er sich anwenderfreundliche anpassen lässt [72]. Die Entscheidung für den Einsatz von ryu basiert vor allem auf der Anpassbarkeit für die jeweiligen Anwendungsfälle. Andere Aspekte, wie die Performanz, sind hierbei nachgelagert.

Innerhalb des Controllers ist die Logik der Datenverarbeitung implementiert. Über Flow-Tabellen werden die sogenannten Ströme (englisch Flows) der Netzwerkpakete registriert und abgelegt, um die Weiterleitung von Paketen zu dokumentieren. Anhand von eigens definierten Kriterien kann ein Netzwerkstrom beschrieben und identifiziert werden. In der Regel werden die Media-Access-Control (MAC)-Adressen des Senders und Empfängers aus der Ethernet-Schicht der Datenpakete extrahiert und anhand der ein- und ausgehenden Switchports identifiziert. In der Folge kann innerhalb der Flow-Tabelle dokumentiert werden, welche Sender-Empfänger-Kombinationen über die jeweiligen Ports des Netzwerkswitches verarbeitet werden.

Ist ein Eintrag zu einer identifizierten Kombination im Switch bereits vorhanden, kann der Switch ohne weitere Controller-Aktivität die Weiterleitung selbst durchführen. Ist kein Eintrag in der Flow-Tabelle hinterlegt, werden alle Ausgangsports des Switches mit der eingehenden Nachricht mittels einer Packet-Out Nachricht geflutet. Ein Flow-Tabelleneintrag wird dann generiert, wenn eine Ziel-MAC-Adresse in dem Netzwerk registriert ist. Dies kann zum Beispiel erreicht werden, indem der Empfänger in einer anderen Nachricht bereits als Sender aufgetreten ist. Durch das gezielte Fluten soll der Zielport für die nächste Verbindung ausgemacht werden, indem auf die geflutete Nachricht eine Antwort bei beispielsweise einem Verbindungsaufbau oder auch Hand-Shake zu erwarten ist. Bei dem nächsten Packet-In-Ereignis sollten sowohl Sender als auch Empfänger für den Controller bekannt sein und es kann eine feste Zuordnung in Form eines neuen Eintrags der Flow-Tabelle erfolgen. Im nächsten Schritt wird dieser innerhalb des Switches mittels der sogenannten Flow-Mod-Nachricht abgespeichert. Folglich beschreibt die Umsetzung des SDN-Controllers die Ausgestaltung der Flow-Tabellen. Das ryu-Projekt liefert bereits einige Beispielimplementierungen für das OpenFlow Protokoll in Version 1.3, das auch in der Validierung verwendet wird.

Der identifizierte Datenverkehr basiert größtenteils auf der Prozessebene der Stationsautomatisierung mit Datenübertragungen nach IEC 61850-8-1 GOOSE und IEC 61850-9-2LE SV auf Multicast-Kommunikation. Eine Multicast-Kommunikation verursacht nach der Standard-Implementierung des ryu-Controllers jeweils eine Anfrage des beteiligten Switches an den Controller. Der Controller muss auf jede Packet-In-Nachricht aufgrund einer identifizierten Multicast-MAC-Adresse mit einer Packet-Out-Nachricht reagieren und den Switch dazu bringen, alle Ausgangsports über die zugehörige Flood-Aktion zu fluten, die in OpenFlow 1.3 definiert ist. Der Grund hierfür ist, dass die fest definierten Ziel-MAC-Adressen, die bei Multicast genutzt werden, nicht als Teilnehmer im Netzwerk registriert werden können, da sie nicht als Sender in Erscheinung treten.

Die Eigenschaft bei jedem Multicast-Paket den Controller anzufragen, würde in einer Überlastung des Controllers münden. Dementsprechend wird in der nachfolgenden Umsetzung die Implementierung des ryu-Controllers angepasst, sodass nicht bei jeder Multicast-Nachricht ein Packet-In-Ereignis generiert wird. Es soll bei der Identifizierung einer Multicast-Nachricht einer bestimmten Sender-MAC Adresse an eine definierte Multicast-Ziel-Adresse ein neuer Eintrag in der Flow-Tabelle angelegt werden. So wird ausgeschlossen, dass ein nachfolgendes Paket wiederum ein Packet-In-Ereignis auslöst, sondern der Switch direkt eine Flutung der Ausgangsports mit der eingehenden Nachricht vornimmt.

Zusammengefasst wird vom verwendeten ryu-Controller eine Flow-Tabelle aus folgenden Elementen erzeugt:

- Sender-MAC-Adresse: Im Ethernet-Header enthalte Information über den Sender des Flows.
- Empfänger-MAC-Adresse: Im Ethernet-Header enthalte Information über den Empfänger des Flows.
- Eingangsport: Virtueller Port des OpenFlow-Switches, an dem das Datenpaket eingeht.
- Switch-ID: Feste Zuordnung zu einem der angeschlossenen OpenFlow-Switches an den zentralen ryu Controller.

Die Arbeitsgruppe IEEE 802.1X [82] definierte einige MAC-Adressbereiche für feste Multicast-Anwendungen. Der Standard IEC 61850 legt in gleicher Weise feste MAC-Adressbereiche für Protokollanwendungen fest, wodurch innerhalb der Nutzung von IEC 61850 auf das Vorhandensein von Multicast-Nachrichten und der verwendeten Anwendung geschlossen werden kann. Die Adressbereiche aus Tabelle 5.1 werden von der IEC 61850 verwendet und nachfolgend für die Controller-Implementierung genutzt.

Tabelle 5.1: MAC-Adressbereiche für die Multicast-Protokollanwendungen nach IEC 61850

Protokollanwendung	MAC-Adressbereich
IEC 61850-8-1 GOOSE	01-0C-CD-01-00-00 . . . 01-0C-CD-01-01-FF
IEC 61850-8-1 GSSE	01-0C-CD-02-00-00 . . . 01-0C-CD-02-01-FF
IEC 61850-9-2 Sampled Values	01-0C-CD-04-00-00 . . . 01-0C-CD-04-01-FF

Die definierten Multicast-Adressen werden zur Optimierung des ryu-Controllers für den Anwendungsfall der Stationsautomatisierung verwendet, um das gezielte Fluten aufgrund von Multicast-Anwendungen ohne wiederholte Controller-Aktivität zu realisieren. Folglich wird die Abfrage nach den definierten Ziel-MAC-Adressen im Controller implementiert, und ein neuer Flow-Tabelleneintrag wird bei Auftreten der Adressbereiche direkt angelegt. Diese Anpassung stellt eine schnelle Verarbeitung der auftreffenden Multicast-Pakete sicher und verhindert die Überlastung des Controllers, da auf dem Prozessbus eine hohe Anzahl von Multicast-Paketen zu erwarten ist. Pro

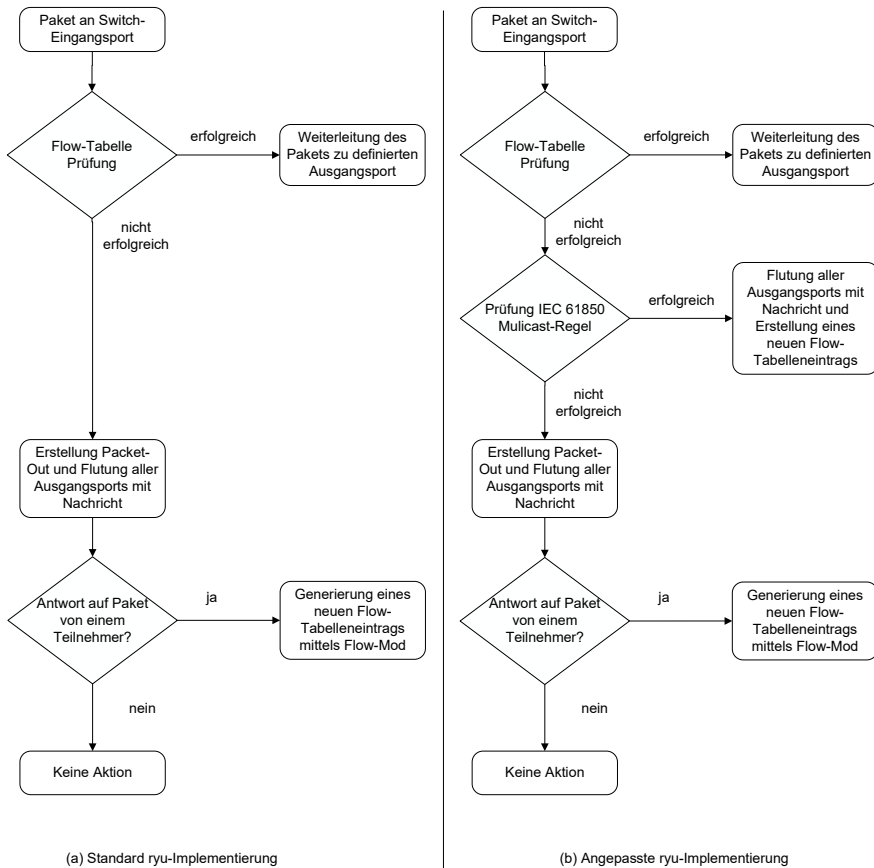


Abbildung 5.2: Ablaufpläne (a) der ryu-Implementierung und (b) der beschriebenen Erweiterung

SV-Publisher würden folglich 4000 Controller Anfragen pro Sekunde erfolgen, die hiermit zu einer einzigen initialen Anfrage reduziert werden kann.

Die vorgenommene Anpassung der ryu-Implementierung ist in Abbildung 5.2 anhand eines Ablaufplans dargestellt. Dabei zeigt (a) den Ablauf bei unbekanntem Nachrichten mit der Standard-ryu-Implementierung, wohingegen (b) die Erweiterung zur Abfrage der neuen Flow-Regel zeigt.

5.2.1.2 OpenFlow-Switch-Implementierung

Innerhalb der Modellierungsumgebung Containernet werden die OpenFlow-Switches initialisiert. Sie können über das OpenFlow 1.3 Protokoll [69] mit fest definierten Befehlen und einem gesicherten Kanal mit dem SDN-Controller kommunizieren. Eine Anpassung der Standardimplementierung ist nicht notwendig, kann aber für zukünftige Erweiterungen in Betracht gezogen werden, um die Verarbeitungsgeschwindigkeit des Switches zu erhöhen. Eine mögliche Implementierung stellt OpenVSwitch dar [83].

5.2.1.3 Containernet-Implementierung

Über das für Containernet zugrundeliegende Mininet können die verwendeten Verbindungen, hier Links genannt, zwischen den virtuellen Ethernet Adapters der Container konfiguriert werden. Mittels der zu definierenden Parameter der verwendeten Link-Implementierung wird beispielsweise die maximale Übertragungsbandbreite konfiguriert. Die Parameter dienen der Abbildung von realen Einschränkungen durch beispielsweise das Übertragungsmedium, die in der Virtualisierung substituiert wird. Die möglichen Anpassungen werden direkt im ausführenden Linux-Kernel adressiert und sind über Parameter in Mininet adaptierbar.

Für die hier vorgenommenen Implementierungen werden keine zusätzlichen Beschränkungen der Links einbezogen, sondern von einer nahezu idealen Virtualisierung ausgegangen, um möglichst wenig negative Beeinflussung der Übertragungszeiten zu erzielen.

5.2.1.4 Implementierung der virtuellen IEDs

In Abschnitt 5.1.2 wurden bereits beispielhafte Schutzsysteme genannt, die als IEDs im Umspannwerk platziert werden können. Deren Funktionalität soll in ein vIED nach dem Konzept aus Abschnitt 4.2.1.2 überführt werden. Die beiden eingeführten Schichten müssen demnach in einem Docker-Container implementiert werden. [84]

Unter Nutzung des IEC 61850 Stacks der Bibliothek libIEC61850 werden die Prozessbus-Schnittstellen verwendet und als Netzwerkschicht implementiert. Die empfangenden Werte der MUs werden dekodiert und können weiterverwendet werden. Die beiden identifizierten Schichten sind logisch zu sehen und werden in der Folge nicht strikt voneinander getrennt implementiert. Im Rahmen der Arbeit und dem Kontext der Co-Simulation wurden verschiedene Implementierungen von Schutzalgorithmen umgesetzt. Beispielhaft wird eine Implementierung eines Überstromzeitschutzes (Protection Time Overcurrent (PTOC)) beschrieben und in Abbildung 5.3 dargestellt.

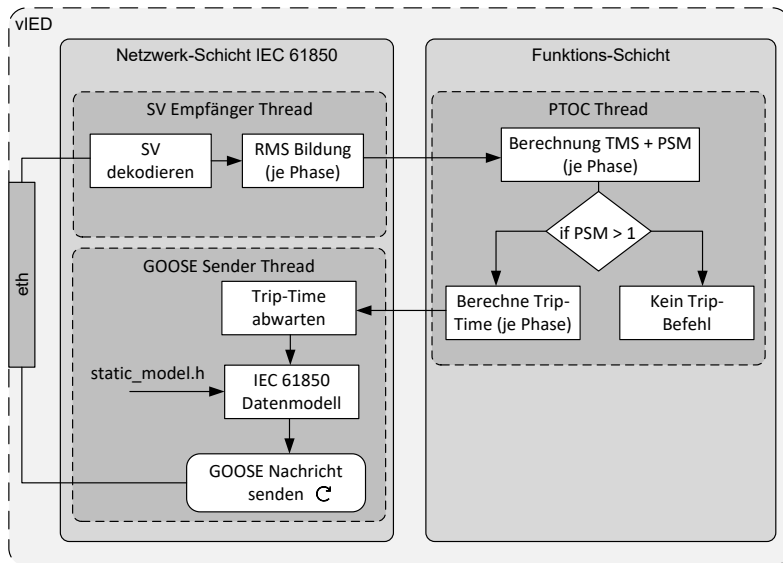


Abbildung 5.3: Implementierter Programmablauf des vIED für den Überstromschutz

Der Überstromschutz basiert auf der Erfassung der Momentanwerte aller drei Phasen und dem Neutralleiter. Diese Werte werden über einen separaten Thread (SV-Empfänger-Thread) mit der libIEC61850 dekodiert. Aus den abgetasteten Momentanwerten werden zunächst je Phase die Mittelwerte als RMS-Werte berechnet. Dieser Mittelwert wird anschließend je Phase an die Funktionsschicht übergeben und in einem eigenen PTOC-Thread weiterverarbeitet. Dabei werden für den Überstromschutz je Phase die sogenannten Plug Setting Multiplier (PSM) und Time Multiplier Setting (TMS) berechnet. Der PSM-Wert gibt an, wie kritisch der Fehler ist. Wenn der PSM-Wert größer als eins ist, gilt der Fehler als kritisch. Der TMS-Wert bestimmt hingegen die Auslösezeit (Trip-Time), also wie lange der Fehler anhalten muss, bis der Auslösebefehl versendet wird. Die genauere Methodik zur Berechnung der Werte wird hier nicht weiter ausgeführt, sie ist beispielsweise an [85], [86] angelehnt.

Falls der berechnete PSM-Wert eins überschreitet, soll eine Schalthandlung initiiert werden und die Trip-Time wird berechnet. Dieser Wert wird an die Netzwerkschicht übergeben. In einem Thread für die GOOSE-Nachrichten wird diese Trip-Time abgewartet und wenn der Fehler weiterhin besteht, wird eine Trip-GOOSE-Nachricht generiert. Hierfür wird wiederum die libIEC61850 verwendet und die Werte des zuge-

hörigen Datensatzes werden im Datenmodell geändert. Im Falle einer zu initiierten Schalthandlung wird das Trip-Attribut im Datensatz von „false“ auf „true“ gesetzt. Für das Datenmodell wird die Header-Datei „static_model.h“ der C-Bibliothek libIEC61850 verwendet und über einen Modellgenerator das C-Modell generiert. Ein Sendeservice greift nun auf das geänderte Datenmodell zurück und schickt die neue GOOSE-Nachricht.

Die GOOSE-Nachrichten werden standardmäßig zyklisch gesendet, also auch im Falle, dass das Trip-Attribut weiterhin den Wert „false“ besitzt. Die Zykluszeit kann variabel gesetzt werden. Bei einer Schalthandlung und der Änderung des Wertes des Trip-Attributes wird ein sogenannter Burst-Mechanismus angestoßen, der die GOOSE-Nachrichten in kürzeren Abständen sendet [87]. Die Umsetzung des Mechanismus wird implizit von der libIEC61850 bereitgestellt und hier genutzt. Die erstellten GOOSE-Nachrichten werden dann über die definierten Adressangaben über die Netzwerkschnittstelle versandt und anderen IEDs und Schaltern zur Verfügung gestellt.

In gleicher Art können unterschiedliche Funktionen implementiert werden und in Verbindung mit der Netzwerkschicht als vIEDs aufgebaut werden. Die Implementierung von vereinfachten Funktionen bis hin zu sehr detaillierten Abbildungen wird ermöglicht. Bei dem beschriebenen vIED mit PTOC-Funktion ist ersichtlich, dass lediglich die Prozessfunktion mitsamt der Prozessbusprotokolle umgesetzt ist. Andere Aspekte, die unter Umständen wesentlich für ein bestimmtes Gerät sind, wie Konfigurationsoberflächen oder weitere Services, werden hier vernachlässigt. Jedoch sind sie im Bedarfsfall zusätzlich implementierbar. Die Synchronisierung mit einer möglichen PTP-Implementierung, wie in Abschnitt 3.3.1 beschrieben, wird hier nicht berücksichtigt, aufgrund der impliziten Synchronisierung des Containers zum Host-System. Hierauf wird weiter in Kapitel 6 eingegangen.[84]

5.2.2 Automatisierte Modellierung von Umspannwerken

Neben der eingeführten und verbreiteten Modellierung des Stromnetzes inklusive veröffentlichter Referenznetze ist die Kommunikationsebene von Digitalen Stationen weniger im Fokus der Betrachtung in Bezug auf simulative Nachbildungen. Die Infrastruktur ist nur indirekt über die jeweilige dokumentierte Implementierung beschrieben. Unter durchgängiger Verwendung des IEC 61850 Standards wird für den Datenaustausch innerhalb des Kommunikationsnetzes ein gesamtheitliches Datenmodell in der SCD-Beschreibungsdatei erzeugt. Um ein virtuelles Abbild der Struktur zu erzeugen, wird nachfolgend das eigens entwickelte Framework *VirtualSubstation*

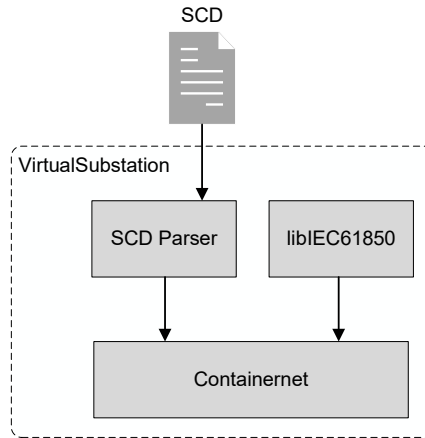


Abbildung 5.4: Workflow des formulierten Frameworks

eingeführt, das die Überführung der realen IEC 61850-Struktur in die Virtualisierung beschreibt.

Es wird vorausgesetzt, dass sämtliche in dem Umspannwerk enthaltene IEDs mit samt deren Kommunikationsservices und Datenobjekten aufgeführt sind. Die SCD-Datei wird in der Regel im Engineering-Prozess von Anlagen erstellt, um die Kommunikationsverbindungen zwischen den Geräten aufzubauen und den Datenaustausch auf Basis der hinterlegten Datenobjekte und Attribute zu beschreiben. Nutzt man nun die SCD-Datei als Beschreibung zur Rekonstruktion in einer Virtualisierungsplattform, erhält man sämtlich relevante Informationen. Darunter zählen neben den Kommunikationsservices auch relevante Konfigurationsinformationen der genutzten Netzwerkadapter und deren Benennung. Der funktionelle Teil für den logischen Aufbau des Netzwerks lässt sich folglich komplett nachbilden. Die notwendigen Schritte der automatisierten Modellierung werden nachfolgend über einen definierten Workflow beschrieben. [88]

5.2.2.1 Workflow der automatisierten Modellierung

Für das Ziel eine dokumentierte IEC 61850 Kommunikationsinfrastruktur zu rekonstruieren, wird als Ausgangsbasis die SCD-Beschreibung verwendet, die nach Abbildung 5.4 die Eingangsgröße für das Framework darstellt. Das Framework beinhaltet die anschließend näher eingeführten drei Teilkomponenten. Die SCD-Beschreibung

geht direkt an einen entwickelten SCD-Parser, der die relevanten Informationen extrahiert und an die nachgelagerte Umsetzung des Containernet-Tools gibt. Das Framework verwendet das vorgestellte Tool Containernet und konfiguriert automatisiert die Netzwerkimplementierung über eine Python-Containernet-Datei. Zusätzlich wird die Bibliothek libIEC61850 als Kommunikationsstack genutzt und ebenfalls an die Containernet-Implementierung übergeben, um angepasste Kommunikationsservices nach der SCD-Beschreibung automatisiert innerhalb der Docker-Container zu instanzieren.

5.2.2.2 SCD-Parser

Basierend auf der SCD-Datei werden alle relevanten Informationen für die Kommunikationsvirtualisierung extrahiert und intern gespeichert. Insbesondere die Entitäten für die Station, die in der SCD als Connected-Access-Points bezeichnet werden, sind entsprechend Abbildung 5.5 zu ermitteln. Es wird sich auf die Modellierung der IEDs beschränkt, da sie aktiv im Rahmen der Netzwerkkommunikation verwendet werden. Der Abschnitt „Kommunikation“ der SCD-Datei definiert die relevanten Informationen der IEDs. Dazu gehören unter anderem Informationen zu:

- Internet Protokoll (IP)-Adressen,
- MAC-Adressen,
- Subnetz- und
- Virtual Local Area Networks (VLAN).

Diese Informationen werden an die Containernet-Implementierung übergeben, um die IEDs zu instanzieren und ihnen die Netzwerkparameter mitzugeben. Neben der reinen Struktur der Container sind auch die Kommunikationsdienste sehr wichtig. Sie sind in den SCD-Dateien vollständig beschrieben. Für die weitere Verarbeitung durch die Bibliothek libIEC61850 werden einzelne ICD-Dateien der jeweiligen IEDs aus der SCD-Datei erstellt und vom Parser bereitgestellt.

5.2.2.3 Bibliothek libIEC61850

Entscheidend für das vorgestellte Framework ist die Verwendung der OpenSource Bibliothek libIEC61850 als Kommunikationsstack. Für die Initialisierung und Implementierung der IEC 61850-Kommunikationsdienste mit libIEC61850 werden die

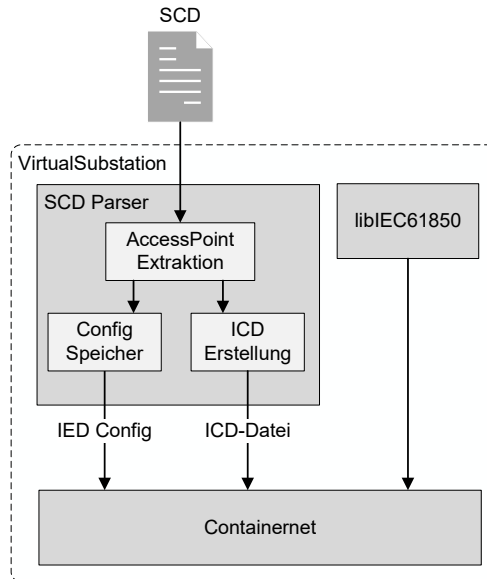


Abbildung 5.5: Workflow des formulierten SCD-Parsers

spezifischen ICD- und Konfigurationsdateien für jedes einzelne IED benötigt. Der verfügbare Modellgenerator wird verwendet, um die Datensätze und Attribute für die jeweiligen IEDs zu initialisieren. Dazu verwendet der Modellgenerator die ICD-Datei und erzeugt ein statisches C-Modell, in dem die zugehörigen Informationen implementiert und anschließend verwendet werden. Für den vorliegenden Ansatz sind die statisch beschriebenen Dienste MMS und GOOSE nach IEC 61850-8-1 von Bedeutung. Die IED-Beschreibung definiert die bereitgestellten Kommunikationsdienste, die über die Bibliothek libIEC61850 erstellt und für weitere Implementierungen zur Verfügung gestellt werden. Diese sind vor allem für die spätere Programmierung notwendig. Hier wird zunächst auf die reine Instanziierung und automatisierte Nutzung von Standarddiensten nach IEC 61850 verwiesen, die von der Bibliothek libIEC61850 zur Verfügung gestellt werden. Für nachfolgende beispielhafte MMS-Services werden die Initiate-Request- und Initiate-Response-Sequenzen für den MMS-Datenaustausch generiert, um anschließend manuelle Client-Abfragen implementieren zu können, auf die der Server regulär antwortet. Die Implementierung nach der Bibliothek libIEC61850 ermöglicht eine Standard-MMS-Server-Konfiguration, die bereits eine Reihe von MMS-

Services bereitstellt und auf adäquate Anfragen der Clients reagiert. Eine Erweiterung und Individualisierung werden zunächst nicht angestrebt. Die SCD- und ICD-Dateien enthalten die in Bezug auf GOOSE bereitgestellten Publisher-Services, die unter Verwendung der Multicast-Eigenschaft als GOOSE-Server-Implementierungen realisiert werden. Auch hier wird ein Mapping auf einen festen Datensatz vorgenommen, der zyklisch gesendet wird.

Eine Publisher-Implementierung von IEC 61850-9-2 SV ist nicht erforderlich, da die Messwerte im Zielbild der Co-Simulation aus Abbildung 4.7 innerhalb der Stromnetzsimulation erzeugt und den IEDs zur Verfügung gestellt werden. Die Generierung von Abtastwerten in der Funktion einer MU ist über die ebenfalls verfügbaren IEC 61850-9-2LE-Implementierungen der libIEC61850 möglich, wird hier jedoch nicht vorgesehen.

5.2.2.4 Implementierung der virtuellen Netzwerkinfrastruktur

Containernet: Für die Implementierung der Netzwerkinfrastruktur wird das Tool Containernet verwendet, das eine Abspaltung von Mininet [75] darstellt [76]. Die Ausführung des Tools wird erweitert, um die notwendigen Container gemäß Abbildung 5.6 zu erstellen und einzubinden. Dazu werden die vom Parser extrahierten Informationen und ICD-Dateien mitsamt einer Dockerfile-Vorlage und dem Tool Docker-Compose verwendet. Darauf basierend wird das abstrahierte Netzwerk automatisiert gestartet und beispielhafte Kommunikationsverbindungen nach IEC 61850 aufgebaut.

Dockerfile, Docker-Compose und Docker Image: Die erstellte Dockerfile-Vorlage für das Framework enthält ein minimales Python-fähiges Basis-Image. Zusätzlich werden wesentliche Kommunikationspakete für Containernet installiert, um die Kommunikation über virtuelle Ethernet-Adapter zu realisieren und zu testen. Konkret wird für die Implementierung die Bibliothek libIEC61850 inklusive der generierten ICD-Datei für die jeweiligen IEDs genutzt. Der Modellgenerator wird in Verbindung mit dem Build-Prozess des Images ausgeführt, um das erforderliche statische Modell zu erzeugen. Die extrahierten Informationen für das entsprechende IED werden gespeichert und konfiguriert, um den zu bauenden Container eines physischen IEDs so ähnlich wie möglich zu gestalten. Darüber hinaus werden die erforderlichen Kommunikationsdienste durch ein automatisch ausführbares Python-Skript konfiguriert, das auf dem Basis-Image im Dockerfile definiert wird. So wird ein Skript vorbereitet und in den Container kopiert, das einen MMS-Server-Dienst aus der libIEC61850 unter Verwendung des generierten statischen Modells startet. In gleicher Weise wird, wenn

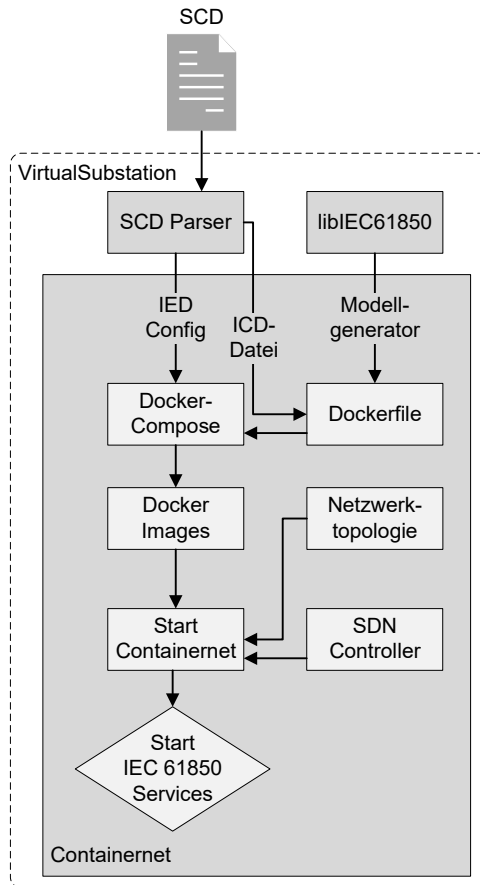


Abbildung 5.6: Workflow der Containernet-Implementierung

die GOOSE-Dienste verfügbar sind, ein Startskript des GOOSE-Servers als Publisher konfiguriert und bereitgestellt. Bei der initialen Implementierung werden detaillierte Funktionsprogrammierungen vermieden. Diese können jedoch in Abhängigkeit von den logischen Knoten als Vorlagen angelegt und bei der Image-Erstellung berücksichtigt werden.

Der nächste Schritt, wie in Abbildung 5.6 skizziert, ist die Erstellung der Docker-Compose-Datei. Diese benötigt wiederum die extrahierten Informationen aus der SCD-Datei. Die zuvor definierte Dockerfile-Vorlage wird verwendet, um das Docker-

Image für jedes einzelne IED zu erstellen. In diesem Schritt wird nur die Build-Option von Docker-Compose verwendet und noch keine Container gestartet. Es werden lediglich die Build- und Initialisierungsoperationen durchgeführt und die einzelnen IED-Container erstellt, die dem System für spätere Prozesse zur Verfügung stehen. Die Benennung der IEDs wird auch für die Images genutzt, um insbesondere die Zuordnung der ICDs und die anschließende Initialisierung der Dienste sicherzustellen. Am Ende dieses Schrittes hat das System für jedes identifizierte IED ein eigenes Docker-Image inklusive der Bibliothek libIEC61850 und der konfigurierten Funktions- und Kommunikationstemplates, die an das eigentliche Containernet-Skript übergeben werden.

