

CONTROL NETWORK PERFORMANCE ENGINEERING

QUALITÄTSORIENTIERTER ENTWURF VON
CSMA-NETZWERKEN DER AUTOMATION

DIPL.-ING. JÖRN PLÖNNIGS

Von der Fakultät Informatik
der Technischen Universität Dresden
zur Erlangung des akademischen Grades
Doktoringenieur (Dr.-Ing.) genemigte

Dissertation

Gutachter:
Prof. Dr.-Ing. habil. Klaus Kabitzsch
Prof. Dr. rer. nat. habil. Peter Buchholz
Prof. Dipl.-Ing. Dr.techn. Dietmar Dietrich

Tag der Einreichung: 2. Februar 2007
Tag der Verteidigung: 23. Mai 2007

KURZFASSUNG

BEIM Entwurf großer Netzwerke der Automation ermöglichen Methoden der Leistungsbewertung den Test und die Dimensionierung eines Netzwerkes bereits vor der Inbetriebnahme und erlauben damit einen effizienten, qualitätsorientierten Entwurf. Es ist jedoch sehr wissensintensiv und aufwendig, entsprechende Analyse- oder Simulations-Modelle aufzustellen und die Ergebnisse auszuwerten, weshalb die Methoden in der Praxis der Automation selten verwendet werden. Viel vertrauter sind dem Entwerfer hingegen die speziellen Software-Tools, mit denen in der Automation Netzwerke entworfen werden. Auf Basis der Datenbanken dieser Tools werden in dieser Arbeit verschiedene Verfahren zur automatischen Systemmodellierung, Verkehrsmodellierung, Leistungsbewertung und Fehlerdiagnose zu einem *Control Network Performance Engineering* kombiniert, welches die Qualitätsbewertung und -beratung nahtlos und ohne Mehraufwand in den Netzwerkentwurf integriert.

ABSTRACT

DURING the design of large automation networks, performance analysis methods can be used for testing and dimensioning the network before implementation and are essential for an efficient and reliable design process. However, setting up the necessary analytical or simulative models is time-consuming, requires in-depth knowledge, and is therefore often not applicable in practice. The network designers are much more used to the design tools used to develop automation networks. Based on these tools' databases various methods for automated system and traffic modeling, performance analysis and diagnoses are combined in the *control network performance engineering* that seamlessly integrates quality analysis and consulting into network design without requiring additional effort.

INHALTSVERZEICHNIS

I. EINFÜHRUNG	1
1. EINLEITUNG	3
2. PROBLEMSTELLUNG	7
2.1. Branchen der Automation und Kommunikation	7
2.2. Entwurf von Netzwerken der Automation	10
2.3. Qualität des Netzwerkentwurfs	13
2.4. Zielsetzung: Verbesserung der Qualität	15
2.4.1. Qualität der Funktionsimplementation	15
2.4.2. Qualität des Kommunikationssystems	17
3. LÖSUNGSKONZEPT	21
3.1. Anforderungen an eine Lösung	21
3.2. Lösungsansätze in der Literatur	23
3.2.1. Leistungsbewertung	23
3.2.2. Automatische Modellierung	25
3.2.3. Automatische Qualitätsbewertung und Fehleranalyse	27
3.3. Control Network Performance Engineering	28
4. NETZWERKPROTOKOLLE DER GEBÄUDEAUTOMATION	31
4.1. Modelle zur Einordnung von Netzwerkprotokollen	31
4.2. Netzwerkprotokolle der Gebäudeautomation	33
II. CONTROL NETWORK PERFORMANCE ENGINEERING	37
5. AUTOMATISCHE MODELLIERUNG DES SYSTEMMODELLS	39
5.1. Grundlegende Aspekte der automatischen Modellierung	39
5.1.1. Informationsquellen zur automatischen Modellierung	41
5.1.2. Beispielnetzwerk	42
5.2. Systemmodell der Bitübertragungsschicht	43
5.3. Systemmodell der Vermittlungsschicht	46

5.4.	Systemmodell der Anwendungsschicht	48
5.5.	Zusammengeführtes Systemmodell der Nachrichtenklassen	51
5.5.1.	Nachrichtenklassen	51
5.5.2.	Übertragungsdienste	53
5.5.3.	Systemmodell der Nachrichtenklassen	54
5.5.4.	Modellierungsgrenzen	57
5.6.	Validierung	58
5.7.	Zusammenfassung	59
6.	AUTOMATISCHE MODELЛИERUNG DES VERKEHRSMODELLS	61
6.1.	Grundlagen der Verkehrsmodellierung	61
6.2.	Modell der Geräteinteraktion	64
6.2.1.	Allgemeiner Modellierungsansatz	64
6.2.2.	Detailliertes Gerätemodell	66
6.2.3.	Abstraktes Gerätemodell	67
6.3.	Zeitverhalten	68
6.3.1.	Bearbeitungszeit der Anwendungsschicht	68
6.3.2.	Bursts	72
6.4.	Quellprozesse	74
6.4.1.	Methoden zur Abtastung	74
6.4.2.	Zeitorientierte Nachrichtengenerierung	76
6.4.3.	Ereignisorientierte Nachrichtengenerierung bei Zustandssignalen	79
6.4.4.	Ereignisorientierte Nachrichtengenerierung kontinuierlicher Signale mit Send-On-Delta	81
6.4.4.1.	Modellierung des Quellprozesses	81
6.4.4.2.	Quellprozess im detaillierten Gerätemodell	83
6.4.4.3.	Quellprozess im abstrakten Gerätemodell	84
6.4.4.4.	Untersuchung charakteristischer Wetterprozesse	86
6.4.4.5.	Untersuchung von Regelungsprozessen	88
6.4.5.	Ereignisorientierte Nachrichtengenerierung kontinuierlicher Signale mit Schwellwerten	90
6.4.6.	Begrenzung der Zwischenankunftszeiten	92
6.4.6.1.	Einfluss im detaillierten Gerätemodell	93
6.4.6.2.	Einfluss im abstrakten Gerätemodell am Beispiel des Send-On-Delta-Algorithmus	95
6.5.	Filterprozesse	97
6.6.	Datenmodell des Gerätemodells	98
6.7.	Validierung	100
6.8.	Modellierungsgrenzen	105
6.9.	Zusammenfassung	106

7. LEISTUNGSBEWERTUNG	109
7.1. Allgemeine Ergebnisanforderungen	109
7.2. Notwendige Leistungskenngrößen	111
7.3. Zusammenhang zwischen den Modellteilen und der Ankunftsra- tenmodifizier u_m	113
7.4. Zusammenfassung	116
8. FEHLERANALYSE UND NUTZERBERATUNG	119
8.1. Einführung	119
8.2. Fehleridentifikation	121
8.2.1. Dienstqualität (Quality of Service)	121
8.2.2. Zusammenhang der Dienstqualität mit den Qualitätsmerk- malen	122
8.2.3. Definition der verwendeten QoS-Charakteristika, QoS-An- forderungen und QoS-Güteklassen	123
8.2.4. Fehleridentifikation mittels QoS-Anforderungen	124
8.2.5. Qualitätsmaß des Netzwerkentwurfs	127
8.2.6. Datenmodell der QoS-Anforderungen und Anwendung	129
8.3. Analyse der Fehlerzusammenhänge	130
8.3.1. Priorisierung von Fehlern durch QoS-Risikoklassen und QoS-Gewichte	130
8.3.2. Fehlerbäume zur Diagnose	131
8.3.3. Fehlerbaum der Kanalauslastung	135
8.3.4. Fehlerbaum der Verzögerungszeit	135
8.3.5. Fehlerbaum der Fehlerwahrscheinlichkeit	138
8.4. Nutzerberatung	140
8.4.1. Einführung	140
8.4.2. Verbesserung einzelner Dienstqualitäten durch Übertragungs- dienste	142
8.4.3. Verbesserung einzelner Dienstqualitäten durch Prioritäten	145
8.4.4. Verbesserung der Lastsituation durch Übertragungsdienste und Prioritäten	146
8.4.5. Verbesserung der Lastsituation durch Anpassung der Ab- tastparameter	148
8.5. Zusammenfassung	151
III. ZUSAMMENFASSUNG	155
9. GRENZEN DER ARBEIT	157
10. ZUSAMMENFASSUNG UND ABGLEICH MIT DER ZIELSTELLUNG	159

INHALTSVERZEICHNIS

11. AUSBLICK	163
APPENDIX	165
DANKSAGUNGSVERZEICHNIS	167
VERZEICHNIS ALLER ZIELE, ANFORDERUNGEN UND PRÄMISSEN	169
ABBILDUNGSVERZEICHNIS	173
TABELLENVERZEICHNIS	175
FORMELVERZEICHNIS	177
ABKÜRZUNGSVERZEICHNIS	181
LITERATURVERZEICHNIS	183
SCHLAGWORTVERZEICHNIS	197

TEIL

I.

EINFÜHRUNG

KAPITEL

1.

EINLEITUNG

*Forschung ist gefährlich: man könnte etwas
Neues entdecken.*

Gerhard Hocher

MIT dem »Intelligenten Haus« werden seit Jahren die Visionen vom Wohnen und Arbeiten der Zukunft verbunden. Im intelligenten Haus wohnt der Mensch nicht nur zum Schutz vor Wettereinflüssen, nein, er lebt im Zentrum eines sich um ihn sorgenden und hegenden Bewusstseins, das seine Wünsche erkennt und gegenwartslos erfüllt. Egal in welchem Raum er sich aufhält, die persönlichen Klimavorstellungen werden realisiert, integrierte Medien liefern nach Bedarf stimmungsgerechte Musik und bieten jederzeit Zugang zu aktuellen und persönlichen Informationen. Dabei verlieren Worte wie Informationsbeschaffung und Datenaustausch im Privaten und auf Arbeit ihre Bedeutung, denn die Allgegenwärtigkeit von Informationen ermöglicht neue raumübergreifende Dimensionen der Teamarbeit. Selbst die Technik, die all dies unterstützt und ermöglicht, verliert ihre eindeutige Lokalisierung und wird zum Teil der Umgebung. Die Intelligenz und letztendlich das Bewusstsein des Hauses entsteht aus dem Beitrag jedes einzelnen technischen Gegenstandes in einem wachsenden Informationsnetzwerk.

Diese vielleicht futuristisch anklingende Skizze entspricht bereits heute dem technisch Möglichen. Moderne Klimaregelungen erlauben ohne weiteres personalisierte Komforteinstellungen. Neue Projektions- und Darstellungsmedien gestatten die unauffällige Integration von Informationen in alle Flächen und Gegenstände, genauso wie das Internet und moderne Funktechnologien den barrierefreien Informationszugang ermöglichen. Letztendlich liegt in der Kompatibilität und Vernetzung der Geräte, der Verteilung von Algorithmen und der dadurch steigenden Komplexität des Systems eine der größten Hürden.

Eine vergleichbare Problematik existiert in der Automatisierungstechnik. Betrachtet man heutige automatisierungstechnische Anlagen, so können sie bereits mehrere tausend Datenpunkte besitzen. Während vor dreißig Jahren ein Haus, eine Maschine oder ein Auto nahezu ohne Elektronik auskam, so ist sie heutzutage unumgänglich. Dabei kann man wie im einleitenden Beispiel über den Sinn und Nutzen vieler komfortorientierter »Features« im direkten Umfeld des Menschen diskutieren, andererseits retten die neuen technischen Möglichkeiten bei Autos Menschenleben und reduzieren den Energie- und Ressourcenbedarf bei der industriellen Produktion oder beim Betrieb eines Gebäudes. Dabei sind die Möglichkeiten, die allein durch das Verknüpfen bereits vorhandener Information entstehen, noch lange nicht ausgeschöpft und ständig kommen neue Informationsquellen hinzu. Doch ist eine Nutzung nur realistisch, sofern die wachsende Komplexität noch beherrscht werden kann.

Für den Entwurf solcher stark vernetzten und komplexen Systeme ist die Reduktion der Komplexität durch Modularisierung und Verteilung von Funktionen und Informationen wesentlich. Dabei verarbeitet jedes Gerät bereits einen Teil seiner Informationen und tauscht autonom fehlende Informationen mit anderen Geräten über ein gemeinsames Netzwerk aus (Kapitel 4). Durch diese Dezentralisierung kann die Komplexität der Einzelsysteme reduziert und die Zuverlässigkeit und Verfügbarkeit des Gesamtsystems gesteigert werden. Das Netzwerk, das alle Geräte miteinander verbindet, wird zu einem elementaren Bestandteil des Systems und zu einer kritischen Fehlerstelle, so dass ein Ausfall des Netzwerks den Ausfall des ganzen Systems nach sich zieht.

Eine absolute und garantierbare Zuverlässigkeit und Verfügbarkeit des Netzwerks kann nicht ohne weiteres erreicht werden. Aufgrund verschiedener Ursachen können Nachrichten bei der Übertragung verloren gehen oder beim Empfänger verspätet eintreffen. Dies kann bei Geräten ein schlechteres und fehlerhaftes Verhalten verursachen mit potentiell fatalen Folgen für die Anlage oder den Menschen. Da das Auftreten der Probleme je nach Netzwerktyp von der Auslastung des Netzwerks und der Konfiguration im Entwurf abhängig ist, gilt es insbesondere dort bereits, später überlastete Netzwerkabschnitte und falsche Parameter zu entdecken. Dem steht entgegen, dass mit einer zunehmenden Verbreitung und sinkenden Margen auch der Entwurf der Netzwerke effizienter gestaltet werden muss. Die Verbesserung der Qualität der Entwürfe muss also mit der Effektivierung einhergehen.

Aufgrund der Komplexität der Systeme sind für eine Qualitätsbewertung passende Software-Tools notwendig. Diese Arbeit entwirft ein solches Tool zur Abschätzung der Leistungsfähigkeit und Qualität von Netzwerkentwürfen. Sie definiert die notwendigen Modelle für das Netzwerk in Kapitel 5 und untersucht die Verkehrseigenschaften in Kapitel 6. Ausgehend von der Leistungsbewertung der Entwürfe in Kapitel 7 wird der Entwurf zuverlässiger Netzwerke vereinfacht, in-

dem mit den Methoden aus Kapitel 8 Entwurfsfehler automatisch gefunden und Lösungsvorschläge angeboten werden.

Dies ermöglicht letztendlich eine erweiterte Entwurfsmethode für Netzwerke, bei der die ständige Leistungsbewertung und Fehlersuche den Entwurf begleitet und damit nicht nur zuverlässigerer Netzwerke ermöglicht, sondern auch die ökonomische Optimierung von Netzwerken und ihren Parametern. Diese in Abschnitt 3.3 detaillierter vorgestellte Methode stellt den Titel der Arbeit, das *Control Network Performance Engineering*¹.

Die Arbeit konzentriert sich auf Netzwerke der Gebäudeautomation, da sie einige besondere Eigenschaften aufweist, die ein Control Network Performance Engineering nahe legt und ermöglicht. Auf diese Besonderheiten im Vergleich zu anderen Branchen der Automation und Kommunikation geht der nächste Abschnitt ein. Hierbei werden insbesondere die Probleme beim Entwurf von verteilten Systemen der Automation diskutiert und hieraus systematisch die Zielstellung und die Voraussetzungen der Arbeit in Kapitel 3 abgeleitet.

¹ Eine Übersetzung wird in Fußnote 6 auf Seite 30 diskutiert.

KAPITEL

2.

PROBLEMSTELLUNG

*Am Anfang jeder Forschung steht das
Staunen. Plötzlich fällt einem etwas auf.*

Wolfgang Wickler

2.1. BRANCHEN DER AUTOMATION UND KOMMUNIKATION

NETZWERKE sind sowohl in der Automation wie auch in der Kommunikati-
onstechnik zum wichtigsten Übertragungsmedium geworden. Obwohl beide
Branchen unterschiedliche Anforderungen stellen, entwickeln sie doch zunehmend
Berührungspunkte bis hin zu gemischten Lösungen. Dieser Abschnitt vergleicht
die Branchen in ihren Anforderungen an das Netzwerk und an den Netzwerk-
entwurf und leitet daraus die Problemstellung in den folgenden Abschnitten ab.
Hierbei ist allen Branchen gemein, dass die Netzwerke auf der einen Seite zuneh-
mend komplexer werden, während auf der anderen Seite der Netzwerkentwurf
effizienter und preiswerter werden soll. Auf diesem sich herausbildenden Gegen-
satz liegt das Augenmerk des Vergleichs. Dabei wird die Komplexität anhand der
Netzwerkgröße und Netzwerkkapazität sowie den Anforderungen an die *Echtzeit*
(en. *Real Time*)¹ und *Betriebssicherheit* (en. *Safety*)² untersucht. Die Effizienz

1 Echtzeitfähigkeit ist nach der VDI 3687 [218] gegeben, wenn die auftretenden Verzögerungs-
zeiten in der Kommunikation den Prozess nicht negativ beeinflussen oder stören.

2 Sicherheit beschreibt die Freiheit von unververtretbaren Risiken nach der IEC 61511 [91]. Um
Zweideutigkeiten zu vermeiden, wird in diesem Sinne explizit der Begriff Betriebssicherheit
verwendet. Den Begriff *Risiko* (en. *Risk*) definiert die [91] als Kombination der Wahr-
scheinlichkeit des Auftretens eines Schadens und des Schweregrades desselben.

2. PROBLEMSTELLUNG

des Entwurfes steht im Zusammenhang zu vorhandenen Mustern³ im Entwurf, der Investition pro Gerät und der Betriebsdauer. Die branchenspezifischen Eigenschaften sind in Tabelle I zusammengefasst und werden im Verlauf des Kapitels 2 zur Problemanalyse herangezogen. Die Tabelle II bewertet abschließend die Branchen in ihrem Problembezug.

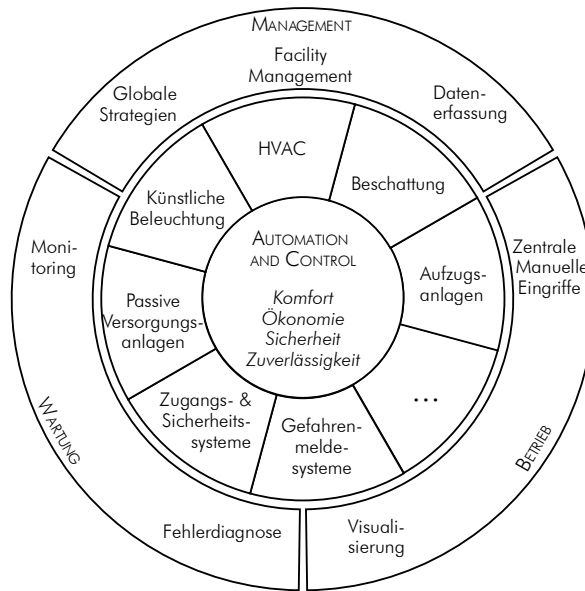


ABBILDUNG 2.1. Funktionen der Gebäudeautomation (ergänzt KASTNER et al. [115]).

Der Schwerpunkt der Arbeit liegt in der *Gebäudeautomation* (en. *Building Automation and Control*). Sie umfasst nach der ISO 16484-1 [96] alle automatisierungstechnischen Anlagen der *Technischen Gebäudeausrüstung* (TGA). Darunter zählen neben Heizungs-, Kühl- und Air-Conditioning-Systemen (HAVAC⁴), die Beschattung, Beleuchtung und weitere im Innenkreis der Abbildung 2.1 dargestellten Anlagen⁵. Die unterschiedlichen Anlagentypen werden in ihrer Hardware ausführlich von DANIELS [31] und SO & CHAN [205] behandelt, wohingegen

3 Unter Mustern werden Entwurfsteile oder Vorgänge verstanden, die innerhalb eines Entwurfes wiederholt auftreten.

4 HAVAC: en. Heating, Ventilation, Air-Conditioning

5 Nach [96] zählen *Informationstechnische Anlagen* wie Gefahrenmeldesysteme, Systeme zur Zutrittskontrolle und Zeiterfassung, Video-Überwachung und Personensuche nicht zur Gebäudeautomation, nutzen jedoch zunehmend die gleichen Netzwerke [188] und werden deshalb in dieser Arbeit nicht separiert.

KRANZ [130] und der VDI [225] in einigen Beispielen auf die Softwarefunktionen eingehen. Hieraus wird in Unterabschnitt 5.1.2 ein Beispiel vorgestellt.

In der Vergangenheit waren die Anlagen der Gebäudeautomation in einzelne handwerkliche Aufgabenbereiche aufgeteilt. Diese sogenannten *Gewerke* wurden separat entworfen und realisiert, bis hin zu einer eigenen, zentralistisch organisierten Verkabelung. Die Entwicklung leistungsfähiger, dezentraler Kommunikationsnetzwerke ermöglicht diese zunehmend gemeinsam zu nutzen, wodurch die Trennung zwischen den Gewerken abnimmt [129]. Dadurch entstehen zum Teil sehr große Netzwerke mit mehreren tausend Geräten [87], die aufgrund sich wiederholender Raumstrukturen große Muster aufweisen [76] und verhältnismäßig schnell entworfen werden können. Doch nicht nur die Gerätezahl steigt, sondern auch die unterschiedlichen, gewerkespezifischen Anforderungen an die Echtzeit und Betriebssicherheit treffen in den Netzwerken zusammen. Zusätzlich erlaubt die Verfügbarkeit von Informationen anderer Gewerke den Einsatz und die Entwicklung synergetischer Algorithmen auf Feldebene, die bisher — wenn überhaupt — der Managementebene (Abschnitt 4.1) vorbehalten waren. Dadurch können zum Beispiel der Energiebedarf reduziert [32, 33, 170] oder abstrakte Metainformationen für neue Komfort- und Sicherheitsfunktionen gewonnen werden [182]. Diese Zunahme der *Komplexität* der Netzwerke, die sich sowohl in einem Wachstum der Gerätezahl, einer höheren Interaktion, als auch in einer Diversifizierung der Echtzeit- und Betriebssicherheitsanforderungen ausdrückt, ist branchenübergreifend zu beobachten.

Während sich die Gebäudeautomation auf den Zweckbau konzentriert, geht die *Heimautomation* (en. *Home Automation*) auf die Bedürfnisse des Wohnbaus ein und integriert zusätzlich Hausgeräte und Multimediatechnik [76]. Hierzu analysieren die Studien von BRETTHAUER et al. [12] und BROY et al. [14] die Entwicklungstendenzen. Im Vergleich zur Gebäudeautomation sind die Anlagen kleiner und dadurch individueller, da sich keine Muster ausprägen. Die Anforderungen an die Echtzeit- und Betriebssicherheit werden meist vernachlässigt.

Die industrielle Automation der *Fertigungs-* und *Prozesstechnik* umfasst die automatisierungstechnischen Anlagen in der mechanisch orientierten Fertigung und der prozessorientierten physikalischen und chemischen Industrie. Die Anlagen können sich räumlich sehr weit erstrecken, umfassen aber meist weniger Geräte als die Gebäudeautomation [131, 196]. Da die Anlagen je nach Aufgabe sehr individuell entworfen werden, besitzen sie wenig Muster. Die Echtzeit- und Betriebssicherheitsanforderungen umfassen in der industriellen Automation das ganze Spektrum von gering bis sehr hoch.

Die *Fahrzeugtechnik* erlebte in den letzten Jahren eine deutliche Zunahme der Netzwerkkomplexität bis zu mehreren hundert Geräten [13]. Da heutzutage zunehmend kritische Aufgaben elektronisch gelöst werden, sind die Anforderungen an Echtzeit und Betriebssicherheit in der Fahrzeugtechnik sehr hoch [13, 126, 136] und für weitere Entwicklungen wird als Voraussetzung eine hohe Netzwerkzu-

verlässigkeit gesehen [136]. Der Entwurf eines Fahrzeugtyps ist individuell und beinhaltet nur wenige Muster. Er wird erst bei der Produktion vervielfältigt.

So wie die Fahrzeugtechnik den Siegeszug der Automationsnetzwerke demonstriert, verdeutlicht das Internet die Vernetzung in der Kommunikationstechnik. Als nahezu weltweit verfügbares Kommunikationsmedium findet es wachsende Verbreitung in der Automation. Insbesondere im Zweckbau bieten sich die Computernetzwerke der *Bürokommunikation* als Medium für die Gebäudeautomation an, so dass in [23, 73, 129] die Verschmelzung zu *Computer Integrated Buildings* prognostiziert wird. Deshalb ist ein Vergleich der Computernetzwerke der Bürokommunikation mit den Netzwerken der Automation unumgänglich. Computernetzwerke bestehen in der Bürokommunikation zum Großteil aus Arbeitsplatzrechnern, weshalb sie zahlenmäßig kleiner sind und es weniger klare Entwurfsmuster und determinierte Interaktionsmuster gibt. Die Anforderungen an Echtzeit und Betriebssicherheit sind meistens gering.

Bei den bisher besprochenen Netzwerken der Bürokommunikation und der Automatisierungstechnik handelt es sich im Allgemeinen um sogenannte lokale Netzwerke (LAN⁶). Daneben gibt es großräumige Netzwerke die in MAN⁷, WAN⁸ und GAN⁹ eingeteilt werden [26, 211]. Diese hier zusammenfassend als WAN bezeichneten Netzwerke aggregieren üblicherweise LANs, wodurch eine hohe Anzahl an Endgeräten entsteht. Auch in den anderen Eigenschaften gibt es deutliche Unterschiede zu LANs, weshalb WANs in Tabelle I separat betrachtet werden.

Um die Problemstellung aus diesem Vergleich herauszuarbeiten, untersucht der nächste Abschnitt detaillierter das Entwurfsvorgehen, bevor in Unterabschnitt 2.4.2 die Branchen in Bezug auf die Eignung für die Zielsetzung der Arbeit abschließen bewertet werden. Da das Entwurfsvorgehen in den Branchen sehr unterschiedlich ist, wird das Entwurfsvorgehen am Beispiel der Gebäudeautomation vorgestellt und daraus werden die entscheidenden Eigenschaften abgeleitet.

2.2. ENTWURF VON NETZWERKEN DER AUTOMATION

Unter dem *Entwurf* eines Gebäudeautomationssystems wird der Prozess von der Planung bis zur Inbetriebnahme der automatisierungstechnischen Anlagen der TGA bezeichnet. Die Abbildung 2.2 aus [38] gibt einen chronologischen Überblick über die einzelnen Entwurfsschritte, die repräsentativ für alle Netzwerke der Automation sind. Für die Arbeit sind nur die Schritte wichtig, die das Kommunikationssystem betreffen. Sie sind in der Abbildung *kursiv* geschrieben und

6 LAN: en. Local Area Network

7 MAN: en. Metropolitan Area Network (Großstadtnetzwerk)

8 WAN: en. Wide Area Network (Weitverkehrsnetzwerk)

9 GAN: en. Global Area Network (Globale Netzwerke)

TABELLE I.

Vergleich verschiedener Branchen der Automatisierungstechnik und Bürokommunikation in: (Oben) Geräteanzahl, ausgeprägte Entwurfsmuster, Betriebsdauer der Geräte, mittlere Anforderungen bzgl. Echtzeit und Betriebssicherheit [131, 181, 196, 218]; (Unten) Netzwerkkapazität, Entwurfskosten und Investition pro Gerät, Vorhandensein von elektronischem Entwurfswissen, Eigenschaften des Nachrichtenverkehrs.

	Gerätezahl Kapazität	Muster Kosten	Betriebsdauer Datenbanken	Anforderungen Verkehr
Gebäudeautomation	$< 2 \cdot 10^4$ klein	häufig mittel	10-30 Jahre ja	mittel gleichmäßig
Heimautomation	< 100 klein	selten klein	5 Jahre ja	gering gleichmäßig
Fertigungs- & Prozesstechnik	< 1000 klein	selten groß	10-30 Jahre teilweise	hoch z. T. burst
Fahrzeugtechnik	< 500 klein	selten mittel	10 Jahre generierbar	sehr hoch z. T. burst
Bürokommunikation	< 1000 groß	selten klein	5 Jahre nein	gering burst
WAN	$< 10^7$ sehr groß	selten groß	10 Jahre nein	gering burst

werden von FISCHER [64] als *Engineering* bezeichnet. Hierunter fallen nach der ISO 16484-1 [96] die Konfigurierung und Inbetriebnahme der Anlage.

Die *Konfigurierung* (en. *Configuration*) umfasst mehrere Arbeitsschritte und ist der Inbetriebnahme zeitlich vorgelagert (Abbildung 2.2). Es handelt sich dabei um einen abstrakten softwarebasierten Entwurf des Systems, der ohne die reale Anlage durchgeführt wird. Das System wird in einem geeigneten *Entwurfstool* (z. B. [2, 47, 59, 169, 209]) nahezu vollständig geplant und parametrisiert. Das Ergebnis der Konfigurierung ist die *Konfiguration*, die im Folgenden als *Netzwerkentwurf* bezeichnet wird, wobei das *Netzwerk* sowohl das Komponentennetzwerk als auch das Kommunikationsmedium umfasst. KRIESEL et al. [131] und SCHWARZ [201] schätzen, dass die Konfigurierung eines Automationsnetzwerks mit 35 Prozent den größten Teil der Kosten¹⁰ in seiner gesamten Lebensdauer ausmacht und damit prädestiniert für Einsparungen ist.

Nach der Montage der Geräte in der Anlage wird bei der *Inbetriebnahme* (en. *Commissioning*) der vollständige Netzwerkentwurf von dem Entwurfstool in die reale Anlage übertragen, wodurch die Geräte ihre zuvor festgelegten logischen Adressen, Kommunikationsbeziehungen und Betriebsparameter zugewiesen be-

¹⁰ neben 30% Softwareanpassung, 25% Wartung & Instandhaltung und 10% Hard- & Softwarekosten

2. PROBLEMSTELLUNG

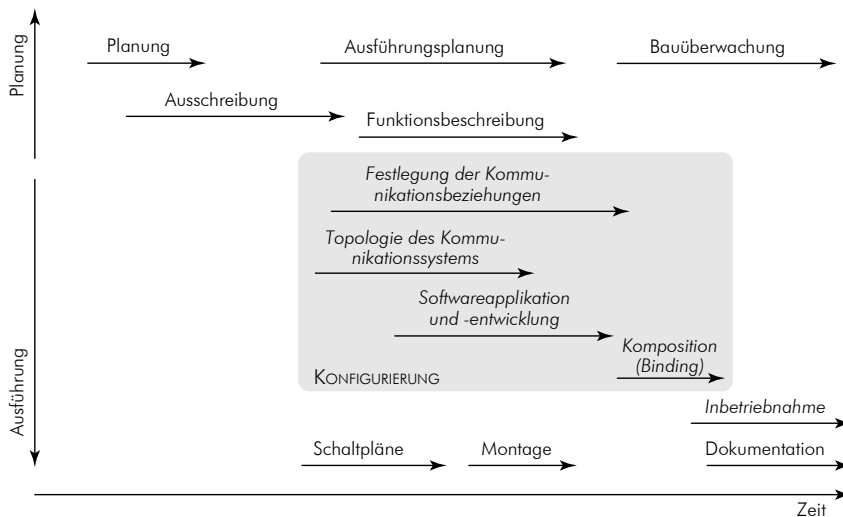


ABBILDUNG 2.2. Entwurfsprozess von Systemen der Gebäudeautomation [38].

kommen. Diese zeitliche Trennung von Konfiguration und Inbetriebnahme verkürzt insbesondere den Aufenthalt auf der Baustelle und damit die Bauzeit. Dadurch besteht allerdings bei der Konfiguration nicht die Möglichkeit für praktische Messungen, um die Funktion der Anlage zu prüfen und sie passend zu dimensionieren und sie können erst bei der Inbetriebnahme durchgeführt werden. Mit der Konsequenz, dass immer wieder Dimensionierungsentscheidungen revidiert werden. Erschwerend kommt hinzu, dass die Inbetriebnahme am Ende der Projektzeit liegt und durch den Kunden fest terminiert ist, so dass Fehler schnell behoben werden müssen.

Diese Problematik verdeutlicht sich in den Ergebnissen einer Umfrage der TU Dresden [109, 181]. Alle befragten Systemintegratoren gaben an, dass Probleme nicht nur häufiger erst bei der Montage und Inbetriebnahme auftreten, sondern auch Nachbesserungen unumgänglich sind und einen zur Montage vergleichbaren Zeitaufwand haben. Dabei sind 70 % der genannten Problemursachen der Konfigurationsphase zuzuordnen¹¹. Sinnvollerweise sollten die Probleme bereits in dieser Phase identifiziert und behoben werden, um damit das finanzielle und terminliche Risiko der Konfiguration und die Kosten für die Inbetriebnahme und Nachbesserungen zu reduzieren. In diesem Zusammenhang diskutiert der nächste Abschnitt den Begriff der Qualität eines Netzwerkentwurfs.

¹¹ neben 33 % Inbetriebnahme/Montage und 13 % Betrieb (Überschneidungen möglich)

2.3. QUALITÄT DES NETZWERKENTWURFS

Unter *Qualität* wird entsprechend der ISO 9000 [100] der Grad verstanden, in dem inhärente Merkmale eines Produktes die gestellten Anforderungen erfüllen. Die *Qualität des Netzwerkentwurfs* bezieht sich demnach nicht auf die Qualität des Entwurfsvorgehens oder der Entwurfsdokumentation, sondern auf die Qualität des resultierenden Netzwerks als Produkt. Die Anforderungen an ein Netzwerk leiten sich aus den Anforderungen an das Gesamtsystem ab. Für ein Gebäudeautomationssystem zum Beispiel wurden aus [5, 7, 21, 35, 121, 149] die folgenden Anforderungen zusammengestellt:

- Funktionale Anforderungen
 - A1 Gewährleistung der Zufriedenheit der im Gebäude arbeitenden Menschen.
 - A2 Umfassende Betriebszustandsübersicht für den Anwender.
 - A3 Minimaler Bedienungs- und Wartungsaufwand.
 - A4 Gebäudeautomation soll dem Umweltschutz dienen.
 - A5 Planbarkeit der Instandhaltungsaufwendungen und Erhöhung der Gebäudelebensdauer.
 - A6 Herstellerunabhängigkeit und spätere Erweiterbarkeit.
 - A7 Akzeptanz des Anwenders (Ästhetik, Lernbarkeit, Kosten)
- Qualitative Anforderungen
 - A8 Garantie der Einhaltung geforderter Prozessparameter innerhalb vorgegebener Toleranzen.
 - A9 Gewährleistung eines optimalen, sicheren und gefahrlosen Betriebes.
 - A10 Gezielte, schnelle Reaktionsmöglichkeiten des Systems bei Störungen.
 - A11 Verringerung der Ausfallzeiten.

Die ersten sieben Anforderungen (A1 bis A7) beziehen sich auf die zu realisierenden Funktionen in der Gebäudeautomation. Sie werden bereits vor der Konfigurierung des Netzwerks festgelegt und stellen damit keine direkten Anforderungen an die Konfigurierung; vorausgesetzt, sie werden in der Konfigurierung korrekt umgesetzt. Damit definiert sich das erste *Qualitätsmerkmal* des Netzwerkentwurfs:

QUALITÄTSMERKMAL 1: KORREKTE FUNKTIONSIMPLEMENTATION

Während der Konfigurierung muss jede Funktion in der Form umgesetzt werden, dass die Geräte die geforderte Funktionalität in Abwesenheit von Kommunikationsfehlern korrekt erfüllen können.

Bei diesem Qualitätsmerkmal werden Kommunikationsfehler ausgeschlossen. Sie sind für die Arbeit von besonderem Interesse und werden deshalb mit spezifischen Qualitätsmerkmalen separat behandelt. Diese Qualitätsmerkmale der Kommunikation leiten sich aus den unteren Anforderungen A8 bis A11 ab und

lassen sich auf alle Netzwerke der Automation übertragen [28, 70, 218]. Werden diese Qualitätsmerkmale nicht auf den ganzen Netzwerkentwurf bezogen, sondern nur auf einzelne Kommunikationsbeziehungen, so spricht man von *Dienstqualität* (QoS¹²) [28, 206, 208, 212], die in Unterabschnitt 8.2.1 noch einmal thematisiert wird. Aus der Anforderung A8 leitet sich das folgende Qualitätsmerkmal ab:

QUALITÄTSMERKMAL 2: GARANTIERBARKEIT

Die Einhaltung geforderter Prozessparameter muss in Bezug auf das Kommunikationsmedium innerhalb vorgegebener Toleranzen garantiert und nachgewiesen werden.

Aus den Anforderungen A9 bis A11 folgen dazu die Prozessparameter:

QUALITÄTSMERKMAL 3: BETRIEBSSICHERHEIT

Ein sicherer und gefahrloser Betrieb muss in Bezug auf das Kommunikationsmedium im Rahmen von Qualitätsmerkmal 2 garantiert werden.

QUALITÄTSMERKMAL 4: ECHTZEITVERHALTEN

Eine rechtzeitige Reaktion des Systems bei Störungen muss in Bezug auf das Kommunikationsmedium im Rahmen von Qualitätsmerkmal 2 garantiert werden.

QUALITÄTSMERKMAL 5: VERFÜGBARKEIT

Die Verfügbarkeit des Systems soll in Bezug auf das Kommunikationsmedium im Rahmen von Qualitätsmerkmal 2 abgeschätzt werden.

Neben diesen technischen Merkmalen gibt es auch ökonomische Merkmale wie die Installationskosten. Sie beeinflussen nach DEWSBURY et al. [36] die Akzeptanz der Lösung und haben in der Praxis einen entscheidenden Stellenwert. Dies bestätigt STILLER [208], der die größtmögliche Auslastung der Netzwerkressourcen als Maximalziel der Netzbetreiber ansieht. Ökonomische Merkmale sind über Kostenmodelle quantifizierbar. Das Aufstellen solcher Kostenmodelle ist methodisch von HASSEL [83] beschrieben und mit entsprechendem Aufwand und Kenntnissen möglich. Es soll deshalb nicht Gegenstand dieser Arbeit sein, sondern vielmehr wird ein technisch orientierter Ansatz verfolgt, bei dem ein akzeptables Risiko nach den Qualitätsmerkmalen 2 bis 5 definiert und die Kosten minimiert werden sollen.

QUALITÄTSMERKMAL 6: ANGEMESSENE KOSTEN

Das Kommunikationsmedium soll den technischen und ökonomischen Anforderungen nach angemessen dimensioniert werden, wobei die technischen Mindestanforderungen erfüllt sein müssen und eine Minimierung der Kosten angestrebt wird.

Die aufgeführten Qualitätsmerkmale bilden den Rahmen für eine Bewertung der Qualität eines Netzwerkentwurfs. Die konkrete Ausprägung und Relevanz der einzelnen Qualitätsmerkmale hängt stark mit der Branche (Tabelle I) und den realisierten Funktionen zusammen. Neben diesen Qualitätsmerkmalen gibt

¹² QoS: en. Quality of Service

es noch weitere Merkmale, die für die Qualität eines Netzwerkentwurfs relevant sein können, wie zum Beispiel eine gute Dokumentation. DEWSBURY [35] bietet hierzu für die Heimautomation eine umfangreiche Klassifikation.

Weichen die Eigenschaften eines Netzwerks von den gestellten Anforderungen ab, so wird dies in der ISO 9000 [100] als *Fehler* (en. *Fault*) bezeichnet und in dieser Arbeit spezifischer als *Entwurfsfehler*. Der verwendete Fehlerbegriff unterscheidet sich somit von der technischen Definition als abnormaler Zustand¹³ und bezieht sich auf Qualitätsmerkmale. Probleme, die nicht im direkten Zusammenhang mit der Konfigurierung stehen, wie Alterung, Verschleiß oder Fehlbedienung, beeinflussen nicht die Entwurfsqualität und werden damit von der Betrachtung ausgeschlossen. Zur Steigerung der Qualität von Netzwerkentwürfen müssen Entwurfsfehler vermieden, erkannt und behoben werden.

2.4. ZIELSETZUNG: VERBESSERUNG DER QUALITÄT

2.4.1. QUALITÄT DER FUNKTIONSIMPLEMENTATION

Um Entwurfsfehler zu vermeiden, werden hauptsächlich Erfahrungswerte angewendet, die nicht nur spezifische Lösungen umfassen wie *Entwurfsmuster* (en. *Design Pattern*), sondern auch grundsätzliche Entwurfparadigmen, -sprachen und -methoden. Hierbei liefert eine *Entwurfsmethode* ein Vorgehensmodell, die *Entwurfssprache* definiert Regeln für die Notation des Entwurfes und ein *Entwurfsparadigma* beschreibt grundlegende Denkweisen und Ansätze. Je nach Tiefe der Synthese reichen die Erfahrungswerte von der individuellen Best Practice bis hin zu meist allgemeinen, formalisierten Standards. Der Vorteil von formalisierten Standards liegt in der hohen Bewährtheit und der besseren Transparenz für andere, die diese Standards auch kennen, wodurch der Austausch gefördert und die Überprüfbarkeit erleichtert wird. Der Nachteil ist die geringere Individualität bei spezifischen Problemen.

Ein Beispiel hierfür ist der bei Netzwerken der Automation übliche Entwurf mit Funktionsblöcken (Abschnitt 5.4). Diese ursprüngliche Best Practice erlaubt eine Systemanalyse der Entwurf Funktionen in separierbaren Modulen und verringert dadurch die Komplexität für den Entwerfer, erhöht die Transparenz für andere und begünstigt die Wiederverwendung. Mit [57, 58, 97] gibt es in nahezu allen Branchen der Automation Standards, die formale Sprachen zur Beschreibung, Darstellung von und den Umgang mit diesen Funktionsblöcken regeln. Diese Standardisierung der formalen Sprachen ist somit gut etabliert und akzeptiert. Die Standardisierung von einzelnen Funktionsblöcken und Entwurfsmustern ist

¹³ Die [91, §11] definiert einen Fehler als abnormalen Zustand, der eine Verminderung oder den Verlust der Fähigkeit einer Funktionseinheit, eine geforderte Funktion auszuführen, verursachen kann.

hingegen mit größeren Umsetzungs- und Akzeptanzschwierigkeiten verbunden, da sie spezifischer sind und dadurch schwerer zu verallgemeinern und wiederzuverwenden. Je nachdem, wie geeignet diese Erfahrungswerte für eine spezifische Entwurfsaufgabe und den Entwerfer sind, verbessern sie die Entwurfsqualität. So sind zum Beispiel Funktionsblöcke durch ihre starke Kapselung nicht für jede Aufgabenstellung geeignet.

Viele moderne Entwurfstools nutzen nicht nur solche Methoden und Sprachen, sondern überprüfen auch die definierten Syntax- und Konsistenzregeln. So unterstützen die Entwurfstools [2, 47, 59, 169] beispielsweise den funktionsblockorientierten Entwurf und prüfen, inwiefern Ausgangsdatenpunkte auf Eingangsdatenpunkte unterschiedlicher Funktionsblöcke gebunden werden. Diese formalen Tests mindern syntaktische Entwurfsfehler beim Festlegen der Kommunikationsbeziehungen, eine Teilaufgabe der Konfigurierung, die im Weiteren als *Komposition* bezeichnet wird. Damit wird das Qualitätsmerkmal 1 durch die Entwurfstools auf syntaktischer Ebene bereits unterstützt.

Je weiter die Entwurfsfehler bei der Komposition auf der semantischen Ebene liegen, desto schwieriger ist eine Fehlererkennung. Semantische Fehler sind zum Beispiel Probleme der Interoperabilität von Geräten, das heißt, zwei Geräte können zwar miteinander fehlerfrei Nachrichten austauschen (Syntax), aber dennoch keine verwertbare Kommunikation aufbauen, da sie die Inhalte der Nachrichten nicht verstehen oder einordnen können.

Dieses Interoperabilitätsproblem versucht man durch eine Standardisierung verschiedener semantischer Tiefen zu erreichen, beginnend bei semantischen Nachrichtentypen (z. B. Temperatur: -274 bis 6279°C) [56, 144], interoperablen Datenpunkten [98, 192], bis zu funktionalen Geräteprofilen [56, 98, 145]. DIETRICH et al. [40] und KABITZSCH et al. [108] geben hierzu einen umfassenderen Einblick. Weiterführende Ansätze versuchen, semantische Informationen für einen automatischen Entwurf zu nutzen [22, 37, 114, 192] und dadurch die Interoperabilitätsproblematik zu vermeiden. Auf einer höheren semantischen Ebene sind Probleme durch inkonsistente, zustandsabhängige Parametrierungen einzuordnen, bei denen die Geräte ihre Nachrichten zwar prinzipiell verstehen können, aber im aktuellen Zustand falsch interpretieren. Hier ermöglichen formale Verifikationsmethoden (z. B. Zustandsreichbarkeit, Blockade) oder die Simulation von Teilsystemen [186, 227, 229] detaillierte Tests. Sie erfordern allerdings ein detailliertes Anwendungsmodell, das nur aufwendig zu erstellen ist, so dass diese Methoden vorwiegend bei kritischen Anwendungen eingesetzt werden [84, 213]. In dieser Arbeit werden solche detaillierten, semantischen Tests nicht vorgenommen, allerdings können die in Unterabschnitt 3.2.1 vorgestellten Verfahren zur automatischen Modellierung den Aufwand reduzieren.

Um Wechselwirkungen durch solche semantischen Entwurfsfehler für diese Arbeit auszuschließen, folgt die erste *Prämisse*. Weitere Prämissen werden im Verlauf der Arbeit ergänzt und gelten ab dem Zeitpunkt der Definition als Vor-

aussetzung und Einschränkung für die folgenden Arbeitsschritte. Sie stellen die Randbedingungen der Arbeit dar und werden im Appendix noch einmal zusammengefasst. Es gilt:

PRÄMISSE 1: KORREKTE FUNKTIONSIMPLEMENTATION

In der Arbeit wird Qualitätsmerkmal 1 vorausgesetzt. Dementsprechend ist davon auszugehen, dass die Funktionen in den Geräten korrekt implementiert sind und in der Konfigurierung eine jede Funktion so eingebunden wird, dass die Geräte die geforderte Funktionalität in Abwesenheit von Kommunikationsfehlern korrekt erfüllen können.

Neben den Entwurfsfehlern durch falsche Komposition können Geräte selbst im Verhalten von ihrer Spezifikation abweichen, Programmierfehler enthalten oder technisch defekt sein. Solche Fehlerquellen müssen durch eine durchgehende Qualitätskontrolle vom Geräteentwurf, der Produktion bis zur Montage geprüft werden und liegen außerhalb des Verantwortungsbereichs der Konfigurierung.

2.4.2. QUALITÄT DES KOMMUNIKATIONSSYSTEMS

Das Kommunikationsmedium steht im Zentrum des Nachrichtenaustausches unter den Geräten und hat damit großen Einfluss auf die Qualität des Netzwerkentwurfs. Bei einer idealen Kommunikation kommen alle Nachrichten unverzüglich und verlustfrei an. In der Realität sind lange Übertragungszeiten, Nachrichtenverluste und -veränderungen nicht auszuschließen. Dadurch entstehen unterschiedliche Störungen, wie zum Beispiel instabile Regelkreise oder inkonsistente Schaltzustände [141, 175, 206, 228]. Insgesamt standen von den in der Umfrage [109, 181] bewerteten Problemursachen 42% im Zusammenhang mit dem Verhalten des Kommunikationsmediums und eine Verbesserung der Kommunikationsqualität ermöglicht demzufolge eine Qualitätsverbesserung des Netzwerkentwurfs.

Einen wichtigen Einfluss auf die Qualität der Kommunikation hat die Netzwerkauslastung. Mit ihr nehmen die Übertragungszeiten und Nachrichtenverluste zu, so dass infolgedessen regelmäßig Störungen entstehen. In diesem Fall ist der Netzwerkentwurf *unterdimensioniert*. Entsprechend den Anforderungen an die Echtzeit und die Betriebssicherheit liegt hierin ein Qualitätsverlust des Netzwerkentwurfs, der schwere Folgen haben kann und ein entsprechendes *finanzielles Risiko*¹⁴ birgt. Dieses finanzielle Risiko bestimmt letztendlich, ob ein Qualitätsverlust toleriert werden kann; ein Nachrüsten der Netzwerke notwendig ist oder jegliche Überlastungsprobleme drastisch vermieden werden müssen. Eine Vermeidungsstrategie ist bewusstes *Überdimensionieren*, so dass die geplante Kapazität den

¹⁴ Das finanzielle Risiko ist eine Kombination der Wahrscheinlichkeit des Auftretens eines Schadens und der finanziellen Kosten zum Beheben desselben. Es entspricht einer geldwerten Bewertung des Risikos in der Definition der IEC 61511 [91]. Es wird hier verwendet, um einen vergleichbaren Kostenbegriff zu schaffen.

2. PROBLEMSTELLUNG

TABELLE II.

Bewertung verschiedener Branchen in ihrer Eignung für eine genaue Dimensionierung und Qualitätsbewertung. Bewertet wurde, inwiefern die Eigenschaften aus Tabelle I den förderlichen Eigenschaften im Tabellenkopf entsprechen, wobei die Symbole + zutreffend, o neutral und – unzutreffend bedeuten.

Förderliche Eigenschaft	Große	Regelmäßige	Lange	Mittlere
	Gerätezahl	Muster	Betriebsdauer	Anforderungen
	Geringe	Mittlere	Vorhanden	Gleichmäßiger
	Kapazität	Investition	Datenbanken	Verkehr
Gebäudeautomation	+	+	+	+
Heimautomation	–	–	–	+
Fertigungs- & Prozesstechnik	+	–	+	–
Fahrzeugtechnik	o	–	+	–
Bürokommunikation	+	–	–	+
WAN	+	–	+	+
	o	+	–	–

Bedarf deutlich übersteigt. Die überflüssigen Ressourcen sind selten von Nachteil, verursachen aber ab einer gewissen Netzwerkgröße höhere Installationskosten.

Das Verhältnis und die Höhe der finanziellen Risiken und direkten Kosten über die Betriebsdauer beeinflussen letztendlich, ob eine genaue Dimensionierung des Netzwerks erstrebenswert ist. Überwiegt deutlich das finanzielle Risiko, so ist eine Überdimensionierung sinnvoll. Sind hingegen die direkten Kosten größer, so ist unter Umständen auch eine Unterdimensionierung zu tolerieren¹⁵. Ein ausgeglichenes Verhältnis zwischen Risiken und Kosten legt eine genaue Dimensionierung nahe. Sie lohnt sich insbesondere bei großen Netzwerken, da sie einerseits eine kritische Netzwerkauslastung erreichen können und da andererseits signifikante Einsparungen bei den Installationskosten möglich sind. Diese Effekte verstärken sich bei einer geringen Netzwerkkapazität und hohen Investitionen pro Gerät. Eine lange Betriebsdauer ist für die Nachhaltigkeit der Dimensionierung entscheidend. Diese für eine genaue Dimensionierung förderlichen Eigenschaften werden in Tabelle II genutzt, um die in Tabelle I verglichenen Branchen zu bewerten. Die zusätzlich aufgeführten Eigenschaften werden in den nächsten Kapiteln ergänzt.

¹⁵ Ein Beispiel ist der Internetanschluss zu Hause. Auch wenn man regelmäßig das Internet nutzt, akzeptiert man bei einem Analogmodem die langsame Verbindung aufgrund des geringen Preises.

In der Bürokommunikation und der Heimautomation sind die finanziellen Risiken, Kosten und Betriebszeiten im Allgemeinen so klein, dass eine genaue Auslegung nicht lohnenswert ist und sowohl eine Unter- als auch eine Überdimensionierung toleriert werden kann. Bei WANs wiederum sind sowohl Betriebsdauer als auch die Kosten um einiges größer und die Netzwerke werden genau dimensioniert [26]. In der industriellen Automation und im Fahrzeugbau überwiegt das finanzielle Risiko. Deshalb wird sicherheitshalber überdimensioniert und es werden deterministische Netzwerke eingesetzt, die zwar meist teurer und weniger leistungsfähig als nicht-deterministische Netzwerke sind, deren Qualitätsverluste allerdings begrenzt und abschätzbar sind.

In direktem Zusammenhang mit dieser Dimensionierungsfrage steht nun die Qualitätsbewertung der Kommunikation. Sie ist erst dann lohnenswert, wenn sich keine Vermeidungsstrategien wie Überdimensionierung oder deterministische Netzwerke rentieren und wenn ein vorhandenes finanzielles Risiko oder reduzierte, direkte Kosten den Mehraufwand für eine Verbesserung der Qualität rechtfertigen.

Die Gebäudeautomation eignet sich besonders gut für eine genaue Dimensionierung und Qualitätsbewertung, wie die Tabelle I zeigt. So gibt es große Netzwerke, lange Betriebszeiten, einige Prozesse mit hohen Anforderungen an die Echtzeit und Betriebssicherheit (Tabelle I) sowie ausgeglichene Kostenverhältnisse. Deshalb stellt sich diese Arbeit die Aufgabe, eine solche Qualitätsbewertung für die Gebäudeautomation zu entwickeln. Dadurch wird die Qualität von Netzwerkentwürfen verbessert und infolgedessen können Kosten reduziert werden. Das notwendige Lösungskonzept wird im folgenden Kapitel erarbeitet. Hierfür werden die konkreten Anforderungen analysiert und vorhandene Lösungsansätze diskutiert.

KAPITEL

3.

LÖSUNGSKONZEPT

*Denn der wird nimmermehr das Ziel gewinnen,
der von Gedanken sieht sich überschwommen,
die kraftauflösend durcheinanderrinnen.*

Dante Alighieri

3.1. ANFORDERUNGEN AN EINE LÖSUNG

IM letzten Kapitel wurden die Notwendigkeit und die Rentabilität einer Qualitätsbewertung von Netzwerkentwürfen in der Gebäudeautomation erörtert, um die Kosten für Konfigurierung, Inbetriebnahme und Nachbesserungen zu reduzieren. Dies formt die *Zielstellung* der Arbeit:

ZIEL:

Zielstellung der Arbeit ist die Erhöhung der Qualität von Netzwerkentwürfen von Steuerungs- und Regelungsnetzwerken durch eine Entwurfsunterstützung zur Vermeidung, Identifikation und Behebung von Entwurfsfehlern mit Hilfe einer begleitenden, automatisierten Qualitäts- und Leistungsbewertung der Kommunikationsmedien.

An dieses Ziel stellen sich weitere Anforderungen, die sich aus der beschriebenen Entwurfssituation in der Gebäudeautomation ergeben. Entscheidend für eine Qualitätsbewertung des Netzwerkentwurfs ist, dass sie keinen zusätzlichen Aufwand erzeugt, da die Entwurfszeit bereits ausgereizt ist. Damit stellt sich:

ANFORDERUNG 1: EFFIZIENZ

Die Bewertung der Qualität von Netzwerkentwürfen und die Identifikation und Behebung von Entwurfsfehlern darf die Konfigurierung nicht wesentlich verlangsamen.

Dies bedeutet implizit, dass von dem Systemintegrator nicht erwartet werden kann, dass er sich umfangreiches Wissen aneignet. Es folgt:

ANFORDERUNG 2: VERSTÄNDLICHKEIT

Die Bewertung der Qualität von Netzwerkentwürfen und die Identifikation und Behebung von Entwurfsfehlern sollte vom Entwerfer kein umfangreiches Zusatzwissen erfordern.

Mit diesen Anforderungen bieten sich einfache, schnelle Lösungen an wie Daumenregeln, Extrapolation oder Überschlagsrechnungen, die auf einzelnen Erfahrungen basieren und sie verallgemeinern (z. B. [160], Empfehlungen in [51]). Solche starken Verallgemeinerungen können bei der Konfigurierung eine Orientierung bieten, erfassen aber nicht die hohe Spezifität von Automatisierungssystemen und ihr nichtlineares Verhalten bei Größenänderung. Damit erlauben sie keine Garantierbarkeit entsprechend Qualitätsmerkmal 2, aus dem die folgende Anforderung resultiert:

ANFORDERUNG 3: GENAUIGKEIT

Die resultierenden Qualitätsabschätzungen müssen in Bezug auf die Qualitätsmerkmale 2 bis 6 eine hinreichend genaue Aussage erlauben, um die wesentlichen Fehler korrekt zu identifizieren.

In den bisherigen Anforderungen liegt ein Widerspruch verborgen, der im nächsten Abschnitt deutlicher hervortritt. So sind mit Anforderung 3 komplexere Ansätze erforderlich, die wiederum berechnungsintensiver und zeitaufwendiger sind und damit Anforderung 1 und 2 widerstreben. Dieser Widerspruch wird durch drei weitere Anforderungen reduziert. Zum einen wird ein Ansatz verfolgt, der es erlaubt, zwischen den Anforderungen zu skalieren, so dass die verwendeten Methoden und Modelle an die vorhandenen Parameter und Anforderungen angepasst werden können.

ANFORDERUNG 4: DIFFERENZIERTHEIT

Die Methoden und Modelle sollten in ihrem Detaillierungsgrad an die vorhandenen Systeminformationen und die Anforderungen an die Ergebnisse anpassbar sein.

Zum anderen sollte bei der Größe und Komplexität der Netzwerke der Gebäudeautomation die Lösung weitestgehend ohne Aufwand für den Anwender anwendbar sein, so dass er sich vielmehr auf die Ergebnisauswertung und seine Handlungen konzentrieren kann.

ANFORDERUNG 5: AUTOMATISIERUNG

Die Lösung muss weitestgehend automatisiert sein, so dass sie nur geringe Eingriffe des Anwenders benötigt.

Zusätzlich sollte sich die Lösung nahtlos in den üblichen Entwurfsprozess eingliedern, um einen reibungslosen Einsatz zu erlauben.

ANFORDERUNG 6: INTEGRATION

Der Lösung muss sich in den vorhandenen Entwurfsprozess nahtlos integrieren.

Diese Anforderungen werden durch COOLING [27] bestätigt. Er hat die Probleme bei der Softwareentwicklung von Echtzeitsystemen untersucht und kommt zu dem Schluss, dass Entwerfer häufig zu optimistisch sind und glauben, Leistungsprobleme schnell lösen zu können — sofern sie »entdeckt« werden — und damit ein häufig verschwiegenes Geschäftsrisiko eingehen¹. Eine Lösung stellt für ihn die Leistungsbewertung der Entwürfe dar, dennoch wird sie nur selten verwendet. Als Gründe führt er die Einarbeitungszeit, Bearbeitungszeit, entsprechende Kosten, Zweifel an der Genauigkeit und die im Nachhinein in Frage gestellte Notwendigkeit auf. Genau diese Gründe werden durch die obigen Anforderungen widerlegt. Inwiefern sich die Leistungsbewertung als Lösung eignet, untersucht der folgende Abschnitt.

3.2. LÖSUNGSANSÄTZE IN DER LITERATUR

3.2.1. LEISTUNGSBEWERTUNG

Um die Qualität der Kommunikation zu bewerten, eignen sich besonders Techniken der *Leistungsbewertung* (en. *Performance Evaluation*) von Kommunikationssystemen. Darunter versteht man die Untersuchung von Systemen mit dem Ziel der quantitativen Bestimmung und Bewertung ihrer *Leistungskenngrößen* [9, 232]. Die Anwendungsbereiche liegen nach [9, 112, 134] in der Vorhersage der Leistungsfähigkeit von Systementwürfen, in der Bewertung und dem Vergleich realisierter Systeme und der Ermittlung von Engpässen im Betrieb. Die gestellte Aufgabe ist im ersten Bereich einzuordnen.

Historisch etablierte sich die Leistungsbewertung von Kommunikationssystemen mit den ersten Telekommunikations- und Datenkommunikationsnetzwerken [120] und ist seitdem weit verbreitet. Umfassende Werke wie [112, 120, 214] geben einen guten Überblick über die Methoden. Sie unterteilen sich in Messverfahren, simulative und analytische Verfahren mit ansteigendem Abstraktionsgrad des Modells. Abbildung 3.1 aus [214] verdeutlicht dies noch einmal. Voraussetzung für *Messungen* ist das reale System. Simulative Ansätze dehnen sich nahezu über alle Abstraktionsstufen aus: beginnend bei *HIL-Simulation* (HIL²) am realen System, über die *detaillierte Simulation*, die ein detailgetreues Systemmodell nutzt,

1 »Designers are (blindly) optimistic that performance problems — if they arise — can be easily overcome. [...] The problems [...] is that performance problems are not predicted, they are *discovered*. If all goes well, fine. But if things do go wrong, however, the business consequences may be serious. [...] It is also our experience that many performance problems have been quietly 'swept under the carpet', especially in the IT world.«, [27, S. 739]

2 en. Hardware-in-the-Loop

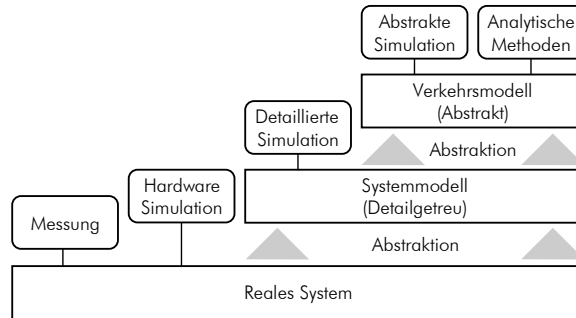


ABBILDUNG 3.1. Methoden zur Leistungsbewertung (nach [214]).

bis hin zur *abstrakten Simulation*, die Modelle mit einem hohen Abstraktionsgrad nutzt, die auch das Verkehrsverhalten der Geräte modellieren. Diese abstrakten Modelle bilden auch die Basis für die *analytischen Methoden*, die in exakte und approximative Methoden unterteilt werden, je nachdem ob sie für das Modell eine mathematisch korrekte Lösung oder nur eine Näherung bieten.

Von diesen Methodenklassen eignen sich für die Bewertung von Netzwerkentwürfen nur simulative- und analytische Methoden, da davon auszugehen ist, dass während der Konfigurierung noch kein reales System vorhanden ist. Diesbezüglich gilt:

PRÄMISSE 2: KEIN REALES NETZWERK VORHANDEN

Bei der Konfigurierung ist kein reales Netzwerk vorhanden oder aus anderen Gründen ist eine Messung zur direkten Leistungsbewertung nicht sinnvoll.

Grundsätzlich besitzen Simulationen den Vorteil, dass durch den geringen Abstraktionslevel auch komplexes Verhalten vergleichsweise einfach zu modellieren ist. Dadurch liefern sie für einzelne stichprobenhafte Experimente schnell genaue Ergebnisse. Um allerdings statistisch relevante Aussagen über das meist komplexe Systemverhalten zu gewinnen, erfordern sie umfangreiche Experimente, die insgesamt sehr berechnungs- und zeitaufwendig sind. Mit zunehmendem Abstraktionslevel nimmt zwar der Ausführungsaufwand ab, aber auch die Ergebnisgenauigkeit, so dass analytische Modelle im Vergleich zur detaillierten Simulation zwar sehr schnell zu lösen, aber ungenauer sind. Entscheidend für die Genauigkeit ist, ob die Randbedingungen der Abstraktionen erfüllt sind. Dementsprechend werden detaillierte Modellierungskennnisse benötigt. Entsprechend empfehlen sich simulative Ansätze zur detaillierten Untersuchung kleiner bis mittlerer Beispiele, wohingegen sich analytische Ansätze bei grundsätzlichen Untersuchungen großer Systeme anbieten. Welche Methode sich allerdings für eine Qualitätsbewertung von Netzwerkentwürfen besonders eignet, ist aufgrund der unterschiedlichen An-

forderungen 1 bis 3 schwer festzulegen.

Besonders im WAN-Bereich haben sich Methoden der Leistungsbewertung im Entwurf etabliert. Einen umfassenden Überblick über die gängigen Tools zum Entwurf und zur Leistungsbewertung der Netzwerke geben GARCIA & HACKBARTH [71]. Wie in Abschnitt 2.1 schon diskutiert wurde, lohnt sich bei WANs eine gute Dimensionierung und rechtfertigt den zusätzlichen Modellierungsaufwand zur Leistungsbewertung. Aufgrund des Stellenwerts, den eine gute Dimensionierung der Netzwerke spielt, spricht man in diesem Umfeld auch vom *Network Performance Engineering* [26] (Abschnitt 3.3). Den Bezug zu den Netzwerken der Automation schaffen LUNTOVSKYY et al. [150], die eine integrierte Entwurfsumgebung für Computernetzwerke der Bürokommunikation bis hin zur Gebäudeautomation skizzieren.

Im Bereich der industriellen Netzwerke gibt es mit [104, 105, 139, 140, 213, 218] einige umfassende Beispiele, die sich den besonderen Gegebenheiten der Automatisierungstechnik zuwenden, die hauptsächlich in den hohen Echtzeitanforderungen liegen. Bei der Analyse von Netzwerken der Gebäudeautomation kommen vor allem simulative Ansätze zur Anwendung, von direkten Protokollimplementationen in [44, 89, 159, 190, 202, 226] bis hin zu abstrakteren Zustandsmaschinenmodellen [19, 81, 245]. Analytische Methoden werden in [15, 16, 157, 176] verwendet.

Diese Auswahl an Arbeiten verdeutlicht, dass unterschiedliche Methoden zur Leistungsbewertung von Kommunikationsnetzwerken vorhanden und etabliert sind. Allerdings erfordern die Methoden entsprechende Modelle von dem zu untersuchenden Netzwerk. Diese wurden bei einem Großteil der Arbeiten von Hand erstellt und sind an die einzelnen Untersuchungsbeispiele angepasst. Bei den teilweise großen Netzwerken in der Gebäudeautomation und den geringen Entwurfsbudgets ist einem Systemintegrator dieser Zusatzaufwand im Entwurfsprozess nach den Anforderungen 1 und 2 nicht zuzumuten und macht ein Konzept zur automatischen Modellierung nach Anforderung 5 erforderlich. Nur dadurch können die vorhandenen Methoden der Leistungsbewertung für eine Qualitätsanalyse im Entwurfsprozess effizient und sinnvoll genutzt werden.

3.2.2. AUTOMATISCHE MODELLIERUNG

Zur Leistungsbewertung müssen zwei Aspekte über das Netzwerk bekannt sein: Dies ist zum einen der strukturelle Aufbau des Netzwerks, welcher im *Systemmodell* erfasst wird, und zum anderen das Kommunikationsbedürfnis der Geräte, also wie oft Nachrichten gesendet werden, welches im *Verkehrsmodell* modelliert wird. Beide Modelle zusammen bilden das *Leistungsmodell*, das speziell auf eine Methode der Leistungsbewertung angepasst ist.

Vorhandene Ansätze zur automatischen Modellierung von Leistungsmodellen im Feldbusbereich dienen vor allem der Modularisierung und Wiederverwendung

von Modellteilen. HINTZE & KUCERA [85] entwarfen eine Methode, bei der die Modelle durch den Anwender in formalen Sprachen beschrieben und danach in ein Simulationsmodell transferiert werden. Vergleichbar ist der Ansatz mit CASTELPIETRA et al. [20], bei denen das System- und Verkehrsmodell in einer speziellen Deklarationssprache spezifiziert und aus modularen Simulationskomponenten zusammengesetzt wird. Ähnlich dazu verwenden TOMURA et al. [213] komposite Entwurfsmuster zum Aufbau der simulierten Zustandsautomaten. Im Bereich der Computernetzwerke gibt es von FAHMY & DOULIGERIS [60] und FAHMY et al. [62] bereits sehr frühe Arbeiten zur automatischen Modellierung von Netzwerken und Simulationsmodellen, die mit der Nutzung eines lernfähigen Expertensystems weit in den Entwurf hineinreichen. Obwohl diese Ansätze bereits die Generierung des Simulationsmodells durch wiederverwendbare Module teilweise automatisieren, muss der Anwender dennoch in der einen oder anderen Weise das Modell zuvor spezifizieren.

In dem verwandten *Software Performance Engineering* finden sich weitergehende Ansätze. WOODSIDE et al. [243] schlugen die Integration der Leistungsbewertung von Programmen direkt in die Entwicklungssoftware vor, wobei der Programmcode automatisch in Leistungsmodelle überführt und bewertet wird. Die Schwierigkeit der Umsetzung eines ähnlichen Ansatzes bei der Leistungsbewertung von Netzwerken liegt in den fehlenden elektronisch lesbaren Informationsquellen zur Modellierung. Insbesondere bei Netzwerken der Bürokommunikation und zum Teil auch bei WANs besteht nicht die Notwendigkeit zur Modellierung des Netzwerks außer zum Zwecke der Leistungsbewertung, so dass sich dieser Zusatzaufwand bei der Dimensionierung und Optimierung auszahlen muss. Im Gegensatz dazu entsteht bei der Komposition der Anwendung in Netzwerken der Automation notwendigerweise ein detailliertes Modell, das sich als Basis für die automatische Modellierung anbietet³. Zusätzlich haben sich in der Gebäudeautomation in den letzten Jahren »offene⁴« *Entwurfsdatenbanken* herausgebildet, die von verschiedenen *Entwurfstools* (Abschnitt 2.2) gemeinsam genutzt werden. Dadurch ist das notwendige Entwurfswissen ohne eine Bindung an ein proprietäres Entwurfstool zugänglich. Als wissenschaftliche Frage ist zu klären, wie eine automatische Modellierung eines Leistungsmodells auf Basis einer Entwurfsdatenbasis erfolgen kann? Dabei ist im Detail die Struktur eines passenden Modells zu diskutieren, der Umfang der Informationsquellen zu untersuchen und gegebenenfalls neue Quellen zu erschließen. Dabei wird sich in Kapitel 5 zeigen, dass wichtige Informationen für das Leistungsmodell nicht aus der Entwurfsdatenbank gewonnen werden können, weshalb das entwickelte Verfahren als automatische Modellierung und nicht als Modelltransformation klassifiziert wird. Diese auto-

3 Gleiches gilt für alle komponentenbasierten verteilten Systeme.

4 Die Struktur der Datenbasis ist bekannt und die Inhalte sind über Schnittstellen zugänglich. Dies erfordert keine unentgeltlichen, freien Datenbasen.

matische Modellierung des Leistungsmodells aus einer Entwurfsdatenbank bildet den ersten Schwerpunkt dieser Arbeit.

Unter anderem fehlen Informationen zum Kommunikationsbedürfnis der Geräte, da sie zur Konfigurierung und dem Betrieb des Netzwerks nicht unmittelbar von Bedeutung sind. Aus diesem Grund muss das Verkehrsmodell anders erstellt werden, wobei die Gebäudeautomation dabei wieder Vorteile bietet. In Computernetzwerken wird die Kommunikation durch den willkürlich handelnden Menschen bestimmt und ist ständig wechselnden Kommunikationsbeziehungen unterworfen, wodurch ein äußerst schlecht zu modellierender Verkehr entsteht (Abschnitt 6.1). In der Gebäudeautomation bestimmen hingegen regelmäßige Regelungs- und Steuerungsprozesse den Verkehr, deren Kommunikationsbeziehungen bereits in der Komposition festgelegt werden und sich in einem Netzwerkentwurf mehrfach wiederholen. Diese Gegebenheiten erlauben ein angepasstes und detailliertes Verkehrsmodell, welches den zweiten wesentlichen und neuartigen Beitrag dieser Arbeit in Kapitel 6 darstellt.

3.2.3. AUTOMATISCHE QUALITÄTSBEWERTUNG UND FEHLERANALYSE

Die automatische Modellierung erlaubt nun den Einsatz der Leistungsbewertung, um die Qualität von Netzwerkentwürfen in der Gebäudeautomation innerhalb der aufgestellten Anforderungen zu beurteilen. Die Leistungsbewertung bestimmt die *Leistungskenngrößen* der einzelnen Netzteilnehmer, die in Abschnitt 7.2 vorgestellt werden. Diese Leistungskenngrößen stellen Qualitätsmerkmale dar. Sie müssen für eine Qualitätsaussage mit ihren Anforderungen verglichen werden. Die Spezifikation von Qualitätsanforderungen und die Bewertung derselben erlauben die Methoden der *Dienstqualität* (QoS⁵) [28, 206, 208, 212]. Sie definieren die Anforderungen an die Qualität einer einzelnen Verbindung und unterscheiden sich damit von dem bisher genutzten globalen Begriff der Qualität eines ganzen Netzwerkentwurfs, welcher dazu in Zusammenhang gestellt werden muss. Die Formalisierung der Definition von QoS-Anforderungen im Entwurf und die Verteilung im Netzwerk sind mit den Arbeiten von FROLUND & KOISTINEN [67] und CAMPBELL et al. [18] bereits lange in der Diskussion. Dennoch fehlen Möglichkeiten, wie die verbindungsorientierte Definition der Dienstqualität effizient nach Anforderung 1 für ein großes Netzwerk mit festen Kommunikationsbeziehungen entworfen werden kann. Hierfür muss eine automatische Lösung nach Anforderung 5 zur Spezifikation und Bewertung der Qualität eines Netzwerkentwurfs gefunden werden.

Die Qualitätsbewertung deckt Fehler als nicht erfüllte Qualitätsanforderungen auf. Die Ursachen der Fehler können umfangreich anhand der Leistungskenn-

5 QoS: en. Quality of Service

größen untersucht werden. Nach den Anforderungen 1 bis 6 muss der Anwender konsequenterweise bei der Fehlerbehebung beraten werden. Ähnliche Aufgaben werden im Betrieb durch die *Netzwerkdiagnose* im Rahmen des *Netzwerkmanagements* erfüllt. STEINDER & SETHI [207] geben einen Überblick über die Methoden, wobei der Schwerpunkt und die Schwierigkeit im Modellieren der im Betrieb dynamisch wechselnden Netzwerkeigenschaften liegen. KESHAV & SHARMA [118] denken als Wissensbasis im Entwurf bereits an, die Ergebnisse einer Leistungsbewertung zu nutzen. Dies empfiehlt sich insbesondere bei den statischen Entwürfen in der Automation. Damit leiten sich neue und tiefere Ansätze zur Diagnose ab, die bis zur Beratung bei der Problemlösung reichen. Sie bilden mit Kapitel 8 den dritten und abschließenden Schwerpunkt der Arbeit.

3.3. CONTROL NETWORK PERFORMANCE ENGINEERING

Aus der Analyse der Literatur folgt, dass die Methoden zur Leistungsbewertung grundsätzlich vorhanden sind und eine Qualitätsbewertung von Netzwerkentwürfen erlauben. Allerdings fehlen Verfahren zur automatischen Modellierung des Systems und des Verkehrsmodells sowie zur Auswertung und Fehleranalyse. Erst damit können die Methoden nahtlos in den Entwurfsprozess integriert werden, so dass ein Systemintegrator, der kein Methodenexperte der Leistungsbewertung ist, dennoch von einer Qualitätsverbesserung profitieren kann. Dadurch entsteht ein neuer, erweiterter Entwurfsprozess, der in diesem Abschnitt bis zur Abbildung 3.2 aufgestellt wird.

Ausgangsbasis ist der Systemintegrator, der einen Netzwerkentwurf in einem Entwurfstool bearbeitet. Hieraus resultiert eine Entwurfsdatenbank oder eine andere elektronisch lesbare Datenquelle, so dass gilt:

PRÄMISSE 3: ELEKTRONISCH LESBARER NETZWERKENTWURF

Es gibt eine elektronisch lesbare Repräsentation des Netzwerkentwurfs, die das zu untersuchende Netzwerk in seinen wesentlichen strukturellen und parametrischen Eigenschaften widerspiegelt.

Diese Prämisse ist im Umfeld der Gebäudeautomation erfüllt und die automatische Modellierung kann als erstes *Teilziel* formuliert werden:

TEILZIEL 1: MODELLIERUNG DES LEISTUNGSMODELLS

Es wird ein Konzept zur automatischen Modellierung von Leistungsmodellen aus Entwurfsdatenbasen für ausgewählte Netzwerke der Gebäudeautomation entwickelt.

Dieser Aufgabe widmen sich die Kapitel 5 und 6.

Ist das Modell erstellt, so kann die Leistungsbewertung erfolgen. Dabei soll in dieser Arbeit aufgrund der unterschiedlichen Anforderungen keine Einschränkung

auf eine Methode vorgenommen, sondern sowohl simulative als auch analytische Verfahren gemäß Anforderung 4 unterstützt werden (Kapitel 7).

Am Ende der Leistungsbewertung müssen die bestimmten *Leistungskenngrößen* auf *Qualitätskenngrößen* abgebildet werden; die Qualität bezüglich der gestellten Anforderungen bewertet und Fehler identifiziert werden. Es folgen die Ziele:

TEILZIEL 2: BEWERTUNG DER QUALITÄT DES KOMMUNIKATIONSMEDIUMS

Die Qualität des Kommunikationsmediums wird anhand der in den Qualitätsmerkmalen 2 bis 5 festgelegten Kriterien bewertet.

TEILZIEL 3: IDENTIFIKATION VON ENTWURFSFEHLERN

Werden Anforderungen an die Qualität des Kommunikationsmediums nicht erfüllt, so werden diese als Entwurfsfehler identifiziert.

Aufgrund der Anzahl an Ergebnissen und der Komplexität der Zusammenhänge muss der Anwender konsequenterweise auch bei der Fehlerbehebung beraten werden.

TEILZIEL 4: BERATUNG ZUR FEHLERBEHEBUNG

Werden Entwurfsfehler identifiziert, so werden die Ursachen analysiert und der Anwender wird zum Zwecke der Fehlerbehebung beraten.

Die Entscheidung für eine Fehlerbehebung sollte letztendlich dem Entwerfer obliegen. In einem weiteren Entwurfsschritt passt er den Netzwerkentwurf entsprechend seiner Entscheidungen an und eröffnet damit die Möglichkeit für eine wiederholte Analyse.

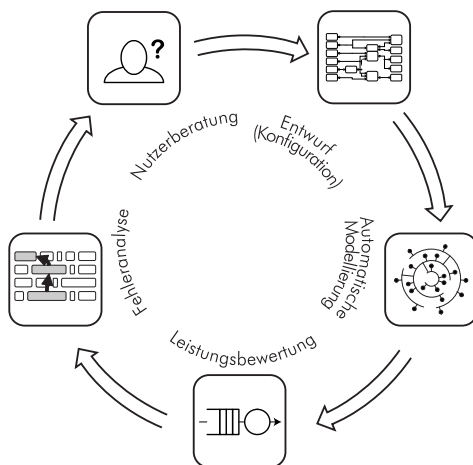


ABBILDUNG 3.2. Integration des Control Network Performance Engineering in die Konfiguration.

Damit resultiert aus den Teilzielen 1 bis 4 ein geschlossener Prozess zur Identifikation und Behebung von Entwurfsfehlern in einem Netzwerkentwurf, der in Abbildung 3.2 dargestellt ist. Dieser Prozess erfüllt die Zielstellung unter den gestellten Anforderungen 1 bis 6. Hierfür reicht es an sich aus, am Ende der Konfigurierung eine Qualitätsbewertung des Netzwerkentwurfs durchzuführen und Fehler gegebenenfalls zu beheben.

Um Fehler nicht nur zu beheben, sondern von vornherein zu vermeiden, muss der Entwerfer ständig über die Qualität seines aktuellen Netzwerkentwurfs informiert sein. Nur so kann er sich anbahnende Fehler früh erkennen und vermeiden. Wird jeder seiner Entwurfsschritte von einer Qualitätsbewertung begleitet, so entwickelt er ein Gefühl für die Wirkung seiner Entwurfshandlungen und kann das Netzwerk unter Qualitätsaspekten gezielt dimensionieren.

Damit ermöglicht die entwurfsbegleitende Leistungsbewertung, Qualitätsbewertung und Fehleranalyse (Abbildung 3.2) einen leistungs- und qualitätsorientierten Entwurf für Automationsnetzwerke, der als *Control Network Performance Engineering* (CNPE) bezeichnet werden soll⁶. Der zweite Teil dieser Arbeit widmet sich diesem Control Network Performance Engineering, doch zunächst werden im nächsten Kapitel einige Grundlagen zu den untersuchten Netzwerken der Gebäudeautomation vorgestellt.

6 Eine Übersetzung ins Deutsche ist schwierig. »Control Networks« sind übersetzt Regelungs- und Steuerungsnetzwerke, die häufiger als »Feldbussysteme« oder allgemeiner als »Netzwerke der Automation« bezeichnet werden. Jedoch werden als »Control Networks« im Englischen insbesondere Netzwerke der Gebäudeautomation verstanden, so dass sowohl der Fokus als auch das Umfeld der Arbeit treffend bezeichnet ist. Ähnlich unzureichend und generell unüblich ist die Übersetzung von »(Network) Performance Engineering«. Es bezeichnet primär das Paradigma, bei dem die Leistung des späteren Systems den Entwurf leitet und mit regelmäßigen Leistungsbewertungen überwacht wird. Deshalb könnte eine Übersetzung »leistungsorientierter Netzwerkentwurf« lauten. Allerdings impliziert das Paradigma, dass Vorstellungen von der gewünschten Systemleistung bestehen. Im Kontext der Arbeit sind dies Qualitätsanforderungen und Leistung ist ein Qualitätsmerkmal. Deshalb ist »qualitätsorientierter Netzwerkentwurf« eine treffendere Übersetzung. Letztendlich führt eine Übersetzung nur zu einer individuellen Umschreibung der im Englischen etablierten und äußerst zutreffenden Fachbegriffe. Deshalb ist der Titel der Arbeit in Englisch.

KAPITEL

4.

NETZWERKPROTOKOLLE DER GEBÄUDEAUTOMATION

*Alles Gescheite ist schon gedacht worden, man
muss nur versuchen, es noch einmal zu denken.*

Johann Wolfgang von Goethe

4.1. MODELLE ZUR EINORDNUNG VON NETZWERKPROTOKOLLEN

BEVOR im Abschnitt 4.2 detaillierter auf konkrete Netzwerkprotokolle eingegangen wird, werden vorher zwei Modelle vorgestellt, die grundlegende Eigenschaften von Netzwerkprotokollen der Automation beschreiben und eine Einordnung vereinfachen. Ein *Netzwerkprotokoll* ist eine exakte Vereinbarung aus Regeln und Formaten, nach der Daten zwischen zwei und mehreren Geräten und Prozessen in einem Netzwerk ausgetauscht werden, so dass alle Beteiligten einander störungsfrei verstehen.

Das wichtigste Modell im Umgang mit Netzwerkprotokollen ist das *OSI-Referenzmodell*, das die zur Kommunikation offener Systeme (OSI¹) wesentlichen Bestandteile in sieben aufeinander aufbauenden Schichten aufteilt, die in Abbildung 4.2 aufgeführt werden. Eine gute Einführung in das Modell bieten [102, 131, 196, 210, 211].

Neben diesem OSI-Referenzmodell wird in der Automation häufig eine Automationspyramide [135, 218] verwendet, um die unterschiedlichen Funktions-

¹ OSI: en. Open System Interconnection

4. NETZWERKPROTOKOLLE DER GEBÄUDEAUTOMATION

und Ausrüstungsebenen einer Anlage einzuteilen. In der ISO 16484-2 [97] wurde daraus das *Strukturmodell der Gebäudeautomation* spezialisiert, welches das Netzwerk entsprechend der Abbildung 4.1 in ein Management-, Automations- und Feldgerätenetzwerk unterteilt [64, 77, 115, 129]. In der Managementebene liegen die Leitzentrale und das Facility Management. Sie realisiert die zentrale Überwachung und Steuerung eines Gebäudekomplexes und trägt hierfür alle relevanten Informationen zusammen. In der Automationsebene sind die lokalen Steuerungen eines Gebäudes oder einer Etage einzuordnen und in der Feldebene werden die konkreten Sensoren, Aktoren und dezentralen Steuerungen realisiert. Die Gerätelokalisierung entspricht meist auch dieser Funktionshierarchie.

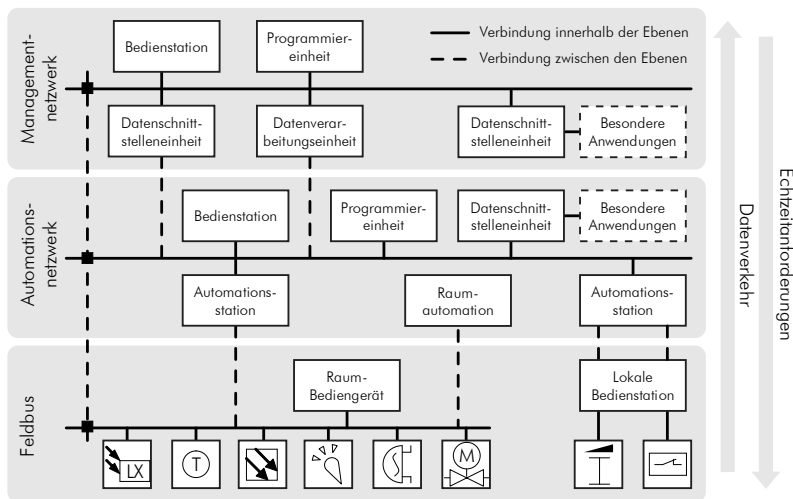


ABBILDUNG 4.1. Strukturmodell der Gebäudeautomation (nach [97]; Symbolerklärung in Abbildung 5.3).

Die Ebenen stellen unterschiedliche Anforderungen an die Kommunikation. So sind die Echtzeitbedingungen auf Feldebene wichtiger und der Datenverkehr geringer. Auf der Managementebene liegt durch die Datenbündelung ein erhöhter Datenverkehr vor, der allerdings durch die Dezentralisierung der Steuerung geringere Echtzeitanforderungen besitzt. Durch die unterschiedlichen Anforderungen haben sich in den Ebenen spezialisierte Netzwerke entwickelt. Sie werden entsprechend ihrer Eignung für spezielle Ebenen als *Feldbus*, *Automatisierungs-* oder *Managementnetzwerk* bezeichnet. Der wachsende Informationsbedarf der Feldgeräte und eine Konsolidierung des Marktes führten in den letzten Jahren dazu, dass immer mehr Kommunikationssysteme die Anforderungen aller Ebene erfüllen und sowohl Feldbussysteme in der Managementebene als auch Managementprotokolle im Feldbereich eingesetzt werden [64, 115].

4.2. NETZWERKPROTOKOLLE DER GEBÄUDEAUTOMATION

Es gibt zahlreiche Arbeiten, die Netzwerkprotokolle der Gebäudeautomation klassifizieren und vergleichen [14, 40, 64, 131, 149, 185, 195, 196, 199, 218, 235]. Von den unterschiedlichen Protokollen sind derzeit BACnet, Konnex und LON am meisten verbreitet. Speziell zu diesen Protokollen geben KASTNER et al. einen detaillierten und aktuellen Überblick in [115]. Die drei Protokolle werden im Folgenden kurz beschrieben und dann in den für die Modellierung relevanten Punkten in Kapitel 5 direkt verglichen.

Das Building Automation and Control Network (BACnet) wird seit 1987 von der American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE) entwickelt, 1995 erstmalig standardisiert und 2003 von der ISO als internationaler Standard etabliert [99]. BACnet ist ein Protokoll der Management- und Automationssebene und besitzt ein geschrumpftes 4-Schichten-Modell. Zu der Anwendungsschicht und einer Netzwerkschicht kann der Anwender für die unteren zwei Schichten entsprechend seinen Anforderungen flexibel zwischen ARCNET (ANSI 878.1), Ethernet (ISO 8803-3), BACnet/IP, PTP (Point-To-Point) via EIA RS-232, MS/TP (Master-Slave/Token-Passing) via EIA RS-485 oder LonTalk (das LON-Basis-Protokoll) wählen. Weitere Informationen zu BACnet können [130] entnommen werden.

Der *Europäische Installations-Bus* (EIB²) ist eine Entwicklung deutscher Firmen unter der Führung von Siemens von 1991. Aus EIB entstand nach einem Zusammenschluss mit BatiBus und EHS³ der *Konnex-Standard* (KNX), der 2003 zum europäischen Standard erhoben wurde [55]. Damit vereint Konnex, das auch als EIB/KNX bezeichnet wird, die wichtigsten Feldbussysteme der Gebäudeautomation im europäischen Sektor neben LON. Konnex implementiert vom OSI-Referenzmodell alle Schichten bis auf die Darstellungs- und Sitzungsschicht [236]. Es nutzt einen CSMA/CA⁴ Zugriffsverfahren[39, 131], bei dem sich Telegramme mit dominanten logischen Pegeln durchsetzen.

Das Local Operating Network (LON) *Local Operating Network* bezeichnet synonym die *LonWorks*-Technologie und wurde 1989 durch die amerikanische Firma Echelon entwickelt. Es basiert auf dem *LonTalk*-Protokoll, das in der EIA/CEA-709.1-B-2002 [51] und der EN 14908 [54] standardisiert wurde und als Teil von BACnet auch ISO-Standard ist. Es nutzt ein spezielles predictive p-persistent CSMA/CD⁵ Zugriffsverfahren[131], bei dem die Anzahl von Arbitrierungsfenstern an den zu erwartenden Verkehr angepasst wird⁶. Jedes sendewillige Gerät

2 EIB: en. European Installation Bus

3 EHS: en. European Home System

4 CSMA/CA: en. Carrier Sense Media Access / Collision Avoidance

5 CSMA/CD: en. Carrier Sense Media Access / Collision Detection

6 Weshalb das Protokoll zum Teil auch als CSMA/CA bezeichnet wird.

4. NETZWERKPROTOKOLLE DER GEBÄUDEAUTOMATION

wählt aus den Arbitrierungsfenstern zufällig eines aus, wobei das Gerät mit dem niedrigsten Fenster senden darf [40, 108]. Seit 2005 gibt es in Europa die Initiative »SafetyLON«, die ein auf LON aufsetzendes Safety-Protokoll für den Einsatz in Systemen mit sehr hohen Anforderungen an die Betriebssicherheit nach der IEC 61508 [90] entwickeln will, wie sie für Entrauchungsanlagen oder Fluchttürsteuerungen benötigt werden.

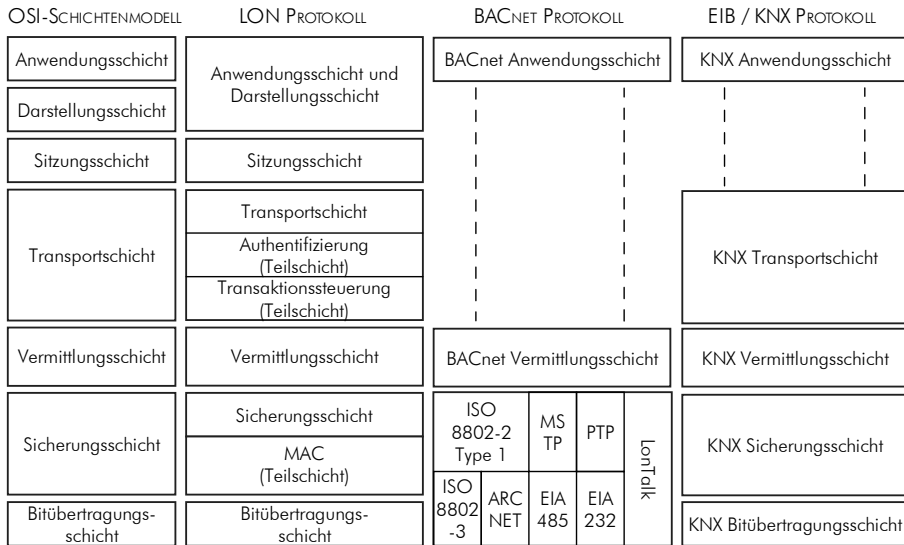


ABBILDUNG 4.2. Vergleich verschiedene Protokolle der Gebäudeautomation mit dem OSI-Referenzmodell.

Alle drei Protokolle unterstützen verschiedene physische *Übertragungsmedien* wie Twisted-Pair, Koaxial, Power-Line, Funk, Infrarot oder Glasfaser und sind offene Standards. Dies ist in der Gebäudeautomation eine wichtige Eigenschaft, um herstellerunabhängige Systeme aufzubauen (siehe A6 in Abschnitt 2.3). Beachtenswert ist, dass alle Protokolle vorzugsweise auf nicht-deterministische Carrier Sense Media Access (CSMA) Zugriffsmechanismen setzen. BACnet als einziger Standard unterstützt durch seine variablen unteren zwei Schichten auch deterministische Protokolle wie MS/TP oder ARCNET. Dies verdeutlicht noch einmal die weniger kritischen Echtzeitanforderungen in der Gebäudeautomation im Vergleich zu anderen industriellen Automatisierungsgebieten (Tabelle I u. II). Die Abbildung 4.2 vergleicht die realisierten Schichten der einzelnen Protokolle mit dem OSI-Referenzmodell.

Neben diesen etablierten Protokollen entwickeln sich derzeit zwei weitere interessante Konkurrenten. Zum einen drängt Ethernet und TCP/IP weiter in den Feldbereich vor. Durch die Ausbreitung des Internets ist Ethernet in vielen Zweckbauten installiert und aufgrund der hohen Stückzahlen entsprechend preiswert. Dadurch bietet sich eine Nutzung für die Gebäudeautomation insbesondere in der Managementebene an. Dementsprechend ist es bereits das von BACnet am meisten verwendete Protokoll [130] und drängt von dort weiter in den Feldbereich vor. Eine weitere interessante Entwicklung liegt in den drahtlosen Sensornetzwerken [163, 165, 167], die besonders für den Heimbereich durch ihre einfache, kabelfreie Installation geeignet sind.

Die in dieser Arbeit vorgestellten Prinzipien lassen sich auf alle vorgestellten, drahtgebundenen Netzwerkprotokolle mit begrenztem Transformationsaufwand anwenden. Dies wird in einzelnen Beispielen gezeigt. Als Demonstrationsbeispiel wurde aus den vorgestellten Protokollen das LON-Protokoll ausgewählt. Zum einen ist es als Feldgerätenetzwerk stärker mit der Konfigurierung der Geräte verbunden als BACnet und zum anderen besser mit Entwurfstools ausgestattet (vgl. Prämisse 3). Im Vergleich zu Konnex wird LON häufiger im Zweckbau eingesetzt, wohingegen der Schwerpunkt von Konnex in der Heimautomation liegt [117, 132]. Dadurch sind die entworfenen Netzwerke in LON üblicherweise größer als bei Konnex und eignen sich gemäß Unterabschnitt 2.4.2 besser für eine Qualitätsbewertung.

TEIL
II.

CONTROL NETWORK
PERFORMANCE ENGINEERING

KAPITEL

5.

AUTOMATISCHE MODELLIERUNG DES SYSTEMMODELLS

*Modelle sollten sich bemühen,
dem Portrait ähnlich zu sehen.*

Salvador Dalí

5.1. GRUNDLEGENDE ASPEKTE DER AUTOMATISCHEN MODELLIERUNG

ENTSCHEIDENDEN Einfluss auf die Modellierung hat die spätere Anwendung des Modells. Entsprechend des gewählten Vorgehens zur Leistungsbewertung ist das Modell nahe am physischen System oder abstrakter und auf die genutzte analytische Methode abgestimmt (Abbildung 3.1). Zwischen diesen beiden Modellierungstiefen kann es weitere Zwischenmodelle geben.

Das in dieser Arbeit vorgeschlagene Modell ist ein Intermediärmodell, das sich noch am physischen System orientiert, aber einen einfachen und schnellen Übergang in ein detailliertes Simulationsmodell oder abstraktes analytisches Modell erlaubt. Dies hat zwei Vorteile: Einerseits kann der Anwender sich im Modell genauso wie im Netzwerkentwurf orientieren (Anforderung 2), andererseits wird keine Einschränkung auf eine Methode der Leistungsbewertung gemacht, sondern sie kann nach Anforderung 4 an die Bedürfnisse der Anforderungen 1 bis 3 angepasst werden. Das Modell und die Implementierung wurden in Zusammenarbeit mit MARIO NEUGEBAUER, ALEXANDER KARAVAN und DANIEL WENDT [234] im BMBF-Projekt »NetPlan« entwickelt. Erste Konzepte entstammen bereits der Diplomarbeit von THOMAS FLORSTEDT [65]. Einzelne Aspekte des Modells sind in [108, 110, 166, 176, 178, 179, 234] veröffentlicht worden.

Das Modell ist in der Unified Modelling Language (UML) [171] beschrieben, die sich im Umgang mit Leistungsmodellen bewährt hat [4, 194, 246]. RUMBAUGH et al. [191] und SEEMANN & VON GUDENBERG [203] geben gute Einführungen in die Terminologie und Verwendung der Sprache. Die Bezeichner der modellierten Objekte sind in englischer Sprache, um konsistent zu der vorhandenen Softwareimplementierung zu sein. Zusätzlich werden zu allen Elementen, die später in der Analyse wiederverwendet werden, die mathematischen Mengensymbole eingeführt. Alle Mengen sind in Fraktur gesetzt, so beschreibt beispielsweise \mathcal{D} die Menge aller Geräte in einem Netzwerk, $d \in \mathcal{D}$ ein beliebiges Element hieraus und $|\mathcal{D}|$ die Mächtigkeit der Menge \mathcal{D} , also die Anzahl der Geräte im Netzwerk.

Um die unterschiedlichen Modellaspekte getrennt zu behandeln, besteht das Modell aus einem *Systemmodell*, das das Netzwerk in seiner Struktur widerspiegelt, und einem *Verkehrsmodell*, das die Nachrichtenerzeugung modelliert (Kapitel 6). Für das Modell der Leistungsbewertung liefert später das Systemmodell die Struktur und das Verkehrsmodell die Parametrierung. Dies erlaubt eine effiziente Leistungsbewertung nach Anforderung 1, da aufgrund der komplexen Abhängigkeit der Modellkomponenten mehrere Iterationen notwendig sind und durch die Modelltrennung die Veränderungen auf das Verkehrsmodell beschränkt bleiben, so dass keine strukturellen Änderungen notwendig sind.

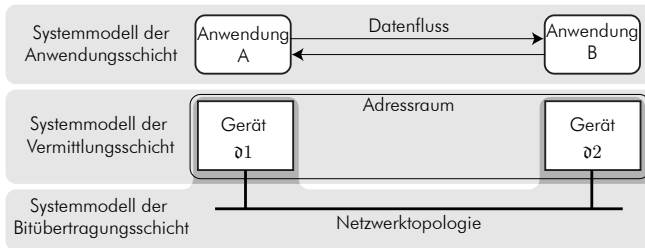


ABBILDUNG 5.1. Schematische Aufteilung des Systemmodells in seine Teilmodelle.

Das Systemmodell ist in weitere Teilmodelle untergliedert, die jeweils die Bitübertragungs-, Vermittlungs- und Anwendungsschicht des OSI-Referenzmodells widerspiegeln. Diese Teilmodelle stellen unterschiedliche Informationsebenen eines Netzwerkentwurfs dar, die bei der Konfigurierung festgelegt werden. Das Systemmodell der Bitübertragungsschicht modelliert die *Topology* des Netzwerks, also wie die Geräte durch Kabel physisch verbunden sind. Über diese Topologie tauschen die Geräte miteinander Nachrichten aus und verwenden dabei Adressen. Der *Adressraum* der Geräte wird durch das Systemmodell der Vermittlungsschicht modelliert. Das Systemmodell der Anwendungsschicht wiederum erfasst, wie die Anwendungen der Geräte miteinander interagieren. Alle Teilmodelle be-

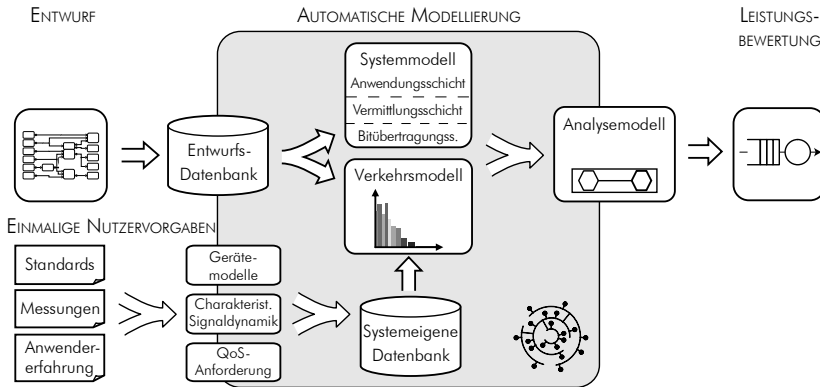


ABBILDUNG 5.2. Detaillierte Struktur der automatischen Modellierung.

sitzen als gemeinsamen Bezugspunkt das Gerät. Die Abbildung 5.1 stellt dies schematisch dar.

Basierend auf der Abbildung 3.2 des Control Network Performance Engineering gibt die Abbildung 5.2 einen detaillierten Einblick in den Prozess der automatischen Modellierung. Die in der Abbildung zur Modellierung herangezogenen Informationsquellen werden im nächsten Abschnitt diskutiert.

5.1.1.1. INFORMATIONSQUELLEN ZUR AUTOMATISCHEN MODELLIERUNG

Nach Prämisse 3 ist eine elektronisch lesbare Repräsentation des Netzwerkentwurfs eine Voraussetzung für eine automatische Leistungsbewertung im CNPE. Ein Beispiel hierfür ist das LonWorks Network Operating System (LNS) [48]. Es wird von verschiedenen Tools zur Entwicklung, Integration und Überwachung von LON-Systemen eingesetzt [2, 47, 169]. Hierbei dient es nicht nur dem Speichern des Netzwerkentwurfs, sondern kann auch online an das Netzwerk angeschlossen werden, um die Inbetriebnahme vorzunehmen oder einzelne Werte im Netzwerk zu überwachen. Die unterschiedlichen Tools greifen über eine COM-Schnittstelle¹ [152] auf die objektorientierte Datenbank zu. Eine ähnliche Datenbank besitzt das Standard-Entwurfswerkzeug ETS 3² [59] von Konnex, das mit der DCOM-Programmierschnittstelle³ Falcon [125] einen Zugang zur Entwurfsdatenbank bereitstellt. Für BACnet existiert derzeit keine vergleichbare, etablierte Entwurfsdatenbank.

1 COM: en. Component Object Model

2 ETS: en. EIB Tool Software

3 DCOM: en. Distributed Component Object Model

Die LNS-Entwurfsdatenbank beinhaltet alle Informationen, die während des Netzwerkentwurfs relevant sind (Abschnitt 2.2), beginnend bei der Topologie über die Adressierung bis zum Anwendungsmodell. Damit können die wesentlichen Teile des Systemmodells konstruiert werden (Abschnitt 5.2 bis 5.5). Für das Verkehrsmodell sind zusätzliche Informationen über das Kommunikationsbedürfnis der Geräte notwendig. Hierfür wird ihr Verhalten in *Gerätemodellen* modelliert, die für verschiedene Entwürfe wiederverwendbar sind und angepasst werden. Sie werden in einer separaten, systemeigenen Datenbank abgelegt, die, wie in Abbildung 5.2 dargestellt, in den Prozess des CNPE eingeht.

Einige detaillierte Modellparameter des Netzwerks und der Geräte fehlen in der Entwurfsdatenbank. Stattdessen weisen die Inbetriebnahmetools Standardwerte zu oder die Geräte werden bei der Herstellung parametrisiert. Hierbei kann zwischen systemspezifischen und gerätespezifischen Standardwerten unterschieden werden. Systemspezifische Parameter werden von allen Geräten verwendet, zum Beispiel Netzwerkparameter. Als Informationsquelle können Standards wie [54], Empfehlungen [143] und allgemeine Gerätedokumentationen [161] dienen. Gerätespezifische Parameter wiederum können spezifischen Gerätedokumentationen entnommen werden und ergänzen das Gerätemodell. Diese Standardparameter sind wiederverwendbar und Teil der systemeigenen Datenbank.

Auf diese Weise kann auf Basis der Entwurfsdatenbank und der systemeigenen Datenbank die Modellierung gemäß Anforderung 5 automatisch erfolgen. Der Zusatzaufwand des Anwenders beschränkt sich auf die einmalige Bereitstellung der Gerätemodelle und gerätespezifischen Parameter, die auch aus Messungen generiert werden können [75].

5.1.2. BEISPIELNETZWERK

In dieser Arbeit wird zur Verdeutlichung der Modellierung und Analyse ein Beispielnetzwerk verwendet, das einen Büroraum in einem Verwaltungsgebäude mit einem hohen Ausstattungsgrad nach dem Beispiel 8 der VDI 6015 [221] entspricht. Es wurde ausgewählt, da es viele für die spätere Netzwerkmodellierung relevante Eigenschaften besitzt.

Im Büroraum sind eine Raumtemperaturregelung, eine Konstantlichtregelung, eine Beschattungsanlage sowie ein Rauchmeldesystem realisiert. Dafür gibt es unter den Fenstern zwei Flächenheizkörper mit elektrischen Stellventilen 07, 08 sowie ein Anzeigegerät 06 zur Raumtemperatur-, Beleuchtungs- und Jalousiesteuerung mit Sollwertversteller, Betriebsartentaster und Display. Der Raumtemperaturfühler 05, Helligkeitssensor 02 und die Fensterkontakte 09, 011 sind jeweils separat angebracht. Zusätzlich gibt es neben einem Präsenzmelder 08 eine Außenjalousie mit Elektroantrieb 03 und Sonnenscheinautomatik 01. Die Raumtemperaturregelung wechselt bei Anwesenheit von Personen aus dem Stand-by in den Komfort-Betrieb. Zusätzlich wird bei offenem Fenster die Heizung aus-

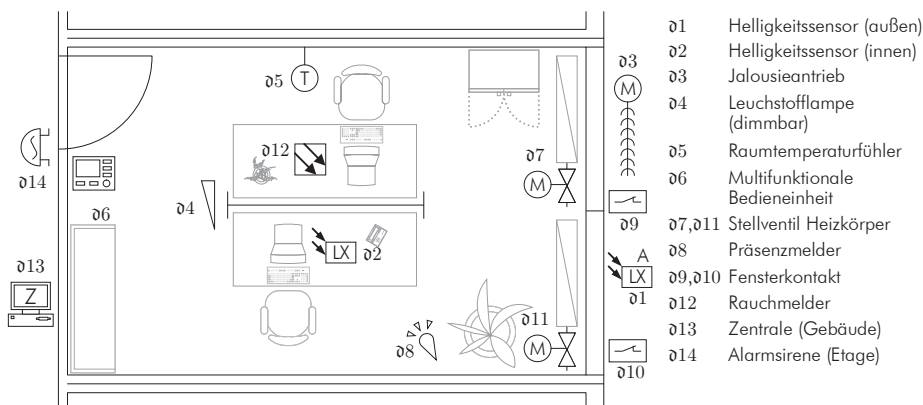


ABBILDUNG 5.3. Installationsplan des verwendeten Beispiels.

geschaltet. Auf die Raumlufttechnik, eine zentrale Messwerterfassung und Sollwertschiebung wurde zur Bewahrung der Übersichtlichkeit verzichtet. Zusätzlich wurde ein Rauchmelder $\varnothing 12$ in den Raum und eine Alarmsirene $\varnothing 14$ in den Flur integriert, wie es die VDI 6010 [220] und das Beispiel 10 der VDI 6015 [221] vorgeschlagen. Die Zentrale $\varnothing 13$ überwacht zum Brandschutz den Rauchmelder ständig und zur Zutrittskontrolle den Bewegungsmelder und die Fensterkontakte. Die Sirene wird im Alarmfall aktiviert.

Um im späteren Verlauf der Arbeit größere Beispiele zu generieren, wird dieser Einzelraum zu einem Gebäude von m Etagen mit jeweils n Einzelräumen vervielfältigt. Dadurch wird ein nachvollziehbar skalierendes Beispiel geschaffen, das sich auf die Aussagen von GRZEMBA [76] und CHRISTEN et al. [23] stützt, nach denen sich auch hochkomplexe Gebäude auf maximal 20 verschiedene Raumprofile reduzieren lassen. Diese wichtige Eigenschaft der Gebäudeautomation (vgl. Tabelle I) vereinfacht nicht nur den Einsatz von Entwurfsmustern in der Konfigurierung, sondern wird auch bei der Modellierung im weiteren Verlauf ausgenutzt.

5.2. SYSTEMMODELL DER BITÜBERTRAGUNGSSCHICHT

Das *Systemmodell der Bitübertragungsschicht* modelliert die *Topologie* eines Netzwerks, die im Wesentlichen festlegt, an welchen *Kanälen* (en. *Channel*) die *Geräte* (en. *Device*) angeschlossen sind und wie die Kanäle durch *Router* verbunden sind. BACnet, Konnex und LON unterstützen nach Abschnitt 4.2 meist freie Topologien [40, 211] und können mit unterschiedlichen Übertragungsmedien

realisiert sein. Dadurch können sie identisch modelliert werden, wobei in dieser Arbeit davon ausgegangen wird, dass:

PRÄMISSE 4: KABELGEBUNDEN

Das zu untersuchende Netzwerk nutzt kabelgebundene Übertragungsmedien, beziehungsweise setzt kurze Funk- oder Infrarotstrecken ein, die sich im Verhalten durch ein Kabel approximieren lassen.

Die Abbildung 5.4 zeigt die Topologie eines einzelnen Büorraumes aus dem Beispiel aus Unterabschnitt 5.1.2. In dem Beispiel ist jeweils eine Etage des Bürogebäudes durch einen Kanal verbunden, so dass sowohl die Geräte eines Büorraumes $\partial 2$ bis $\partial 12$ als auch die Alarmsirene $\partial 14$ der Etage an dem gleichen Kanal $c1$ angeschlossen sind. Ein leistungsstarker Kanal $c0$, der sogenannte *Backbone*, verbindet die einzelnen Kanäle $c1 \dots c_m$ der m Etagen miteinander. Die entstehende Baumstruktur ist sehr typisch für die Gebäudeautomation und unter anderem Grundlage der Konnex Topologie. Die Kanäle sind durch *Router* verbunden, die Nachrichten gezielt weiterleiten können. Neben Routern gibt es je nach Netzwerkprotokoll vergleichbar arbeitende Geräte, die eine routende Funktion übernehmen, also Nachrichten gezielt weiterleiten (Bridge, Coupler, Hub, Repeater, Router, Switch), zusätzlich in andere Protokollformate umwandeln (Gateway) oder kapseln (Tunnel-Router) [40, 115, 133, 211]. In dem Beispiel sind neben den Routern nur die zentralen Geräte des Gebäudes, der Außenlichtsensor $\partial 1$ und die Zentrale $\partial 13$ direkt an dem Backbone angeschlossen.

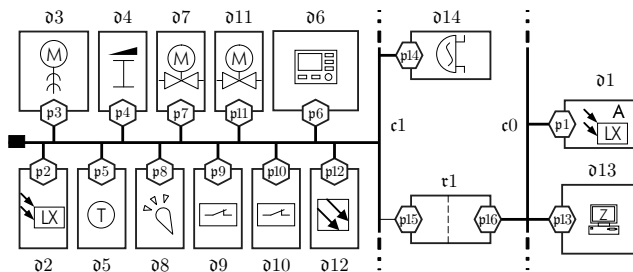


ABBILDUNG 5.4. Topologie am Beispiel des Einzelraums.

Diese Elemente des physischen Netzwerks und ihre Beziehung zueinander werden durch das UML-Modell in Abbildung 5.5 repräsentiert. Korrespondierend zum Kanal, Gerät und Router gibt es die Klassen *Channel* \mathcal{C} , *Device* \mathcal{D} und *AbstractRouter* \mathcal{R} . Diese Klassen sind alle der Klasse *Network* komposit untergeordnet, welche das gesamte Modell bündelt. Die unterschiedlichen Routervarianten wie Router, Gateway oder Bridge leiten sich im UML-Modell von der Elternklasse *AbstractRouter* ab. Die physische Verbindung zwischen einem Kanal und einem Gerät und zwischen mehreren Kanälen und einem Router wird

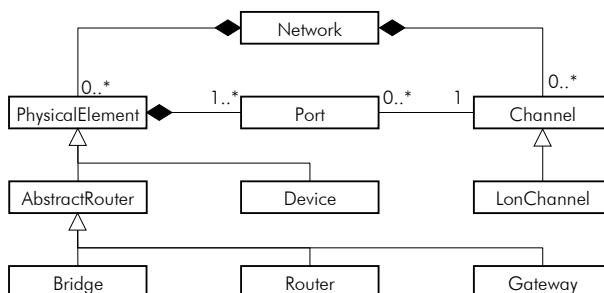


ABBILDUNG 5.5. UML-Darstellung des Systemmodells der Bitübertragungsschicht.

über die zusätzliche Klasse *Port* \mathfrak{P} ausgedrückt. Da sowohl *Device* als auch *AbstractRouter* mindestens einen *Port* besitzen, leiten sich beide Klassen von der Elternklasse *PhysicalElement* ab. Damit werden auch redundante Kanäle bei SafetyLON [244] unterstützt.

Die Klassen besitzen für das Beispiel LON die in Aufzählung 1 aufgeführten Attribute. Die Symbole der Attribute kennzeichnen, ob die Informationen in der LNS-Datenbank verfügbar, wiederherstellbar oder mit Standardwerten belegt sind. Für BACnet oder Konnex muss unter Prämisse 4 nur die Klasse *Channel* (z. B. *LonChannel*) an die spezifischen Kanalparameter angepasst werden. Dadurch kann prinzipiell auch ein Netzwerkentwurf modelliert werden, der verschiedene Netzwerkprotokolle enthält.

AUFZÄHLUNG 1. Auszug aus den Attributen der Klassen aus Abbildung 5.5

(● In LNS vorhanden (sofern definiert); ⊗ Aus Standards zuzuordnen;

★ Default aus Standards/Messungen; – Unbekannt;

¹ Standardisiert für gängige Transceiver und über TransceiverID zuzuordnen).

PhysicalElement

- name (Bezeichner in Entwurfstool)
- manufacturer (Gerätehersteller)
- modelType (Gerätetypbezeichner)

Port

- ★ bufferSize (Warteschlangengröße)
- prioritySlot (Zugewiesener Prioritätsslot)

LonChannel

- name (Bezeichner Entwurfstool)
- ⊗ beta1Time¹ (Dauer Beta1-Slot)
- ⊗ beta2Time¹ (Dauer Beta2-Slot)
- ★ bitError (Fehlerrate des Mediums)
- ⊗ bitRate¹ (Bitrate des Mediums)
- ⊗ collisionDetection¹ (Kollisionserkennung Ja/Nein)
- ⊗ maxDistance^{1,2} (Max. Kabellänge)
- ⊗ preambleTime¹ (Preamblelänge)
- priorities (Anzahl Prioritätsslot)

Für das Beispiel LON kann die Struktur und ein Großteil der in Aufzählung 1 angegebenen Attribute aus der LNS-Datenbank gelesen werden. Nur die `Port.bufferSize` und die `LonChannel.bitError` sind nicht vorhanden. Für die `bufferSize` werden Empfehlungen aus Betriebshandbüchern [161] und Standards [54] genommen und für den `bitError` übliche Werte aus Messungen, die in Unterabschnitt 8.3.5 diskutiert werden. Die Bedeutung der Attribute wird im Laufe der Arbeit noch erklärt.

5.3. SYSTEMMODELL DER VERMITTLUNGSSCHICHT

Damit die Geräte im Netzwerk gezielt Nachrichten austauschen können, benutzen sie eindeutige, logische *Adressen*, die jedem Gerät zugewiesen sind. Das *Systemmodell der Vermittlungsschicht* modelliert diese logische Adressierung.

Sowohl BACnet, LON und Konnex nutzen hauptsächlich ein numerisches, hierarchisch organisiertes Adressformat [115]. BACnet unterstützt durch seine austauschbaren unteren Schichten unterschiedliche Adressformate wie IP-, MAC- oder LON-Adressen [115, 130]. Bei Konnex besitzen die Geräte eine eindeutige Adresse, die der Topologie folgt und sich aus *Zone.Linie.Device* zusammensetzt. In LON besteht die eindeutige Adresse aus *Domain.Subnetz.Node*⁴. Diese eindeutige Adresse erlaubt eine verbindungslose *Punkt-zu-Punkt-Kommunikation* (en. *Point-to-Point Communication*) zwischen zwei Geräten, die als *Unicast-Adressierung* bezeichnet wird. Hierbei gibt es bei LON die Einschränkung, dass Sender und Empfänger Mitglied der gleichen Domäne sein müssen.

Daneben unterstützen Konnex und LON zusätzlich die *Multicast-Adressierung*, bei der eine Nachricht mehrere Empfänger besitzt. Hierfür werden die Geräte in Gruppen zusammengefasst und diese Gruppe als Adresse verwendet. Bei LON müssen gruppierte Geräte der gleichen Domäne angehören, in Konnex können diese Gruppen verschiedene Zonen übergreifen.

Die dritte Adressierungsmöglichkeit ist die *Broadcast-Adressierung*, bei der alle Geräte eines logischen Segmentes angesprochen werden, also alle Geräte einer Zone oder Linie in Konnex oder einer Domäne oder Subnetz in LON oder BACnet/IP.

Die Abbildung 5.6 zeigt die logische Segmentierung des Beispiels aus Unterabschnitt 5.1.2 in LON. Die Zentrale 1.1.1 und der Außenlichtsensor 1.1.13 bilden das Subnetz 1.1. Zusätzlich liegt die Alarmsirene einer Etage in einem eigenen Subnetz (im Bsp. 1.2). Alle Geräte eines Büroraumes sind jeweils einem Subnetz zugeordnet (im Bsp. 1.3). Die Heizungsventile eines jeden Raumes sind gruppiert,

⁴ Bei LON kann zusätzlich die Neuron-ID adressiert werden, eine weltweit eindeutige 48Bit-Adresse des Hardware-Chips. Diese Adresse wird nur zu Konfigurierungszwecken genutzt und kann im Modell auf eine *Domain.Subnetz.Node*-Adresse abgebildet werden.

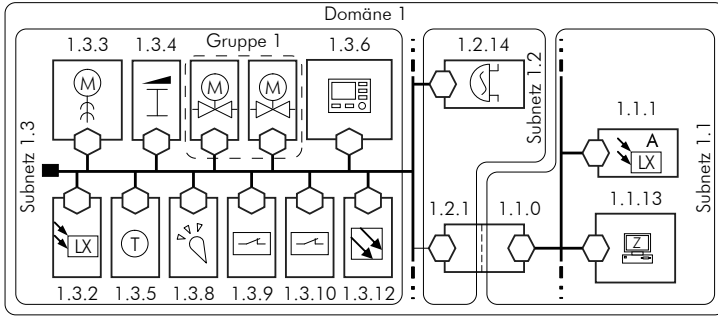


ABBILDUNG 5.6. Adressierung am Beispiel des Einzelraums.

damit die Einzelraumregelung beide mit einer Nachricht ansprechen kann. Alle Geräte liegen innerhalb einer Domäne.

Die Abbildung 5.5 demonstriert die grundsätzliche Modellierung in UML für das Beispiel LON. Schlüsselobjekt des Modells ist die Klasse `LogicalElement` von der sich alle adressierbaren Klassen `Domain`, `Subnet`, `Node` ableiten. Durch entsprechende Komposition wird die Hierarchie `Domain.Subnet.Node` gebildet. In der Abbildung 5.5 ist die Klasse `Group` dem `Network` untergeordnet. Dies stellt eine Verallgemeinerung der Unterordnung von Gruppen unter Domains bei LON in Bezug auf Konnex dar. Die Assoziation von `Node` zu `Port` stellt die Verbindung zum Modell der Bitübertragungsschicht und den einzelnen Geräten her. Dabei kann ein `Port` mehrere `Node`-Adressen besitzen, wie es in LON theoretisch möglich ist. Die Konnex-Adressierung ist durch eine Abbildung von `Zone.Linie.Device` auf `Domain.Subnet.Node`-Klassen durch das gleiche Modell darstellbar. Bei BACnet/IP mit IPv4 würde noch eine vierte Adressebene, zum Beispiel zwischen `Subnet` und `Node`, hinzukommen. Auch andere Adressformate lassen sich einfach in das Modell integrieren, indem sie die Klasse `LogicalElement` ableiten.

Das Systemmodell der Vermittlungsschicht lässt sich im Beispiel LON vollstän-

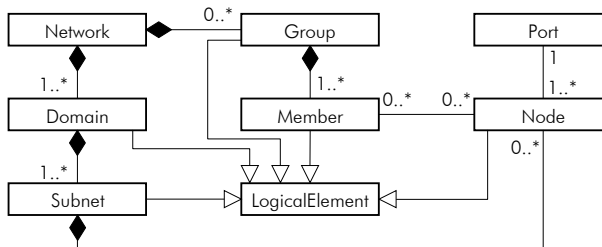


ABBILDUNG 5.7. UML-Darstellung des Systemmodells der Vermittlungsschicht.

dig aus der LNS-Datenbank exportieren. Die einzelnen Klassen besitzen neben ihrer Adresse keine weiteren Attribute.

5.4. SYSTEMMODELL DER ANWENDUNGSSCHICHT

Die *Anwendungsschicht* stellt im OSI-Referenzmodell dem übergeordneten Anwendungsprozess direkt Funktionalität zum einfachen Versand von Nachrichten zur Verfügung. Zum einen sind dies *Netzwerkmanagement-Dienste*, die zum Beispiel die Zuweisung von Adressen zu den Geräten ermöglichen. Zum anderen sind dies spezielle, gemeinsam genutzte Variablen, die bei BACnet als *Datenpunkte*, bei LON als *Netzwerk-Variable* und bei Konnex als *Gruppenobjekt* (en. *Group Object*) bezeichnet werden. Ändert die Anwendung im Betrieb eine solche gemeinsam genutzte Variable, so wird automatisch eine Nachricht erzeugt und an alle bei der Komposition festgelegten Empfängergeräte gesendet. Bei Eintreffen der Nachricht ändert der Empfänger die gemeinsam genutzte Variable und benachrichtigt die Anwendung. Auf diese Weise können beide Anwendungen den Variablenwert synchronisieren, ohne sich über das Versenden von Nachrichten zu kümmern.

Dahinter steht bei allen Protokollen ein auf Funktionsblöcken basierender Anwendungsentwurf [115, 241], der in Abschnitt 2.4 schon angesprochen wurde. Dazu werden abgeschlossene Funktionen eines Gerätes in *Funktionsblöcke* (en. *Function Block*) gekapselt, die nach außen nur Schnittstellen in Form von *Datenpunkten* (en. *Data Point*) anbieten [97, 219], hinter denen sich gemeinsam genutzte Variablen verbergen. Meist werden Eingangsdatenpunkte \mathfrak{J} und Ausgangsdatenpunkte \mathfrak{D} unterschieden, die jeweils nur schreibenden oder lesenden netzwerkseitigen Zugriff erlauben. Prinzipiell gibt es in BACnet und Konnex auch Datenpunkte, die beides erlauben, sie werden allerdings zur Verwendung nicht empfohlen [130]. Ein Gerät kann mehrere Funktionsblöcke implementieren, wie die Bedieneinheit $\mathfrak{d6}$ in Abbildung 5.8. Dieses Funktionsblock-Konzept ist auch in anderen Disziplinen der Automatisierung etabliert [57, 58, 138].