Netzwerktopologie: Um die Virtualisierung eines physikalischen Netzes mit Containernet zu realisieren, kann die Netzwerkinfrastruktur auf verschiedene Weisen konfiguriert werden. Um ein reales Umspannwerk entsprechend seiner SCD-Datei abbilden zu können, sollte die identische Netzwerktopologie mit der Anbindung der einzelnen IEDs über die Netzwerkkomponenten, wie beispielsweise Switches und Router, übernommen werden. Die SCD-Dateien enthalten jedoch keine dedizierten Informationen über die physikalische Netzwerkstruktur. Lediglich die logischen Kommunikationsbeziehungen sind hier beschrieben und können vom Framework genutzt werden. Für den vorliegenden Ansatz wird zunächst von einer vollständigen Sterntopologie des Netzwerks ausgegangen, um die Kommunikation zwischen allen IEDs gewährleisten zu können. Das Protokoll IEC 61850 verlangt die Unterscheidung zwischen Prozess- und Stationsbus, die beispielsweise jeweils eine Sterntopologie aufweisen können. Zunächst wird diese Topologie durchgängig verwendet.

SDN-Controller: Containernet basiert auf der Verwendung von NFV, die den Einsatz von SDN-Systemen vorsieht. Das virtuelle Netzwerk wird also durch einen SDN-Controller aufgebaut, der die Flow-Tabellen für die OpenVSwitch-Implementierungen im Containernet bereitstellt. Für den Aufbau des Netzwerks ist die Wahl des Controllers und dessen Implementierung von großer Relevanz. Dies ist insbesondere für die angestrebte Kopplung an eine Echtzeit-Stromnetzsimulation über den Prozessbus mit Echtzeitbedingungen interessant. Die in Abschnitt 5.2.1.1 beschriebene ryu-Implementierung findet hier Anwendung und ermöglicht die Abbildung des virtuellen Netzwerkes.

Containernet und libIEC61850-Initialisierung: Die Kernkomponente der Containernet-Implementierung innerhalb des Frameworks ist das Containernet-Skript, das

die Initialisierung des Netzwerks und den Start der Docker-Container übernimmt. Die zuvor erstellten Docker-Images werden in der definierten Netzwerktopologie instanziiert und über Switches und den SDN-Controller miteinander verbunden. Containernet erstellt zusätzlich zu den, von Docker bereitgestellten Netzwerkadaptern weitere Netzwerk-Schnittstellen, denen die aus der SCD extrahierten Konfigurationen zugewiesen werden. Auf diese Weise wird eine Zuordnung der extrahierten IP- und MAC-Adressen vorgenommen, um eine Verbindung über das virtuelle Netzwerk herzustellen. Die Container werden anschließend in der definierten Topologie angeordnet und können einander erreichen. Um schließlich den gewünschten IEC 61850-Kommunikationsverkehr automatisch zu generieren, werden die implementierten Kommunikationsskripte für MMS und GOOSE über die jeweiligen Container gestartet. Zusätzliche Dienste können auch manuell über die erreichbaren Kommandozeilen der einzelnen Container bedient und getestet werden. [88]

5.3 Kopplung der Simulatoren auf dem Prozessbus

Die beiden beschriebenen Simulatoren werden im Rahmen der Co-Simulation miteinander gekoppelt. Dabei wird, wie in Abschnitt 4.3 eingeführt, der Prozessbus als Kopplungspunkt ausgemacht. Der Prozessbus wird in der Regel in den IEC 61850-Umspannwerken durch einen Netzwerkswitch umgesetzt, an dem sämtliche Geräte direkt angeschlossen sind, um Daten auszutauschen. Dementsprechend gilt für die Co-Simulation, dass der Prozessbus die MUs und Schalter innerhalb der Stromnetzsimulation mit den vIEDs innerhalb der Kommunikationsnetzwerkvirtualisierung verbindet. Der Prozessbus wird durch einen virtuellen OpenFlow-Switch innerhalb der Virtualisierung realisiert. Die Kopplung der beiden Simulatoren ermöglicht dem Simulator der Stromnetzsimulation seine Daten in den virtuellen OpenFlow-Switch einzuspeisen und Daten hiervon zu erhalten. Der Datenaustausch zwischen den Simulatoren wird in diesem Fall über eine direkte Netzwerkverbindung ohne den Weg über den Koordinator umgesetzt, um eine maximale Datenaustauschgeschwindigkeit entsprechend der Echtzeitanforderungen zu gewährleisten.

Neben dem Datenaustausch beschreibt der zweite Aspekt der Kopplung die Synchronisierung der beiden Simulatoren, wie es in Abbildung 4.6 dargestellt ist. Demnach verbinden sich beide Simulatoren zu einem PTP-Master unter Verwendung des zugehörigen Protokolls. Die Verbindung wird über separate Netzwerkadapter aufgebaut, die explizit nicht für die Kopplung der Simulatoren genutzt wird. Die Umset-

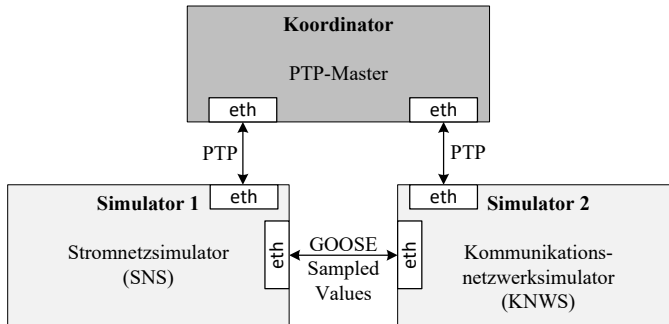


Abbildung 5.7: Umsetzung der in Abschnitt 4.3 eingeführten Kopplung der Simulatoren

zung der Kopplung ist schematisch in Abbildung 5.7 skizziert und zeigt jeweils die zwei Netzwerkschnittstellen der Simulatoren und die dabei verwendeten Protokolle.

Aus der dargestellten Kopplung werden also technische Anforderungen von zwei getrennten Netzwerk-Schnittstellen deutlich, wodurch eine Trennung des Netzwerkverkehrs durch den Datenaustausch und der Synchronisierung erzielt wird. Demnach sind die vIEDs nicht direkt mit dem PTP-Master verbunden und es kann auf eine eigenständige Implementierung der Synchronisierung innerhalb der vIEDs verzichtet werden. Die zugrundeliegenden Container beziehen die Synchronitätseigenschaften von dem Simulator und dessen Synchronisierung.

Der virtuelle Prozessbus-Switch verteilt die aufkommenden Daten an die unterschiedlichen Teilnehmer. Die Charakteristik der Publish-Subscriber-Protokolle SV und GOOSE erleichtert die Routing-Aufgaben des OpenFlow-Switches, indem Multicast-Vorgaben umgesetzt werden, um eingehende Nachrichten von einem virtuellen Netzwerkport auf andere auszugeben. Eine genaue Beschreibung der Routing-Prozesse im OpenFlow-Switch wird im Rahmen der Modellierung der Verzögerungszeit in Abschnitt 5.4 vorgestellt. Der Datenaustausch erfordert Verbindungen zwischen den einzelnen vIEDs und dem OpenFlow-Switch sowie die Implementierung zur Verbindung des Hardware-Netzwerkadapters des Simulators, der für den Datenaustausch genutzt wird. Dieser Netzwerkadapter wird im Simulator mithilfe der Schnittstellen-Implementierung von Containernet auf einen virtuellen Netzwerkadapter gespiegelt, der von Containernet genutzt werden kann. Zudem wird ein Link zum OpenFlow-Switch hergestellt, um den Datenaustausch zum Prozessbus zu rea-

lisieren. Für die Umsetzung im Rahmen der Co-Simulation wird auf ein Beispiel aus dem Mininet-Projekt zurückgegriffen.

Abgeleitet kann aus der beschriebenen Umsetzung der Datenaustausch über den virtuellen Prozessbus zwischen den verschiedenen Endgeräten Simulatoren-übergreifend realisiert werden. Die eingeführten Prozessbus-Protokolle können ausgetauscht werden und in den beiden Simulatoren weiterverwendet und verarbeitet werden.

5.4 Modellierung der Verzögerungszeit des Gesamtansatzes

Für die eingeführten Anforderungen an die Co-Simulation und dem Anwendungsfall der Digitalen Stationen mit der Abbildung der Stationsautomatisierung ist ersichtlich, dass die Übertragungszeit von großer Bedeutung ist. Dementsprechend hat die eingeführte Virtualisierung einen maßgeblichen Einfluss auf diese, da nach Abbildung 3.3 auf dem Weg zwischen den beiden physischen Geräten die Strukturen und Vorgänge des virtuell nachgebildeten Netzwerkes hinzukommen. Der Einfluss und die Größe der Verzögerungen des SDN-Netzwerkes sollen in der Folge über ein Verzögerungsmodell abgeschätzt und mathematisch mittels der Warteschlangen-Modellierung beschrieben werden. [89]

In einer SDN-Infrastruktur ist vornehmlich die Kommunikation zwischen Switch und Controller von Interesse. Sie tritt auf, wenn ein unbekannter Netzwerkstrom (Flow) vom Switch identifiziert und eine Anfrage an den Controller sendet. Im Allgemeinen wird die Kommunikation zwischen Switch und Controller gemäß [70] und [90] in mehrere Phasen unterteilt, wenn ein Flow zum ersten Mal auftritt und noch kein zugehöriger Eintrag in den Flow-Tabellen existiert. Nachfolgend ist der Verarbeitungsprozess eines eintreffenden Pakets in einem SDN-Netzwerk anhand von sieben Schritten beschrieben. Eine schematische Abbildung der zu durchlaufenden Schritte ist in Abbildung 5.8 dargestellt und soll die Identifikation von Verzögerungen innerhalb der Verarbeitungskette unterstützen. Folgende Schritte umfasst die Verarbeitungskette:

1. Ein Paket wird an den OpenFlow-Switch gesendet und kommt in dessen Eingangswarteschlange, wodurch eine Wartezeit resultiert.
2. Der Switch vergleicht den Header des Pakets mit den bekannten Informationen aus seiner eigenen Flow-Tabelle. Zusammen mit Schritt 1 ergibt sich die von einem Switch benötigte Berechnungszeit als Kehrwert der Servicerate.

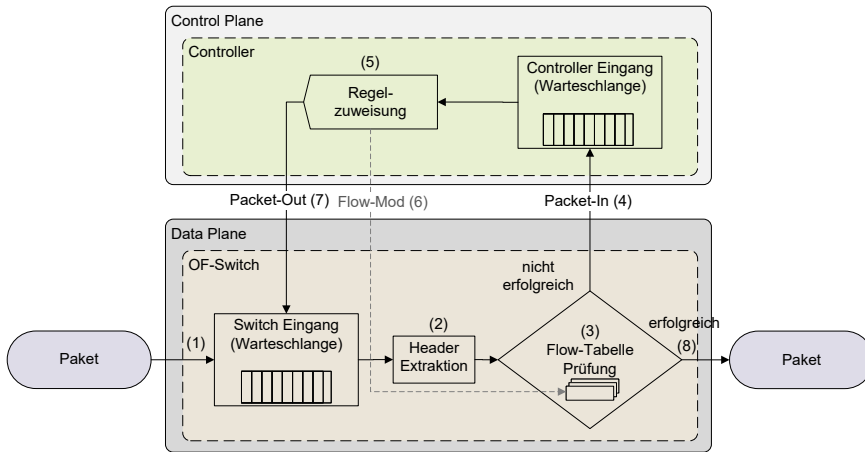


Abbildung 5.8: Relevante SDN-Prozesse innerhalb der Co-Simulation

3. Wenn noch keine passende Flow-Regel existiert, wird vom Switch eine Packet-In-Nachricht erstellt. Die Packet-In-Nachricht wird über einen OpenFlow-Kanal an den Controller gesendet. Die Packet-In-Nachricht wird an die Eingangswarteschlange des Controllers weitergeleitet und wartet auf ihre Verarbeitung.
4. Der Controller liest die Packet-In-Nachricht und erstellt die Regelsatzanpassung (Flow-Mod). Die Packet-In-Nachricht wird verarbeitet und die ausgehenden Nachrichten werden erstellt. Dies geschieht innerhalb der Berechnungszeit des Controllers, die durch die Servicerate definiert ist.
5. Der Controller aktualisiert die Flow-Tabellen der Switches mittels Flow-Mod. Die Switches empfangen die Flow-Mod-Nachricht über den OpenFlow-Kanal und aktualisieren ihre eigenen Flow-Tabellen.
6. Parallel dazu erstellt der Controller eine Packet-Out-Nachricht, die an die Eingangswarteschlange des OpenFlow-Switches mit höherer Priorität zurückgegeben wird. Dieses Paket wird nun vom Switch auf die gleiche Weise wie in Schritt 3 weiterverarbeitet [91].
7. Das Paket kann nun unter Verwendung der neu erstellten Flow-Regel erfolgreich weitergeleitet werden. Alle nachfolgenden Pakete des Flows werden direkt auf der Grundlage der Weiterleitungsregel behandelt, und es ist keine weitere Aktion der Steuerungsebene erforderlich.

Die Verzögerung, die sich aus dem SDN-Ansatz ergibt, besteht zum einen aus einer festen Komponente der Übertragung zwischen der Daten- und Steuerungsebene. Zum anderen gibt es eine variable Komponente, die sich aus den Berechnungszeiten der jeweiligen Instanzen (Switch und Controller) ergeben. Diese Verzögerungszeit ist als Variable zu definieren, da sie vom aktuellen Zustand, wie beispielsweise der Auslastung der Warteschlange, abhängt [90]. Die SDN-Verzögerung resultiert primär aus dem etablierten Prozess zwischen Switch und Controller, falls ein Datenstrom neu ist und noch nicht in der Flow-Tabelle des Switches gespeichert ist. Hier findet eine Verarbeitung beider Komponenten statt. Von daher ist eine Abschätzung dieser Verzögerungszeit für den gewählten Ansatz der Co-Simulation sehr wichtig und wird im Folgenden näher untersucht. Ziel ist es, eine Aussage über die auftretende Verzögerungszeit treffen zu können, die sich aus der Struktur und realisierten Implementierungen der Co-Simulation ergibt. Für die Modellierung werden vergleichbare Literaturstellen herangezogen.

Auf der Grundlage der SDN-Struktur werden Warteschlangenmodelle für den Erwartungswert der Verzögerung $E(\Delta T)$ eines OpenFlow-Switches und eines Controllers aufgestellt. Klassische Warteschlangenmodelle basieren auf einer Systembeschreibung, die eine Verteilung von Eingangsgröße, Servicezeit und Anzahl der Server erfordert. Panev [70] wählt eine Warteschlangenmodellierung für eine Single-Controller-Topologie, bei der eine separate Modellierung des Controllers und des Switches vorgesehen ist. Für beide werden getrennte Warteschlangen definiert, wobei global angenommen wird, dass jeder Switch die gleiche Servicerate hat. Eine weitere Einschränkung von Panev ist, dass die Puffergröße des Switches und des Controllers als unendlich groß definiert wird. Die Servicerate wird als unabhängig von der Arbeitslast angenommen. Diese Randbedingungen stellen eine Idealisierung der Rechenressourcen der Hardwareplattform dar und können hier in gleicher Weise für eine erste Betrachtung genutzt werden.

Für die Modellierung der Eingangsverteilungen bei SDN-Ansätzen werden häufig Poisson-Verteilungen verwendet [70], wobei die bereits erwähnten Einschränkungen einer Single-Controller-Architektur berücksichtigt werden. Oftmals wird ein M/M/1-System definiert. Demnach werden die Markov-Verteilungen (M) sowohl für den Ankunfts- als auch den Bearbeitungsprozess für eine einzelne Bearbeitungseinheit (1) angenommen. Leland [92] hingegen sagt, dass ein Ethernet-basiertes Netzwerk mit einer Gleichverteilung (G) genauer modelliert werden kann, was auf ein M/G/1-System nach Iqbal [18] hindeutet. Für die beiden Modelle des Switches (S) und des Controllers (C) müssen unterschiedliche interne Prozesse berücksichtigt und in das

entsprechende Warteschlangenmodell integriert werden. In der folgenden Notation nach Panev werden n Switches mit dem Index i vom gleichen Typ angenommen. Die mathematische Modellierung der Erwartungswerte der Verzögerungen vom Switch $E(\Delta T_S)$ und Controller $E(\Delta T_C)$ nach [70] ergeben sich somit zu:

$$E(\Delta T_S) = \frac{\sum_{i=1}^n \left(\Lambda_i \cdot \frac{1}{\mu_i - \Lambda_i} \right)}{\sum_{i=1}^n \Lambda_i} \quad (5.1)$$

$$E(\Delta T_C) = \frac{1}{\mu_C - \sum_{i=1}^n q_i^{pc} \lambda_i} \quad (5.2)$$

mit:

λ_i	Ankunftsrate,
μ_i	Servicerate eines Switches,
μ_C	Servicerate des Controllers,
q_i^{pc}	Wahrscheinlichkeit eines Packet-In-Ereignisses,
Λ_i	Gesamtankunftsrate von Paketen am Switch i .

Die Darstellung der Formel (5.1) ist im Vergleich zur Veröffentlichung von Panev [70] leicht angepasst und in Bezug auf die Laufvariablen korrigiert. Die Beschreibung der durchschnittlichen Teilnehmer in der Warteschlange dient hier als Grundlage, aus der sich nach dem Little'schen Gesetz eine durchschnittliche Wartezeit in Abhängigkeit von der Ankunftsrate ableiten lässt, wobei ein stationärer Zustand angenommen wird. Das Little'sche Gesetz beschreibt den Zusammenhang zwischen der durchschnittlichen Anzahl von Elementen in einem System, der durchschnittlichen Ankunftsrate von Elementen und der durchschnittlichen Zeit, die ein Element im System verbringt. Unter Einbeziehung der Servicerate nach dem Jackson-Theorem ergibt sich die durchschnittliche Verweildauer innerhalb der jeweiligen Komponente. Das Jackson-Theorem gibt an, dass die Servicerate gleich der Anzahl der bedienten Anfragen pro Zeiteinheit ist. Um ein System mit mehreren Switches zu beschreiben, wird für Formel (5.1) ein über die Ankunftsraten gewichteter Mittelwert angesetzt. [70]

Bei einem Packet-in-Ereignis für das Auftreten eines unbekanntes Flows ergibt sich $E(\Delta T_i^{tot})$ als die durchschnittliche Interaktionszeit zwischen einem Switch i und dem Controller C zu:

$$E(\Delta T_i^{tot}) = E(\Delta T_{S_i}) + E(\Delta T_C) + E(\Delta T_{i_{prop}}^C) + E(\Delta T_{i_{prop}}^{Cmax}) \quad (5.3)$$

mit:

- $E(\Delta T_{S_i})$ Erwartungswert der Verzögerung am Switch i ,
- $E(\Delta T_C)$ Erwartungswert der Verzögerung am Controller,
- $E(\Delta T_{i_{prop}}^C)$ Erwartungswert der Ausbreitungszeit von Switch i zum Controller,
- $E(\Delta T_{i_{prop}}^{Cmax})$ Erwartungswert der maximalen Ausbreitungsverzögerung einer Flow-Mod-Nachricht.

Zhang [90] beschreibt zusätzlich die Differenzierung für das Multiple-Switch-Szenario, das dadurch gekennzeichnet ist, dass die Ausbreitungszeit der Flow-Mod-Nachricht zwischen den Switches unterschiedlich ist. Dies ist insbesondere bei den untersuchten verteilten mobilen Systemen notwendig. Die in dieser Arbeit neu entworfene Co-Simulation arbeitet hingegen zentral auf einem Host und führt zu der Vereinfachung, dass die Ausbreitungszeit immer gleich ist und nicht zwischen den verschiedenen Switches variiert. Somit kann diese als eine feste Verzögerungskomponente festgelegt werden. Es wird zudem angenommen, dass der Weg vom Switch zum Controller die gleiche Zeit benötigt wie der umgekehrte Weg.

Das Modell beschreibt den Fall, dass das Paket noch nicht in der Flow-Tabelle gespeichert ist. Existiert ein Eintrag, erfolgt keine Kommunikation zum Controller, und der Weg verkürzt sich auf die durchschnittliche Verweildauer innerhalb des Switches $E(\Delta T_S)$. Das Modell dient dazu, die Verzögerung bei normalem Verhalten abzuschätzen, um eine Überlastung der Switches und des Controllers zu vermeiden. Der Pfad berücksichtigt den Einfluss des Controllers sowie die Wahrscheinlichkeit, dass ein Paket zum Controller gesendet werden muss, um die mittlere Auslastung der Controller-Warteschlange genau zu modellieren. Die mittlere Verweildauer des Systems hängt somit von der Ankunftsrate (Datenrate im Netz), der Anzahl der Switches und der Wahrscheinlichkeit von Packet-In-Nachrichten ab.

Direkt ersichtlich aus den Gleichungen (5.1) und (5.2) ist der Gültigkeitsbereich des Modells mit der Annahme des Little'schen Gesetzes, wenn sich das System in einem stabilen Zustand befindet. Daraus folgt, dass das System nur dann stabil ist, wenn die Servicerate größer ist als die Ankunftsrate am Switch und die wahrscheinlichkeitsgewichtete Ankunftsrate am Controller. Wenn die Ankunftsrate die Servicerate übersteigt, ist die Annahme nach Little nicht mehr gültig und das System geht in einen instabilen Zustand über. Das System divergiert in dem Bereich, in dem Service-rate und Ankunftsrate nahezu identisch sind. Die Randbedingungen nach [70] werden

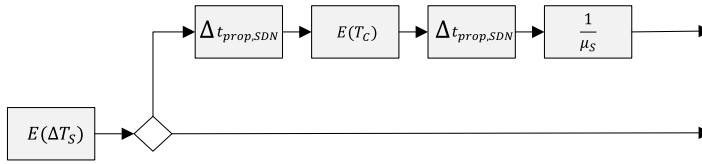


Abbildung 5.9: Sequenzdiagramm des Vorwärtspfads von Paketen

dahingehend bewertet, dass begrenzte Puffer einen signifikanten Einfluss auf die Verzögerung haben, insbesondere bei hoher Auslastung der Switches und Controller. Andererseits hat die Modellierung des Switches als einzelne Warteschlange keine großen Auswirkungen. Folglich muss die gewählte Implementierung des SDN-Ansatzes als Einflussgröße in einer realen Anwendung berücksichtigt werden, beispielsweise durch die Auswahl der Controller-Implementierung.

Als Grundlage wird das SDN-Verzögerungs-Modell von Panev verwendet, das in [93] herangezogen wird. Es wird das Single-Controller-Szenario mit n identischen OpenFlow-Switches angenommen, was bedeutet, dass die Ankunfts- und Serviceraten der Switches gleich sind. Dadurch ist die Wahrscheinlichkeit, dass ein Switch ein Packet-In-Ereignis auslöst, ebenfalls identisch. Die Beschränkung auf eine Single-Controller-Topologie ist gültig, da der Containernet-basierte Ansatz der Co-Simulation auf einem zentralen Controller basiert, der mit verschiedenen OpenFlow-Switches auf der Datenebene verbunden ist.

Jarschel [19] wählt eine ähnliche Darstellung, berücksichtigt jedoch nicht die erhöhte Priorität in der Eingangswarteschlange des Switches sowie die Übertragungszeit zwischen Switch und Controller. Das Modell von Panev wird um die Verarbeitung von Paketen durch den Controller und deren Rücksendung an den Switch erweitert. Hier hat das Paket eine erhöhte Priorität in der Eingangswarteschlange und kann diese direkt wieder verlassen. Es wird erneut vom Switch verarbeitet, wobei die Berechnungszeit aus der angenommenen Servicerate des Switches resultiert. Das resultierende Sequenzdiagramm zur Verarbeitung der Pakete ist in Abbildung 5.9 dargestellt. Zudem wird $\Delta t_{prop,SDN}$ als Ausbreitungsverzögerung ergänzt, die sich durch den Übergang zwischen zwei SDN-Komponenten auf dem Rechner ergibt.

Zusammengeführt ergibt sich die hier eingestellte erwartete Verzögerung der gesamten SDN-Infrastruktur $E(\Delta T_{SDN})$ zu:

$$\begin{aligned} E(\Delta T_{SDN}) &= E(\Delta T_S) + E(\Delta T_C) + 2\Delta t_{prop,SDN} + \frac{1}{\mu_S} \\ &= \frac{\sum_{i=1}^n \left(\Lambda_i \cdot \frac{1}{\mu_i - \Lambda_i} \right)}{\sum_{i=1}^n \Lambda_i} + \frac{1}{\mu_C - \sum_{i=1}^n q_i^{pc} \lambda_i} + 2\Delta t_{prop,SDN} + \frac{1}{\mu_S} \end{aligned} \quad (5.4)$$

mit:

$\Delta t_{prop,SDN}$ feste Ausbreitungsverzögerung zwischen zwei SDN-Komponenten.

Im Vergleich zur Panev-Modellierung wird nicht zwischen der Ankunftsrate λ_i und der Gesamtankunftsrate Λ_i unterschieden, die durch eine Routing-Wahrscheinlichkeit zwischen zwei Switches eingeführt wurde. Hier wird davon ausgegangen, dass die virtuellen Instanzen über die OpenFlow-Switches verbunden sind und daher der gesamte Verkehr über die Switches läuft. Abgeleitet gilt vereinfachend die Ankunftsrate λ_i :

$$\begin{aligned} \lambda_i &= \Lambda_i \\ \lambda_1 &= \lambda_2 = \lambda_3 = \dots = \lambda_S \end{aligned} \quad (5.5)$$

Aus dieser Vereinfachung wird auch der gewichtete Mittelwert aus Formel (5.1) aufgelöst. Setzt man (5.5) in (5.4) ein, können die Summen der verschiedenen Switches zusammengefasst werden. Als Ergebnis erhält man die erwartete Gesamtverzögerung des Modells in Gleichung 5.6.

$$E(\Delta T_{SDN}) = \Delta t_{prop,SDN} + \frac{1}{\mu_S - \lambda_S} + \frac{1}{\mu_C - n \cdot \lambda_S \cdot q_S^{pc}} + \frac{1}{\mu_S} \quad (5.6)$$

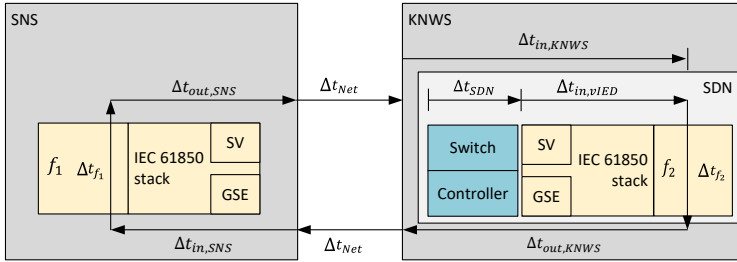


Abbildung 5.10: Übertragungszeitkomponenten innerhalb der Co-Simulation für das Gesamtverzögerungsmodell

5.4.1 Gesamtverzögerungsmodell

Für das Gesamtverzögerungsmodell der Co-Simulation werden die externen und internen Einflüsse der SDN-basierten Netzwerkvirtualisierung gemeinsam in ein mathematisches Modell überführt. Relevant für die Untersuchungen der neu entworfenen Co-Simulationen sind Situationen, in denen die beiden Simulatoren interagieren. Dies ist beispielsweise bei einer Schutzauslösung der Fall. Hier muss zusätzlich die Hard- und Softwarebasis der Stromnetzsimulation genauer untersucht und einbezogen werden. In der vorliegenden Arbeit ist jedoch die Beschränkung auf die Effekte außerhalb des Stromnetzsimulators eingeführt. Es wird angenommen, dass die Stromnetzsimulation den formulierten Bedingungen aus 4.1.2 entspricht und folglich die Echtzeitanforderungen erfüllt und sowohl interne Berechnungen zur Lösung der Modellgleichungen Δt_{f_1} , als auch das Ver- und Entpacken $\Delta t_{in,sim}$ und $\Delta t_{out,sim}$ der Informationen aus den Übertragungsprotokollen innerhalb der Simulationsschrittweite Δt_{SNS} erfolgt.

Abgeleitet setzt sich das hier beschriebene Gesamtmodell aus den Einflüssen von der Übertragung zwischen den beiden Simulatoren Δt_{Net} und den Verzögerungen auf dem KNWS Δt_{KNWS} zusammen. Die Verzögerungszeit Δt_{KNWS} tritt sowohl im Eingangspfad mit $\Delta t_{in,KNWS}$ als auch im Ausgangspfad mit $\Delta t_{out,KNWS}$ auf. Das vorgeschlagene Verzögerungsmodell ist richtungsunabhängig und erlaubt es, Δt_{SDN} für beide Pfade in gleicher Weise anzuwenden. Die einzelnen genannten Verzögerungskomponenten sind in Abbildung 5.10 dargestellt.

Insgesamt wird von den folgenden Vereinfachungen des Modells ausgegangen:

$$\begin{aligned}
\Delta t_{SNS} &= \Delta t_{f_1} = \Delta t_{in,SNS} = \Delta t_{out,SNS} \\
\Delta t_{vIED} &= \Delta t_{in,vIED} = \Delta t_{out,vIED} \\
\Delta t_{KNWS} &= \Delta t_{in,KNWS} = \Delta t_{out,KNWS} \\
&= \Delta t_{eth} + \Delta t_{prop,eth} + \Delta t_{SDN} + 2\Delta t_{veth} + \Delta t_{prop,CN} + \Delta t_{vIED}
\end{aligned} \tag{5.7}$$

mit:

Δt_{SNS}	Verzögerung innerhalb des SNS,
Δt_{KNWS}	Verzögerung innerhalb des KNWS,
Δt_{eth}	Verzögerung vom realen Netzwerkadapter,
$\Delta t_{prop,eth}$	Verzögerung zur Weitergabe an die Virtualisierung,
Δt_{veth}	Verzögerung vom virtuellen Netzwerkadapter,
$\Delta t_{prop,CN}$	Verzögerung zwischen Switch und vIED,
Δt_{vIED}	Verzögerung vom vIED.

Die neu eingeführten Komponenten sind nicht vollständig in Abbildung 5.10 enthalten. Eine weitere Detaillierung wird im Rahmen der Validierung in Abbildung 6.5 dargestellt und integriert.