Werden Funktionsblöcke für eine bestimmte Geräteklasse oder Funktion standardisiert, so spricht man in diesem Fall in der Gebäudeautomation von *Profilen* oder *Geräteprofilen* [55, 98, 145]. Die Profile sind im Allgemeinen so spezifiziert, dass sie im Umfeld ihres Gewerks zueinander passen, wodurch die Interoperabilität der implementierenden Geräte verbessert wird.

Bei der Konfigurierung des Netzwerks wird meist auf vorprogrammierte Geräte mit bestehenden Funktionsblöcken zurückgegriffen. Um einen Prozess zu realisieren, verbindet der Systemintegrator einzelne Datenpunkte verschiedener Funktionsblöcke durch logische Verbindungen (*Komposition*), die als *Bindings* in LON, *Gruppe* in Konnex oder gemeinsame Datenpunkte in BACnet bezeichnet werden. Dadurch werden beide Datenpunkte im Betrieb als gemeinsam genutzte Variablen

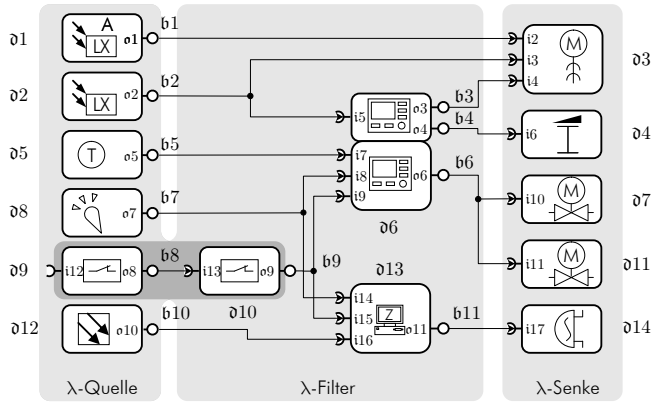


ABBILDUNG 5.8. Systemmodell der Anwendungsschicht am Beispiel des Einzelraums.

synchronisiert. Damit legt der Systemintegrator bereits bei der Konfigurierung die Kommunikationsbeziehungen fest. Dies stellt eine wichtige Eigenschaft von Netzwerken der Automation für die weitere Modellierung dar (Tabelle I), die mit der folgenden Prämisse vorausgesetzt wird.

PRÄMISSE 5: UNVERÄNDERLICHE KOMMUNIKATIONSBEZIEHUNGEN

Es gibt im untersuchten Netzwerk a priori bekannte unveränderliche, gerichtete Kommunikationsbeziehungen.

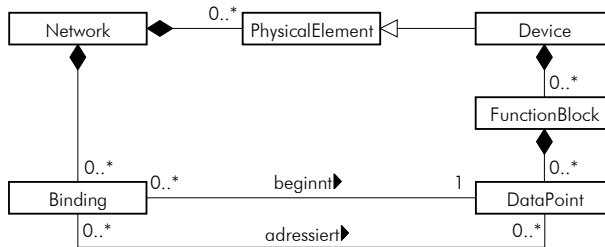


ABBILDUNG 5.9. UML-Darstellung des Systemmodells der Anwendungsschicht.

Das *Systemmodell der Anwendungsschicht* in Abbildung 5.9 modelliert die beschriebenen Funktionsblöcke, Datenpunkte und Verbindungen in den Klassen *FunctionBlock*, *DataPoint* und *Binding* \mathfrak{B} in multipler Komposition. Die zwei Assoziationen zu *DataPoint* der Klasse *Binding* modellieren eine 1-zu- n -Verbindung, mit einem Sender und n Empfängern. In Konnex kann es, durch die Gruppierung von Datenpunkten⁵ mit Schreib/Lese-Zugriff, prinzipiell auch m -

⁵ Bei Konnex werden Datenpunkte, anstelle von gerichteten Bindings, gruppiert. Der Sender

zu- n -Verbindungen geben. Sie können im Modell durch m einzelne 1-zu- n -Verbindungen abgebildet werden. Dies entspricht auch dem physischen Verhalten. Ein Datenpunkt mit Schreib/Lese-Zugriff wird im Weiteren als Element der Menge der Eingangs- \mathcal{I} und Ausgangsdatenpunkte \mathcal{O} behandelt.

AUFZÄHLUNG 2. Auszug aus den Attributen der Klassen aus Abbildung 5.9 (Legende in Aufzählung 1).

FunctionBlock	Binding
<ul style="list-style-type: none"> • name (Bezeichner in Entwurfstool) • profileName (Geräteprofilname) • configurationProperties (Konfigurationsparameter) 	<ul style="list-style-type: none"> • serviceType (Übertragungsdienst) • prior (Nutzr Priorität) • retries (Wiederholungen) • repeatIntervalTimer (Timer wiederholte Nachrichten) • transmissionTimer (Timer bestätigte Nachrichten) • failureRetries (Max. Wiederholungen bei Fehlern)
<p>DataPoint</p> <ul style="list-style-type: none"> • name (Bezeichner im Entwurfstool) • accessDirection (Lesen/Schreiben/...) • variableType (Variablen Typ) 	

Die Attribute der Klassen sind in Aufzählung 2 zusammengestellt. Sie sind für LON, BACnet und Konnex verwendbar und im Beispiel LON vollständig in der LNS-Datenbank verfügbar. Das Attribut `configurationProperties` eines `FunctionBlock` listet die verschiedenen im Netzwerkentwurf eingestellten Konfigurationsparameter auf. Das Attribut `variableType` eines `DataPoints` spezifiziert den semantischen Datentyp eines Datenpunktes, angefangen bei der Form der digitalen Repräsentation als: *Ganzzahl*⁶, *Gleitkommazahl*⁷, *Boolean* oder *Aufzählungsdatentyp* (en. *Enumeration*); über die physikalische Einheit, bis zum Definitionsbereich, der Quantisierung und ähnlichen Parametern. Diese semantischen Datentypen sind meist standardisiert: in LON als *Standard Netzwerk Variablen Typen* (SNVT⁸) [144], in Konnex als *Interworking-Datapoint-Types* [55] und in BACnet als *Standard Object Properties* [98, 130]. Der `serviceType` gibt den *Übertragungsdienst* eines Bindings an, der im nächsten Abschnitt eine Bedeutung für die weitere Modellierung gewinnt.

ist dabei Teil der Gruppe. Dieses Schema wird entsprechend auf eine logische Multi- oder Unicast-Adressierung abgebildet.

6 Ganzzahl: en. Integer

7 Gleitkommazahl: en. Floating point

8 SNVT: en. Standard Network Variable Type

5.5. ZUSAMMENGEFÜHRTES SYSTEMMODELL DER NACHRICHTENKLASSEN

5.5.1. NACHRICHTENKLASSEN

Die bisher vorgestellten Teilmodelle des Systemmodells erfassen die unterschiedlichen Sichtweisen auf einen Netzwerkentwurf. Diese Teilmodelle werden im *Systemmodell der Nachrichtenklassen* zusammengeführt, das sich einfach in ein Simulationsmodell und analytisches Modell überführen lässt.

Eine *Nachrichtenklasse* fasst alle Nachrichten zusammen, die den gleichen Weg durch das Netzwerk nehmen und die gleiche Größe haben. Vorerst sei als These formuliert, dass alle Nachrichten, die von einem beliebigen Datenpunkt o an einen Datenpunkt i eines anderen Gerätes gesendet werden, eine Nachrichtenklasse $m_{o \rightarrow i}$ mit bekannter, unveränderlicher Nachrichtengröße und Weg durch das Netzwerk bilden. Dadurch können diese Nachrichten im Netzwerk in Nachrichtenklassen zusammengefasst werden und müssen nicht einzeln betrachtet werden. Diese These wird im Folgenden näher begründet.

TABELLE III.

Zusammensetzung der Nachrichtengröße in Bit zur Synchronisation von Datenpunkten in LON bei unterschiedlichen Übertragungsdiensten mit der Größe der Adresse S_{ADD} (24bit) inklusive DomainID $S_{DID} \in \{0, 8, 24, 48\}$ und der Größe des Gruppenfeldes $S_{MS} \in \{0, 8, 16, 24, 32, 40, 48, 56, 64\}$ (Abkürzungen nach 11).

Nachrichtentyp	$S_{MPDU}/$ S_{LPDU}	S_{NPDU}	$S_{TPDU}/S_{SPDU}/$ $S_{AuthPDU}$	S_{APDU}
Zu bestätigende Nachricht (AckD)	24	$8 + S_{ADD}$	8	$16 + S_{App}$
Wiederholte Nachricht (Rep)	24	$8 + S_{ADD}$	8	$16 + S_{App}$
Anforderung (Req)	24	$8 + S_{ADD}$	8	16
Antwort (Res)	24	$8 + S_{ADD}$	8	$16 + S_{App}$
Bestätigung (Ack)	24	$8 + S_{ADD}$	8	0
Erinnerung (Rem)	24	$8 + S_{ADD}$	$16 + S_{MS}$	0
Erinnerung&Nachricht (RemMsg)	24	$8 + S_{ADD}$	$16 + S_{MS}$	$16 + S_{App}$
Authentifiz. Aufforderung (Chall)	24	$8 + S_{ADD}$	80	0
Authentifiz. Antwort (Reply)	24	$8 + S_{ADD}$	80	0

Die *Nachrichtengröße* S_{phy} bezeichnet die physische Größe einer Nachricht auf dem Kanal in Bits. Sie setzt sich aus der *Informationsgröße* S_{app} auf der Anwendungsschicht und den auf den einzelnen OSI-Schichten hinzugefügten Protocol Data Units (PDU) zusammen. Die Informationsgröße wird durch den programmierten Variablentypen des Datenpunktes unveränderlich festgelegt. Die Größen der PDUs sind in den Protokollstandards festgelegt und hängen bei LON von

dem verwendeten Adressierungsschema und dem Übertragungsdienst der Nachricht ab (Tabelle III). Da den Bindings feste Adressen und Übertragungsdienste bei der Konfigurierung zugewiesen werden, ist damit die Nachrichtengröße aller Nachrichten eines Bindings festgelegt. Damit gilt in der Gebäudeautomation:

PRÄMISSE 6: UNVERÄNDERLICHKEIT DER NACHRICHTENGRÖSSE

Der Variablentyp, die Adressierung und die Übertragungsdienste der Kommunikationsbeziehungen aus Prämisse 5 sind im Betrieb unveränderlich und im Netzwerkentwurf bekannt.

Die Tabelle IV gibt für das Beispiel aus Unterabschnitt 5.1.2 die unterschiedlichen Informationsgrößen.

TABELLE IV.

Nachrichtengröße in Bit mit $S_{\text{DID}} = 8$ Bit für das Beispiel Unterabschnitt 5.1.2 in LON (SNVTs nach den LonMark Functional Profiles [144, 145]; $\|\mathfrak{M}_b\|$ Anzahl der Nachrichtenklassen; ¹ Der eigentliche binäre Zustandstyp SNVT_switch wird als kontinuierlichen Variable zwischen (0 [On] ... 1 [Off]) genutzt; ² Größe der ersten Nachrichtenklasse).

Binding	Quelle	Größe	SNVT	S_{app}	Dienst	S_{phy}^2	$\ \mathfrak{M}_b\ $
b1	o1	Beleuchtungsstärke	SNVT_lux	16	Unack	104	1
b2	o2	Beleuchtungsstärke	SNVT_lux	16	Rep 3x	104	1
b3	o3	Jalousiestand	SNVT_setting	32	Rep 3x	120	1
b4	o4	Lampendimmung	SNVT_switch ¹	8	Rep 3x	96	1
b5	o5	Temperatur	SNVT_temp_p	16	Unack	104	1
b6	o6	Ventilöffnung	SNVT_lev_percent	16	Unack	104	1
b7	o7	Anwesenheit	SNVT_occupancy	8	Ack	96	4
b8,b9	o8,o9	Fensteröffnung	SNVT_switch	8	Ack	96	3, 4
b10	o10	Rauchmeldung	SNVT_switch ¹	8	Ack	96	3
b11	o11	Alarm	SNVT_switch ¹	8	Auth	96	14

Die Unveränderlichkeit der Nachrichtenwege setzt zum einen voraus, dass sich die Netzwerktopologie im Betrieb nicht ändert und zum anderen, dass der Nachrichtenweg zwischen zwei Geräten eindeutig ist. Die erste Voraussetzung ist unter Prämisse 3 erfüllt und die zweite zumindest bei Konnex grundsätzlich gegeben, da nur eine Baumtopologie mit Zone.Linie.Device erlaubt ist [115]. BACnet impliziert eindeutige Wege, da es nur lernende Router [130] erlaubt, die bei einfachen Routingalgorithmen eine statische, eindeutige Topologie voraussetzen. In LON sind prinzipiell mehrwegige Ringtopologien [40] möglich. Sie werden aber nur selten mit speziellen Routern realisiert, die im fehlerfreien Normalbetrieb den Ring in eine eindeutige Linie auflösen [87]. Die in SafetyLON möglichen redundanten Ports sind mit physisch getrennten Netzwerken verbunden, so dass die Prämis-

se 7 für die einzelnen Netzwerke erfüllt ist. Damit gilt ohne Einschränkung in der Gebäudeautomation:

PRÄMISSE 7: EINDEUTIGKEIT DER WEGE

Zwischen zwei Geräten kann eine Nachricht maximal einen eindeutigen Weg nehmen.

5.5.2. ÜBERTRAGUNGSDIENSTE

Die schon mehrmals angesprochenen *Übertragungsdienste* (en. *Message Service Type*) realisieren Mechanismen zur Fehlersicherung der Übertragung durch mehrfache, geregelte Übertragungsversuche. Die Realisierung der Übertragungsdienste ist fast immer protokollspezifisch, in den grundlegenden Mechanismen jedoch vergleichbar und wird am Beispiel LON erklärt.

Bei einem *bestätigten Übertragungsdienst* (en. *Acknowledged*) sendet der Empfänger bei Erhalt der Nachricht (Acknowledged-Request; AckD) eine Bestätigung (Acknowledgment; Ack). Bleibt die Bestätigung beim Sender innerhalb eines `binding.transmissionTimers` aus, so geht er davon aus, dass der Empfänger die zu bestätigende Nachricht nicht erhalten hat, und wiederholt seine Nachricht mit einer Erinnerung bis zu `binding.failureRetries`-mal. Hierbei wird bei einer Gruppenadressierung an mehr als 16 Empfänger, in LON die Erinnerung (Reminder; Rem) getrennt gesendet und ansonsten mit der ersten Nachricht zusammen (Reminder-Message; RemMsg). Alle Nachrichten (AckD, Ack, Rem), die zwischen Sender und Empfänger innerhalb eines Übertragungsdienstes versendet werden, bilden eine *Transaktion*.

Neben dem bestätigten Übertragungsdienst gibt es noch den *unbestätigt wiederholten Übertragungsdienst* (en. *Unacknowledged Repeated*), bei dem die Nachricht im Abstand eines *Wiederholungstimers* (`binding.repeatIntervalTimer`) genau `binding.retries`-mal hintereinander gesendet wird. Ist `binding.retries` genau eins, so tritt der Sonderfall des *unbestätigten Übertragungsdienstes* (en. *Unacknowledged*) auf.

Die Sitzungsschicht in LON realisiert zusätzlich einen *beantworteten Übertragungsdienst* (en. *Request (Req) / Response (Res)*), bei dem der Sender den Empfänger auffordert, ihn den aktuellen Wert eines Datenpunktes zu übermitteln. Zusätzlich gibt es den *authentifizierten Übertragungsdienst* (en. *Authenticated*), der mit den anderen Diensten kombiniert werden kann und den Empfänger auffordert, die Echtheit des Senders zu überprüfen (mittels Challenge (Chall), Reply) [40]. Damit erfüllt der Dienst die wachsenden Anforderungen an die *Sicherheit* (en. *Security*) in der Automation [200, 215].

BACnet erlaubt unbestätigte, bestätigte und beantwortete Übertragungsdienste mit zusätzlicher Segmentierung großer Nachrichten. In Konnex gibt es nur einen bestätigten Übertragungsdienst, der für die gängige Gruppenadressierung optimiert wurde. Hierbei antworten alle Empfänger innerhalb eines speziellen

Segmentes noch in der gleichen Nachricht, wobei negative Bestätigungen positive überschreiben.

5.5.3. SYSTEMMODELL DER NACHRICHTENKLASSEN

Das *Systemmodell der Nachrichtenklassen* besteht im Wesentlichen aus den eingeführten Nachrichtenklassen. Aus der These, dass alle Nachrichten, die von einem Datenpunkt σ an einen anderen Datenpunkt i gesendet werden, eine Nachrichtenklasse $\mathfrak{m}_{\sigma \rightarrow i}$ bilden, folgt, dass jedes Binding mindestens eine Nachrichtenklassen in Richtung des Informationsflusses besitzt. Durch die Übertragungsdienste zerfallen die Bindings sogar in mehrere Nachrichtenklassen, da in LON zum Beispiel auf die ersten Nachrichten eines bestätigten Übertragungsdienstes, die Bestätigung mit einer anderen Größe und auf einem entgegengesetzten Weg folgt. Sollten zusätzlich Erinnerungen notwendig sein, so bilden sie die dritte Nachrichtenklasse. Dabei sind die Regeln, wie die Nachrichtenklassen für die Bindings bei bestimmten Übertragungsdiensten gebildet werden, äußerst protokollspezifisch und werden im Folgenden für LON demonstriert.

Grundsätzlich gibt es immer eine Nachrichtenklasse in Richtung des Steuerflusses von σ nach i . Bei einem wiederholten Übertragungsdienst haben alle Wiederholungen die gleichen Wege und Nachrichtengrößen und bilden keine neue Klasse. Beim bestätigten, beantworteten und authentifizierten Übertragungsdienst mit n Empfängern gibt es eine Nachrichtenklasse für die Erinnerung (entweder Reminder-Message oder nur Reminder) und dann n Nachrichtenklassen für die Antworten und gegebenenfalls wieder n Klassen für deren Antworten, etc. So gibt es in LON pro Binding $\mathfrak{b} \in \mathfrak{B}$ mit $\|\mathfrak{P}_{\mathfrak{b}}\|$ Empfängerports

$$\|\mathfrak{M}_{\mathfrak{b}}\| = \begin{cases} 1, & \text{unbestätigt;} \\ 2 + \|\mathfrak{P}_{\mathfrak{b}}\|, & \text{bestätigt, beantwortet;} \\ 2 + 2 \|\mathfrak{P}_{\mathfrak{b}}\|, & \text{unbestätigt \& authentifiziert;} \\ 2 + 3 \|\mathfrak{P}_{\mathfrak{b}}\|, & \text{bestätigt, beantwortet \& authentifiziert.} \end{cases} \quad (5.1)$$

Nachrichtenklassen $\mathfrak{M}_{\mathfrak{b}}$. Zur Demonstration werden die Übertragungsdienste am Binding $\mathfrak{b}1$ verglichen, wobei die Reihenfolge der Nachrichtenklassen nicht der chronologischen Ordnung entspricht.

$$\begin{aligned} \mathfrak{m}_{\mathfrak{b}1} | \text{wiederholt} &\Rightarrow \{ \overbrace{\mathfrak{w}_{\mathfrak{p}1 \rightarrow \mathfrak{p}3}} \}, \\ \mathfrak{m}_{\mathfrak{b}1} | \text{bestätigt} &\Rightarrow \{ \overbrace{\mathfrak{w}_{\mathfrak{p}1 \rightarrow \mathfrak{p}3}}^{\text{Bestätigungsaufforderung}}, \overbrace{\mathfrak{w}_{\mathfrak{p}1 \rightarrow \mathfrak{p}3}}^{\text{Erinnerung}}, \overbrace{\mathfrak{w}_{\mathfrak{p}3 \rightarrow \mathfrak{p}1}}^{\text{Bestätigung}} \}, \\ \mathfrak{m}_{\mathfrak{b}1} | \text{beantwortet} &\Rightarrow \{ \overbrace{\mathfrak{w}_{\mathfrak{p}1 \rightarrow \mathfrak{p}3}}^{\text{Anforderung}}, \overbrace{\mathfrak{w}_{\mathfrak{p}1 \rightarrow \mathfrak{p}3}}^{\text{Erinnerung}}, \overbrace{\mathfrak{w}_{\mathfrak{p}3 \rightarrow \mathfrak{p}1}}^{\text{Antwort}} \}, \end{aligned}$$

$$\begin{aligned}
 m_{b1}|_{\text{authentifiz.}} &\Rightarrow \left\{ \overbrace{w_{p1 \rightarrow p3}}^{\text{Anforderung}}, \overbrace{w_{p1 \rightarrow p3}}^{\text{Erinnerung}}, \overbrace{w_{p3 \rightarrow p1}}^{\text{Challenge}}, \overbrace{w_{p1 \rightarrow p3}}^{\text{Reply}} \right\}, \\
 m_{b1}|_{\text{authentifiz.} \wedge \text{beantwortet}} &\Rightarrow \left\{ \overbrace{w_{p1 \rightarrow p3}}^{\text{Anforderung}}, \overbrace{w_{p1 \rightarrow p3}}^{\text{Erinnerung}}, \overbrace{w_{p3 \rightarrow p1}}^{\text{Challenge}}, \overbrace{w_{p1 \rightarrow p3}}^{\text{Reply}}, \overbrace{w_{p3 \rightarrow p1}}^{\text{Bestätigung}} \right\}.
 \end{aligned}$$

Bei einer Multicast-Verbindung wie dem Binding $b2$ folgt zum Beispiel

$$m_{b2}|_{\text{bestätigt}} \Rightarrow \left\{ \overbrace{w_{p2 \rightarrow p3 \wedge p6}}^{\text{Bestätigungsaufforderung}}, \overbrace{w_{p2 \rightarrow p3 \wedge p6}}^{\text{Erinnerung}}, \overbrace{w_{p3 \rightarrow p2}, w_{p6 \rightarrow p2}}^{\text{Bestätigungen}} \right\}.$$

Um den Weg $w_{p_o \rightarrow p_i}$ zu bestimmen, den die Nachrichten einer Klasse nehmen, ist es notwendig, den Senderport p_o und die Empfängerports p_i zu identifizieren. Hierzu werden die Elemente des Systemmodells zusammengeführt. Aus dem Systemmodell der Anwendungsschicht in Abbildung 5.9 sind alle Empfängerdatenpunkte eines Bindings bekannt. Für jeden Datenpunkt lässt sich über den `FunctionBlock` auf das aggregierende `Device` schließen und aus dem Systemmodell der Bitübertragungsschicht auf den zugehörigen `Port`⁹. Um die Adressierung zu bestimmen, kann im Systemmodell der Vermittlungsschicht dem `Port` der zugehörige `Node` zugeordnet werden. Gibt es mehr als einen Empfänger-`Node`, so werden sie in einer neuen `Group` zusammengefasst¹⁰. Die resultierende `Group` oder der `Node` kann als logische Adresse vom Binding benutzt werden. Damit lässt sich für jedes Binding ein `LogicalElement` als Absender- und Empfängeradresse bestimmen, die auf eine Menge von `Nodes` und `Ports` abzubilden sind. Diese Abhängigkeiten spiegelt das Modell der `MessageClass` in Abbildung 5.11 wieder.

Anhand des Sende- und der Empfängerports lässt sich unter Prämisse 7 der Weg, den eine Nachricht durch das Netzwerk nimmt, eindeutig bestimmen. Entsprechende Wegesuchalgorithmen liefert die Graphentheorie [123, 197]. Im Modell besteht ein Weg `Route` aus `Ports`, `AbstractRoutern` und `Channels`, die sich zur einfachen Erweiterbarkeit von `Routeable` ableiten. Die Wege sind komposit, so teilen sich alle Ports an den Kanälen cC und cD den Weg $w_{cC \rightarrow cD}$. Ist weiterhin Kanal cE über einen Router zu cD verbunden, so führt der Weg $w_{cC \rightarrow cE}$ über den Weg $w_{cC \rightarrow cD}$. Die Komposition $w_{cC \rightarrow cD} \subset w_{cC \rightarrow cE}$ ermöglicht eine speichereffiziente Implementierung und effiziente Analyse.

Bei einer Multicast-Verbindung mit mehreren Empfängerports kommt es zu einer Aufspaltung der Wege. Zum einen an Routern, wenn die Empfänger in verschiedenen Kanälen liegen, und sonst an den Kanälen zu den Empfängerports.

⁹ Besitzt ein Gerät mehrere Ports, so sind diese nach Prämisse 7 getrennt zu behandeln.

¹⁰ Dies ist notwendig, da die Informationen zur Gruppenbildung und die Bindings in der LNS-Datenbank nicht direkt abgespeichert werden. In den Datenpunkten werden nur die Empfänger abgespeichert. Diesen werden allerdings zur Adressierung *Selektoren* zugewiesen, die bei einem Binding gleich sind, so dass die Bindings und Gruppen eindeutig wiederhergestellt werden können.

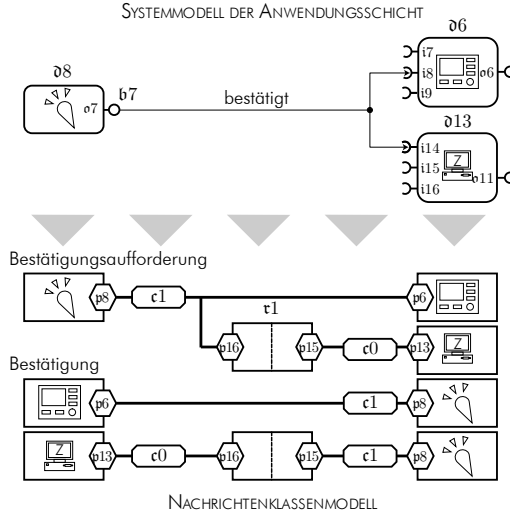


ABBILDUNG 5.10. Beispiel der Nachrichtenklassen für das Binding $b7$ in Unterabschnitt 5.1.2 bei bestätigtem Übertragungsdienst ohne die Reminder-Nachrichtenklasse.

Dadurch ergibt sich ein Baum, der sich vom Sendeport aus beginnend aufspaltet. Durch Prämisse 7 ist ein Überschneiden von Ästen auszuschließen. Die Abbildung 5.10 zeigt den Weg für das Binding $b7$ des Beispiels.

Für das Beispiel aus Unterabschnitt 5.1.2 führt der Weg von dem Helligkeitssensor außen $d1$ zur Jalousie $d3$ über den Router $r1$

$$w_{p1 \rightarrow p3} = \{p1, c0, p16, r1, p15, c1, p3\}.$$

Der Helligkeitssensor im Raum $d2$ sendet nicht nur zur Jalousie $d3$, sondern auch an das Raumbediengerät $d6$. Da aber alle Geräte am gleichen Kanal $c1$ liegen, folgt mit der Gruppenadressierung

$$w_{p2 \rightarrow p3 \wedge p6} = \left\{ p2, c1, \left\langle \begin{matrix} p3 \\ p6 \end{matrix} \right\rangle \right\}.$$

Auch der Fensterkontakt $d10$ sendet an zwei Empfänger: das Raumbediengerät $d6$ und die Zentrale $d13$. Diese liegt aber am Kanal $c0$, muss also über den Router $r1$ erreicht werden. Entsprechend ist

$$w_{p10 \rightarrow p6 \wedge p13} = \left\{ p10, c1, \left\langle \begin{matrix} p6 \\ p15, r1, p16, c0, p13 \end{matrix} \right\rangle \right\}.$$

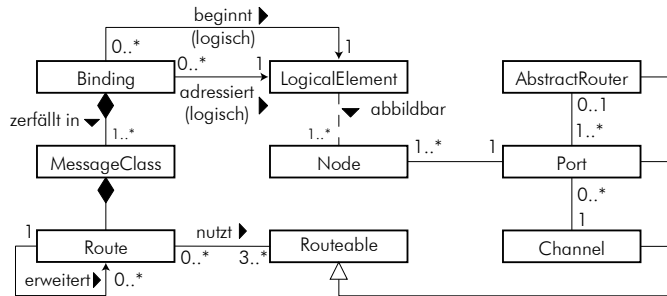


ABBILDUNG 5.11. UML-Modell des zusammengeführten Systemmodells.

Damit können die Wege und Nachrichten- und Informationsgrößen für die Nachrichtenklassen bestimmt werden. Alle notwendigen Attribute der Klassen in Aufzählung 1 können aus der LNS-Datenbank direkt ausgelesen oder mit Hilfe des Protokollstandards rekonstruiert werden. Als zusätzliches Attribut führt die Aufzählung 3 den *arrivalRateModifier* (*Ankunftsratenmodifier*) auf. Die *Ankunftsrate* λ_{app} beschreibt die Anzahl an Nachrichten, die pro Zeiteinheit von der Anwendungsschicht über die Nachrichtenklasse übertragen werden sollen. Sie wird im Verkehrsmodell im nächsten Kapitel für jedes Binding bestimmt. Der *Ankunftsratenmodifier* korrigiert das Verhältnis dieser Ankunftsrate der Anwendungsschicht λ_{app} zu den auf Bitschicht pro Nachrichtenklasse übertragenen Nachrichten λ_{phy} . So ist der Ankunftsratenmodifier zum Beispiel 0, wenn keine Erinnerungen in einem bestätigten Übertragungsdienst notwendig sind; genauso wie er bei einem wiederholten Nachrichtendienst mit r Wiederholungen $r + 1$ ist. Auf die Bestimmung des Modifiers geht der Abschnitt 7.3 ein, während sich das nächste Kapitel dem Netzwerkmodell zuwendet.

AUFZÄHLUNG 3. Auszug aus den Attributen der Klassen aus Abbildung 5.11 (Legende in Aufzählung 1).

MessageClass

- ⊗ messageSize (Nachrichtengröße)
- ⊗ informationSize (Informationsgröße)
- messageType (AckD, Ack, Req, Res, etc.)
- arrivalRateModifier (Ankunftsratenmodifier)

5.5.4. MODELLIERUNGSGRENZEN

Es kann nicht ausgeschlossen werden, dass im Netzwerk neben den bisher modellierten Nachrichtenklassen noch weitere existieren. Das bisher aus Bindings

abgeleitete Modell erfasst nur Nachrichtenklassen, die im Strukturmodell der Gebäudeautomation aus Abschnitt 4.1 als horizontale Kommunikation einzuordnen sind [115]. FISCHER [64] bezeichnet diesen Verkehr als *Prozess-Kommunikation* und unterscheidet zusätzlich die *Management-Kommunikation* (vertikale Kommunikation), die zum Beispiel durch die Überwachung von Sensoren durch das zentrale Prozessleitsystem auf Managementebene entsteht und die *Engineering-Kommunikation* zur Konfigurierung und Inbetriebnahme des Netzwerks. Sowohl die Engineering- als auch teilweise die Management-Kommunikation nutzen das vorgestellte Binding-Konzept nicht und werden nicht als Nachrichtenklasse erfasst. Da die Engineering-Kommunikation allerdings der Konfigurierung und Inbetriebnahme dient, ist sie im Betrieb weitestgehend auszuschließen und muss nicht modelliert werden.

Dies gilt nicht für die Management-Kommunikation. In größeren Systemen der Gebäudeautomation ist davon auszugehen, dass eine Managementebene mit zumindest einer zentralen Prozessleitwarte vorgesehen ist. Da diese häufig mittels Gateways spezielle Managementprotokolle oder Computernetzwerke nutzen (Abschnitt 4.1), werden sie nicht im gleichen Entwurfstool geplant und sind nicht Teil der Entwurfsdatenbank. Demnach darf die Management-Kommunikation nach Prämisse 3 keinen relevanten Einfluss auf das Netzwerkverhalten besitzen oder muss andernfalls im Modell durch entsprechende zusätzliche Funktionsblöcke mit passenden Datenpunkten und Bindings von Hand integriert werden. Damit wird Prämisse 3 ausdrücklich mit der folgenden Prämisse verschärft:

PRÄMISSE 8: FUNKTIONSBLOCKBASIERT

Der Verkehr wird vorwiegend von Funktionsblöcken und ihren Datenpunkten erzeugt, die in dieser Form im Modell modelliert sind.

Mit dem Ankunftsdatenmodifier und dieser Prämisse werden die Einflüsse des Verkehrsmodells auf das Systemmodell der Nachrichtenklassen schon deutlich. Das Verkehrsmodell wird im nächsten Kapitel vorgestellt.

5.6. VALIDIERUNG

Zur Validierung wurden verschiedene LON-Netzwerkentwürfe getestet. Die Tests umfassten zum einen eigene Entwürfe, die gezielt überprüften, ob die Struktur des Systemmodells und die in den Aufzählungen 1 bis 4 aufgeführten Attribute der Instanzen korrekt ausgelesen werden. Zum anderen wurden die bereits vorhandenen Entwürfe getestet, um die Kompatibilität zu realen Entwürfen sicherzustellen. Sie lassen sich fehlerfrei importieren und die entsprechenden Netzwerkmodelle spiegeln den ursprünglichen Netzwerkentwurf wider.

Hierbei werden auch *virtuelle Funktionsblöcke* erkannt und modelliert, die nach Bedarf auf Geräten erstellt werden. Sollen zum Beispiel im Entwurfstool Nachrichten aktiv gepollt werden, so wird im Netzwerkinterface ein solcher virtueller

Funktionsblock mit einem Datenpunkt und einem Binding erstellt. Auf diese Weise wird in LON das Funktionsblockkonzept auch bei zusätzlichen Funktionen konsequent verfolgt.

Dieses Validierungsergebnis ist zu erwarten, da die Struktur der LNS-Datenbank weitestgehend bekannt ist und sich das vorgestellte Datenmodell an den vorhandenen Informationen orientiert. Zusätzlich stellen Entwurfstools [2, 47, 169] selbst gewisse Konsistenzbedingungen des Datenmodells sicher, wie zum Beispiel, das `Bindings` Start- und Zieldatenpunkte haben und diese physisch erreichbar sind. Dadurch handelt es sich bei der automatischen Modellierung des Systemmodells vor allem um eine Aufgabe einer gewissenhaften Implementierung.

5.7. ZUSAMMENFASSUNG

Das vorgestellte Modell stellt den Kern des Datenmodells dar, das zur weiteren Analyse verwendet wird. Die Struktur und die Attribute des Systemmodells sind direkt oder indirekt nahezu vollständig aus der LNS-Datenbank zu exportieren, wie die Aufzählungen 1 bis 3 aufführen. Nur die Attribute `port.bufferSize` und `LonChannel.bitError` sind nicht vorhanden und werden durch Defaultwerte aus Standards [54, 161] und Messungen ergänzt (Unterabschnitt 8.3.5). Der `MessageClass.arrivalRateModifier` wird durch das Tool entsprechend Abschnitt 7.3 bestimmt. Diese gute Exportfähigkeit wird durch eine Validierung an unterschiedlichen Netzwerken hinterlegt.

Das Modell ist in seiner dargestellten Struktur auch für andere Netzwerke der Gebäudeautomation mit abschätzbaren Anpassungen nutzbar. Hierfür sind im Modell bereits die abstrakten Klassen `Routeable`, `PhysicalElement`, `LogicalElement`, `AbstractRouter` und `Channel` als Erweiterungspunkte vorgesehen. Die Tabelle V fasst die wesentlichen Unterschiede und Bestandteile des Modells zusammen.

Zusätzlich besitzt das Modell in seiner Implementation dynamische Erweiterungspunkte, so dass jedes Objekt um zusätzliche Datenobjekte zweckmäßig erweitert werden kann, die als *ExtendedAttribute* bezeichnet werden. Dies wird bei der weiteren Analyse nicht nur genutzt, um Daten, Zwischenergebnisse und Resultate zu ergänzen, sondern auch um Modelle unterschiedlicher Detaillierung austauschbar zu gestalten. Um die Übersicht zu wahren, wird auf die explizite Darstellung der *ExtendedAttributes* und ihrer Verwaltungsobjekte verzichtet, allerdings handelt es sich bei allen in den folgenden Kapiteln definierten Objekten um solche *ExtendedAttributes*.

Damit ist das aufgestellte Systemmodell aus einer Entwurfsdatenbank in seiner dargestellten Struktur vollständig modellierbar. Die aufgeführten Attribute reduzieren sich auf die in der Entwurfsdatenbank vorhandenen Informationen oder können mit Standardwerten und Empfehlungen ergänzt werden. Die Modellie-

5. AUTOMATISCHE MODELLIERUNG DES SYSTEMMODELLS

TABELLE V.
Vergleich der Eignung des Systemmodells für LON, Konnex und BACnet.

	LON	Konnex	BACnet (IPv4)
BITÜBERTRAGUNGSSCHICHT			
Topologie	frei, schleifenfrei	Linie, Baum	frei, schleifenfrei
Anpassungen	-	Anpassung Channel	Anpassung Channel
VERMITTLUNGSSCHICHT			
Hierarchie	dreistufig	dreistufig	z. T. vierstufig (IP)
Stufen	Domain.Subnet.Node	Zone.Linie.Device	verschieden
Adressarten	Unicast, Multicast, Broadcast	Unicast, Multicast, Broadcast	Unicast, Broadcast
Anpassungen	-	Abbildung auf Domain.Subnet.Node	Erweiterung als LogicalElement
ANWENDUNGSSCHICHT			
Funktionsblöcke	ja	ja	ja
Datenpunkte	schreibend / lesend	schreibend & lesend	schreibend & lesend
Verbindungen	1 : n	m : n	1 : 1
Anpassungen	-	Abbildung m : n auf m × (1 : n) Verb.	-
NACHRICHTENKLASSEN			
Nachrichtendienst	unbestätigt, bestätigt, beantwortet, wiederholt, authentifiziert	bestätigt	unbestätigt, bestätigt, beantwortet, segmentiert
Anpassungen	-	Generierung der Nachrichtenklassen	Generierung der Nachrichtenklassen

zung des Systemmodells erfolgt vollständig automatisch nach Anforderung 5, so dass der Anwender nicht gefordert ist. Um das Teilziel 1 zu erfüllen, wird das Verkehrsmodell im nächsten Kapitel ergänzt.

KAPITEL

6.

AUTOMATISCHE MODELLIERUNG DES VERKEHRSMODELLS

*Man kann [...] nicht studieren, ohne
bisweilen die Empfindung zu haben, als wöhne
den mathematischen Formeln selbständiges Leben
und eigener Verstand inne, als seien dieselben
klüger als wir, klüger sogar als ihr Erfinder, als
gäben sie uns mehr heraus, als seinerzeit in sie
hingelegt wurde.*

Heinrich Hertz

6.1. GRUNDLAGEN DER VERKEHRSMODELLIERUNG

Das Verkehrsmodell beschreibt, wie häufig Nachrichten im Netzwerk entstehen. Es gibt verschiedene Ansätze zur Verkehrsmodellierung, die in dieser Arbeit im Abstraktionslevel des Modells, in der Modellierungstiefe des Systems und dem Modellierungszugang unterschieden werden. Der Abstraktionslevel des Modells bestimmt die Detailtreue des Modells, vom exakten Modell bis hin zu stark abstrahierten Modellen. Die Modellierungstiefe bezieht sich auf die Granularität des Modells, also ob das Quellverhalten einzelner Geräte bis hin zu ihrem Anwendungsverhalten modelliert wird oder approximativ das Kanalverhalten. Der Modellierungszugang wiederum legt fest, wie die Modellparameter bestimmt werden, etwa durch Messungen, Anwendungsmodelle oder theoretische Annahmen. Diese Kriterien hängen zusammen und beeinflussen die Eignung eines

Modells für die spätere Leistungsbewertung, wie es in Abbildung 3.1 prinzipiell dargestellt wird.

Referenzmessungen bieten einen direkten Modellierungszugang und können zum Teil ohne weitere Abstraktion eingesetzt werden. So sind im WAN-Bereich die ursprünglichen Nachrichtenquellen für die Betrachtungen häufig irrelevant und Referenzmessungen an Switches und Kanälen werden zur Leistungsbewertung verwendet [45, 137, 174, 240]. Eine der wichtigsten Erkenntnisse ist, dass der Verkehr in diesem Bereich vor allem burstartig¹ ist [45, 137, 174]. Kurzzeitige *Bursts* entstehen zum Beispiel durch die Segmentierung großer Inhalte oder durch die stoßweise Erzeugung vieler Anfragen, wie beim Öffnen einer Internetseite. Überlagern sich viele solcher Burstquellen, so führt dies nicht zu einer Glättung des Verkehrsverhaltens, sondern es entstehen längere Burstperioden mit darin enthaltenden charakteristischen Burstspitzen. Dies führt dazu, dass sich das Burstverhalten unabhängig von der Zeitskalierung ausprägt, die sogenannte *Selbstähnlichkeit* [137, 240].

Es gibt verschiedene Ansätze, diesen Verkehr zu abstrahieren, üblich sind vor allem (Hidden) Markov-Modelle inklusive On/Off-Prozesse, fluide Modelle, autoregressive Modelle und *langzeitabhängige* Erweiterungen² zu den beiden Letztgenannten [1, 68, 142, 173, 233]. Diese komplexen Modelle erlauben allerdings meist keine analytische Leistungsbewertung oder einfachen Simulationen, sondern benötigten umfangreiche Simulationen.

Zu den nicht abstrahierten Messspuren stellen analytische Modelle den Gegenpol in der Abstraktion des Verkehrsmodells dar. Sie besitzen nur wenige charakteristische Parameter und erlauben eine schnelle analytische Leistungsbewertung. Dadurch eignen sie sich besonders für die Untersuchung des prinzipiellen Systemverhaltens in einem großen Parameterraum bei theoretischer Parametrisierung. Eine der einfachsten Modellannahme ist Poisson- beziehungsweise exponentialverteilter Verkehr³ [45, 174]. Sie hat den Vorteil, dass wenn sich unterschiedliche Quellen mit Poisson-Verkehr überlagern, wieder ein Poisson-Verkehr resultiert, dessen Erwartungswert der Ankunftsrate die Summe der Beteiligten ist. Insbesondere im Internet haben sich diese Modellannahmen aufgrund des Burstverhaltens als unrealistisch herausgestellt, werden allerdings aufgrund der Einfachheit und Schnelligkeit häufig zur groben Auslegung verwendet [26, 173].

Im Bereich der Industrieautomation hat JASPERNEITE [105, 106] gezeigt, dass es auch im industriellen Umfeld Burstverhalten gibt. Er untersucht eine einzelne Fertigungszelle mit einer zentralen speicherprogrammierbaren Steuerung (SPS)

-
- 1 Zeitabschnitte mit sehr großem Verkehrsaufkommen wechseln sich mit Abschnitten mit geringem ab.
 - 2 Die fraktalen Eigenschaften der Selbstähnlichkeit und Burstperioden lassen sich nicht mehr gut durch die klassischen Modelle erfassen, die hauptsächlich Kurzzeitabhängigkeiten in der Autokorrelation modellieren.
 - 3 Die Zwischenankunftszeit ist exponentialverteilt und die Ankunftsrate Poisson-verteilt.

und mehreren verteilten Fertigungs- und Bedienstationen. Den burstartigen Verkehr führt er auf den Fertigungsprozess zurück, da die zentrale SPS je nach Fertigungsschritt zeitweilig nur mit einzelnen Fertigungssteuerungen kommuniziert. Er modelliert den Verkehr quellspezifisch durch On/Off-Prozesse.

An diesem Beispiel zeigt sich, dass das Anwendungsverhalten ausschlaggebend für die Verkehrsgenerierung ist. In der Modellierungstiefe gehen Anwendungsmodelle über die Modelle des Quell- und Kanalverhaltens hinaus. SCHROM [199] und TOMURA et al. [213] modellieren das Anwendungsverhalten durch Petrinetze und können so die Interaktion der Geräte abbilden. RÜPING et al. [190] gehen noch weiter und interpretieren direkt den Neuron C Code der Geräte und können mit zusätzlichen Simulatoren für technische Prozesse das ganze Systemverhalten simulieren. Bei diesen äußerst detaillierten Modellen gibt es zwei Schwierigkeiten. Zum einen sind umfangreiche Systemkenntnisse notwendig, vom Quellcode der Geräte über ein deterministisches Hardware-Verhalten bis zu den Prozessmodellen. Diese Informationen gehen weit über das bei der Komposition vorhandene Entwurfswissen hinaus⁴. Als zweite Schwierigkeit räumen RÜPING et al. in [89, 190] die langen Simulationszeiten ein.

In dieser Arbeit wird ein Ansatz verfolgt, der das Anwendungsverhalten und die Geräteinteraktion in reduziertem Maße modelliert und sich an dem vorhandenen Systemmodell der Anwendungsschicht bedient. Gemäß den Anforderungen 1, 3 und 4 wird wieder ein Modell angestrebt, das sowohl eine detaillierte Simulation als auch eine schnelle analytische Lösung erlaubt. Dafür werden in den nächsten Abschnitten sukzessiv Modelle aufsteigender Abstraktion aufgestellt. Mögliche Bursteigenschaften beeinflussen die Genauigkeit des Modells und finden deshalb besondere Erwähnung. Als Modellierungszugang wird in dieser Arbeit ein generischer, theoretischer Ansatz gewählt, bei dem das Modell in Teilen theoretisch vormodelliert und dann mit Hilfe des Systemmodells an den Netzwerkentwurf angepasst wird. In Erweiterung wird in der Diplomarbeit von MARIO GÜRTLER [75] ein automatisierter Messzugang diskutiert. Grundlegende Aspekte der Modellierung entstammen der Zusammenarbeit mit MARIO NEUGEBAUER und BIRGIT GUTSCHE und wurden verschiedentlich in [108, 164, 176, 180] veröffentlicht.

⁴ Dies hat unterschiedliche Gründe: So wird der Quellcode der Geräte nicht offen gelegt; die Hardware ist in ihrem Zeitverhalten nicht-deterministisch (Unterabschnitt 6.3.1) und die Prozessmodelle sind äußerst komplex [108]. In Zukunft könnten semantische Gerätedatenbanken und automatisch modellierte Gebäudesimulationen [187] weitere Informationszugänge schaffen. Das verwendete Modell würde eine automatische Modellzuweisung erlauben.

6.2. MODELL DER GERÄTEINTERAKTION

6.2.1. ALLGEMEINER MODELLIERUNGSANSATZ

Nach Prämisse 8 kann davon ausgegangen werden, dass der Verkehr im Netzwerk vorwiegend von Funktionsblöcken und ihren Datenpunkten erzeugt wird. Das *Gerätemodell* modelliert dieses Verkehrsverhalten auf der Anwendungsschicht.

Um eine Nachricht zu erzeugen, ändert die Anwendung eines Gerätes den Ausgangsdatenpunkt $\mathfrak{o} \in \mathfrak{D}$ und automatisch wird eine Nachricht an den im Systemmodell der Anwendungsschicht festgelegtem Eingangsdatenpunkt $\mathfrak{i} \in \mathfrak{I}$ gesendet. Der Verkehr an den Eingangs- und Ausgangsdatenpunkten kann ohne Einschränkung als diskreter *Ereignisstrom* aufgefasst werden, der durch die Zwischenankunftszeiten $T_{Z;k_i} = t_{k_i} - t_{k_i-1}$ und $T_{Z;k_o} = t_{k_o} - t_{k_o-1}$ zweier aufeinander folgender Ereignisse mit dem spezifischen Index $k_i \in \mathbb{N}$ und $k_o \in \mathbb{N}$ eindeutig ausgedrückt wird. Dies stellt die Modellierungsbasis dar.

Zur Verdeutlichung wird der Fensterkontakt $\mathfrak{d}10$ aus Unterabschnitt 5.1.2 vorgestellt. Der Fensterkontakt hat einen Eingangsdatenpunkt $\mathfrak{i}13$ und einen Ausgangsdatenpunkt $\mathfrak{o}9$, wie in Abbildung 5.8 dargestellt. Der Eingang hat einen Ereignisstrom mit $\mathfrak{T}_{Z;\mathfrak{i}13}$, der durch Nachrichten vom Fensterkontakt $\mathfrak{d}9$ herrührt und einen Ereignisstrom am Ausgang mit $\mathfrak{T}_{Z;\mathfrak{o}9}$. Der Fensterkontakt erzeugt am Ausgang $\mathfrak{o}9$ eine Nachricht, wenn das Fenster geöffnet oder geschlossen wird oder wenn eine Nachricht am Eingangsdatenpunkt $\mathfrak{i}13$ eintrifft, so dass beliebig viele Fensterkontakte in Reihe geschaltet werden können⁵. Der Ereignisstrom am Ausgangsdatenpunkt hat also zwei Ursachen, einerseits einen externen Ereignisstrom durch die Fensteröffnung und zweitens einen am Eingangsdatenpunkt.

Verallgemeinert man dieses Beispiel, so bildet sich der Ereignisstrom, der hier auch als *Ankunftsprozess* bezeichnet wird, aus einem *Quellprozess* und einem *Filterprozess* als Reaktionen auf die Ankunftsereignisse an den Eingangsdatenpunkten. Dieses Verhalten lässt sich unter anderem durch das allgemeine stochastische *Petrinetz* in Abbildung 6.1 ausdrücken, das allerdings nicht die Komplexität von [199, 213] erreicht, sondern alle internen Zustände der Anwendung kapselt. Stattdessen wird das Anwendungsverhalten nur durch die individuelle *Verzögerungszeit* $\tau_{i,o}(k_i)$ der Transition zwischen dem Eingangsereignis k_i und dem Ausgangsereignis k_o ausgedrückt. Die Verzögerungszeit erfasst zwei Fälle: Wenn ein Ausgang nicht auf das Eingangsereignis k_i reagiert, ist $\tau_{i,o}(k_i)$ unendlich⁶. Sonst entspricht die Verzögerungszeit $\tau_{i,o}(k_i)$ der *Bearbeitungszeit* $\tau_{B;i,o} = t_{k_o} - t_{k_i} \geq 0$ der Anwendung, um die Reaktion k_o auf k_i hervorzurufen. Damit ergibt sich die Menge der Ereigniszeitpunkte des Ausganges $\mathfrak{T}_{Z;o}$ aus den um $\tau_{i,o}(k_i)$ verschob-

⁵ Dies entspricht dem Verhalten der Fensterkontakte der Firma Warema [231].

⁶ Die Eingangszustände besitzen unbegrenzte Kapazität, so dass keine Blockade entsteht.

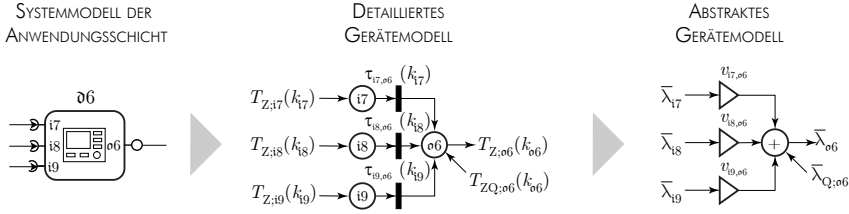


ABBILDUNG 6.1. Gerätemodell.

benen endlichen Eingangszeitpunkten $\mathfrak{T}_{F;i,o} = \{t_{k_i} + \tau_{i,o}(k_i) \mid \tau_{i,o}(k_i) < \infty\}$ der Eingänge $i \in \mathcal{I}_d$ des Gerätes d und den Quellzeitpunkten $\mathfrak{T}_{Q;o}$ mit $k_q \in \mathbb{N}$ zu

$$\mathfrak{T}_{Z;o} = \mathfrak{T}_{Q;o} \cup \bigcup_{i \in \mathcal{I}_d} \mathfrak{T}_{F;i,o}. \quad (6.1)$$

Dieses Modell vernachlässigt den Einfluss von komplexen, internen Zustandsvariablen und Zusammenhängen mehrerer Eingänge. Sendet ein Gerät zum Beispiel nur eine neue Nachricht, nachdem zwei Eingänge aktualisiert wurden, wird das Verhalten im Modell auf das letzte, auslösende Ankunftsereignis reduziert. Dies ist keine drastische Einschränkung, da die Ankunftsereignisse in der Anwendung separat bearbeitet werden (Unterabschnitt 6.3.1) und dadurch solche Abhängigkeiten umständlich zu implementieren sind. Deshalb kann angenommen werden:

PRÄMISSE 9: ENTKOPPLUNG DER BEARBEITUNG

Der Zusammenhang zwischen dem Sendeereignis eines Ausgangsdatenpunktes o_d und dem Ankunftsereignis eines Eingangsdatenpunktes i_d kann isoliert von internen Zuständen und anderer Eingängen betrachtet werden.

Aufgrund des nicht-deterministischen Geräte- und Netzwerkverhaltens werden die Zwischenankunftszeiten $T_{Z;k_i}$, $T_{Z;k_o}$ und $T_{Q;k_q}$ der Ereignisströme sowie die Verzögerungszeiten $\tau_{i,o}(k_i)$ als Realisierungen eines *stochastischen Prozesses* angenommen und durch seine charakteristischen Eigenschaften modelliert. Der Spezialfall einer unendlichen Verzögerungszeit, bei dem kein Übergang von einem Eingangs- zu einem Ausgangsereignis auftritt, wird durch die komplementäre *Übergangswahrscheinlichkeit*

$$v_{i,o} = 1 - \mathbb{P}[\tau_{i,o}(k_i) = \infty] \quad (6.2)$$

separat modelliert.

6.2.2. DETAILLIERTES GERÄTEMODELL

Auf der Basis von Gleichung (6.1) wird das exakte Modell weiter abstrahiert. Im ersten Modellierungsschritt wird davon ausgegangen, dass die untersuchten Zufallsprozesse jeweils *Erneuerungsprozesse* sind und alle Realisierungen statistisch unabhängig und identisch verteilt sind. Dies bedeutet unter anderem, dass die Zwischenankunftszeiten $T_{Z;k_i}$ keine kritische Autokorrelation oder Bursts mit Auswirkungen auf das Netzwerkverhalten besitzen (Unterabschnitt 6.3.2).

PRÄMISSE 10: STATISTISCHE UNABHÄNGIGKEIT

Auf der Anwendungsschicht sind die Zwischenankunftszeiten an den Datenpunkten und die Bearbeitungszeiten statistisch unabhängig und identisch verteilt oder liegen außerhalb der für das Netzwerkverhalten relevanten Grenzen (bei LON etwa 100–200 ms).

Unter dieser Annahme können die Zufallsprozesse durch ihre *Dichtefunktion* (PDF⁷) hinreichend genau modelliert werden. Ferner kann die Dichtefunktion f_C des Zufallsprozesses C , der sich aus der Addition zweier Zufallsvariablen A und B ergibt, als Faltung der Dichtefunktionen f_A und f_B ausgedrückt werden, so dass

$$f_C(t) = (f_A \oplus f_B)(t) = \int_{-\infty}^{\infty} f_A(u) f_B(t-u) \, du. \quad (6.3)$$

Bei der Verarbeitung zwischen Ein- und Ausgang verzerrt die Varianz der Bearbeitungszeit die Zwischenankunftszeiten. In Prämisse 12 wird später festgelegt, dass die Varianz der Bearbeitungszeit im Verhältnis zur Varianz der Zwischenankunftszeit deutlich geringer ist. Dann kann der verzerrende Einfluss der Bearbeitungszeit auf die Verteilung der Zwischenankunftszeit eines Ausgangs vernachlässigt werden und es reicht aus, die Übertragungswahrscheinlichkeit $v_{i,o}$ zu betrachten. Entfallen Zwischenankunftszeiten, so akkumulieren sich die Zeiten bis zur nächsten erfolgreichen Nachricht. Aufgrund der statistischen Unabhängigkeit ist dann

$$f_{T_{Z;i,o}^*}(t) = \sum_{k=1}^{\infty} v_{i,o} (1 - v_{i,o})^{k-1} \bigoplus^k f_{T_{Z;i}}(t). \quad (6.4)$$

Ein Ausgangsprozess an o lässt sich nach Gleichung (6.1) aus dem Einfluss aller Eingänge $i \in \mathcal{I}_o$ mit jeweils $v_{i,o}$ und einem Quellprozess zusammensetzen. Hierbei überlagern sich die Ankunftsdaten der einzelnen Prozesse additiv. Mit Gleichung (6.3) und (6.4) folgt für (6.1) die Darstellung

$$f_{\lambda_o}(\lambda) = f_{\lambda_{Q;o}}(\lambda) \oplus \bigoplus_{i \in \mathcal{I}} f_{\lambda_{i,o}^*}(\lambda). \quad (6.5)$$

⁷ PDF: en. Probability Distribution Function

Womit sich die Dichtefunktion $f_{\lambda_{\circ}}$ der Ankunftsrate eines Ausgangsdatenpunktes \circ zusammensetzt aus der PDF der Ankunftsrate des Quellprozesses $\lambda_{Q;\circ}$ und der verzerrten Dichtefunktionen $f_{\lambda_{i;\circ}^*}$ der Eingänge. Stellt man das Gleichungssystem für alle Ausgangsdatenpunkte auf, so erhält man das *detaillierte Gerätemodell*.

Um den Kreis zu schließen und die Ankunftsrate der Eingangsdatenpunkte zu bestimmen, werden die Bindings aus dem Systemmodell der Anwendungsschicht verwendet. Bei der Übertragung der Nachrichten über das Netzwerk entsteht zwischen \circ und i beim Binding $\mathfrak{b}_{\circ,i}$ eine Verzögerungszeit $\tau_{\mathfrak{b}_{\circ,i}}$. Sie wird im detaillierten Gerätemodell vernachlässigt und nur die Fehlerwahrscheinlichkeit $p_{\mathfrak{b}_{\circ,i}}^-$ für den Verlust der Nachricht auf dem Weg von \circ zu i betrachtet, welche später in Abschnitt 7.3 untersucht wird. Liegen mehrere Bindings \mathfrak{B}_i am Eingang i an, so ist

$$\begin{aligned} f_{T_{Z;\mathfrak{b}_{\circ,i}}^*}(t) &= \sum_{k=1}^{\infty} (1 - p_{\mathfrak{b}_{\circ,i}}^-) (p_{\mathfrak{b}_{\circ,i}}^-)^{k-1} \bigoplus^k f_{T_{Z;\circ}}(t), \\ f_{\lambda_i}(\lambda) &= \bigoplus_{\mathfrak{b}_{\circ,i} \in \mathfrak{B}_i} f_{\lambda_{\mathfrak{b}_{\circ,i}}^*}(\lambda). \end{aligned} \quad (6.6)$$

Hierbei sind Zyklen auf der Anwendungsschicht problematisch. Wird zum Beispiel das Binding \mathfrak{b}_9 auf den Eingang $i12$ gebunden, so erzeugt jede Nachricht, die im Datenpunkt $\circ 9$ generiert wird, im gleichen Datenpunkt wieder eine Nachricht, sofern sie nicht verloren geht. Dadurch entsteht eine bis ins Unendliche wachsende Ankunftsrate. Zyklusfreiheit ist nicht zwingend erforderlich, wenn auf dieser Strecke mehr Nachrichten verloren gehen, als generiert werden, gibt es einen Fix-Punkt. Es gilt deshalb:

PRÄMISSE 11: EINGESCHRÄNKTE ZYKLEN IM VERKEHRSMODELL

Existiert ein Zyklus im Systemmodell der Anwendungsschicht und dem Verkehrsmodell, so muss die durch die Übergangswahrscheinlichkeiten und Fehlerwahrscheinlichkeiten von Bindings im Zyklus abgehende Rate an Nachrichten größer sein, als die von Quellprozessen eingetragene Ankunftsrate.

6.2.3. ABSTRAKTES GERÄTEMODELL

Im zweiten Modellierungsschritt wird die Annahme der Erneuerungsprozesse auf *Poisson-Prozesse* oder exponentialverteilte Prozesse weiter eingeschränkt. Entscheidend ist, dass die Dichtefunktion durch den Erwartungswert hinreichend beschrieben werden kann und diese Eigenschaft bei einer Überlagerung erhalten bleibt. Dann vereinfacht sich Gleichung (6.1) zu

$$\mathbb{E}[\lambda_{\circ}] = \mathbb{E}[\lambda_{Q;\circ}] + \sum_{i \in \mathcal{J}} v_{i,\circ} \mathbb{E}[\lambda_i] \quad (6.7)$$

und für ein Gerät folgt die vollständige Darstellung in Matrixform

$$\vec{\lambda}_{\mathfrak{D}_\circ} = \vec{\lambda}_{Q;\mathfrak{D}_\circ} + \mathbf{V}_\circ \vec{\lambda}_{\mathfrak{T}_\circ} \quad (6.8)$$

mit den Vektoren der Erwartungswerte der Ankunftsrate aller Eingangsdatenpunkte $\vec{\lambda}_{\mathfrak{T}_\circ}$, der Ausgangsdatenpunkte $\vec{\lambda}_\circ$, der zugehörigen Quellprozesse $\vec{\lambda}_{Q;\mathfrak{D}_\circ}$ und der Matrix der Übertragungswahrscheinlichkeiten \mathbf{V}_\circ . Diese Gleichung wird im Folgenden als *abstraktes Gerätemodell* bezeichnet und entspricht der rechten Darstellung in Abbildung 6.1. Auf die explizite Darstellung des Erwartungswerts in der Form $\mathbb{E}[\lambda_\circ]$ wird verzichtet und näherungsweise der Mittelwert $\bar{\lambda}_\circ$ verwendet. Es folgt aus Gleichung (6.6)

$$\bar{\lambda}_i = \sum_{b_{\circ,i} \in \mathfrak{B}_i} (1 - p_{b_{\circ,i}}^-) \cdot \bar{\lambda}_\circ. \quad (6.9)$$

Basierend auf dem abstrakten Gerätemodell wurden in [180] drei Spezialfälle identifiziert. Ein Funktionsblock beziehungsweise ein Datenpunkt, der nur über Quellprozesse verfügt ($\vec{\lambda}_{\mathfrak{D}_\circ} = \vec{\lambda}_{Q;\mathfrak{D}_\circ}$), wird als λ -Quelle bezeichnet ($\mathfrak{D}1, \mathfrak{D}2, \mathfrak{D}5, \mathfrak{D}8, \mathfrak{D}12$ im Beispiel), ein Funktionsblock, der nur Daten empfängt ($\vec{\lambda}_{\mathfrak{D}_\circ} = \emptyset$), als λ -Senke ($\mathfrak{D}3, \mathfrak{D}4, \mathfrak{D}7, \mathfrak{D}11, \mathfrak{D}14$) und ein Funktionsblock, der Daten verarbeitet aber keine eigenen erzeugt ($\vec{\lambda}_{\mathfrak{D}_\circ} = \mathbf{V}_\circ \vec{\lambda}_{\mathfrak{T}_\circ}$), als λ -Filter ($\mathfrak{D}6, \mathfrak{D}13$). Diese Klassifikation wurde in Abbildung 5.8 angedeutet, wobei die Fensterkontakte $\mathfrak{D}9$ und $\mathfrak{D}10$ einen Mischtypen zwischen λ -Quelle und λ -Filter darstellen. Die Klassifikation λ -Quelle, λ -Filter, λ -Senke deckt sich in vielen Fällen mit der in der Automation üblichen Prozesskette von Sensor, Controller, Aktor und wird in dieser Arbeit zur Vereinfachung weiterverwendet, ohne dass darin eine Beschränkung der Gültigkeit der Ergebnisse liegt.