Für die Beschränkung des Modells auf die relevante Größe der Übertragungszeitkomponente Δt_{TT} , also der Übertragungszeit analog zu Gleichung (3.3), ergibt sich folgendes Gesamtmodell in Gleichung 5.8:

$$\begin{aligned}
\Delta t_{TT} &= \Delta t_{SNS} + \Delta t_{Net} + \Delta t_{eth} + \Delta t_{prop,eth} + \Delta t_{SDN} + 2\Delta t_{veth} \\
&\quad + \Delta t_{prop,SDN} + \Delta t_{vIED} \\
&= \Delta t_{SNS} + \Delta t_{Net} + \Delta t_{eth} + \Delta t_{prop,eth} + 2\Delta t_{prop,SDN} + \frac{1}{\mu_S} + \frac{1}{\mu_S - \lambda_S} \\
&\quad + \frac{1}{\mu_C - n \cdot \lambda_S \cdot q_S^{pc}} + 2\Delta t_{veth} + \Delta t_{prop,SDN} + \Delta t_{vIED}
\end{aligned} \tag{5.8}$$

Das Verzögerungsmodell beschreibt folglich das Verhalten und den Einfluss der Übertragungszeit bedingt durch die SDN-Komponenten in Form von Warteschlangen und Verarbeitungszeiten und die internen Übergabestrecken der Netzwerkpakete im virtuellen Netzwerk innerhalb des KNWS. Ein möglicher Einfluss des Betriebssystem-bedingten Scheduling zur sequenziellen Abarbeitung der Prozesse wird nicht betrachtet. [79], [89]

5.4.2 Integration des Verzögerungsmodells in das Framework VirtualSubstation

In der Initialisierungsphase der Virtualisierung mittels des vorgestellten VirtualSubstation-Frameworks werden direkt die Details der Zielkonfiguration extrahiert. Dabei sind die Konfigurationsdaten mitunter Einflussgrößen für die Verzögerungsberechnung. Die Struktur der Netzwerkumgebung mit den Informationen zu der Anzahl von Komponenten und den vorhandenen GOOSE- und SV-Server ergeben eine Berechnungsgrundlage für die erwartete Verzögerung innerhalb der Co-Simulation. Vor allem die hochfrequent abgetasteten Messwerte über MUs, die in der Stromnetzsimulation platziert werden, bilden den Haupteinflussfaktor für die Ankunftsrate an den Prozessbus-Switches unter der Annahme, dass der Prozessbus die Mess- und Steuernachrichten bündelt. Für eine korrekte Abbildung werden zunächst die Abgänge aus der SCD-Datei extrahiert und für jeden Abgang ein Prozessbusswitch mit einer Sterntopologie aufgebaut. Die jeweiligen vIEDs werden mit dem Switch verbunden. Über die angegebenen SV- und GOOSE-Server kann anschließend eine Abschätzung der Ankunftsrate an den jeweiligen Switches getroffen werden.

Das Ziel der Integration der Übertragungszeitberechnung in das Virtualisierungsframework ist die Aussage über eine mögliche Verletzung der Echtzeitanforderungen aus der IEC 61850-5. Damit das Co-Simulationssystem als valide und die Aussagen als gültig bezeichnet werden können, muss ausgeschlossen werden, dass die Anforderungen in einer Situation verletzt werden. Hierfür werden die Informationen verarbeitet und die erwartete Verzögerung unter der abgeschätzten Ankunftsrate und der angenommenen Übergabewahrscheinlichkeit zwischen Switch und Controller im Initialisierungsprozess berechnet. Die eingeführte Berechnung der Verzögerung erzeugt einen Erwartungswert für die benötigte Übertragungszeit einer Nachricht zwischen den Simulatoren unter der angenommenen Verteilungsfunktion.

6 Validierung der Co-Simulation

Der eingeführte Ansatz der Co-Simulation wurde für den Anwendungsfall der Stationsautomatisierung in Digitalen Stationen formuliert und enthält verschiedene Alleinstellungsmerkmale. Um die Gültigkeit des Ansatzes zu überprüfen, werden im Folgenden verschiedene Aspekte der Validierung dargestellt.

6.1 Validierungsvorgehen

Neben den Forschungsfragen werden im folgenden Abschnitt die bisher gewonnenen Erkenntnisse der vorliegenden Arbeit validiert, bewiesen oder näher betrachtet. Diese sind in Tabelle 6.1 zusammengefasst und um die später beschriebenen Validierungsmethoden ergänzt.

Tabelle 6.1: Validierungsgegenstände und -vorgehen

Nr.	These	Vorgehen des Nachweises
1	Die Virtualisierung kann das Netzwerkverhalten einer Digitalen Station adäquat nachbilden und emulieren.	Nachweis über zwei aufgenommene Netzwerkmitschnitte
2	Der Koordinator übernimmt die Aufgabe der Synchronisierung mit einer nicht-Co-Simulations-spezifischen Implementierung.	Beschreibung der Synchronisierung über gemeinsames Netzwerk
3	Die Kopplung aus Simulation und Virtualisierung kann grundlegend die Anforderungen zur Übertragungszeit aus IEC 61850 einhalten.	Messung anhand einer beispielhaften virtuell implementierten Schutzfunktion
4	Das aufgestellte mathematische Verzögerungsmodell mittels Warteschlangen-Ansatz besitzt Gültigkeit für die aufgebaute Virtualisierung.	Messung verschiedener Kombinationen der Container-, Switch-Anzahl und Variation des Netzwerktraffics mit anschließendem Vergleich mit dem Warteschlangenmodell
5	Das Framework VirtualSubstation kann ein vorgegebenes Netzwerk aus einer SCD-Beschreibung adäquat nachbilden.	Analyse des virtualisierten Netzwerkes mit Initialisierung von Services

6.1.1 Adäquate Abbildung durch die Nutzung einer Virtualisierung

Die erste Grundannahme für das vorgestellte Co-Simulationssystem ist die prinzipielle Abbildbarkeit der Netzwerkebene einer Digitalen Station innerhalb einer Virtualisierungsumgebung. Dabei steht vor allem die generelle Möglichkeit im Vordergrund, einen IEC 61850-Netzwerkverkehr zu erzeugen, der in realen IEC 61850-Stationen auftritt. Es müssen identische Kommunikationsbeziehungen mittels der Virtualisierung nachgebildet und das Netzwerkverhalten emuliert werden. Die Verwendung von einzelnen Docker-Containern zur virtuellen Abbildung von IEDs einer Digitalen Station inklusive der SDN-basierten Virtualisierungsmethodik stellt die Basis dar.

Als Hauptkriterium zur Validierung, ob das abgebildete virtuelle Netzwerk einem realen Stationsnetzwerk entsprechen kann, wird der Netzwerkverkehr des Prozessbusses herangezogen. Dies ist der zentrale Punkt der Co-Simulation, an dem die Simulatoren und deren Modellierungen Einfluss auf den Netzwerkverkehr haben. Alle abgebildeten vIEDs empfangen Daten vom Prozessbus und speisen diesen. In der Folge soll untersucht werden, ob sich der Netzwerkverkehr eines Co-Simulationsaufbaus mit dem Verhalten eines Realaufbaus vergleichen lässt.

Zu diesem Zweck werden zwei Aufbauten benötigt, deren Netzwerkverkehr mitgeschnitten und anschließend verglichen werden kann. Für beide Fälle wird ein identisches Simulationsszenario gewählt, über das beide Ebenen der Co-Simulation (SNS und KNWS) angesprochen werden. Für einen reproduzierbaren Testaufbau wird ein Simulationsszenario innerhalb der Stromnetzsimulation umgesetzt, das einen Kurzschluss in einem Netzausschnitt zu einem fest definierten Zeitpunkt umsetzt. Die zugehörigen Momentanwerte von Strom- und Spannung werden über die beschriebenen IEC 61850 SV-Schnittstellen an einen externen Prozessbus geleitet. Die beiden zu vergleichenden Aufbauten setzen dabei den Prozessbus unterschiedlich um. Bei dem Aufbau der virtuellen Station befindet sich der Switch zur Umsetzung des Prozessbusses virtualisiert innerhalb von Containernet. Dementsprechend wird der SNS, wie für die Co-Simulation beschrieben, direkt mit dem KNWS verbunden. Bei dem Vergleichsaufbau ohne Virtualisierung wird der SNS mit einem realen Switch verbunden, der den Prozessbus bildet und die Daten an weitere angeschlossene Systeme verteilt. Die notwendige Synchronisierung in beiden Versuchsaufbauten wird über die Anbindung der einzelnen Geräte an einen PTP-Master realisiert. Im Aufbau der virtuellen Station wird jeweils ein separater Netzwerkadapter verwendet. In der realen Station ist der PTP-Master an den Prozessbus-Switch angeschlossen. Der nachfolgend auszu-

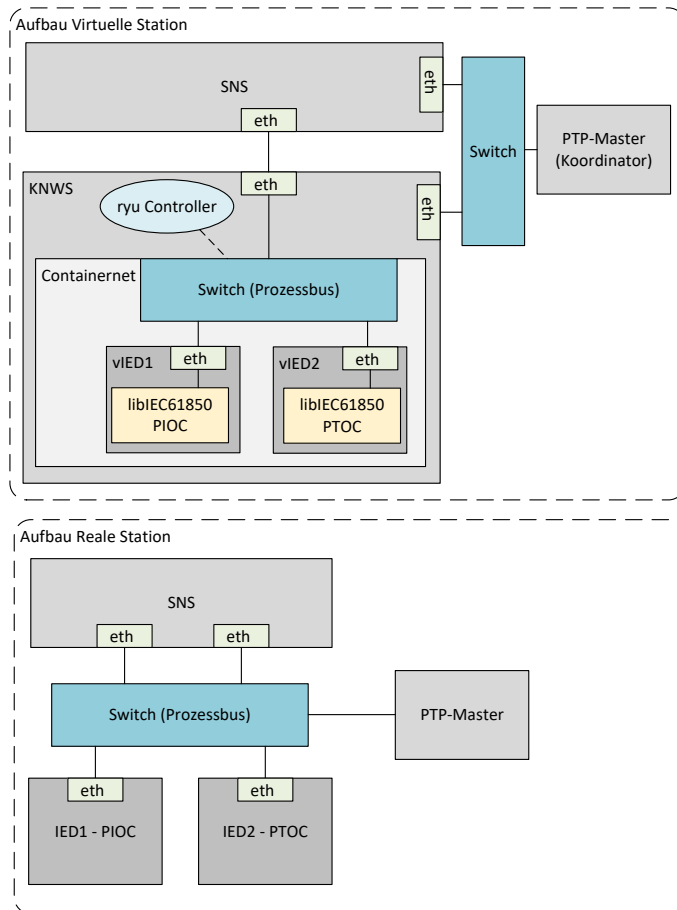


Abbildung 6.1: Aufbauten zur Validierung des Netzwerkverhaltens

wertende Netzwerkverkehr wird jeweils auf dem Prozessbus-Switch durch Spiegelung sämtlicher Ports aufgenommen. In beiden Aufbauten werden zwei Schutzsysteme eingebunden, zum einen virtualisiert als vIEDs, und zum anderen als reale Systeme vom Hersteller ABB. Dabei wird ein System für den instantanen und eins für den zeitgesteuerten Überstromschutz (Protection Instantaneous Overcurrent (PIOC), PTOC) verwendet. Der grundlegende Aufbau ist in Abbildung 6.1 dargestellt.

Die Validierung umfasst die Erzeugung des Netzwerkverkehrs in beiden Aufbauten und den Vergleich der aufgenommenen Daten. Dabei werden Merkmale des Netzwerkverkehrs herausgearbeitet und analysiert. Ziel ist es, festzustellen, in welchen Punkten das nachgebildete virtuelle Netzwerk von dem realen Laboraufbau abweicht [94].

Für den Vergleich der Netzwerkmitschnitte werden die generierten Packet Capture (PCAP)-Dateien statistisch ausgewertet. Hierfür wird ein Ansatz gewählt, der die relevanten Merkmale aus den PCAP-Dateien extrahiert und diese mit einem Ähnlichkeitsmaß vergleicht. Das Vorgehen lehnt sich an die in [95] ausgeführten Techniken der Validierung von Netzwerk-Traffic-Generatoren an. Für die Merkmale des Netzwerkverkehrs werden unterschiedliche Ähnlichkeits- und Distanzmaße in Erwägung gezogen, um die Vergleichbarkeit herzustellen. Als prinzipiell geeignete Maße werden der Pearson-Korrelationskoeffizient, die Kosinus-Ähnlichkeit und die euklidische Distanz aus der Literatur identifiziert.

Im Rahmen der Arbeit wird sich auf die Kosinus-Ähnlichkeit beschränkt, da die Vorarbeiten aus [94] zeigen, dass der Netzwerkverkehr mittels dieses Maßes am ehesten verglichen werden kann. Die Aussage basiert auf einer Kreuzvalidierung mit synthetischem Netzwerkverkehr. Die Kosinus-Ähnlichkeit (CS) misst den Kosinus des Winkels zwischen zwei Vektoren [96]. Sie ignoriert die Größe der Vektoren und konzentriert sich auf ihre Ausrichtungen, wodurch sie die semantische Ähnlichkeit effektiv erfasst. Formell lässt sich die Kosinus-Ähnlichkeit berechnen nach

$$CS = \frac{\sum_{i=1}^n x_i y_i}{\sqrt{\sum_{i=1}^n x_i^2} \sqrt{\sum_{i=1}^n y_i^2}} \quad (6.1)$$

mit \vec{x} und \vec{y} als Beobachtungsvektoren der Dimension n der betrachteten Messpunkte. Die Kosinus-Ähnlichkeit beschreibt ein Wertintervall von $[-1, 1]$, wobei die Vektoren bei der Ausprägung von eins identisch sind, bei null keine Ähnlichkeit aufweisen und bei Minus eins entgegengesetzt sind.

Die relevanten Merkmale der Netzwerkmitschnitte werden in Anlehnung an [95], [97] definiert und aus den PCAP-Dateien extrahiert. Dabei werden die Merkmale in fünf Kategorien unterschieden und diese in einer Datenvorverarbeitung mittels des OpenSource Tools PyShark [98] berechnet. Die Auswahl der Merkmale ist angelehnt an die Protokoll-Analyse des weit verbreiteten Netzwerk-Analyse-Tools Wireshark [99]. Als Resultat ergeben sich folgende Merkmalskategorien:

- Paket-basierte Merkmale: Extrahierbare Merkmale eines jeden aufgenommenen Netzwerkpakets.
- Konversations-basierte Merkmale: Netzwerkpakete werden in erkannten Konversationen zwischen zwei Endpunkten zusammengefasst und aus diesen Konversationen extrahierbare Merkmale abgeleitet.
- Endpunkt-basierte Merkmale: Netzwerkpakete werden den erkannten Endpunkten zugeordnet und zusammengefasst. Zugehörige Merkmale werden statistisch aus diesen Paketzusammenfassungen generiert.
- Protokoll-basierte Merkmale: Die Netzwerkpakete werden je erkanntem Netzwerkprotokoll zusammengefasst und relevante Merkmale extrahiert.
- I/O-basierte Merkmale: Unter die I/O-basierten Merkmale fallen zusätzliche Statistiken aus den paket-basierten Merkmalen, wobei Pakete sekundenweise zusammengefasst und ausgewertet werden.

Die in Tabelle 6.2 aufgeführten Merkmale des Netzwerkverkehrs dienen dem Vergleich des Verhaltens der Co-Simulation mit dem des realen Aufbaus. Jedes Merkmal ist einer bestimmten Kategorie zugeordnet. So findet sich beispielsweise das Merkmal „Quelle“ in den ersten beiden Kategorien. Dies bedeutet, dass bei den paket-basierten Merkmalen jedes einzelne Paket hinsichtlich der enthaltenen Quellinformation betrachtet wird. Bei den konversations-basierten Merkmalen wird der Netzwerkverkehr zunächst in die einzelnen Konversationen unterteilt, wodurch die Pakete jeder Konversation zusammengefasst werden, und anschließend wird die Quelle dieser Konversationen analysiert.

6.1.2 Synchronität der Simulatoren ohne spezifische Koordinator-Umsetzung

Im Vergleich zu anderen in der Literatur vorgestellten Co-Simulationsansätzen zeichnet sich der in dieser Arbeit entwickelte Ansatz dadurch aus, dass die beiden Simulatoren direkt gekoppelt werden können, ohne dass eine speziell entworfene Koordinationsinstanz erforderlich ist. Der eingesetzte Koordinator fungiert als allgemeingültige Synchronisationsinstanz, die durch einen PTP-Master realisiert wird. Der Datenaustausch erfolgt unabhängig von dieser Koordination. Somit übernimmt der Koordinator lediglich eine Teilmenge der definierten Aufgaben und ist nicht spezifisch

Tabelle 6.2: Verwendete Merkmale des Netzwerkverkehrs

Kategorie	Merkmal
Paket-basierte Merkmale	Zeitstempel Quelle Ziel Paketlänge
Konversations-basierte Merkmale	Quelle Ziel Paketanzahl Gesamtlänge Paketrate Byte-Rate
Endpunkt-basierte Merkmale	Adressen TX Paketanzahl TX Bytes RX Paketanzahl RX Bytes
Protokoll-basierte Merkmale	Protokoll Paketanzahl Gesamtlänge Paketrate Byte-Rate
I/O-basierte Merkmale	Zeitstempel Paketanzahl Bytes

auf die vorliegende Co-Simulation zugeschnitten. Um die Gültigkeit dieses Aufbaus nachzuweisen, wird auf die verwendete Synchronisierung der einzelnen Simulatoren verwiesen. Über eine PTP-Infrastruktur sind diese synchron mit der festgelegten Genauigkeit zum installierten Zeitgeber und demnach auch die Komponenten untereinander. Es wird angenommen, dass die instanziierten vIEDs innerhalb des Virtualisierungsrechners den Zeitstempel des Simulators bekommen und folglich Teil der PTP-Infrastruktur sind. Auf diese Aussage stützen sich beispielsweise auch die Autoren von [100], [101]. Dabei müssen die einzelnen Docker-Container die Hardware-Clock des Virtualisierungshosts nutzen. Aufgrund des gemeinsam verwendeten Kernels zwischen Host und Container kann sichergestellt werden, dass die gleiche Zeitbasis genutzt und diese vom Host geteilt wird. Bei Nutzung einer Synchronisierungsinfrastruktur, wie PTP, kann laut Abschnitt 5.2.1.4 angenommen werden, dass auch die Container und alle instanziierten IEDs im virtuellen SDN-Netzwerk synchron sind.

Die zweite Hauptaufgabe des Koordinators in Co-Simulationen stellt die Orchestrierung und Sicherstellung des geregelten Datenaustauschs dar, wie es in Abbildung 2.3 eingeführt und beschrieben ist. In alternativen Ansätzen muss demnach erreicht werden, dass zum einen das Übergabeformat der Informationen über definierte Schnittstellen der Simulatoren bereitgestellt werden kann und die Informationen über den Koordinator zu den korrekten Zeitpunkten übergeben werden. Diese Aufgabe kommt primär bei Systemen zum Tragen, bei denen die Simulationszeit von der Ausführungszeit abweicht. Dies kann bei Offline-Simulation, deren einzelne Berechnungsschritte angestoßen werden, oder auch bei beschleunigten Simulationen der Fall sein.

In dem vorliegenden Ansatz der Co-Simulation entspricht die Simulationszeit der Ausführungszeit. Die Simulationen finden online in Echtzeit statt. Infolgedessen entfällt die Aufgabe der externen Taktung der Simulatoren und der Übergabe von Informationen zwischen diesen. Sie können über die implementierten Protokollschnittstellen zur Ausführungszeit direkt an den jeweils anderen Simulator übertragen werden, ohne die Koordinatorinstanz zu adressieren. Durch den Wegfall der Aufgabe des Datenaustauschs des Koordinators wird der potenziell negative Einfluss des Koordinators auf die Echtzeitfähigkeit der Co-Simulation implizit vermieden. Beide Simulatoren erfüllen die Anforderung der Echtzeitfähigkeit und werden über den Koordinator synchronisiert.

Der Nachweis der Synchronität und Echtzeitfähigkeit der Co-Simulation wird durch die Analyse von erfasstem Netzwerkverkehr erbracht, wobei auf die relevanten Merkmale der Netzwerkpakete verwiesen wird.

6.1.3 Anforderungen an die IEC 61850-Übertragungszeit

Die in Abschnitt 3.3.2 dargelegten Zeitanforderungen, die für die Formulierung der Co-Simulation verwendet wurden, müssen nach IEC 61850-5 eingehalten werden. Für den grundlegenden Nachweis insbesondere der Übertragungszeitrestriktion wird ein Minimalbeispiel einer virtualisierten Schutzfunktion aufgebaut. Für das Minimalbeispiel wird das virtuelle Kommunikationsnetzwerk so weit wie möglich vereinfacht und besteht lediglich aus einem vIED, das direkt mit einem virtuellen Switch verbunden ist. Die Verbindung zwischen dem SNS und dem KNWS wird direkt ohne zusätzliche reale Netzwerkkomponenten vorgenommen. Hieraus werden äußere Einflüsse auf ein Minimum reduziert. Es resultiert der nachfolgende Testaufbau, wie er in Abbildung 6.2 dargestellt ist.

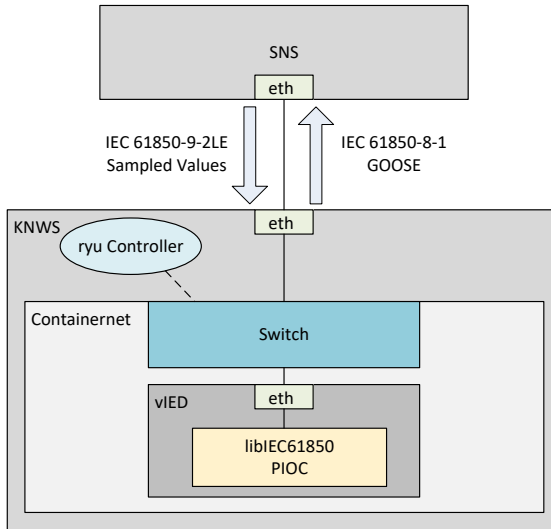


Abbildung 6.2: Minimalbeispiel zur Validierung der Anforderungen zur Übertragungszeit

Für die erste Annäherung besteht das vIED aus einem PIOC und einer, auf der Bibliothek `libIEC61850` basierenden IEC 61850-Implementierung mit den Prozessbus-Protokollen IEC 61850-8-1 GOOSE und IEC 61850-9-2LE SV. Innerhalb der Container-Netzwerk-Virtualisierung wird ein OpenFlow-Switch verwendet, der mit einem ryu-SDN-Controller gesteuert wird. Das gewählte Minimalbeispiel eignet sich trotz der Vereinfachung für die Prüfung der Einhaltung der Übertragungszeiten, da die Zielarchitektur der Virtualisierung verwendet wird und alle identifizierten Verzögerungskomponenten aus 5.4 enthalten sind. Jedoch werden alle zusätzlich auftretenden Effekte wie beispielsweise Controller-Auslastung so weit wie möglich vermieden.

Um Aussagen über die entstandenen Verzögerungen sowie die Validität der Annahmen im Vergleich zu realen IED-Implementierungen zu treffen, werden Messungen der aufgetretenen Übertragungszeiten für das Minimalbeispiel durchgeführt. Für die Betrachtung der Ähnlichkeit zu realen IEDs können beispielsweise die ermittelten Größen mit Referenzliteratur zu kommerziellen Systemen [102] verglichen werden.

Die Abbildung 6.3 zeigt die definierten und untersuchten Evaluationsszenarien. Dabei entspricht das Szenario (i) einem realen IED, in dem die Funktions- und Protokollimplementierungen direkt auf dem System vorgenommen sind und die Programme

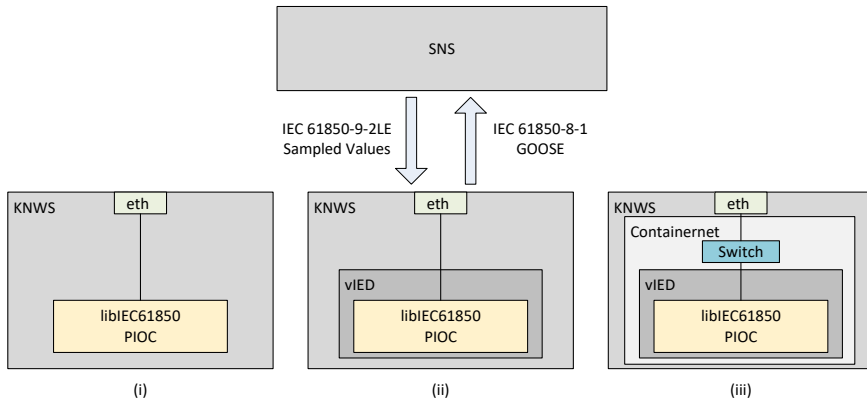


Abbildung 6.3: Evaluationsszenarien mit (i) der Funktionalität direkt laufend auf dem physischen Gerät, (ii) einer virtualisierten Funktion mit Standard Docker-Netzwerk und (iii) dem eingeführten Minimalbeispiel

direkt vom Betriebssystem der realen Komponente ausgeführt werden. Im zweiten Szenario (ii) wird die gleiche Funktionalität als Docker-Container unter Nutzung des Standardnetzwerkes von Docker instanziiert. Hierbei entspricht das vIED bereits der Zielarchitektur. Das vIED, also der Docker-Container, enthält wiederum die identische Implementierung der Schutzfunktion und der Protokollumsetzung nach IEC 61850 wie in dem Szenario (i). Innerhalb der realen Komponente müssen, die am realen Netzwerkadapter eintreffenden Netzwerkpakete lediglich über ein Bridge-Interface an den Container und dessen virtuellen Adapters übergeben werden. Die implementierten Funktionen können anschließend auf die übertragenen Daten zugreifen und die Schutzfunktion umsetzen. Ausgehende Nachrichten werden identisch verarbeitet und gelangen so über den Container zur Schnittstelle der realen Komponente. Das Szenario (iii) entspricht dem Zielbild, wie es in Abbildung 6.2 dargestellt ist. Es beinhaltet zusätzlich zur Container-Implementierung noch das virtuelle Netzwerk in Form der SDN-Architektur mit einem OpenFlow Switch und einem SDN-Controller. Die Funktion und Protokollimplementierung bleiben unverändert.

Für die Validierung des neuentworfenen Co-Simulationsansatzes werden in der Folge die drei Aufbauvarianten mit dem gleichen Überstrom-Ereignis innerhalb der Stromnetzsimulation angeregt. Es werden sowohl das Verhalten in der Stromnetzsimulation als auch der resultierende Netzwerkverkehr ausgewertet. Zusätzlich spielt bei der Auswertung der Übertragungszeit der Zeitpunkt eine besondere Rolle, zu

dem die Informationen auf der Sender- und Empfängerseite außerhalb des Protokolls vorliegt. Betrachtet werden hierbei seitens des Senders der Zeitpunkt, bevor die Information in das Protokoll verpackt wird, und seitens des Empfängers der Zeitpunkt, nachdem die Information entpackt wird. Dies ist Abbildung 3.3 zu entnehmen.

Da der Fokus auf den Übertragungszeitanteilen der Virtualisierung der Kommunikationsebene liegt, ist eine simulative Untersuchung des Zeitbedarfs für das Ver- und Entpacken der Netzwerkpakete erforderlich. Diese Größe kann nicht allein durch die Analyse des Netzwerkverkehrs ermittelt werden. Daher wird in den nachfolgenden Simulationen das Senden eines GOOSE-Befehls zur Schutzanregung untersucht. Innerhalb der Funktionsschicht des vIED wird ein Zeitstempel integriert, der nach dem Auslösen des Überstrom-Ereignisses und vor der Übergabe der Information an die Protokollschicht generiert wird. Dadurch umfasst die zeitliche Betrachtung auch das Verpacken der Information im GOOSE-Protokoll, sodass der erzeugte Zeitstempel zur Berechnung der Übertragungszeit verwendet werden kann.

Für die Stromnetzsimulation wird die, vom Hersteller garantierte Simulationsschrittweite herangezogen. In dieser Zeit soll ebenfalls die Information in ein Übertragungsprotokoll überführt, beziehungsweise aus einem Protokoll extrahiert und in die Simulation integriert werden. Im Rahmen der Arbeit wird auf eine nähere Untersuchung und Betrachtung der Stromnetzsimulation verzichtet. Dementsprechend wird keine zusätzliche Validierung in der Art durchgeführt, sondern für die Berechnung der Übertragungszeit die Annahme der Simulationsschrittweite verwendet. Gestützt wird die Annahme durch beispielsweise die Untersuchungen von Memon [103]. Eine grobe Einschätzung der Gültigkeit dieser Aussage lässt sich jedoch auch aus den aufgenommenen zeitlichen Verläufen innerhalb der Stromnetzsimulation, also der Auslösezeit, durch die Betrachtung der elektrisch simulierten Signale herstellen. Für die Messungen wird das gesendete GOOSE Paket von der Stromnetzsimulationen empfangen und in ein binäres Simulationssignal überführt. Somit endet die Übertragungszeitberechnung mit der Detektion der Flanke durch die Änderung des Binärwertes in der Simulation. Zu diesem Zeitpunkt wurde die Information aus dem Protokoll extrahiert und steht der Simulation in verwertbarer Art zur Verfügung. Die Messung der Übertragungszeit ist in Abbildung 6.4 schematisch dargestellt.

Die erzeugten Messreihen werden in dem Abschnitt der Durchführung dargestellt und die ermittelte Übertragungszeit der drei Szenarien betrachtet und untereinander verglichen. Der Einfluss der Virtualisierung kann daraus experimentell abgeleitet werden. Wie eingeführt, wird hier zunächst nur ein Minimalbeispiel betrachtet. Weiterführende, komplexere Szenarien mit einer größeren Anzahl von Containern,

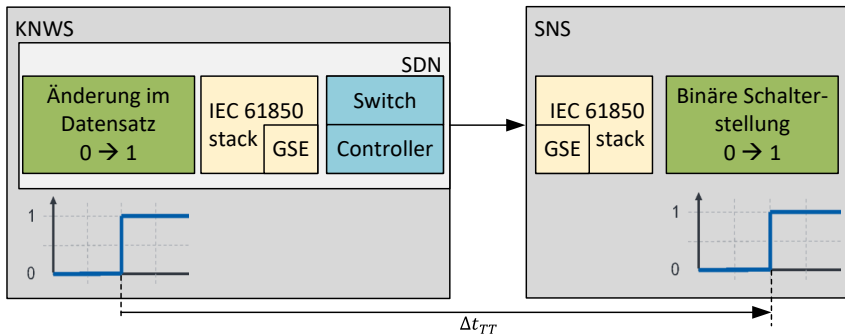


Abbildung 6.4: Schematische Darstellung der Übertragungszeitmessung zwischen dem KNWS und SNS

Switchen oder erhöhtem Netzwerkverkehrsaufkommen werden im nachfolgenden Abschnitt betrachtet, nachdem die grundlegende Einhaltung der zu erfüllenden Zeitanforderung hier nachgewiesen ist.

6.1.4 Mathematische Berechnung von auftretenden Verzögerungen

Das in Abschnitt 5.4 eingeführte Verzögerungsmodell für SDN-Infrastrukturen und den gewählten Ansatz der Co-Simulation beschreibt zunächst theoretisch den Zusammenhang zwischen den unterschiedlichen Einflussvariablen und der auftretenden Übertragungszeit beim Datenaustausch in der Co-Simulation. Folgende Einflussvariablen wurden diesbezüglich identifiziert:

- Anzahl der Container
- Anzahl der Switches
- auftretender Datenverkehr
- Übergabewahrscheinlichkeit zwischen Switch und Controller
- Controller-Interaktion bezüglich der zu untersuchenden Paketübertragung

Im Rahmen der Validierung werden Messdatensätze erzeugt, die Variationen der Einflussvariablen beinhalten und die resultierenden Übertragungszeiten ermitteln lassen.

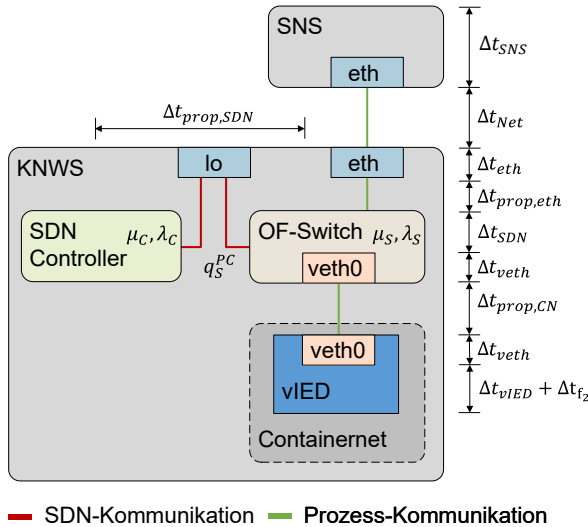


Abbildung 6.5: Komponenten des Verzögerungsmodells im gewählten Versuchsaufbau

Anschließend folgt ein Vergleich der Messdaten mit den resultierenden Ergebnissen aus dem mathematischen Verzögerungsmodell. Das Ziel ist hierbei aufzuzeigen, dass das verwendete Warteschlangenmodell für die Co-Simulation und dessen Datenaustausch gültig ist und verwendet werden kann, um eine Einschätzung der Übertragungszeit für unterschiedliche Konstellationen der Netzwerkinfrastruktur zu generieren.

Für die Validierung werden dementsprechend die relevanten Daten der unterschiedlichen Konstellationen für einen festen Testaufbau messtechnisch erfasst, in dem sich die einzelnen Komponenten des Verzögerungsmodells aus Gleichung (5.8) direkt anhand der unterschiedlichen Zeitstempel der Pakete im Netzwerk, die in den Traffic-Aufzeichnungen generiert werden, bestimmen lassen. Dabei werden unterschiedliche reale und virtuelle Netzwerkschnittstellen überwacht und Informationen im PCAP-Format aufgezeichnet. Der Versuchsaufbau ist in Abbildung 6.5 dargestellt und beschreibt die unterschiedlichen Messpunkte und Verzögerungskomponenten.

An dem Beispiel eines GOOSE-Pakets, das von dem implementierten vIED an die Stromnetzsimulation geschickt wird, werden in der Folge die simulativen Untersuchungen durchgeführt. Hierzu werden die Daten messtechnisch erfasst und hinsichtlich der sich ergebenden Zeitstempel ausgewertet. In dieser Art der Validierung spielt

die Implementierung einer Funktion innerhalb des vIEDs keine Rolle. Demnach benötigt es keiner Auswertung der Round-Trip-Time (RTT) durch beispielsweise einer Schutzanregung bis hin zu einer umgesetzten Schalthandlung innerhalb der Stromnetzsimulation. Das Verzögerungsmodell konzentriert sich auf die Anteile der Übertragungszeit, die durch die Virtualisierung entstehen, wodurch ein einfaches Senden eines GOOSE-Pakets innerhalb des vIEDs ausreichend ist. Das GOOSE-Paket wird im vIED generiert und verschickt und kann über unterschiedlichen Netzwerkadapter mittels einer Paketaufzeichnung erkannt werden. Die Zeitstempel der jeweiligen Adapter im Netzwerkmittschnitt werden nachfolgend verwendet. Im Fall einer SDN-Controller-Interaktion durchläuft also das GOOSE-Paket den in Abbildung 5.9 dargestellten Pfad. Folglich kann das Paket an den Netzwerkadaptern des vIED, des OpenFlow-Switches, des SDN-Controllers (beide über das Loopback-Interface (lo)), dem realen Adapter des Hosts und dem Stromnetzsimulator aufgenommen werden. Bei der SDN-Kommunikation ist das GOOSE-Paket dabei in die OpenFlow Nachrichten (Packet-In, Packet-Out und Flow-Mod) eingebettet. Durch Bildung von Differenzen der Zeitstempel lassen sich die einzelnen Komponenten des Verzögerungsmodells berechnen.

Die notwendige Variation der einzelnen Einflussvariablen des Verzögerungsmodells können dementsprechend automatisiert erzeugt, das resultierende virtuelle Netzwerk instanziiert und das zu überwachende GOOSE-Paket innerhalb des vIEDs versendet werden. Die automatisierte Generierung der Variationen und Durchführung der Simulationen sind in dem Programmablaufplan der Abbildung 6.6 dargestellt.

Die Variationen werden nach folgendem Schema vorgegeben und in dem initialen Schritt des Programmablaufplans der automatisierten Ausführung übergeben:

- Definition Anzahl Container und Switche: Neben der reinen Variation der Größen steht auch die Variation der Anzahl von Containern pro Switch im Vordergrund.
 - GOOSE bekannt?: Soll eine Interaktion des Controllers herbeigeführt werden und im Simulationsfall der Controller über die Weiterleitung des Pakets entscheiden? Wenn das GOOSE-Paket bereits bekannt ist, kann es direkt über die vorhandene Flow-Tabelle vom Switch korrekt weitergeleitet werden.
 - Start SV-Publisher: Soll neben dem GOOSE-Paket zusätzlicher Netzwerkverkehr im virtuellen Netzwerk erzeugt werden? Hierdurch lässt sich die Ankunftsrate λ generieren. Der zugehörige SV-Stream wird über einen separaten Container erzeugt und belastet den Controller nur beim ersten Paketversand, bis
-

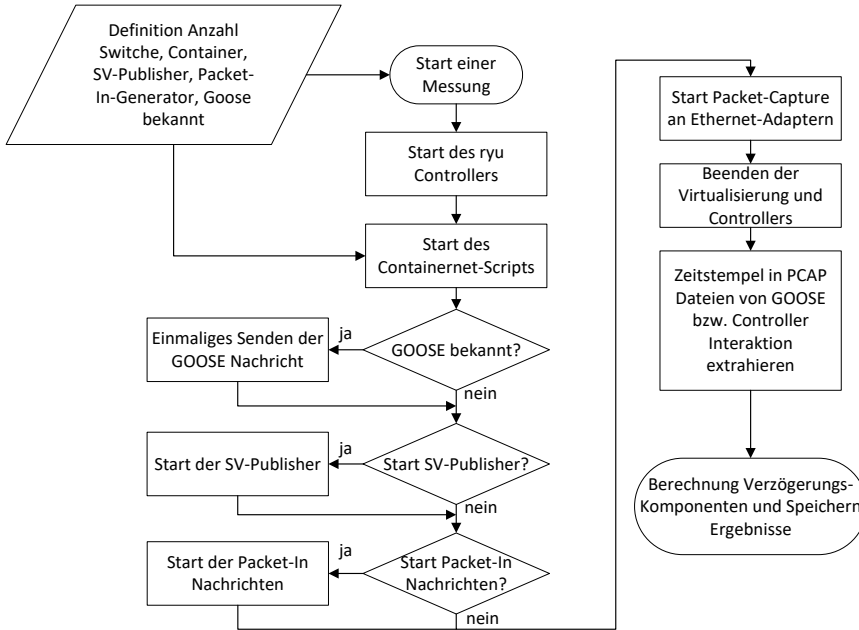


Abbildung 6.6: Programmablaufplan zur automatisierten Erstellung von Datensätzen unter Variation der Einflussvariablen

die Flow-Regel initiiert wurde. Bei der anschließenden Datenaufnahme findet also keine Controller-Interaktion aufgrund der SV-Pakete statt.

- Start Packet-In-Nachrichten: Soll der Controller über eine definierte Rate an Packet-In-Nachrichten belastet werden? Hier werden Internet Control Message Protocol (ICMP) Nachrichten in einem separaten Container erzeugt, die eine Controller-Interaktion generieren. Der Typ der erzeugten Pakete spielt dabei keine Rolle. Es muss nur sichergestellt werden, dass sie eine Controller-Interaktion erzielen, indem noch kein Eintrag in den Flow-Tabellen der Switches enthalten ist. Über dieses Konstrukt lässt sich die Wahrscheinlichkeit eines Packet-In-Ereignisses q_i^{pc} für das relevante GOOSE-Paket generieren.

Für die Validierung und zielgerichteten Untersuchung der Einflussvariablen werden jeweils eine oder mehrere Größen fixiert, wohingegen die jeweils anderen Größen in einem vorgegebenen Bereich variiert werden. Es wird jeweils die Übertragungszeit gemessen und die Abhängigkeiten dargestellt. Dabei wird in drei Kategorien der Variationen unterteilt:

1. Variation der Anzahl der Container pro Switch und Anzahl der Switche: Zunächst werden die Anzahl der SV-Publisher und die Rate der Packet-In-Ereignisse auf null festgelegt. Die Variationen werden zweimal durchgeführt: einmal für den Fall, dass die GOOSE-Nachricht bekannt ist und keine Controller-Interaktion stattfindet, und einmal für den umgekehrten Fall.
2. Variation der Anzahl der SV-Publisher und Anzahl der Switche: Die Anzahl der Container wird auf sechs festgelegt. Wichtig ist dabei, dass sie größer als eins ist, um mindestens einen dedizierten Container zu gewährleisten, über den zusätzliche SV-Publisher generiert werden können. In diesem Fall können bis zu fünf zusätzliche SV-Publisher instanziiert werden. Die Wahl der Anzahl sechs basiert auf den später dargestellten Ergebnissen aus der ersten Kategorie. Die Packet-In-Rate wird weiterhin auf null gesetzt. Analog zur ersten Kategorie wird der Weg über den Controller durch das vorgeschaltete Packet-In-Ereignis einmal erzwungen und einmal vermieden.
3. Variation der Rate von Packet-In-Ereignissen und Anzahl der Switche: Hier wird der Einfluss der Packet-In-Rate und der damit verbundenen Auslastung des Controllers untersucht. Dabei bleibt die Anzahl der Container bei sechs und die der SV-Publisher bei zwei konstant, während der Controller unterschiedlich stark belastet wird. Das vorausgehende Packet-In-Ereignis der GOOSE-Nachricht wird dabei analog zu den ersten beiden Kategorien berücksichtigt.

Die drei Kategorien mit den jeweils fixierten und zu variierenden Einflussgrößen sind in Tabelle 6.3 zusammengefasst.

6.1.5 Nachbildung eines Netzwerkes mit dem Framework VirtualSubstation

Das eingeführte Framework VirtualSubstation aus Abschnitt 5.2.2 hat das Ziel, aus der Beschreibung eines IEC 61850 basierten Umspannwerkes eine virtuelle Netzwerkinfrastruktur und eine Co-Simulation aufzubauen. Demnach gilt es nachzuweisen, dass aus der Beschreibung in Form einer SCD-Datei alle notwendigen Informationen

Tabelle 6.3: Kategorien der Variationen für die Validierung des Verzögerungsmodells

Nr.	Konstant	Variabel	Alternierend
1	<ul style="list-style-type: none"> • Anzahl SV-Publisher • Packet-In-Rate 	<ul style="list-style-type: none"> • Anzahl Container pro Switch • Anzahl Switche 	<ul style="list-style-type: none"> • GOOSE-Packet-In-Ereignis
2	<ul style="list-style-type: none"> • Packet-In-Rate • Anzahl Container pro Switch 	<ul style="list-style-type: none"> • Anzahl SV-Publisher • Anzahl Switche 	<ul style="list-style-type: none"> • GOOSE-Packet-In-Ereignis
3	<ul style="list-style-type: none"> • Anzahl Container pro Switch • Anzahl SV-Publisher 	<ul style="list-style-type: none"> • Packet-In-Rate • Anzahl Switche 	<ul style="list-style-type: none"> • GOOSE-Packet-In-Ereignis

automatisiert extrahiert und in die Virtualisierung überführt werden können. Aus den jeweils enthaltenen IEDs werden Docker-Container generiert und mit den relevanten Informationen, wie IP-Adressen und Kommunikationsservices innerhalb von Containernet instanziiert. Hierbei wird für die Validierung die Funktion der unterschiedlich möglichen IEDs vernachlässigt. Dies würde jeweils eine beispielhafte Implementierung und ein Mapping zum Datenmodell erfordern, um deren elektrischen Verhalten korrekt abzubilden und zu integrieren. In dem Framework wird lediglich der Fokus darauf gelegt, die Container an sich in einer SDN-Infrastruktur zu implementieren und deren Netzwerkschicht automatisiert zu integrieren. Eine Beispielimplementierung einer Schutzfunktion wurde bereits in Abschnitt 5.2.1.4 eingeführt und könnte um weitere Funktionen erweitert werden, um auch die Funktion der unterschiedlichen IEDs in das Framework zu integrieren.

Für den beschriebenen Funktionsumfang ist innerhalb der Validierung also vorzusehen, dass eine SCD-Beschreibung überführt und beispielhafte Kommunikationsservices gestartet werden können. Zu diesem Zweck wird eine vereinfachte SCD-Datei verwendet und als Eingang für das Framework herangezogen. Für die Gültigkeit des Frameworks wird anschließend das erzeugte Netzwerk anhand der instanziierten Container und einem beispielhaft gestarteten GOOSE-Publisher-Service und MMS-Client-Server-Verbindung validiert.

6.2 Durchführung der Validierung

Die eingeführten Aussagen müssen über die jeweils beschriebenen Wege validiert werden. In den folgenden Abschnitten werden diese Validierungen durchgeführt, und die Ergebnisse werden beschrieben und analysiert, um eine abschließende Aussage über die Gültigkeit des vorliegenden Co-Simulationsansatzes für die Stationsautomatisierung zu ermöglichen. Die Abschnitte sind direkt den zuvor eingeführten Beschreibungen zugeordnet und können entsprechend referenziert werden.

Für die Testaufbauten im Rahmen der Validierungen wurde Rechentechnik mit folgenden technischen Spezifikationen für den KNWS genutzt:

- Hardware: Dell EMC Power Edge R640
- Prozessor: 2x Intel Xeon Gold 6146, 3.2 GHz, 12C/24T
- Arbeitsspeicher: 64 GB (4 x 16 GB RDIMM, 2666 MT/s)
- Netzwerk: Intel X710 Dual Port 10GbE und i350 Dual Port 1GbE
- Betriebssystem: Ubuntu 20.04 LTS

6.2.1 Validierung der adäquaten Abbildung durch die Nutzung einer Virtualisierung

Zur Validierung der adäquaten Abbildung wird der Netzwerkverkehr gemäß dem in Abschnitt 6.1.1 beschriebenen Vorgehen verglichen. Dies geschieht auf der Grundlage einer implementierten Schutzauslösung des PTOC und erfolgt anhand der drei Evaluationsszenarien aus Abbildung 6.3. Die notwendigen Merkmale des Netzwerkverkehrs sind Tabelle 6.2 zu entnehmen. Dementsprechend können für die Durchführung zwei aufgenommene Netzwerkverkehrs-Dateien im PCAP-Format verwendet werden. Es werden die beiden Aufbauten aus Abbildung 6.1 herangezogen. Die erzeugten Datensätze werden in der Form vergleichbar erzeugt, dass das abgebildete Szenario der Schutzauslösung reproduzierbar von der Stromnetzsimulation erzeugt wird. Die ausgehenden SV-Pakete werden über einen definierten Zeitraum identisch an den realen Testaufbau und anschließend an die virtuell abgebildete Infrastruktur gegeben. Die nachgebildeten Schutzsysteme des Herstellers ABB im Testaufbau sind an standardisierten Schutzfunktionen orientiert und implementiert. Details zu der Programmierung der realen Schutzsysteme sind nicht bekannt, wodurch eine gewisse Abweichung der Funktion erwartet werden kann. Jedoch wird die Ähnlichkeit der

implementierten Schutzlogiken als ausreichend hoch angenommen. Dementsprechend folgt in beiden Aufbauten nach einer bestimmten Zeit der Grenzwertüberschreitung des Stroms ein Schaltbefehl zur Schalterauslösung in der Stromnetzsimulation. Um das Datenmodell zwischen den virtuellen und realen IEDs möglichst identisch inklusive der Parametrierungen für Ziel-MAC-Adressen zu gestalten, wird die CID-Datei aus den realen IEDs exportiert und für den Build-Prozess der Netzwerkschicht für die vIEDs verwendet. Folgende maßgebliche Merkmale werden von den realen IEDs herangezogen, um die vIEDs zu implementieren:

- GOOSE-Quell-MAC-Adresse:
 - PIOC: 00:00:23:34:36:00
 - PTOC: 00:00:23:34:35:e7
- GOOSE-Ziel-MAC-Adresse:
 - PIOC: 01:0c:cd:01:00:01
 - PTOC: 01:0c:cd:01:00:02
- GOOSE-ID:
 - PIOC: ABB1LD0/LLN0.PIOC
 - PTOC: ABB2LD0/LLN0.PTOC

Die GOOSE-Quell-MAC-Adresse wird nicht aus dem CID extrahiert, sondern aus einem Mitschnitt ausgelesen und repräsentiert die MAC-Adresse des realen Adapters des IEDs. Diese MAC-Adresse kann dem Containernet-Skript direkt bei der Instanziierung des jeweiligen Containers übergeben werden, sodass der virtuelle Netzwerkadap-ter des Containers die gleiche MAC-Adresse erhält wie der reale Adapter des IEDs.

Neben den GOOSE-Konfigurationen sind die SV-Konfigurationen für die virtuelle Nachbildung essenziell. Dabei werden die Konfigurationen in beiden Aufbauten über den SNS vorgenommen, da dieser beide Systeme mit Messwerten speist. Demnach werden identische SV-IDs und SV-Ziel-MAC-Adressen parametrier-
t. Nur die SV-Quell-MAC-Adressen und die Anzahl der Netzwerkadap-ter unterscheiden sich, wie auch der Abbildung 6.1 zu entnehmen ist. So werden für den realen Aufbau zwei getrennte Netzwerkadap-ter verwendet, um jeweils einen SV-Stream zu übertragen und von dem jeweiligen IED die GOOSE-Pakete entgegenzunehmen. Im virtuellen Aufbau hingegen werden sämtliche Daten zwischen dem SNS und dem KNWS über einen physischen Netzwerkadap-ter übertragen, was zu einer Abweichung zwischen der

Co-Simulation und dem Realaufbau führt. Der Laboraufbau wird in Anlehnung an reale Umspannwerke gewählt, obwohl auch nur eine Verbindung für zwei SV-Publisher genutzt werden könnte. Folglich ergeben sich folgende Konfigurationen:

- SV-Quell-MAC-Adresse:
 - Reale Station: b4:96:91:54:21:be und b4:96:91:54:21:bf
 - Virtuelle Station: b4:96:91:58:00:6d
- SV-Ziel-MAC-Adresse:
 - Messung für PIOC: 01:0c:cd:04:01:01
 - Messung für PTOC: 01:0c:cd:04:01:02
- SV-ID:
 - Messung für PIOC: SV-ABB1-PIOC
 - Messung für PTOC: SV-ABB2-PTOC

Für beide Testaufbauten werden Datensätze in Form von PCAP-Dateien erstellt und beinhalten die aufgenommenen Netzwerkpakete an jeweils einem Mirrorport des zentralen Switches. In dem realen Aufbau wird hier direkt ein Mirrorport am Netzwerkschicht eingerichtet und der ausgeleitete Datenverkehr über ein separates Gerät aufgezeichnet. In der Virtualisierung wird der Verkehr an einem Port des OpenFlow-Switches verwendet und direkt von dem Virtualisierungsrechner aufgenommen.

Die erzeugten zwei PCAP-Dateien gleichen sich folglich in der zeitlichen Länge und werden durch die Berechnung der definierten Merkmale gleichartig weiterverarbeitet und können miteinander verglichen werden. Der Inhalt der PCAP-Dateien beschreibt primär den SV-Prozessbusverkehr, der von der Stromnetzsimulation erzeugt wird und ist ausschnittsweise in Tabelle 6.4 dargestellt. Ein zyklisch sendender GOOSE-Publisher macht generell den zweitgrößten Anteil des Netzwerkverkehrs in beiden Mitschnitten aus. Zum Zeitpunkt der Schalterauslösung wird ergänzend der bereits eingeführte Burst-Prozess initiiert. Hierbei wird das eigentlich periodisch auftretende Heartbeat-Verhalten der GOOSE-Publisher um eine sequenzielle Erhöhung des Übertragungsintervalls ergänzt. Dies soll im Fall einer Störung gezielt auf das eingetretene Ereignis hindeuten. Das Burst-Verhalten wird nach einer bestimmten Zeit wieder abgestellt und es verbleibt das Heartbeat-Verhalten [87].

Der Vergleich der beiden Datensätze anhand des Ähnlichkeitsmaßes ist in Tabelle 6.5 dargestellt. Der Durchschnittswert der verschiedenen Merkmale, also die berechnete Ähnlichkeit zwischen den Datensätzen, beträgt hieraus 0,75. Zur Interpretation

Tabelle 6.4: Ausschnitt einer aufgenommenen PCAP-Datei der virtuellen Station

Zeitstempel [s]	Quell-MAC	Ziel-MAC	Typ	Länge [byte]
12.191549	b4:96:91:58:00:6d	01:0c:cd:04:01:01	SV	122
12.191549	b4:96:91:58:00:6d	01:0c:cd:04:01:02	SV	122
12.191615	00:00:23:34:36:00	01:0c:cd:01:00:01	GOOSE	212
12.191799	b4:96:91:58:00:6d	01:0c:cd:04:01:01	SV	122
12.191825	b4:96:91:58:00:6d	01:0c:cd:04:01:02	SV	122
12.192049	b4:96:91:58:00:6d	01:0c:cd:04:01:01	SV	122
12.192049	b4:96:91:58:00:6d	01:0c:cd:04:01:02	SV	122
12.192299	b4:96:91:58:00:6d	01:0c:cd:04:01:01	SV	122
12.192323	b4:96:91:58:00:6d	01:0c:cd:04:01:02	SV	122
12.192549	b4:96:91:58:00:6d	01:0c:cd:04:01:01	SV	122
12.192549	b4:96:91:58:00:6d	01:0c:cd:04:01:02	SV	122
12.192741	00:00:23:34:36:00	01:0c:cd:01:00:01	GOOSE	212
12.192799	b4:96:91:58:00:6d	01:0c:cd:04:01:01	SV	122
12.192829	b4:96:91:58:00:6d	01:0c:cd:04:01:02	SV	122
12.193051	b4:96:91:58:00:6d	01:0c:cd:04:01:01	SV	122
12.193051	b4:96:91:58:00:6d	01:0c:cd:04:01:02	SV	122
12.193299	b4:96:91:58:00:6d	01:0c:cd:04:01:01	SV	122
12.193328	b4:96:91:58:00:6d	01:0c:cd:04:01:02	SV	122
12.193355	b4:96:91:58:00:6d	01:0c:cd:04:01:01	SV	122
12.193355	b4:96:91:58:00:6d	01:0c:cd:04:01:02	SV	122

des Ähnlichkeitswertes werden insbesondere die Merkmale mit geringer Ähnlichkeit betrachtet, die durch strukturelle Unterschiede zwischen den beiden Testaufbauten erklärbar sind.

Unter der Kategorie der Paket-basierten Merkmalen ist der Wert für die Quelle-Adresse geringer als die durchschnittliche Ähnlichkeit. Der Grund ist bereits durch die SV-Publisher beschrieben, wonach der reale Aufbau zwei getrennte Netzwerkadapter verwendet und bei der Virtualisierung lediglich ein Netzwerkadapter des SNS genutzt wird. Eine höhere Ähnlichkeit wird erreicht, wenn der SNS denselben Netzwerkadapter verwendet, der auch für die Virtualisierung genutzt wird. Der Testaufbau wurde in dieser Weise gewählt, da nach dem IEC 61850 Standard für jeden SV-Publisher ein separater Adapter vorgesehen werden sollte. Für die Co-Simulation hingegen wird eine einzelne Verbindung angestrebt, um eine hohe Flexibilität und Automatisierung in der Konfiguration erreichen zu können. Weiterhin weichen die SV-Quelle-Adressen der beiden Testumgebungen voneinander ab. Hierdurch ist strukturell eine Abweichung dieses Merkmals erwartbar. Dieses Merkmal kann von der Stromnetzsimulation nicht

Tabelle 6.5: Ergebnisse der Kosinus-Ähnlichkeit der verglichenen Merkmale zwischen dem Aufbau der realen und virtuellen Station

Kategorie	Merkmal	Ergebnis der Kosinus-Ähnlichkeit
Paket-basierte Merkmale	Zeitstempel	0,9883
	Quelle	0,5453
	Ziel	0,9496
	Paketlänge	0,996
Konversations-basierte Merkmale	Quelle	0,2754
	Ziel	0,3316
	Paketanzahl	0,9999
	Gesamtlänge	0,9999
	Paketrage	0,9999
	Byte-Rate	0,9999
Endpunkt-basierte Merkmale	Adressen	0,2004
	TX Paketanzahl	0,7071
	TX Bytes	0,7071
	RX Paketanzahl	0,5
	RX Bytes	0,5
Protokoll-basierte Merkmale	Protokoll	0,1181
	Paketanzahl	0,9999
	Gesamtlänge	0,9999
	Paketrage	0,9999
	Byte-Rate	0,9999
I/O-basierte Merkmale	Zeitstempel	1,0
	Paketanzahl	0,9986
	Bytes	0,9987

umgesetzt werden, außer wenn mehrere Verbindungen zwischen den Simulatoren aufgebaut und die MAC-Adressen des Stromnetzsimulators angeglichen werden.

Die gleiche Begründung ist für die Konversations-basierte Quellen zutreffend. Zusätzlich sind die Ähnlichkeiten der Merkmale der Quellen und Ziele geringer, da der Realaufbau nicht strikt von dem Labornetzwerk bei der Datenaufzeichnung getrennt werden kann. Dies ist begründet durch die PTP-Synchronisierung der einzelnen Geräte im Versuchsaufbau, wobei der PTP-Master nur über eine komplexere Laborinfrastruktur erreicht werden kann und am Prozessbus-Switch anliegt, wie in Abbildung 6.1 dargestellt wurde. Die Ankopplung an den PTP-Master ist im Netzwerkmitschnitt der Virtualisierung nicht sichtbar, da der Mitschnitt am virtuellen Switch des KNWS aufgenommen wird, wohingegen die Synchronisierung über einen weiteren Kanal um-

gesetzt ist. Folglich sind in dem Traffic-Mitschnitt des Realaufbaus noch zusätzliche Protokolle und Konversationen, wie PTP-Nachrichten, enthalten. Dies prägt sich in den Konversations-basierten Merkmalen stärker aus, da in der Paket-basierten Betrachtung der hohe Anteil der Kommunikation mit dem SV-Protokoll diesen Unterschied überlagert. Die zusätzlichen Konversationen zwischen Teilnehmern wirken sich ebenfalls auf den Ähnlichkeitswert der Ziel-Adressen aus. Dieser liegt somit unter dem Gesamtdurchschnitt von 0,75.

Bei den Endpunkt-basierten Merkmalen spielt vor allem auch das zusätzliche Vorhandensein von Endpunkten der Labor-Infrastruktur und die Unterschiede der SV-Quell-Adressen eine Rolle, um den Wert des Merkmals der Adressen zu begründen. Die unterschiedliche Anzahl an Endpunkten lässt sich hier ableiten und bewirkt auch die Ergebnisse der anderen Endpunkt-basierten Merkmale, die unter der durchschnittlichen Ähnlichkeit von 0,75 liegen.

Das Merkmal „Protokoll“ hat ebenfalls einen unter dem Durchschnitt von 0,75 befindlichen Wert in der Kategorie der Protokoll-basierten Merkmale. Die Begründung hierfür ist identisch zu der zuvor beschriebenen, nicht vollständigen Abkoppelung von der Laborumgebung. Aus den zusätzlichen Verbindungen und Endpunkten resultieren auch zusätzliche vorhandene Protokolle im Mitschnitt des Realaufbaus. Insgesamt sind in dem Realaufbau sieben verschiedene Protokolle vorhanden, wohingegen die virtuelle Station ausschließlich die SV- und GOOSE-Pakete sowie eine geringe Anzahl von sonstigen User Datagram Protocol (UDP)-Paketen beinhaltet. In dem Realaufbau wurden neben PTP noch die Protokolle Spanning Tree Protokoll (STP), Link Layer Discovery Protokoll (LLDP), Internet Group Management Protokoll (IGMP) und Multicast Domain Name System Protokoll (MDNS) mit aufgezeichnet. Das zusätzliche Vorhandensein dieser Protokolle ist ein Grund für den geringen Ähnlichkeitswert.

Insgesamt zeigt die Auswertung der unterschiedlichen Merkmale des Netzwerkverkehrs, dass die Virtualisierung das Netzwerkverhalten eines Realaufbaus gut nachbilden kann. Einige strukturelle Unterschiede in den beiden betrachteten Testaufbauten bestehen jedoch aufgrund der gewollten singulären Direktverbindung zwischen den Simulatoren innerhalb der Co-Simulation. Der Laboraufbau stützt sich dabei auf den Aufbau einer möglichst realitätsnahen Digitalen Station mit separaten Netzwerkkad-aptern für jede MU und somit SV-Stream. Eine vollständige Separation der umliegenden Laborumgebung war hingegen im Testaufbau nicht möglich. Die dargestellten Ergebnisse zeigen die erwartete Abweichung. Betrachtet man aber das Netzwerkverhalten auf dem Prozessbus mit den maßgeblichen Prozessbus-Protokollen, kann gefol-

gert werden, dass die Virtualisierung sehr hohe Ähnlichkeitswerte im Vergleich zum Realaufbau aufweist und das Verhalten gut nachbildet. Maßgeblich hierfür ist eine sehr gute Beschreibung des realen Aufbaus durch eine vollständige SCD-Datei der Station. Hierdurch lassen sich die Komponenten und Strukturen hinreichend genau nachbilden.

6.2.2 Validierung der Synchronität der Simulatoren ohne spezifische Koordinator-Umsetzung

Die Korrektheit des Datenaustauschs, dessen Validierungsvorgehen in Abschnitt 6.1.2 beschrieben wurde, kann aus den Testdaten und nachfolgend dargestellten Simulationsergebnissen des Szenarios (iii) entnommen werden. Ein beispielhafter Simulationsmitschnitt der Stromnetzsimulation mit ausgeführter Schalthandlung ist in Abbildung 6.7 dargestellt. Daraus lässt sich ableiten, dass die Ereignisse und Netzwerkpakete korrekt verarbeitet werden und eine funktionierende Schutzauslösung ermöglichen. Es sind keine Ungenauigkeiten oder Synchronisierungsfehler erkennbar. Die PTP-Synchronisierung muss in Anlehnung an [100] lediglich zwischen den physischen Geräten, also der SNS- und KNWS-Hardware geprüft und sichergestellt werden. Hier wird im Rahmen der Arbeit lediglich auf die Verwendung von einer frei verfügbaren, etablierten PTP-Implementierung verwiesen, die durch korrekte Nutzung des Synchronisierungsprotokolls und Weitergabe des verteilten Zeitsignals eine Zeitgleichheit der Geräte sicherstellt. Eine detaillierte Analyse der Zeitstempel ist hier nicht notwendig, da die gesammelten Testdaten aus den anderen Validierungskategorien darauf hindeuten, dass sie korrekt verwendet wurden. Zudem sind die Zeitstempel beider Simulatoren konsistent. Ein Auszug aus einer aufgenommenen PCAP-Datei ist in Tabelle 6.4 dargestellt und zeigt die synchronen Netzwerkpakete anhand der Darstellung der Zeitstempel und belegt somit die Korrektheit der Synchronisierung.