6.3. ZEITVERHALTEN

6.3.1. BEARBEITUNGSZEIT DER ANWENDUNGSSCHICHT

Die Bearbeitungszeit der Anwendung ist aufgrund der komplexen, nicht-deterministischen Prozessorarchitekturen der Geräte nicht genau abzuschätzen. Dies wird am Aufbau eines Neuron-Chips erklärt, der in LON-Netzwerken am meisten vertreten ist. Als Beispiel dient die Bearbeitungszeit zwischen dem Eintreffen einer Nachricht für den Eingangsdatenpunkt $i5$ und dem Senden der Nachricht vom Ausgangsdatenpunkt $o4$ des Lichtreglers auf dem Gerät $\mathfrak{D}6$. Dies entspricht den Zeitpunkten t_{R2} und t_{S2} in Abbildung 6.2, die einen Überblick über das Zeitverhalten gibt.

Der Neuron-Chip besitzt drei Prozessoren (CPU⁸), die sich die sieben Schichten des OSI-Referenzmodells aus Abbildung 4.2 aufteilen. Eine korrekt eintreffende

⁸ CPU: en. Central Processing Unit

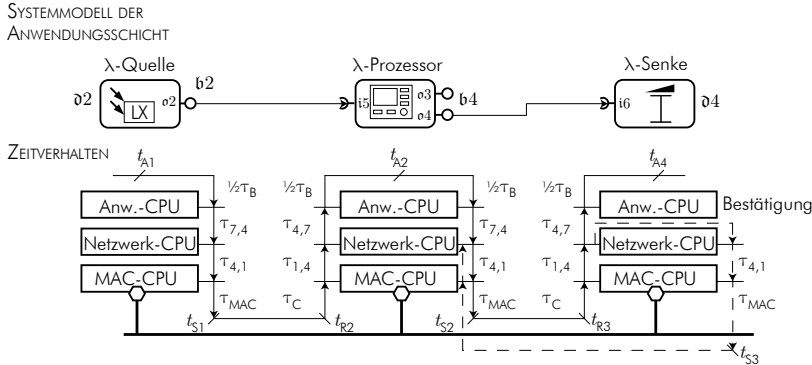


ABBILDUNG 6.2. Zeitverhalten bei der Gerätekommunikation.

Nachricht für i5 durchläuft bis zur Bearbeitung in der Anwendung alle sieben Schichten und die drei CPUs. Um mögliche Leistungsengpässe abzufangen, tauschen die CPUs die Nachrichten über Warteschlangen aus. Die Bearbeitungszeiten pro Nachricht hängen bei der MAC-CPU⁹ und Netzwerk-CPU¹⁰ hauptsächlich von der zu verarbeitenden Nachrichtengröße ab. Die Bearbeitungszeiten der Anwendungs-CPU¹¹ besitzen eine größere Varianz, da die Anwendung zyklisch, ereignisorientiert und interruptfrei arbeitet. Dadurch wird beim Eintreffen einer neuen Nachricht der laufende Zyklus erst zu Ende bearbeitet, bevor das neue Ereignis angegangen wird. Dabei werden in jedem Zyklus alle Ereignisbedingungen in einer festgelegten Reihenfolge getestet, wozu pro Bedingung in etwa 1 ms benötigt wird [161]. Jede durchlaufene CPU und Schicht trägt etwas zu der gesamten Verzögerungszeit bei, wobei die im Verkehrsmodell verwendete Bearbeitungszeit τ_B nur einen Teil ausmacht.

Da der Neuron-Chip die CPUs in einem Chip integriert, ist es schwierig, die einzelnen CPU-Zeiten separat zu bestimmen. Aus diesem Grunde werden nicht die CPU-Zeiten modelliert, sondern es werden die Verzögerungszeiten zum MAC-Zugriff τ_{MAC} , zur Adressierung $\tau_{1,4}$, der Vorverarbeitung $\tau_{4,7}$ und Bearbeitung im Anwendungsprozessor τ_B modelliert. Die gesamte Verzögerungszeit vom Eintreffen der Nachricht t_{s1} bis zum Senden der Reaktion t_{R2} setzt sich demzufolge zusammen aus

$$t_{S2} - t_{R2} = \tau_{1,4} + \tau_{4,7} + \tau_B + \tau_{7,4} + \tau_{4,1} + \tau_{MAC}. \quad (6.10)$$

9 Bitübertragungs- und Sicherungsschicht

10 Vermittlungs- bis Darstellungsschicht

11 Anwendungsschicht

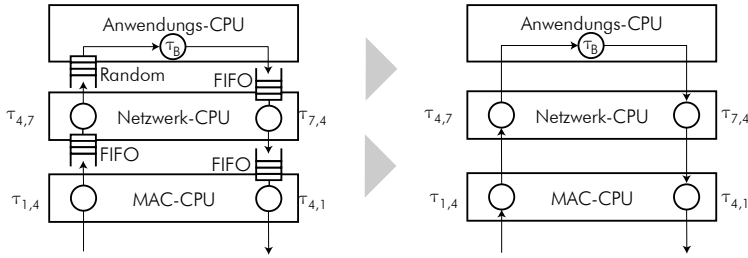


ABBILDUNG 6.3. Zeitverhalten eines Gerätes.

Diese von der Prozessorarchitektur abweichende Einteilung der Zeiten erlaubt es, dass wichtige Dienste der unteren Schichten, direkt modelliert werden können. So dauert das Routing einer Nachricht oder das Erzeugen einer Bestätigung

$$t_{S3} - t_{R3} = \tau_{1,4} + \tau_{4,1} + \tau_{MAC}. \quad (6.11)$$

Die Abbildung 6.3 zeigt links, wie diese Neuron-Chip-Architektur ohne Buszugriffszeiten als ein Warteschlangenmodell mit einer FIFO-Behandlung¹² und zufälligem Bearbeitungsmodus modelliert werden kann¹³ [112, 120, 214]. Zur Modellierung wird das vereinfachte Modell auf der rechten Seite ohne Warteschlangen genutzt, da dann die Verzögerungszeiten nicht durch Wartezeiten verfälscht werden. Hierbei wird vorausgesetzt, dass die betrachteten Warteschlangen bis auf den modellierten Medienzugriff nicht ausgelastet sind, also die einzelnen Bearbeitungszeiten $\tau_{*,\delta}$ deutlich kleiner als die Zwischenankunftszeiten $T_{Z;,\delta}$ sind, so dass gilt:

PRÄMISSE 12: BEARBEITUNGSZEITEN

Die Bearbeitungszeiten der einzelnen Schichten $\tau_{*,\delta}$ eines Gerätes sind, ausgenommen vom Medienzugriff, in Erwartungswert und Varianz deutlich kleiner als die Zwischenankunftszeiten aller ankommenden Nachrichten, womit gilt $\mathbb{E}[\tau_{*,\delta}] \ll \mathbb{E}[T_{Z;,\delta}] \wedge \text{Var}[\tau_{*,\delta}] \ll \text{Var}[T_{Z;,\delta}]$.

Mit dieser Prämisse können die einzelnen Bearbeitungszeiten der Schichten $\tau_{*,\delta}$ direkt aus Messungen abgeschätzt werden, ohne den Einfluss der Warteschlangen zu beachten. Die Tabelle VI vergleicht Messungen zu den Bearbeitungszeiten von Neuron-Chips. SCHMALEK [195], MOTOROLA [160] und LOY & PRINZ [148] haben jeweils die Verzögerungszeit von dem Hardware-Eingang einer Anwendung zum Hardware-Ausgang einer anderen Anwendung gemessen. Dabei handelt es sich um einfache Anwendungen. Zusätzlich hat SCHMALEK [195] eine komplexe Anwendung im gleichen Aufbau untersucht, wobei sich die gemessene Ver-

¹² FIFO: ein First-In-First-Out.

¹³ Durch ein Prozessorsharing-Modell ist zusätzlich der blockierende, gemeinsame Speicherzugriff und durch separate priorisierte Warteschlange die priorisierte Bearbeitung modellierbar.

TABELLE VI.

Verschiedene Messergebnisse zur Verzögerungszeit eines Neuron-Chips (Bereinigt um τ_{MAC} , bestehend aus der Sendezeit der Telegramme, 7.5 x Beta2 und Präambel; Beta1 entfällt unter Prämisse 12; alle Zeiten $[\tau_*] = \text{ms}$, $[S_{\text{App}}] = \text{Byte}$, $[C_{\text{D}}] = \text{MHz}$).

Abstraktion	τ	C_{D}	S_{App}	$\bar{\tau}$	σ_{τ}	$\tau \mid_{10 \text{ MHz}}$	$\sigma_{\tau} \mid_{10 \text{ MHz}}$
SCHMALEK [195]	$t_{\text{A2}} - t_{\text{A1}}$	5	1	18.6	2.5	9.3	1.25
(komplexe Anwendung)				43.4	9	21.7	4.5
LOY & PRINZ [148]	$t_{\text{A2}} - t_{\text{A1}}$	5	1	21.6	-	10.8	-
(Anteilig)				$\bar{\tau}_{\text{B}} = 7.5$	-	3.8	-
				$\bar{\tau}_{7,1} = 8.1$	-	4	-
				$\bar{\tau}_{1,7} = 6$	-	3	-
FUCHS [69]	$t_{\text{S3}} - t_{\text{S2}}$	5	1	4.4	2.3	2.2	1.16
			95	5.5	2.3	2.4	1.17
(routing)	$\tau_{1,3} + \tau_{3,1}$	10	1	2.4	0.8	2.4	0.8
(routing)	$+ \tau_{\text{MAC}}$		50	3.5	0.8	3.4	0.8
MOTOROLA [160]	$t_{\text{A2}} - t_{\text{A1}}$	10	1	6.6	-	6.6	-
			31	14	-	14	-

zögerungszeit in etwa verdoppelt. Daneben hat FUCHS [69] die Antwortzeiten auf eine Bestätigungsnachricht und die Routingzeiten gemessen. Die Messungen von FUCHS [69] und MOTOROLA [160] zeigen, dass die Bearbeitungszeiten linear von der physischen Nachrichtengröße $S_{\text{phy};i}$ in Bits abhängen. Daneben gibt es eine umgekehrt proportionale, lineare Abhängigkeit vom Prozessortakt C_{D} , so dass eine Verdopplung des Taktes zu einer Halbierung der Bearbeitungszeit führt [69, 189]. Diese Abhängigkeit wurde genutzt um in Tabelle VI die Verzögerungszeiten auf einen 10 MHz-Chip zu normieren.

Da die Messungen unterschiedliche Geräte, Medien und Messpunkte nutzen, erweist sich ein direkter Vergleich als schwierig. So liegen die gemessenen Verzögerungszeiten von MOTOROLA [160] unterhalb der anderen Messungen, besitzen aber eine größere Abhängigkeit von der Nachrichtengröße als bei FUCHS [69]. Geht man davon aus, dass der Sender nach LOY & PRINZ [148] im Verhältnis 3:2 mehr Aufwand hat als der Empfänger, der Einfluss der Nachrichtengröße auf den Sender größer [69, 160] und die Varianz unabhängig von der Nachrichtengröße ist [69], folgen die Abschätzung für die Zeiten in Tabelle VII unter der Verwendung eines linearen Modells mit

$$\bar{\tau}_* = C_{10;\text{D}}(a_{\tau_*} + b_{\tau_*} \cdot S_{\text{app};i}); \quad \sigma_{\tau_*} = C_{10;\text{D}} \cdot c_{\tau_*}; \quad (6.12)$$

und dem Verhältnis des Prozessortaktes zum Normtakt $C_{10;\text{D}}$ mit

$$C_{10;\text{D}} = \frac{10 \text{ MHz}}{C_{\text{D}}}. \quad (6.13)$$

TABELLE VII.

Parameter der Bearbeitungszeiten im Neuron-Chip für Gleichung (6.12) bei einer einfachen $\tau_{B,s}$, normalen $\tau_{B,n}$, und komplexen Anwendung $\tau_{B,c}$.

	$\tau_{7,4}$	$\tau_{4,1}$	$\tau_{1,4}$	$\tau_{4,7}$	$\tau_{B,s}$	$\tau_{B,n}$	$\tau_{B,c}$
a_{τ_*} in ms	2.0	1.6	0.95	1.6	3.8	8	16
b_{τ_*} in ms/bit	$18 E^{-3}$	$2 E^{-3}$	$0.5 E^{-3}$	$10 E^{-3}$	0	0	0
c_{τ_*} in ms	0.5	0.62	0.48	0.45	0.75	1.2	4.4

Mit dieser Abschätzung liegt die Anwendungs-zu-Anwendungs-Verzögerung zweier 10 MHz Chips für eine 1 Byte Nachricht bei 10 ms und damit im Mittel von SCHMALEK [195] und LOY & PRINZ [148] aus Tabelle VI. Die Antwortzeit auf eine Bestätigungsaufforderung wird zu 2.6 ms abgeschätzt und ist damit vergleichbar zu den Messungen von FUCHS [69]. Diese Zeiten werden standardmäßig unter der Annahme einer Anwendung mittlerer Komplexität mit einer normalverteilten Bearbeitungszeit mit $\bar{\tau}_{B,n} = 8$ ms und $\sigma_{\tau_{B,n}} = 1.2$ ms in der weiteren Analyse verwendet.

Diese Messungen wurden an Neuron-Chips der Firma Echelon vorgenommen, der in LON-Netzwerken die meiste Verwendung findet. Zum Vergleich sei RÖSSLER [189] aufgeführt, der die mittleren Anwendungs-zu-Anwendungs-Verzögerungen seiner EIA-709.1-Implementation im Bereich von 162 ms schätzt. Bei diesem deutlichen Unterschied kann bei Verwendung anderer Implementationen ein separates Gerätemodell nur nahe gelegt werden. Genaue Modelle zum Zeitverhalten der Anwendung einzelner Geräte können entsprechend Abschnitt 6.6 in der systemeigenen Datenbank ergänzt werden. GÜRTLER [75] diskutiert entsprechende Methoden, um sie aus Messungen von Protokollspuren zu identifizieren.

6.3.2. BURSTS

Um die Eignung des Modells innerhalb der Anforderung 3 zu beurteilen, muss die Burstsituation im Netzwerk diskutiert werden. Wie eingangs erwähnt, entsteht bei der Überlagerung von Burstquellen kein sich ausmittelnder, gleichmäßiger Ankunftsprozess, sondern die Bursteigenschaft bleibt erhalten und resultiert in Selbstähnlichkeit und Langzeitabhängigkeit [173, 240].

Liegen die niedrigeren Zwischenankunftszeiten bei einem Burst im Bereich der Bearbeitungszeiten der Geräte, so füllen sich die Warteschlangen der Geräte in kurzer Zeit. Als Folge entstehen größere Wartezeiten und unter Umständen sogar Nachrichtenverluste bei den Nachrichten, die nach dem letzten Abschnitt nur beim MACs modelliert werden. Dabei zeigen RYU & ELWALID [193] und GROSSGLAUSER & BOLOT [74], das für das Verhalten vor allem die Kurzzeit-Autokorrelation relevant ist und der relevante *Korrelationshorizont* in starkem

Zusammenhang zur Warteschlangenlänge (`port.bufferSize`) steht. Eine Abschätzung des Korrelationshorizonts nach GROSSGLAUSER & BOLOT [74] ist schwierig, da außerdem die Verteilung der Zwischenankunftszeiten und die von der Netzwerkauslastung abhängigen Bearbeitungszeiten eine Rolle spielen. Allerdings liegt die Warteschlangenlänge bei LON unter Zehn [54, 161]. Daraus sind die für das Netzwerkverhalten relevanten Grenzen der Zwischenankunftszeiten aus Prämisse 10 für ein gering bis mittel ausgelastetes Netzwerk, bei dem Bursts das Netzwerkverhalten besonders stark beeinflussen, auf ungefähr 100–200 ms abzuschätzen. Zwischenankunftszeiten, die in diesem Bereich liegen, müssen die Prämisse 10 nach statistischer Unabhängigkeit erfüllen.

Bursts in diesem Zeitbereich können auf der Anwendungsschicht entstehen oder erst in den unteren Schichten. In den unteren Schichten liegen die Ursachen meist bei den Übertragungsdiensten, da als weiterer Grund die Segmentierung von Nachrichten nur selten unterstützt wird (Kapitel 4). So entstehen kurzzeitige Bursts beim *wiederholten Übertragungsdienst*, da die Nachrichten mehrmals hintereinander gesendet werden. Ein *bestätigter Übertragungsdienst* an eine große Gruppe von Empfänger führt außerdem zu einer großen Anzahl an Bestätigungen, die ungefähr gleichzeitig auf den Kanal zugreifen. Für das Verkehrsmodell ist dieses Burstverhalten nicht relevant, da es die Anwendungsschicht modelliert.

Auch auf der Anwendungsschicht können Bursts durch reine Burstprozesse oder parallele, gekoppelte Prozesse entstehen. Ein Burstprozess eines Gerätes auf der Anwendungsschicht entsteht im Beispiel von JASPERNEITE [105] durch einen zyklischen Steuerungsprozess, der periodisch mit einem Gerät intensiv Daten austauscht. Solche zentralistisch organisierten Prozesses-Steuerungen sind in der Gebäudeautomation selten und das Netzwerk wird durch verteilte Prozesse dominiert. In den nächsten Abschnitten wird gezeigt, dass diese selten einen relevanten Burstcharakter haben.

Gekoppelte, parallele Prozesse entstehen, wenn ein Binding an eine Gruppe von λ -Filtern sendet und diese an ihren Ausgängen nahezu gleichzeitig Antworten senden oder eine eingehende Nachricht eine Reaktionen an mehreren Ausgängen erzeugt. Im Beispiel liegt eine solche Prozesskopplung am Gerät $\mathfrak{d6}$ an den Ausgängen $\mathfrak{o4}$ und $\mathfrak{o5}$ vor, die beide im Mittel nach $\bar{\tau}_B$ Nachrichten erzeugen, wenn eine Nachricht am Eingang eintrifft. Die Abbildung 6.4(a) zeigt die Wirkung der Prozesskopplung im Vergleich zum ungekoppelten Prozess. Durch die Kopplung entstehen mehr kurze Zwischenankunftszeiten um $\bar{\tau}_B$, die durch längere Zwischenankunftszeiten wieder ausgeglichen werden, so dass der Mittelwert beider Verteilungen gleich ist. Damit sind im abstrakten Gerätemodell die berechneten Erwartungswerte zwar fehlerfrei, allerdings sind Rückschlüsse auf das detaillierte Gerätemodell immer fehlerbehaftet.

Die Abbildung 6.4(b) bildet die Autokorrelationsfunktion der Zwischenankunftszeiten des Gerätes $\mathfrak{d6}$ ab. Die aufgetragenen Korrelationskoeffizienten drücken eine positive Korrelation mit 1, eine negative mit -1 und keine Korrelation mit 0

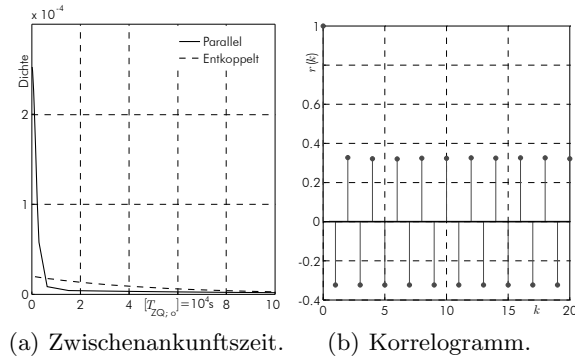


ABBILDUNG 6.4. Entwicklung der Zwischenankunftszeiten zweier gekoppelter Ausgänge bei einer mittleren Bearbeitungszeit $\bar{\tau}_B = 8$ ms.

aus. Sie sollten für den Lag $k > 0$ den abgeschätzten 95%igen Konfidenzintervall nicht verlassen [10]. Dieser liegt nah bei Null und wird deutlich überschritten, so dass Prämisse 10 verletzt wird. Die Auswirkungen auf das Geräteverhalten sind bei zwei gekoppelten Prozessen noch relativ gering, steigen aber mit der Anzahl paralleler Prozesse. Diese Prozesskopplung kann in diesem Verkehrsmodell nicht erfasst werden¹⁴.

6.4. QUELLPROZESSE

6.4.1. METHODEN ZUR ABTASTUNG

Quellprozesse bilden im Gerätemodell (6.8) den Anteil am generierten Verkehr, der in keinem Kausalzusammenhang zu einem Ankunftsprozess an einem Eingangsdatenpunkt steht. Er wird durch Zwischenankunftszeit $T_{Q;\sigma}$ beschrieben.

Quellprozesse entstehen in der Gebäudeautomation hauptsächlich in Sensoren oder Eingabegeräten, die Daten aus der Umwelt aufnehmen. Eine weitere Ursache können zyklische Prozesse sein, wie zum Beispiel ein PID-Regler der regelmäßig seinen Integralteil nachführen muss. Im Beispiel handelt es sich bei allen Sensoren (λ -Quellen) und den Reglerausgänge $\sigma 3, \sigma 4, \sigma 6$ um Quellprozesse. In beiden Fällen werden äußere oder innere Signale überwacht, wie zum Beispiel bei Sensoren oder Reglern. Die Signale werden in dieser Arbeit eingeteilt in *kontinuierliche Signale* ($\sigma 1, \sigma 2, \sigma 5, \sigma 12$) und in *Zustandssignale* ($\sigma 8, \sigma 9, \sigma 10$). Zustandssignale besitzen eine zählbare Menge an möglichen Zuständen, wie beispielsweise $\{0, 1\}$ (binär),

¹⁴ Allerdings werden Prozesskopplungen in der Simulation in Kapitel 7 mit betrachtet.

$\{\text{Ein, Aus, Fehler}\}$, $\{\text{Jan, \dots, Dez}\}$. Die Nummerierung der Zustände kann als Wertediskretisierung aufgefasst werden, wodurch ein darstellbarer, zeitkontinuierlicher Zustandsverlauf $g_o(t) \in \mathbb{N}$ entsteht.

Zur Verarbeitung werden diese Signale zeit- und wertediskretisiert. Die Basis der Zeitdiskretisierung ist die Zykluszeit der Anwendung, die unter Prämisse 12 vergleichbar ist mit der Bearbeitungszeit der Anwendung $\tau_{B;o}$. Um hierbei einen Parametrierungsfehler auszuschließen wird angenommen, dass:

PRÄMISSE 13: ZYKLUSZEIT ERFÜLLT ABTASTTHEOREM

Die Zykluszeit ist ausreichend klein, um ein rekonstruierbares Abbild des Signals zu liefern. Dies bedeutet bei kontinuierlichen Signalen, dass die Zykluszeit das Abtasttheorem erfüllt [184, 204] und bei Zustandssignalen, dass die Zyklusfrequenz mehr als doppelt so groß ist wie die maximale Änderungsfrequenz der Zustände.

Nicht in jedem Zyklus wird eine neue Nachricht mit dem aktuellen Signalwert generiert, sondern dies geschieht unter zwei Paradigmen. Das erste Paradigma ist die *zeitorientierte Nachrichtengenerierung*, bei dem in einem regelmäßigen Abstand $T_{A;o}$ eine neue Nachricht generiert wird. Die zeitorientierte Nachrichtengenerierung empfiehlt sich immer dann, wenn ein aktueller Signalwert zwingend erforderlich ist. Dies ist in der Gebäudeautomation vor allem bei Regelungsprozessen kontinuierlicher Prozesse üblich. In Unterabschnitt 6.4.2 werden die Quellprozesse mit zeitorientierter Nachrichtengenerierung näher untersucht.

Allerdings werden bei der zeitorientierten Nachrichtengenerierung auch dann Nachrichten erzeugt, wenn sich der Signalwert nicht signifikant ändert. Um den unnötigen Verkehr zu sparen, werden Nachrichten in der Gebäudeautomation nach [109, 181] häufiger ereignisorientiert generiert. Unter die *ereignisorientierte Nachrichtengenerierung* fallen alle Erzeugungsprozesse, die den abgetasteten Signalwert in seinem Neuigkeitswert beurteilen und danach über eine Versendung entscheiden. Bei Zustandssignalen wird beispielsweise nur dann eine Nachricht generiert, wenn sich der Zustand geändert hat. Der resultierende Quellprozess wird in Unterabschnitt 6.4.3 behandelt.

Bei kontinuierlichen Signalen werden Nachrichten in ähnlicher Weise bei einer signifikanten Änderung zum letzten Wert gesendet. Die in der Gebäudeautomation meistverbreitete Methode ist die *Send-On-Delta Nachrichtengenerierung*, die von der LonMark für Sensoren empfohlen wird [147]. MISKOWICZ [156] schlägt eine alternative Methode vor, die das Fehlerintegral zum gehaltenen letzten Wert bildet und bei Überschreitung eines Grenzwerts sendet. Dieses Verfahren besitzt den Vorteil, das Diskretisierungsfehler die beim Send-On-Delta entstehen, vermieden werden und das Fehlerquadrat konstant ist. Das ist insbesondere in PID-Reglern für den Integralteil wichtig. Da diese Methode allerdings keine Verbreitung hat, wird sie in dieser Arbeit nicht weiter untersucht, sondern der Send-On-Delta-Algorithmus in Unterabschnitt 6.4.4 fokussiert. Auch die Alarmgene-

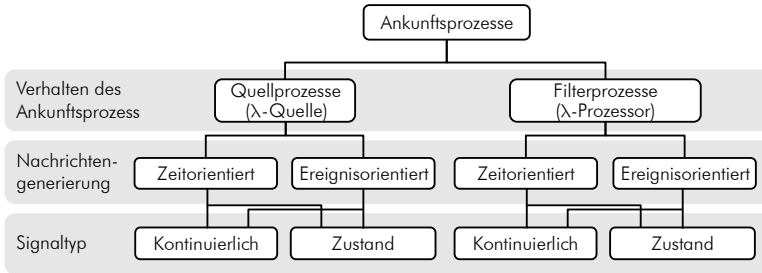


ABBILDUNG 6.5. Klassifikation der Ankunftsprozesse.

rierung durch Schwellwertüberwachung am Rauchmelder $\varnothing 12$ fällt unter die ereignisorientierte Nachrichtengenerierung und wird in Unterabschnitt 6.4.5 näher behandelt.

Bei der ereignisorientierten Nachrichtengenerierung werden die realisierten Zwischenankunftszeiten häufig begrenzt. Zum einen kann dies durch eine sogenannte *Max-Send-Time* $T_{U;\varnothing}$ geschehen, die den maximalen Abstand zweier Telegramme definiert und damit dem Empfänger ermöglicht bei Nichterhalt der Nachricht in dieser Zeit in einen sicheren Zustand zu wechseln. Zum anderen definiert die *Min-Send-Time* $T_{L;\varnothing}$ den minimalen Abstand zweier Nachrichten und verhindert eine Überlastung des Kanals bei starken Änderungen. Der Einfluss der Parameter auf die Quellprozesse wird in Unterabschnitt 6.4.6 untersucht.

Die Abbildung 6.5 stellt die vollständige Klassifikation der Ankunftsprozess dar. Die Klassifikation erfolgt auf den Ebenen des Verhaltens von Ankunftsprozessen, der Nachrichtengenerierung und des Signaltyps. Die weiteren Abschnitte wenden sich den einzelnen Elementen zu und modellieren das Quellverhalten für das detaillierte und abstrakte Gerätemodell.

6.4.2. ZEITORIENTIERTE NACHRICHTENGENERIERUNG

Die zeitorientierte Nachrichtengenerierung entspricht einer zyklischen, nahezu äquidistanten Abtastung mit einer erhöhten Abtastzeit $T_{A;\varnothing}$. Der Quellprozess auf Anwendungsschicht besitzt prinzipiell die gleiche Zwischenankunftszeit. Allerdings wird die Abtastzeit durch einen internen Timer realisiert, der bei einem Neuron-Chip eine nicht unbeachtliche Toleranz von 12 ms pro Sekunde und eine Mittelwertverschiebung besitzt, die für kleine Abtastzeiten positiv und für große Abtastzeiten im Sekundenbereich negativ ist [49]. Ferner wird nach Ablauf von $T_{A;\varnothing}$ erst im nächsten Anwendungszyklus ein Sendeereignis auslöst. Dadurch kann die Nachrichtengenerierung nur als n -faches der zufällig verteilten Zykluszeit $\tau_{B;\varnothing}$ realisiert werden.

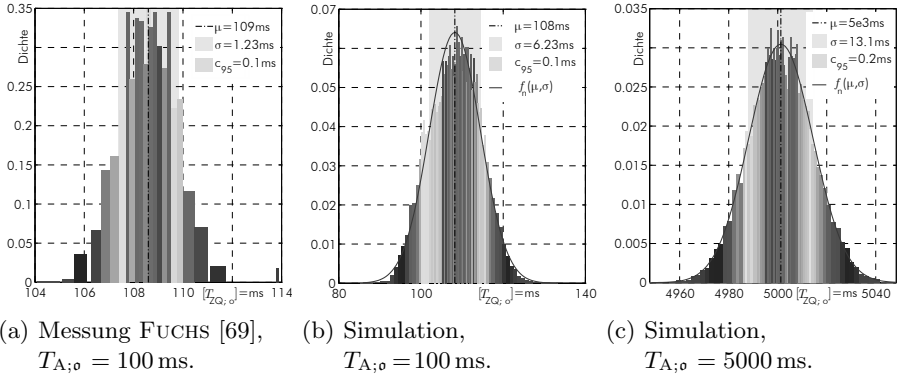


ABBILDUNG 6.6. Histogramme der Zwischenankunftszeit bei zyklischer Nachrichtengenerierung mit wiederholtem Timer mit $\bar{\tau}_{B;d} = 16 \text{ ms}$ und $\sigma_{\tau_{B;d}} = 2.4 \text{ ms}$. Der hellgrau hinterlegte Bereich zeigt die Standardabweichung σ und der grau hinterlegte den 95%igen Konfidenzintervall c_{95} an. $f_n(\mu, \sigma)$ ist die Normalverteilung nach Gleichung (6.15).

Dies hat je nach der Implementation des Timers unterschiedliche Auswirkungen. Beim Neuron-Chip gibt es zwei Arten. Die Abbildung 6.6(a) zeigt das Verhalten des *wiederholten Timers*, das von FUCHS [69] an einem Last-Generator [111] gemessen wurde. Diese Implementation startet sich sofort nach Ablauf neu. Die Bearbeitung erfolgt im nächsten Anwendungszyklus, der unabhängig vom Timer im Mittel mit $\tau_{B;d}/2$ erreicht wird und in $(0, \tau_{B;d})$ gleichverteilt ist. Entsprechend sind die realisierten Zwischenankunftszeiten im Mittel um 8 ms größer als die eingestellten Abtastzeiten, womit sich die Bearbeitungszeit von 16 ms für den 5 MHz-Chip nach Tabelle VII bestätigt. Unter Prämisse 12 entsteht dadurch die in Abbildung 6.6(a) sichtbar glatte Verteilung der Zwischenankunftszeiten, die primär durch Varianz des realisierten Timers $T_{Q;o}^*$ bestimmt wird. Die Toleranzen und die Mittelwertverschiebung des Timers sind für einen Millisekundentimer und Sekundentimer in LON unterschiedlich [49]. Unter der Annahme, dass die Abweichungen normalverteilt sind und die Toleranzen bei 4σ ($\approx 99\%$) liegen, folgt aus [49] für beide

$$\begin{aligned} \bar{T}_{Q;o}^* = T_{A;o}^* = \text{tmr}(T_{A;o}) &= \frac{0.8192}{1000} \left[\left\lfloor \frac{(1000 T_{A;o}) \bmod 1000}{C_d} \right\rfloor \frac{C_d}{0.82} + 1 \right] \\ &+ \frac{0.8192}{1000} \left[\left\lfloor \frac{1001}{C_d} \right\rfloor \frac{C_d}{0.82} + 1 \right] \cdot [T_{A;o}] + \frac{1}{2} \tau_{B;d} \quad (6.14) \end{aligned}$$

$$\sigma_{T_{Q;o}^*}^2 = \sigma_{T_{A;o}^*}^2 = \text{tms}(T_{A;o}) = \left[\frac{1}{4000} [T_{A;o}] (11 \cdot C_d + 1) \right]^2 + \frac{\tau_{B;d}^2}{12} + \sigma_{\tau_{B;d}}^2$$

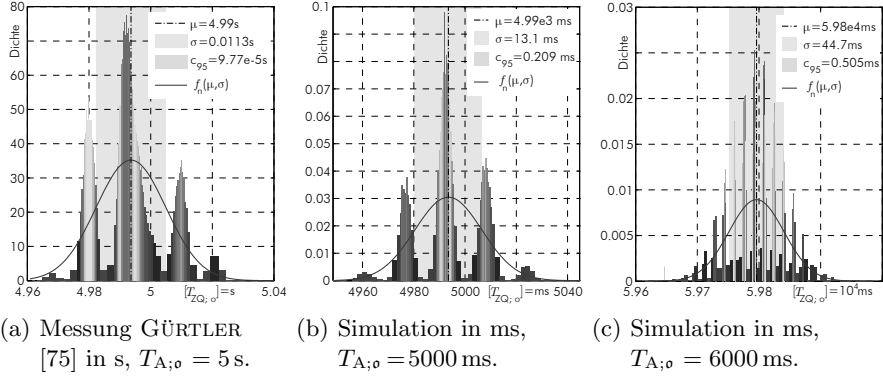


ABBILDUNG 6.7. Histogramme der Zwischenankunftszeit bei zyklischer Nachrichtengenerierung mit einmaligem Timer mit $\bar{\tau}_{B;o} = 16$ ms und $\sigma_{\tau_{B;o}} = 2.4$ ms.

die Normalverteilung als detailliertes Gerätemodell

$$\begin{aligned}
 \mathbf{f}_{T_{Q;o}}(t) &= \mathbf{f}_{\text{tmr}}(t \mid T_{A;o}) \\
 &= \mathbf{f}_n \left(t \mid T_{A;o}^*, \sigma_{T_{A;o}^*} \right) = \frac{1}{\sigma_{T_{A;o}^*} \sqrt{2\pi}} e^{-\frac{1}{2} \left(\frac{t - T_{A;o}^*}{\sigma_{T_{A;o}^*}} \right)^2}. \quad (6.15)
 \end{aligned}$$

Die zweite Implementierungsvariante ist der *einmalige Timer*, dessen Verhalten in Abbildung 6.7(a) dargestellt ist und von GÜRTLER [75] gemessen wurde. Im Gegensatz zu Abbildung 6.6(a) prägt sich deutlich eine multimodale Verteilung aus. Sie entsteht, da der Timer erst nach der Bearbeitung neu gestartet wird und dadurch laufend mit dem Bearbeitungszyklus synchronisiert wird. Entsprechend zeigt Abbildung 6.7(a) sehr deutlich die einzelnen Anwendungszyklen, die auch bei großen Abtastzeiten in Abbildung 6.7(c) nicht verschwinden. Die Anwendungszyklen liegen in etwa 16 ms auseinander, womit sich noch einmal die Abschätzung der Bearbeitungszeit nach Tabelle VII für den 5 MHz-Chip bestätigt. Die Verteilung des realisierten Timers $T_{Q;o}^*$ bildet die Hüllkurve der Reihe der Verteilungen der Anwendungszyklen. Das zeigt sich gut im Vergleich der in Abbildung 6.7(b) und Abbildung 6.6(c) eingezeichneten Normalverteilungen. Zur Modellierung werden unter Prämisse 12 ausschließlich die Anwendungszyklen betrachtet, die im Toleranzbereich der Abtastzeit von $4\sigma_{T_{A;o}^*}$ liegen oder sich innerhalb $4\sigma_{\tau_{B;o}}$ überschneiden. Sie werden als multimodale Normalverteilung dargestellt und ergeben

$$\mathbf{f}_{T_{Q;o}}(t) = \mathbf{f}_{\text{tmr}2}(t \mid \bar{T}_{A;o}) = \sum_{k=k_L}^{k_U} w_n(k) \cdot \mathbf{f}_n(t \mid k \cdot \bar{\tau}_{B;o}, \sigma_{\tau_{B;o}}) \quad (6.16)$$

mit

$$k_L = \left\lfloor \frac{T_{A;o}^* - 4\sigma_{T_{A;o}^*} - 4\sigma_{\tau_{B;b}}}{\bar{\tau}_{B;d}} \right\rfloor, \quad k_U = \left\lceil \frac{T_{A;o}^* + 4\sigma_{T_{A;o}^*} + 4\sigma_{\tau_{B;b}}}{\bar{\tau}_{B;d}} \right\rceil,$$

$$w_n(k) = \frac{\mathbf{F}_n \left(l_U(k) \mid T_{A;o}^*, \sigma_{T_{A;o}^*} \right) - \mathbf{F}_n \left(l_L(k) \mid T_{A;o}^*, \sigma_{T_{A;o}^*} \right)}{\mathbf{F}_n(4 \mid 0, 1) - \mathbf{F}_n(-4 \mid 0, 1)},$$

$$l_L(k) = \max \left(\left(k - \frac{1}{2} \right) \bar{\tau}_{B;d}, T_{A;o}^* - 4\sigma_{T_{A;o}^*} \right),$$

$$l_U(k) = \min \left(\left(k + \frac{1}{2} \right) \bar{\tau}_{B;d}, T_{A;o}^* + 4\sigma_{T_{A;o}^*} \right).$$

Für das Beispiel aus Unterabschnitt 5.1.2 sei angenommen, dass der Helligkeitssensor $\mathfrak{d}2$ und die Beleuchtungsregelung in $\mathfrak{d}6$ zeitorientiert Nachrichten generiert und regelmäßig alle 500 ms eine Nachricht erzeugen soll. Bei einer normalen Anwendung mit 5 MHz lässt sich damit die Zwischenankunftszeit abschätzen zu

$$\bar{T}_{Q;o3} = \bar{T}_{Q;o4} = \bar{T}_{Q;o5} \approx \text{tmr}(500 \text{ ms}) = 507.7 \text{ ms}$$

Die Abbildung 6.6(c) zeigt das Histogramm der Simulation für dieses Beispiel mit einem wiederholten Timer. Die Mittelwertverschiebung ist vergleichsweise klein und kann vernachlässigt werden. Weiterhin kann das detaillierte Modell des multimodalen, einmaligen Timers auf einen einfach normalverteilten, wiederholten Timer vereinfacht werden und beide Fälle bis auf eine Punktverteilung mit einem Dirac bei $T_{A;o}^*$.

6.4.3. EREIGNISORIENTIERTE NACHRICHTENGENERIERUNG BEI ZUSTANDSSIGNALLEN

Zustandssignale fassen abzählbare Mengen zusammen. Sie können durch Nummerierung der Zustände als zeitkontinuierlicher Zustandsverlauf $g_o(t)$ über der Zeit dargestellt werden. Durch die Zeitdiskretisierung des Gerätes in jedem Zyklus mit $\tau_{B;d}$ ergibt sich hieraus der zeitdiskrete Zustandsverlauf $g_o(k_{D;o})$. Aus der Annahme, dass Nachrichten nur generiert werden, wenn sich der Zustand ändert, folgt der gesuchte Quellprozess

$$\mathfrak{T}_{Q;o} = \{ t_{k_{D;o}} \mid |g_o(k_{D;o}) - g_o(k_{D;o} - 1)| > 0 \}. \quad (6.17)$$

Mit einer hinreichend kleinen Zykluszeit nach Prämisse 13 kann davon ausgegangen werden, dass die Zeitdiskretisierung verlustfrei ist. Werden also die Zustandsänderungen im Ursprungssignal bestimmt, entspricht sie dem Quellprozess. Im Unterschied zur zeitorientierten Nachrichtengenerierung kann die Verzögerungszeit zwischen Eintreten des Ereignisses in der Umwelt und der Nachrichtengenerierung im Gerät vernachlässigt werden, da das externe Ereignis von der

internen Verzögerungszeit unabhängig ist und zu keiner Mittelwertverschiebung der Zwischenankunftszeit führt.

Die Hauptschwierigkeit besteht darin, das Ursprungssignal zu bestimmen. Zustandssignale sind vielseitig und liegen im Beispiel an den Fensterkontakten ϑ_9 , ϑ_{10} und dem Präsenzmelder ϑ_8 an. Genauso erfassen Schalter und Taster binäre Zustandssignale. Bei allen aufgezählten Beispielen und generell häufig bei binären Zustandssignalen liegen die Ursachen für die Zustandsänderungen im Menschen. Da der Mensch allerdings willkürlich handelt, ist das Verhalten einer Einzelperson nicht aussagekräftig und zur Modellierung eines repräsentativen Signalverlaufs sind umfangreiche Studien notwendig. Insbesondere in der Gebäudesimulation und zur Berechnung der Energieeffizienz werden entsprechende Nutzungsprofile aufgestellt, wobei sich allerdings die Notwendigkeit zeigt, dass die Nutzungsprofile sehr stark differenziert werden müssen [86, 122]. So gibt die VDI 6020 [222] als Beispiel nur ein Nutzungsprofil zur Fensterlüftung an und die [41] sehr unterschiedliche Nutzungsprofile im Zusammenhang mit der Energieeffizienzbestimmung.

In dieser Arbeit soll auf eine detaillierte Analyse dieser Profile verzichtet werden, da die entstehenden Ankunftsdaten im Vergleich zu zeitorientierten Nachrichtenquellen im Allgemeinen vernachlässigbar ist und grobe Abschätzungen zu keinen signifikanten Änderungen der Ergebnisse führen, so dass Anforderung 3 nach Genauigkeit erfüllt bleibt. So sei für das Beispiel aus Unterabschnitt 5.1.2 für die Fensterkontakte ϑ_9 , ϑ_{10} angenommen, dass das Fenster zwei bis dreimal täglich geöffnet und geschlossen wird, dann ändert sich der Zustand im Schnitt täglich fünfmal, wobei genauso viele Nachrichten erzeugt werden. Die Zwischenankunftszeit ist also

$$\bar{T}_{Q;\vartheta_8} = \bar{T}_{Q;\vartheta_9} \approx (5 \text{ Msg/d})^{-1} = 17280 \text{ s} = 4.8 \text{ h.}$$

Der Belegungszustand des Raumes ändert sich ferner im Schnitt zehnmal pro Tag, womit am Präsenzmelder folgt

$$\bar{T}_{Q;\vartheta_7} \approx (10 \text{ Msg/d})^{-1} = 8640 \text{ s} = 2.4 \text{ h.}$$

In beiden Fällen liegt die Ankunftsrate um 4 Größenordnungen unter der Ankunftsrate des Helligkeitssensors ϑ_2 . Für das detaillierte Gerätemodell wird angenommen, dass die Zwischenankunftszeit exponentialverteilt ist, die sich gut zur Beschreibung seltener Ereignisse eignet.

Komplexere Zustandssignale mit mehr als zwei Zuständen sind in der Gebäudeautomation eher selten. Hierzu wurde die Liste der Standard Enumeration Types von LON analysiert [146]. Sie listet die standardisierten *Aufzählungstypen*, die als Indikator für ein ursprüngliches Zustandssignal angesehen werden. Obwohl es an die 70 solcher komplexer Datentypen gibt, handelt es sich hauptsächlich um zustandsorientierte Steuerungsbefehle und entsprechende, spezifische

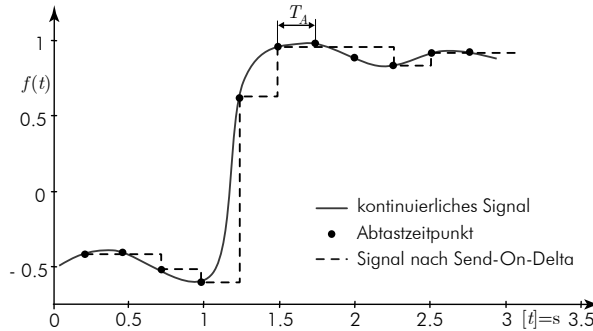


ABBILDUNG 6.8. Signaldigitalisierung und Send-On-Delta an einem beispielhaften Signalverlauf mit einem Delta $\delta_o = 0.1$.

Statusmeldungen der Geräte, so zum Beispiel der Türschloss-, Ventilator- oder Waschmaschinenstatus. Es existieren keine Studien über die Verwendung und Einsatzhäufigkeit dieser Typen. Allerdings ist anzunehmen, dass sie in einer Installation eher einen geringen Teil ausmachen und auch hier gilt, dass sich die Zustände vergleichsweise selten ändern. Damit kann davon ausgegangen werden, dass die statistische Unabhängigkeit nach Prämisse 10 für die ereignisorientierte Nachrichtengenerierung bei Zustandssignalen gegeben ist.

6.4.4. EREIGNISORIENTIERTE NACHRICHTENGENERIERUNG KONTINUIERLICHER SIGNALE MIT SEND-ON-DELTA

6.4.4.1. MODELLIERUNG DES QUELLPROZESSES

Die *Send-On-Delta Nachrichtengenerierung* wird zur bedarfsgerechten Nachrichtenerzeugung bei kontinuierlichen Signalen verwendet. Für das zugrunde liegende Verfahren führt MISKOWICZ [158] verschiedene Bezeichnungen auf wie event-based sampling oder level-crossing sampling. Im Grunde wird eine Nachricht generiert, wenn sich das digitalisierte Signal $f_o(k_{A;o})$ wie in Abbildung 6.8 im Abtastpunkt $t_{k_{A;o}} \in \mathfrak{T}_{A;o}$ seit dem letzten Sendezeitpunkt $t_{k_{Q;o}} \in \mathfrak{T}_{A;o}$ um eine Differenz von mindestens δ_o geändert hat. Hierbei gibt es die Implementierungsvariante, dass nicht in jedem Anwendungszyklus $\tau_{B;\delta}$ abgetastet wird, sondern eine Abtastzeit $T_A \gg \tau_{B;\delta}$ verwendet wird. Dies wird vorerst nicht unterschieden und für $\tau_{B;\delta}$ wird analog T_A verwendet, so dass für den gesuchten Quellprozess $\mathfrak{T}_{Q;o}$ folgt,

$$\mathfrak{T}_{Q;o} = \{t_{k_{Q;o}} = t_{k_{A;o}} \mid |f_o(k_{A;o}) - f_o(k_{Q;o} - 1)| \geq \delta_o\}. \quad (6.18)$$

Sofern $\tau_{B;\delta}$ und $T_{A;\delta}$ im Sinne von Prämisse 13 hinreichend klein sind, ist die Zeitdiskretisierung verlustfrei und das Ursprungssignal kann direkt analysiert werden, wobei Bearbeitungszeiten vernachlässigbar sind.

Die erste Abschätzung betrachtet die Ankunftsrate in jedem Diskretisierungsintervall. Sie wurde mit [180] parallel zu MISKOWICZ [153, 154, 155] veröffentlicht, der in seinen Arbeiten die Effektivität des Send-On-Delta-Algorithmus untersucht. Steigt das Signal linear um eine Einheit pro Diskretisierungsintervall und liegt das Delta bei zwei Einheiten, so entstehen im Mittel anderthalb Nachrichten in jedem Intervall. Dabei wird pro Intervall maximal eine Nachricht generiert. Also ist die mittlere Ankunftsrate in jedem Intervall $k_{A;\delta}$

$$\bar{\lambda}_{Q;\delta}(k_{A;\delta}) = \min\left(\frac{|f'_\delta(k_{A;\delta})|}{\delta_\delta}, 1\right) = \min\left(\left|\frac{f_\delta(k_{A;\delta}) - f_\delta(k_{A;\delta} - 1)}{\delta_\delta (t_{k_{A;\delta}} - t_{k_{A;\delta} - 1})}\right|, 1\right). \quad (6.19)$$

Hierbei handelt es sich um eine Abschätzung, da im Falle eines mit einer kleinen Amplitude rauschenden Signals und einem deutlich größeren Delta die Ankunftsrate zu groß abgeschätzt wird, obwohl selten eine Nachricht generiert wird. Dabei ist die resultierende Ankunftsrate kleiner eins und der abgeschätzte Sendezeitpunkt liegt in einem späteren Intervall. Dieser Extrapolationsfehler mittelt sich bei kleinen Anstiegsschwankungen durch die Überabtastung heraus. Hat der Anstieg allerdings einen Vorzeichenwechsel im nächsten Abtastintervall, so tritt die abgeschätzte Sendeanzahl nie ein.

Diese Problematik wurde umfangreich von MISKOWICZ [153, 154, 155, 158] untersucht. Er grenzt die mittlere Ankunftsrate zwischen dem nach Gleichung (6.19) abgeschätztem Maximum $\bar{\lambda}_{Q;\delta}(k_{A;\delta})$ und dem Minimum $\bar{\lambda}_{Q;\delta}(k_{A;\delta}) - 2 \cdot \bar{e}$ ein, wobei \bar{e} die mittlere Extremwertdichte des Signals ist.

Der Fehler wird in Tabelle IX und XI bewertet und in dieser Arbeit vernachlässigt. In Erweiterung von Prämisse 13 muss äquivalent zum Abtasttheorem gelten, dass Delta mindestens um die Hälfte kleiner ist als die relevante Signalamplitude ($2\delta_\delta < A_f$), damit der Ursprungssignalverlauf verlustfrei wiederhergestellt werden kann. Da der Fehler von dem Verhältnis von δ_δ und T_A zur kleinsten relevanten Signalamplitude und -frequenz abhängt, kann er im Sinne von Prämisse 13 als klein angesehen werden. Mit dieser Aussage schließt auch MISKOWICZ [158] die Modellierung des Quellprozesses ab und untersucht die Effektivität des Send-On-Delta-Algorithmus. Die folgenden weiterführenden Modellierungsschritte nimmt er nicht vor.

Die Analyse eines kontinuierlichen Signalverlaufs $f_\delta(t)$ mit Gleichung (6.19) mit der Abtastzeit $\tau_{B;\delta}$ oder T_A ist genauso aufwendig wie die direkte Anwendung des Send-On-Delta-Algorithmus. Wird eine höhere Schrittweite verwendet¹⁵, steigen der Diskretisierungsfehler und der Fehler der Abschätzung. Um den Aufwand ge-

¹⁵ Dann ist die Ankunftsrate in Gleichung (6.19) allerdings nicht mehr auf 1 Begrenzt.

mäß der Anforderung 1 zu reduzieren, wird deshalb nicht der vollständige Signalverlauf analysiert, sondern die Zwischenankunftszeit wird anhand der Dichtefunktion f_{d_o} des Anstiegsbetrags d_o bestimmt und dabei generalisiert. Hierbei wird ausgenutzt, dass kontinuierliche Signale im Gebäude üblicherweise periodisch sind und eine charakteristische Dynamik haben. So ändert sich die Temperatur in einem Raum über den Tagesverlauf vergleichbar mit den anderen Räumen, auch wenn die einzelnen Raumtemperaturen unterschiedlich sind. Verschiedene solcher charakteristischen Prozesse im Gebäude werden in Unterabschnitt 6.4.4.4 und 6.4.4.5 untersucht. Hieraus dienen für die folgende Modellierung die Temperatur und die Beleuchtungsstärke als Beispiel, da sie eine sehr unterschiedliche Dynamik aufweisen.

6.4.4.2. QUELLPROZESS IM DETAILLIERTEN GERÄTEMODELL

Die Zwischenankunftszeiten des Quellprozesses sind vom Anstieg abhängig. Steigt ein Signal linear um 10 Einheiten pro Sekunde und ist das Delta bei 2 Einheiten, dann werden in einer Sekunde 5 Nachrichten generiert, wohingegen bei einem Anstieg von zwei Einheiten in der gleichen Zeit nur eine Nachricht entsteht. Die Zwischenankunftszeiten sind also umgekehrt proportional zum Anstieg, im Gegensatz zu ihren Häufigkeiten, die proportional sind. Entsprechend ist die Dichte, mit der der Anstieg d_o auftritt, nicht gleich der Dichte von $T_{Q;o} = \delta_o/d_o$, sondern beide Dichtefunktionen stehen in einem nichtlinearen Zusammenhang. Aus dem umgekehrt proportionalen Verhalten der Zwischenankunftszeiten zum Anstieg folgt ein komplementäres Verhalten der *kumulativen Verteilungsfunktionen* (CDF¹⁶). Das heißt mit der Wahrscheinlichkeit $\mathbb{P}[T_{Q;o} \leq T]$, dass die realisierten Zwischenankunftszeiten $T_{Q;o}$ kleiner als T sind, ist die realisierte Signaländerung $d_o \cdot T$ größer als Delta δ_o . Dabei müssen die realisierten Zwischenankunftszeiten noch zusätzlich mit ihrer Häufigkeit gewichtet werden, die proportional zum Anstieg ist. Es resultiert

$$F_{T_{Q;o}}(t) = 1 - \frac{1}{\bar{d}_o} \int_0^{\delta_o/t} \hat{d}_o \cdot f_{d_o}(\hat{d}_o) d\hat{d}_o. \quad (6.20)$$

Hierbei tritt das schon in Unterabschnitt 6.4.4 diskutierte Extrapolationsproblem in verschärfter Form auf, da impliziert wird, dass der Anstieg für mindestens $T_{Q;o}$ anhält, damit sich das abgeschätzte Ankunftsereignis ausprägt. Durch die isolierte Betrachtung der einzelnen Anstiege d_o mittelt sich der Extrapolationsfehler wie in Gleichung (6.19) nicht heraus, sondern kommt bei kleinen Anstiegen voll zum Tragen. Da kontinuierliche Signale allerdings meist stetig sind, ist der Fehler meist gering (Tabelle IX).

¹⁶ CDF: en. Cumulative Distribution Function

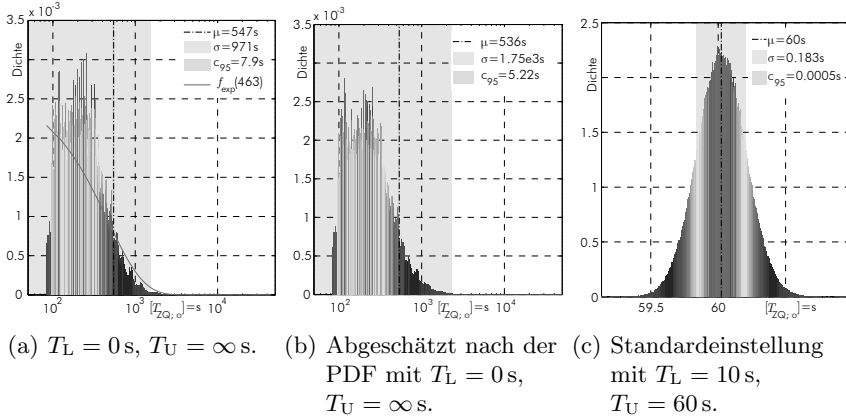


ABBILDUNG 6.9. Histogramme der Zwischenankunftszeiten nach Simulation und Abschätzung des Send-On-Delta-Algorithmus mit $\delta=0.1$ °C auf einer Temperaturspur mit unterschiedlichen Parametern.

Genau genommen können die realisierten Zwischenankunftszeiten nur Vielfache der Abtastzeit $T_{A;o}$ sein. Dadurch prägen sich besonders bei großen Abtastzeiten $T_{A;o} \gg \tau_{B;o}$ der zweiten Implementierungsvariante deutliche multimodale Verteilungen aus (Temperatur (außen) in Abbildung 6.20). Es liegt nahe, dass die Abtastzeiten über einen wiederholten Timer nach Unterabschnitt 6.4.2 realisiert werden und die Realisierungen normalverteilt sind mit dem Erwartungswert $T_{A;o}^*$. Damit setzt sich der Quellprozess aus den einzelnen Verteilungen um $k \cdot T_{A;o}^*$ zusammen und bildet eine multimodale Normalverteilung

$$\mathbf{f}_{T_{Q;o}}(t) = \sum_{k=1}^{\infty} w_{T_A}(k) \cdot \mathbf{f}_n \left(t \mid k \cdot T_{A;o}^*, \sigma_{T_{A;o}^*} \right). \quad (6.21)$$

Der Anteil der einzelnen Verteilungen berechnet sich aus Gleichung (6.20) zu

$$w_{T_A}(k) = \mathbf{F}_{T_{Q;o}}(k \cdot T_{A;o}^*) - \mathbf{F}_{T_{Q;o}}((k-1) \cdot T_{A;o}^*). \quad (6.22)$$

6.4.4.3. QUELLPROZESS IM ABSTRAKTEN GERÄTEMODELL

Die Abbildung 6.11 zeigt, dass der Anstiegsbetrag $d_o \in |f'_o|$ der untersuchten Signale meist exponentialverteilt ist. Dies ist darauf zurückzuführen, dass ein Großteil der hinter den kontinuierlichen Signalen stehenden Prozesse ein ausgeprägtes Verzögerungsverhalten besitzt und mit mehreren Frequenzen oszilliert. Dadurch sind kleine Anstiegsänderungen im Signal typischer als große.

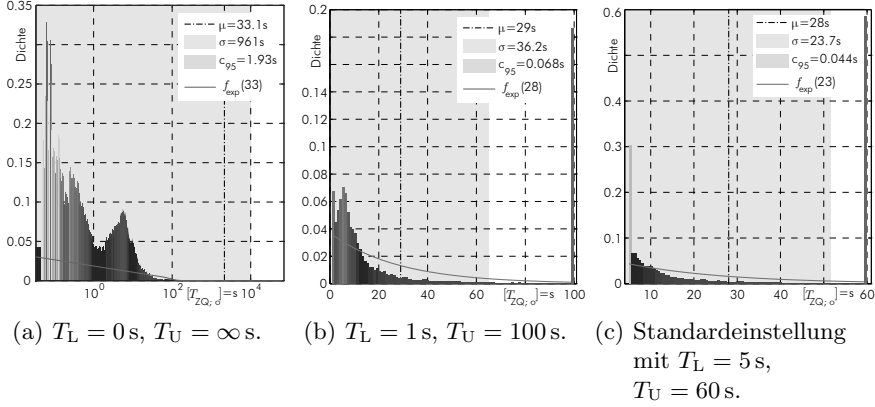


ABBILDUNG 6.10. Histogramme der Zwischenankunftszeiten nach Simulation des Send-On-Delta-Algorithmus mit $\delta=50$ lx auf einer Spur der Beleuchtungsstärke mit unterschiedlichen Parametern.

Aus diesem Grunde wird für f_{d_o} eine exponentielle Verteilung angenommen mit der Dichtefunktion

$$f_{d_o}(d_o) = f_{\text{exp}}(d_o \mid \bar{d}_o) = \begin{cases} 1/\bar{d}_o \cdot e^{-\frac{d_o}{\bar{d}_o}}, & d_o \geq 0; \\ 0, & \text{sonst;} \end{cases} \quad (6.23)$$

der kumulativen Verteilungsfunktion für $d_o \geq 0$

$$F_{d_o}(d_o) = \int_0^{d_o} f_{d_o}(\hat{d}) \, d\hat{d} = 1 - e^{-\frac{d_o}{\bar{d}_o}} \quad (6.24)$$

und dem Erwartungswert \bar{d}_o . Damit folgt aus Gleichung (6.20) für $t \geq 0$

$$F_{T_{\text{soD};0}}(t) = \int_0^t f_{T_{\text{soD};0}}(\hat{t}) \, d\hat{t} = \frac{t \cdot \bar{d}_o + \delta_o}{t \cdot \bar{d}_o} e^{-\frac{\delta_o}{t \cdot \bar{d}_o}} \quad (6.25)$$

mit dem Erwartungswert

$$\bar{T}_{Q;0} = \frac{\bar{d}_o}{\delta_o}. \quad (6.26)$$

Dieser Erwartungswert gilt für beide Implementierungsvarianten, solange $\bar{T}_{Q;0}$ deutlich größer als $T_{A;0}$ ist. Andernfalls berechnet sich aus Gleichung (6.22) für die zweite Variante der Erwartungswert zu

$$\bar{T}_{Q;0} = \sum_{k=1}^{\infty} k \cdot T_{A;0}^* \cdot w_{T_A}(k). \quad (6.27)$$

6.4.4.4. UNTERSUCHUNG CHARAKTERISTISCHER WETTERPROZESSE

Zur Bestimmung der Quellprozesse fehlt die Untersuchung der Dynamik repräsentativer Prozesse. In Unterabschnitt 6.4.3 wurde angesprochen, dass die Gebäudesimulation und die Energieeffizienzbestimmung auch repräsentative Ursprungsdaten benötigen. Zu den in der VDI 6020 [222] für die Simulation empfohlenen Daten gehören die Testreferenzjahre [66]. Das verwendete umfasst den Jahresverlauf von 10 unterschiedlichen Wetterprozessen von 1984 im Gebiet Würzburg mit einem Messwert pro Stunde. Die Messauflösung von einer Stunde ist für einen Großteil der langsamen Wetterprozesse wie Temperatur, Luftdruck oder Himmelsbedeckung ausreichend. Die Beleuchtungsstärke, die als direkte und diffuse Strahlung gemessen wird, ist allerdings deutlich dynamischer, wenn sich zum Beispiel eine Wolke vor die Sonne schiebt. Aus diesem Grunde wurden stattdessen die detaillierteren Verläufe mit 100 Werten pro Tag aus der Simulationssoftware TransSys [119] verwendet. Diese Jahresverläufe wurden in ihrem unterschiedlichen Verhalten untersucht und verglichen und dazu mittels kubischen Splines [29] interpoliert.

TABELLE VIII.
Charakteristika des Anstiegs unterschiedlicher physikalischer Größen
($|f'|_{\min} = 0$, $[f'] = \text{Einheiten/s}$).

Nr.	Größe	Einheit	$ f' _{\text{mean}}$	$ f' _{\text{max}}$	f_{\min}	\bar{f}	f_{\max}
1	Himmelsbedeckung	-	1.94 E^{-5}	1.54 E^{-4}	0.00	63.1	1.00
2	Windrichtung	°	2.01 E^{-2}	1.42 E^{-1}	0.00	1.80 E^2	3.60 E^2
3	Windgeschwindigkeit	m/s	3.20 E^{-4}	3.57 E^{-3}	0.00	2.63	22.4
4	Niederschlag	mm/h	1.67 E^{-5}	2.55 E^{-3}	0.00	8.01 E^{-2}	7.70
5	Luftdruck	hpa	9.10 E^{-5}	7.67 E^{-4}	9.49 E^2	9.85 E^2	1.01 E^3
6	Lufttemperatur außen	°C	1.83 E^{-4}	1.19 E^{-3}	-16.9	8.99	32.6
7	Relative Luftfeuchte	-	8.67 E^{-6}	6.27 E^{-5}	2.40 E^{-1}	7.79 E^{-1}	1.00
8	Direkte Sonnenstrahlung	W/m ²	1.01 E^{-2}	12.8	0.00	62.8	6.85 E^2
9	Diffuse Himmelsstrahlung	W/m ²	5.51 E^{-3}	2.48	0.00	64.9	3.69 E^2
10	Beleuchtungsstärke außen	lx	8.82	1.47 E^3	0.00	1.35 E^4	1.06 E^5
11	Lufttemperatur innen	°C	1.02 E^{-4}	1.09 E^{-2}	17.7	21.0	22.6
12	Beleuchtungsstärke innen	lx	7.36	4.43 E^2	0.00	7.46 E^2	1.30 E^4
13	CO ₂ -Gehalt	ppm	7.81 E^{-2}	4.53 E^{-1}	1.99	8.75 E^2	1.24 E^3
14	Heizungsstellwert	-	1.24 E^{-4}	4.16 E^{-2}	0.00	1.15 E^{-2}	1.00
15	Kühldeckenstellwert	-	2.80 E^{-5}	5.00 E^{-2}	0.00	7.98 E^{-2}	1.00
16	Leuchtenstellwert	-	5.11 E^{-5}	2.75 E^{-2}	0.00	1.37 E^{-1}	6.93 E^{-1}
17	Ventilatorstellwert	-	8.31 E^{-5}	2.16 E^{-3}	0.00	3.17 E^{-1}	1.00
18	Stellwert allgemein	-	7.16 E^{-5}	3.03 E^{-2}	0.00	1.36 E^{-1}	1.00

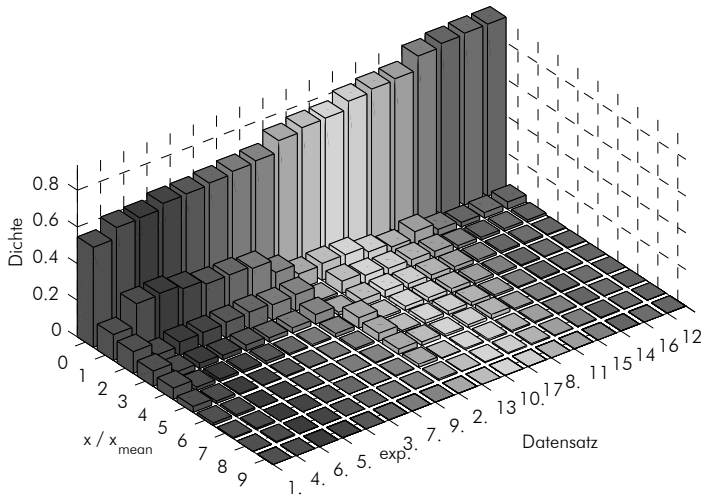


ABBILDUNG 6.11. Histogramme der Anstiege verschiedenen Wettergrößen nummeriert nach Tabelle VIII.

Die Tabelle VIII fasst die charakteristischen Anstiege der untersuchten Wettergrößen im oberen Teil mit den Nummern 1 bis 10 zusammen, aus denen mit Gleichung (6.26) und der noch nicht eingeführten Gleichung (6.38) die Zwischenankunftszeit abgeschätzt werden können. Zusätzlich zeigt Abbildung 6.11 die Histogramme der Anstiege der Wetterdaten.

Von den charakteristischen Jahresverläufen sind insbesondere die Temperatur und die Beleuchtungsstärke interessant, da beide Signale sehr unterschiedliche Eigenschaften besitzen. Dies verdeutlicht Abbildung 6.12 der Jahresverläufe beider Signale. So oszilliert die Temperatur in Abbildung 6.12(a) mit verschiedenen Frequenzen gleichmäßig um wenige Grad Celsius pro Tag, wohingegen die Beleuchtungsstärke in Abbildung 6.12(b) täglich auf bis zu 120000 lx springt und nachts für mehrere Stunden nahezu Null ist. Die resultierenden Zwischenankunftszeiten einer Send-On-Delta Nachrichtengenerierung werden in Abbildung 6.13 für einen Ausschnitt von drei Tagen dargestellt. Deutlich sind bei der Beleuchtungsstärke die großen Zwischenankunftszeiten nachts und die kleinen tagsüber zu erkennen, die Off- und On-Phasen bilden. Damit gibt es scheinbar in Widerspruch zu Prämisse 10 keine statistische Unabhängigkeit der Zwischenankunftszeiten, sondern Burstverhalten mit On/Off-Phasen. Allerdings dauert die Off-Phase bei der Beleuchtungsstärke im Beispiel 10 Stunden und liegt damit weit außerhalb relevanter Netzwerkzeiten. Die Abbildung 6.13(c) verdeutlicht, dass es in der On-Phase keine kurzzeitigen Bursts gibt. Deshalb muss zur Modellierung des Verkehrs kein On/Off-Modell verwendet werden, sondern es ist ausreichend, die Beleuchtungs-

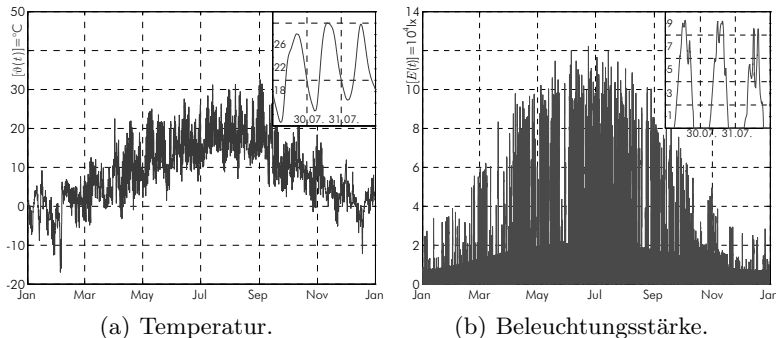


ABBILDUNG 6.12. Jahresverläufe der Außentemperatur und Beleuchtungsstärke von 1984 aus [66] mit vergrößerten Bereich vom 29.07.84 bis 31.07.84

stärke mit den Verfahren aus [75] in ein Nacht- und ein Tag-Szenario zu zerlegen. Hierauf wurde in dieser Arbeit verzichtet, um die Dynamik der Beleuchtungsstärke mit zu beachten.

Ein weiteres Kriterium für die statistische Unabhängigkeit in Prämisse 10 ist die Autokorrelation. Die deutliche Autokorrelation der Temperatur in Abbildung 6.14(a) besitzt bei einer mittleren Zwischenankunftszeit von 547s keine Relevanz für das Netzwerkverhalten. Bei einer Beschränkung mit einer Max-Send-Time von 60s verschwindet die Autokorrelation in Abbildung 6.14(b). Die Beleuchtungsstärke mit einer mittleren Zwischenankunftszeit von 33s besitzt in Abbildung 6.14(c) eine vernachlässigbare Autokorrelation. Damit kann für die untersuchten Signale Prämisse 10 bestätigt werden.

6.4.4.5. UNTERSUCHUNG VON REGELUNGSPROZESSEN

Durch die hohe Anzahl an Räumen in einem Gebäude und die vielen damit verbundenen Quellprozesse ist der Verlauf physikalischer Prozesse im Gebäude besonders wichtig. Deshalb wurde das Anstiegsverhalten an einer Simulation eines Einzelraums in Matlab untersucht, die in [44, 108] geschildert ist und dem Beispiel in Unterabschnitt 5.1.2 mit einer zusätzlichen Kühldecke entspricht. Die Abbildung 6.15 zeigt den Jahresverlauf und einen Ausschnitt von drei Tagen der Temperatur und Beleuchtungsstärke. Im Dreitagesverlauf ist der Einfluss der Nachtabschaltung zu erkennen, die sowohl Kühldecke als auch Beleuchtungsregelung deaktiviert. Im Vergleich zu Abbildung 6.12 zeigt sich ein ähnlicher, aber gestauchter Jahresverlauf. Dies ist darauf zurückzuführen, dass die äußeren Wettergrößen einen großen Einfluss auf die Prozesse im Gebäude haben. Obwohl die Raumtemperatur von der Raumgröße, den Wandeigenschaften, dem einfallen-

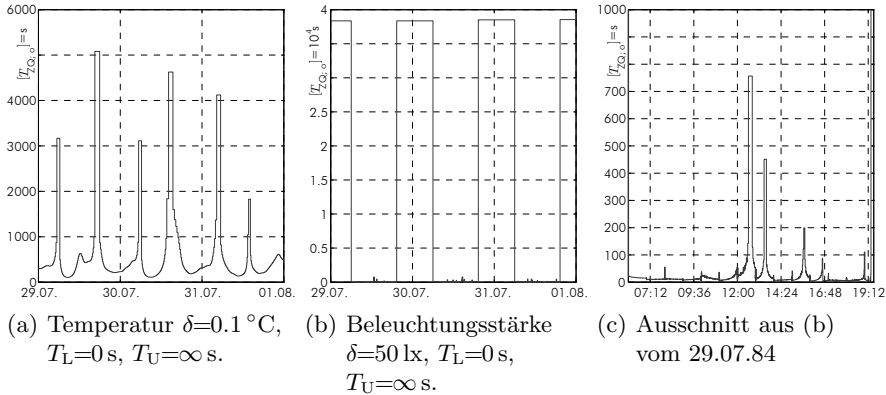


ABBILDUNG 6.13. Zwischenankunftszeiten vom 29.07.84 bis zum 31.07.84 nach Simulation des Send-On-Delta-Algorithmus.

den Licht und wärmeerzeugenden Personen und Geräten abhängt, lässt sich der Quellprozess ohne Simulation des Einzelfalls aus den Wettergrößen gut charakterisieren. Es gilt hierbei meist $|f'|_{\text{mean};\text{in}} \leq |f'|_{\text{mean};\text{out}}$ für die gleichen physikalischen Größen, womit die abgeschätzten Zwischenankunftszeiten in der Regel kleiner und für das Netzwerk kritischer sind.

Dies bestätigt Tabelle VIII, welche die charakteristischen Anstiege der Simulation im unteren Teil mit den Nummern 11 bis 18 auführt. Die Anstiege in Abbildung 6.11 sind wie bei den Wettergrößen exponential verteilt. Die Tabelle VIII charakterisiert zudem die Stellwerte der PID-Regler, der Heizung, Kühlung, Belüftung und Beleuchtung. Sie besitzen einen vergleichbaren mittleren Anstiegsbetrag von ungefähr $7.16 \cdot 10^{-5}$, der als Standardwert für Stellwerte im Bereich (0,1) genutzt werden kann.

Die Charakteristika werden in Unterabschnitt 6.4.6.2 benutzt, um die unterschiedlichen Verfahren zur Abschätzung des Quellprozesses der Send-On-Delta Nachrichtengenerierung anhand des Beispiels zu vergleichen und den Einfluss der Min- und Max-Send-Time zu bewerten. Die Modellierung des Dynamikverhaltens von Regelkreisen mit Charakteristika erfordert allerdings, dass direkte Rückkopplungen des Netzwerkverhaltens auf die Prozesse ausgeschlossen werden. Dabei handelt es sich zum Beispiel um Schwingungen oder Instabilitäten, die durch hohe Verzögerungszeiten im Netzwerk auftreten [141, 175, 206, 228].

PRÄMISSE 14: DÄMPFUNG DER UMGEBUNG

Die Einflüsse der Totzeiten und Dämpfungseigenschaften der Umgebungsprozesse auf die Ankunftsrate sind bedeutend größer, als die Einflüsse des Netzwerkverhaltens.

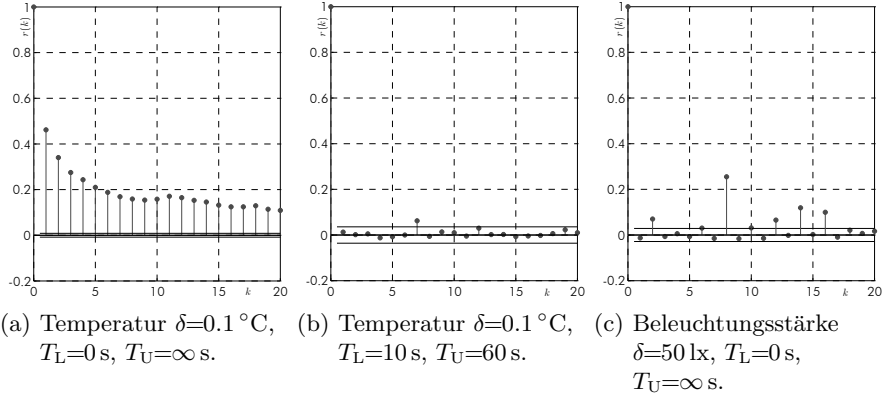


ABBILDUNG 6.14. Korrelogramme der Zwischenankunftszeiten.

6.4.5. EREIGNISORIENTIERTE NACHRICHTENGENERIERUNG KONTINUIERLICHER SIGNALE MIT SCHWELLENWERTEN

Unter *ereignisorientierter Nachrichtengenerierung mit Schwellwerten* werden in dieser Arbeit Prozesse verstanden, die das Ursprungssignal überwachen und nur bei der Über- oder Unterschreitung eines oder mehrerer Grenzwerte $\mathfrak{s} \in \mathfrak{S}_o$ senden. Der Quellprozess ist

$$\mathfrak{T}_{Q;o} = \left\{ t_{k_{Q;o}} = t_{k_{D;o}} \mid \begin{array}{l} (f_o(k_{D;o}) \geq \mathfrak{s} \wedge f_o(k_{D;o} - 1) < \mathfrak{s}) \\ \vee (f_o(k_{D;o}) < \mathfrak{s} \wedge f_o(k_{D;o} - 1) \geq \mathfrak{s}) \end{array} \right\}. \quad (6.28)$$

Ein möglicher Ansatz zur Abschätzung der Übertragungswahrscheinlichkeit bei Schwellwertfiltern bietet die schon diskutierte Send-On-Delta Nachrichtengenerierung. Gibt es mehrere Grenzen $\mathfrak{s} \in \mathfrak{S}_o$, die nahezu äquidistant sind, kann der mittlere Abstand zwischen den Grenzen als ein Delta aufgefasst werden und die bekannten Algorithmen zur Abschätzung des Quellprozesses herangezogen werden.

Gibt es nur wenige Grenzen, muss ein anderer Ansatz gewählt werden. Hierzu wurde der Jahresverlauf der Temperatur und Beleuchtungsstärke in einhundert Bereiche wertediskretisiert und die positiven und negativen Grenzgänge gezählt. Die Abbildung 6.16(b) stellt das resultierende Histogramm dar und vergleicht es mit dem der eingenommenen Werte in Abbildung 6.16(a). Beide Verteilungen stimmen nicht zwangsläufig überein, da ein häufig eingenommener Zustand stabiler ist und nicht ständig wechselt. Die Analyse der Grenzgänge kann zur Bestimmung der realisierten Zwischenankunftszeiten genutzt werden. Hierfür werden für

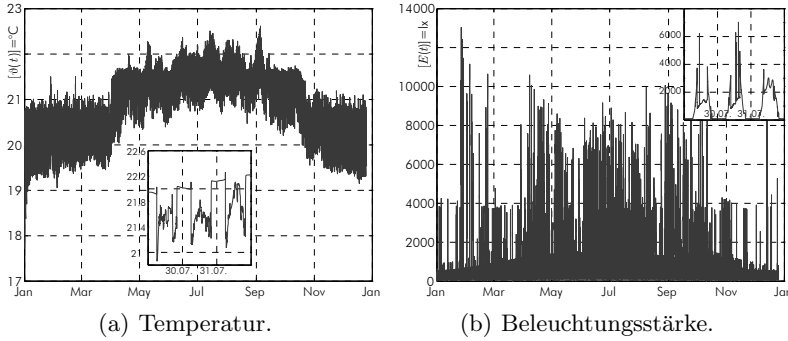


ABBILDUNG 6.15. Jahresverläufe der Raumtemperatur und Raumbeleuchtungsstärke nach Simulation mit vergrößerten Bereich vom 29.07.84 bis 31.07.84

jede Grenze die Grenzgänge pro Jahr als Ankunftsrate genommen und additiv überlagert. Es gilt also für alle Grenzen $s \in \mathfrak{S}_o$ im detaillierten Gerätemodell

$$f_{\lambda_{Q;o}}(\lambda) \approx \bigoplus_{s \in \mathfrak{S}_o} f_{\lambda_{Q;s}}(\lambda) \quad (6.29)$$

und im abstrakten Modell

$$\bar{\lambda}_{Q;o} \approx \sum \bar{\lambda}_{Q;s}. \quad (6.30)$$

Die hierfür notwendigen exponentiell verteilten Zwischenankunftszeiten sind, wie in Abbildung 6.16(c) und 6.17(c) beispielhaft dargestellt, gegeben.

Ist ein beispielhafter Signalverlauf nicht für eine solche Analyse vorhanden, muss ein theoretischer Zugang gewählt werden. Die Problemstellung der Grenzüberschreitung ist in der Literatur als *Level-Crossing-Problem* bekannt [8, 242]. Hierbei handelt es sich um die Fragestellung, wie oft ein zufälliger Bewegungsprozess, einen bestimmten Schwellwert passiert. Zur Lösung muss der Prozess modelliert werden, zum Beispiel als Wiener-Prozess. Weitere Details können [8] oder [242] entnommen werden.

Der Rauchmelder $\vartheta 12$ im Beispiel implementiert eine ereignisorientierte Nachrichtengenerierung mit Schwellwert. Im Alarmfall sendet er eine Nachricht an die Zentrale, wenn der CO_2 -Gehalt im Raum einen Grenzwert von 1200 ppm übersteigt. Die Analyse der CO_2 -Signalspur aus der Simulation ergibt 77 Grenzgänge im Jahr mit einer mittleren Ankunftsrate

$$\bar{T}_{Q;o7} \approx (2.4876 \text{ E}^{-6} \text{ Msg/s})^{-1} = 111.7 \text{ h.}$$

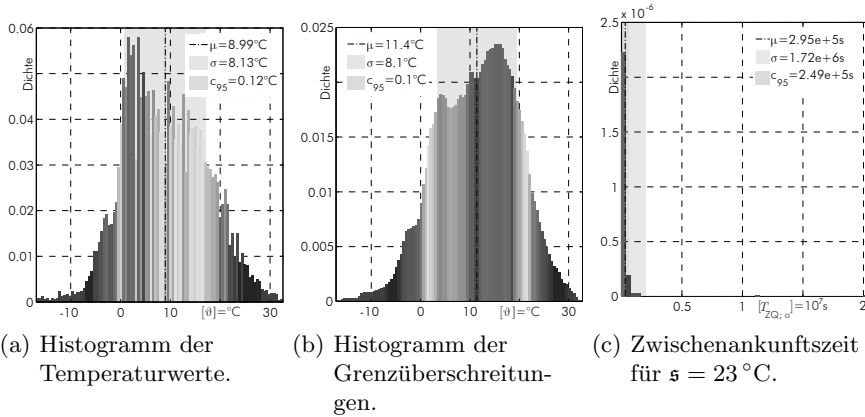


ABBILDUNG 6.16. Histogramme der Grenzüberschreitung am Signal der Temperatur.

Er generiert damit äußerst selten eine Nachricht. Um allerdings sicherzustellen, dass der Rauchmelder nicht defekt ist, wird die Zwischenankunftszeit zusätzlich begrenzt und es wird festgelegt, dass der Rauchmelder mindestens alle 60 Sekunden eine Nachricht erzeugen muss. Diese Begrenzung der Zwischenankunftszeit thematisiert der folgende Unterabschnitt.

6.4.6. BEGRENZUNG DER ZWISCHENANKUNFTSZEITEN

Es gibt zwei Parameter, die verwendet werden, um die Zwischenankunftszeit eines beliebigen, eingeführten Quellprozess $T_{Q;0}^*$ zur ereignisorientierten Nachrichtengenerierung zu begrenzen. Zum einen definiert die *Max-Send-Time* $T_{U;0}$ den maximalen Abstand zweier Telegramme und ermöglicht damit dem Empfänger bei Nichterhalt der Nachricht in dieser Zeit in einen sicheren Zustand zu wechseln. Zum anderen regelt die *Min-Send-Time* $T_{L;0}$ den minimalen Abstand zweier Nachrichten und verhindert eine Überlastung des Kanals, bei starken Änderungen. Damit gilt für die Zwischenankunftszeit prinzipiell $T_{L;0} \leq t_{k_{Q;0}} \leq T_{U;0}$. Allerdings werden diese Zeiten innerhalb des Gerätes als einmaliger Timer nach Gleichung (6.16) realisiert, weshalb es zu einer Mittelwertverschiebung wie bei der zeitorientierten Nachrichtengenerierung in Unterabschnitt 6.4.2 kommt. Somit gilt tatsächlich $T_{L;0}^* \leq t_{k_{Q;0}} \leq T_{U;0}^*$ mit

$$T_{L;0}^* = \text{tmr}(T_{L;0}), \quad T_{U;0}^* = \text{tmr}(T_{U;0}). \quad (6.31)$$

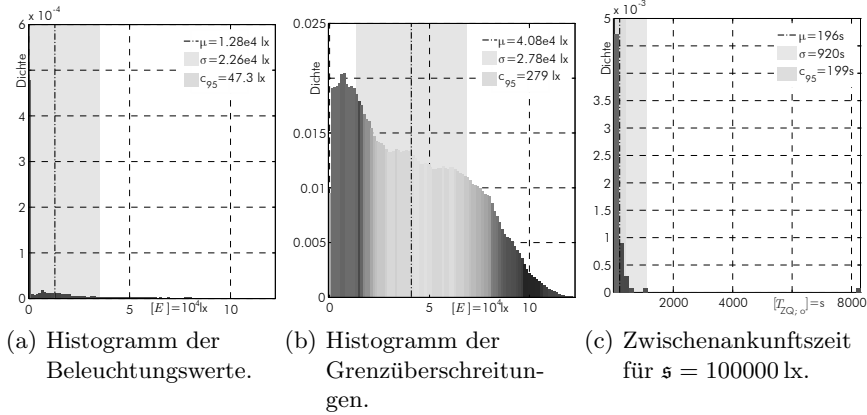


ABBILDUNG 6.17. Histogramme der Grenzüberschreitung am Signal der Beleuchtungsstärke.

und Gleichung (6.14). Der ungefilterte Quellprozess $T_{Q;0}^*(k_{D;0})$ ist begrenzt zu

$$T_{Q;0}^*(k_{D;0}) = \max(T_{L;0}^*; \min(T_{U;0}^*; T_{Q;0}^*(k_{D;0}))). \quad (6.32)$$

6.4.6.1. EINFLUSS IM DETAILLIERTEN GERÄTEMODELL

Die resultierende Dichtefunktion der begrenzten Zwischenankunftszeit setzt sich aus drei Verteilungen zusammen. Agiert nur die Max-Send-Time, so entsteht ein zyklischer Prozess nach Gleichung (6.16) mit $f_{\text{tmr}2;0}(t | T_{U;0})$, wie Abbildung 6.9(c) zeigt. In gleicher Weise existiert für $\delta_0/d_0 \ll T_{L;0}$ ein Prozess der Min-Send-Time mit $f_{\text{tmr}2;0}(t | T_{L;0})$. Im dritten Fall ist nur der ursprüngliche Quellprozess $T_{Q;0}^*$ aktiv.

Die drei Verteilungen sind unterschiedlich stark an der resultierenden Verteilung beteiligt. Der Anteil der Max-Send-Time steht im Zusammenhang mit der Wahrscheinlichkeit, dass die realisierte Zwischenankunftszeit größer ist als $\mathbb{P}[T_{Q;0}^* > T_{U;0}^*]$, ist aber dieser nicht gleich. Dies wird in Abbildung 6.10(a) zu Abbildung 6.10(c) deutlich. Die Abbildung 6.10(a) zeigt das Histogramm der Simulation der Zwischenankunftszeit ohne Min- und Max-Send-Time einer Beleuchtungsstärke. Setzt man für eine neue Simulation die Max-Send-Time auf 60s, so wird bei den 3.5% der Stichprobe, die kleiner sind, die Max-Send-Time früher feuern. In Abbildung 6.10(c) zeigt sich allerdings eine Spitze bei der Max-Send-Time, die mit 32% deutlich größer ist. Dies liegt darin begründet, dass bei einer Max-Send-Time von 5s in $\bar{d}_0/\delta_0 = 20\text{s}$ etwa 4 Nachrichten generiert wer-

den, während bei $\bar{d}_o/\delta_o = 10$ s nur halb so viele Nachrichten entstehen. Damit muss $\mathbb{P}[20 \text{ s} \geq T_{Q;o}^* > 25 \text{ s}]$ bei der Abbildung auf die Zwischenankunftszeit doppelt so stark gewichtet werden wie $\mathbb{P}[10 \text{ s} \geq T_{Q;o}^* > 15 \text{ s}]$. Dementsprechend setzt sich der Anteil der Max-Send-Time-Verteilung zusammen aus

$$\begin{aligned} w_U &= \sum_{n=1}^{\infty} n \mathbb{P} [nT_{U;o}^* < T_{Q;o}^* \leq (n+1)T_{U;o}^*] \\ &= \sum_{n=1}^{\infty} n \left[\mathbf{F}_{T_{Q;o}^*}((n+1)T_{U;o}^*) - \mathbf{F}_{T_{Q;o}^*}(nT_{U;o}^*) \right]. \end{aligned} \quad (6.33)$$

In analoger Weise reduziert sich die Fläche, die kleiner als Min-Send-Time ist, von 31 % in Abbildung 6.10(a) auf 13 % in Abbildung 6.10(b). Ursache ist, dass von ungefilterten Telegrammen, deren Zwischenankunftszeiten in Summe kleiner als die Min-Send-Time sind, nur das letzte gesendet wird und sich dadurch die Gesamtzahl reduziert. Ist $T_{Q;o}^*$ zwanzigfach kleiner als $T_{U;o}^*$ und gilt diese Bedingung zwanzig Telegramme hintereinander, so entfallen 19 Telegramme. Da $T_{Q;o}^*$ eine Zufallsgröße ist und nicht zwanzigmal hintereinander gleich, entsteht bei der folgenden Gleichung ein Extrapolationsfehler für kleine $T_{Q;o}^*$, der vernachlässigt wird. Es gilt für

$$\begin{aligned} w_L &= \sum_{n=1}^{\infty} \frac{1}{n} \mathbb{P} \left[\frac{T_{L;o}^*}{n+1} \leq T_{Q;o}^* < \frac{T_{L;o}^*}{n} \right] \\ &= \sum_{n=1}^{\infty} \frac{1}{n} \left[\mathbf{F}_{T_{Q;o}^*} \left(\frac{T_{L;o}^*}{n} \right) - \mathbf{F}_{T_{Q;o}^*} \left(\frac{T_{L;o}^*}{n+1} \right) \right]. \end{aligned} \quad (6.34)$$

Die Min-Send-Time und Max-Send-Time beschneiden die Dichtefunktion der ungefilterten Zwischenankunftszeit $f_{T_{Q;o}^*}(t)$. Sie geht damit nur mit einem Anteil ein, welcher sich berechnet zu

$$w_\delta = 1 - \mathbb{P} [T_{Q;o}^* < T_{L;o}^*] - \mathbb{P} [T_{Q;o}^* > T_{U;o}^*] = \mathbf{F}_{T_{Q;o}^*}(T_{U;o}^*) - \mathbf{F}_{T_{Q;o}^*}(T_{L;o}^*). \quad (6.35)$$

Da die Grenzen $T_{L;o}^*$ und $T_{U;o}^*$ selbst Zufallszahlen sind, sind die Übergänge fließend. Das Minimum und Maximum statistisch unabhängiger Zufallszahlen wird über die kumulative Verteilungsfunktion bestimmt. Mit $w_S = w_L + w_U + w_\delta$ ergibt sich aus Gleichung (6.32)

$$\begin{aligned} \mathbf{F}_{T_{Q;o}^*}(t) &= \frac{w_L}{w_S} \mathbf{F}_{\text{tmr2}}(t \mid T_{L;o}^*) \\ &\cdot \left[1 - \left(1 - \frac{w_U}{w_S} \mathbf{F}_{\text{tmr2}}(t \mid T_{U;o}^*) \right) \left(1 - \mathbf{F}_{T_{Q;o}^*}(t) \right) \right] \end{aligned} \quad (6.36)$$

6.4.6.2. EINFLUSS IM ABSTRAKTEM GERÄTEMODELL AM BEISPIEL DES SEND-ON-DELTA-ALGORITHMUS

Eine sehr einfache Schätzung der mittleren Zwischenankunftszeit eines gefilterten Quellprozesses ist direkt nach Gleichung (6.32) möglich, indem

$$\bar{T}_{Q;\circ} \approx \max \left(T_{L;\circ}^*; \min \left(T_{U;\circ}^*; \bar{T}_{Q;\circ}^* \right) \right) \quad (6.37)$$

bestimmt wird. Diese Abschätzung ist gültig, sofern die mittlere Zwischenankunftszeit $\bar{T}_{Q;\circ}^*$ des ungefilterten Quellprozesses deutlich im Bereich der Min- oder Max-Send-Time liegt ($\bar{T}_{Q;\circ}^* \ll T_{L;\circ}^* \vee T_{U;\circ}^* \ll \bar{T}_{Q;\circ}^*$) oder deutlich zwischen diesen Bereichen ($T_{L;\circ}^* \ll \bar{T}_{Q;\circ}^* \ll T_{U;\circ}^*$), so dass der Einfluss der anderen Parameter auf das Zentrum der Verteilung gering ist. Auf diese Weise lässt sich der Rauchmelder $\mathfrak{d}12$ aus Unterabschnitt 6.4.5 nahezu fehlerfrei abschätzen. Er hatte mit 40200s eine sehr große mittlere Zwischenankunftszeit, die bei einer Max-Send-Time von 60 Sekunden vollständig kompensiert wird.

$$\bar{T}_{Q;\circ 10} \approx \max(0; \min(60.008 \text{ s}; 40200 \text{ s})) = 60.008 \text{ s}.$$

Aus Gleichung (6.36) leitet sich eine genauere Berechnung ab. Ohne die fließenden Grenzübergänge zu betrachten, ist der Erwartungswert

$$\begin{aligned} \bar{T}_{Q;\circ} &= \frac{w_L}{w_S} \mathbb{E}[\mathbf{f}_{\text{tmr}2}(t \mid T_{L;\circ})] + \frac{w_U}{w_S} \mathbb{E}[\mathbf{f}_{\text{tmr}2}(t \mid T_{U;\circ})] \\ &\quad + \mathbb{E} \left[\mathbf{f}_{T_{Q;\circ}^*}(T_{L;\circ}^* \leq t \leq T_{U;\circ}^*) \right] \\ &\approx \frac{w_L}{w_S} T_{L;\circ}^* + \frac{w_U}{w_S} T_{U;\circ}^* + \frac{w_\delta}{w_S} \mathbb{E} \left[\mathbf{f}_{T_{Q;\circ}^*}(T_{L;\circ}^* \leq t \leq T_{U;\circ}^*) \right]. \end{aligned} \quad (6.38)$$

Für das Beispiel des Send-On-Delta-Algorithmus kann dieser Erwartungswert mit der Annahme eines exponentiell verteilten Anstiegsbetrags aus Unterabschnitt 6.4.4.3 bestimmt werden. Allerdings besitzen die unendlichen Reihen (6.33) und (6.34) in diesem Fall keine geschlossene Lösung, weshalb sie näherungsweise in ein Integral aufgelöst werden, indem für n ein linearer Verlauf angenommen wird. Für die Parameter in Gleichung (6.38) folgt damit

$$\begin{aligned} w_U &\approx \int_{T_{U;\circ}^*}^{\infty} \frac{t}{T_{U;\circ}^*} \mathbf{f}_{T_{Q;\circ}^*}(t) dt = \frac{\delta_\circ}{T_{U;\circ}^* \cdot \bar{d}_\circ} \left(1 - e^{-\frac{\delta_\circ}{T_{U;\circ}^* \cdot \bar{d}_\circ}} \right); \\ w_L &\approx \int_0^{T_{L;\circ}^*} \frac{t}{T_{L;\circ}^*} \mathbf{f}_{T_{Q;\circ}^*}(t) dt = \frac{\delta_\circ}{T_{L;\circ}^* \cdot \bar{d}_\circ} e^{-\frac{\delta_\circ}{T_{L;\circ}^* \cdot \bar{d}_\circ}} \quad \text{und} \\ \mathbb{E} \left[\mathbf{f}_{T_{Q;\circ}^*}(T_{L;\circ}^* \leq t \leq T_{U;\circ}^*) \right] &= \frac{\delta_\circ}{\bar{d}_\circ} \left(e^{-\frac{\delta_\circ}{T_{U;\circ}^* \cdot \bar{d}_\circ}} - e^{-\frac{\delta_\circ}{T_{L;\circ}^* \cdot \bar{d}_\circ}} \right). \end{aligned} \quad (6.39)$$

Mit diesen Parametern für Gleichung (6.38) kann das abstrakte Gerätemodell der Send-On-Delta Nachrichtengenerierung direkt aus dem Mittelwert des Anstiegsbetrags abgeschätzt werden. Dieser ist auch bei Prozessen mit unbekanntem Anstiegsverteilungen für einen Anwender einfach anzugeben und erlaubt damit einen einfachen Modellierungszugang nach Anforderung 2 und eine gute Schätzung nach Anforderung 3.

Bei einem Send-On-Delta-Algorithmus nach der zweiten Implementierungsvariante sinkt die Genauigkeit, wenn $T_{L;\sigma}$ und $T_{U;\sigma}$ nicht deutlich größer als die Abtastzeit T_A sind. Dann gilt

$$\begin{aligned}
 w_U &= \sum_{k=\lceil T_{U;\sigma}^*/T_{A;\sigma}^* \rceil}^{\infty} \left[\frac{k \cdot T_{A;\sigma}^*}{T_{U;\sigma}^*} \right] w_{T_A}(k); & w_\delta &= \sum_{k=\lceil T_{L;\sigma}^*/T_{A;\sigma}^* \rceil}^{\lfloor T_{U;\sigma}^*/T_{A;\sigma}^* \rfloor} w_{T_A}(k); \\
 w_L &= \sum_{k=1}^{\lfloor T_{L;\sigma}^*/T_{A;\sigma}^* \rfloor} \left[\frac{T_{L;\sigma}^*}{k \cdot T_{A;\sigma}^*} \right]^{-1} w_{T_A}(k); & & \\
 \mathbb{E} \left[\mathbf{f}_{T_{Q;\sigma}^*} (T_{L;\sigma}^* \leq t \leq T_{U;\sigma}^*) \right] &= \sum_{k=\lceil T_{L;\sigma}^*/T_{A;\sigma}^* \rceil}^{\lfloor T_{U;\sigma}^*/T_{A;\sigma}^* \rfloor} k \cdot T_A^* \cdot w_{T_A}(k).
 \end{aligned} \tag{6.40}$$

Die Tabelle IX vergleicht die unterschiedlichen Ansätze zur Bestimmung des Quellprozesses eines begrenzten Send-On-Delta-Algorithmus mit Gleichung (6.32), (6.37), (6.38) und (6.39) für den Helligkeitssensor außen und den Temperaturfühler. Die relativen Fehler zu den Simulationsergebnissen des Send-On-Delta-Algorithmus auf eine Spline-Interpolation des Jahresverlaufs werden in den Klammern dahinter angegeben. In einer Java-Implementation benötigt die Simulation pro Spur etwa 60 Minuten und Gleichung (6.32) ganze 90 Minuten. Die Gleichung (6.37), (6.38) und (6.39) und berechnen sich hingegen unter einer Millisekunde¹⁷. Dabei liefert die Abschätzung direkt aus der Dichtefunktion des Anstiegs mit Gleichung (6.38) sehr gute Ergebnisse. Ist der Anstieg nicht ideal exponentialverteilt, so resultiert Gleichung (6.39) in großen Abweichungen. So ist der Fehler bei der Beleuchtungsstärke σ_1 mit den langen Nachtphasen deutlich und fällt beim reinen Tag-Szenario σ_1^* . Damit stellt die Abschätzung nach Gleichung (6.38) einen guten Kompromiss zwischen den Anforderungen 3 und 1 dar und Gleichung (6.39) einen akzeptablen.

¹⁷ Um die Dichtefunktion des Anstiegsbetrags wird die Spur einmalig mit $\tau_{B;\sigma} = 8 \text{ ms}$ in etwa 90 Minuten analysiert.

TABELLE IX.

Vergleich unterschiedlicher Verfahren zur Abschätzung des Erwartungswerts der Zwischenankunftszeit anhand einer Simulation (Lufttemperatur σ_5 mit $\delta_{\sigma_5} = 0.1^\circ\text{C}$, Beleuchtungsstärke σ_1 (σ_1^* Tagsüber) mit $\delta_{\sigma_1} = 50\text{lx}$, Heizungsstellwert σ_6 mit $\delta_{\sigma_6} = .05$; Zeiten in s, relativer Fehler in Klammern).

	T_L	T_U	Simulat.	Gl. (6.32)	Gl. (6.38)	Gl. (6.39)	Gl. (6.37)
σ_1	0	∞	5.697	5.667 (0.52%)	5.682 (0.26%)	5.667 (0.52%)	5.667 (0.52%)
σ_1	5	60	11.95	11.92 (0.25%)	11.95 (0.04%)	8.310 (30.4%)	5.667 (52.6%)
σ_1^*	0	∞	16.26	16.10 (1.00%)	16.19 (0.48%)	16.10 (1.00%)	16.10 (1.00%)
σ_1^*	5	60	18.17	18.05 (0.66%)	18.12 (0.25%)	16.21 (10.7%)	16.10 (11.4%)
σ_5	0	∞	548.0	536.5 (2.09%)	536.5 (2.09%)	536.5 (2.09%)	536.5 (2.09%)
σ_5	10	60	60.02	60.01 (0.03%)	60.02 (0.01%)	60.02 (0.01%)	60.02 (0.01%)
σ_6	0	∞	49.11	48.60 (1.05%)	48.68 (0.87%)	48.60 (1.05%)	48.60 (1.05%)
σ_6	10	60	36.79	36.77 (0.04%)	36.79 (0.01%)	38.97 (5.94%)	48.60 (32.1%)
Fehlermittel				(0.71%)	(0.50%)	(6.46%)	(12.60%)

6.5. FILTERPROZESSE

Ein Filterprozess verarbeitet einkommende Nachrichten an einem Eingangsdatenpunkt i und erzeugt neue Nachrichten an einem Ausgangsdatenpunkt σ . Er wird durch die Verzögerungszeit $\tau_{i,\sigma}$ aus Unterabschnitt 6.3.1 und die Übergangswahrscheinlichkeit $v_{i,\sigma}$ charakterisiert, die an dieser Stelle thematisiert wird.

Im einfachsten Fall ist die Übergangswahrscheinlichkeit $v_{i,\sigma} = 1$ und auf jede eintreffende Nachricht wird eine ausgehende generiert. Dieses Verhalten ist sehr typisch für Geräte mit einer Steueraufgabe, da sie nur bei Änderung eines Werts einen aktualisierten Stellbefehl erzeugen. Ein anderer üblicher Anwendungsfall sind Geräte, die Eingangswerte durchschleifen, um zum Beispiel die Ausführung einer Aktion zu überprüfen, oder die den Eingangswert verarbeiten.

Auch bei Reglern ist ein solches direktes Übergangsverhalten möglich. Um allerdings den Integralteil eines PID-Reglers nachzuführen, muss er regelmäßig ausgeführt werden. Deshalb wird ein PID-Regler entweder zyklisch Nachrichten generieren oder er wendet auf das Stellsignal einen Send-On-Delta-Algorithmus an. In beiden Fällen wird der PID-Regler als Quellprozess ohne den Zusammenhang zu den Eingängen modelliert, auch wenn dieser funktional existiert. Ein solches Beispiel wurde in Unterabschnitt 6.4.4.5 untersucht.

Im Gegensatz dazu wird ein Mehrpunktregler nur dann ein neues Stellsignal senden, wenn ein eintreffender Istwert einen eingestellten Schwellwert passiert oder dieser geändert wird. Ähnlich agiert ein Alarmmelder. In beiden Fällen handelt es sich um eine schwellwertbasierte Filterung des Eingangssignals. Nach Unterabschnitt 6.4.5 können zum Beispiel die in Tabelle VIII aufgeführten Cha-

rakteristika als Eingangssignale analysiert werden und die Zwischenankunftszeiten $T_{QV;o}$ am Ausgang bestimmt werden. Allerdings werden diese nicht als Quellprozess modelliert, sondern als Filterprozess. Dadurch wird das Verhalten verallgemeinert und mögliche Schwankungen der Eingangsankunftsrate, zum Beispiel durch eine andere Parametrierung, werden mit berücksichtigt. Die Übergangswahrscheinlichkeit des Filterprozesses bestimmt sich aus

$$v_{i,o} = \frac{\bar{\lambda}_{QV;o}}{\bar{\lambda}_{Z;i}}. \quad (6.41)$$

Damit ist zur Bestimmung eines Filterprozesses immer der verarbeitete Signalverlauf notwendig. Darauf werden dann die bekannten Verfahren zur Berechnung eines Quellprozesses angewendet und dieser mit dem Eingangsprozess in Zusammenhang gesetzt. In allen Fällen kann der resultierende Filterprozess nach Unterabschnitt 6.4.6 wieder durch eine Min- und Max-Send-Time begrenzt werden.

Allerdings ist es schwierig, den verarbeiteten Signalverlauf zu bestimmen. Während bei Sensoren (λ -Quellen) die Vielfalt noch eingeschränkt ist und Aufstellungen wie Tabelle VIII möglich sind, gibt es bei λ -Filtern eine größere Gerätevielfalt an Steuerungen, Reglern, Filtern, etc. mit einer noch größeren Breite an Implementierungsvarianten. Detaillierte Prozessmodelle könnten eine Analysebasis bilden und problemlos in der systemeigenen Datenbank abgelegt werden, wie der nächste Abschnitt erörtert. Allerdings liegt der Programmcode vieler Geräte nicht offen und ein zusätzlicher Modellierungsaufwand widerspricht den Anforderungen 1 und 2. Deshalb sind bei komplexem oder unbekanntem Geräteverhalten die Verfahren von GÜRTLER [75] zu empfehlen, welche die Übergangswahrscheinlichkeiten direkt aus Messungen an bestehenden Netzwerken modellieren.

6.6. DATENMODELL DES GERÄTEMODELLS

Die Abbildung 6.18 zeigt das Datenmodell der Gerätemodelle. Es vervollständigt die Darstellung des Netzwerks im Systemmodell aus Kapitel 5 um die Komponenten des Verkehrsmodells. Wie in Abschnitt 5.7 beschrieben, handelt es sich bei den neuen Klassen um dynamische Datenobjekte (`ExtendedAttribute`), die das Basismodell erweitern. Für die Basis-Klassen des Systemmodells der Anwendungsschicht `Device`, `FunctionBlock` und `DataPoint` gibt es jeweils Profil-Klassen. Diese Profile stellen die Basis des Gerätemodells dar. Sie sind in der systemeigenen Datenbank gespeichert und werden dem Gerät anhand der Hersteller- und Typbezeichnung (Aufzählung 1) zugewiesen. Mehrere Geräte vom gleichen Typ teilen sich das Profil, wodurch eine effiziente Speichernutzung erzielt wird, da durch den hohen Wiederholungsgrad in Netzwerken der Gebäudeautomation auch bei großen Netzwerken die Zahl der eingesetzten Gerätetypen weit unter hundert

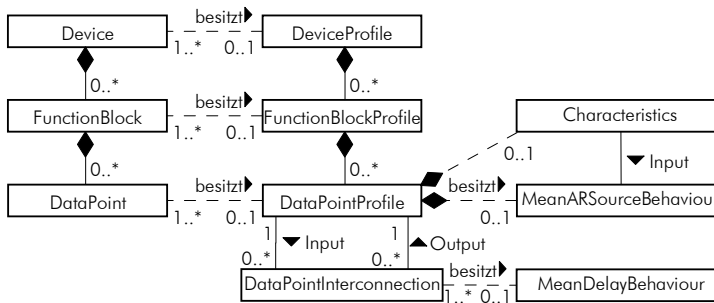


ABBILDUNG 6.18. UML-Darstellung des Gerätemodells.

bleibt [109, 181]. Das `DataPointProfile`, das die Basis-Klasse `DataPoint` erweitert, kann mehrere `DataPointInterconnections` besitzen, die die internen Verbindungen zweier Datenpunkte im Profil repräsentieren und als Attribut die Übergangswahrscheinlichkeit $v_{i,o}$ besitzen. Das Zeitverhalten eines Gerätes wird durch die Klasse `MeanDelayBehaviour` in der Form der Gleichung (6.12) erfasst.

Um das Quellverhalten nach Gleichung (6.38) zu bestimmen, ergänzt `MeanARSourceBehaviour` die Parametrierung der Min- / Max-Send-Time- und Send-On-Delta-Parameter und die Klasse `Characteristic` das Anstiegsverhalten. Dabei wird ausgenutzt, dass die zyklische Nachrichtengenerierung mit $T_L = T_U = T_A$ und die ereignisorientierte Generierung bei Zustandssignalen mit $\delta = 1$ einen Unterfall der ereignisorientierten Send-On-Delta Nachrichtengenerierung mit begrenzten Zwischenankunftszeiten darstellt. Ein `Characteristic` wird dem `DataPointProfile` dynamisch über Schlüsselwörter zugeordnet. Sie beschreiben die Semantik und bilden Schlüsselwortfamilien, in denen sich ausschließende Schlüsselwörter mit dem gleichen Kontext zusammengefasst werden. So ist zum Beispiel die Familie der »physikalischen Größen = {Temperatur, Beleuchtungsstärke, Luftfeuchte, etc.}« und die der »Tageszeiten = {Tag, Nacht}«. In Unterabschnitt 6.4.4.4 wurde beschrieben, dass es sinnvoll ist, die Beleuchtungsstärke in ein Tag- und ein Nachtszenario zu zerlegen. Dementsprechend gibt es ein `Characteristic` für den Tag, eines für die Nacht und eines, das beide vereint. Über globale Schlüsselwörter können die `Characteristic` passender Szenarios schnell ausgewählt werden. Das Zeitverhalten ist in `MeanDelayBehaviour` abgelegt. Die Attribute dieser Klassen werden in Aufzählung 4 vorgestellt. Zu den Klassen `MeanARSourceBehaviour`, `MeanDelayBehaviour` und `Characteristic` gibt es nicht aufgeführte erweiterte Varianten, die die entsprechenden Verteilungsfunktionen implementieren.

Diese Klassen bilden das Gerätemodell und müssen vom Anwender definiert werden. Dabei kann aus der Entwurfsdatenbank nur der strukturelle Teil der Klas-

AUFZÄHLUNG 4. Auszug aus den Attributen der Klassen aus Abbildung 6.18 (Legende in Aufzählung 1).

<p>DeviceProfile</p> <ul style="list-style-type: none"> • clockSpeed (Bezeichner) • modelType (Standardisierter Typ) • manufacturer (Informationsgröße) <p>FunctionBlockProfile</p> <ul style="list-style-type: none"> • profileName (Geräteprofilname) • configurationProperties (Default Konfigurationsparameter) <p>DataPointProfile</p> <ul style="list-style-type: none"> • dataPointName (Bezeichner) • snctUnvtName (Variablentyp) • bitLength (Informationsgröße) - keywordClass (Schlüsselworte) <p>MeanARSourceBehaviour</p> <ul style="list-style-type: none"> - deltaCP (Konfigurationsvariable) - deltaValue (Default Wert) - minStCP (Konfigurationsvariable) - minStValue (Default Wert) - maxStCP (Konfigurationsvariable) - maxStValue (Default Wert) 	<p>Characteristic</p> <ul style="list-style-type: none"> - riseMin (Minimum Anstiegsbetrag) - riseMean (Mittelw. Anstiegsbetrag) - riseMax (Maximum Anstiegsbetrag) - valueMin (Minimum Wertebereich) - valueMax (Maximum Wertebereich) - keywordClassList (Schlüsselworte) <p>MeanDelayBehaviour</p> <ul style="list-style-type: none"> ★ serviceTimeIto4 ($\bar{\tau}_{1,4}, \sigma_{\tau_{1,4}}$) ★ serviceTime4to7 ($\bar{\tau}_{4,7}, \sigma_{\tau_{4,7}}$) ★ serviceTime7to4 ($\bar{\tau}_{7,4}, \sigma_{\tau_{7,4}}$) ★ serviceTime4to1 ($\bar{\tau}_{4,1}, \sigma_{\tau_{4,1}}$) ★ serviceTimeApp ($\bar{\tau}_B, \sigma_{\tau_B}$) <p>DataPointInterconnection</p> <ul style="list-style-type: none"> - gain ($v_{i,o}$)
---	--

sen entnommen werden, also über welche Funktionsblöcke und Datenpunkte ein Gerät verfügt. Die Attribute von **Characteristic**, **DataPointInterconnection** und **MeanARSourceBehaviour** müssen durch den Anwender definiert werden. Für **MeanDelayBehaviour** werden standardmäßig die in Unterabschnitt 6.3.1 analysierten Werte benutzt.

6.7. VALIDIERUNG

Bei der Validierung gibt es zwei Aspekte, die zu unterscheiden sind. Dies ist einerseits die *Genauigkeit* der vorgestellten Abschätzungen für Quell- und Filterprozesse anhand praktischer Signale. Davon ist andererseits die *Übertragbarkeit* der Gerätemodelle zu trennen, also inwiefern ein verallgemeinertes Geräteverhalten auch für ein Gerät in einem anderen Netzwerk gültig ist.

Die Abbildung 6.19 gibt einen Überblick über die gemessenen Zwischenankunftszeiten in einem Musterraum. Dabei handelt es sich um eine 1.5x1x1 Meter große Holzkammer, in der typische Sensoren, Regler und Aktoren verbaut wurden,

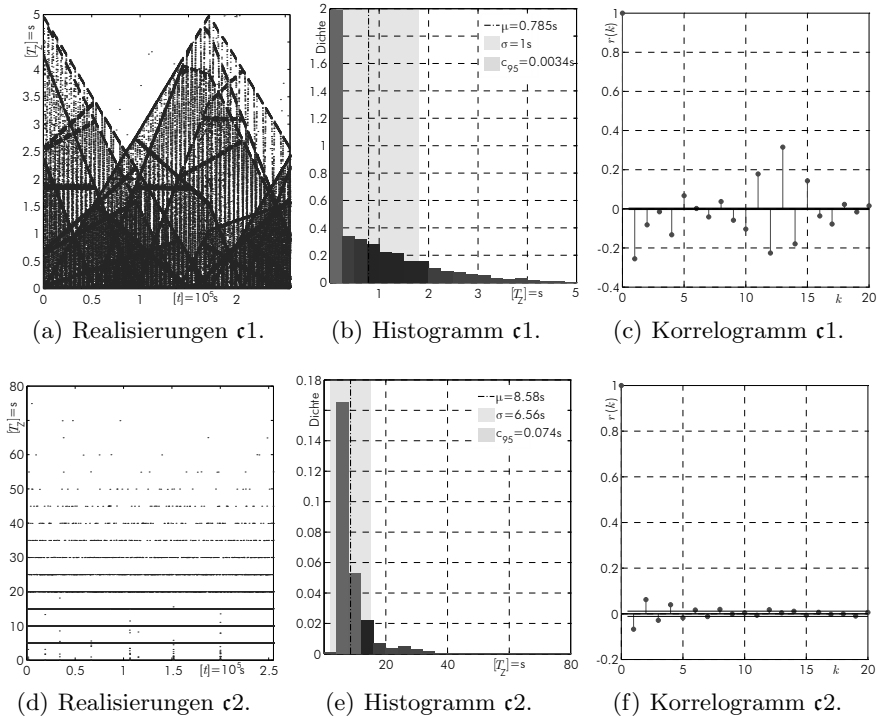


ABBILDUNG 6.19. Zwischenankunftszeiten einer Messung am Musterraum mit zwei Kanälen c1 und c2.

ähnlich dem Beispiel aus Unterabschnitt 5.1.2. Die meisten Geräte sind am ersten von zwei Kanälen angeschlossen, weshalb die Zwischenankunftszeiten im Mittel kleiner sind. Der zweite Kanal besitzt nach Abbildung 6.19(e) eine exponentiell verteilte Zwischenankunftszeit, während am ersten Kanal deutlich Korrelationen in Abbildung 6.19(c) und Muster in Abbildung 6.19(a) sichtbar sind.

Diese Korrelation steht im Zusammenhang mit dem Geräteverhalten. Beide Kanäle enthalten vorwiegend λ -Quellen, die nach dem Send-On-Delta-Algorithmus und zum Teil mit Min- und Max-Send-Time arbeiten. Die Abbildung 6.20 gibt einen Überblick über ihr Quellverhalten. Von links nach rechts sind die Zwischenankunftszeiten, ihre Histogramme und Korrelogramme sowie der Signalverlauf aufgetragen. Der Vergleich der Mittelwert der Histogramme mit den Geräteparametern aus Tabelle X zeigt auf, dass bei der Innen- und Außenbeleuchtungsstärke sowie bei der Luftqualität fast ausschließlich die Max-Send-Time agiert. Dies liegt an den geringen Signalschwankungen im Musterraum, die im Vergleich

6. AUTOMATISCHE MODELLIERUNG DES VERKEHRSMODELLS

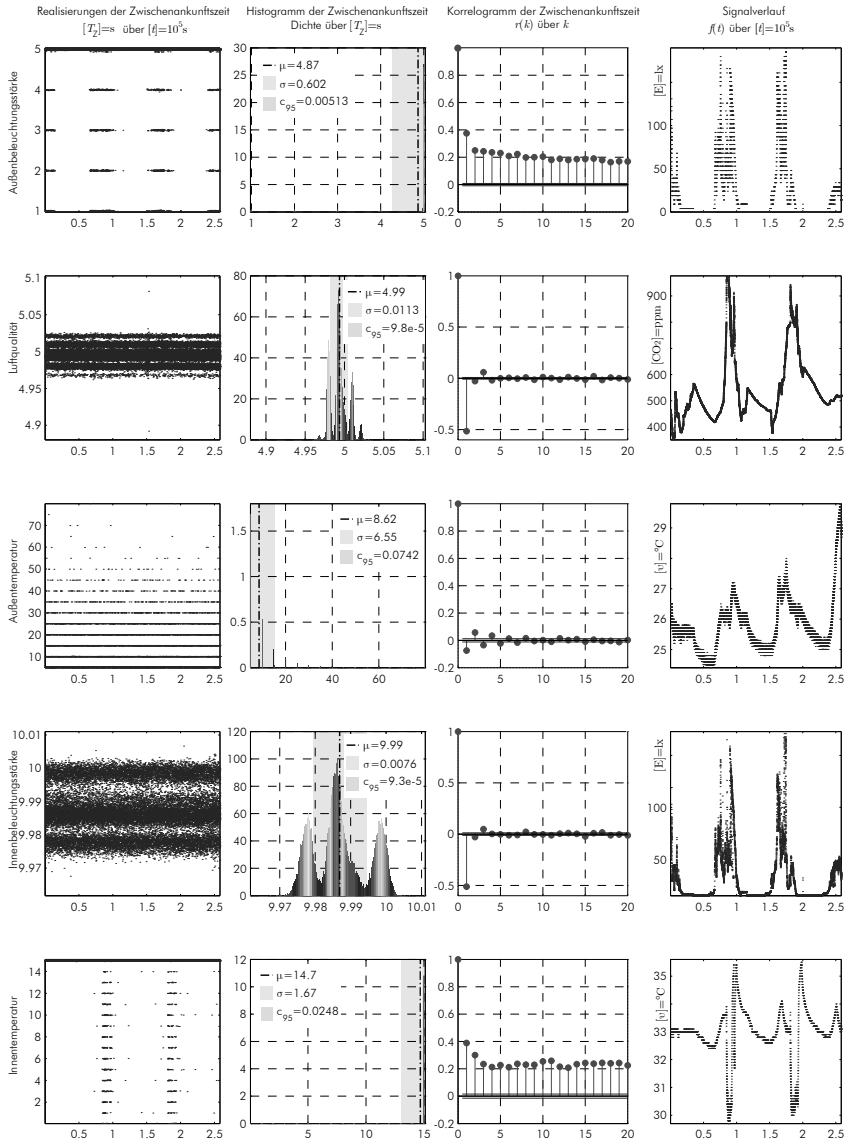


ABBILDUNG 6.20. Verkehrsstatistiken ausgewählter Geräte aus dem Musterraum. Die Beschriftungen der Diagramme stehen über der ersten Zeile und gelten spaltenweise. Das gemessene Gerät steht links und gilt pro Zeile.

TABELLE X.
Parameter der Geräte aus der Messung.

Nr.	Größe	δ_o	T_L	T_U	T_A
1	Außenbeleuchtungsstärke	5 lx	1 s	5 s	1 s
2	Innenbeleuchtungsstärke	1 lx	1 s	10 s	1 s
3	Außentemperatur	0.1 °C	10 s	80 s	5 s
4	Innentemperatur	0.1 °C	0 s	15 s	1 s
5	Luftqualität (zyklisch)	-	5 s	5 s	5 s

zu einem normalen Raum durch die Bauweise des Musterraums und den Standort im Labor noch zusätzlich gedämpft sind.

Durch die dominante Max-Send-Time entstehen im ersten Kanal hauptsächlich zyklische Quellprozesse. Obwohl die eingestellten Max-Send-Times von 5, 10 und 15 Sekunden Vielfache voneinander sind, etabliert sich keine feste Reihenfolge bei den Sendungen. Stattdessen zeigt Abbildung 6.19(a) scheinbar viele überlagerte Dreieckssignale, die sich im Korrelogramm in Abbildung 6.19(c) als Phasen ausprägen. Sie entstehen, durch die in Unterabschnitt 6.4.2 diskutierte, ungleiche Mittelwertverschiebung der Timer. Dadurch sind die mittleren Zwischenankunftszeiten der Geräte keine Vielfache mehr und wandern leicht zueinander. Bei größeren Netzwerken mit mehr und aktiveren Prozessen ist davon auszugehen, dass solche Effekte und die Korrelation der Zwischenankunftszeiten abnehmen.

Ein gutes Beispiel für das Verkehrsverhalten einer Send-On-Delta Abtastung ohne Min- oder Max-Send-Time stellt der Außentempersensordar. Seine Zwischenankunftszeiten sind mit einer zugrunde liegenden Abtastzeit $T_A = 10$ s in Abbildung 6.20 deutlich exponentiell verteilt und prägen den zweiten Kanal. Das durch Gleichung (6.4.2) beschriebene Verhalten eines einmaligen Timers kann an der Luftqualität und Innenbeleuchtungsstärke nachvollzogen werden. Die starke Korrelation bei der Außenbeleuchtungsstärke und Innentemperatur entsteht durch die vorherrschende Max-Send-Time, die tagsüber von kurzen zusammenhängenden, dynamischen Signalphasen unterbrochen wird. Es resultiert ein On/Off-Prozess, dessen Einfluss auf die Korrelation durch eine Zerlegung in ein Tag/Nacht-Szenario verschwindet.