6.2.3 Validierung der Anforderungen an die IEC 61850 Übertragungszeit

Das eingeführte Modell der Virtualisierung in den drei Evaluationsszenarien dient als generelle Analyse der möglichen Einhaltung der Übertragungszeit, wie es in Abschnitt 6.1.3 erläutert wurde. In Bezug auf das aufgestellte Verzögerungsmodell ist zu erwähnen, dass im Evaluationsszenario (iii) das GOOSE-Paket für den OpenFlow-Switch bekannt ist. Demnach ist bereits ein Eintrag in der Flow-Tabelle des Switches hinter-

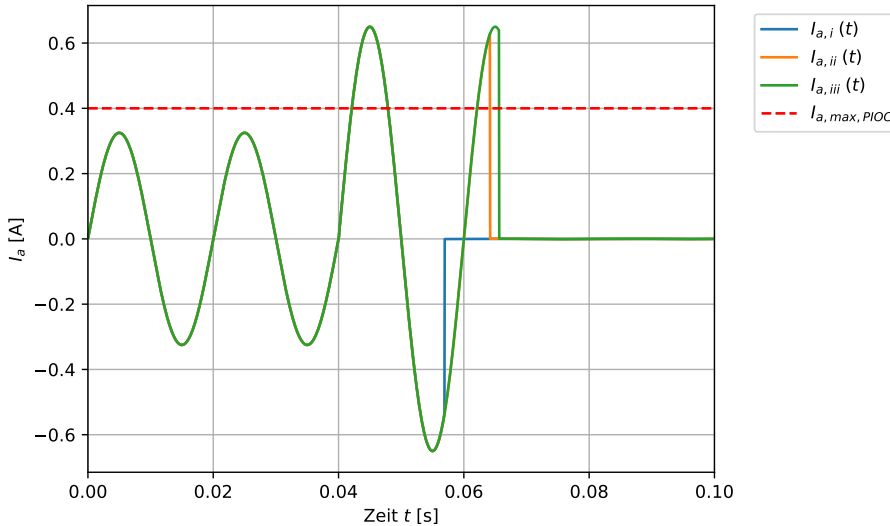


Abbildung 6.7: Stromverläufe in den drei Evaluationsszenarien mit Grenzwertüberschreitung und erfolgter Schalthandlung

legt und keine Interaktion mit dem Controller notwendig. Ein möglicher Unterschied der Ergebnisse zu dem Fall einer Controller-Interaktion ist in dem nachfolgenden Validierungsgegenstand und der Untersuchung der Übertragungszeitanteile dargestellt.

Als erste Ebene wird die RTT zwischen der Strom-Grenzwertüberschreitung I_a an der Phase a und dem folgenden Schaltbefehl in allen drei Evaluationsszenarien untersucht, wie es in Abbildung 6.7 gezeigt wird.

Die drei Stromverläufe sind nach den drei Evaluationsszenarien mittels Indexierung dargestellt. Eine Zunahme der RTT ist in den Szenarien (ii) und (iii) identifizierbar. Diese beiden Szenarien nutzen die Virtualisierung und die IED-Funktion läuft in einem Container. Bei Szenario (i) hingegen wird die Funktion direkt auf dem Host ausgeführt. Der Unterschied zwischen den Szenarien (ii) und (iii) ist nach Abbildung 6.7 hingegen geringer. Der Vergleich der RTT und der Übertragungszeit basiert darauf, dass die Simulationsschrittweite der Stromnetzsimulation eingehalten wird. Außerdem werden die Zeitstempel vor der Übergabe der Schaltinformation an die Netzwerkschicht des IEDs genutzt. Für die Berechnung gilt der Zusammenhang,

$$\Delta t_{RTT} = 3\Delta t_{SNS} + 2\Delta t_{Net} + \Delta t_{PD}, \quad (6.2)$$

mit

Δt_{RTT}	Zeitkomponente für die RTT,
Δt_{SNS}	Simulationsschrittweite innerhalb des SNS,
Δt_{Net}	Zeitkomponente auf dem Weg zwischen den Simulatoren auf dem Netzwerkkabel und
Δt_{PD}	Zeitkomponente für die Netzwerk- und Funktionsschicht des IEDs.

Die Ergebnisse der verschiedenen Komponenten der Verzögerungen in den drei Szenarien sind in Tabelle 6.6 dargestellt. Der Verzögerungsanteil des IEDs muss für eine korrekte Interpretation und Berechnung der Übertragungszeit näher beschrieben werden. Über die Trennung der Zeitkomponenten der Netzwerk- ($\Delta t_{PD,N}$) und Funktionsanteile ($\Delta t_{PD,F}$) des physischen Geräts (PD) gilt für Δt_{PD} unter Vernachlässigung möglicher weiterer interner Prozesse im IED und der Annahme, dass die Netzwerkschicht für das Ver- und Entpacken von Daten die gleiche Zeit benötigt:

$$\Delta t_{PD} = 2\Delta t_{PD,N} + \Delta t_{PD,F}. \quad (6.3)$$

Dabei werden zunächst sämtliche Zeiteinflüsse von der Virtualisierung unter $\Delta t_{PD,N}$ zusammengefasst und noch keine Trennung nach dem Verzögerungsmodell vorgenommen. Im Szenario (ii) wird bei dem IED ohne SDN-Umgebung der Container mittels der Netzwerkbridge direkt mit dem realen Adapter verbunden, wodurch im Vergleich zum ersten Szenario ein geringer Unterschied erwartbar ist. Es kann die Beziehung für die Messung der Übertragungszeit Δt_{TT} aus der RTT gefolgert werden, die sich als

$$\Delta t_{TT} = \Delta t_{SNS} + \Delta t_{Net} + \Delta t_{PD,N} \quad (6.4)$$

beschreiben lässt.

Die dargestellten Ergebnisse zeigen für Δt_{TT} in Tabelle 6.6, dass die nach IEC 61850-5 definierte Anforderung an die Übertragungszeit, wie in Kapitel 3.3.2 ausgeführt, von drei Millisekunden in allen drei Evaluationsszenarien eingehalten werden kann. Die Co-Simulation erfüllt somit für den Testaufbau die Anforderung. Der Vergleich mit einer Betrachtung des Unternehmens ABB [102] zeigt beispielsweise eine RTT von 68 ms für eine vollständige Störbeseitigung mit aktuellen Schutzgeräten.

Der Einfluss der Virtualisierung mit SDN-Umgebung spiegelt sich in den ermittelten Ergebnissen wider. Hier können die Szenarien (i) und (ii) zusammengefasst

Tabelle 6.6: Ergebnisse der Szenarien (i), (ii) und (iii)

Zeitkomponente	Szenario (i)	Szenario (ii)	Szenario (iii)
Δt_{RTT} [ms]	14,80	22,05	23,50
Δt_{PD} [ms]	14,51	21,76	23,21
$\Delta t_{PD,N}$ [ms]	0,03	0,03	0,82
$\Delta t_{PD,F}$ [ms]	14,45	21,07	21,33
Δt_{TT} [ms]	0,15	0,15	0,94

werden und weisen im Vergleich zum Szenario (iii) einen deutlichen Unterschied auf. So ist die Übertragungszeit ungefähr neunmal so groß beim Szenario (iii). Die erlaubte Grenze von drei Millisekunden lässt jedoch genügend Spielraum für die folgenden Untersuchungen von komplexeren Infrastrukturen innerhalb der SDN-Umgebung.

Aus den Ergebnissen wird jedoch auch deutlich, dass die Containerisierung in den Szenarien (ii) und (iii) einen starken Einfluss auf die RTT hat. Der Verzögerungsanteil, der durch die Funktionsschicht der vIEDs verursacht wird, ist erheblich und lässt die Verzögerungskomponente $\Delta t_{PD,N}$ um ungefähr 50 % ansteigen, was auf die verminderte Rechenleistung der Container zurückzuführen ist. Auf eine mögliche Einschränkung auf den Zeitbedarf der Funktionsschicht wird hier nicht näher eingegangen, da diese stark von der Komplexität der IED-Funktionalität abhängig ist. Komplexere Berechnungen und Funktionen steigern den Rechenbedarf, wodurch zum einen höhere Verzögerung erwartbar sind. Andererseits kann der höhere Rechenaufwand die Container an ihre Belastungsgrenzen heranführen, was sich wiederum in zusätzlichen Verzögerungen äußert. Die Effekte werden im Rahmen der Arbeit explizit nicht näher untersucht. [80]

6.2.4 Validierung der mathematischen Berechnung von auftretenden Verzögerungen

Die Durchführung der Tests nach Tabelle 6.3 zur Validierung des aufgestellten Verzögerungsmodells ist in Abschnitt 6.1.4 ausgeführt. Es ergeben sich sechs verschiedene Abbildungen, die jeweils Ergebnisse der Übertragungszeit in Abhängigkeit variierender Parameter der Testkonfigurationen zeigen. Dabei wird jeweils die Anzahl der Switches unter und zusätzlich eine weitere Größe variiert, wie in Tabelle 6.3 vermerkt ist. Demnach werden die beschriebenen Testdatensätze nachfolgend dargestellt und anschließend die Ergebnisse mit dem mathematischen Modell nach Kapitel 5.4 verglichen.

Für den vorliegenden Testaufbau wird die Einschränkung getroffen, dass lediglich die Zeitkomponenten innerhalb des KNWS untersucht sind. Demnach beinhaltet der Testaufbau für die Validierung des mathematischen Modells lediglich den KNWS, ohne eine Anbindung an den SNS mitsamt simulierter Messwerte. So wird der Orchestrierungsaufwand für die Testdurchführungen mit einer Zeitmessung über beide Simulatoren minimiert. Die Anteile der Übertragungen vom KNWS hin zum SNS werden hier nicht gesondert betrachtet, da das aufgestellte Verzögerungsmodell primär für die Untersuchung der SDN-Einflüsse aufgestellt worden ist. Die zusätzlichen Komponenten der RTT wurden in der vorherigen Kategorie bereits exemplarisch gezeigt und sind unbeeinflusst von der Struktur der virtuellen Netzwerkinfrastruktur auf dem KNWS.

Die einzelnen Testdatensätze sind als Comma-Separated Values (CSV)-Dateien erstellt, die die ermittelten Zeitstempel (t_k) aus den PCAP-Dateien und dem erzeugten Zeitstempel vor Verpacken der Schaltinformation in das GOOSE-Protokoll enthalten. Zusätzlich sind die einzelnen Zeitkomponenten als Differenzen zwischen den Zeitstempeln als weitere Elemente in der CSV-Datei enthalten. Das Schema eines Datensatzes einer Testdurchführung ist in Tabelle 6.7 dargestellt.

Die Testdurchführungen zeigen eine Streuung der Ergebnisse bezüglich der messbaren Übertragungszeiten und Zeitkomponenten. Folglich wird bei zwei Messungen nicht zwangsläufig das gleiche, messbare Ergebnis erzielt. Die Streuungen bei mehreren Testläufen bei einer Testkonfiguration zeigt einen vom Verzögerungsmodell nicht abgedeckten Einfluss. Dieser kann mit unterschiedlichen Auslastungssituationen des KNWS, der Netzwerkauslastungen oder Scheduling-Schritten externer Prozesse erklärbar sein. Die Untersuchung der Einflüsse steht nicht im Vordergrund für diese Arbeit.

Für die Betrachtung der Einflussparameter, die von dem mathematischen Modell abgedeckt werden, werden je Testkonfiguration, also beispielsweise einer Kombination aus Switch- und Container-Anzahl, mehrere Messungen durchgeführt. Jede Durchführung wird durch eine komplett neue Initialisierung und ein neues Starten des Controllers, Containernet und der einzelnen Container, wie in Abbildung 6.6 dargestellt, erzielt. Um den Effekt und die Streuung zu verdeutlichen und zu beschreiben, bei welchen messbaren Zeitkomponenten der Übertragungszeit vornehmlich Streuungen auftreten, sind zunächst in Tabelle 6.8 und Abbildung 6.8b die einzelnen Zeitkomponenten von fünf beispielhaften Testdurchführungen dargestellt. Die Daten wurden nach der ersten Validierungskategorie aus Tabelle 6.3 mit einer Controller-Interaktion aufgenommen. Die Messwerte sind mit „...“ gekennzeichnet, da der zugrunde lie-

Tabelle 6.7: Berechnung und Beschreibung der einzelnen Verzögerungskomponenten für die Validierungsdatensätze

	Einheit	Berechnung	Bedeutung
$t_{eth,d}$	s	Messung	Zeitstempel am Netzwerkadapter des Containers
$t_{eth,s1}$	s	Messung	Zeitstempel am OpenFlow-Switch zur Verbindung des Container-Netzwerkadapters
$t_{lo,PI}$	s	Messung	Zeitstempel des Packet-In-Ereignis am Loopback Interface
$t_{lo,FM}$	s	Messung	Zeitstempel des Flow-Mod-Ereignis am Loopback Interface
$t_{lo,PO}$	s	Messung	Zeitstempel des Packet-Out-Ereignis am Loopback Interface
$t_{eth,s2}$	s	Messung	Zeitstempel am OpenFlow-Switch zur Verbindung des realen Netzwerkadapters
$t_{eth,PD}$	s	Messung	Zeitstempel am realen Netzwerkadapter
$t_{d,GSE}$	s	Messung	Erzeugter Zeitstempel vor Erstellen des GOOSE-Pakets
$\Delta t_{PO,PI}$	ms	$t_{lo,PO} - t_{lo,PI}$	Zeitdifferenz zwischen Packet-In- und Packet-Out am Loopback Interface
$\Delta t_{s2,s1}$	ms	$t_{eth,s2} - t_{eth,s1}$	Zeitdifferenz am OpenFlow Switch zwischen Ein- und Ausgang des GOOSE-Pakets
Δt_{PD}	ms	$t_{eth,PD} - t_{eth,d}$	Zeitdifferenz von dem Netzwerkadapter des Containers und dem realen Netzwerkadapter
$\Delta t_{PD,GSE}$	ms	$t_{eth,PD} - t_{d,GSE}$	Zeitdifferenz zwischen Erstellen des GOOSE-Pakets und Messung am realen Netzwerkadapter
$\Delta t_{d,GSE}$	ms	$t_{eth,d} - t_{d,GSE}$	Zeitdifferenz für das Verpacken der GOOSE-Nachricht
$\Delta t_{d,s}$	ms	$t_{eth,s1} - t_{eth,d}$	Zeitdifferenz zwischen dem OpenFlow-Switch und dem Container
$\Delta t_{s1,PI}$	ms	$t_{lo,PI} - t_{eth,s1}$	Zeitdifferenz zwischen Eingang des Pakets am OpenFlow-Switch und Eingang des Packet-In am Loopback Interface
$\Delta t_{PO,s2}$	ms	$t_{eth,s2} - t_{lo,PO}$	Zeitdifferenz zwischen der Packet-Out Nachricht am Loopback Interface und Eingang des Pakets am OpenFlow-Switch
$\Delta t_{s,PD}$	ms	$t_{eth,PD} - t_{eth,s2}$	Zeitdifferenz zwischen dem OpenFlow-Switch und dem realen Netzwerkadapter

Tabelle 6.8: Gemessene und berechnete Werte für fünf Durchgänge eines Datensatzes mit einem Switch und einem Container

	D1	D2	D3	D4	D5
$t_{eth,d}$... 5,310408	... 7,314591	... 9,364703	... 1,245960	... 3,164955
$t_{eth,s1}$... 5,310414	... 7,314597	... 9,364707	... 1,245966	... 3,164958
$t_{lo,PI}$... 5,310786	... 7,315105	... 9,365203	... 1,246945	... 3,165735
$t_{lo,FM}$... 5,312611	... 7,317021	... 9,366952	... 1,248774	... 3,167420
$t_{lo,PO}$... 5,312676	... 7,317086	... 9,367018	... 1,248840	... 3,167484
$t_{eth,s2}$... 5,312806	... 7,317152	... 9,367181	... 1,249051	... 3,167687
$t_{eth,PD}$... 5,312816	... 7,317160	... 9,367193	... 1,249063	... 3,167698
$t_{d,GSE}$... 5,310214	... 7,314395	... 9,364520	... 1,245758	... 3,164794
$\Delta t_{PO,PI}$	1,889331	1,981027	1,814956	1,895103	1,748991
$\Delta t_{s2,s1}$	2,392293	2,555324	2,474080	3,085624	2,728693
Δt_{PD}	2,408207	2,569170	2,490336	3,102364	2,744006
$\Delta t_{PD,GSE}$	2,602195	2,765792	2,664543	3,305176	2,904756
$\Delta t_{d,GSE}$	0,193988	0,196622	0,174207	0,202812	0,160750
$\Delta t_{d,s}$	0,005930	0,005734	0,004607	0,005128	0,004019
$\Delta t_{s1,PI}$	0,372889	0,508126	0,495761	0,978897	0,776900
$\Delta t_{PO,s2}$	0,130073	0,066171	0,163363	0,211624	0,202802
$\Delta t_{s,PD}$	0,009984	0,008112	0,011649	0,011612	0,011294

gende absolute Zeitstempel für die Auswertung unerheblich ist. Zusätzlich ist eine statistische Auswertung der gemessenen Verzögerungskomponenten mit dem Mittelwert \bar{x} , dem Median M , der Standardabweichung s und der Varianz s^2 in Tabelle 6.9 eingeführt.

Zu sehen sind besonders Abweichungen in den Zeitkomponenten, die eine Verbindung zwischen dem Switch und dem Controller beschreiben, wohingegen die Verzögerungskomponenten zwischen den Netzwerkadaptern und dem Switch relativ konstant ausfallen. Besonders die Komponente $\Delta t_{s1,PI}$ weist eine hohe Abweichung zwischen den Testdurchführungen auf, wie in der Standardabweichung und Varianz aus Tabelle 6.9 zu sehen ist. Hier tritt der Abgleich innerhalb des Switches zwischen der empfangenen Nachricht und der Flow-Tabelle auf, inklusive der gegebenenfalls vorhandenen Warteschlange am Switch selbst. Dieser Einfluss der Komponente $\Delta t_{s1,PI}$ ist dann in den zusammengesetzten Komponenten $\Delta t_{PD,GSE}$, Δt_{PD} und $\Delta t_{s2,s1}$ enthalten.

Der Median von jeweils fünf Durchläufen wird für die Auswertung gewählt, da in den Datensätzen eine hohe Varianz festgestellt wurde. Durch die Verwendung des Medians wird der Einfluss von Ausreißern, insbesondere bei einer geringen Stichpro-

Tabelle 6.9: Statistische Auswertung der Verzögerungskomponenten aus Tabelle 6.8

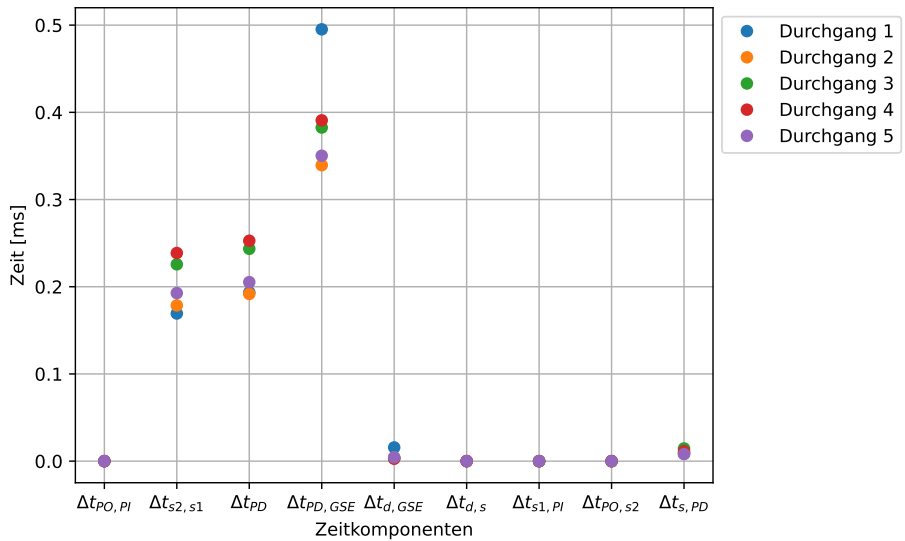
	\bar{x}	M	s^2	s
$\Delta t_{PO,PI}$	1,865681	1,889331	0,007975	0,0892
$\Delta t_{s2,s1}$	2,626823	2,474080	0,071893	0,2681
Δt_{PD}	2,642217	2,490336	0,056546	0,2377
$\Delta t_{PD,GSE}$	2,664172	2,765792	0,027572	0,1653
$\Delta t_{d,GSE}$	0,193195	0,193988	0,000129	0,0114
$\Delta t_{d,s}$	0,005523	0,005734	0,000000	0,0003
$\Delta t_{s1,PI}$	0,626753	0,495761	0,042418	0,2060
$\Delta t_{PO,s2}$	0,158626	0,130073	0,002070	0,0455
$\Delta t_{s,PD}$	0,010074	0,011294	0,000000	0,0010

bengröße verringert, da sie den Median weniger stark beeinflussen als den Mittelwert [104]. Daher basieren die folgenden Abbildungen und Diskussionen auf den Medianen.

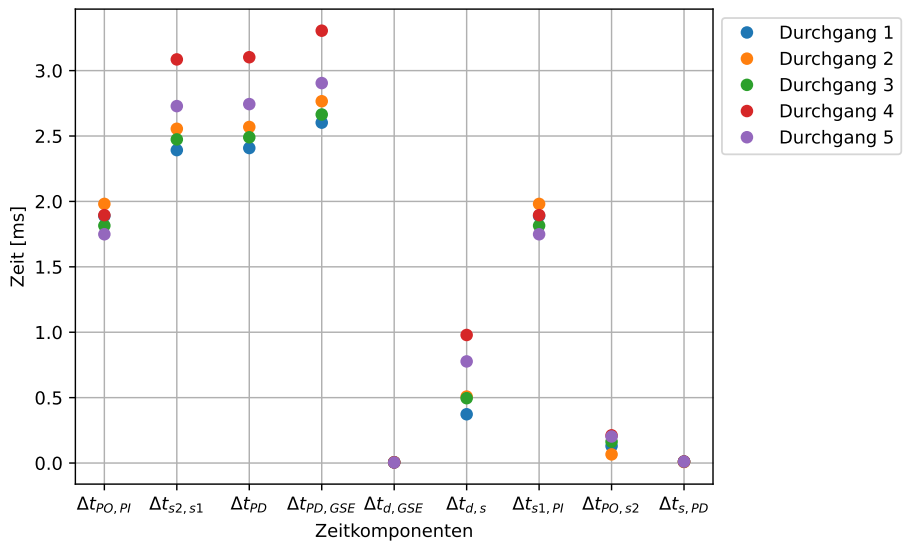
Zunächst wird der Einfluss der Anzahl der Container pro Switch und die Anzahl der Switches untersucht, um das Verzögerungsmodell experimentell zu parametrieren. Hierzu wird die Anzahl der SV-Publisher sowie die Packet-In-Rate auf null gesetzt. Demnach tritt ausschließlich Datenverkehr in Form des GOOSE-Pakets auf, ohne jegliche weitere Beanspruchung der Switches oder des Controllers durch andere Prozesse. Es wird jeweils ein Test mit und ohne Controller-Interaktion durchgeführt. Auf Basis der aufgenommen Messungen können die notwendigen Modellparameter für das Gesamtverzögerungsmodell aus Gleichung (5.8) extrahiert werden.

Für die Bestimmung der Servicerate steht im Versuchsaufbau lediglich die Zeitkomponente $\Delta t_{s2,s1}$ zur Verfügung, die messtechnisch erfasst werden kann und die Switch-Aktivität umfasst. Diese Komponente beinhaltet jedoch auch die Verzögerungen durch die Warteschlange. Daher kann die Servicerate nicht isoliert von der Warteschlange aus den Messdaten abgeleitet werden. Um eine Verzögerungskomponente, die um die Warteschlange bereinigt ist, für den Bedienerprozess des Switches zu definieren, wird ein Referenzwert aus [105] herangezogen:

$$\mu_s = \frac{1}{0,006 \text{ ms}} = 166.666,67 \frac{1}{\text{s}} \quad (6.5)$$



(a) Zeitkomponenten der Übertragungszeit ohne Controller-Aktivität



(b) Zeitkomponenten der Übertragungszeit mit Controller-Aktivität

Abbildung 6.8: Zeitkomponenten der Übertragungszeit bei fünf Testdurchführungen mit identischen Testparametern (oben: ohne Controller-Aktivität, unten: mit Controller-Aktivität nach Tabelle 6.8)

Ebenso stehen für die Ableitung der Servicerate des verwendeten Controllers nicht ausreichend Informationen aus den Messwerten zur Verfügung. Die Messung der Verzögerungskomponente $\Delta t_{PO,PI}$ durch die Auswertung der Packet-In- und Packet-Out-Pakete am Loopback-Interface lässt zwar eine Näherung zu, beinhaltet aber wiederum neben der Verarbeitung auch die Warteschlange des Controlllers. Aus diesem Grund wird auch hier eine Parametrierung aus [105] verwendet:

$$\mu_c = \frac{1}{0,23 \text{ ms}} = 4347,82 \frac{1}{\text{s}} \quad (6.6)$$

In der Folge wird jeweils das berechnete Ergebnis des Verzögerungsmodells einbezogen. Die zu untersuchende Zeitkomponente des Modells wird als $\Delta t_{PD,GSE}$ definiert, wodurch sich das Gesamtmodell gemäß Gleichung 6.7 zu folgendem Term reduziert:

$$\begin{aligned} \Delta t_{PD,GSE,math} = & \Delta t_{eth} + \Delta t_{prop,eth} + 2\Delta t_{prop,SDN} + \frac{1}{\mu_S} + \frac{1}{\mu_S - \lambda_S} \\ & + \frac{1}{\mu_C - n \cdot \lambda_S \cdot q_S^{pc}} + 2\Delta t_{veth} + \Delta t_{prop,CN} + \Delta t_{vIED} \end{aligned} \quad (6.7)$$

Im jeweils zweiten Pfad findet keine Controller-Aktivität statt. Das GOOSE-Paket ist bereits bekannt, und die Verbindung ist als Flow-Regel in der Tabelle der Switches hinterlegt. In diesem Fall reduziert sich die Gleichung (6.7) für die Übertragungszeit zu:

$$\Delta t_{PD,GSE,math} = \Delta t_{eth} + \Delta t_{prop,eth} + \frac{1}{\mu_S - \lambda_S} + 2\Delta t_{veth} + \Delta t_{prop,CN} + \Delta t_{vIED} \quad (6.8)$$

Für weitere Parametrierungen der Gleichungen (6.7) und (6.8) werden die Daten aus Tabelle 6.8 verwendet, und die Mediane für folgende Zeitkomponenten ermittelt:

$$\begin{aligned} \Delta t_{eth} + \Delta t_{prop,eth} &= \Delta t_{s,PD} = 0,011612 \text{ ms} \\ \Delta t_{prop,CN} + \Delta t_{veth} &= \Delta t_{s1,PI} = 0,495761 \text{ ms} \\ \Delta t_{vIED} + \Delta t_{veth} &= \Delta t_{d,GSE} = 0,193988 \text{ ms} \end{aligned} \quad (6.9)$$

Die Gleichungen (6.7) und (6.8) dienen dazu, in den verschiedenen Validierungskategorien den Vergleich zwischen dem mathematischen Modell und den Testdatensätzen herzustellen. Im mathematischen Modell werden analog zu den Kategorien die zugehörigen Parameter variiert, um vergleichbare Beziehungen abzubilden. Besonders

hervorzuheben sind die Parameter der Übergabewahrscheinlichkeit und Ankunftsrate aus dem Verzögerungsmodell, die in den Testdatensätzen durch Variation der Anzahl der SV-Publisher und der generierten Pakete hergestellt werden. Der Zusammenhang zu den Modell-Parametern wird in der jeweiligen Validierungskategorie in Tabelle 6.3 beschrieben.

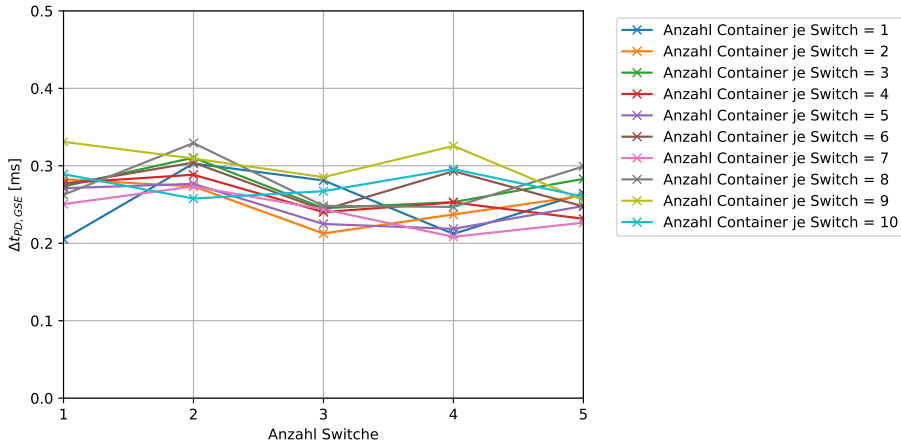
6.2.4.1 Kategorie 1

Die Testdaten für die erste Validierungskategorie des Verzögerungsmodells aus Tabelle 6.3 zeigen die Abbildungen 6.9a und 6.9b, indem jeweils die Komponente $\Delta t_{PD,GSE}$ in Abhängigkeit der Anzahl der Switches gemessen wurde. Es werden, wie vorher beschrieben, jeweils Mediane von fünf Testdurchführungen dargestellt. Die Abbildung 6.9a zeigt den Fall ohne Controller-Interaktion, also bei bekanntem GOOSE-Paket, wohingegen die Abbildung 6.9b den Fall mit Controller-Interaktion veranschaulicht.

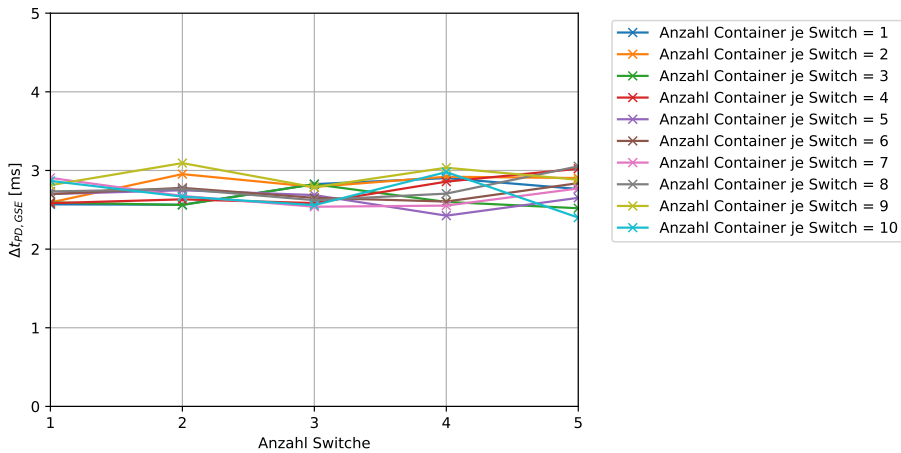
Die dargestellten Messungen zeigen, dass die Übertragungszeit größer ist, wenn der Controller durch eine Packet-Out- und Flow-Mod-Aktivität ins Spiel kommt. Sowohl mit als auch ohne Interaktion lässt sich kein klarer Zusammenhang zwischen der Container- und Switch-Anzahl ausmachen. Hier tritt lediglich eine Streuung zwischen den einzelnen Testdurchführungen auf.

Abbildung 6.10 zeigt die Ergebnisse des mathematischen Modells, bei dem die Anzahl der Container nicht direkt in die Gleichungen (6.7) und (6.8) einfließt, wodurch auch kein Zusammenhang erwartbar ist. Infolgedessen wird die Anzahl der Switches erhöht, wobei angenommen wird, dass die Ankunftsrate am jeweiligen Switch und die Übergabewahrscheinlichkeit an den Controller vernachlässigbar gering sind. Dementsprechend wurde das Modell mit den Annahmen $\lambda_S = q_S^{pc} = 0,001$ parametrisiert, um nicht durch null zu teilen.

Auswertung: Die dargestellten Testdatensätze zeigen eine nicht vorhandene Abhängigkeit der Übertragungszeit von der Anzahl der instanziierten Container, wenn diese jeweils nur vernachlässigbar geringen Verkehr an Paketen erzeugen. Dies entspricht dem aufgestellten Gesamtmodell der Übertragungszeit. Zum Vergleich der Zeitbereiche wurde der Mittelwert aus den verschiedenen Messungen aus den Abbildungen 6.9a und 6.9b als Stichprobentest gebildet. Im Fall ohne Controller-Aktivität ergibt sich ein Mittelwert von 0,2651 ms. Laut parametrisierten Modell ist ein Wert von 0,2116 ms zu erwarten. Eine prozentuale Abweichung von 25,3 % ist zu verzeichnen. Im Fall der Controller-Aktivität ergibt sich ein Mittelwert von 2,7352 ms, wobei das Modell einen Wert von 2,1817 ms vorgibt. Eine Abweichung von ebenfalls ca. 25,3 %



(a) $\Delta t_{PD,GSE}$ ohne Controller-Interaktion



(b) $\Delta t_{PD,GSE}$ mit Controller-Interaktion

Abbildung 6.9: $\Delta t_{PD,GSE}$ in Abhängigkeit der Anzahl verwendeter Switches und Container in Validierungskategorie 1

ist festzustellen. In dieser Betrachtung wurde die Streuung der einzelnen Testdurchführungen durch die Bildung der Mediane bereits berücksichtigt. Abschließend kann festgehalten werden, dass für die Validierungskategorie eins das aufgenommene Verhalten der Übertragungszeit dem mathematischen Verzögerungsmodell entspricht und dieses unter Berücksichtigung von Streuung einzelner Messwerte und der berechneten Abweichungen gültig ist. Die Abweichung der Stichproben weisen allerdings auf eine

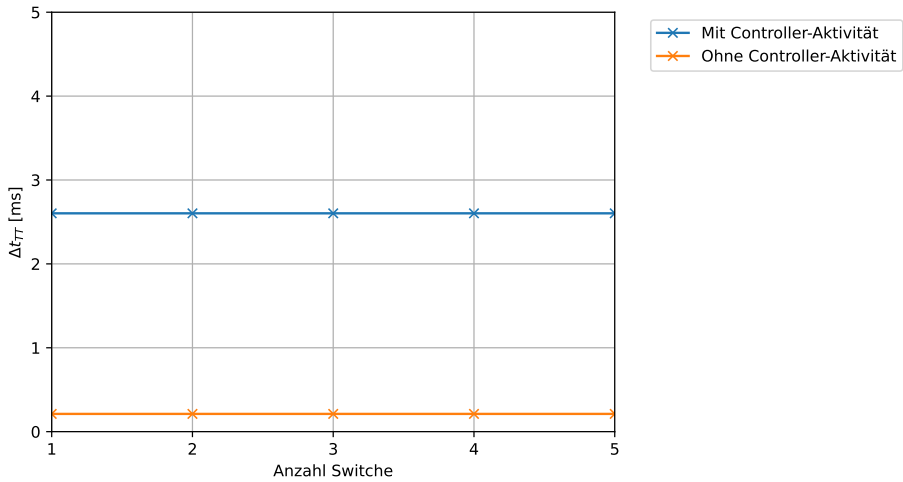
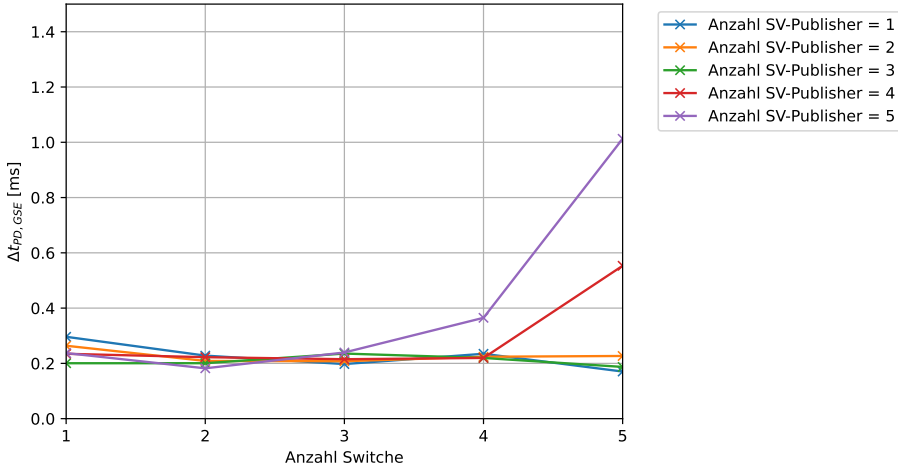


Abbildung 6.10: Δt_{TT} in Abhängigkeit der Anzahl der Switches in den Fällen mit und ohne Controller-Aktivität unter Nutzung des mathematischen Verzögerungsmodells

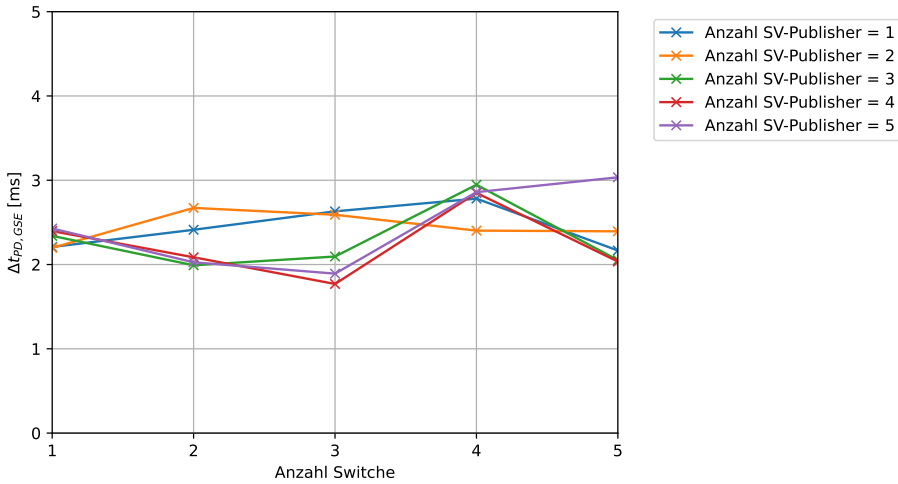
nicht optimale Modellparametrierung in Bezug auf die Servicerate vom Switch hin, die in beiden Fällen von gleicher Bedeutung ist.

6.2.4.2 Kategorie 2

Die Testdaten für die zweite Validierungskategorie des Verzögerungsmodells aus Tabelle 6.3 sind in Abbildung 6.11a und 6.11b visualisiert, indem wiederum die Komponente $\Delta t_{PD,GSE}$ in Abhängigkeit der Anzahl der Switches gemessen wurde. Abbildung 6.11a zeigt den Fall ohne Controller-Interaktion, also bei bekanntem GOOSE-Paket, wohingegen Abbildung 6.11b den Fall mit Controller-Interaktion darstellt. Farblich unterschieden ist die implementierte Anzahl von SV-Publishern. In der nachfolgenden Betrachtung sind jeweils sechs Container an einen Switch angeschlossen, wovon maximal fünf als einzelne SV-Publisher genutzt werden können. Der sechste Container ist nicht als SV-Publisher implementiert, um das Senden des GOOSE-Pakets aus diesem ohne zusätzliche Einflüsse beispielsweise durch das Senden von SV-Paketen zu untersuchen. Die SV-Publisher generieren hier die Ankunftsrate λ_i an einem OpenFlow-Switch analog zu der Verzögerungsmodellierung. Es werden die SV-Publisher-Implementierungen nach IEC 61850-9-2LE unter der Abtastung von 80 Werten je Sinus des Wechselstroms, also 4 kHz, verwendet. Diese Abtastung entspricht der Ankunftsrate an den Switchen.



(a) $\Delta t_{PD,GSE}$ ohne Controller-Interaktion



(b) $\Delta t_{PD,GSE}$ mit Controller-Interaktion

Abbildung 6.11: $\Delta t_{PD,GSE}$ in Abhängigkeit der Anzahl verwendeter Switches und SV-Publisher in Validierungskategorie 2

Die Testdaten aus Abbildung 6.11a zeigen jeweils eine ungefähr konstante Verzögerung analog zu Abbildung 6.9a bei geringer Gesamt-Paket-Rate, also bei geringer Anzahl von Switches und SV-Publishern. Lediglich die Verläufe bei Nutzung von vier und fünf SV-Publishern stellen ein signifikant ansteigendes Verhalten bei Erhöhung

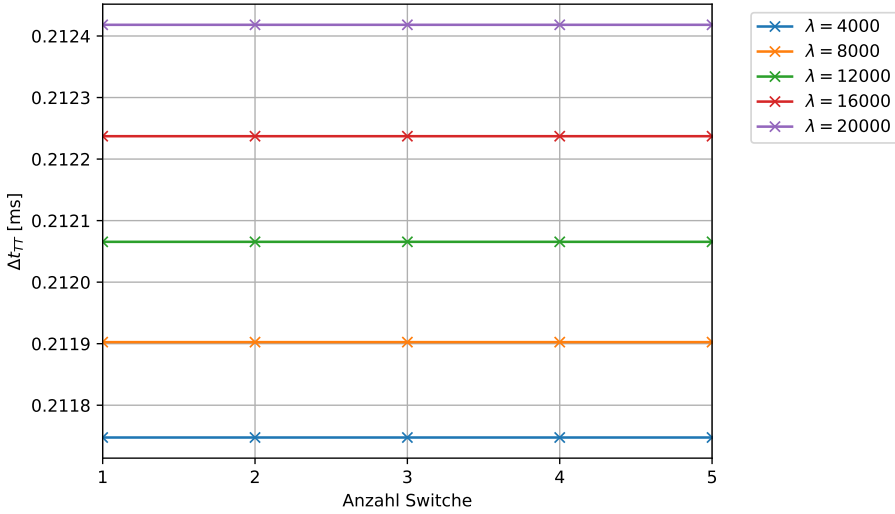
der Switch-Anzahl dar. Dies zeigt, dass die Gesamt-Paket-Rate einen Einfluss auf die Übertragungszeit hat. Bei Verwendung von weniger als vier SV-Publishern bleibt die Übertragungszeit bei einer ansteigenden Switch-Anzahl weiter konstant.

Die Abbildung 6.11b zeigt eine konstante Verzögerung für die verschiedenen untersuchten Kombinationen aus Switch-Anzahl und SV-Publisher-Anzahl. Auffällig hingegen ist die gestiegene Standardabweichung der Messdaten für sechs Container im Vergleich zu Abbildung 6.9b, in dem die Messungen ohne SV-Traffic zusammengefasst sind. Die Standardabweichung für die Messdaten bei sechs Containern ohne SV-Publisher hat 0,08449 betragen, in den Messungen aus 6.11b hingegen 0,166 bei zwei SV-Publishern bis zu 0,4474 bei fünf SV-Publishern. Das erhöhte Aufkommen von SV-Verkehr führt demnach zu einer Erhöhung der Streuung der Messwerte.

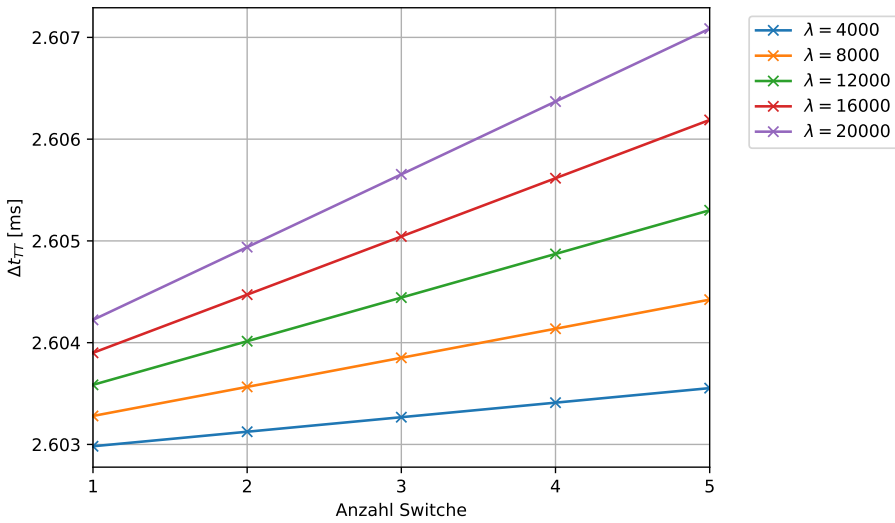
Die folgenden Abbildungen 6.12a und 6.12b zeigen die berechneten Zusammenhänge nach dem Verzögerungsmodell.

Auswertung: Bei den Messungen ohne Controller-Aktivität, wie in Abbildung 6.11a dargestellt, zeigt sich, dass die Verzögerung bei hohen Ankunftsdaten mit der Implementierung von vier bis fünf SV-Publishern mit zunehmender Anzahl der Switches ansteigt. Dies könnte auf Belastungsgrenzen hinweisen, die nicht im Verzögerungsmodell in Abbildung 6.12a berücksichtigt werden und beispielsweise durch die Gesamt-Auslastung des Hosts sowie die Virtualisierung entstehen. Insbesondere weicht die gemessene Verzögerung bei vier Publishern und fünf Switches sowie bei fünf Publishern und vier Switches in Abbildung 6.11a zunehmend von dem aufgestellten Verzögerungsmodell ab, das keine Abhängigkeit von der Anzahl der Switches aufweist. Demnach bleibt das Modell bis zu einer Gesamt-Paket-Rate innerhalb der Virtualisierung von etwa 60k Pakete pro Sekunde (PPS) bei drei Switches und fünf SV-Publishern beziehungsweise 64k PPS bei vier Switches und vier SV-Publishern gültig. Bei höheren Paket-Raten steigt die Abweichung der Messungen von den errechneten Modellwerten signifikant. Ein ähnliches Verhalten wird in [93] beschrieben.

Das Verhalten des Verzögerungsmodells aus Abbildung 6.12b ist in den Testdaten mit Controller-Aktivität aus Abbildung 6.11b nicht unmittelbar erkennbar. Die Steigung der Verzögerung in Abhängigkeit von der Anzahl der Switches in Abbildung 6.12b ist jedoch sehr gering und kann aufgrund der zuvor berechneten Standardabweichungen in den durchgeführten Messungen nicht erfasst werden. Die beobachtete Zunahme der Streuung im Vergleich zur vorherigen Validierungskategorie in Abbildung 6.9b deutet auf eine signifikante Auffälligkeit hin und schränkt die Anwendbarkeit des mathematischen Verzögerungsmodells ein.



(a) Δt_{TT} ohne Controller-Aktivität



(b) Δt_{TT} mit Controller-Aktivität

Abbildung 6.12: Δt_{TT} in Abhängigkeit der Anzahl der Switches und SV-Publisher unter Nutzung des mathematischen Verzögerungsmodells

6.2.4.3 Kategorie 3

In der Kategorie 3 zur Validierung des Verzögerungsmodells aus Tabelle 6.3 werden die Auswertungen durchgeführt, in denen die Übergabewahrscheinlichkeit eines verschickten Pakets variiert wird. Um dies zu realisieren, wird unbekannter Netzwerkverkehr erzeugt, der nicht in den Flow-Regeln der OpenFlow-Switches enthalten ist, um eine definierte Anzahl von Controller-Aktionen herbeizuführen. Um nun die Übergabewahrscheinlichkeit über diese unbekannt Pakete zu steuern, wird in der Folge eine feste Anzahl von Containern und SV-Publishern parametrisiert, während die Anzahl der unbekannt Pakete variiert wird. In den Testdaten, die in Abbildung 6.13a und 6.13b gezeigt werden, sind sechs Container und zwei SV-Publisher verwendet. Die Anzahl der Container beeinflusst, wie in Kategorie eins in Abschnitt 6.2.4.1 gezeigt, nicht die Übertragungszeit. Hierzu wird an jedem Switch eine Ankunftsrate von 8000 PPS durch die SV-Publisher generiert. Hinzu kommen nun die unbekannt Pakete, die zusätzlich von einem eigenen Container an den Switch weitergegeben werden. In den folgenden Untersuchungen werden folgende Raten der unbekannt Pakete (Packet-In-Ereignis) vorgegeben.

$$\text{Packet-In-Rate} = [400, 800, 1200, 1600, 2000] \text{ PPS} \quad (6.10)$$

Unter Berücksichtigung der Packet-In-Rate und der Ankunftsrate, erzeugt durch die SV-Publisher, lässt sich jeweils die zutreffende Übergabewahrscheinlichkeit berechnen und wird für die Abbildungen 6.14a und 6.14b entsprechend verwendet.

In Abbildung 6.13a wird ersichtlich, dass kein erkennbarer Zusammenhang zwischen der Packet-In-Rate, also der Übergabewahrscheinlichkeit eines Pakets, und der Übertragungszeit besteht. Da der Controller aus der Betrachtung bei einem bekanntem GOOSE-Flow entfällt, unterstreicht das aufgezeichnete Verhalten die Annahmen und auch die mathematische Berechnung aus Abbildung 6.14a. Auffällig ist jedoch bei den Testdaten, dass insgesamt eine geringere Übertragungszeit vorliegt als in den zuvor aufgezeichneten Datensätzen der vorherigen Kategorien. Da jedoch kein Zusammenhang zu den hier definierten Parametern zu sehen ist, wird dieser Fakt als Streuung innerhalb der Testdurchführungen verstanden und nicht weiter untersucht. Es sind keine Veränderungen am Testaufbau erfolgt.

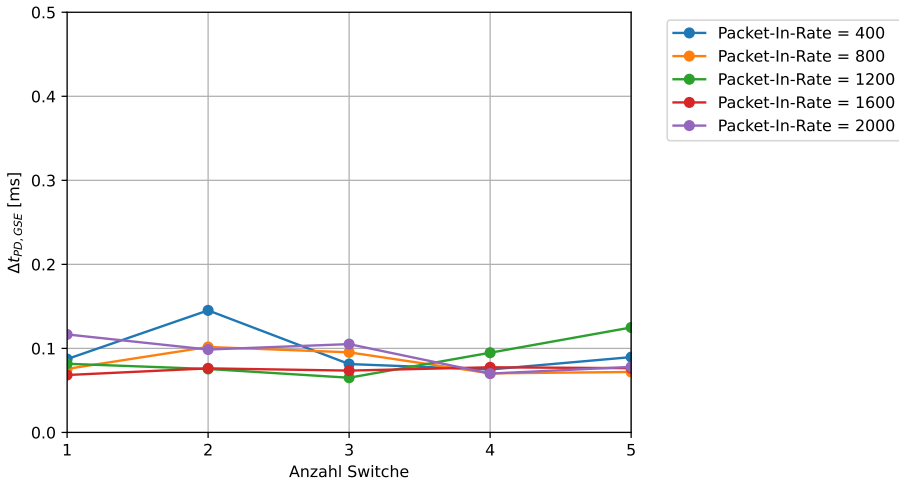
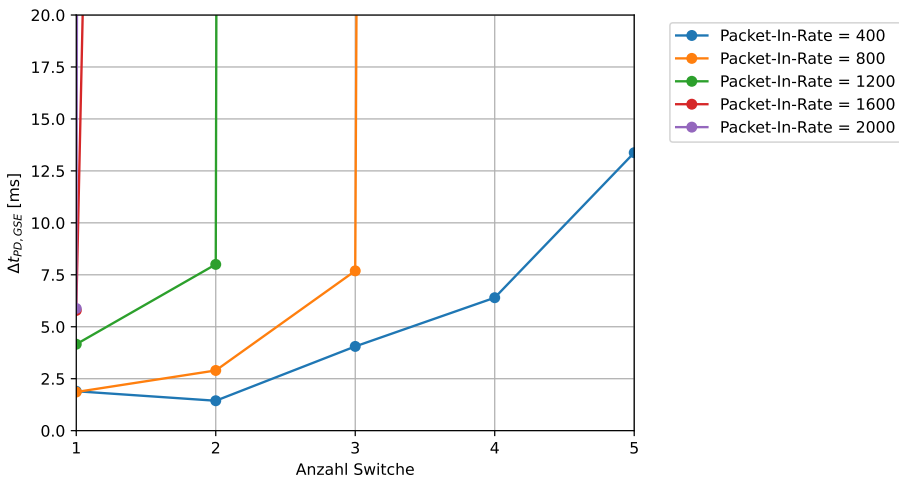
(a) $\Delta t_{PD,GSE}$ ohne Controller-Interaktion(b) $\Delta t_{PD,GSE}$ mit Controller-InteraktionAbbildung 6.13: $\Delta t_{PD,GSE}$ in Abhängigkeit der Anzahl verwendeter Switches und Packet-In-Rate in Validierungskategorie 3

Abbildung 6.13b hingegen zeigt eine starke Abhängigkeit zwischen der gemessenen Übertragungszeit, der Packet-In-Rate und der Anzahl der Switches. Die Packet-In-Rate und auch die SV-bedingte Ankunftsrate bezieht sich, wie in Kategorie 2 in Abschnitt 6.2.4.2, auf jeden einzelnen Switch. Demnach wird der Controller bei seiner

Verarbeitung und in der Warteschlange erheblich durch den Netzwerkverkehr belastet. Zunehmend mehr Controller-Aktivität bei steigender Switchanzahl und Packet-In-Rate ist die Folge. Das gemessene GOOSE-Paket verharrt so länger in der Warteschlange und der Controller ist ab einer bestimmten Auslastung nicht mehr in der Lage, die Pakete geregelt zu verarbeiten. Es entsteht ein signifikanter Anstieg der Verzögerung. Die höchste parametrisierte Packet-In-Rate von 2000 PPS (unbekannte Pakete) an jedem Switch lässt sich dementsprechend nicht in dem abgebildeten Ausschnitt in Abbildung 6.13b darstellen. Es wird aus den Messungen ersichtlich, dass die Packet-In-Rate einen sehr hohen Einfluss auf die Übertragungszeit bei notwendiger Controller-Aktivität durch das GOOSE-Paket hat.

In den Abbildungen 6.14a und 6.14b sind die durch das Verzögerungsmodell berechneten Übertragungszeiten für die Variation der Switchanzahl und der Übergabewahrscheinlichkeit dargestellt. Analog zu den Testdaten werden die parametrisierten Packet-In-Raten als Anteile der Ankunftsrate am Switch $\lambda_{i,PI}$ in die Übergabewahrscheinlichkeit für jeden Switch i umgerechnet und in den Abbildungen verwendet.

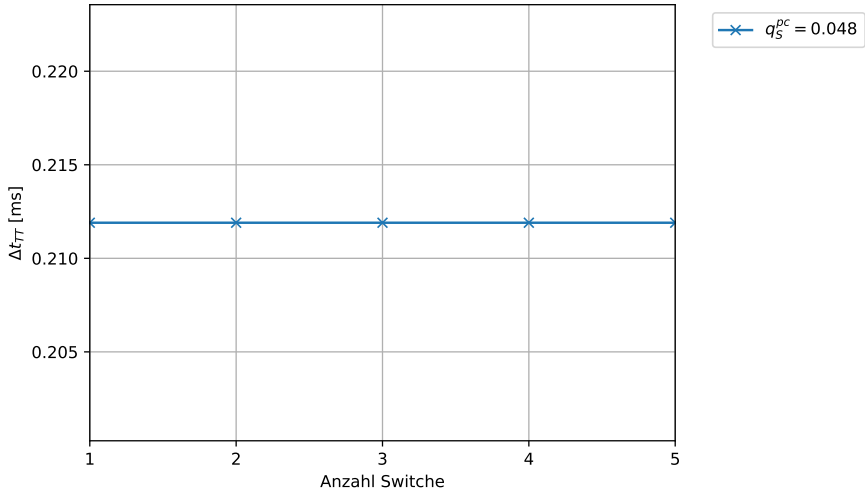
Die Übergabewahrscheinlichkeit bei zwei SV-Publishern und der Ankunftsrate $\lambda_{i,SV}$ ergibt sich zu:

$$q_S^{pc} = \frac{\lambda_{i,PI}}{\lambda_{i,SV} + \lambda_{i,PI}} \quad (6.11)$$

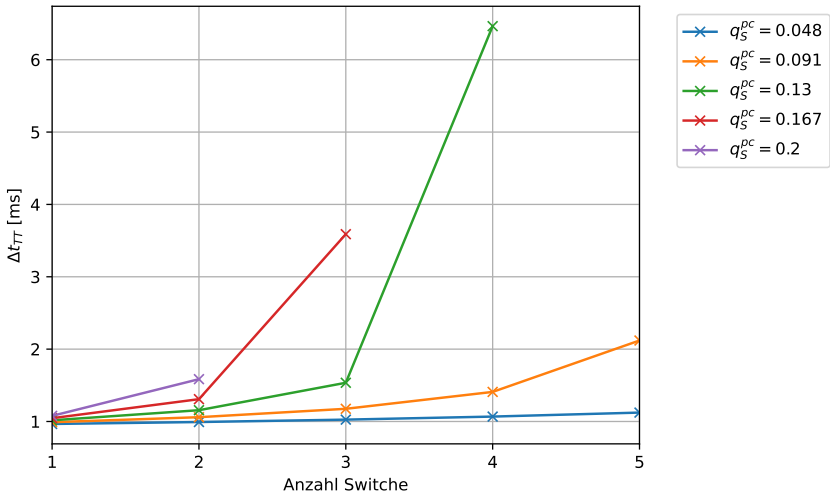
In Abbildung 6.14a wird lediglich die erste parametrisierte Übergabewahrscheinlichkeit, beziehungsweise Packet-In-Rate dargestellt, da in dem Verzögerungsmodell ohne Controller-Aktivität keine Abhängigkeit von der Übergabewahrscheinlichkeit enthalten ist. Folglich wird der konstante Wert für alle weiteren Übergabewahrscheinlichkeiten identisch sein.

Auswertung: Das dargestellte Verhalten der berechneten Übertragungszeiten ohne Controller-Aktivität entspricht dem der Testdaten aus Abbildung 6.13a. Die Übertragungszeit bleibt unter Berücksichtigung der bei der Messung auftretenden Streuung konstant. Die gemessene Übertragungszeit weicht jedoch um ca. 58,9 % vom berechneten Modell ab.

Bei der Untersuchung der Controller-Aktivität in dieser Validierungskategorie lässt sich ein ähnliches Verhalten zwischen den Testdaten und der mathematischen Berechnung feststellen. In der mathematischen Berechnung werden nur die Werte geplottet, in denen die Verzögerungsmodellgleichung im Geltungsbereich liegt. Dies ist der Fall, solange der Term $\mu_C - n \cdot \lambda_S \cdot q_S^{pc}$ positiv ist. Das Modell ist folglich nicht gültig, wenn die Servicerate des Controllers kleiner als die Rate der ankommenden Paketen



(a) Δt_{TT} ohne Controller-Aktivität



(b) Δt_{TT} mit Controller-Aktivität

Abbildung 6.14: Δt_{TT} in Abhängigkeit der Anzahl der Switches und Übergabewahrscheinlichkeit unter Nutzung des mathematischen Verzögerungsmodells

am Controller ist, da das Ergebnis des Modells negativ wird. Im dargestellten Gültigkeitsbereich sind die Steigungen der einzelnen Messverläufe deutlich größer als beim mathematischen Modell. Der angenommene Wert der Servicerate des Controllers μ_C

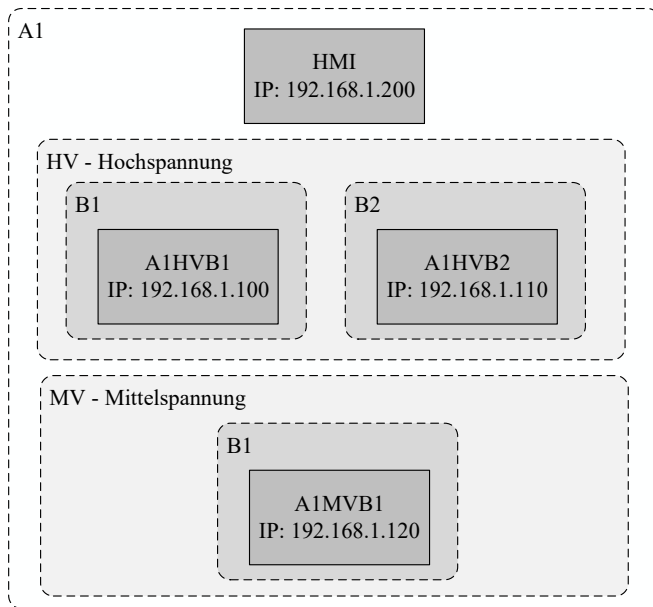
hat einen starken Einfluss auf die dargestellten Ergebnisse. Der angenommene Modellparameter aus der Literatur, wie oben beschrieben, führt für den untersuchten Testaufbau unter Nutzung des ryu-Controllers zu der Abweichung in den Abbildungen 6.13b und 6.14b. Die Servicerate im Testaufbau ist demnach geringer, wodurch ein Überlaufen der Warteschlange früher zustande kommt und ein signifikanter Anstieg der Übertragungszeit die Folge ist. Das generelle Verhalten eines überlasteten Controllers kann aber in beiden Abbildungen ähnlich gezeigt werden. Für eine exakte Parametrierung der Switch- und Controller-Serviceraten wird auf die Möglichkeiten nach beispielsweise [105] verwiesen. Sie stellen jedoch keinen Untersuchungsgegenstand für die vorliegende Arbeit dar. Der Gültigkeitsnachweis des Modellverhaltens ist für die Abschätzungen zur Übertragungszeit der Co-Simulation ausreichend.