Tabelle XI vergleicht das abgeschätzte Quellverhalten der Geräte mit dem gemessenen. Für jede Quelle listet die erste, mit einem Stern gekennzeichnete Zeile die theoretischen Ergebnisse auf, die sich ergeben würden, wenn man aus den bisher verwendeten Signalverläufen in Tabelle VIII die korrespondierenden anwenden würde. Dadurch wird die Übertragbarkeit der Modelle getestet. Mit einem mittleren Fehler von fast 200 % zeigt sich, dass die Übertragbarkeit nicht gewährleistet ist. Ursache ist der starke Dämpfungscharakter des Musterraumes, der in keinem Vergleich mit einem normalen Raum steht, so dass der Musterraum

TABELLE XI.

Vergleich unterschiedlicher Verfahren zur Abschätzung des Erwartungswerts der Zwischenankunftszeit bei Send-On-Delta anhand der gemessenen Signalspuren (Nummerierung und Parameter in Tabelle X; * theoretische Ergebnisse nach Tabelle VIII; F - Fehlermittel).

Nr.	Mes.	Simulation	Gl. (6.32)	Gl. (6.38)	Gl. (6.39)	Gl. (6.37)
1*	4.87	2.389 (50.9%)	1.732 (64.4%)	1.737 (64.3%)	1.296 (73.4%)	1.004 (79.4%)
1		4.916 (0.95%)	4.751 (2.45%)	4.749 (2.48%)	4.999 (2.66%)	5.005 (2.78%)
2*	9.99	2.283 (77.2%)	1.591 (84.1%)	1.602 (84.0%)	1.154 (88.5%)	1.004 (90.0%)
2		8.648 (13.4%)	7.660 (23.3%)	7.690 (23.0%)	6.559 (34.3%)	9.873 (1.17%)
3*	8.62	80.03 (828%)	80.02 (828%)	80.02 (828%)	80.02 (828%)	80.02 (828%)
3		15.46 (79.3%)	6.563 (23.9%)	5.734 (33.5%)	6.534 (24.2%)	6.574 (23.6%)
4*	14.7	15.01 (2.12%)	15.01 (2.08%)	15.01 (2.07%)	15.01 (2.10%)	15.01 (2.10%)
4		13.94 (5.20%)	12.58 (14.4%)	12.59 (14.3%)	14.00 (4.73%)	15.01 (2.10%)
5*	4.99	5.009 (0.38%)	5.005 (0.30%)	5.005 (0.30%)	5.005 (0.30%)	5.005 (0.30%)
5		5.010 (0.39%)	5.005 (0.30%)	5.005 (0.30%)	5.005 (0.30%)	5.005 (0.30%)
F*		(191.7%)	(195.8%)	(195.7%)	(198.5%)	(199.9%)
F		(19.85%)	(12.87%)	(14.72%)	(12.84%)	(5.99%)

nicht repräsentativ ist, um die Übertragbarkeit zu beurteilen.

Die zweite Zeile jeder Quelle in Tabelle XI vergleicht die gemessenen Zwischenankunftszeiten mit den Ergebnissen der Analyse der übertragenen Werteverläufe, um die Genauigkeit zu prüfen. Die Ergebnisse sind meist qualitativ gut und vergleichbar mit Tabelle IX. Nur die Außentemperatur weist bereits bei der Simulation des Send-On-Delta-Algorithmus einen großen Fehler auf. Er führt sich auf einen Interpolationsfehler des mit 0.1 °C und 10s sehr grob werte- und zeitdiskretisierten Signalverlaufs zurück, der damit Prämisse 13 verletzt.

Die gute Genauigkeit bestätigt sich auch bei λ -Filtern. Die Tabelle XII vergleicht die nach Gleichung (6.7) abgeschätzten mittleren Ankunftsrate zweier Steuerungen, die auf jede ankommende Nachricht reagieren.

Damit kann zusammengefasst werden: Die Übertragbarkeit der für Tabelle VIII untersuchten Prozesse auf den Musterraum kann nicht bestätigt werden. Allerdings ist der Musterraum hierfür auch ungeeignet und eine repräsentative Untersuchung müsste unterschiedliche Räume und Gebäude analysieren. Davon losgelöst konnte die Genauigkeit der Gerätemodelle durch die Messung bestätigt werden.

TABELLE XII.

Vergleich der Abschätzung der mittleren Ankunftsrate zweier λ -Filter mit einer Übergangswahrscheinlichkeit $v_{i,o}$ von eins zur Messung (F - Fehler).

(a) Kühlungssteuerung					(b) Lüftungssteuerung						
$\bar{\lambda}_i$	0.0095	0.0074	0.0767	F	$\bar{\lambda}_i$	0.0077	0.0110	0.0046	0.2003	0.0767	F
$v_{o,i}$	1	1	1		$v_{o,i}$	1	1	1	1	1	
$\bar{\lambda}_o^*$		0.0935			$\bar{\lambda}_o^*$			0.2999			
$\bar{\lambda}_o$		0.0935		0.0%	$\bar{\lambda}_o$			0.3004			0.2%
$v_{o,i}$	0	1	0		$v_{o,i}$	1	0	0	0	0	
$\bar{\lambda}_o^*$	-	0.0074	-		$\bar{\lambda}_o^*$	0.0075	-	-	-	-	
$\bar{\lambda}_o$	-	0.0074	-	0.0%	$\bar{\lambda}_o$	0.0077	-	-	-	-	3.7%
					$v_{o,i}$	0	0	1	0	0	
					$\bar{\lambda}_o^*$	-	-	0.0046	-	-	
					$\bar{\lambda}_o$	-	-	0.0046	-	-	0.0%

6.8. MODELLIERUNGSGRENZEN

Die Grenzen der Modellierung des Verkehrsmodells liegen vor allem in den genannten Prämissen und der Übertragbarkeit des Geräteverhaltens.

Das in Abschnitt 6.3 und 6.4 modellierte Grundverhalten der Geräte im Zeit- und Quellverhalten konnte durch Simulation und verschiedene Messungen bestätigt werden. Hierbei gibt es praktische Abweichungen im Geräteverhalten, so wurden in Unterabschnitt 6.4.2 bereits zwei unterschiedliche Varianten, einen Timer zu realisieren, vorgestellt und damit gewiss nicht alle abgehandelt. Entsprechend gibt es noch mehr Varianten, die komplexeren Algorithmen zu implementieren. Allerdings offenbart Unterabschnitt 6.4.2 auch einen Vorteil des komplexen Geräteverhaltens, da es nach dem zentralen Grenzwertsatz meist in gut modellierbaren Normalverteilungen resultiert¹⁸.

Grundsätzlich sind die Abweichungen zur Realität im detaillierten Gerätemodell größer als im abstrakten. Die Genauigkeit der Dichtefunktionen des detaillierten Gerätemodells hängt vor allem von der Prämisse 10 zur statistischen Unabhängigkeit und Abwesenheit von Bursts ab, die praktisch nicht zu garantieren sind. Die Zwischenankunftszeiten sind im Vergleich zum Netzwerkverhalten allerdings üblicherweise sehr groß, so dass Ungenauigkeiten in den Verteilungen und Voraussetzungen zu vernachlässigen sind, da sie keine starke Auswirkung auf das Geräteverhalten haben. Das abstrakte Gerätemodell rechnet mit Erwartungswerten, die die Prämisse 10 nicht benötigen, wie sich auch bei der Diskussion der Bursts in Unterabschnitt 6.3.2 zeigt. Die Genauigkeit der aufgestellten Abschät-

¹⁸ Da Normalverteilungen invariant gegenüber der Faltung sind, kann dadurch in der Implementation die Faltung im detaillierten Gerätemodell stark vereinfacht werden.

zungen zum Quellverhalten bestätigen Tabelle IX und Tabelle XI. Deshalb ist die Genauigkeit der Gerätemodelle im Rahmen der Notwendigkeit gegeben.

Kritischer ist die Übertragbarkeit der Gerätemodelle einzuschätzen, insbesondere da sich Fehler durch die Wiederverwendung der Gerätemodelle potenzieren können. Dass die Übertragbarkeit der Gerätemodelle nicht uneingeschränkt gegeben sein kann, liegt in der Natur der Vorhersage. Die Genauigkeit der Algorithmen lässt sich anhand der untersuchten und verallgemeinerten Ursprungssignale nachweisen, die Übertragbarkeit des Verhaltens jedoch nur in umfangreichen Experimenten, die bisher aus praktischen Gründen nicht durchgeführt werden konnten. Dabei deckt das in Abschnitt 6.7 betrachtete kleine Beispiel bereits die Schwierigkeiten auf. Mit seinem besonderen Dynamikverhalten ist der Musterraum zwar nicht repräsentativ, um die Übertragbarkeit zu bewerten, allerdings demonstriert er die Notwendigkeit, die Gerätemodelle zu differenzieren. Dies bestätigen die Erfahrungen über die Nutzerprofile in der Gebäudesimulation [86, 122]. Hier zeigt die zunehmende Etablierung der Verfahren zur Energieverbrauchsabschätzung auch die Praktikabilität solcher Ansätze.

Die Differenzierung kann letztendlich nur durch ein großes Repertoire an Gerätemodellen und charakteristischen Signalverläufen erreicht werden. Hierbei erlauben die im letzten Abschnitt 6.6 angesprochenen Schlüsselwörter eine einfache Ausdifferenzierung nach bestimmten Anwendungsszenarien, so dass zum Beispiel eine Temperatur in eine Raumtemperatur und Kesseltemperatur unterschieden wird. Diese große Anzahl an Gerätemodellen kann allerdings nicht von Hand erstellt werden, da ein umfangreiches Modellierungs- und Prozesswissen erforderlich ist. Die Arbeit von GÜRTLER [75] erforscht die automatische Modellierung aus Messungen an realen Netzwerken, die in diesem Punkt Abhilfe schaffen kann. Die hier vorgestellten Modelle dienen wiederum der Anpassung an Parameter im Netzwerkentwurf.

6.9. ZUSAMMENFASSUNG

Das aufgestellte Verkehrsmodell ermöglicht die automatische Modellierung des Nachrichtenaufkommens in einem bestehenden Netzwerk. Es erfüllt damit zusammen mit dem Systemmodell das Teilziel 1, insbesondere unter den Anforderungen 3 und 5. Das Verkehrsmodell kann sehr spezifisch an das Netzwerk angepasst werden, da der Verkehr auf die einzelnen produzierenden Geräte heruntergebrochen wird und zusätzlich die Geräteinteraktionen modelliert werden. Diese Modellierung fügt sich in einzelnen Gerätemodellen der Geräte zusammen, die in einer systemeigenen Datenbank zur Wiederverwendung abgespeichert werden. Das ermöglicht die automatische, modulare Generierung des Verkehrsmodells aus dem Systemmodell der Anwendungsschicht heraus. Weiterhin werden in den Gerätemodellen die Verkehrseigenschaften nicht direkt charakterisiert, sondern es

werden die üblichen Verfahren zur Erzeugung von Nachrichten in der Gebäudeautomation untersucht und der Zusammenhang zwischen ihren Parametern und dem Verkehrsverhalten modelliert. Dadurch ist es möglich, anhand der im Netzwerkentwurf festgelegten Geräteparameter die spezifische Verkehrscharakteristik der Gerätemodelle zu bestimmen. Damit kann das Verkehrsmodell sowohl strukturell als auch parametrisch gut an einen spezifischen Netzwerkentwurf angepasst werden.

ALGORITHMUS 1. Algorithmus zur Berechnung der mittleren Ankunftsrate mit dem abstrakten Gerätemodell.

1. Wenn eine ideale Reihenfolge $\mathfrak{V} \neq \emptyset$ besteht, dann nutze sie $\mathfrak{E} = \mathfrak{V}$, sonst identifiziere die Ausgangsdatenpunkte mit Quellanteil $\mathfrak{E} \subseteq \mathfrak{D} \mid \bar{\lambda}_{Q,e} > 0$ (`DataPoint` oder `DataPointProfile` besitzt `MeanARSourceBehaviour`)
 2. Für jedes $e \in \mathfrak{E}$
 3. Wenn $e \in \mathfrak{D} \wedge \mathfrak{B}_e \neq \emptyset$ (Ausgang & ausgehende Bindings)
 4. Bestimme die passende `Characteristic`-Instanz
 5. Berechne die mittlere Ankunftsrate nach `MeanARSourceBehaviour`
 6. Für alle eingehenden internen Verbindungen
 7. Addiere $v_{i,o} \bar{\lambda}_i$ zur Ankunftsrate
 8. Für alle ausgehenden Bindings
 9. Füge alle Zieldatenpunkte ans Ende von \mathfrak{E} sofern nicht enthalten
 10. Wenn $e \in \mathfrak{J}$ und interne Verbindungen ausgehen
 11. Berechne die mittlere Ankunftsrate nach Gleichung (6.9)
 12. Erweitere \mathfrak{E} um alle intern verbundenen Datenpunkte sofern nicht enthalten
 13. Entferne e aus \mathfrak{E} und füge e zu \mathfrak{V}
-

Die unterschiedlichen Modelle werden anhand von Simulationen und Messungen in den Tabellen IX und XI verglichen und bewertet. Dabei erweisen sich für das detaillierte Modell die Gleichung (6.38) und im abstrakten Modell die Gleichung (6.39) als guter Kompromiss zwischen den Anforderungen 3 und 5. Der Algorithmus 1 fasst die Bestimmung des Verkehrsmodells am Beispiel des abstrakten Gerätemodells zusammen. Der Vorteil dieses Algorithmus ist, dass zuerst die Ankunftsraten reiner λ -Quellen bestimmt und dann sequentiell die Wirkungsketten der verbundenen Datenpunkte analysiert werden. Dabei bildet sich eine ideale Reihenfolge \mathfrak{V} zur Lösung heraus, bei der kein Datenpunkt berechnet wird, bevor nicht alle eingehenden Größen bekannt sind. Soll die Ankunftsrate im Laufe einer weiteren Iteration aktualisiert werden, so ist diese ideale Reihenfolge bekannt.

Die Tabelle XIII fasst noch einmal die Ergebnisse für den Beispielraum aus Unterabschnitt 5.1.2 zusammen. Bei dieser Gegenüberstellung ist besonders zu beachten, dass die mittlere Ankunftsrate $\bar{\lambda}$ der zyklisch abgetasteten Beleuchtungsstärke σ_2 fünf Größenordnungen über der ereignisorientierten Fensteröffnung σ_8

TABELLE XIII.
Ankunftsdaten für das Beispiel.

Größe		Klassifizierung	$[\bar{T}_Z] = s$	$[\bar{\lambda}] = \frac{M_{sg}}{s}$	Gleichung	
b1	o1	Beleuchtungsstärke	SOD λ -Quelle	11.951	0.084	(6.26), (6.38)
b2	o2	Beleuchtungsstärke	SOD λ -Quelle	0.5077	1.939	(6.14)
b3	o3	Jalousiestand	zykl. λ -Quelle	0.5077	1.939	(6.14)
b4	o4	Lampendimmung	zykl. λ -Quelle	0.5077	1.939	(6.14)
b5	o5	Temperatur	SOD λ -Quelle	60.019	0.0166	(6.26), (6.38)
b6	o6	Ventilöffnung	SOD λ -Filter	36.789	0.0272	(6.26), (6.38)
b7	o7	Anwesenheit	ereign. λ -Quelle	8640	$1.2 \cdot 10^{-4}$	-
b8	o8	Fensteröffnung	ereign. λ -Quelle	17280	$5.8 \cdot 10^{-5}$	-
b9	o9	Fensteröffnung	\sim & λ -Filter	8640	$2 \cdot 5.8 \cdot 10^{-5}$	-, (6.8)
b10	o10	Rauchmeldung	ereign. λ -Quelle	60.008	0.0166	(6.37)
b11	o11	Alarm	λ -Filter	59.156	0.0169	(6.8)

liegt. Diese starken Unterschiede sind typisch in die Gebäudeautomation. Auch in der Messung aus Abschnitt 6.7 traten alle Geräte, die auf Benutzeraktionen reagierten, kaum in Erscheinung. Dies erschwert die Leistungsbewertung, wie sich im nächsten Kapitel zeigen wird.

KAPITEL

7.

LEISTUNGSBEWERTUNG

Der Zufall ist der einzig legitime Herrscher des Universums.

Napoleon Bonaparte

7.1. ALLGEMEINE ERGEBNISANFORDERUNGEN

IM Unterabschnitt 3.2.1 wurden die grundlegenden Konzepte der Leistungsbewertung eingeführt und geschlussfolgert, dass entsprechende Methoden vorhanden und etabliert sind. Hierzu ergänzt dieses Kapitel keinen neuen methodischen Ansatz, sondern definiert, welche Ergebnisse und Anforderungen zur Diagnose im nächsten Kapitel notwendig sind. Ohne eine Einschränkung auf eine Methode der Leistungsbewertung zu machen, werden im Folgenden zwei implementierte Beispiele für LON vorgestellt.

Die für LON implementierte Leistungsbewertung nutzt einen analytischen und einen simulativen Ansatz in Kombination, um die konträren Anforderungen 1 und 3 gemäß Anforderung 4 zu erfüllen. Die Abbildung 7.1 zeigt den vollständigen Algorithmus. Aufgrund der starken Abhängigkeiten der einzelnen Modellteile, die in Abschnitt 7.3 thematisiert wird, ist bei der analytischen Lösung eine *Fix-Punkt-Analyse* notwendig. Dabei wird das Modell mehrmals analytisch gelöst (①–③) und jedes Mal werden die Parameter des System- und Verkehrsmodells an die neuen Ergebnisse angepasst (①). Wenn die neue Lösung nur noch geringfügig um ε von der alten abweicht (③), wird die Iteration im Mittel nach 10 Schritten beendet. Danach wird mit dem vorparametrierten Modell eine Simulation (④) durchgeführt, um die detaillierten Ergebnisse zu bestimmen. Die abhängigen Teile des Verkehrsmodells werden dabei mit simuliert, so dass keine Fix-Punkt-Analyse notwendig ist. Da die Simulation mit einer Komplexität von $O(n^2)$ bei großen

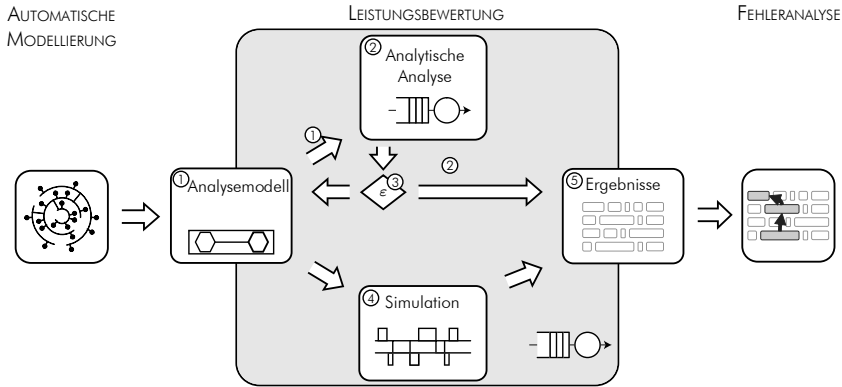


ABBILDUNG 7.1. Detaillierte Struktur der Leistungsbewertung.

Netzwerken deutlich länger als eine analytische Berechnung mit $O(n)$ braucht (vgl. Abbildung 9.1(a) auf Seite 158), wird so eine effiziente und genaue Lösung nach Anforderung 1 und 3 erreicht.

BUCHHOLZ [15, 16, 108, 176] entwickelte die hier genutzte analytische Analyse. Es handelt sich um eine Warteschlangenanalyse des von LON verwendeten p -persistent CSMA-Algorithmus, wobei ein jedes Gerät mit einer ausgehenden Warteschlange modelliert wird. Der Kanal stellt prinzipiell eine Bearbeitungsstation mit lastabhängiger Bedienrate dar, die alle angebotenen Gerätewarteschlangen bearbeitet. Der Ansatz dekomponiert diese zentrale Bearbeitungsstation in einzelne $M/M/1/b_\delta$ -Bearbeitungsstationen für die Geräte mit verknüpfter Bedienrate und einer Buffergröße b_δ . Dadurch kann das Geräteverhalten in seinen Mittelwerten gut abgeschätzt werden. Voraussetzung sind exponentielle Ankunfts- und Bedienprozesse nach Prämisse 10 ohne Bursts, wofür sich das einfache Ankunftsratenmodell gut eignet.

Bei der Simulation handelt es sich um eine ereignisorientierte Simulation des LON-Protokolls [40, 51, 54]. Die Simulation ist im Sinne von Anforderung 1 auf eine hohe Performance ausgerichtet. Darauf hat die zentrale Ereignisverwaltung einen großen Einfluss mit einem Scheduling-Aufwand von $O(\log(n))$ ¹. Dieser Aufwand wird reduziert, indem zentrale Ereignisse vermieden werden und Kanäle, Geräte und Transaktionen lokale Ereignisfolgen in einem Divide-And-Conquer-Ansatz selbst verwalten. Durch die Simulation der Transaktionen werden Burstprozesse einzelner Übertragungsdienste (Unterabschnitt 6.3.2) auf der Bitübertragungsschicht berücksichtigt. Auch Bursts bei gekoppelten Prozessen werden beachtet, indem Filterprozesse direkt simuliert werden. Außerdem werden belie-

¹ Die angegebene Komplexität bezieht sich auf den Sortieraufwand der Ereignisse in chronologischer Reihenfolge. Hierfür wird ein Rot-Schwarz-Baum mit $O(\log(n))$ verwendet.

bige², statistisch unabhängige Quellprozesse mit einem detaillierten Verkehrsmodell unterstützt.

Ein ähnlicher Ansatz ist auch für andere Protokolle wie Konnex oder BACnet möglich, genauso wie andere Methoden der Leistungsbewertung denkbar sind. Entscheidend ist, dass die im folgenden Abschnitt definierten Leistungskenngrößen für die weitere Analyse bereitgestellt werden.

7.2. NOTWENDIGE LEISTUNGSKENNGRÖSSEN

Die im Folgenden vorgestellten Leistungskenngrößen werden für die weitere Betrachtung benötigt, so dass gilt:

PRÄMISSE 15: LEISTUNGSKENNGRÖSSEN

Die Leistungsbewertung muss die in Abschnitt 7.2 definierten Leistungskenngrößen in ausreichender Genauigkeit nach Anforderung 3 bereitstellen.

Grundlage ist das in Abschnitt 5.5 eingeführte *Systemmodell der Nachrichtenklassen*. Demnach wird ein Binding \mathfrak{b} auf der Anwendungsschicht zerlegt in mehrere Nachrichtenklassen $\mathfrak{m} \in \mathfrak{M}_{\mathfrak{b}}$ auf der Bitübertragungsschicht, die zusammen eine Transition bilden. Jede Nachrichtenklasse $\mathfrak{m} \in \mathfrak{M}_{\mathfrak{b}}$ besitzt einen Weg $\mathfrak{W}_{\mathfrak{m}}$ durch das Netzwerk unter Verwendung von Ports \mathfrak{P} , Kanälen \mathfrak{C} und Routern \mathfrak{R} auf der Bitübertragungsschicht. Für diese Elemente werden die Leistungskenngrößen allgemeingültig anhand eines beliebigen Elementes $\mathfrak{e} \in \mathfrak{W} \cup \mathfrak{M} \cup \mathfrak{B}$ definiert.

DURCHSATZ Unter dem *Durchsatz* (en. *Throughput*) $\vartheta_{\mathfrak{e}}$ in Msg/s wird in der Leistungsbewertung meist die Anzahl der bearbeiteten Aufträge pro Zeiteinheit verstanden, die auch als *Abgangsrate* bezeichnet wird [9, 112].

DATENDURCHSATZ: Bei Netzwerken wird meist vom *Datendurchsatz* Θ in bit/s gesprochen, der die übertragenen Daten pro Zeiteinheit erfasst [39, 195]. Synonym zum Datendurchsatz wird in dieser Arbeit auch der Begriff *Last* verwendet. Der Datendurchsatz bestimmt sich aus der Abgangsrate $\vartheta_{\mathfrak{e}}$ und entweder der Nachrichtengröße S_{phy} oder der Informationsgröße S_{app} , je nachdem ob das betrachtete Element \mathfrak{e} zu der Bitübertragungsschicht oder Anwendungsschicht gehört.

$$\Theta_{\mathfrak{e}} = \begin{cases} \vartheta_{\mathfrak{e}} \cdot S_{\text{phy};\mathfrak{e}}, & \mathfrak{e} \in \mathfrak{M} \cup \mathfrak{W}; \\ \vartheta_{\mathfrak{e}} \cdot S_{\text{app};\mathfrak{e}}, & \mathfrak{e} \in \mathfrak{B}. \end{cases} \quad (7.1)$$

² Voraussetzung ist eine streng monoton wachsende Verteilungsfunktion [34].

AUSLASTUNG: Die *Auslastung* (en. *Utilisation*) ρ_{ϵ} in Prozent definiert das Verhältnis zwischen vorhandener und genutzter Kapazität. Für einen Bearbeiter mit der *Bedienrate* μ_{ϵ} folgt

$$\rho_{\epsilon} = \frac{\vartheta_{\epsilon}}{\mu_{\epsilon}}. \quad (7.2)$$

Bei einem Kanal ist der maximale Datendurchsatz $\Theta_{\max;\epsilon}$ bekannt und oft wird näherungsweise

$$\rho_{\epsilon} = \frac{\Theta_{\epsilon}}{\Theta_{\max;\epsilon}} = \frac{\vartheta_{\epsilon} \cdot \overline{S}_{\text{phy};\epsilon}}{\mu_{\epsilon} \cdot \overline{S}_{\text{phy};\epsilon}}. \quad (7.3)$$

benutzt³. Die Auslastung ist für Bindings nicht definiert.

ERFOLGS- UND FEHLERWAHRSCHEINLICHKEIT: Das Verhältnis der Abgangsrate zur Ankunftsrate definiert die *Erfolgswahrscheinlichkeit* p_{ϵ}^{+} , dass das Element verlassen wird

$$p_{\epsilon}^{+} = \frac{\overline{\vartheta}_{\epsilon}}{\overline{\lambda}_{\epsilon}} \quad (7.4)$$

und komplementär die *Fehlerwahrscheinlichkeit* p_{ϵ}^{-}

$$p_{\epsilon}^{-} = 1 - p_{\epsilon}^{+}. \quad (7.5)$$

VERZÖGERUNGSZEIT: Jedes Element ϵ besitzt ferner eine *Verzögerungszeit* (en. *Delay*) τ_{ϵ} in Sekunden, die mit p_{ϵ}^{-} unendlich ist⁴. Bei einem Binding bezieht sich diese Verzögerungszeit auf die Informationsübertragung von der initiierenden Anwendung bis zur Ankunft bei den empfangenden Anwendungen. Dies ist üblicherweise die Verzögerungszeit vom Sender zu einem Empfänger, außer beim beantworteten Übertragungsdienst, dort dauert die Verzögerungszeit von der Anfrage bis zum Eintreffen der Antwort auf der Anwendungsschicht des Senders.

Diese Leistungskenngrößen ändern sich dynamisch abhängig vom aktuellen Zustand der Geräte und des Netzwerks. Deshalb werden sie als Zufallsprozesse über

3 Dies ist eine Näherung, da $\mu_{\epsilon} \cdot \overline{S}_{\text{phy};\epsilon} \neq \Theta_{\max;\epsilon}$ und μ_{ϵ} nicht konstant ist. So gibt es bei den Protokollen zusätzliche Ruhezeiten zwischen den Nachrichten (z. B. β_1, β_2 bei LON), die nicht zu $S_{\text{phy};\epsilon}$ zählen und die reale Bedienrate reduzieren. Entsprechend ist bei dieser Abschätzung eine hundertprozentige Auslastung nicht zu erreichen.

4 Damit sind prinzipiell für $p_{\epsilon}^{-} > 0$ keine Momente mehr für die Verzögerungszeit definiert. In der Implementation ist die PDF für die Verzögerungszeit nur für $t < \infty$ definiert und für den Dirac bei Unendlich wird die Fehlerwahrscheinlichkeit mit abgespeichert. Die Momente werden dann innerhalb $t < \infty$ bestimmt. Die unendlichen Verzögerungszeit vereinfacht später die Fehleranalyse.

der Zeit oder den Realisierungen aufgenommen. Hierbei wird statistische Unabhängigkeit der Kenngrößen nicht gefordert, sondern neben der Dichtefunktion und dem Mittelwert auch die Autokorrelation erfasst. Die implementierte analytische Analyse kann hiervon nur die Mittelwerte abschätzen, so dass die Dichtefunktion und die Autokorrelationsfunktion durch Simulation bestimmt werden müssen. Aus Performancegründen erstellt die implementierte Simulation diese Verteilungen nur für die Leistungskenngrößen, die in Kapitel 8 ausgewertet werden⁵.

7.3. ZUSAMMENHANG ZWISCHEN DEN MODELLTEILEN UND DER ANKUNFTSRATENMODIFIER u_m

Die Ergebnisse der Leistungsbewertung haben Auswirkungen auf die Parametrisierung des System- und Verkehrsmodells. In Unterabschnitt 5.5.3 wurde der *Ankunftsratemodifier* u_m als Verhältnis zwischen der Ankunftsrate der Nachrichtenklasse $m \in \mathfrak{M}_b$ des Bindings b eingeführt. Aus dem Verkehrsmodell ist die mittlere Ankunftsrate eines Bindings bekannt, so dass die mittlere Ankunftsrate einer Nachrichtenklasse dieses Bindings wie folgt lautet

$$\bar{\lambda}_m = u_m \cdot \bar{\lambda}_b. \quad (7.6)$$

Die Übertragung ist allerdings nicht fehlerfrei und jedes durchlaufene Wegelement $w \in \mathfrak{W}_m$ ist mit einer Fehlerwahrscheinlichkeit p_w^- belegt. Diese Fehlerwahrscheinlichkeiten kombinieren sich auf dem Teilweg $\mathfrak{W}_{p_o, w} \subseteq \mathfrak{W}_m$ von dem Ursprungsort p_o der Nachrichtenklasse bis zu dem Wegelement $w \in \mathfrak{W}_m$. Nach Prämisse 7 ist dieser Teilweg eindeutig und unverzweigt. Unter der Annahme, dass die Fehlerwahrscheinlichkeiten der Elemente statistisch unabhängig sind, wird das Wegelement w mit der Wahrscheinlichkeit $p_{l;w}^+$ erfolgreich erreicht und mit $p_{o;w}^+$ erfolgreich verlassen, wobei

$$p_{o;w}^+ = \prod_{w^* \in \mathfrak{W}_{p_o, w}} (1 - p_{w^*}^-) \quad \text{und} \quad p_{l;w}^+ = \frac{p_{o;w}^+}{1 - p_w^-}. \quad (7.7)$$

Daraus folgt die Erfolgswahrscheinlichkeit p_m^+ der Nachrichtenklasse m . Sie ist für die unterschiedlichen Empfängerdatenpunkte \mathfrak{G} unterschiedlich. Ist $i \in \mathfrak{G}$ ein Empfängerdatenpunkt hieraus und p_i der Zielpunkt des Gerätes mit i , so ist die Erfolgswahrscheinlichkeit der Nachrichtenklasse für diesen Port

$$p_m^+ | i = p_{o;p_i}^+. \quad (7.8)$$

⁵ Für die QoS-Anforderungen entsprechend Unterabschnitt 8.2.3 definiert sind.

Durch die Verluste liegt an den Wegelementen nicht die volle Ankunftsrate einer Nachrichtenklasse an, sondern die mittlere Ankunftsrate der Nachrichtenklasse \mathbf{m} am Wegelement \mathbf{w} reduziert sich auf

$$\bar{\lambda}_{\mathbf{m}|\mathbf{w}} = p_{\mathbf{I};\mathbf{w}}^+ \cdot \bar{\lambda}_{\mathbf{m}} = p_{\mathbf{I};\mathbf{w}}^+ \cdot u_{\mathbf{m}} \cdot \bar{\lambda}_{\mathbf{b}}. \quad (7.9)$$

Hieraus bestimmt sich die mittlere Ankunftsrate aller Nachrichtenklassen, die das Wegelement \mathbf{w} insgesamt passieren, zu

$$\bar{\lambda}_{\mathbf{w}} = \sum_{\mathbf{m} \in \mathfrak{M} \mid \mathbf{w} \in \mathfrak{W}_{\mathbf{m}}} \bar{\lambda}_{\mathbf{m}|\mathbf{w}} = \sum_{\mathbf{m} \in \mathfrak{M} \mid \mathbf{w} \in \mathfrak{W}_{\mathbf{m}}} p_{\mathbf{I};\mathbf{w}}^+ \cdot \bar{\lambda}_{\mathbf{m}} \quad (7.10)$$

mit der mittleren Nachrichtengröße

$$\bar{S}_{\text{phy};\mathbf{w}} = \frac{1}{\bar{\lambda}_{\mathbf{w}}} \sum_{\mathbf{m} \in \mathfrak{M} \mid \mathbf{w} \in \mathfrak{W}_{\mathbf{m}}} \bar{\lambda}_{\mathbf{m}|\mathbf{w}} \cdot \bar{S}_{\text{phy};\mathbf{m}}. \quad (7.11)$$

Bei einem Port wird zusätzlich zwischen ein- und ausgehendem Verkehr unterschieden. Besitzt \mathbf{p} einen Kanal \mathbf{c} als Vorgänger $\mathbf{w}_{i-1} \in \mathfrak{W}$, so ist

$$\bar{\lambda}_{\mathbf{I};\mathbf{w}=\mathbf{p}} = \sum_{\mathbf{m} \in \mathfrak{M} \mid (\mathbf{w} \in \mathfrak{W}_{\mathbf{m}} \wedge \mathbf{w}_{i-1} \in \mathcal{C})} \bar{\lambda}_{\mathbf{m}|\mathbf{w}} \quad \text{und} \quad \bar{\lambda}_{\mathbf{O};\mathbf{w}=\mathbf{p}} = \bar{\lambda}_{\mathbf{w}} - \bar{\lambda}_{\mathbf{I};\mathbf{w}}. \quad (7.12)$$

Die mittleren Ankunftsraten der durchlaufenden Elemente werden zur Parametrierung der analytischen Leistungsbewertung verwendet. Dafür wird an den Warteschlangen der Geräte der ausgehende Verkehr der Geräteports $\bar{\lambda}_{\mathbf{O};\mathbf{p}}$ nach Gleichung (7.12) angelegt. Aus den Ergebnissen der Leistungsbewertung wird nun der Ankunftsratenmodifier $u_{\mathbf{m}}$ bestimmt.

Im einfachsten Fall handelt es sich um einen unbestätigten, wiederholten Übertragungsdienst mit $r_{\mathbf{b}}^+ = \text{binding.retries}$ Wiederholungen. Dann werden für jede Nachricht auf der Anwendungsschicht $r_{\mathbf{b}}^+$ Nachrichten auf der Bitübertragungsschicht erzeugt und folglich ist der Ankunftsratenmodifier $u_{\text{rpt}} = r_{\mathbf{b}}^+$ für die einzige Nachrichtenklasse $\mathbf{m} = \text{rpt}$.

Bei einem bestätigten Nachrichtentyp gibt es einen Ankunftsratenmodifier $u_{\mathbf{m}}$ für jede Nachrichtenklasse. Angenommen das Binding \mathbf{b} hat nur einen Empfängerdatenpunkt \mathbf{i} , dann folgt eine Unicast-Adressierung mit drei Nachrichtenklassen für die erste Nachricht (AckD), die Bestätigungen (Ack) und möglichen Erinnerungen (Rem), die den Inhalt der ersten Nachricht enthalten. Die erste Nachricht wird immer gesendet und u_{AckD} ist 1. Bestätigungen werden gesendet, wenn die erste Nachricht oder eine Erinnerung beim Empfänger ankommt und

$$u_{\text{Ack}} = p_{\text{AckD}}^+ \cdot u_{\text{AckD}} + p_{\text{Rem}}^+ \cdot u_{\text{Rem}}. \quad (7.13)$$

Kommt die erste Nachricht oder die Bestätigung nicht an, wird bis zu $r_{\mathbf{b}}^-$ -mal eine Erinnerung gesendet, wobei $r_{\mathbf{b}}^- = \text{binding.failureRetries}$ ist. Für die erste

Erinnerung ist $\bar{\lambda}_{\text{Rem};1} = (1 - p_{\text{AckD}}^+ p_{\text{Ack}}^+) \bar{\lambda}_{\text{AckD}}$ und für jede folgende Erinnerung $\bar{\lambda}_{\text{Rem};n} = (1 - p_{\text{Rem}}^+ p_{\text{Ack}}^+) \bar{\lambda}_{\text{Rem};n-1}$. Dementsprechend ist

$$\begin{aligned} u_{\text{Rem}} &= (1 - p_{\text{AckD}}^+ \cdot p_{\text{Ack}}^+) \sum_{i=0}^{r-1} (1 - p_{\text{Rem}}^+ \cdot p_{\text{Ack}}^+)^i \\ &= (1 - p_{\text{AckD}}^+ \cdot p_{\text{Ack}}^+) \frac{1 - (1 - p_{\text{Rem}}^+ \cdot p_{\text{Ack}}^+)^{r_b^-}}{p_{\text{Rem}}^+ \cdot p_{\text{Ack}}^+}. \end{aligned} \tag{7.14}$$

Im Falle einer Gruppenadressierung wird durch eine Erinnerung nur von den ausstehenden Empfängern die Bestätigung eingefordert. Für einen einzelnen Empfänger ist die Transaktion beendet, sobald er seine Bestätigung erfolgreich abgesetzt hat. Damit gibt es aus Sicht des Empfängers keinen Unterschied zur Unicast-Adressierung und sein Ankunftsatenmodifier für die Bestätigungen $u_{\text{Ack};i}$, kann durch die isolierte Betrachtung dieser Verbindung als Unicast nach Gleichung (7.13) bestimmt werden. Da so lange Erinnerungen gesendet werden, bis die letzte Bestätigung eintrifft, lässt sich der Ankunftsatenmodifier der Erinnerungen u_{Rem} aus dem maximalen $u_{\text{Rem};i}$ der einzeln betrachteten Unicast-Adressierungen der Empfänger $i \in \mathfrak{G}$ nach Gleichung (7.14) abschätzen⁶.

Wird eine Gruppe mit mehr als 16 Empfängern adressiert, so wird jede Erinnerung getrennt von der ursprünglichen Nachricht gesendet und beide müssen ihr Ziel erreichen. Es ist also

$$\begin{aligned} u_{\text{AckD}} &= 1 + u_{\text{Rem}}, \\ u_{\text{Rem}} &= \max(u_{\text{Rem};i} \mid i \in \mathfrak{G}), \\ u_{\text{Ack};i} &= p_{\text{AckD};o,i}^+ \left(u_{\text{AckD}} + p_{\text{Rem};i}^+ \cdot u_{\text{Rem};i} \right) \quad \text{und} \\ u_{\text{Rem};i} &= \left(1 - p_{\text{AckD};o,i}^+ \cdot p_{\text{Ack};i}^+ \right) \frac{1 - \left(1 - p_{\text{AckD};o,i}^+ \cdot p_{\text{Rem};i}^+ \cdot p_{\text{Ack};i}^+ \right)^{r_b^-}}{p_{\text{AckD};o,i}^+ \cdot p_{\text{Rem};i}^+ \cdot p_{\text{Ack};i}^+}. \end{aligned} \tag{7.15}$$

Der Ankunftsatenmodifier beim beantworteten Übertragungsdienst folgt gemäß $u_{\text{Req}} = u_{\text{AckD}}$ und $u_{\text{Res}} = u_{\text{Ack}}$. Bei den authentifizierten Übertragungsdiensten ergeben sich die Ankunftsatenmodifier in analoger Weise unter Berücksichtigung der Abhängigkeiten der einzelnen Nachrichtenklassen.

⁶ Prinzipiell müsste die Kombinationen aller Teiltransaktionen (Eine Erinnerung und ihre Bestätigungen) aufgestellt werden, um die Wahrscheinlichkeiten zu bestimmen, in welcher Teiltransaktion welcher Empfänger seine Bestätigung erfolgreich absetzen kann. Dabei muss beachtet werden, dass die Fehlerwahrscheinlichkeiten der Empfänger nicht statistisch unabhängig sind, da sich Teilwege überdecken. Dies wiederum nutzt die vorgestellte Abschätzung aus, so dass sie ungenau wird, wenn mehrere Empfänger mit unterschiedlichen Wegen eine vergleichbar große Fehlerwahrscheinlichkeit haben.

Der Ankunftsratenmodifier bestimmt sich demnach aus den Fehlerwahrscheinlichkeiten der durchlaufenden Elemente. Diese werden jedoch erst in der Leistungsbewertung bestimmt, welche die Ankunftsratenmodifier zur Parametrierung nutzt. Hierin besteht die erste Abhängigkeit zwischen parametrischen Teilen des Systemmodells der Nachrichtenklassen und der Leistungsbewertung.

Die zweite Abhängigkeit bezieht sich auf das Verkehrsmodell. Hier wurde in Gleichung (6.6) die Fehlerwahrscheinlichkeit $p_{\mathfrak{b},\mathfrak{o},i}^-$ für den Verlust der Nachricht auf dem Weg von i zu \mathfrak{o} benutzt. Diese bestimmt sich nur aus der Fehlerwahrscheinlichkeit der Nachrichtenklassen, die die Information übertragen, und nicht aus der vollständigen Transaktion. Für die üblichen Übertragungsdienste ohne ihre authentifizierten Pendants gilt

$$p_{\mathfrak{b},\mathfrak{o},i}^- = \begin{cases} (p_{\text{rpt};\mathfrak{o},i}^-)^{r_{\mathfrak{b}}^+}, & \text{wiederholt;} \\ p_{\text{AckD};\mathfrak{o},i}^- (p_{\text{Rem};\mathfrak{o},i}^-)^{r_{\mathfrak{b}}^-}, & \text{bestätigt } \|\mathfrak{G}\| \leq 16; \\ (p_{\text{AckD};\mathfrak{o},i}^-)^{r_{\mathfrak{b}}^-+1}, & \text{bestätigt } \|\mathfrak{G}\| > 16; \\ (1 - p_{\text{Req};\mathfrak{o},i}^+ p_{\text{Res};i}^+) (1 - p_{\text{Rem};\mathfrak{o},i}^+ p_{\text{Res};i}^+)^{r_{\mathfrak{b}}^-}, & \text{beantwortet } \|\mathfrak{G}\| \leq 16; \\ (1 - p_{\text{Req};\mathfrak{o},i}^+ p_{\text{Res};i}^+)^{r_{\mathfrak{b}}^-+1}, & \text{beantwortet } \|\mathfrak{G}\| > 16. \end{cases} \quad (7.16)$$

Diese beiden Abhängigkeiten werden durch die mehrfache Iteration in der Fixpunkt-Analyse aufgelöst, wobei als Startbedingung ein verlustfreies System mit $p_{\mathfrak{c}}^+ = 1$ angenommen wird. Der Algorithmus 2 bestimmt letztendlich die Leistungskenngrößen eines Netzwerks.

ALGORITHMUS 2. Algorithmus der analytischen Leistungsbewertung.

1. Wiederhole über n
 2. Aktualisiere das Verkehrsmodell
 3. Berechne die Ankunftsratenmodifier $u_{\mathfrak{m}} \forall \mathfrak{m} \in \mathfrak{M}$ analog zu Gleichung (7.15)
 4. Berechne die Ankunftsrate an allen Netzwerkelementen nach Gleichung (7.10)
 5. Analytische Leistungsbewertung nach [15, 16, 108, 176]
 6. Berechne die Verluste der Nachrichtenklassen nach Gleichung (7.16)
 7. Bis $|\sum_{\mathfrak{c} \in \mathfrak{C}} \Theta_{\mathfrak{c};n} - \sum_{\mathfrak{c} \in \mathfrak{C}} \Theta_{\mathfrak{c};n-1}| < \varepsilon$
-

7.4. ZUSAMMENFASSUNG

Die Grenzen der ausgewählten Methoden liegen vor allem in der Natur der Methoden selbst. So setzt die analytische Analyse exponentiell verteilte Ankunftsprozesse und Bedienraten auf der Bitübertragungsschicht voraus und vernachlässigt damit insbesondere das burstartige Verhalten einiger Übertragungsdienste.

Auch für die weitere Analyse sind die resultierenden Mittelwerte unzureichend und höhere statistische Momente notwendig.

TABELLE XIV.

Ergebnisse der Leistungsbewertung für das Beispiel mit 2 Etagen à 10 Räumen ($\bar{\Theta}_\epsilon$ nach Gl. (7.1), $\bar{\Theta}_{\Sigma;b}$ nach Gl. (8.24), $\bar{\Theta}_{\Pi;b}$ nach Gl. (8.25) (vgl. Abb. 8.2); $[\bar{\lambda}_\epsilon] = \frac{\text{Msg}}{\text{s}}$; $[\bar{\vartheta}_\epsilon] = \frac{\text{Msg}}{\text{s}}$; $[\bar{\Theta}_\epsilon] = \frac{\text{bit}}{\text{s}}$; $[\bar{\Theta}_{\Pi;b}] = \frac{\text{bit}}{\text{Msg}}$; $[\bar{\rho}_\epsilon] = \%$; $[\rho_\epsilon^+] = \%$; $[\bar{\tau}_\epsilon] = \text{ms}$).

ϵ	Größe	Dienst	$\bar{\lambda}_\epsilon$	$\bar{\vartheta}_\epsilon$	$\bar{\Theta}_\epsilon$	$\bar{\Theta}_{\Sigma;b}$	$\bar{\Theta}_{\Pi;b}$	ρ_ϵ^+	$\bar{\tau}_\epsilon$
b1	Beleuchtungsstärke	Unack	0.064	0.055	0.887	7.06	127	87.22	31.55
b2	Beleuchtungsstärke	Rep 3x	1.970	1.965	31.44	615	313	99.77	38.63
b3	Jalousiestand	Unack	1.965	1.688	54.02	236	140	85.90	35.01
b4	Lampendimmung	Rep 3x	1.965	1.960	15.68	567	289	99.75	40.28
b5	Temperatur	Unack	0.017	0.014	0.229	1.73	121	85.81	31.49
b6	Ventilöffnung	Unack	0.017	0.014	0.232	1.73	120	86.73	31.99
b7	Anwesenheit	Ack	1.2 E^{-4}	1.2 E^{-4}	9.3 E^{-4}	0.03	239	100.0	49.27
b8	Fensteröffnung	Ack	5.9 E^{-5}	5.9 E^{-5}	4.7 E^{-4}	0.01	165	100.0	47.37
b9	Fensteröffnung	Ack	1.2 E^{-4}	1.2 E^{-4}	9.3 E^{-4}	0.03	239	100.0	46.02
b10	Rauchmeldung	Ack	0.200	0.200	1.598	33.6	168	99.99	49.38
b11	Alarm	Ack, Auth	4.000	3.990	31.92	5052	1266	99.74	158.9
c1	Backbone	-	50.00	50.00	5581	$\bar{\rho}_\epsilon = 3.95$	100.0	-	-
c2	Kanal 2	-	167.5	160.5	16706	$\bar{\rho}_\epsilon = 63.6$	95.79	-	-

Diese können durch die Simulation bestimmt werden. Obwohl die Simulation auf eine hohe Performance hin ausgerichtet ist, benötigt sie deutlich länger, um Ergebnisse im statistisch signifikanten Umfang zu sammeln. Insbesondere wenn die Bindings in einem Netzwerk sehr unterschiedliche Ankunftsraten haben, müssen sehr viele Ereignisse für die aktiven Bindings bearbeitet werden, bis ausreichend Stichproben bei den inaktiven vorliegen. Dass diese Unterschiede in den Ankunftsraten nicht ungewöhnlich für die Gebäudeautomation sind, zeigt das Verkehrsmodell für den Büroraum in Tabelle XIII. Die Simulation für die Ergebnisse in Tabelle XIV mit zwei Fluren und jeweils 10 Büroräumen dauert zwei Stunden⁷, in denen über eine Milliarde Ereignisse bearbeitet werden und ein Monat virtuelle Zeit vergeht. Dazu im Vergleich benötigt die analytische Berechnung drei Sekunden. Der Speicherverbrauch der Simulation liegt bei 24 MByte und bei der Analyse bei 16 MByte, wobei bereits 15 MByte auf das Modell abfallen. Um die Simulation gemäß Anforderung 1 effizienter zu gestalten, könnten Ansätze zur Simulation seltener Ereignisse [17, 107] helfen⁸. Der Anwender braucht sich über dieses Problem keine Gedanken machen. Im Sinne der Anforderungen 2, 3 und 5

⁷ Diese Simulation hatte höhere Anforderung bzgl. der statistisch Signifikanz. Bei Standardereignissen benötigt sie 43 Minuten.

⁸ Im Beispiel ist es bereit sinnvoll, die Ankunftsraten der Bindings b7 bis b9 zu verhundertfachen. Bei nahezu identischen Ergebnissen fällt die Simulationszeit auf 30 Sekunden.

passt die Simulation automatisch ihre Laufzeit an, bis Stichproben im statistisch signifikanten Umfang für eine Qualitätsbewertung zusammengetragen wurden.

Für den Großteil der Bindings bietet die implementierte Leistungsbewertung dennoch eine effiziente und genaue Analyse gemäß den Anforderungen 1 und 3. Die für die weitere Analyse in Kapitel 8 notwendigen Leistungskenngrößen wurden definiert und die Zusammenhänge zwischen dem Systemmodell, Verkehrsmodell und der Leistungsbewertung offen gelegt. Damit kann, wie in Abschnitt 3.3 vorgestellt, die Fehleranalyse und Nutzerberatung im folgenden Kapitel 8 anschließen. Dort werden auch die Ergebnisse der Tabelle XIV als praktisches Beispiel einfließen und detaillierter interpretiert.

KAPITEL

8.

FEHLERANALYSE UND NUTZERBERATUNG

*Wer die Ursache nicht kennt,
nennt die Wirkung Zufall.*

Werner Mutsch

8.1. EINFÜHRUNG

IM letzten Kapitel wurden die Leistungskenngrößen der Elemente eines Netzwerks definiert. Dabei wurde bewusst darauf verzichtet, die Ergebnisse in Tabelle XIV zu interpretieren oder genau zu erklären, wie sich zum Beispiel die Verzögerungszeit eines Bindings zusammensetzt. Dies entspricht der realen Situation, in der die Simulation Verteilungen für Bindings und Geräte erzeugt, ohne dass der Anwender auf Anhieb die Zusammenhänge versteht. Für ein Netzwerk mit fünf Kanälen, 100 Geräten und 400 Bindings liegen nach acht Sekunden über 5500 Ergebnisse vor. Dadurch ist es selbst für einen Anwender, der Detailkenntnisse zur Leistungsbewertung besitzt, aufwendig, entsprechende Schlussfolgerungen zu ziehen. Deshalb ist es nach den Anforderungen 1 und 2 notwendig, den Anwender bei der Auswertung der Ergebnisse zu unterstützen.

Die Auswertung teilt sich nach Abbildung 3.2 in die Fehleranalyse und Nutzerberatung, die aufgrund ihres Zusammenhangs in diesem Kapitel gemeinsam behandelt werden. Die *Fehleranalyse* bewertet die Leistungskenngrößen, um Entwurfsfehler zu identifizieren, die als Abweichung der Eigenschaften des Netzwerkentwurfs von den gestellten Qualitätsanforderungen 2 bis 5 definiert sind. Allerdings sind diese Qualitätsanforderungen wie Betriebssicherheit oder Echtzeitverhalten noch unkonkret und müssen zerlegt und quantifiziert werden, wozu im nächsten Abschnitt die *Dienstqualität* eingesetzt wird.

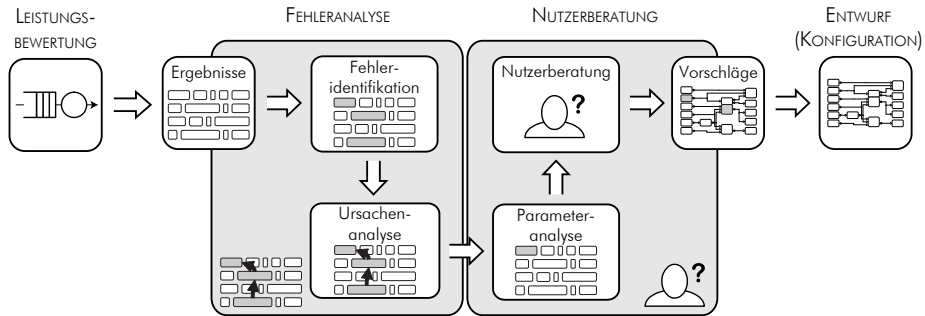


ABBILDUNG 8.1. Detaillierte Struktur der Fehleranalyse und Nutzerberatung.

Daraufhin können die Ergebnisse der Leistungsbewertung gegen die gestellten Anforderungen der Dienstqualität geprüft und Entwurfsfehler identifiziert werden. Zur Analyse der Fehlerursachen¹ werden die Entwurfsfehler zum einen priorisiert, so dass der Anwender kritische Entwurfsfehler sofort identifizieren kann und diese nicht durch unbedeutende überdeckt werden. Der zweite Teil konzentriert sich auf die Analyse der Fehlerzusammenhänge mit Hilfe von Fehlerbäumen.

Eine ähnliche Aufgabe besitzt die *Netzwerkdiagnose* im Betrieb von Netzwerken. Sie überwacht als Teil des *Netzwerkmanagements* die Netzwerkleistung durch Monitoring und Messungen. Werden gehäuft Übertragungsfehler beobachtet, versucht sie, die Ursachen festzustellen, sie zu beheben oder durch Rerouting zu vermeiden. Hierzu werden Methoden der künstlichen Intelligenz angewendet [116, 207], wozu unter anderem die modellbasierte Diagnose und das modellbasierte Schließen [80, 95, 183] zählen. Die notwendigen Entscheidungsbaume und Regeln werden im Betrieb angelernt. Dies ist in der Konfigurierung aufgrund des nach Prämisse 2 fehlenden Netzwerks nicht möglich. Stattdessen bieten aber die bekannten Modelle und Leistungskenngrößen umfangreiche Informationen. So nutzen FAHMY & DOULIGERIS [61, 62] bereits Simulationsergebnisse zum Trainieren ihres Expertensystems zur Entwurfsberatung und KESHAV & SHARMA [118] nutzen sie zur Netzwerkdiagnose im Entwurf. In dieser Arbeit wird der Kontext mit Hilfe von Fehlerbäumen strukturiert [183], so dass dem Anwender in der *Nutzerberatung* in Abschnitt 8.4 zielführende Lösungsvorschläge unterbreitet werden können. Darüber hinaus kann sich der Anwender bei der Parametrierung gängiger Lösungen beraten lassen.

¹ In der Fehlerklassifikation wird häufig zwischen Fehler, Fehlerwirkung und Fehlerursache (Fehlerzustand / Defekt) unterschieden [43, 207] oder noch diverser klassifiziert [198]. Eine eindeutige Zuordnung ist bei komplexen Systemen meist schwierig, da Fehler, Wirkung und Ursache stark zusammenhängen und zum Teil nicht zu isolieren sind. Deshalb wird hierauf bewusst verzichtet und der Begriff Fehler nicht nur für nicht erfüllte Qualitätsmerkmale, sondern auch für die Ursachen verwendet. Deutlicher wird dies im Unterabschnitt 8.3.2.

TABELLE XV.
QoS-Charakteristika nach ISO/IEC 13236 [101].

Gruppe	QoS-Charakteristikum	Spezialisierung
Zeit	Verzögerungszeit	
Übereinstimmung	zeitliche Übereinstimmung	zeitliche Datenübereinst.
Kapazität	Datendurchsatz	Nutzdatendurchsatz Subsystemdurchsatz
Integrität	Restfehler Übertragungsfehler	
Sicherheit	Sicherheit	
Kosten	Kosten	Kommunikationskosten
Zuverlässigkeit	Verfügbarkeit Zuverlässigkeit Fehlertoleranz	Kanalverfügbarkeit

Die grundlegenden Konzepte der vorgestellten Fehleranalyse wurden in [177] veröffentlicht und mit der Unterstützung der Diplomarbeit von BORIS KOLMAN [124] implementiert. Sie beziehen sich speziell auf das Beispiel LON, sind dennoch grundsätzlich auf BACnet und Konnex übertragbar, indem das Formelmateriale an das andere Protokollverhalten angepasst wird.

8.2. FEHLERIDENTIFIKATION

8.2.1. DIENSTQUALITÄT (QUALITY OF SERVICE)

Dienstqualität beziehungsweise *Quality of Service* (QoS) ist bei Netzwerken ein häufig verwendeter Begriff mit entsprechend vielseitiger Bedeutung, von der reinen Zusammenfassung von Gütekriterien [101, 103] bis zu praktischen Maßnahmen zur Gewährleistung des QoS [24]. Für diese Arbeit wird die Dienstqualität als Bewertungskriterium im Sinne der ISO/IEC 13236 [101] benutzt, die Dienstqualität als »Menge von Güten, die auf das gesamte Verhalten eines oder mehrerer Objekte bezogen sind« erklärt.

Zur Quantifizierung der Dienstqualität definiert die ISO/IEC 13236 [101] verschiedene *QoS-Charakteristika*. Hierbei wird zwischen *generischen QoS-Charakteristika*, ihren *Ableitungen* und ihren *Spezialisierungen* unterschieden. Als Ableitung können mathematische, statistische Funktionen dienen, wie der minimale, mittlere oder maximale Durchsatz; wohingegen Spezialisierungen den Anwendungsfall eingrenzen. Tabelle XV gibt Beispiele für die wichtigsten Charakteristika der ISO/IEC 13236 [101] wieder.

Ein Anwender oder Betreiber stellt an diese QoS-Charakteristika spezifische Anforderungen. Sie werden oft als *QoS-Anforderungen* (en. *QoS Requirement*) bezeichnet, in konkreten Werten auch als *QoS-Parameter* und selten als *QoS-Syntax*. Diese QoS-Anforderungen müssen mit einer bestimmten Garantie vom System erfüllt werden. Hierbei werden verschiedene *Garantieklassen* unterschieden [63, 208]: die *bestmögliche Bereitstellung* (en. *Best-Effort Service*), die keine Garantien bietet, aber auch keine Überwachungsmechanismen benötigt sowie die *garantierte Bereitstellung* (en. *Guaranteed Service*), die weiter in *statistische (weiche) Garantie* (en. *Soft Guarantees*) und *deterministische (harte) Garantie* (en. *Hard Guarantees*) unterteilt wird.

Deterministische Garantien erfordern, dass die QoS-Anforderungen für alle Fälle nachweislich erfüllt sind. Da die in der Gebäudeautomation verwendeten CSMA-Netzwerke nicht-deterministisch sind, können solche Garantien nicht ohne zusätzlichen technischen Aufwand gegeben werden. Sie sind bei den meisten Anwendungen in der Gebäudeautomation auch nicht notwendig.

Bei statistischen Garantien müssen die Anforderungen nur für einen Anteil ε aller Fälle erfüllt werden. Hierbei ist die Häufung einzelner Ausfälle nicht ausgeschlossen. Ein Ausfall wird im Weiteren als *QoS-Defekt* bezeichnet.

Da QoS-Defekte häufig korreliert auftreten [26, 46, 72] und die meisten Prozesse wie Regelungen nur kurze QoS-Defektfolgen tolerieren, sind statistische Garantien nicht immer ausreichend. Deshalb fordern *schwach harte Garantien* (en. *Weakly Hard Guarantees*) [6, 230, 237] die Gültigkeit von statistischen Garantien innerhalb beschränkter, gleitender Zeitintervalle [6, 162] oder aufeinander folgender Realisierungen. Den letzten Fall bezeichnet man als *(m, k)-harte Garantie* (en. *(m, k)-firm Guarantees*) [78, 79, 128, 237], da maximal $(k - m)$ QoS-Defekte innerhalb von k aufeinander folgenden Realisierungen toleriert werden.

8.2.2. ZUSAMMENHANG DER DIENSTQUALITÄT MIT DEN QUALITÄTSMERKMALEN

In Abschnitt 2.3 wurden für einen Netzwerkentwurf die *Qualitätsmerkmale*: Garantierbarkeit, Betriebssicherheit, Echtzeitverhalten, Verfügbarkeit und angemessene Kosten herausgearbeitet. Diese Qualitätsmerkmale beziehen sich auf das durch das Netzwerk beeinflusste Anwendungsverhalten, während die Dienstqualität Anforderungen von Einzelverbindungen definiert. Damit besitzen Qualitätsmerkmale und QoS-Charakteristika unterschiedliche Abstraktionslevels und Bezugssysteme, wodurch eine quantitative Abbildung erschwert wird.

Dementsprechend stehen Echtzeit- und Verzögerungsverhalten, Betriebssicherheit und Integrität sowie Kosten und Kapazität zwar im Zusammenhang zueinander, dieser ist allerdings stark durch die spezifische Anwendung im Einzelfall geprägt. So ist die Betriebssicherheit bei dem Rauchmelder $\delta 12$ wichtiger als beim Temperatursensor $\delta 5$ und erfordert Integrität und Echtzeitverhalten. Aus

diesem Grund wird auf eine Zuordnung und Quantifizierung der Qualitätsmerkmale durch QoS-Charakteristika verzichtet und stattdessen gefordert:

PRÄMISSE 16: QUANTIFIZIERBARKEIT DER QUALITÄTSMERKMALE

Die geforderten Qualitätsmerkmale werden vollständig als Anforderungen an die Dienstqualität für einzelne Verbindungen spezifiziert.