6.2.5 Validierung der Nachbildung eines Netzwerkes mit dem Framework VirtualSubstation

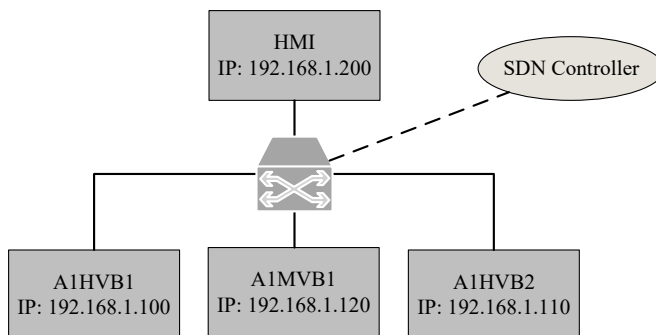
Zur Überprüfung der Funktionalität des Frameworks VirtualSubstation kommt eine beispielhafte SCD-Beschreibungsdatei zum Einsatz. Die SCD-Datei besteht aus der Beschreibung von vier IEDs auf unterschiedlichen Spannungsebenen eines Umspannwerkes und ist schematisch in Abbildung 6.15a dargestellt.

Das Ziel des Frameworks ist es, die vier IEDs zu identifizieren und mitsamt beinhalteteter Meta-Informationen das virtuelle Äquivalent in Containernet nachzubilden. Folglich sollen vier Container mit jeweils einer IEC 61850 Netzwerkschicht instanziiert werden. Die jeweiligen Informationen aus der SCD-Datei werden in vier separate ICD-Dateien überführt, wobei jedes ICD ein eigenes IED beschreibt. Die ICD-Dateien werden innerhalb der jeweiligen Container verwendet, um automatisiert mit der Bibliothek libIEC61850 die Netzwerkschicht der Container aufzubauen. Demnach werden beispielsweise GOOSE-Publisher oder MMS-Server automatisch erstellt und können separat gestartet werden.

Das Ergebnis der Ausführung des Frameworks auf die SCD-Datei mündet in dem Vorhandensein der vier Container, die aufgrund fehlender Topologieinformationen in der SCD-Datei durch eine Sternkonfiguration miteinander verbunden werden. Dementsprechend ist jeder Container in der Lage, mit den jeweils anderen zu kommunizieren. Die Kommunikation stellt der separat gestartete SDN-Controller sicher. Schematisch ist das Ergebnis des Frameworks in Abbildung 6.15b abgebildet.



(a) Schematischer Aufbau der verwendeten SCD-Datei



(b) Schematisches Ergebnis der Framework-Ausführung

Abbildung 6.15: Schematische Darstellung der verwendeten SCD-Datei und des daraus resultierenden Framework-Ergebnisses

Für den Nachweis der Funktionsfähigkeit dient neben der schematischen Darstellung der instanziierten Container ein beispielhafter Netzwerkmitschnitt, der den Datenverkehr zwischen zwei vIEDs zeigt. Hier wird ein exemplarischer GOOSE-Publisher im vIED A1HVB1, ein MMS-Server innerhalb von A1MVB1 und ein MMS-Client im Human Machine Interface (HMI) gestartet. Für die MMS-Kommunikation wird ein Reporting-Service vom HMI abonniert. Die resultierenden Netzwerkpakete werden am OpenFlow-Switch aufgenommen. Die Pakete sind als Screenshots des Analysetools Wireshark in Abbildungen 6.16a und 6.16b dargestellt.

Die Netzwerkmitsschnitte zeigen, dass die Container erfolgreich gestartet und korrekt mit den Meta-Informationen, wie der IP-Adresse, verknüpft sind. Weiterhin sind die Netzwerkschichten der vIEDs erfolgreich initiiert und erlauben einen Datenaustausch unter den vIEDs. Die notwendige SDN-basierte Netzwerkinfrastruktur konnte eingerichtet werden und die Kommunikationsfähigkeit hergestellt werden. Die Grenzen des Frameworks liegen vor allem im zusätzlichen Bedarf einer Beschreibung der Netzwerktopologie, um ein Umspannwerk mit der Trennung des Prozess- und Stationsbusses umsetzen zu können. Aktuell sind sämtliche vIEDs und Schnittstellen für den Datenaustausch über einen zentralen Switch verbunden. Kritisch kann dies bei der Durchmischung der hochfrequenten SV-Multicast Nachrichten mit anderen Services sein, die gegebenenfalls eine negative Auswirkung auch auf die Übertragungszeit der Co-Simulation haben könnte. Dieser mögliche Einfluss wurde jedoch aus dem Betrachtungsraum der Arbeit gelassen, um die generelle Funktionsfähigkeit des Ansatzes nachzuweisen. [88]

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	2a:03:f7:ce:cc:b0	Iec-Tc57_01:...	GOOSE	165	
2	1.000977856	2a:03:f7:ce:cc:b0	Iec-Tc57_01:...	GOOSE	165	
3	2.000943785	2a:03:f7:ce:cc:b0	Iec-Tc57_01:...	GOOSE	165	

< >

> Frame 1: 165 bytes on wire (1320 bits), 165 bytes captured (1320 bits) on interface s1-eth1, id 0
 > Ethernet II, Src: 2a:03:f7:ce:cc:b0 (2a:03:f7:ce:cc:b0), Dst: Iec-Tc57_01:00:04 (01:0c:cd:01:00:04)
 > 802.1Q Virtual LAN, PRI: 4, DEI: 0, ID: 0
 v GOOSE
 APPID: 0x0001 (1)
 Length: 147
 > Reserved 1: 0x0000 (0)
 Reserved 2: 0x0000 (0)
 v goosePdu
 gocbRef: A1HVB1LD0/LLN0\$G0\$gcbSwitchgear
 timeAllowedtoLive: 500
 dataSet: A1HVB1LD0/LLN0\$G0\$gcbSwitchgear
 goID: A1HVB1LD0/LLN0\$G0\$gcbSwitchgear

< >

```

0000 01 0c cd 01 00 04 2a 03 f7 ce cc b0 81 00 80 00
0010 88 b8 00 01 00 93 00 00 00 00 61 81 88 80 1f 41
0020 31 48 56 42 31 4c 44 30 2f 4c 4c 4e 30 24 47 4f
0030 24 67 63 62 53 77 69 74 63 68 67 65 61 72 81 02
0040 01 f4 82 1f 41 31 48 56 42 31 4c 44 30 2f 4c 4c
0050 4e 30 24 47 30 24 67 63 62 53 77 69 74 63 68 67
0060 65 61 72 83 1f 41 31 48 56 42 31 4c 44 30 2f 4c
0070 4c 4e 30 24 47 4f 24 67 63 62 53 77 69 74 63 68
0080 67 65 61 72 84 08 62 56 da 93 22 4d d2 0a 85 01
0090 01 86 01 00 87 01 00 88 01 01 89 01 00 8a 01 01

```

(a) GOOSE-Kommunikation von A1HVB1

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.1.200	192.168.1.120	MMS	253	initiate-RequestPDU
2	0.300232017	192.168.1.120	192.168.1.200	MMS	209	initiate-ResponsePDU
3	0.415096711	192.168.1.200	192.168.1.120	MMS	102	01 confirmed-RequestPDU
4	0.715317901	192.168.1.120	192.168.1.200	MMS	111	01 confirmed-ResponsePDU
22	3.816757652	192.168.1.200	192.168.1.120	MMS	88	conclude-RequestPDU
23	4.117046644	192.168.1.120	192.168.1.200	MMS	88	conclude-ResponsePDU

< >

> Frame 4: 111 bytes on wire (888 bits), 111 bytes captured (888 bits) on interface HMI-eth0, id 0
 > Ethernet II, Src: f2:4a:a5:f1:cb:d8 (f2:4a:a5:f1:cb:d8), Dst: 52:ee:bf:52:af:20 (52:ee:bf:52:af:20)
 > Internet Protocol Version 4, Src: 192.168.1.120, Dst: 192.168.1.200
 > Transmission Control Protocol, Src Port: 102, Dst Port: 44578, Seq: 144, Ack: 224, Len: 45
 > TPKT, Version: 3, Length: 45
 > ISO 8073/X.224 COTP Connection-Oriented Transport Protocol
 > ISO 8327-1 OSI Session Protocol
 > ISO 8327-1 OSI Session Protocol
 > ISO 8823 OSI Presentation Protocol
 v MMS
 v confirmed-ResponsePDU
 invokeID: 1
 v confirmedServiceResponse: getNameList (1)
 v getNameList
 v listOfIdentifier: 1 item
 Identifier: A1MVB1H1Q01
 moreFollows: False

< >

```

0000 52 ee bf 52 af 20 f2 4a a5 f1 cb d8 08 00 45 00
0010 00 61 ac 37 40 00 40 06 09 cf c0 a8 01 78 c0 a8
0020 01 c8 00 66 ae 22 ef c4 ee d8 a2 95 d6 8b 80 18
0030 00 55 84 e4 00 00 01 01 08 0a 15 70 d0 f6 e5 65
0040 71 17 03 00 00 2d 02 f0 80 01 00 01 00 61 20 30
0050 1e 02 01 03 a0 19 a1 17 02 01 01 a1 12 a0 0d 1a
0060 0b 41 31 4d 56 42 31 48 31 51 30 31 81 01 00

```

(b) MMS-Kommunikation zwischen A1MVB1 und HMI

Abbildung 6.16: Wireshark-Aufzeichnungen der im Rahmen der Validierung erfassten Kommunikationsströme (oben: GOOSE-Kommunikation, unten: MMS-Kommunikation)

6.3 Fazit der Validierung

In den verschiedenen vorangegangenen Abschnitten wurden, die im Rahmen der Arbeit aufgestellten Aussagen näher untersucht und mit praktischen Umsetzungen validiert. Dabei ist festzustellen, dass der eingeführte Ansatz der Co-Simulation für die Stationsautomatisierung im untersuchten Bereich gültig ist, mit den jeweils ausgeführten Beschränkungen, wie beispielsweise der Limitierung an Netzwerkverkehr.

Die generelle Möglichkeit, das Netzwerkverhalten nachzubilden, wurde nachgewiesen, indem Netzwerkpakete mit einem Realaufbau erzeugt wurden. Es gibt jedoch nachgewiesene Einschränkungen hinsichtlich der Struktur des Testaufbaus, beispielsweise der verwendeten Netzwerkadapter, die einen Einfluss auf den Inhalt der Netzwerkpakete haben können. Dennoch kann das Verhalten eines realen IEC 61850 Aufbaus grundlegend nachgebildet werden.

Die für eine Co-Simulation notwendige Zeitgleichheit der einzelnen Simulatoren konnte mithilfe von Referenzliteratur erklärt und mittels eingeführter Netzwerkmittschnitte demonstriert werden.

Anschließend konnte anhand eines Minimalbeispiels erstmalig der Einfluss der Virtualisierung auf die Größe der Übertragungszeit abgebildet werden. Es konnte experimentell nachgewiesen werden, dass unter Nutzung der SDN-Struktur der Virtualisierung die Restriktionen des Standards IEC 61850 eingehalten werden können.

Auf Basis dieser generellen Aussage wurde der Einfluss der Struktur und des Umfangs der virtuellen Umgebung auf die Übertragungszeit untersucht. Hierfür wurden umfangreiche Tests durchgeführt, mit verschiedenen Variationen der Strukturparameter, wie der Anzahl verwendeter virtueller Switches und abgebildetem Netzwerkverkehr. Der Vergleich mit dem eingeführten Verzögerungsmodell zeigt das erwartbare Verhalten mit der Ergänzung, dass eine auftretende Obergrenze des Gesamt-Netzwerkverkehrs für die Gültigkeit des mathematischen Modells existiert. Wird ein größeres Aufkommen von Netzwerkverkehr simuliert, so ist das Verzögerungsmodell außerhalb des Gültigkeitsbereiches und es müssen noch weitere Betrachtungen beispielsweise unter Berücksichtigung der Rechenoperationen auf dem Virtualisierungsrechner und hieraus entstehenden Abhängigkeiten durchgeführt werden. Für die vorliegende Arbeit wird folglich die Aussage über die Gültigkeit des Modells nur bis zu der experimentell bestimmten Grenze von ca. 60k PPS nachgewiesen.

Abschließend wurde die validierte Virtualisierung zur Abbildung der Kommunikationsinfrastruktur der Stationsautomatisierung verwendet und in dem Framework VirtualSubstation integriert, um anhand von Beschreibungsdateien ein reales Netzwerk nachzubilden. Der experimentelle Nachweis der Funktionsfähigkeit des Frameworks wurde mit einer beispielhaften Infrastruktur und gestarteten Kommunikationsservices erbracht.

Zusammengefasst sind somit alle wesentlichen Aussagen der vorliegenden Arbeit zur Entwicklung der Co-Simulation untersucht und validiert worden. Das abschließende Fazit aus den validierten Aussagen ist, dass über das mathematische Warteschlangenmodell die Übertragungszeit in der Co-Simulation abgeschätzt werden kann. Basierend auf der Struktur, beispielsweise durch die Informationen aus den SCD-Beschreibungsdateien, lässt sich vor der Implementierung der Co-Simulation beurteilen, ob die Struktur unter Einhaltung der Restriktionen an die Übertragungszeit simuliert werden kann. Dabei gilt die Restriktion von 3 ms für die Übertragungszeit. Hierbei zeigen sowohl das mathematische Modell als auch die simulativen Untersuchungen, dass zur Einhaltung der erlaubten Übertragungszeit eine Controller-Aktivität vermieden werden sollte. Das bedeutet, dass kritische Nachrichten, wie Befehle über das GOOSE-Protokoll bereits im Vorhinein für die OpenFlow-Switches anhand ihrer Flow-Tabellen bekannt sein sollten. Dementsprechend wird abgeleitet empfohlen, dass diese Nachrichten in der SCD-Datei identifiziert und vor der eigentlichen Simulation bereits einmalig erzeugt und in dem virtuellen Netzwerk verschickt werden, um einen Eintrag in der Flow-Tabelle der Switches zu erzeugen und eine erneute Controller-Aktivität zu vermeiden. Alternativ kann auch direkt aus der SCD-Datei eine Flow-Tabelle erstellt und an die zu instanzierenden Switches gegeben werden. Unter Vermeidung der Controller-Aktivität zeigen die Auswertungen der Validierungskategorien, dass vor allem die Gesamt-Paket-Rate im virtuellen Netzwerk einen Einfluss auf die Übertragungszeit hat und die experimentell ermittelte obere Grenze von ca. 60k PPS nicht überschritten werden sollte. Eine Einhaltung der erlaubten Übertragungszeit kann hierdurch gewährleistet werden, mit der Einschränkung der identifizierten Streuung der Messwerte.

7 Zusammenfassung und Ausblick

7.1 Zusammenfassendes Fazit

Die vorliegende Arbeit stellt einen neuen Ansatz einer Co-Simulation für die Stationsautomatisierung in Digitalen Stationen dar und verwendet dabei eine Zusammenführung einer Stromnetzsimulation mit einer neuartigen Virtualisierung der Kommunikationsinfrastruktur. Im Rahmen der Arbeit konnte ein Bedarf an einer gemeinschaftlichen Simulation des dynamischen Verhalten des Stromnetzes und des Kommunikationsnetzwerks in digitalisierten Infrastrukturen herausgearbeitet werden. Dabei konnte festgestellt werden, dass vor allem ein Bedarf von dynamischen Untersuchungen besteht, um beispielsweise die Interaktion zwischen IEDs, wie Schutzgeräten, und dem Stromnetz näher untersuchen zu können. Mögliche Untersuchungen können im Bereich von Störungen, Ausfällen oder IT-Angriffen auf vernetzte Automatisierungs- und Steuerungsgeräte im Umspannwerk liegen. Hierbei kann eine Simulation wichtige Daten generieren, um die Verflechtungen und Abhängigkeiten im Falle einer Störung zu analysieren. Weiterhin können simulierte Daten erforderlich und sinnvoll sein, um beispielsweise Detektions- oder Monitoringsysteme zu entwickeln und zu testen. [106], [107]

Aus diesem Bedarf heraus wurde ausgehend von einer Literaturrecherche und Analyse von bestehenden Ansätzen der Co-Simulation die Anforderungen für das entwickelte System herausgearbeitet. Eine flexible Simulationsumgebung der Kommunikationsnetzwerkinfrastruktur unter der Implementierung von realen Übertragungsprotokollen und Funktionen der Stationsautomatisierung bildet eine Lücke in den recherchierten Ansätzen. Die eingeführte Co-Simulation hat das Potenzial, diese Lücke zu schließen und eine Simulationsplattform zu schaffen, die den Bedarf an Entwicklungs- und Testmöglichkeiten deckt.

Abgeleitet aus den identifizierten Lücken veröffentlichter Co-Simulationen und den funktionalen und kommunikationstechnischen Anforderungen der IEC 61850 basierten Stationsautomatisierung wurde ein eigener Ansatz der Co-Simulation formuliert. Diese Formulierung stellt den Kern der Arbeit dar, wonach eine Stromnetzsimulation mit einer Virtualisierung der Kommunikation verbunden wird. Die Virtualisierung der Sekundärsysteme für den Anwendungsfall des Umspannwerkes über dedizierte Container und der Kommunikationsinfrastruktur durch SDN-Ansätze stellt einen neuar-

tigen Beitrag der Arbeit dar. Hierbei schafft der Ansatz die Möglichkeit, IEDs detailgetreu anhand der zwei Schichten – Funktion und Netzwerk – nachzubilden und so als virtuelle Instanz in einem parametrierbaren virtuellen SDN-Netzwerk zu platzieren. Demnach kann anhand von textuellen Beschreibungsdateien die Instanziierung der virtuellen Netzwerke erfolgen und eine Kommunikationsumgebung der Stationsautomatisierung in die Virtualisierung überführt werden. Die aufgebauten Schnittstellen ermöglichen es, eine Ad-hoc-Co-Simulation mit der Stromnetzsimulation aufzubauen. Die Kopplung der Simulatoren erfolgt direkt durch die Nutzung gleichartiger Schnittstellen, ohne einen Einfluss des Koordinators auf den Datenaustausch. Innerhalb der Arbeit wurden die Grundanforderungen an die Stromnetzsimulation formuliert, ohne den Simulator im Detail zu untersuchen. Die Dynamik und Echtzeitfähigkeit von Stromnetzsimulatoren stellen bereits umfangreich bearbeitete Forschungsgebiete dar.

Der erarbeitete Ansatz der Co-Simulation wurde anschließend innerhalb der Arbeit durch eine Beschreibung der Umsetzung näher untersetzt. Die genutzten Implementierungen für die abschließende Validierungsphase der vorliegenden Arbeit sind eingeführt und beschrieben. Dabei stellt die Kommunikationsebene, wie bereits beschrieben, den Schwerpunkt dar. Der SDN- und Container-basierte Virtualisierungsansatz wird über die Nutzung des Tools Containernet und Docker-Containern maßgeblich geformt. Eine Anwendung im Kontext der Stationsautomatisierung ist dabei ein Alleinstellungsmerkmal der vorliegenden Arbeit. Die automatisierte Überführung von IEC 61850-Beschreibungsdateien von Umspannwerken in die Virtualisierung demonstriert eine Möglichkeit, den Ansatz für die Co-Simulation zu nutzen.

Sämtliche neu gewonnenen Aussagen der Arbeit sind im Abschnitt 6 zusammengefasst und wissenschaftlich nachgewiesen. Dabei konnten fünf Kernaussagen der Arbeit und des Ansatzes identifiziert werden. Für jede Kernaussage wurde zunächst ausgeführt, unter welchen Bedingungen, Vorgehensweisen und Implementierungen die Aussagen näher untersucht werden können. Die Durchführung der einzelnen Validierungen sind anschließend umfangreich im Rahmen der Arbeit beschrieben. Die Ergebnisse der Testdurchführungen zeigen experimentelle Nachweise der unterschiedlichen Aussagen und bekräftigen die Gültigkeit und auch die Grenzen der eingeführten Co-Simulation. Die umfangreiche Untersuchung des wichtigen Kriteriums der Übertragungszeit und Vergleich mit dem mathematisch aufgestellten Verzögerungsmodell weisen explizit auf den Gültigkeitsbereich hin. Alle herausgearbeiteten Aussagen der Arbeit konnten folglich validiert oder die abgeleiteten Grenzen identifiziert werden.

Insgesamt wurde innerhalb der Arbeit ein neuartiges Systemdesign der Co-Simulation durch Nutzung einer Virtualisierung entworfen und experimentell validiert, um

eine Möglichkeit der gesamtheitlichen Simulation der Stationsautomatisierung zu schaffen. Weiterhin kann die Gültigkeit der Co-Simulation bereits vor Ausführung über ein eingeführtes Verzögerungsmodell anhand der Übertragungszeit eingeschätzt und bewertet werden. Hieraus lässt sich bestimmen, ob ein gegebenes Netzwerk aufgrund der Topologie- und Größeneigenschaften innerhalb der erlaubten Restriktionen simulierbar ist. Die Co-Simulation kann so beispielsweise Daten einer Digitalen Station erzeugen oder eine Laufzeit-Simulationsumgebung bereitstellen und bietet im Vergleich zu den alternativen Ansätzen der Literatur einen deutlichen Mehrwert in der detailgetreuen Abbildung des Anwendungsfalls.

7.2 Reflexion der Forschungsfragen

Die in Abschnitt 1.3 formulierten Forschungsfragen wurden in den verschiedenen Arbeiten und Beschreibungen behandelt. Im Folgenden werden die relevanten Erkenntnisse zur Bewertung dieser Forschungsfragen zusammengefasst.

Forschungsfrage 1: Abstraktionsfähigkeit des Kommunikationsnetzwerks und Kombination zu einer Co-Simulation Der vorgestellte Ansatz der Virtualisierung wurde für die Emulation des Netzwerks der Stationsautomatisierung eingeführt und im Rahmen der Co-Simulation mit einer Echtzeit-Stromnetzsimulation gekoppelt. Auf die Gültigkeit der Abstraktionsfähigkeit und Gültigkeit der Kopplung wurde insbesondere im Rahmen der umfassenden Validierung eingegangen und beweist die These der Ad-hoc-Kopplung der beiden gewählten Simulatoren mit direktem Datenaustausch.

Forschungsfrage 2: Herausforderungen der Kopplung und Zeitanforderungen

Die eingeführten Zeit- und Kopplungsanforderungen orientieren sich an den Vorgaben des Kommunikations-Standards IEC 61850. Folglich wurden die Bedingungen der Übertragungszeit und Synchronität realer Infrastrukturen für die Co-Simulation übernommen und gelten hier als Maßstab der Gültigkeit. Weiterhin gilt das Kriterium der Echtzeitfähigkeit für beide verwendeten Simulatoren und für das gekoppelte Gesamtsystem.

Besonderes Augenmerk wurde in der Arbeit auf die Übertragungszeit gelegt, die mathematisch über ein formuliertes Warteschlangenmodell abgeschätzt werden kann. Die Gültigkeit des Modells wurde zudem im Rahmen der Validierung experimentell untersucht und inklusive erkannter Einschränkungen nachgewiesen.

Forschungsfrage 3: Modellierung von Netzwerk- und Automatisierungskomponenten Für die Abbildung der Kommunikationsebene, einschließlich der Netzwerk- und Automatisierungskomponenten, wurde im Rahmen dieser Arbeit eine SDN-basierte Virtualisierung eingeführt. Anhand exemplarischer Implementierungen konnte sowohl das Verhalten des Kommunikationsnetzwerks durch die Korrektheit der implementierten Übertragungsprotokolle als auch die Übertragungszeiten im Vergleich zu Vorgaben aus relevanten Standards für reale Netzwerke demonstriert werden. Der Ansatz der vIEDs für die Automatisierungskomponenten ist zudem geeignet für den Einsatz virtualisierter Funktionen in realen Anlagen. Das eingeführte Framework zur Überführung einer Beschreibungsdatei in die Virtualisierung des Netzwerks ermöglicht eine Modellierung der Kommunikationsebene. Die durchgeführten Validierungen belegen auch hier die Gültigkeit der eingeführten Konzepte zur Abbildung der Kommunikationsebene.

7.3 Ausblick, mögliche Erweiterungen und Verbesserungen

Der eingeführte Ansatz der Co-Simulation stellt einen ersten Systementwurf für die Simulation der Stationsautomatisierung von Digitalen Stationen dar und kann zukünftig vor allem einen Baustein für domänenübergreifende Untersuchungen liefern. Die Untersuchungen und Validierungen des Ansatzes zeigen zusätzlich Weiterentwicklungs- und Optimierungspotenzial und werden nachfolgend kurz beschrieben.

Priorisierung von Paketen Die Untersuchungen in Bezug auf die Übertragungszeit der Pakete zeigt, dass die sequenzielle Verarbeitung der Pakete innerhalb der SDN-Komponenten (Switch und Controller) den maßgeblichen Anteil der Verzögerung bewirkt. Um auch außerhalb der experimentell und berechneten Grenzen zur Einhaltung der Übertragungszeitbeschränkungen Simulationen korrekt durchführen zu können, ist eine mögliche Erweiterung die Priorisierung der unterschiedlichen Pakete in Anlehnung an die Prioritäten der Übertragungsklassen nach IEC 61850-5, siehe Tabelle 3.3, für die SDN-Komponenten. Die Verarbeitung von hochpriorien Paketen wird beispielsweise von den OpenFlow-Switches vorgezogen, um für diese die Vorgabe der Übertragungszeit zu garantieren. Hierfür muss jedoch eine Vorverarbeitung entworfen und implementiert werden, um die Priorisierungsinformationen zu verarbeiten.

In zukünftigen Untersuchungen kann ein Ansatz dahingehend erarbeitet und getestet werden, um die Grenzen beispielsweise in Bezug auf die maximale Gesamt-Paketrate erweitern zu können.

Verwendung von angepassten SDN-Controllern Neben den OpenFlow-Switches steht vor allem die Controller-Aktivität im Vordergrund der Übertragungszeit. Die dargestellten Ergebnisse zeigen, dass der Aufbau für die Einhaltung der Übertragungszeit-Grenzen bestehende Flow-Regeln erfordert. Unbekannte Pakete, bei denen eine Controller-Aktivität notwendig wird, können in den Testdurchführungen nicht oder nicht zuverlässig die Grenzen der Übertragungszeit einhalten. Die Analysen ergeben, dass die Verarbeitungszeit aufgrund der Zeitanteile der Warteschlange und der Servicerate des Controllers zu hoch ist, weshalb eine verbesserte und angepasste Implementierung des SDN-Controllers erforderlich ist. Einige Untersuchungen aus der Literatur (z. B. [81], [108]), zeigen verschiedene Controller-Implementierungen und deren Performanz. Die gewählte ryu-Implementierung könnte für nachfolgende Untersuchungen ersetzt oder adaptiert werden, um eine höhere Verarbeitungsgeschwindigkeit zu erreichen. Eine detaillierte Untersuchung im Rahmen der Co-Simulation ist hierzu notwendig und sollte zukünftig in Betracht gezogen werden.

Verwendung von mehreren SDN-Controllern Ergänzend zu der reinen Verbesserung der Controller-Implementierung kann eine weitere Ergänzung in Bezug auf die SDN-Controller eine mögliche Erweiterung der Struktur hin zu Multi-Controller-Ansätzen sein. Aus der Validierung mitsamt der Testdaten ist ersichtlich, dass die Anzahl der Switches einen Einfluss bei Verwendung eines einzelnen Controllers hat. Untersuchungen zu Multi-Controller-Ansätzen [109]–[111] zeigen, dass unter Umständen die Verringerung der Anzahl der Switches pro Controller einen positiven Effekt auf die Übertragungszeit haben kann. Dies könnte für den Anwendungsfall der Stationsautomatisierung von Umspannwerken bedeuten, dass in den dargelegten Konzepten und Implementierungen jeder Abgang oder Prozessbus eines Umspannwerkes über einen dedizierten virtuellen OpenFlow-Switch realisiert ist und zukünftig über einen eigenen Controller erweitert werden könnte. Hieraus kann eine Eins-zu-Eins-Beziehung zwischen Switch und Controller resultieren und die Ankunftsrate am Controller verringert werden. Hinzu kämen jedoch gegebenenfalls eine zusätzliche Orchestrierung und Koordination der verschiedenen Controller.

Erweiterung durch Zeit-Steuerungstools Abseits der verwendeten SDN-Infrastruktur zu synchronen Simulationen des Kommunikationsverhaltens in Form der zur Laufzeit synchronen Virtualisierung kann eine mögliche Erweiterung der Co-Simulation sein, dass ein Zeit-Steuerungstool eingesetzt wird [112]. Mit dieser Erweiterung kann die Zeitbasis der Virtualisierung und deren Komponenten manipuliert werden. Prinzipiell sind hierdurch beschleunigte oder verlangsamte Betrachtungen der Kommunikationsebene möglich. Vor allem eine verlangsamte Ausführung kann Vorteile bringen in den seriellen Ausführungen der SDN-Komponenten und beispielsweise die Beschränkung der maximalen Gesamt-Paketrate auflösen. Jedoch würde eine solche Erweiterung den Verzicht auf einen spezifischen Koordinator zum Datenaustausch der Co-Simulation wiederum infrage stellen, da beide Simulatoren nicht mehr synchron zur Laufzeit wären und synchronisiert, beschleunigt oder verlangsamt werden müssen. Zusätzlich besteht ein realer Datenkanal zwischen den beiden Simulatoren, der ebenfalls mit betrachtet werden muss. Der formulierte Ansatz der Co-Simulation wäre folglich nicht mehr vollumfassend anwendbar.

Verwendung eines Echtzeitbetriebssystems Innerhalb der Ergebnisse und Umsetzungen ist ersichtlich geworden, dass einige Verzögerungen in der Virtualisierung auch durch Grenzen des Virtualisierungshosts entstehen. Vor allem in der parallelen Ausführung der verschiedenen Berechnungen und Prozesse der Container und SDN-Komponenten hat das Scheduling des Betriebssystems einen Einfluss auf die Dauer und gegebenenfalls auch Priorität der Abarbeitung. Die bei der Echtzeit-Definition beschriebenen Echtzeitbetriebssysteme, wie RTLinux [42], können eingesetzt werden, um besonders im Fall von komplexen Netzwerken oder hohen Datenaufkommen die zusätzlichen Verzögerungseffekte zu minimieren. Entsprechende Analysen für die Ausführung von Virtualisierungen wurden beispielsweise in [113] untersucht.

Erweiterung des Verzögerungsmodells Die experimentelle Untersuchung der Übertragungszeit wurde in Bezug auf das formulierte mathematische Verzögerungsmodell betrachtet. Dabei wurden Abweichungen vom Modell festgestellt, die eine Modellexerweiterung erforderlich machen, um möglichst ähnliche Ergebnisse zu erzielen. Besonders hervorzuheben ist die Nichtberücksichtigung der Verarbeitungslogik und der Einflüsse des Virtualisierungshosts. Ersichtlich wurde dies in der begrenzten Gültigkeit des Verzögerungsmodells bei der Gesamt-Paketrate innerhalb der Virtualisierung. Die sequenzielle Verarbeitung über den Host wird über eine Betriebssystem-basiertes Scheduling umgesetzt. Jede Berechnung auf dem Host hat eine Allokation von Res-

sourcen zur Folge. Hierunter fallen auch die Bearbeitungs- und Berechnungsschritte der einzelnen Container und SDN-Elemente. Besonders bei einer hohen Summe an parallelen Prozessen auf dem Host ist mit einer zusätzlichen Verzögerung zu rechnen, die nicht über das reine Warteschlangenmodell der SDN-Komponenten abgedeckt ist. Die Ergebnisse der Arbeit deuten darauf hin, dass die Einflüsse im Untersuchungsbe- reich nicht so stark waren, dass sie das aufgestellte Modell vollständig beeinträchtigen. Es ist jedoch zu erwarten, dass das Modell bei der Erweiterung um Host-Prozesse an weitere Grenzen stoßen wird. Ein Beispiel hierfür konnte bereits in den angestellten Untersuchungen gezeigt und beschrieben werden. Soll also die Co-Simulation für er- heblich größere und umfangreichere Netzwerke verwendet werden, bedarf es einer Mo- dellenerweiterung um die Einflüsse, die vom Virtualisierungshost hinzukommen können, um die auftretenden Verzögerungen besser abschätzen zu können. Auf die generelle Aussage des Ansatzes der Co-Simulation hingegen hat eine Modellerweiterung keine Auswirkung.

Literaturverzeichnis

- [1] Hua Lin, S. Sambamoorthy, S. Shukla, J. Thorp und L. Mili, „Power system and communication network co-simulation for smart grid applications“, in *ISGT 2011*, IEEE, Jan. 2011, S. 1–6.
 - [2] W. Li und X. Zhang, „Simulation of the smart grid communications: Challenges, techniques, and future trends“, *Computers and Electrical Engineering*, Jg. 40, Nr. 1, S. 270–288, Jan. 2014.
 - [3] C. Steinbrink, F. Schlogl, D. Babazadeh, S. Lehnhoff, S. Rohjans und A. Naranan, „Future perspectives of co-simulation in the smart grid domain“, in *2018 IEEE International Energy Conference, ENERGYCON 2018*, 2018, S. 1–6.
 - [4] C. Gomes, C. Thule, D. Broman, P. G. Larsen und H. Vangheluwe, *Co-simulation: State of the art*, 2017. Adresse: <https://arxiv.org/abs/1702.00686>.
 - [5] D. Rösch, A. Kummerow, S. Ruhe, K. Schäfer, C. Monsalve und S. Nicolai, „Cybersecurity in digital substations: Cyber-physical system modelling, assessment and analysis“, *At-Automatisierungstechnik*, Jg. 68, Nr. 9, S. 720–737, Sep. 2020.
 - [6] C. Gomes, C. Thule, D. Broman, P. G. Larsen und H. Vangheluwe, „Co-Simulation: A Survey“, *ACM Comput. Surv.*, Jg. 51, Nr. 3, Mai 2018.
 - [7] T. Westermann, *Modellbildung und Simulation*, 2. Aufl. Heidelberg: Springer Berlin, Heidelberg, 2021.
 - [8] M. Geimer, T. Krüger und P. Linsel, „Co-Simulation, gekoppelte Simulation oder Simulationskopplung? Ein Versuch der Begriffsvereinheitlichung“, *O+P Ölhydraulik und Pneumatik*, Jg. 50, Nr. 11-12, S. 4–8, 2006.
 - [9] R. Schmoll, *Co-Simulation und Solverkopplung - Analyse komplexer multiphysikalischer Systeme*. Kassel university press GmbH, 2015, S. 1–146.
 - [10] P. Palensky, A. A. Van Der Meer, C. D. Lopez, A. Joseph und K. Pan, „Cosimulation of Intelligent Power Systems: Fundamentals, Software Architecture, Numerics, and Coupling“, *IEEE Industrial Electronics Magazine*, Jg. 11, Nr. 1, S. 34–50, Mär. 2017.
 - [11] R. Nollau, *Modellierung und Simulation technischer Systeme*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009.
-

-
- [12] H. Luke, „The origins of the sampling theorem“, *IEEE Communications Magazine*, Jg. 37, Nr. 4, S. 106–108, Apr. 1999.
- [13] E. D. N. Ndih und S. Cherkaoui, „Simulation methods, techniques and tools of computer systems and networks“, in *Modeling and Simulation of Computer Networks and Systems*, Elsevier, 2015, S. 485–504.
- [14] B. K. Choi und D. Kang, Hrsg., *Modeling and Simulation of Discrete-Event Systems*. Wiley, Okt. 2013.
- [15] T. Sauerbier, *Theorie und Praxis von Simulationssystemen*. Wiesbaden: Vieweg+Teubner Verlag, 1999.
- [16] D. Baum, *Grundlagen der Warteschlangentheorie*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.
- [17] T. Sauerbier und O. Mildenerger, „Warteschlangensysteme“, in *Theorie und Praxis von Simulationssystemen*, Wiesbaden: Vieweg+Teubner Verlag, 1999, S. 137–165.
- [18] A. Iqbal, U. Javed, S. Saleh, J. Kim, J. S. Alowibdi und M. U. Ilyas, „Analytical Modeling of End-to-End Delay in OpenFlow Based Networks“, *IEEE Access*, Jg. 5, S. 6859–6871, 2017.
- [19] M. Jarschel, S. Oechsner, D. Schlosser, R. Pries, S. Goll und P. Tran-Gia, „Modeling and performance evaluation of an OpenFlow architecture“, *Proceedings of the 2011 23rd International Teletraffic Congress, ITC 2011*, S. 1–7, 2011.
- [20] M. Jarschel, F. Lehrieder, Z. Magyari und R. Pries, „A flexible OpenFlow-controller benchmark“, in *Proceedings - European Workshop on Software Defined Networks, EWSDN 2012*, IEEE, Okt. 2012, S. 48–53.
- [21] S. Azodolmolky, R. Nejabati, M. Pazouki, P. Wieder, R. Yahyapour und D. Simeonidou, „An analytical model for software defined networking: A network calculus-based approach“, *GLOBECOM - IEEE Global Telecommunications Conference*, S. 1397–1402, 2013.
- [22] A. Herzog, *Simulation mit dem Warteschlangensimulator*, Ser. Studienbücher Wirtschaftsmathematik. Wiesbaden: Springer Fachmedien Wiesbaden, 2021.
- [23] A. Roy, J. L. Pachau und A. K. Saha, „An overview of queuing delay and various delay based algorithms in networks“, *Computing*, Jg. 103, Nr. 10, S. 2361–2399, Okt. 2021.
-

-
- [24] W. Yue, Q. Li, S. Jin und Z. Ma, *Queueing Theory and Network Applications*, Ser. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2017, Bd. 10591.
- [25] R. Beuran, *Introduction to Network Emulation*. Jenny Stanford Publishing, Nov. 2012.
- [26] R. P. Goldberg, „Architecture of virtual machines“, in *Proceedings of the June 4-8, 1973, national computer conference and exposition on - AFIPS '73*, New York, USA: ACM Press, 1973, S. 309.
- [27] J. Smith und Ravi Nair, „The architecture of virtual machines“, *Computer*, Jg. 38, Nr. 5, S. 32–38, Mai 2005.
- [28] S. Soltész, H. Pötzl, M. E. Fiuczynski, A. Bavier und L. Peterson, „Container-based operating system virtualization“, in *Proceedings of the 2nd ACM SIGOPS/EuroSys European Conference on Computer Systems 2007*, New York, NY, USA: ACM, Mär. 2007, S. 275–287.
- [29] V. Nguyen, Y. Besanger, Q. Tran und T. Nguyen, „On Conceptual Structuration and Coupling Methods of Co-Simulation Frameworks in Cyber-Physical Energy System Validation“, *Energies*, Jg. 10, Nr. 12, Nov. 2017.
- [30] M. Specht und S. Rohjans, „ICT and Energy Supply: IEC 61970/61968 Common Information Model“, in *Standardization in Smart Grids*, Springer Berlin Heidelberg, 2013, S. 99–114.
- [31] M. Uslar, M. Specht, S. Rohjans, J. Trefke und J. M. Vasquez Gonzalez, *The Common Information Model CIM*, Ser. Power Systems. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, Bd. 66.
- [32] R. Mackiewicz, „Overview of IEC 61850 and Benefits“, in *2006 IEEE PES Power Systems Conference and Exposition*, IEEE, 2006, S. 623–630.
- [33] T. Kostic, O. Preiss und C. Frei, „Towards the formal integration of two upcoming standards: IEC 61970 and IEC 61850“, in *Large Engineering Systems Conference on Power Engineering, 2003*, IEEE, 2003, S. 24–29.
- [34] M. T. Le, V. H. Nguyen, Q. T. Tran, Y. Besanger, T. Braconnier, A. Labonne und H. Buttin, „CIM compliant multiplatform approach for cyber-physical energy system assessment“, in *2017 IEEE Innovative Smart Grid Technologies - Asia (ISGT-Asia)*, IEEE, Dez. 2017, S. 1–6.
-

-
- [35] D. Broman, L. Greenberg, E. A. Lee, M. Masin, S. Tripakis und M. Wetter, „Requirements for hybrid cosimulation standards“, in *Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control*, New York, NY, USA: ACM, Apr. 2015, S. 179–188.
- [36] W. Li, M. Ferdowsi, M. Stevic, A. Monti und F. Ponci, „Cosimulation for Smart Grid Communications“, *IEEE Transactions on Industrial Informatics*, Jg. 10, Nr. 4, S. 2374–2384, Nov. 2014.
- [37] L. Barbierato, E. Pons, A. Mazza, E. F. Bompard, V. S. Rajkumar, P. Palensky, E. Macii, L. Bottaccioli und E. Patti, „Stability and Accuracy Analysis of a Real-time Co-simulation Infrastructure“, in *2021 IEEE International Conference on Environment and Electrical Engineering and 2021 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I&CPS Europe)*, IEEE, Sep. 2021, S. 1–6.
- [38] B. Morgenstern, „Definitionen und Begriffe der Informationsverarbeitung nach DIN 41859 und DIN 44300“, in *Elektronik*, Wiesbaden: Vieweg+Teubner Verlag, 1997, S. 1–2.
- [39] P. M. Menghal und A. J. Laxmi, „Real time simulation: Recent progress & challenges“, in *2012 International Conference on Power, Signals, Controls and Computation*, IEEE, Jan. 2012, S. 1–6.
- [40] S. S. Noureen, N. Shamim, V. Roy und S. B. Bayne, „Real-Time Digital Simulators: A Comprehensive Study on System Overview, Application, and Importance“, *International Journal of Research and Engineering*, Jg. 4, Nr. 11, S. 266–277, Dez. 2017.
- [41] D. Mahrenholz und S. Ivanov, „Real-Time Network Emulation with ns-2“, in *Eighth IEEE International Symposium on Distributed Simulation and Real-Time Applications*, IEEE, 2004, S. 29–36.
- [42] M. Barabanov und V. Yodaiken, „Getting Started with RTLinux“, *Finite State Machine Labs Inc*, 2001.
- [43] J. A. Stankovic und R. Rajkumar, „Real-Time Operating Systems“, *Real-Time Systems*, Jg. 28, Nr. 2/3, S. 237–253, Nov. 2004.
- [44] F. Reghenzani, G. Massari und W. Fornaciari, „The Real-Time Linux Kernel“, *ACM Computing Surveys*, Jg. 52, Nr. 1, S. 1–36, Jan. 2020.
-

-
- [45] S. C. Müller, H. Georg, J. J. Nutaro, E. Widl, Y. Deng, P. Palensky, M. U. Awais, M. Chenine, M. Kuch, M. Stifter, H. Lin, S. K. Shukla, C. Wietfeld, C. Rehtanz, C. Dufour, X. Wang, V. Dinavahi, M. O. Faruque, W. Meng, S. Liu, A. Monti, M. Ni, A. Davoudi und A. Mehrizi-Sani, „Interfacing Power System and ICT Simulators: Challenges, State-of-the-Art, and Case Studies“, *IEEE Transactions on Smart Grid*, Jg. 9, Nr. 1, S. 14–24, Jan. 2018.
- [46] C. Rehtanz und X. Guillaud, „Real-time and co-simulations for the development of power system monitoring, control and protection“, in *19th Power Systems Computation Conference, PSCC 2016*, IEEE, Jun. 2016, S. 1–20.
- [47] T. Blockwitz, M. Otter, J. Akesson, M. Arnold, C. Clauss, H. Elmqvist, M. Friedrich, A. Junghanns, J. Mauss, D. Neumerkel, H. Olsson und A. Viel, „Functional Mockup Interface 2.0: The Standard for Tool independent Exchange of Simulation Models“, in *Proceedings of the 9th International MODE-LICA Conference, September 3-5, 2012, Munich, Germany*, Bd. 76, Munich, Nov. 2012, S. 173–184.
- [48] IEEE, *IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA): Framework and Rules*, IEEE Std 1516–2010, 2010. Adresse: <https://standards.ieee.org/standard/1516-2010.html> (Zugriff am 12. 01. 2026).
- [49] B. P. Zeigler, A. Muzy und E. Kofman, „Chapter 6 - Basic Formalisms: DEVS, DESS, DTSS“, in *Theory of Modeling and Simulation (Third Edition)*, Third Edition, Academic Press, 2019, S. 153–165.
- [50] K. H. Georg, „Co-Simulation Based Performance Evaluation of ICT Infrastructures for Smart Grids“, Diss., Technische Universität Dortmund, 2015.
- [51] M. S. Abdelrahman, I. Kharchouf, T. L. Nguyen und O. A. Mohammed, „A Hybrid Physical Co-Simulation Smart Grid Testbed for Testing and Impact Analysis of Cyber-Attacks on Power Systems: Framework and Attack Scenarios“, *Energies*, Jg. 16, Nr. 23, S. 7771, Nov. 2023.
- [52] C. Steinbrink, A. A. van der Meer, M. Cvetkovic, D. Babazadeh, S. Rohjans, P. Palensky und S. Lehnhoff, „Smart grid co-simulation with MOSAIK and HLA: a comparison study“, *Computer Science - Research and Development*, Jg. 33, Nr. 1-2, S. 135–143, Feb. 2018.
- [53] C. Steinbrink, C. Köhler, M. Siemonsmeier und T. van Ellen, „Lessons learned from CPES co-simulation with distributed, heterogeneous systems“, *Energy Informatics*, Jg. 1, Nr. S1, S. 327–335, Okt. 2018.
-

-
- [54] C. Steinbrink, M. Blank-Babazadeh, A. El-Ama, S. Holly, B. Lüers, M. Nebel-Wenner, R. P. Acosta, T. Raub, J. S. Schwarz, S. Stark, A. Nieße und S. Lehnhoff, „CPES testing with MOSAIK: Co-Simulation planning, execution and analysis“, *Applied Sciences (Switzerland)*, Jg. 9, Nr. 5, S. 923, Mär. 2019.
- [55] B. Palmintier, D. Krishnamurthy, P. Top, S. Smith, J. Daily und J. Fuller, „Design of the HELICS high-performance transmission-distribution-communication-market co-simulation framework“, in *2017 Workshop on Modeling and Simulation of Cyber-Physical Energy Systems (MSCPES)*, IEEE, Apr. 2017, S. 1–6.
- [56] L. Barbierato, P. Rando Mazzarino, M. Montarolo, A. Macii, E. Patti und L. Bottaccioli, „A comparison study of co-simulation frameworks for multi-energy systems: the scalability problem“, *Energy Informatics*, Jg. 5, Nr. S4, S. 53, Dez. 2022.
- [57] S. Ciraci, J. Daily, J. Fuller, A. Fisher, L. Marinovici und K. Agarwal, „FNCS: A framework for power system and communication networks co-simulation“, in *Simulation Series*, Bd. 46, 2014.
- [58] W. Zhong, M. Liu und F. Milano, „A Co-Simulation Framework for Power Systems and Communication Networks“, in *2019 IEEE Milan PowerTech*, IEEE, Jun. 2019, S. 1–6.
- [59] S. C. Müller, H. Georg, M. Küch und C. Wietfeld, „INSPIRE -Co-Simulation von energie-und IKT-Systemen zur evaluation von smart-Grid-Applikationen“, *At-Automatisierungstechnik*, Jg. 62, Nr. 5, S. 315–324, 2014.
- [60] H. Georg, C. Wietfeld, S. C. Müller und C. Rehtanz, „A HLA based simulator architecture for co-simulating ICT based power system control and protection systems“, in *2012 IEEE Third International Conference on Smart Grid Communications (SmartGridComm)*, IEEE, Nov. 2012, S. 264–269.
- [61] A. A. Smadi, B. T. Ajao, B. K. Johnson, H. Lei, Y. Chakhchoukh und Q. Abu Al-Haija, „A Comprehensive Survey on Cyber-Physical Smart Grid Test-bed Architectures: Requirements and Challenges“, *Electronics*, Jg. 10, Nr. 9, S. 1043, Apr. 2021.
- [62] E. Padilla, *Substation Automation Systems*. Wiley, Okt. 2015.
- [63] G. Clarke, D. Reynders und E. Wright, „Intelligent electronic devices (IEDs)“, in *Practical Modern SCADA Protocols*, Elsevier, 2003, S. 312–315.
-

-
- [64] S. Mohagheghi, „A fuzzy cognitive map for data integrity assessment in a IEC 61850 based substation“, in *IEEE PES General Meeting*, IEEE, Jul. 2010, S. 1–7.
- [65] H. S. Oluwatosin, „Client-Server Model“, *IOSR Journal of Computer Engineering*, Jg. 16, Nr. 1, S. 57–71, 2014.
- [66] J. Arrillaga und N. R. Watson, *Computer modelling of electrical power systems: Second edition*. Wiley, Feb. 2013, S. 1–369.
- [67] R. Wójtowicz, R. Kowalik und D. D. Rasolomampionona, „Next generation of power system protection automation-Virtualization of protection systems“, *IEEE Transactions on Power Delivery*, Jg. 33, Nr. 4, S. 2002–2010, Aug. 2018.
- [68] C. Anderson, „Docker [Software engineering]“, *IEEE Software*, Jg. 32, Nr. 3, S. 102–c3, Mai 2015.
- [69] Open Networking Foundation, *OpenFlow Switch Specification Version 1.3.0*. Adresse: <https://opennetworking.org/wp-content/uploads/2014/10/openflow-spec-v1.3.0.pdf> (Zugriff am 12.01.2026).
- [70] S. Panev, „a Delay Modelling in Sdn-Based Mobile Networks With Single Controller“, *Journal of Electrical Engineering and Information Technologies*, Jg. 4, Nr. 1-2, S. 41–49, 2019.
- [71] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown und S. Shenker, „NOX“, *ACM SIGCOMM Computer Communication Review*, Jg. 38, Nr. 3, S. 105–110, Jul. 2008.
- [72] Ryu SDN Project, *Ryu*, Adresse: <https://ryu-sdn.org/> (Zugriff am 12.01.2026).
- [73] P. Berde, M. Gerola, J. Hart, Y. Higuchi, M. Kobayashi, T. Koide, B. Lantz, B. O’Connor, P. Radoslavov, W. Snow und G. Parulkar, „ONOS“, in *Proceedings of the third workshop on Hot topics in software defined networking*, New York, NY, USA: ACM, Aug. 2014, S. 1–6.
- [74] J. Medved, R. Varga, A. Tkacik und K. Gray, „OpenDaylight: Towards a model-driven SDN controller architecture“, in *Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014, WoWMoM 2014*, IEEE, Jun. 2014, S. 1–6.
- [75] B. Lantz, B. Heller und N. McKeown, „A network in a laptop“, in *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, New York, NY, USA: ACM, Okt. 2010, S. 1–6.
-

-
- [76] M. Peuster, H. Karl und S. Van Rossem, „MeDICINE: Rapid prototyping of production-ready network services in multi-PoP environments“, *2016 IEEE Conference on Network Function Virtualization and Software Defined Networks, NFV-SDN 2016*, S. 148–153, 2017.
- [77] S. Hubschneider, S. Kochanneck, B. Bohnet, M. Suriyah, I. Mauser, T. Leibfried, H. Schmeck und M. Braun, „Requirements for Power Hardware-in-the-Loop Emulation of Distribution Grid Challenges“, in *2018 53rd International Universities Power Engineering Conference (UPEC)*, IEEE, Sep. 2018, S. 1–6.
- [78] Q. Huang, S. Jing, J. Li, D. Cai, J. Wu und W. Zhen, „Smart Substation: State of the Art and Future Development“, *IEEE Transactions on Power Delivery*, Jg. 32, Nr. 2, S. 1098–1105, Apr. 2017.
- [79] D. Rösch, S. Nicolai und P. Bretschneider, „Container-based Virtualization of an IEC 61850 Substation Co-Simulation Approach“, in *2022 10th Workshop on Modelling and Simulation of Cyber-Physical Energy Systems, MSCPES 2022*, IEEE, Mai 2022, S. 1–6.
- [80] D. Rösch, S. Nicolai und P. Bretschneider, „Combined simulation and virtualization approach for interconnected substation automation“, in *2021 6th International Conference on Smart and Sustainable Technologies, SpliTech 2021*, IEEE, Sep. 2021, S. 1–6.
- [81] M. Z. Abdullah, N. A. Al-awad und F. W. Hussein, „Performance Comparison and Evaluation of Different Software Defined Networks Controllers“, *International Journal of Computing & Network Technology*, Jg. 06, Nr. 02, S. 36–41, Mai 2018.
- [82] IEEE, *IEEE Working Group 802.1X*. Adresse: <https://1.ieee802.org/security/802-1x/> (Zugriff am 12. 01. 2026).
- [83] Linux Foundation, *Open vSwitch*. Adresse: <https://www.openvswitch.org/> (Zugriff am 12. 01. 2026).
- [84] D. Rösch, K. Schäfer und S. Nicolai, „The Implementation and Evaluation of Virtualized Protection Intelligent Electronic Devices into a Virtual Substation“, *Electricity*, Jg. 5, Nr. 2, S. 385–396, Jun. 2024.
- [85] C. Kramer und W. Elmore, „Flexible inverse overcurrent relaying using a microprocessor“, *IEEE Transactions on Power Delivery*, Jg. 5, Nr. 2, S. 915–923, Apr. 1990.
-

-
- [86] S. Kannuppaiyan und V. Chenniappan, „Numerical inverse definite minimum time overcurrent relay for microgrid power system protection“, *IEEEJ Transactions on Electrical and Electronic Engineering*, Jg. 10, Nr. 1, S. 50–54, Jan. 2015.
- [87] M. A. Aftab, S. Roostae, S. Suhail Hussain, I. Ali, M. S. Thomas und S. Meh-fuz, „Performance evaluation of IEC 61850 GOOSE-based inter-substation communication for accelerated distance protection scheme“, *IET Generation, Transmission and Distribution*, Jg. 12, Nr. 18, S. 4089–4098, Okt. 2018.
- [88] D. Rösch, O. Farghaly, R. Suri, P. Dahane, S. Nicolai und P. Bretschneider, „VirtualSubstation: An IEC 61850 framework for a Containernet based virtual substation“, in *2022 57th International Universities Power Engineering Conference: Big Data and Smart Grids, UPEC 2022 - Proceedings*, IEEE, Aug. 2022, S. 1–6.
- [89] D. Rösch, T. Bauer, S. Nicolai und P. Bretschneider, „Delay modeling for virtualization-based Co-simulation of IEC 61850 substations“, *Journal of Cleaner Production*, Jg. 368, S. 133 179, Sep. 2022.
- [90] X. Zhang, K. Hengster-Movric und M. Sebek, „Distributed observer and controller design for state-output decomposed systems“, in *2016 IEEE Conference on Control Applications (CCA)*, IEEE, Sep. 2016, S. 450–455.
- [91] Y. Goto, H. Masuyama, B. Ng, W. K. Seah und Y. Takahashi, „Queueing analysis of software defined network with realistic openflow-based switch model“, *Proceedings - 2016 IEEE 24th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, MASCOTS 2016*, S. 301–306, 2016.
- [92] W. Leland, M. Taqqu, W. Willinger und D. Wilson, „On the self-similar nature of Ethernet traffic (extended version)“, *IEEE/ACM Transactions on Networking*, Jg. 2, Nr. 1, S. 1–15, 1994.
- [93] Z. Shang und K. Wolter, *Delay Evaluation of OpenFlow Network Based on Queueing Model*, arXiv:1608.06491, 2016.
- [94] D. Rösch, Z. Li, S. Nicolai und J. Seitz, „Traffic-Based Validation of Virtualized Communication Networks“, in *2024 9th International Conference on Smart and Sustainable Technologies, SpliTech 2024*, IEEE, Jun. 2024, S. 1–6.
-

-
- [95] S. Molnar, P. Megyesi und G. Szabo, „How to validate traffic generators?“ , in *2013 IEEE International Conference on Communications Workshops (ICC)*, IEEE, Jun. 2013, S. 1340–1344.
- [96] J. Y. Chung, B. Park, Y. J. Won, J. Strassner und J. W. Hong, „Traffic Classification Based on Flow Similarity“, in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2009, S. 65–77.
- [97] J. M. Chahira, „Model for Intrusion Detection Based on Hybrid Feature Selection Techniques“, *International Journal of Computer Applications Technology and Research*, Jg. 09, Nr. 03, S. 115–124, Mär. 2020.
- [98] KimiNewt, *PyShark*. Adresse: <https://github.com/KimiNewt/pyshark/> (Zugriff am 12.01.2026).
- [99] The Wireshark Foundation, *Wireshark*, Adresse: <https://www.wireshark.org/> (Zugriff am 12.01.2026).
- [100] S. T. Arzo, D. Scotece, R. Bassoli, D. Barattini, F. Granelli, L. Foschini und F. H. P. Fitzek, „MSN: A Playground Framework for Design and Evaluation of MicroServices-Based sdN Controller“, *Journal of Network and Systems Management*, Jg. 30, Nr. 1, S. 19, Jan. 2022.
- [101] F. Paraiso, S. Challita, Y. Al-Dhuraibi und P. Merle, „Model-Driven Management of Docker Containers“, in *2016 IEEE 9th International Conference on Cloud Computing (CLOUD)*, IEEE, Jun. 2016, S. 718–725.
- [102] S. Meier, T. Werner und C. Popescu-Cirstucescu, „Performance considerations in digital substations“, in *13th International Conference on Development in Power System Protection 2016 (DPSP)*, 2016, S. 1–9.
- [103] A. A. Memon und K. Kauhaniemi, „Real-Time Hardware-in-the-Loop Testing of IEC 61850 GOOSE-Based Logically Selective Adaptive Protection of AC Microgrid“, *IEEE Access*, Jg. 9, S. 154 612–154 639, 2021.
- [104] F. P. Hartwig, G. Davey Smith, A. F. Schmidt, J. A. C. Sterne, J. P. T. Higgins und J. Bowden, „The median and the mode as robust meta-analysis estimators in the presence of small-study effects and outliers“, *Research Synthesis Methods*, Jg. 11, Nr. 3, S. 397–412, Mai 2020.
- [105] A. Abdulghaffar, A. Mahmoud, M. Abu-Amara und T. Sheltami, „Modeling and evaluation of software defined networking based 5G core network architecture“, *IEEE Access*, Jg. 9, S. 10 179–10 198, 2021.
-

-
- [106] D. Rösch, S. Ruhe, K. Schäfer und S. Nicolai, „Local anomaly detection analysis in distribution grid based on IEC 61850-9-2 le SV voltage signals“, in *SEST 2019 - 2nd International Conference on Smart Energy Systems and Technologies*, IEEE, Sep. 2019, S. 1–6.
- [107] D. Rösch, T. Bauer, A. Kummerow, M. Kühne, S. Nicolai und P. Bretschneider, „Transformation in substation automation: Cyber-Resilient Digital Substations (CyReDS) in power grids“, *At-Automatisierungstechnik*, Jg. 71, Nr. 9, S. 789–801, Sep. 2023.
- [108] J. Ali, S. Lee und B.-h. Roh, „Performance Analysis of POX and Ryu with Different SDN Topologies“, in *Proceedings of the 2018 International Conference on Information Science and System*, New York, NY, USA: ACM, Apr. 2018, S. 244–249.
- [109] T. Hu, Z. Guo, P. Yi, T. Baker und J. Lan, „Multi-controller Based Software-Defined Networking: A Survey“, *IEEE Access*, Jg. 6, S. 15 980–15 996, 2018.
- [110] A. Alowa und T. Fevens, „Towards Minimum Inter-Controller Delay Time in Software Defined Networking“, *Procedia Computer Science*, Jg. 175, S. 395–402, 2020.
- [111] K. Smida, H. Tounsi, M. Frikha und Y.-Q. Song, „Delay Study in Multi-controller Software Defined Vehicular Network Using OpenDaylight for Emergency Applications“, in *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*, IEEE, Jun. 2019, S. 615–620.
- [112] J. Yan und D. Jin, „A lightweight container-based virtual time system for software-defined network emulation“, *Journal of Simulation*, Jg. 11, Nr. 3, S. 253–266, Aug. 2017.
- [113] C. Scordino, I. M. Savino, L. Cuomo, L. Miccio, A. Tagliavini, M. Bertogna und M. Solieri, „Real-Time Virtualization For Industrial Automation“, in *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, IEEE, Sep. 2020, S. 353–360.
-

Rösch: Ein Beitrag zur Echtzeit-Co-Simulation der Stationsautomatisierung

Die Digitalisierung der Energieversorgung führt zu einer engen Verzahnung von Stromnetz und Kommunikationsinfrastruktur. In digitalen Umspannwerken entstehen komplexe Abhängigkeiten zwischen elektrischen und informationstechnischen Systemen, die neue Anforderungen an Analysemethoden stellen.

Diese Arbeit stellt einen Co-Simulationsansatz für die Stationsautomatisierung digitaler Umspannwerke vor. Der Ansatz ermöglicht eine echtzeitfähige Analyse von Stromnetz und Kommunikationsinfrastruktur unter anwendungsspezifischen Randbedingungen und koppelt Stromnetzsimulation und virtualisierte Kommunikationssysteme auf Basis des Standards IEC 61850.

Das entwickelte Co-Simulationssystem nutzt unabhängig betreibbare Simulatoren mit direkter Kopplung zum Datenaustausch. Die Architektur erfüllt relevante Echtzeitanforderungen und grenzt sich damit von bestehenden Co-Simulationslösungen ab. Konzeption und Validierung mit Fokus auf das zeitliche Verhalten bilden die zentralen Beiträge dieser Arbeit.