Damit gilt ein Netzwerkentwurf als frei von Entwurfsfehlern, wenn alle Anforderungen an die Dienstqualität erfüllt sind. Der folgende Abschnitt konkretisiert, wie die QoS-Anforderungen durch den Anwender definiert werden, während der übernächste Abschnitt präzisiert, was als QoS-Fehler gilt.

8.2.3. DEFINITION DER VERWENDETEN QoS-CHARAKTERISTIKA, QoS-ANFORDERUNGEN UND QoS-GÜTEKLASSEN

Aus den in Tabelle XV aufgeführten QoS-Charakteristika führen THOMESSE [212] und SOUCEK & SAUTER [206] die Zeit, Kapazität und Integrität als besonders wichtig für Feldbussysteme auf. Diese QoS-Charakteristika werden in dieser Arbeit durch die folgenden Spezialisierungen konkretisiert:

ZEIT

- *Verzögerungszeit* eines Bindings τ_b : Anwendungs-zu-Anwendungs-Verzögerung der Informationsübertragung nach Abschnitt 7.2. Dies ist wichtig für Qualitätsmerkmal 4 (Echtzeitverhalten) und teilweise für Qualitätsmerkmal 3 (Betriebssicherheit).
- *Jitter* eines Bindings $\sigma_{\tau_b}^2$: Varianz der Verzögerungszeit. Jitter sollte insbesondere bei isochronen Datenströmen gering sein, zum Beispiel bei zyklischen Regelungen. Dies ist für Qualitätsmerkmal 4 wichtig.

KAPAZITÄT

- *Auslastung* eines Kanals ρ_c : Auslastung eines Kanals im Verhältnis zu seiner maximalen Bitrate $\Theta_{\max;c}$ nach Gleichung (7.3). Die Auslastung im Arbeitspunkt dient Qualitätsmerkmal 6 (angemessene Kosten).

INTEGRITÄT

- *Fehlerwahrscheinlichkeit* eines Bindings p_b^- : Wahrscheinlichkeit für den Totalverlust der Informationsübertragung unter gegebenenfalls mehrmaliger Wiederholung nach Abschnitt 7.3. Wichtig für die Qualitätsmerkmale 3 und 5 (Verfügbarkeit) und mit Einfluss auf Qualitätsmerkmal 4.

THOMESSE [212] benennt ferner die folgenden üblichen QoS-Anforderungen für ein beliebiges QoS-Charakteristikum $\xi \in \{\tau_b, \sigma_{\tau_b}^2, \rho_c, p_b^-\}$

- oberer Grenzwert ξ^{\cap} ,
- obere Warngrenze $\xi^{\cap} \leq \xi^{\cap}$,
- Arbeitspunkt $\xi^{\circ} \leq \xi^{\cap}$,
- untere Warngrenze $\xi^{\cup} \leq \xi^{\circ}$,
- unterer Grenzwert $\xi^{\cup} \leq \xi^{\cup}$.

Diese QoS-Anforderungen können nach Bedarf für die aufgeführten QoS-Charakteristika τ_b , $\sigma_{\tau_b}^2$, ρ_c festgelegt werden. Die Fehlerwahrscheinlichkeit p_b^- ist in ihren Realisierungen eine binäre Größe, entweder eine Übertragung ist erfolgreich (0) oder nicht (1). Deshalb gilt für sie $p_b^{-\cap} = p_b^{-\cap} = p_b^{-\circ} = p_b^{-\cup} = p_b^{-\cup} = 1$. Die Gültigkeit der Grenzen wird über die Garantieklassen bestimmt.

Die Garantieklassse und ihre Parameter müssen zusätzlich zu den QoS-Anforderungen festgelegt werden. Deterministische Garantien sind aufgrund des Netzwerkverhaltens auszuschließen und nur statistische Garantien oder (m, k) -harte Garantien zulässig. Der nächste Abschnitt führt ein, wie die QoS-Anforderungen anhand dieser Garantien bewertet werden. Dazu ergänzt der Unterabschnitt 8.2.6, wie dies gemäß den Anforderungen 1 und 5 effizient und automatisch realisiert werden kann.

8.2.4. FEHLERIDENTIFIKATION MITTELS QoS-ANFORDERUNGEN

Die Abbildung 8.2(a) stellt das Histogramm der Verzögerungszeit des Bindings **b5** aus der Leistungsbewertung dar. Die deutliche Erlang-Verteilung ist häufig zu messen [69] und liegt auch den anderen Übertragungsdiensten in Abbildung 8.2 zugrunde, bei denen nur Wiederholungen auftreten. Ist das Netzwerk gering ausgelastet, sind die Warteschlangen der Geräte meist leer und die normalverteilten Bearbeitungszeiten aus Abschnitt 6.3 dominieren. Bei wachsender Auslastung nehmen die Wartezeiten deutlich zu, die Verteilung wird rechtsschief, bis sie in eine sehr breite Normalverteilung übergeht. In diesem Beispiel ist das Netzwerk gering ausgelastet und die Nachrichten werden im Mittel mit 27 ms schnell übertragen. Anhand dieses Beispiels und des allgemeinen QoS-Charakteristikums ξ wird erklärt, wie die Dienstqualität unter unterschiedlichen Garantieklassen bewertet wird.

Bei statistischen Garantien muss zu den QoS-Anforderungen an ξ die erlaubte Toleranz $\varepsilon_{\xi x}$ angegeben werden. Nur $\varepsilon_{\xi^{\circ}}$ ist mit 50% bereits vorgegeben. Die Grenzen besitzen unterschiedliche Bedeutungen. Es gilt als *QoS-Fehler*, wenn

$$\varepsilon_{\xi^{\cap}} \leq \mathbb{P}[\xi \geq \xi^{\cap}] = 1 - \mathbf{F}_{\xi}(\xi^{\cap}) \vee \varepsilon_{\xi^{\cup}} \leq \mathbb{P}[\xi \leq \xi^{\cup}] = \mathbf{F}_{\xi}(\xi^{\cup}) \quad (8.1)$$

und als *QoS-Warnung*, wenn

$$\varepsilon_{\xi^{\cap}} \leq \mathbb{P}[\xi \geq \xi^{\cap}] = 1 - \mathbf{F}_{\xi}(\xi^{\cap}) \vee \varepsilon_{\xi^{\cup}} \leq \mathbb{P}[\xi \leq \xi^{\cup}] = \mathbf{F}_{\xi}(\xi^{\cup}). \quad (8.2)$$

Unter diesen Bedingungen werden die aus der Leistungsbewertung resultierenden Dichtefunktionen bewertet. In Abbildung 8.2(a) liegen zum Beispiel 5%

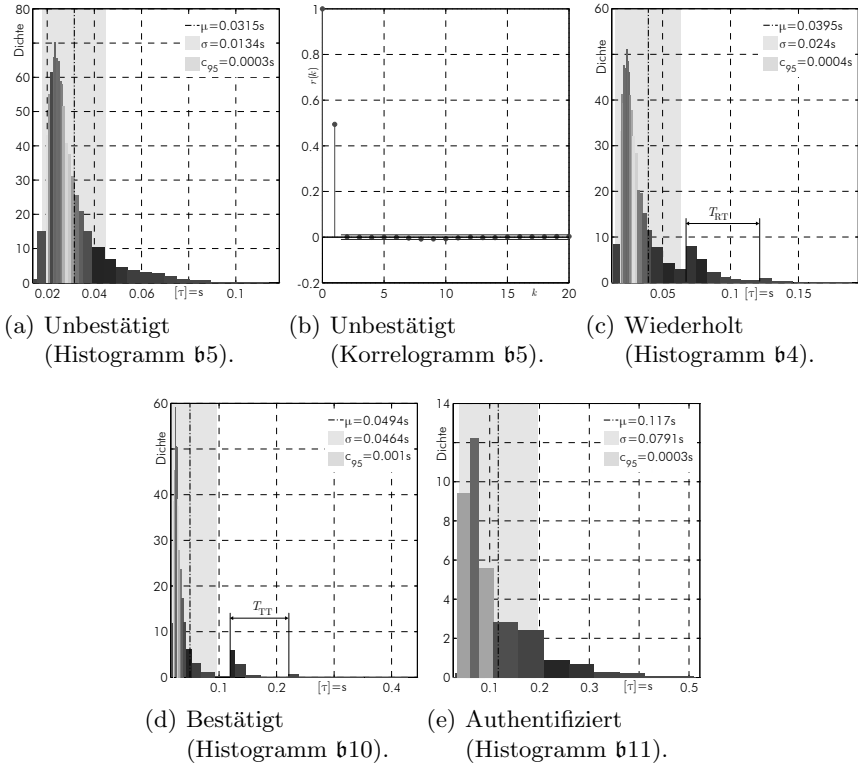


ABBILDUNG 8.2. Typische Verzögerungszeiten unterschiedlicher Übertragungsdienste.

der Verteilung unter 20 ms. Wird die untere Grenze τ_b^ψ auf 20 ms gesetzt mit einer Toleranz von $\varepsilon_{\tau_b^\psi} = 4\%$, so wird dies als gescheiterte QoS-Anforderung und QoS-Fehler identifiziert, allerdings nicht bei $\varepsilon_{\tau_b^\psi} = 6\%$.

Bei (m, k) -harten Garantien müssen mindestens m_{ξ^x} Fälle von k_{ξ^x} erfolgreich sein. Bei CSMA-Netzwerken kann hierfür keine deterministische Garantie gegeben werden [78, 79], weshalb wieder ein zulässige Toleranz ε_{ξ^x} definiert wird. Um die Wahrscheinlichkeit eines QoS-Fehlers zu bestimmen, nutzen HAMDAOUI & RAMANATHAN [79] eine Markov-Kette, die vergleichbar mit Abbildung 8.3 ist. Die Übergangswahrscheinlichkeiten bestimmen sie nur aus der Korrelation zweier aufeinander folgender Antwortzeiten. HAMDAOUI & RAMANATHAN [79] bezeichnen diese geringe Korrelation selbst als Schwachstelle ihres Ansatzes.

Deshalb wird ihr Ansatz in dieser Arbeit um ein *AR-Modell*² erweitert, um

2 AR: Autoregressiv

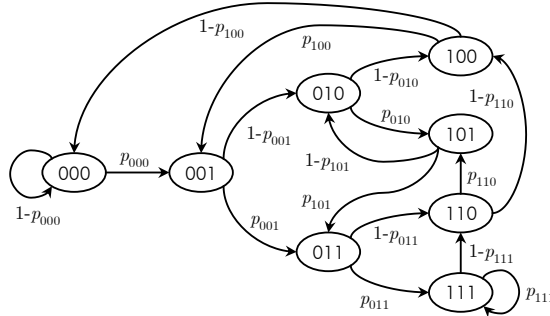


ABBILDUNG 8.3. Markov-Kette zur Berechnung der Wahrscheinlichkeit eines (2, 3)-QoS-Fehlers in Anlehnung an HAMDIAOUI & RAMANATHAN [79].

die in der Leistungsbewertung bestimmten Autokorrelationsfunktionen (Abbildung 8.2(b)) nutzen zu können. Hierbei wird angenommen, dass die Autokorrelation eines QoS-Charakteristikum $r_\xi(j)$ gleich der ihrer QoS-Defekte ist. Es werden k aufeinander folgende Realisierungen³ $M_\xi(1), \dots, M_\xi(k) = \{0, 1\}$ in ihrem Zustand des QoS-Defekts mit 1 als defekt betrachtet. War also bei $k = 3$ die vorletzte Nachricht defekt ($M_\xi(2) = 1$), so ist die neue Nachricht mit $r_\xi(2) \geq 0$ wieder defekt oder mit $r_\xi(2) < 0$ nicht. Zusätzlich muss der Einfluss der anderen Zustände betrachtet werden und die Wahrscheinlichkeit eines neuen QoS-Defekts

$$p_{\xi^x} = \begin{cases} \mathbb{P}[\xi \geq \xi^x], & \chi \in \{\cap, \sqcap\}; \\ \mathbb{P}[\xi \leq \xi^x], & \chi \in \{\cup, \odot\}. \end{cases} \quad (8.3)$$

Dann ist

$$p_{M_\xi(1), \dots, M_\xi(k)} = \frac{p_{\xi^x} + \sum_{j=1}^k r_\xi(j) \begin{cases} M_\xi(k-j+1), & r_\xi(j) \geq 0 \\ M_\xi(k-j+1) - 1, & r_\xi(j) < 0 \end{cases}}{\sum_{j=0}^k |r_\xi(j)|}. \quad (8.4)$$

Mit dieser Gleichung lassen sich die zustandsabhängigen Übergangswahrscheinlichkeiten der in Abbildung 8.3 beispielhaft dargestellten Markov-Kette bestimmen und daraus die Zustandswahrscheinlichkeiten $\pi_{M_\xi(1), \dots, M_\xi(k)}$. Sie dürfen die

3 Im Idealfall wird k durch (m, k) -parametriert. Besitzt die Autokorrelationsfunktion allerdings relevante Koeffizienten $r_\xi(i)$ für $i > k$ sollte k größer gewählt werden und in Gleichung (8.5) die fehlerhaften Untermengen (m, k) in den Kombinationen bestimmt werden. Zur einfacheren Darstellung wird der Idealfall angenommen.

definierte Toleranz ε_ξ nicht verletzen, so dass ein QoS-Fehler oder eine QoS-Warnung auftritt, sofern

$$\varepsilon_{\xi^x} \leq \sum \left\{ \pi_{M_\xi(1), \dots, M_\xi(k)} \left| \sum_{i=1}^k M_\xi(i) > k_{\xi^x} - m_{\xi^x} \right. \right\}. \quad (8.5)$$

8.2.5. QUALITÄTSMASS DES NETZWERKENTWURFS

Wie in Unterabschnitt 8.2.2 schon angesprochen wurde, ist die Qualität eines Netzwerkentwurfs schwierig zu quantifizieren. Nach Prämisse 16 gilt ein Netzwerkentwurf als frei von Entwurfsfehlern, wenn alle seine QoS-Anforderungen erfüllt sind. Auf dieser Basis wird ein Qualitätsmaß für den Netzwerkentwurf entwickelt, das sich aus der Qualität der QoS-Anforderungen ableitet.

Das Qualitätsmaß Q_{ξ^x} einer QoS-Anforderung drückt aus, wie gut sie erfüllt ist. Mit der Definition

$$Q_{\xi^x} = \begin{cases} -|\varepsilon_{\xi^x} - p_{\xi^x}|, & \chi = \odot; \\ \varepsilon_{\xi^x} - p_{\xi^x}, & \text{sonst} \end{cases} \quad (8.6)$$

ist es im Bereich $-1 \leq Q_{\xi^x} \leq 1$ negativ, wenn eine QoS-Anforderung nicht erfüllt wird und im anderen Fall positiv. Bei (m, k) -harten Garantien wird anstelle von p_{ξ^x} aus Gleichung (8.3) die Gleichung (8.5) verwendet.

Als Qualitätsmaß eines Netzwerkentwurfs wird zum einen das mittlere Qualitätsmaß aller QoS-Anforderungen Ξ im Netzwerk

$$\bar{Q} = \frac{1}{\|\Xi\|} \sum_{\xi^x \in \Xi} Q_{\xi^x} \quad (8.7)$$

verwendet. Hierbei gilt allerdings für die Addition der Qualitätsmaße von QoS-Anforderungen

$$Q_{\xi_1^x} + Q_{\xi_2^x} = \begin{cases} Q_{\xi_1^x}, & Q_{\xi_1^x} < 0 \wedge Q_{\xi_2^x} \geq 0; \\ Q_{\xi_2^x}, & Q_{\xi_1^x} \geq 0 \wedge Q_{\xi_2^x} < 0; \\ Q_{\xi_1^x} + Q_{\xi_2^x}, & \text{sonst,} \end{cases} \quad (8.8)$$

damit erfüllte QoS-Anforderungen keine gescheiterten auslöschen. Ist das Qualitätsmaß positiv, so ist das Netzwerk in allen seinen QoS-Anforderungen gut ausgelegt. Bestehen QoS-Fehler, QoS-Warnungen oder ist der Arbeitspunkt verletzt, so fällt das Qualitätsmaß ins Negative. Der Nachteil dieses Maßes liegt darin, dass durch Hinzufügen von erfüllten QoS-Anforderungen das Qualitätsmaß steigt. Wird stattdessen von allen Anforderungen das niedrigste Qualitätsmaß

$$Q_{\min} = \min \{Q_{\xi^x} \mid \xi^x \in \Xi\} \quad (8.9)$$

TABELLE XVI.

Qualität der QoS-Anforderungen im Beispiel ($\varepsilon_{\tau_e^X} = 1 - F_n(R_e \mid \mu = 0, \sigma = 1)$;
 $[\tau_e^X] = s$; $[p_{\tau_e^X}] = \%$; $[\varepsilon_{\tau_e^X}] = \%$; $[Q_{\tau_e^X}] = E^{-2}$; $\bar{Q} = -1.03$; $Q_{\min} = -8.00$).

e	Größe	R_e	$\varepsilon_{\tau_e^X}$	τ_e^\cap	$p_{\tau_e^\cap}$	$Q_{\tau_e^\cap}$	$G_{\tau_e^\cap}$	τ_e^\cap	$p_{\tau_e^\cap}$	$Q_{\tau_e^\cap}$	$G_{\tau_e^\cap}$
b1	Beleuchtungsstärke	1	15.9	0.5	12.7	3.20	4.03	-	-	-	-
b2	Beleuchtungsstärke	2	2.28	0.2	0.23	2.04	5.02	0.5	0.23	2.04	7.02
b3	Jalousiestand	2	2.28	0.5	14.0	-11.7	-5.12	-	-	-	-
b4	Lampendimmung	2	2.28	0.2	0.25	2.03	7.02	0.5	0.25	2.03	5.02
b5	Temperatur	1	15.9	1.0	13.5	2.32	4.02	-	-	-	-
b6	Ventilöffnung	1	15.9	1.0	12.4	3.44	4.03	-	-	-	-
b7	Anwesenheit	1	15.9	0.5	0.00	15.9	4.16	1.0	0.00	15.9	6.16
b8	Fensteröffnung	1	15.9	0.5	0.00	15.9	4.16	1.0	0.00	15.9	6.16
b9	Fensteröffnung	1	15.9	0.5	0.00	15.9	4.16	1.0	0.00	15.9	6.16
b10	Rauchmeldung	3	0.13	0.5	0.00	0.13	6.00	1.0	0.00	0.13	8.00
b11	Alarm	3	0.13	0.5	0.41	-0.27	-6.00	1.0	0.28	-0.15	-8.00

verwendet, so erhält man ein Qualitätsmaß, das unabhängig von der Anzahl der Anforderungen ist. Dies ist allerdings ein Vor- und Nachteil gleichermaßen, weshalb zur Bewertung der Qualität eines Netzwerks immer beide Qualitätsmaße herangezogen werden sollten.

Jetzt können die Ergebnisse in Tabelle XIV sinnvoll bewertet werden. Die Tabelle XVI definiert QoS-Anforderungen an die einzelnen Bindings im Netzwerk und bewertet die Qualität der einzelnen Verbindungen. Die Anforderungen an die Verzögerungszeit sind nicht besonders hoch und liegen bei der Konstantlichtregelung bei 200 ms und sonst bei 500 ms oder 1 s. Diese Anforderungen können von nahezu allen Bindings erfüllt werden. Nur die Bindings **b3**, **b11** scheitern und ihr Qualitätsmaß $Q_{\tau_e^X}$ ist negativ. Ein Blick in die Tabelle XIV zeigt, dass bei **b3** die mittlere Verzögerungszeit bei 35.01 ms liegt, also deutlich unter 500 ms. Ursache des Scheiterns bei **b3** ist die Fehlerwahrscheinlichkeit, die bei 14 % liegt und damit $\varepsilon_{\tau_e^X} = 2.28 \%$ (2σ -Grenze) deutlich übersteigt. Abbildung 8.2(e) zeigt, dass bei **b11** zusätzlich die Verzögerungszeit die untere Warngrenze bei 500 ms überschreitet, weshalb der Qualität niedriger ist als beim oberen Grenzwert. Diese Hintergründe kann ein erfahrener Anwender aus den Tabellen XIV und XIV ablesen. Ein weniger erfahrener Anwender benötigt hierbei Unterstützung, die in den Abschnitten 8.3 und 8.4 erläutert wird.

8.2.6. DATENMODELL DER QoS-ANFORDERUNGEN UND ANWENDUNG

Das Systemmodell aus Kapitel 5 muss noch um die QoS-Anforderungen vervollständigt werden. Das Datenmodell der QoS-Anforderungen besteht aus den Klassen `SoftQoSRequirement` und `MkQoSRequirement`, die sich von der ersten ableiten. Die Klassen definieren einzelne QoS-Anforderungen wie einen Grenzwert oder Arbeitspunkt und beinhalten die notwendigen Parameter, die in Aufzählung 5 aufgeführt werden. Es handelt sich um dynamische Datenobjekte (`ExtendedAttribute`), die den Klassen `Channel` und `Binding`, aber auch `DataPoint` und `DataPointProfile` zugewiesen werden können. Werden den beiden letzten Klassen `SoftQoSRequirements` oder `MkQoSRequirements` zugewiesen, so werden diese automatisch im Netzwerk verteilt. Dadurch können generische QoS-Anforderungen nach Anforderung 1 definiert werden, mit denen ein Anwender die QoS-Anforderungen nicht für das ganze Netzwerk händisch zuweisen muss, sondern mit denen er globale Standardanforderungen definieren kann, die dennoch spezifischen Prozessbezug besitzen.

AUFZÄHLUNG 5. Auszug aus den Attributen der Klassen der QoS-Anforderungen.

<code>SoftQoSRequirement</code>	<code>MkQoSRequirement</code>
<ul style="list-style-type: none"> • value $\mathbb{P}[\xi \geq \xi^x]$ • qosCharacteristic ξ (Error/Utilisation/Delay/Jitter) • qosRequirementType χ (Upper/Lower Warn/Fault;Target) • qosRisk (R_{ξ^x}) • epsilon (ϵ_{ξ^x}) 	<ul style="list-style-type: none"> • m_{ξ^x} • k_{ξ^x}

Der Algorithmus 3 beschreibt die automatische Verteilung. Grundlage sind die wiederverwendbaren, generischen Gerätemodelle aus der systemeigenen Datenbank. Hierin sind die `DataPointProfiles` direkt um QoS-Anforderungen erweiterbar⁴. Dadurch werden sie auf alle instanzierenden Datenpunkte und gebundene Bindings übertragen. Wird für das `DataPointProfile` des Heizungsventils eine Komfortanforderung von 60 Sekunden definiert, so erhalten automatisch `i10` und `i11` und das `Binding` `b6` die gleichen QoS-Anforderungen. Unter der Annahme, dass Prozesse ähnliche QoS-Anforderungen an die vorhergehenden und nachfolgenden Prozesse stellen, werden diese Anforderungen auf die Bindings `b5`, `b7`, `b9` erweitert, da sie über einen Verstärkungsfaktor $v_{i,o} > 0$ im Verkehrsmodell

⁴ Die systemeigene Datenbank ist aufgebaut wie das Datenmodell und kann genauso erweitert werden.

zu **b6** verbunden sind. Für das **DataPointProfile** von **i11** wurde eine Sicherheitsanforderung von 200 ms definiert. Sie verteilt sich in gleicher Weise auf **b7** bis **b10** und ergänzt in **b7** und **b9** die Komfortanforderungen von Binding **b6**.

ALGORITHMUS 3. Algorithmus zur Verteilung der QoS-Anforderungen.

¹ Sofern in Zeile 12 nicht *weiter* aufgerufen wird.

1. Identifiziere die Datenpunkte mit QoS-Anforderungen $\mathcal{E} \subseteq \mathcal{J} \cup \mathcal{D}$ (**DataPoint** oder **DataPointProfile** besitzt **SoftQoSRequirement**)
 2. Für jedes $\epsilon \in \mathcal{E}$
 3. Wenn $\epsilon \in \mathcal{D}$ (Ausgangsdatenpunkt)
 4. Für alle ausgehenden Bindings $b \in \mathfrak{B}_\epsilon$ führe Zeile 12 aus
 5. Für alle Zieldatenpunkte i führe Zeile 12 aus und¹ füge i ans Ende von \mathcal{E}
 6. Für alle intern verbundenen Eingangsdatenpunkte i führe Zeile 12 aus und¹ füge i ans Ende von \mathcal{E}
 7. Wenn $\epsilon \in \mathcal{J}$ (Eingangsdatenpunkt)
 8. Für alle eingehenden Bindings $b \in \mathfrak{B}_\epsilon$ führe Zeile 12 aus
 9. Für den Quelldatenpunkt o führe Zeile 12 aus und¹ füge o ans Ende von \mathcal{E}
 10. Für alle intern verbundenen Ausgangsdatenpunkt o führe Zeile 12 aus und¹ füge o ans Ende von \mathcal{E}
 11. Entferne ϵ aus \mathcal{E}

 12. Existiert eine QoS-Anforderung, die gleich ist, dann *weiter*, sonst wende aktuelle QoS-Anforderung an
-

8.3. ANALYSE DER FEHLERZUSAMMENHÄNGE

8.3.1. PRIORISIERUNG VON FEHLERN DURCH QoS-RISIKOKLASSEN UND QoS-GEWICHTE

Die Priorisierung von identifizierten QoS-Fehlern ist für eine gezielte Darstellung und Weiterverarbeitung wichtig. Hierfür wird in diesem Abschnitt eine Gewichtsfunktion für die QoS-Anforderungen und QoS-Fehler entwickelt.

In Unterabschnitt 8.2.2 wurde als Beispiel angebracht, dass die Betriebssicherheit und die resultierenden QoS-Anforderungen beim Rauchmelder **d12** wichtiger als beim Temperatursensor **d5** sind, da ein Ausfall des Temperatursensors kaum katastrophalen Folgen hat. Das *Risiko* [91] des Prozesses, als Kombination der Auftrittswahrscheinlichkeit und des Schweregrades eines Schadens, bestimmt also zu einem wichtigen Teil die Priorität der QoS-Anforderungen.

Da das Risiko eines Prozesses schwer zu quantifizieren ist, werden meist Risikoklassen verwendet. Die Prozesse in der Gebäudeautomation besitzen im All-

gemeinen nur ein geringes Risiko im Sinne der IEC 61511 [91], weshalb drei einfache Risikoklassen ausreichend sind.

- *Sicherheitsanforderungen* ($R_{\xi^x} = 3$) - Es besteht durch einen QoS-Fehler ein Risiko für Personen.
- *Funktionsanforderungen* ($R_{\xi^x} = 2$) - Es besteht durch einen QoS-Fehler ein Risiko für technische Geräte, aber nicht für Personen.
- *Komfortanforderungen* ($R_{\xi^x} = 1$) - Es besteht kein Risiko für Personen oder Geräte, sondern ein QoS-Fehler führt zu einem reduzierten Komfort.

Für einen Prozess, der noch nicht einmal Komfortanforderungen stellt, wird angenommen, dass keine QoS-Anforderungen definiert werden. Die Risikoklasse $R_{\xi^x} \in \{1, 2, 3\}$ wird mit den QoS-Anforderungen festgelegt.

Zur Priorisierung wird aus dieser Risikoklasse das *QoS-Gewicht* G_{ξ^x} bestimmt, welches darüber hinaus bewertet, wie gut eine QoS-Anforderung erfüllt ist. So ist ein QoS-Fehler, bei dem die obere und untere Grenze mit 10% der Dichte überschritten wird, stärker verletzt, als bei einer Anforderung mit 1% der gleichen Risikoklasse. Grundlage des QoS-Gewichts ist das Qualitätsmaß einer Anforderung Q_{ξ^x} aus Gleichung (8.6), welches zusätzlich mit der Risikoklasse und der Art der QoS-Anforderung bewertet wird, so dass ein QoS-Fehler immer höher gewichtet wird als eine QoS-Warnung. Deshalb sei das QoS-Gewicht

$$G_{\xi^x} = \operatorname{sgn}(Q_{\xi^x}) \begin{cases} 5 + R_{\xi^x} + |Q_{\xi^x}|, & \chi \in \{\mathbb{m}, \mathbb{u}\}; \\ 3 + R_{\xi^x} + |Q_{\xi^x}|, & \chi \in \{\cap, \cup\}; \\ R_{\xi^x} + |Q_{\xi^x}|, & \chi = \odot; \end{cases} \quad (8.10)$$

mit der Signum-Funktion $\operatorname{sgn}(x)$. Genauso wie das Qualitätsmaß ist das QoS-Gewicht negativ, wenn die QoS-Anforderung nicht erfüllt wird. Im späteren Verlauf der Arbeit werden Elemente wie Bindings oder Kanäle mit Hilfe des QoS-Gewichts sortiert. Damit zwei gescheiterte QoS-Warnungen nicht einen QoS-Fehler überdecken, werden immer die kleinsten Gewichte der Elemente verglichen. Beispielsweise gilt bei zwei Bindings \mathbf{b}_1 und \mathbf{b}_2 mit den QoS-Anforderungen $\Xi_{\mathbf{b}_1}$ und $\Xi_{\mathbf{b}_2}$

$$G_{\mathbf{b}_1} < G_{\mathbf{b}_2} \quad \text{wenn} \quad \min(G_{\xi^x} \mid \xi^x \in \Xi_{\mathbf{b}_1}) < \min(G_{\xi^x} \mid \xi^x \in \Xi_{\mathbf{b}_2}). \quad (8.11)$$

An anderen Stellen werden QoS-Gewichte addiert. Damit erfüllte Anforderungen keine gescheiterten auslöschen, gilt für die Addition von QoS-Gewichten die Vorschrift aus Gleichung (8.8). Die QoS-Gewichte für das Beispiel werden mit in Tabelle XVI aufgeführt.

8.3.2. FEHLERBÄUME ZUR DIAGNOSE

Die Zusammenhänge von QoS-Fehlern werden mittels Fehlerbäume analysiert und dargestellt. *Fehlerbäume* sind eine graphische Methode aus der Zuverlässig-

keitsanalyse, um komplexe Ereignisse auf atomare Ereignisse herunterzubrechen [42]. Hierzu wird das komplexe Ereignis aufgetragen und dann werden alle Ereignisse, die dazu führen, logisch aufgeschlüsselt. Auf diese Weise wird das System so lange dekomponiert, bis an den Blättern nur noch atomare Ereignisse vorliegen. So zum Beispiel: Autounfall \Rightarrow Bremssystem \Rightarrow Bremsschlauch \vee Bremspedal. In einem möglichen zweiten Schritt werden die atomaren Ereignisse in ihrer Auftretswahrscheinlichkeit bewertet. Jetzt rechnet man den Fehlerbaum wieder über die logischen Verknüpfungen hoch, um die Wahrscheinlichkeit des Wurzelereignisses zu erhalten.

Fehlerbäume finden auch in der Diagnose zur Strukturierung des Diagnosewissens Verwendung [3, 94, 183, 216]. In dieser Arbeit erfüllen sie drei Zwecke: Sie dienen der Darstellung der Zusammenhänge der QoS-Fehler für den Nutzer, so dass er gemäß Anforderung 2 die verschiedenen Ursachen nachvollziehen kann. Zweitens sind die relevanten Ursachen für den Anwender im Sinne von Anforderung 1 schnell zu identifizieren, wenn der Fehlerbaum mit Wahrscheinlichkeiten belegt wird. Drittens kann das Tool in gleicher Weise die relevanten Ursachen identifizieren und in einem weiteren Schritt den Nutzer gezielt beraten (Anforderung 5).

Für die Diagnose geschieht die Gewichtung des Fehlerbaums in umgekehrter Richtung von der Wurzel zu den Blättern, den atomaren Ursachen, wobei die gleichen Konsistenzbedingungen gelten. Setzt sich das komplexe Ereignis C aus den Ereignissen A und B zusammen, dann gilt bei einem Oder-Glied

$$\mathbb{P}[C] = \mathbb{P}[A \vee B] = \mathbb{P}[A] + \mathbb{P}[B] - \mathbb{P}[A \wedge B] \quad (8.12)$$

und für ein Und-Glied

$$\mathbb{P}[C] = \mathbb{P}[A \wedge B] = \mathbb{P}[A] \cdot \mathbb{P}[B|A] = \mathbb{P}[B] \cdot \mathbb{P}[A|B]. \quad (8.13)$$

Sind die Ereignisse A und B unabhängig voneinander, ist $\mathbb{P}[A \wedge B] = \mathbb{P}[A] \cdot \mathbb{P}[B]$ und sind A und B disjunkt, so ist $\mathbb{P}[A \wedge B] = \emptyset$.

Das Wurzelereignis ist immer der identifizierte QoS-Fehler oder die QoS-Warnung mit der Wahrscheinlichkeit $p_{\xi^x} = \mathbb{P}[\xi \geq \xi^x]$ mit $\chi \in \{\cap, \cup\}$. Die aufgetragenen Teilereignisse erfüllten nicht von Natur aus die Konsistenzbedingungen. So bildet sich der QoS-Fehler Auslastung in Abbildung 8.4 aus den unscharfen Ereignissen hoher Datendurchsatz und geringe Bitrate entsprechend Gleichung (7.3). Beides sind weder Wahrscheinlichkeiten noch unabhängig quantifizierbare Ereignisse, so dass $\mathbb{P}[A \wedge B]$ nicht zu bestimmen ist. Deshalb wird bei Und-Gliedern häufig ein Ereignis als Sechseck dargestellt. Ein so dargestelltes Ereignis ist nicht quantifizierbar und wird nicht gewichtet. Dadurch wird der hohe Datendurchsatz von der geringen Bitrate entkoppelt und kann bewertet werden.

Gründe für einen hohen Datendurchsatz können unter anderem falsche Routingtabellen oder zu wenige Router sein. Letzteres ist wieder als Route dargestellt

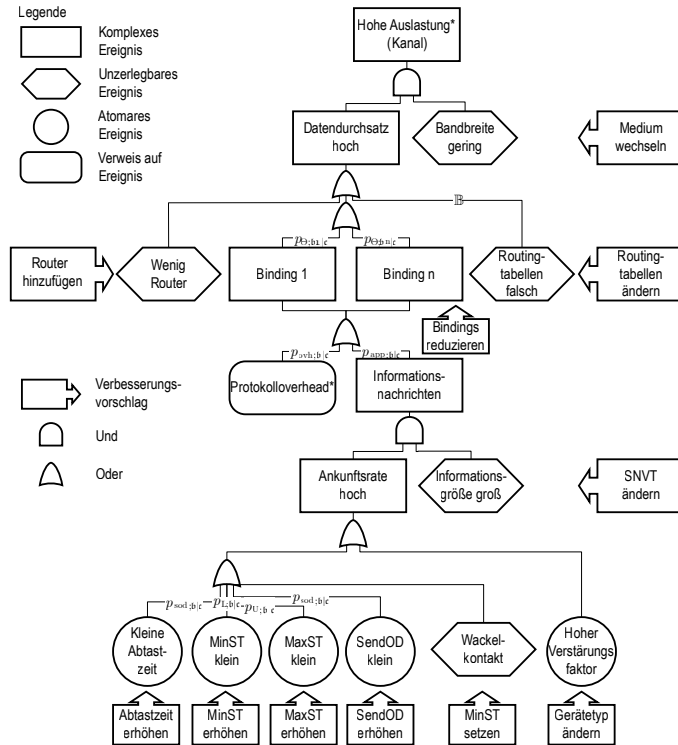


ABBILDUNG 8.4. Fehlerbaum der Kanalauslastung (Ergänzt in Abb. 8.5).

und nicht quantifizierbar. Auch falsche Routingtabellen entfallen bei der Gewichtung. Die Verbindung ist mit \mathbb{B} gekennzeichnet, was bedeutet, dass es sich um einen Booleschen Wert (Wahr/Falsch) handelt. Sie werden bei der Diagnose separat getestet, in diesem Fall durch Validation der Routingtabellen gegen das Systemmodell. Wird der Test bestanden, entfallen sie bei der Gewichtung, ansonsten müssen sie zuvor korrigiert werden.

Der QoS-Fehler Auslastung p_{ρ_x} reduziert sich auf die Ursache eines hohen Datendurchsatzes der passierenden Bindings. Eine jedes Binding b beteiligt sich im Mittel mit dem Datendurchsatz

$$\bar{\Theta}_{b|c} = \sum_{m \in \mathcal{M}_b \mid c \in \mathcal{W}_m} \bar{\Theta}_{m|c} \quad (8.14)$$

seiner passierenden Nachrichtenklassen nach Gleichung (7.9) am Datendurchsatz $\bar{\Theta}_c$ des Kanals. Die Datendurchsätze sind disjunkt, so dass aus der Konsistenz-

bedingung des Oder-Gliedes $\mathbb{P}[C] = \mathbb{P}[A \vee B] = \mathbb{P}[A] + \mathbb{P}[B]$ folgt

$$p_{\rho_c^x} = \sum_{b \in \mathfrak{B}} p_{\bar{\Theta}; b|c} \quad (8.15)$$

und

$$p_{\bar{\Theta}; b|c} = \frac{\bar{\Theta}_{b|c}}{\bar{\Theta}_c} p_{\rho_c^x}. \quad (8.16)$$

Der Datendurchsatz $\bar{\Theta}_{b|c}$ eines Bindings kann weiter in den zur Informationsübertragung notwendigen Datendurchsatz $\bar{\Theta}_{\text{app}; b|c}$ und den vermeidbaren⁵ Protokolloverhead $\bar{\Theta}_{\text{ovh}; b|c}$ zerlegt werden. Diese Datendurchsätze werden aus den Ergebnissen der Leistungsbewertung bestimmt. Der Datendurchsatz $\bar{\Theta}_{\text{app}; b|c}$ ist die Last aller passierenden, informationsübertragenden Nachrichtenklassen ohne Wiederholungen (AckD, Res, Reply) und der Protokolloverhead der Datendurchsatz der restlichen Nachrichtenklassen. Um die Gewichtung zu bestimmen, wird wieder die Konsistenzregel angewendet. Da die Datendurchsätze disjunkt sind, gilt

$$p_{\bar{\Theta}; b|c} = p_{\text{ovh}; b|c} + p_{\text{app}; b|c} \quad (8.17)$$

und

$$p_{\text{ovh}; c} = \frac{\bar{\Theta}_{\text{ovh}; b|c}}{\bar{\Theta}_{\text{ovh}; b|c} + \bar{\Theta}_{\text{app}; b|c}} p_{\rho; b|c}. \quad (8.18)$$

Auf diese Weise kann ein Großteil der Fehlerbäume gewichtet werden, obwohl unscharfe Ereignisse benannt sind. Dies wird in den dargestellten Fehlerbäumen immer durch eine Wahrscheinlichkeit auf den Verbindungen gekennzeichnet. Dadurch, dass die Gewichte immer die Konsistenzregeln erfüllen, ist die Vergleichbarkeit gewahrt und sie verhalten sich intuitiv. So sind alle Gewichte reine Wahrscheinlichkeiten und Teilereignisse haben meist kleinere Wahrscheinlichkeiten⁶. Die vollständigen Gleichungen der Gewichte werden aufgrund ihres Umfangs nicht aufgeführt, sondern nur die Fehlerbäume in ihren Zusammenhängen erklärt. Dabei beziehen sich die Fehlerbäume immer auf eine Verletzung der oberen Grenzen der QoS-Charakteristika. Die Fehlerbäume der unteren Grenzen wurden nicht aufgestellt, da sie eine geringere praktische Bedeutung haben. Sie führen sich auf ähnliche Einflussfaktoren zurück.

5 Im Sinne des Datendurchsatzes vermeidbar, dass bestätigte Übertragungsdienste zur Gewährleistung der Integrität wichtig sind, spielt bei für die Kapazität keine Rolle.

6 Eine Ausnahme sind Und-Glieder, die allerdings meist vereinfacht werden (Sechseck).

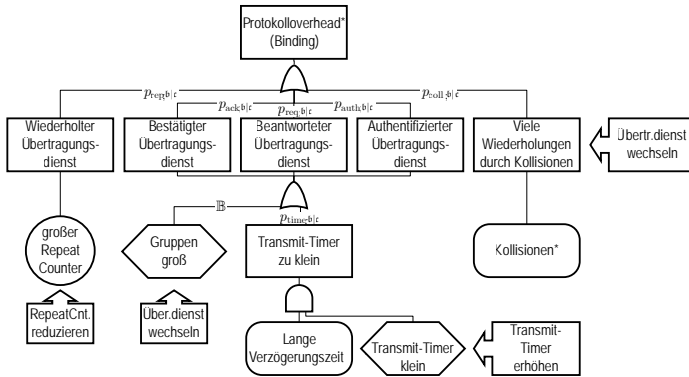


ABBILDUNG 8.5. Fehlerbaum des Protokolloverheads ergänzend zu Abb. 8.4.

8.3.3. FEHLERBAUM DER KANALAUSLASTUNG

In Unterabschnitt 8.2.3 wurde die Kanalauslastung ρ_c als QoS-Charakteristikum definiert. Der entsprechende Fehlerbaum ist in Abbildung 8.4 und 8.5 dargestellt. Die ersten Ereignisse wurden im letzten Abschnitt erklärt, so dass nur noch auf die unteren eingegangen wird.

Ursache für einen hohen Datendurchsatz an Informationsnachrichten ist eine große Informationsgröße $S_{App;b}$ und eine hohe Ankunftsrate. Die Informationsgröße ist selten veränderlich, weshalb sie als Sechseck ausgeschlossen wird. Die Ankunftsrate ist auf den Quelldatenpunkt des Bindings zurückzuführen, wo die in Kapitel 6 diskutierten Ankunftsprozesse und ihre Parameter Einfluss haben. Hierbei kann die Min-, Max-Send-Time und der Abtastprozess mit w_L , w_U und p_δ zueinander gewichtet werden.

Der Fehlerbaum des Protokolloverheads wird separat in Abbildung 8.5 dargestellt. Ein solcher Verweis auf andere Fehlerbäume wird durch ein Ereignis mit runden Ecken und durch ein Sternchen am Namen des Ziels gekennzeichnet. Protokolloverhead entsteht bei allen Übertragungsdiensten, die nicht unbestätigt sind. Der Datendurchsatz, den sie erzeugen, kann durch eine falsche Wahl oder Parametrierung des Übertragungsdienstes übermäßig groß sein. Dies steht immer im Zusammenhang zu den real auftretenden Verzögerungszeiten, Übertragungsfehlern und Kollisionen, deren Fehlerbäume in den nächsten Abschnitten behandelt werden.

8.3.4. FEHLERBAUM DER VERZÖGERUNGSZEIT

Der in Abbildung 8.6 dargestellte Fehlerbaum bezieht sich speziell auf die Verzögerungszeit eines Bindings. Der Jitter kann ähnlich behandelt werden.

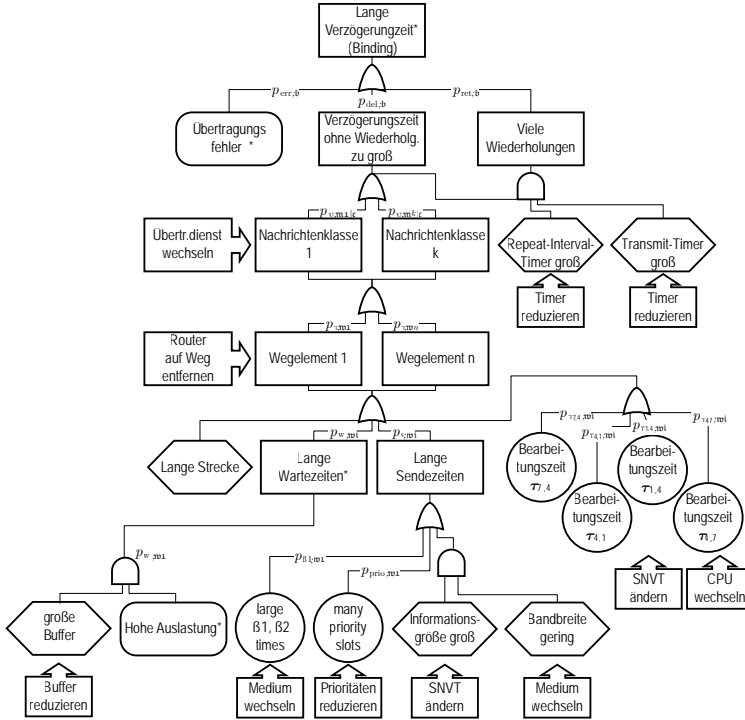


ABBILDUNG 8.6. Fehlerbaum für die Verzögerungszeit.

Der erste Schritt ist, den Fehlerbaum in seine einzelnen Elemente zu zerlegen. Bei einem Binding ist die Verzögerungszeit immer nur die Zeit bis zur erfolgreichen Informationsübertragung, unabhängig davon ob und wann die Transaktion erfolgreich abgeschlossen wird. Bei einem unbestätigten Übertragungsdienst ist die Verzögerungszeit des Bindings identisch mit der einzigen Nachrichtenklasse. Bei einem wiederholten Übertragungsdienst werden die Nachrichten im Abstand von `binding.repeatIntervalTimer` $T_{RT;b}$ wiederholt, wobei anzunehmen ist, dass die Verteilung Gleichung (6.15) folgt. Die einzelnen Nachrichten erreichen ihr Ziel mit einer Erfolgswahrscheinlichkeit $p_{rpt;o,i}^+$. Da die Verzögerungszeit den Fall p_b^- mit einer unendlichen Verzögerung enthält, ist direkt

$$f_{\tau_{b;o,i}}(t) = \sum_{i=0}^{r_b^+ - 1} p_{rpt;o,i}^+ (p_{rpt;o,i}^-)^i \left[f_{\tau_{rpt;o,i}}(t) \oplus f_{tmr}(t \mid i \cdot T_{RT;b}) \right]. \quad (8.19)$$

Bei einem bestätigten Übertragungsdienst trifft die Information mit $p_{AckD;o,i}^+$

als erste Nachricht ein, sonst bleibt die Bestätigung aus und eine Erinnerung wird generiert. Sie wird erst gesendet, nachdem der `binding.transmission-Timer` $T_{\text{TT};\mathfrak{b}}$ abgelaufen ist, welcher zu der Verzögerungszeit der Erinnerung hinzukommt und nach Gleichung (6.16) verteilt ist.

$$\mathbf{f}_{\tau_{\mathfrak{b};\mathfrak{o},\mathfrak{i}}}(t) = \begin{cases} p_{\text{AckD};\mathfrak{o},\mathfrak{i}}^+ \cdot \mathbf{f}_{\tau_{\text{AckD};\mathfrak{o},\mathfrak{i}}}(t) + p_{\text{AckD};\mathfrak{o},\mathfrak{i}}^- \cdot \\ \sum_{i=1}^{\tau_{\mathfrak{b}}^-} p_{\text{Rem};\mathfrak{o},\mathfrak{i}}^+ (p_{\text{Rem};\mathfrak{o},\mathfrak{i}}^-)^{i-1} \mathbf{f}_{\tau_{\oplus\mathfrak{t};\mathfrak{o},\mathfrak{i}}}(t), & \|\mathfrak{G}\| \leq 16; \\ \sum_{i=0}^{\tau_{\mathfrak{b}}^-} p_{\text{AckD};\mathfrak{o},\mathfrak{i}}^+ (p_{\text{AckD};\mathfrak{o},\mathfrak{i}}^-)^i \mathbf{f}_{\mathfrak{a}\oplus\mathfrak{t};\mathfrak{o},\mathfrak{i}}(t), & \|\mathfrak{G}\| > 16; \end{cases} \quad (8.20)$$

mit

$$\begin{aligned} \mathbf{f}_{\tau_{\oplus\mathfrak{t};\mathfrak{o},\mathfrak{i}}}(t) &= \mathbf{f}_{\tau_{\text{Rem};\mathfrak{o},\mathfrak{i}}}(t) \oplus \mathbf{f}_{\text{tmr}2}(t \mid i \cdot T_{\text{TT};\mathfrak{b}}) \quad \text{und} \\ \mathbf{f}_{\mathfrak{a}\oplus\mathfrak{t};\mathfrak{o},\mathfrak{i}}(t) &= \mathbf{f}_{\tau_{\text{AckD};\mathfrak{o},\mathfrak{i}}}(t) \oplus \mathbf{f}_{\text{tmr}2}(t \mid i \cdot T_{\text{TT};\mathfrak{b}}). \end{aligned}$$

In ähnlicher Weise bildet sich die Verteilung der Verzögerungszeit der authentifizierten Varianten und des antwortenden Übertragungsdienstes. Beim letzteren ist die Übertragung mit $(1 - p_{\text{Req};\mathfrak{o},\mathfrak{i}}^+ p_{\text{Res};\mathfrak{o},\mathfrak{i}}^+)$ erfolgreich und besitzt eine Verzögerungszeit $\mathbf{f}_{\tau_{\text{Req};\mathfrak{o},\mathfrak{i}}}(t) \oplus \mathbf{f}_{\tau_{\text{Res};\mathfrak{i}}}(t)$ (vgl. Gl. (7.16)).

Im Fehlerbaum wird unterschieden, ob der Anteil der Verteilung, der die Grenze überschreitet, durch Wiederholungen entstanden ist oder die Verzögerungszeit schlechthin zu groß ist. Im ersten Fall haben Fehlerraten und die Wiederholungstimer zusätzlichen Einfluss, im anderen muss die Verzögerungszeit der Nachrichtenklassen weiter zerlegt werden. Sie setzt sich aus den Verzögerungszeiten der einzelnen Wegelemente zusammen, so dass analog zu Gleichung (7.7) die Verzögerungszeit bis zum Verlassen des Wegelementes \mathfrak{w} gleich ist mit

$$\mathbf{f}_{\tau_{\mathfrak{w}}}(t) = \bigoplus_{\mathfrak{w}^* \in \mathfrak{W}_{\mathfrak{p}_\mathfrak{o}, \mathfrak{w}}} \mathbf{f}_{\tau_{\mathfrak{w}^*}}(t). \quad (8.21)$$

Damit weist die Nachrichtenklasse \mathfrak{m} für den Zielport $\mathfrak{p}_\mathfrak{i}$ eine Verzögerungszeit $\mathbf{f}_{\tau_{\mathfrak{m} \mid \mathfrak{i}}}(t) = \mathbf{f}_{\tau_{\mathfrak{p}_\mathfrak{i}}}(t)$ auf. Hierbei wird angenommen, dass die Verzögerungszeiten der einzelnen Wegelemente und Nachrichtenklassen statistisch unabhängig sind. Unabhängig von dieser Annahme addiert sich der Erwartungswert der Verzögerungszeit der Nachrichtenklasse aus den Erwartungswerten der Wegelemente⁷, so dass sich die Gewichte bestimmen zu

$$\bar{\tau}_{\mathfrak{m} \mid \mathfrak{i}}(t) = \sum_{\mathfrak{w}^* \in \mathfrak{W}_{\mathfrak{p}_\mathfrak{o}, \mathfrak{p}_\mathfrak{i}}} \bar{\tau}_{\mathfrak{w}^*}(t). \quad (8.22)$$

⁷ Bei der Berechnung des Erwartungswerts, werden die unendlichen Verzögerungszeiten nicht mitgerechnet.

Die Verzögerungszeiten der einzelnen Elemente können weiter zerlegt werden, wie Abbildung 6.3 demonstriert. An einem ausgehenden Port besteht die Zeit aus der Bearbeitungszeit $\tau_{4,1}$ und der Buszugriffszeit τ_{MAC} , die die Wartezeit τ_{W} und die Sendezeit $\tau_{\text{S}} = S_{\text{phy};\text{m}}/\Theta_{\text{max};c}$ beinhaltet. Die Wartezeit hängt stark mit der Ankunftsrate am Gerät und der Netzwerkauslastung zusammen, während die anderen Zeiten von der Nachrichtengröße abhängen. Die Verzögerungszeit eines eingehenden Ports ist die Bearbeitungszeit $\tau_{1,4}$, die eines Kanals die vernachlässigbare Laufzeit der Strecke $l_{\mathbf{w}_{i-1},\mathbf{w}_{i+1}}$ zwischen den kommunizierenden Ports $t_c = 3l_{\mathbf{w}_{i-1},\mathbf{w}_{i+1}}/2c \approx 0 \text{ s}$. Für einen Router werden nur die Verzögerungszeiten der Ports $\tau_{1,4}$ und $\tau_{4,1}$ veranschlagt. Am ersten und letzten Port kommen noch $\tau_{7,4}$ und $\tau_{4,7}$ hinzu.

Die Verteilung der Verzögerungszeit beinhaltet bereits den Fehlerfall der Übertragung mit $p_{\mathbf{b}}^-$. Auf diese Weise wird bei der Überprüfung der oberen Grenzen immer bedacht, dass die Nachricht gar nicht ankommt. Dieser Verlust ist im Fehlerbaum gleich am Anfang integriert und wird im nächsten Abschnitt detailliert untersucht.

8.3.5. FEHLERBAUM DER FEHLERWAHRSCHEINLICHKEIT

Das Vorgehen bei der Fehlerwahrscheinlichkeit \mathbf{b} eines Bindings ähnelt dem Vorgehen bei der Verzögerungszeit. Die Fehlerwahrscheinlichkeiten werden nach den unterschiedlichen Übertragungsdiensten bis auf die Nachrichtenklassen zurückgeführt und von dort weiter auf die einzelnen Wegelemente. Die formalen Zusammenhänge wurden in Abschnitt 7.3 dargestellt und bilden den Hintergrund von Abbildung 8.7.

Bei eingehenden Ports und Routern kann die Fehlerwahrscheinlichkeit nicht quantifiziert werden, da in der Leistungsbewertung hierfür keine Bearbeiter modelliert wurden. Prinzipiell gibt es bei Eingangsports bereits Warteschlangen und bei Routern eine Kapazitätsbegrenzung, wenn Verkehr von beiden Seiten ankommt, die wiederum Auswirkung auf diese Warteschlangen hat. Das nicht quantifizierbare Ereignis ist bei Bedarf im Fehlerbaum zu erweitern.

Bei ausgehenden Ports können aufgrund langer Wartezeiten gemäß Abbildung 8.6 die Warteschlangen überlaufen oder beim Buszugriff Kollisionen auftreten, wenn mehrere Ports die gleichen Arbitrierungsfenster wählen (Abschnitt 4.2). Im üblichen Fall werden die beteiligten Telegramme zerstört. Einige Transceiver erkennen Kollisionen und starten einen neuen Übertragungsversuch mit einer neuen Arbitrierung. Dadurch wird die Fehlerwahrscheinlichkeit zu Lasten einer höheren Verzögerungszeit reduziert. Ursache für Kollisionen sind eine hohe Netzwerkauslastung [16, 127] oder mehrfach vergebene Prioritäten.

Bei der Übertragung auf dem Kanal kann die Nachricht verändert werden. Fehlen Endwiderstände, so treten interferierende Reflektionen auf [82] oder wird die Leitungslängen ohne Repeater überschritten, fällt die Signalstärke zu stark ab.

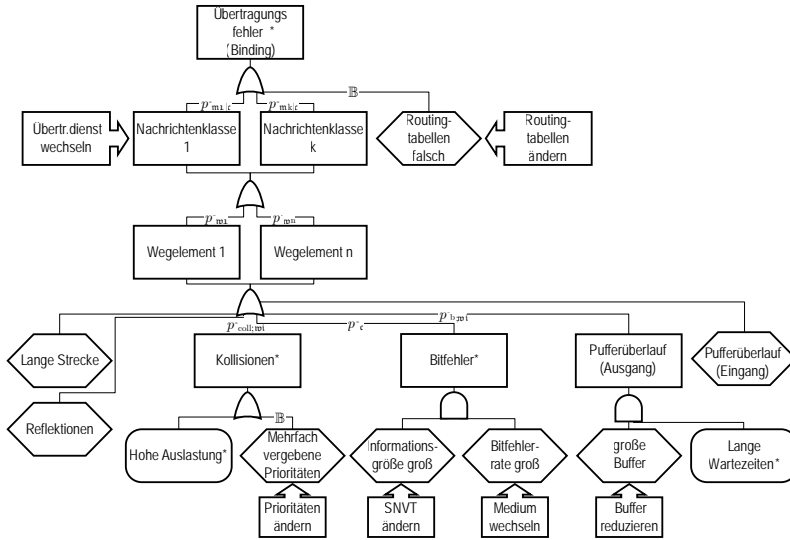


ABBILDUNG 8.7. Fehlerbaum für die Übertragungsfehler.

Beides lässt sich im Netzwerkentwurf nicht diagnostizieren, sondern nur *Bitfehler* können abgeschätzt werden. Ihr Auftreten hängt stark von den Eigenschaften des Mediums und den elektromagnetischen Interferenzen der Umgebung ab. Bei kabelgebundenen Netzwerken veranschlagt man eine Fehlerwahrscheinlichkeit von $p_{\text{bit};c}^- \approx 1 \text{ E}^{-6}, \dots, 1 \text{ E}^{-5}$ pro Bit [127], bei Lichtwellenleitern ungefähr 3 E^{-7} [26] und bei kabellosen Übertragungen dienen $1 \text{ E}^{-5}, \dots, \text{ E}^{-3}$ als Orientierung [238, 239]. Daraus kann die Fehlerwahrscheinlichkeit abgeschätzt werden zu

$$p_c^- = \left(1 - p_{\text{bit};c}^-\right)^{S_{\text{phy};m}}. \quad (8.23)$$

In der Praxis treten Bitfehler selten vereinzelt auf, sondern sind stark korreliert und burstartig. Dadurch sinkt insbesondere bei großen Paketen die mittlere Paketfehlerrate, da sich Bitfehler innerhalb weniger Pakete häufen [26]. Die Gleichung (8.23) beschreibt dementsprechend ein Worst-Case-Szenario mit höherer Paketfehlerrate. Sie wird nur bei der analytischen Leistungsbewertung benutzt. In der Simulation werden die korrelierten Bitfehler mittels eines *Gilbert-Elliott-Modells* erzeugt, das vergleichbar mit einem On/Off-Modell für die Fehlerrate ist. Es hat sich in verschiedenen Szenarios bewährt [52, 72, 237].

8.4. NUTZERBERATUNG

8.4.1. EINFÜHRUNG

Mit Hilfe der im letzten Abschnitt diskutierten Fehlerbäume können die Hintergründe identifizierter QoS-Fehler einfach aufgedeckt werden. Die Gewichtung der QoS-Fehler und Ursachen unterstützt den Anwender in einer gezielten Analyse. Letztendlich implizieren die atomaren Ursachen immer Lösungsmöglichkeiten, die als Schlagwort im Fehlerbaum grün angegeben sind. Diese Lösungsmöglichkeiten sind auch dem Tool in Form von Textbausteinen bekannt. Während der Anwender durch den Fehlerbaum wandert, werden daraus aktuell passende Lösungsvorschläge erzeugt und dem Anwender angezeigt. Die Abbildung 8.8 zeigt einen Prototyp der Diagnoseansicht des NetPlan-Tools von KOLMAN [124]. Auf der linken Seite sind die identifizierten Fehler aufgelistet, auf der rechten Seite die Gewichte der Ursachen und darunter sind die Erklärungen und Lösungsvorschläge zu den Fehlern. Dadurch erhält der Anwender umfangreiche Informationen bei der Analyse der Fehlerzusammenhänge und muss sie nicht von alleine verstehen.

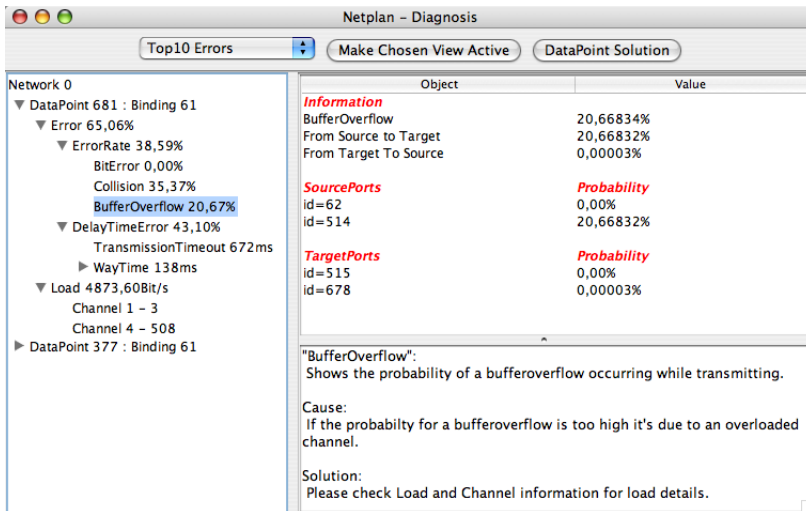


ABBILDUNG 8.8. Prototyp der Diagnoseansicht des NetPlan-Tools aus [124].

Zusätzlich kann der Anwender das Tool gezielt nach Ursachen für den ausgewählten QoS-Fehler suchen lassen, wobei der Fehlerbaum anhand seiner Gewichte mit dem Algorithmus 4 analysiert wird. Dadurch werden die effektivsten Lösungsmöglichkeiten für den Anwender automatisch bestimmt und ihm vorgeschlagen. Er kann auf diese Weise den Netzwerkentwurf gezielt verändern.

ALGORITHMUS 4. Algorithmus zur gezielten Ursachenanalyse im Fehlerbaum.

1. Anwender wählt einen QoS-Fehler aus
 2. Gewichte den Fehlerbaum mit den Leistungskenngrößen
 3. Wähle alle atomaren, quantifizierten Blätter des Fehlerbaums:
 4. Sortiere die Blätter absteigend nach ihrem Gewicht
 5. Wähle die Top- n Blätter mit dem größten Gewicht aus und für jedes Blatt:
 6. Bestimme den Lösungsvorschlag des Blattes (Hauptvorschlag)
 7. Sammle alle Lösungsvorschläge von der Wurzel bis zum Blatt inklusive nicht-quantifizierter Ursachen in Und-Verknüpfungen (Nebenvorschläge)
 8. Kombiniere die Lösungsvorschläge ähnlicher Ursachen
 9. Präsentiere Vorschläge dem Anwender
-

Bei üblichen Lösungsstrategien kann sich der Anwender darüber hinaus in der Parametrierung des Netzwerks beraten lassen. So ist es sinnvoll, bei einzelnen, isolierten QoS-Fehlern von Bindings die Übertragungsdienste und Prioritäten anzupassen. Ist ein ganzer Kanal überlastet, kann ein Vorschlag zur Lastreduzierung berechnet werden. Diesem Thema widmen sich die Abschnitte 8.4.4 und 8.4.5 im Detail.

Bei der Bestimmung dieser Parametrierungsvorschläge entsteht oft ein *Pareto-Optimierungsproblem* [50, 172, 247, 248], da die Verbesserung der Qualität der einen QoS-Anforderung in der Verschlechterung der Qualität einer anderen resultiert, ohne dass eine eindeutig beste Lösung definiert werden kann. Prinzipiell stellen das Qualitätsmaß eines Netzwerkentwurfes oder sein QoS-Gewicht, das zusätzlich die QoS-Risikoklassen einbezieht, jeweils eine monokriterielle Zielfunktion bereit, dessen Maximum eine quantitativ beste Lösung darstellt. Diese Zielfunktion ist allerdings trügerisch, da die Auswirkungen auf die Qualität der Prozesse nicht betrachtet werden. Eine ideale Zielfunktion müsste die unterschiedlichen Prozesse diesbezüglich stärker differenzieren, was praktisch weder effizient noch trivial ist und somit nach den Anforderungen 1 und 2 ausscheidet. Deshalb existiert, anstatt von einer eindeutig besten Lösung, eine nicht vollständig bestimmbare Menge praktikabler Kompromisse, die als *Pareto-Menge* oder *Pareto-Optimum* bezeichnet wird. Um diese Lösungsmenge zu bestimmen, werden zwei Strategien verfolgt. Zum einen wird zuerst versucht, nur die Netzwerkparameter anzupassen. Dadurch können Anforderungen erfüllt werden, ohne die Prozesse selbst zu ändern, so dass die vom Anwender vorgesehene Funktion ermöglicht wird. Erst wenn diese Strategie nicht erfolgreich ist, wird durch die Abtastparameter direkter Einfluss auf die Prozesse genommen.

Als nächste Strategie werden die QoS-Risikoklassen genutzt, um die Eingriffe zu priorisieren. Ziel ist es, dass die Anforderungen von Prozessen mit einer hohen Risikoklasse stärker durchgesetzt werden und dass sie vor dem Eingriff im Interesse anderer Anforderungen geschützt sind. Dementsprechend gilt

- Sicherheitsanforderungen - Die Anforderungen müssen erfüllt werden. Eingriffe sind nur zulässig, um die eigenen Anforderungen durchzusetzen.
- Funktionsanforderungen - Die Anforderungen sollten dringend erfüllt werden. Eingriffe, die nicht die eigenen Anforderungen durchsetzen, sollten vermieden werden.
- Komfortanforderungen - Die Anforderungen sollten erfüllt werden. Eingriffe sind auch im Interesse fremder Anforderungen möglich.

Letztendlich kann das Tool zwar die Auswirkungen der Parameter auf die Netzwerkleistung abschätzen, aber nicht auf die Prozesse. Deshalb ist die letzte entscheidende und ausführende Instanz immer der Systemintegrator. Er muss alle Änderungen am Netzwerkentwurf selbst vornehmen, so dass die Beratung keine automatische Parametrierung umfasst.

8.4.2. VERBESSERUNG EINZELNER DIENSTQUALITÄTEN DURCH ÜBERTRAGUNGSDIENSTE

Einzelne QoS-Fehler können oft durch eine passende Auswahl und Parametrierung des Übertragungsdienstes behoben werden. Die Übertragungsdienste eignen sich unterschiedlich für QoS-Anforderungen. Die Tabelle XVII gibt eine Übersicht über die empfohlenen Übertragungsdienste nach den geforderten QoS-Charakteristika. Dabei ist der unbestätigte Übertragungsdienst die beste Wahl für eine schnelle und lastarme Übertragung, bei der die Integrität keine große Rolle spielt. Bei hohen Integritätsanforderungen empfiehlt sich der bestätigte oder wiederholte Übertragungsdienst, wobei der erste nur bei Gruppennachrichten unter vier Empfängern weniger Last erzeugt. Auch unter Zeitaspekten ist der bestätigte Übertragungsdienst nicht die beste Wahl, da sein Timer üblicherweise größer ist, so dass Wiederholungen später eingeleitet werden und folglich später ankommen. Der beantwortete und authentifizierte Übertragungsdienst schneidet in allen Punkten noch schlechter ab. Er realisiert allerdings spezielle Funktionen, die zum Teil benötigt werden.

TABELLE XVII.

Empfohlene Übertragungsdienste nach den geforderten QoS-Charakteristika
 (¹ Bei einer Gruppengröße unter vier, sonst: Wiederholt).

	Zeit	Kapazität	Integrität	Kapazität & Integrität
Zeit	Unbestätigt	Unbestätigt	Wiederholt	Wiederholt
Kapazität	Unbestätigt	Unbestätigt	Bestätigt ¹	Bestätigt ¹
Integrität	Wiederholt	Bestätigt ¹	Bestätigt ¹	Bestätigt ¹

Diese Empfehlungen spiegeln sich auch in den Ergebnissen aus Tabelle XIV wieder und die QoS-Warnung des Bindings $\mathfrak{b}3$ lässt sich durch einen bestätigten oder wiederholten Nachrichtendienst beheben. Die Empfehlungen können allerdings nicht pauschalisiert werden, da die Nachrichtengröße, der Weg und die Dienstparameter über den optimalen Übertragungsdienst entsprechend der gestellten QoS-Anforderungen entscheiden. Zum Vorteil des Anwenders kann das Tool diese Zusammenhänge in der *Übertragungsdienstberatung* gut abschätzen und ihm eine quantifizierte Entscheidungsbasis bieten. Hierfür wählt er das zu bewertende Binding aus und dann werden die unterschiedlichen Nachrichtendienste in ihren Eigenschaften Last, Verzögerungszeit und Integrität nach Algorithmus 5 abgeschätzt. Das geschieht auf Basis der vorhandenen Leistungskenngrößen der Wegelemente unter der Annahme, dass die Änderung des Übertragungsdienstes keinen signifikanten Einfluss auf die Kenngrößen der Netzwerkelemente hat. Die notwendigen Abschätzungen der Fehlerwahrscheinlichkeit und Zeit wurden in den Abschnitten 7.3 und 8.3.4 definiert. Für die Last wird zuerst der mittlere *akkumulierte Datendurchsatz* $\bar{\Theta}_{\Sigma;\mathfrak{b}}$ abgeschätzt, den alle Nachrichtenklassen des Bindings im gesamten Netzwerk physisch erzeugen, also

$$\bar{\Theta}_{\Sigma;\mathfrak{b}} = \sum_{\mathfrak{m} \in \mathfrak{M}_{\mathfrak{b}}} \sum_{\mathfrak{w} \in \mathfrak{W}_{\mathfrak{m}}} \bar{\Theta}_{\mathfrak{m}|\mathfrak{w}} = \sum_{\mathfrak{m} \in \mathfrak{M}_{\mathfrak{b}}} \sum_{\mathfrak{w} \in \mathfrak{W}_{\mathfrak{m}}} p_{\mathfrak{I};\mathfrak{w}}^+ \cdot \bar{\lambda}_{\mathfrak{m}} \cdot S_{\text{phy};\mathfrak{m}}. \quad (8.24)$$

Setzt man diesen akkumulierten Datendurchsatz eines Bindings ins Verhältnis zu seinem Datendurchsatz, so gibt der resultierende mittlere *relative Datendurchsatz* $\bar{\Theta}_{\Pi;\mathfrak{b}}$ an, wie viele Bits pro Nachricht (bit/Msg) auf der Anwendungsschicht im Netzwerk physisch erzeugt werden.

$$\bar{\Theta}_{\Pi;\mathfrak{b}} = \frac{\bar{\Theta}_{\Sigma;\mathfrak{b}}}{\bar{\vartheta}_{\mathfrak{b}}}. \quad (8.25)$$

Beide Durchsätze werden in Tabelle XIV mit angegeben und eignen sich gut, um mit $\bar{\Theta}_{\Sigma;\mathfrak{b}}$ die lastverursachenden Bindings im Netzwerk zu identifizieren und mit $\bar{\Theta}_{\Pi;\mathfrak{b}}$ den Overhead der Übertragungsdienste einzuschätzen. Diesbezüglich dienen sie in Unterabschnitt 8.4.5 auch zur Lösungsfindung.

Zur Abschätzung werden die Übertragungsdienste automatisch parametrisiert. Hierzu wird als Erstes die Anzahl an notwendigen Wiederholungen $r_{\mathfrak{b}}^+$ oder $r_{\mathfrak{b}}^-$ bestimmt. Hierzu wird bei statistischen Garantien die Gleichung (7.16) direkt in (8.1) eingesetzt und die Ungleichung nach $r_{\mathfrak{b}}^+$ oder $r_{\mathfrak{b}}^-$ aufgelöst. Die Anzahl an notwendigen Wiederholungen ist dann von der Fehlerwahrscheinlichkeit der Übertragung und der erlaubten Toleranz $\varepsilon_{\xi^{\mathfrak{m}}}$ abhängig. Besitzt ein Binding mehrere QoS-Anforderungen an die Integrität und Zeit, so muss die kritischste, also kleinste, Toleranz ausgewählt werden. Dadurch wird der Worst-Case der QoS-Anforderungen an die Zeit erfasst, dass bereits $\varepsilon_{\tau^{\mathfrak{m}}}$ Nachrichten durch Fehler verloren

ALGORITHMUS 5. Algorithmus der Übertragungsdienstberatung.

1. Anwender wählt ein Binding aus
 2. Für jeden Übertragungsdienst
 3. Erstelle die Nachrichtenklassen
 4. Parametriere die Nachrichtenklasse automatisch
 5. Schätze die Fehlerwahrscheinlichkeiten nach Gleichung (7.16) ab
 6. Schätze den akkumulierten Datendurchsatz nach Gleichung (8.24) ab
 7. Schätze die Verzögerungszeit nach den Gleichungen (8.19) bis (8.21) ab
 8. Bestimme die QoS-Gewichte der Nachrichtenklassen nach Gleichung (8.10)
 9. Sortiere die Nachrichtenklassen nach ihrem QoS-Gewicht absteigend
 10. Präsentiere den Vergleich dem Anwender
-

gehen. Zusätzlich wird zur Auslegung ein Sicherheitsfaktor s verwendet. Der Sicherheitsfaktor sollte bei mindestens $s \geq 1.2$ liegen, so dass mit 20% überdimensioniert wird, um dadurch statistische Schwankungen in der Leistungsbewertung und Unterschiede zum realen Systemverhalten auszugleichen. Zusammenfassend gilt beim wiederholten Übertragungsdienst

$$r_b^+ \geq \left\lceil \frac{\ln \left(\min \left(\varepsilon_{\tau^m}, \varepsilon_{p_b^{-m}} \right) \right) - \ln(s)}{\ln \left(p_{\text{rpt};o,i}^- \right)} \right\rceil \quad (8.26)$$

und beim bestätigten Übertragungsdienst

$$r_b^- \geq \begin{cases} \left\lceil \frac{\ln \left(\min \left(\varepsilon_{\tau^m}, \varepsilon_{p_b^{-m}} \right) \right) - \ln \left(s \cdot p_{\text{AckD};o,i}^- \right)}{\ln \left(p_{\text{Rem};o,i}^- \right)} \right\rceil - 1, & \|\mathfrak{G}\| \leq 16; \\ \left\lceil \frac{\ln \left(\min \left(\varepsilon_{\tau^m}, \varepsilon_{p_b^{-m}} \right) \right) - \ln(s)}{\ln \left(p_{\text{AckD};o,i}^- \right)} \right\rceil, & \|\mathfrak{G}\| > 16. \end{cases} \quad (8.27)$$

Bei (m, k) -harten Garantien wird die gleiche Abschätzung genutzt und dann r_b^+ oder r_b^- linear variiert, bis die QoS-Anforderung mit s gerade erfüllt ist.

Im zweiten Schritt werden der Transmission- und Repeat-Interval-Timer ausgelegt. Die EN 14908 [54] empfiehlt einen Transmission-Timer, bei dem mindestens 99% aller Pakete innerhalb dieser Zeit ankommen, so dass die Last durch Wiederholungen niedrig ist. Bestehen allerdings Zeitanforderungen, so sind die Timer meist zu groß und die notwendigen Wiederholungen kommen nicht innerhalb der oberen Grenze der Verzögerungszeit an. Deshalb wird aus der Verteilung der Verzögerungszeit des Bindings das Quantil t_q der oberen Grenze bei $1 - \varepsilon_{\tau^m} \cdot s$ bestimmt und entsprechend den Verhältnissen der mittleren Verzögerungszeit in

Gleichung (8.19) und (8.20) können die Timer abgeschätzt werden. Bei einem wiederholten Übertragungsdienst ist der Anteil der Zeiten

$$t_q \approx \bar{\tau}_{\text{rpt};o,i} \cdot (1 - (p_{\text{rpt};o,i}^-)^r) + \sum_{i=1}^r i \cdot T_{\text{RT};b} \cdot p_{\text{rpt};o,i}^+ \cdot (p_{\text{rpt};o,i}^-)^i \quad (8.28)$$

mit t_q so, dass $\mathbf{F}_{\tau_{b;o,i}}(t_q) = 1 - s \cdot \varepsilon_{\tau^m}$. Daraus folgt

$$T_{\text{RT};b} \leq \frac{\left(\bar{\tau}_{\text{rpt};o,i} \cdot (p_{\text{rpt};o,i}^+)^r - t_q\right) \left(p_{\text{rpt};o,i}^- - 1\right)^2}{p_{\text{rpt};o,i}^+ \cdot p_{\text{rpt};o,i}^- \left[\left(p_{\text{rpt};o,i}^-\right)^r \left(1 + r \left(1 - p_{\text{rpt};o,i}^-\right)\right) - 1 \right]} \quad (8.29)$$

Unter der Annahme $\tau_{\text{AckD};o,i} \approx \tau_{\text{Rem};o,i}$ gilt diese Gleichung auch für $T_{\text{TT};b}$ mit $\tau_{\text{AckD};o,i}$ als $\tau_{\text{rpt};o,i}$. Bei (m, k) -harten Garantien werden die Gleichungen (8.19) und (8.20) noch einmal optimiert⁸.

Diese Parametrierung ist in der Hinsicht optimal, dass die Timer und die Wiederholungsrate aufeinander abgestimmt sind und dabei eine geringe Last erzeugt wird. So ist es unnötig, Wiederholungen einzuplanen, die zu spät ankommen. Genauso müssen die Timer erlauben, dass alle notwendigen Wiederholungen rechtzeitig ankommen. Davon losgelöst hat auch die Warteschlangenlänge in den Geräten einen Einfluss auf die Fehlerwahrscheinlichkeit und die Verzögerungszeit eines Bindings. Dabei gibt es einen Trade-off zwischen beiden Leistungskenngrößen für unterschiedliche Warteschlangenlängen und somit eine bestimmbare, optimale Warteschlangenlänge⁹. Auf eine Auslegung wurde verzichtet, da die Warteschlangenlängen der Geräte selten von extern zu parametrieren sind.

8.4.3. VERBESSERUNG EINZELNER DIENSTQUALITÄTEN DURCH PRIORITÄTEN

Die Priorisierung ist eine andere Möglichkeit, um die Dienstqualität einzelner Bindings zu verbessern. Eine priorisierte Nachricht wird bei der Übertragung bevorzugt behandelt, durchläuft separate Warteschlangen und besitzt eigene Zeitschlitze beim Buszugriff. Dadurch hat sie gerade bei ausgelasteten Kanälen eine geringere Verzögerungszeit und Fehlerwahrscheinlichkeit. So lässt sich im Beispiel die Verzögerungszeit und Fehlerwahrscheinlichkeit des Bindings **b11** halbieren und die QoS-Anforderung erfüllen.

8 Der Faltungsaufwand wird reduziert, indem die gefalteten Verteilungen einmalig bestimmt und danach nur noch zueinander verschoben werden.

9 Im WANs gibt es das »resource dimensioning paradigm«, das eine kleine Warteschlangenlänge und große Bandbreite empfiehlt. Ziel ist es, dass Bursts schnell abklingen. Dabei steigen zwar kurzzeitig die Fehlerwahrscheinlichkeiten, aber die Verzögerungszeiten bleiben kurz. Dadurch kann schneller wiederholt und letztendlich auch die Integrität verbessert werden.

Allerdings hat die Priorisierung auch Nachteile. So werden die prioren Zeitschlitzte nach jedem Telegramm bereitgehalten und reduzieren damit den maximalen Datendurchsatz. Weiterhin können priore Nachrichten geringer priorisierte Nachrichten teilweise blockieren. Wird die Priorität in einem Segment doppelt vergeben, so steigt die Kollisionswahrscheinlichkeit drastisch. Letztendlich behandelt LON Prioritäten nicht-deterministisch und Nachrichten werden automatisch unpriorisiert versendet, wenn der Kanal gering ausgelastet ist oder wenn gerade zuvor eine priore Nachricht von dem Gerät versendet wurde [40]. Deshalb wird die Fehlerwahrscheinlichkeit des Bindings `b11` auch nicht Null.

Diese starken Wechselwirkungen auf die Geräte und die Lastsituation in den Segmenten machen eine neue Leistungsabschätzung notwendig, sobald ein Binding anders priorisiert wird. Deshalb ist es unter der Effizienz-Anforderung 1 nicht sinnvoll, alle Kombinationen für eine Priorisierung durchzuprobieren, sondern die Freiheitsgrade müssen drastisch reduziert werden.

Prinzipiell sind bei LON bis zu 127 *Prioritätsslots* realisierbar [54], die verfügbaren Transceiver unterstützen jedoch maximal 16 Prioritätsslots, üblicherweise nur vier (`LonChannel.priorities`). Diese geringe Anzahl an Varianten kann das Tool bei der *Prioritätsberatung* mit einer festen Prioritätenliste durchprobieren. Zuvor müssen allerdings die zur Priorisierung geeignetsten Bindings bestimmt werden, wobei wahlweise vorhandene Prioritäten berücksichtigt werden. Dabei ist zu beachten, dass zwar Bindings als Prior markiert werden, allerdings jeder sendende Port einen zugewiesenen Prioritätsslot benötigt. Dies gilt auch für Router, so dass wenn eine priorisierte Nachricht von einem Sender mit einem hohen Prioritätsslot gesendet wird und diese einen Router ohne zugewiesenen Slot passiert, so sendet der Router sie ohne Priorität. Dementsprechend sollten sowohl Sender und Router ein Prioritätsslot zugewiesen werden.

Der Algorithmus 6 realisiert die Vergabe der Prioritätsmarkierungen und Prioritätsslot anhand des QoS-Gewichts einer unpriorisierten Parametrierung. Auf diese Weise werden die Geräte mit der höchsten QoS-Risikoklasse und dem größten Bedarf bevorzugt. Die *Prioritätsberatung* listet die Ergebnisse und Konfigurationen der einzelnen Versuche auf und bewertet sie mit dem Qualitätsmaß \bar{Q} oder Q_{\min} aus Unterabschnitt 8.2.5, wodurch auch neu aufgetretene QoS-Fehler Beachtung finden. Der Anwender kann am Ende vergleichen, welche Konfiguration den besten Erfolg verspricht und sich diese anzeigen lassen.

8.4.4. VERBESSERUNG DER LASTSITUATION DURCH ÜBERTRAGUNGSDIENSTE UND PRIORITÄTEN

Gibt es im Netzwerkentwurf einen überlasteten Kanal oder Routerport, so entstehen bei den passierenden Nachrichtenklassen gehäuft QoS-Fehler. In diesem Fall ist es nicht hilfreich, den fehlerhaften Klassen zuverlässigere Übertragungsdienste oder Prioritäten zuzuweisen, da diese nur noch mehr Last produzieren. Häufig

 ALGORITHMUS 6. Algorithmus der Prioritätsberatung.

1. Setze alle nicht durch den Anwender ausgeschlossenen Prioritäten zurück
 2. Leistungsbewertung der Konfiguration
 3. Bestimme alle QoS-Gewichte
 4. Für jeden sendenden Port
 5. Summiere die QoS-Gewichte der passierenden Bindings (vgl. Gleichung (8.8))
 6. Für $i = 1$ bis $\max(\text{LonChannel.priorities})$
 7. Für jeden Kanal
 8. Vergebe i Prioritätsslots aufsteigend nach dem summierten QoS-Gewicht an die Ports und markiere jedes passierende Binding mit QoS-Fehler als Prior
 9. Leistungsbewertung der Konfiguration
 10. Sortiere die Ausgangskonfiguration und Vorschläge nach dem Qualitätsmaß \bar{Q} o. Q_{\min} absteigend
 11. Präsentiere Vergleich dem Anwender
-

stellen im Gegenteil unbestätigte Übertragungsdienste die Lösung dar, da oft aus Vorsicht bestätigte Dienste verwendet werden, ohne die Nachteile abzuschätzen. So ist es im Beispiel aus Unterabschnitt 5.1.2 von Anwendungssicht her nahelegend, die wichtigen Nachrichten des Rauchmelders **b10** und des Alarms **b11** bestätigt und authentifiziert zu übertragen. Vergrößert man kontinuierlich das Beispiel, indem man mehr Büroräume und Flure hinzufügt, so stellt sich als ein Flaschenhals der äußerst leistungsstarke Backbone heraus, da die Bindings **b10** und **b11** diesen mit ihren Antworten an die Zentrale überschwemmen. Die Verzögerungszeiten und Fehlerraten sinken drastisch, wenn stattdessen unbestätigte oder wiederholte Übertragungsdienste eingesetzt werden.

Hieraus folgt als erster Lösungsansatz die *Übertragungsdienstreduktion*, bei der versucht wird, die Überlastsituation ohne Eingriff in die Prozesse zu lösen, indem die Übertragungsdienste und Prioritäten netzwerkweit bedarfsgerecht ausgelegt werden. Die Auslegung setzt vorerst alle Übertragungsdienste auf den unbestätigten Dienst zurück (mit parametrierbaren Ausnahmen), um auf dieser Konfiguration die notwendigen Übertragungsdienste zu vergeben. Der Algorithmus 7 vergibt in mehreren Iterationen die geeigneten Übertragungsdienste und Prioritäten nach Algorithmus 5 und 6, solange das zu einer netzwerkweiten Verbesserung des QoS-Fehlers führt. Dabei werden die Übertragungsdienste nicht alle auf einmal geändert, da ansonsten die Annahme geringer Änderungen des Netzwerkverhaltens von Algorithmus 5 verletzt wird und es zu Instabilitäten im Fix-Punkt-Ansatz und Fehlparametrierungen kommt. Stattdessen werden blockweise die h Bindings mit dem kleinsten QoS-Gewicht angepasst. Das Resultat ist eine Pareto-optimale Vergabe der Übertragungsdienste und Prioritäten nach der QoS-Risikoklasse entsprechend Unterabschnitt 8.4.1.

Auf diese Weise kann eine geringe Überlast, die durch unangemessene Über-

ALGORITHMUS 7. Algorithmus der Übertragungsdienstreduktion.

1. Weise allen Bindings den unbestätigten Übertragungsdienst zu, die nicht antwortend, authentifiziert oder vom Nutzer ausgeschlossen sind
 2. Wiederhole
 3. Vergebe die Prioritäten mit Algorithmus 6
 4. Wähle die Konfiguration mit dem größten Qualitätsmaß \bar{Q} o. Q_{\min}
 5. Für h Bindings mit dem kleinsten, negativen QoS-Gewicht
 6. Bestimme den passenden Übertragungsdienst mit Algorithmus 5
 7. Leistungsbewertung der Konfiguration
 8. Wiederhole solange das Qualitätsmaß \bar{Q} o. Q_{\min} zunimmt
-

tragungsdienste entstanden ist, meist behoben werden. Führt dieser Ansatz nicht zum Ziel, kann die Überlast nur noch durch eine Restrukturierung des Netzwerks (Kapitel 11) oder durch eine Anpassung der Anwendungsparameter gelöst werden. Im letzten Punkt kann das Tool den Anwender weiter beraten.

8.4.5. VERBESSERUNG DER LASTSITUATION DURCH ANPASSUNG DER ABTASTPARAMETER

Ist ein oder sind mehrere Kanäle überlastet und ist die Übertragungsdienstreduktion nicht erfolgreich, muss die Ankunftsrate auf der Anwendungsschicht reduziert werden. Dies ist durch die Anpassung der Abtastparameter und der Min- / Max-Send-Time möglich. Hierbei sind Wechselwirkungen zu beachten. So kann ein zu großes Send-On-Delta in Regelungen zu Schwingungen durch Alias-Frequenzen oder sogar zu Instabilitäten führen. In beiden Fällen entsteht dadurch mehr Verkehr. Unter Prämisse 14 wird davon ausgegangen, dass die Prozesse so ausgelegt sind, dass sie kleineren Änderungen gegenüber tolerant sind. Weiterhin werden nach Unterabschnitt 8.4.1 Bindings mit Sicherheitsanforderungen nicht verändert. Jetzt ist zu klären, wie diese Änderungen unter den restlichen Bindings verteilt werden.

Ein naheliegender Ansatz ist es, die Parametrierung der Bindings zu ändern, die den meisten Durchsatz im Kanal erzeugen. Allerdings können bei Bindings nur die Ursprungsdatenpunkte parametrierbar sein, von denen einige eine Min- und Max-Send-Time und nur λ -Quellen Abtastparameter besitzen. Dadurch sind oft die aktivsten Bindings nicht zu ändern. Im Beispielnetzwerk mit dem überlasteten Backbone wird die Hauptlast durch das Binding b11 erzeugt, das von einem λ -Filter stammt, dessen Lastverhalten nicht parametrierbar ist. Außerdem vernachlässigt dieser Ansatz den Zusammenhang zwischen den Bindings und verschiedenen Kanälen, die vielleicht aus ähnlichen Gründen überlastet sind.

Das Verkehrs- und das Systemmodell erlauben eine weitergehende Analyse der Zusammenhänge in der *Ankunftsratenreduktion*. So haben mehrere überlastete

Kanäle und Ports häufig die gleichen Hauptursachen, welche nicht offensichtlich in passierenden Bindings liegen müssen. So führt sich die Ankunftsrate des Bindings \mathfrak{b}_{11} über den λ -Filter im Verkehrsmodell auf mehrere λ -Quellen zurück. Dabei zeigt sich in Abbildung 5.8, dass auch \mathfrak{b}_{10} die gleichen λ -Quellen besitzt. Verringert man also an diesen Datenpunkten die Ankunftsrate nur etwas, so reduziert sich die Last im ganzen Netzwerk deutlich.

Um den Einfluss der Datenpunkte auf die Netzwerklast zu bestimmen, wird das Verkehrsmodell herangezogen. Die Gleichung (6.8) definiert das Gerätemodell als Matrix der mittleren Ankunftsrate der Ein- und Ausgangsdatenpunkte. Sie lässt sich auf das ganze Netzwerk verallgemeinern. Sind \mathfrak{J} und \mathfrak{D} die Menge aller Ein- und Ausgangsdatenpunkte im ganzen Netzwerk, so ist

$$\vec{\lambda}_{\mathfrak{D}} = \vec{\lambda}_{\mathfrak{Q};\mathfrak{D}} + \mathbf{V}\vec{\lambda}_{\mathfrak{J}} \quad (8.30)$$

mit der netzwerkweiten Matrix \mathbf{V} der Übertragungswahrscheinlichkeiten. In gleicher Form lässt sich der Einfluss der Bindings auf den Verkehr der Eingangsdatenpunkte aus Gleichung (6.9) verallgemeinern zu

$$\vec{\lambda}_{\mathfrak{J}} = \mathbf{P}^+\vec{\lambda}_{\mathfrak{D}} \quad (8.31)$$

mit der netzwerkweiten Matrix \mathbf{P}^+ der Erfolgswahrscheinlichkeiten der Bindings. Aus \mathbf{V} und \mathbf{P}^+ lässt sich nun die gewichtete *Adjazenzmatrix* $\mathbf{A}_{\mathfrak{J}\cup\mathfrak{D}}$ [11, 29, 123] aller Datenpunkte $\mathfrak{J}\cup\mathfrak{D}$ bilden, mit

$$a_{\mathfrak{e},\mathfrak{k}} = \begin{cases} v_{\mathfrak{e},\mathfrak{k}}, & \mathfrak{e} \in \mathfrak{J} \wedge \mathfrak{k} \in \mathfrak{D}; \\ p_{\mathfrak{b}_{\mathfrak{e},\mathfrak{k}}}^+, & \mathfrak{e} \in \mathfrak{D} \wedge \mathfrak{k} \in \mathfrak{J} \wedge \mathfrak{b}_{\mathfrak{e},\mathfrak{k}} \in \mathfrak{B}; \\ 0 & \text{sonst.} \end{cases} \quad (8.32)$$

Es folgt dann aus dem Vektor der Quellprozesse $\vec{\lambda}_{\mathfrak{Q};\mathfrak{J}\cup\mathfrak{D}}$ aller Datenpunkte eine allgemeine Darstellung des Verkehrsmodells.

$$\begin{aligned} \vec{\lambda}_{\mathfrak{J}\cup\mathfrak{D}} &= \mathbf{A}_{\mathfrak{J}\cup\mathfrak{D}}^T \cdot \vec{\lambda}_{\mathfrak{J}\cup\mathfrak{D}} + \vec{\lambda}_{\mathfrak{Q};\mathfrak{J}\cup\mathfrak{D}} \\ \vec{\lambda}_{\mathfrak{J}\cup\mathfrak{D}} &= (\mathbf{E} - \mathbf{A}_{\mathfrak{J}\cup\mathfrak{D}}^T)^{-1} \cdot \vec{\lambda}_{\mathfrak{Q};\mathfrak{J}\cup\mathfrak{D}}. \end{aligned} \quad (8.33)$$

Das Verkehrsmodell ist nur lösbar, wenn $(\mathbf{E} - \mathbf{A}_{\mathfrak{J}\cup\mathfrak{D}}^T)$ regulär ist. Dies ist gleichbedeutend mit Prämisse 11. Hieraus leitet sich die *Sensibilitätsmatrix*

$$\mathbf{Z} = (\mathbf{E} - \mathbf{A}_{\mathfrak{J}\cup\mathfrak{D}}^T)^{-1} \quad (8.34)$$

ab. Durch Multiplikation mit dem Einheitsvektor \vec{e} bestimmt sich der Einfluss der Datenpunkte im Verkehrsmodell zu $\mathbf{Z} \cdot \vec{e}$.

Für die Diagnose ist allerdings der Einfluss auf den Durchsatz der überlasteten Elemente entscheidend. Dies wird gut durch den *relativen Datendurchsatz* aus Gleichung (8.25) ausgedrückt, also der Datendurchsatz, den eine Nachricht auf Anwendungsschicht im ganzen Netzwerk physisch erzeugt. Für jeden Datenpunkt gilt dann für ausgehenden Bindings

$$\bar{\Theta}_{\Pi; \epsilon \in \mathcal{J} \cup \mathcal{D}} = \sum_{\mathfrak{b} \in \mathfrak{B}_\epsilon | \epsilon \in \mathcal{D}} \bar{\Theta}_{\Pi; \mathfrak{b}}, \quad (8.35)$$

so dass der Vektor $\mathbf{Z} \cdot \vec{\bar{\Theta}}_{\Pi; \mathcal{J} \cup \mathcal{D}}$ den Einfluss angibt, den eine Änderung der Ankunftsrate auf den Durchsatz im ganzen Netzwerk hat.

Für die Reduktion der Ankunftsrate sind vor allem die überlasteten Kanäle wichtig. Zusätzlich sollten nach Unterabschnitt 8.4.1 nur Bindings mit Komfortanforderungen geändert werden und vorzugsweise jene mit freien Kapazitäten in den QoS-Anforderungen ($Q_{\xi^x} > 0$). Um beides wird Gleichung (8.35) ergänzt zu

$$\bar{\Theta}_{\Pi; \epsilon \in \mathcal{J} \cup \mathcal{D}}^* = \sum_{\mathfrak{b} \in \mathfrak{B}_\epsilon | \epsilon \in \mathcal{D}} \frac{K_{\mathfrak{b}}}{\bar{\vartheta}_{\mathfrak{b}}} \sum_{\mathfrak{m} \in \mathfrak{M}_{\mathfrak{b}}} \sum_{\mathfrak{w} \in \mathfrak{W}_{\mathfrak{m}} | \mathfrak{w} \in \mathcal{C}} -\min(G_{\epsilon}; 0) \cdot p_{\mathfrak{I}; \mathfrak{w}}^+ \cdot \bar{\lambda}_{\mathfrak{m}} \cdot S_{\text{phy}; \mathfrak{m}} \quad (8.36)$$

mit der noch freien Kapazitäten $K_{\mathfrak{b}}$ in den QoS-Anforderungen $\Xi_{\mathfrak{b}}$ eines Bindings, die bestimmt ist durch das niedrigste, positive Qualitätsmaß

$$K_{\mathfrak{b}} = \min(0; \min(Q_{\xi^x} | \xi^x \in \Xi_{\mathfrak{b}} \wedge R_{\xi^x} = 1)). \quad (8.37)$$

Damit kann der *Sensibilitätsvektor* \vec{z} der Datenpunkte bestimmt werden.

$$\vec{z}_{\mathcal{J} \cup \mathcal{D}} = \mathbf{Z} \cdot \vec{\bar{\Theta}}_{\Pi; \mathcal{J} \cup \mathcal{D}}^*. \quad (8.38)$$

Er spiegelt den Einfluss der Datenpunkte auf den Durchsatz in den überlasteten Kanälen unter Gewichtung der QoS-Risikoklassen wider. Anhand dieser Ergebnisse kann der Anwender die einflussreichsten und, bezüglich der QoS-Risikoklassen, sichersten Änderungspunkte identifizieren. Als weitere Hilfestellung können die Min-Send-Time, Max-Send-Time und die Abtastparameter der Datenpunkte mit w_U , w_L , w_δ in ihrem Einfluss bewertet werden. Der Vorteil des Sensibilitätsvektors ist, dass immer Ausgangsdatenpunkte mit Quellprozessen den größten Einfluss besitzen und damit vorwiegend parametrierbar sind.

Der Sensibilisierungsvektor bietet sich auch für eine Optimierung der Parameter an. Hierbei handelt es sich aufgrund des Netzwerkverhaltens und des Verkehrsmodells um ein stark nichtlineares Problem mit vielen Freiheitsgraden. Es lässt sich zerlegen, indem die Ankunftsrate der Datenpunkte gemäß ihrem Gewicht im Sensibilisierungsvektor reduziert wird. Damit zerfällt das Problem, die Last um $q = 10\%$ zu reduzieren, auf die Frage, die Lasten in den Datenpunkten

ALGORITHMUS 8. Algorithmus der Beratung zur Ankunftsratenreduktion.

1. Bestimme die Adjazenzmatrix des Netzwerks nach Gleichung (8.32)
 2. Berechne die Sensibilitätsmatrix mit Gleichung (8.34)
 3. Berechne den Sensibilitätsvektor $\vec{z}_{\mathcal{J} \cup \mathcal{D}}$ mit Gleichung (8.38)
 4. Entferne aus $\vec{z}_{\mathcal{J} \cup \mathcal{D}}$ alle Datenpunkte, die nicht parametrierbar sind
 5. Sortiere die restlichen Datenpunkte in $\vec{z}_{\mathcal{J} \cup \mathcal{D}}$ absteigend nach ihrem Wert
 6. Gewichte den Einfluss der Min-Send-Time, Max-Send-Time und der Abtastparameter mit w_U, w_L, w_δ
 7. Präsentiere dem Anwender die Ergebnisse
-

um den Anteil $q \cdot z_o / \sum z_{\mathcal{J} \cup \mathcal{D}}$ zu verringern. Dann können gezielt die Modelle des Verkehrsmodells benutzt werden, um in Teiloptimierungen die Min-Send-Time, Max-Send-Time und die Abtastparameter anzupassen. Hierbei ist allerdings zu beachten, dass der Sensibilitätsvektor nur den Einfluss im aktuellen Modellzustand widerspiegelt und nur lokal für kleine Änderungen gültig ist. Der Algorithmus 9 stellt den Optimierungsalgorithmus vor, der das Pareto-Optimum der Ankunftsratenreduktion nach den Gewichtungen in Unterabschnitt 8.4.1 bestimmt. Hierbei werden zwar die QoS-Risikoklassen beachtet, allerdings keine Feinheiten in den Prozessen, so dass auf jeden Fall der Anwender das Ergebnis kontrollieren muss.

ALGORITHMUS 9. Algorithmus der Ankunftsratenoptimierung.

1. Wiederhole
 2. Reduziere die Übertragungsdienste nach Algorithmus 7
 3. Schätze die notwendige Lastreduktion q
 4. Berechne den Sensibilitätsvektor mit Algorithmus 8
 5. Normiere den Sensibilitätsvektor auf $\sum z_{\mathcal{J} \cup \mathcal{D}} \doteq 1$
 6. Für alle Ausgangsdatenpunkte o
 7. Bestimme die anteilhafte Ankunftsratenänderung $q_o = q \cdot s \cdot z_o$
 8. Optimierte die Min-Send-Time, Max-Send-Time und die Abtastparameter im Anteil von w_U, w_L, w_δ , bis die Ankunftsrate um q_o reduziert ist
 9. Wiederhole, bis die QoS-Fehler der Kanäle ihr Minimum erreichen
 10. Präsentiere dem Anwender die Ergebnisse und Änderungen
-

8.5. ZUSAMMENFASSUNG

Die meisten Teilziele aus Abschnitt 3.3 werden durch die Fehleranalyse und Nutzerberatung anvisiert, da sie die Auswertung realisiert und damit die für den Anwender relevanten Ergebnisse liefert.

Das Teilziel 2 wird mit den vorgestellten QoS-Anforderungen erfüllt, die eine Bewertung der Qualität des Kommunikationsmediums ermöglichen. Hierbei werden die Qualitätsmerkmale 2 bis 5 nicht direkt quantifiziert, sondern unter Prämisse 16 auf die unterschiedlichen QoS-Anforderungen der Anwendungen heruntergebrochen. Neben statistischen QoS-Garantien können mit den (m, k) -harten Garantien auch Fehlerfolgen beschränkt werden. Das ist insbesondere für Regelungen von Bedeutung. Das Qualitätsmaß aus Unterabschnitt 8.2.5 führt die QoS-Anforderungen wieder auf ein Maß zusammen, welches dem Anwender als Orientierung dienen kann.

Jeder QoS-Fehler stellt auch einen Entwurfsfehler dar, womit die in Unterabschnitt 8.2.4 vorgestellte Fehleridentifikation das Teilziel 3 erfüllt. Neben den QoS-Fehlern gibt es andere Entwurfsfehler, die durch Sinnfälligkeitstests gefunden werden können, wie zum Beispiel fehlerhafte Routing-Tabellen oder Min-Send-Times, die größer als Max-Send-Times sind. Sie gehen nach Prämisse 16 nicht in die Qualitätsbewertung ein.

Das Teilziel 4 zur Beratung des Anwenders bei der Problemlösung wird durch die Fehlerbäume und die unterschiedlichen Beratungstools zur Parametrierung erfüllt. Dabei ermöglichen die Fehlerbäume dem Anwender vor allem das einfache Verstehen der Ursachen von QoS-Fehlern und ergänzen charakteristische Verbesserungsvorschläge. Darüber hinaus berät die Übertragungsdienst- und Prioritätsberatung den Anwender bei der Verbesserung einzelner QoS-Anforderungen durch eine angemessene Parametrierung. Die Übertragungsdienst- und Ankunfts-ratenreduktion unterstützt den Anwender bei der komplexen Aufgabenstellung, im ganzen Netzwerk die Auslastung zu reduzieren.

Die aufgestellten Algorithmen 6 bis 9 bestimmen jeweils das Pareto-Optimum nach den Strategien in Unterabschnitt 8.4.1. Hierbei kann nicht sichergestellt werden, ob das globale Optimum im Sinne der höchsten Qualität gefunden wird. Vielmehr verwenden die Algorithmen das vorhandene Modell- und Diagnosewissen, um mit einer geringen Anzahl an Iterationen der aufwendigen Leistungsbewertung nach den Anforderungen 1, 3 und 5 das Pareto-Optimum zu finden.

Der Anwender startet die unterschiedlichen Beratungstools selektiv und kann sich dabei an dem Entscheidungsgraphen aus Abbildung 8.9 orientieren. In dem Beispiel vom Büroraum mit zwei Fluren und jeweils 10 Büroräumen kann der Anwender auf diese Weise die QoS-Fehler beim Binding **b3** und **b11** mit Hilfe der Übertragungsdienst- und Prioritätsberatung beheben. Änderungen am Netzwerkentwurf muss er selbst durchführen. Technisch ist es möglich dies zu automatisieren und damit den Anforderungen 1 und 6 gerecht zu werden. Es wurde allerdings aus zwei Gründen darauf verzichtet: Zum einen erhöht sich das Verständnis des Anwenders über das Netzwerkverhalten nur, wenn er das Tool und seine Beratungsfunktionen bewusst nutzt und sich damit Erfahrungen zum gezielten Netzwerk Performance Engineering erarbeitet. Zum anderen kann nur der Anwender die Auswirkungen von Änderungen auf die Prozesse abschätzen und ist

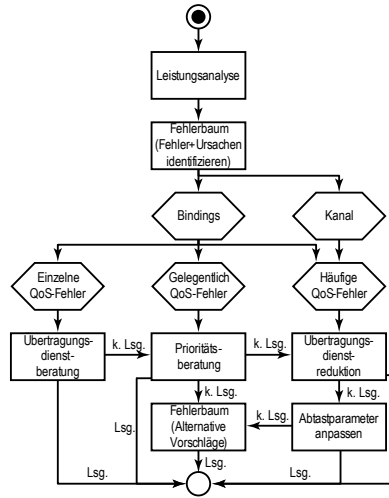


ABBILDUNG 8.9. Entscheidungsbaum zur Auswahl des richtigen Diagnosewerkzeuges (Abbruch jederzeit möglich und nicht eingezeichnet; Lsg. - Lösung; k. Lsg. - keine Lösung).

letztendlich für den Netzwerkentwurf verantwortlich. Eine Erweiterung der Nutzerberatung mit zusätzlichen Qualitätsmodellen zur *Regelqualität* (en. *Quality of Control*) könnte den Anwender in dieser Fragestellung unterstützen (Kapitel 11), allerdings sind die notwendigen Streckenmodelle komplexer und spezifischer als die Verkehrsmodelle [108, 217] sowie schlechter zu verallgemeinern.

Beim Verfassen dieser Arbeit wurde die Nutzerberatung noch implementiert, so dass keine definitiven Aussagen zur Genauigkeit und Performance der Algorithmen getroffen werden können. Erste Zwischenergebnisse bestätigen allerdings die gute Anwendbarkeit der vorgestellten Algorithmen.

TEIL
III.

ZUSAMMENFASSUNG

KAPITEL

9.

GRENZEN DER ARBEIT

*Aber man verlangt vom Forscher, dass er
Beweise liefert, wenn es sich zum Beispiel um
die Entdeckung eines großen Berges handelt,
verlangt man, dass er große Steine mitbringt.*

Antoine de Saint-Exupéry

BISHER wurde das Tool noch nicht in einem großen Versuch getestet. Zum einen hat aus firmenpolitischen, Sicherheits- oder Gewährleistungsgründen noch kein Betreiber sein Netzwerk für Messungen dieser Größenordnung geöffnet. Zum anderen war bisher die Erstellung der Gerätemodelle aus Gerätebeschreibungen aufwendig. Die Arbeit von GÜRTLER [75] ermöglicht diesbezüglich eine Vereinfachung durch die automatische Modellierung aus Messungen.

Die Grenzen der Arbeit werden vor allem durch das Verkehrsmodell vorgegeben, von dem die Genauigkeiten der Leistungsbewertung und Fehleranalyse maßgeblich abhängen. In Abschnitt 6.8 werden die Grenzen des Verkehrsmodells kritisch beleuchtet. Dabei verbergen sich die Probleme nicht in der Genauigkeit der Algorithmen und Modelle, sondern vielmehr in der Passfähigkeit der analysierten Daten. Deshalb ist vor allem eine Differenzierung der Gerätemodelle und der charakteristischen Signalverläufe notwendig. Dabei wird die Übertragbarkeit auf unterschiedliche Netzwerkentwürfe positiv eingeschätzt. Der Vergleich zu Simulationsergebnissen in Tabelle IX und zu Messungen in Tabelle XI hat gezeigt, dass die Abschätzungen in den richtigen Größenordnungen liegen. Letztendlich ist eine präzise Verkehrsmodellierung für die Qualitätsverbesserung bei der Konfigurierung nicht entscheidend, sondern dass die Verhältnismäßigkeiten unter den Geräten stimmen.

Deshalb wird auch die Fehleranalyse und Nutzerberatung positiv eingeschätzt. Obwohl die statistischen und (m, k) -harten Garantien genaue Verteilungen zur Berechnung benutzen, entscheidet sich ein Großteil der Parametrierung nach den Verlustraten und den Mittelwerten der Verzögerungszeiten, die recht gut abzuschätzen sind. Der Sicherheitsfaktor s der Parametrierung gewährleistet zusätzlich eine höhere Toleranz. Solange deshalb die Verhältnismäßigkeiten im Verkehrsmodell stimmen, sind die Geräte zueinander gut ausgelegt.

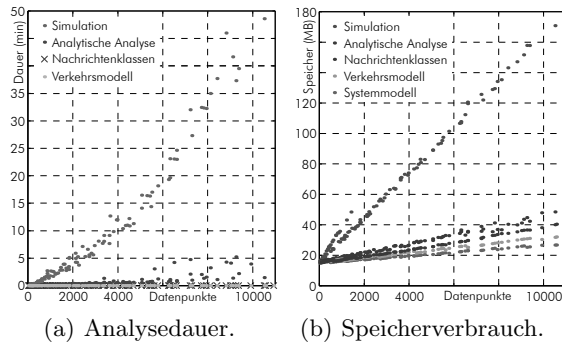


ABBILDUNG 9.1. Vergleich der einzelnen Analyseschritte.

Die Abbildung 9.1 vergleicht die Analysedauer und den Speicherverbrauch der einzelnen Analyseschritte. Hierfür wurde das Beispiel aus Unterabschnitt 5.1.2 entsprechend vervielfältigt¹. Es ist offensichtlich, dass die Simulation die meisten Ressourcen benötigt. Die Analysedauer der Simulation hängt ferner stark von den Unterschieden in den Ankunftsraten der Geräte ab, wie in Abschnitt 7.4 erörtert wurde. Grundsätzlich zeigt sie eine Komplexität von $O(n^2)$ in Abhängigkeit von der Anzahl an Datenpunkten. Der Speicherverbrauch der Simulation entwickelt sich linear dazu. Damit können prinzipiell auch Netzwerke mit zehntausend Datenpunkten² in einer Stunde analysiert werden, wobei allerdings Anforderung 1 nach Effizienz nicht zu halten ist. Eine Erweiterung der analytischen Analyse um eine Abschätzung der Verzögerungszeit aus den Verteilungen der Warteschlangenlängen könnte eine Möglichkeit bieten, die Simulation zu ersetzen.

Die zunehmende Etablierung der Gebäudesimulation zur Energieverbrauchsabschätzung [86, 122] stimmt zuversichtlich, dass diese Probleme bewältigt werden. Unzweifelhaft hat die automatische Modellierung der Simulationsmodelle aus Entwürfen einen wichtigen Beitrag an dieser Entwicklung [25, 30, 88, 151] geleistet.

- 1 Zur Vereinfachung der Simulation wurden die Ankunftsraten der Bindings **b7** bis **b9** verhundertfachen (vgl. Fußnote 8 auf Seite 117).
- 2 Das Beispiel mit fünf Fluren und jeweils 100 Büroräumen.

KAPITEL

10.

ZUSAMMENFASSUNG UND ABGLEICH MIT DER ZIELSTELLUNG

*Wer das Ziel kennt, kann entscheiden;
wer entscheidet, findet Ruhe;
wer Ruhe findet, ist sicher;
wer sicher ist, kann überlegen;
wer überlegt, kann verbessern.*

Konfuzius

IN der Einleitung wurde der zunehmende Bedarf für eine Qualitätsbewertung von Entwürfen verteilter Automationsnetzwerke herausgearbeitet. Hierbei wurde deutlich, dass Methoden zur Leistungsbewertung von Netzwerken diese Aufgabe gut erfüllen können und das derzeit bestehende Problem in der fehlenden Integration dieser zum Teil komplexen Methoden in die Konfigurierung besteht. Ein Systemintegrator, der in kurzer Zeit ein großes Netzwerk planen muss, hat weder die Zeit noch zum Teil die Kenntnisse, solche Methoden anzuwenden. Diese Analyse in der Einleitung bestätigt sich in Gesprächen mit Industrievertretern. Oft schildern sie, dass ein Großteil der Entwurfsfehler erst bei der Inbetriebnahme oder im laufenden Betrieb gefunden wird. Dabei geschieht es, dass ganze Entwurfsteile revidiert werden müssen. Zur Vorsicht werden deshalb kritische Anlageanteile in Testräumen komplett vorinstalliert. Dadurch wird notgedrungen die Trennung von Konfigurierung und Integration aufgehoben und die Inbetriebnahme zu einer Konfigurierungs- und Nachbesserungsphase unter hohem Zeitdruck.

Aus dieser Situation wurden in Kapitel 3 die Zielstellung und die Anforderungen der Arbeit abgeleitet. Als Ziel wurde die Erhöhung der Qualität von Netzwerkentwürfen von Steuer- und Regelungsnetzwerken durch ein Control Network

Performance Engineering formuliert. Dabei wurden die folgenden Teilziele und Anforderungen spezifiziert:

TEILZIELE

- Teilziel 1: Modellierung des Leistungsmodells
- Teilziel 2: Bewertung der Qualität des Kommunikationsmediums
- Teilziel 3: Identifikation von Entwurfsfehlern
- Teilziel 4: Beratung zur Fehlerbehebung

ANFORDERUNGEN

- Anforderung 1: Effizienz
- Anforderung 2: Verständlichkeit
- Anforderung 3: Genauigkeit
- Anforderung 4: Differenziertheit
- Anforderung 5: Automatisierung
- Anforderung 6: Integration

Das Teilziel 1 wird mit den Modellen und Methoden aus Kapitel 5 und 6 umfassend erfüllt. Die automatische Modellierung des Systemmodells aus einer Entwurfsdatenbank ist unkritisch. Sofern das Datenbankenformat offen ist, sind die vorhandenen Informationen und ihre Strukturen bekannt und es kann eine zuverlässige Modellierung implementiert werden. Der Vorteil des vorgestellten Modells liegt in der einfachen Übertragbarkeit auf andere Netzwerke wie BACnet oder Konnex, da die heutigen drahtgebundenen Feldbussysteme sich stark ähneln. Dies wird durch vorgesehene Erweiterungspunkte und eine Beschränkung auf das für die weitere Analyse notwendige Wissen erreicht.

Für die automatische Modellierung des Verkehrsmodells aus Kapitel 6 wurden die gängigsten Verfahren der Nachrichtengenerierung in einem Netzwerk der Gebäudeautomation untersucht. Das Verkehrsmodell besteht aus einzelnen, generischen Gerätemodellen, deren Verkehrsverhalten anhand der im Netzwerkentwurf festgelegten Geräteparameter angepasst wird und die zusätzlich die Geräteinteraktion modellieren. Dadurch wird ein äußerst anpassungsfähiges und genaues Verkehrsmodell geschaffen, das fähig ist, sich an die unterschiedlichen Gerätevarianten und Kontexte anzupassen. Da in der Gebäudeautomation alle Netzwerke funktionsblockbasierte Geräte einsetzen und sich deren Arbeitsweisen ähneln, ist das Verkehrsmodell auch für andere Netzwerke anwendbar.

Die Leistungsbewertung in Kapitel 7 kombiniert eine analytische und simulative Analyse und ermöglicht so, die konträren Anforderungen nach Effizienz und Genauigkeit zu erfüllen. Hierfür wurde bei der Modellierung entsprechend Anforderung 4 auf eine unterschiedliche Granularität des Modells geachtet. Bei der Anpassung der Methode an ein anderes Netzwerk ist die Implementierung einer adäquaten Leistungsbewertung sicherlich am aufwendigsten.

Das Kapitel 8 erfüllt die Teilziele 2 bis 4 mit den in Abschnitt 8.5 diskutierten Maßnahmen. Grundlage zur Bewertung der Qualität von Netzwerkentwürfen ist die Dienstqualität (QoS). Die Methoden in Unterabschnitt 8.2.6 ermöglichen die automatische Verteilung solcher Anforderungen mit Hilfe der generischen Gerä-

temodelle. Auf diese Weise können effizient nach Anforderung 1 umfangreiche Qualitätsanforderungen an ein Netzwerk gestellt und Entwurfsfehler identifiziert werden.

Bei der Lösung der QoS-Fehler kann sich der Anwender auf unterschiedliche Weise beraten lassen. Die vorgestellten Fehlerbäume nutzen das Modellierungswissen, um die Ursachen von Verstößen gegen QoS-Anforderungen aufzudecken und dem Anwender vielseitige Lösungsansätze zu erschließen. Die unterschiedlichen Beratungstools wiederum führen das Wissen zusammen, um konkrete Parametrierungsvorschläge zu machen. Hierbei wird das globale Pareto-Optimierungsproblem gezielt zerlegt und durch praktikable Regeln mit nur wenigen Iterationen der Leistungsbewertung gelöst. Dadurch wird zwar nicht das globale Optimum der Qualität des Netzwerkentwurfs nach Gleichung (8.7) gefunden, aber eine Lösung im Rahmen der Anforderungen 1, 3 und 5.

Damit kann zusammengefasst werden, dass alle aufgestellten Teilziele im Rahmen der Anforderungen erfüllt wurden und folglich das Ziel einer Qualitätsverbesserung von Netzwerkentwürfen erreicht wurde. Der Zusatzaufwand für den Anwender ist sehr gering. Lediglich ein einziges Mal müssen für die verwendeten Gerätefamilien ein Gerätemodell, charakteristische Signaldynamiken und QoS-Anforderungen definiert werden. Mit den Methoden von GÜRTLER [75] reichen dazu Referenzmessungen aus. Danach kann der Anwender weiterhin sein gewohntes Entwurfstool benutzen, um den Entwurf auszuführen. Ohne sein Zutun wird sein Netzwerkentwurf automatisch aus der Tool-Datenbank gelesen, ein System- und Verkehrsmodell erstellt (Abbildung 5.2), die Leistungs- und Qualität bewertet (Abbildung 7.1), Fehler identifiziert und Lösungen entwickelt (Abbildung 8.1). Durch dieses automatisierte Control Network Performance Engineering kann ein Systemintegrator einen qualitativ hochwertigen Netzwerkentwurf in kurzer Zeit gezielt erstellen.

KAPITEL

11.

AUSBLICK

*Voraussagen soll man unbedingt vermeiden,
besonders solche über die Zukunft.*

Mark Twain

DIE automatische Modellierung der Gerätemodelle aus Messungen wurde schon mehrmals als wichtige Ergänzung angesprochen. Zielstellung ist es, die Gerätemodelle aus passiv aufgenommenen Protokollspuren automatisch und ohne steuernden oder korrigierenden Eingriff des Anwenders zu erstellen. Dabei geht es nicht nur darum, das Sendeverhalten in Verteilungsfunktionen zu protokollieren, sondern die aufgestellten detaillierten und abstrakten Gerätemodelle zu erstellen und zu parametrieren, bis hin zu der Beurteilung der Abtastprozesse. Trotz der zum Teil einfachen Gerätemodelle liegt hierin eine komplexe Aufgabe der automatischen Modellierung, die bisher nicht umfangreich erforscht wurde und mit der Arbeit von MARIO GÜRTLER [75] nicht abzuschließen ist.

Neben der automatischen Modellierung der differenzierten Gerätemodelle ist die Bereitstellung dieser Modelle für den Gebrauch eine wichtige Fragestellung. Hierfür bietet sich eine zentral organisierte Datenbank an, in der die Gerätemodelle zusammengetragen und ausgetauscht werden können. Damit sind vor allem organisatorische Fragen verbunden. Die in Abschnitt 6.6 beschriebene Zuordnung der Gerätemodelle zu den Geräten mittels Schlüsselwörtern und den Hersteller- und Gerätetypbezeichnungen bieten nur grundlegende Organisationsansätze, erfassen aber nicht verwandte Geräte oder eine Hierarchisierung der Schlüsselwörter. Die Fragestellung ähnelt dem Problem der funktionalen und kontextbezogenen Klassifikation von Geräten und Geräteprofilen, die auch im automatischen Entwurf eine Rolle spielt [22, 37, 114, 192].

Darüber hinaus ist die Fragestellung des Austauschs der Gerätemodelle zu klären. Wenn Anwender die Möglichkeit besitzen, auf einfache Weise aus Protokollspuren Gerätemodelle zu bilden, so ist der Umgang der Datenbank mit unterschiedlichen Modellvarianten aus vergleichbaren Kontexten zu erforschen. Hier könnten Lernverfahren und eine automatische Klassifikation interessante Fragestellungen bieten.

Von der systemeigenen Datenbank losgelöst, können die in Abschnitt 8.4 vorgestellten Diagnosedienste ergänzt werden. Unter anderem sind die Reorganisation des Netzwerks und die automatische Routerplatzierung herausfordernde Aufgaben. Hierzu bietet das Systemmodell der Anwendungsschicht einen sinnvollen Zugang zur Routerplatzierung, da die Konnektivität der Geräte a priori bekannt ist und es sinnvoll ist, Router dort zu platzieren, wo sie den meisten Verkehr trennen. Allerdings spielen bei der Platzierung von Routern die Raum- und Etagenplanung eine entscheidende Rolle, so dass die Integration von Gebäudemodellen [53, 92, 93, 113] vielversprechende Ansätze ermöglicht.

Darüber hinaus ist die automatische Parametrierung eine interessante Aufgabenstellung, die über die hier vorgestellte Zielstellung des CNPE hinausgeht. Denn es werden nicht mehr nur Fehler gefunden, beseitigt und im Lerneffekt vermieden, sondern dem Anwender wird die lästige Parametrierung vollständig abgenommen. Gerade im Zusammenhang mit Plug&Play [168] und dem automatischen Entwurf [22, 37, 114, 192] können solche Ansätze die Parametrierung aus der Konfigurierung verdrängen, wofür eine hohe Qualität der Entwürfe entscheidend ist.

Über die automatische Parametrierung des Netzwerks hinaus können auch die Anwendungen mit parametrisiert werden. In der Praxis ist ein Großteil der Regelkreise schlecht ausgelegt, da häufig Standardparameter verwendet werden. Die Mechanismen zur automatischen Anwendung von Gerätemodellen könnten dahingehend ausgeweitet werden, dass semantisch verbundene Gerätegruppen automatisch erkannt und parametrisiert werden. Da das Netzwerkverhalten wichtig bei der Auslegung der Regelungen [141, 175, 206, 228] sein kann, können Ergebnisse der Leistungsbewertung gut verwendet werden. Letztendlich erlauben solche Modelle außerdem die Beurteilung der Prozess- und Regelqualität in der Diagnose und die Anpassung der Netzwerkparameter an die Regelungen.

Zu guter Letzt können die hier in der Konfigurierung angewendeten QoS-Anforderungen und gewonnenen Diagnose- und Parametrierungskenntnisse ebenfalls im Betrieb genutzt werden, von einer Qualitätsüberwachung über eine Online-Diagnose bis hin zu einer QoS-Steuerung des Netzwerks.

APPENDIX

DANKSAGUNGSVERZEICHNIS

Da meine Arbeit einigen Verzeichnissen umfasst, die ich ohne die Unterstützung vieler anderer Menschen nicht hätte zusammentragen können, möchte ich dieses Verzeichnis diesen Menschen widmen.

Allen voran möchte ich Herrn Prof. Kabitzsch danken, dass er mit dem NetPlan-Projekt diese Arbeit erst ermöglicht und im Anschluss ihren Fortgang gesichert hat. Mit seiner fortwährenden Bereitschaft zu Diskussionen und kurzfristigem »Triggern«, hat er mit seinen fachlichen Anregungen und praktischen Fragestellungen viel zu der Abgrenzung und Erdung der Arbeit beigetragen. Auch sei ihm für das Vertrauen gedankt, das er damals in den Studenten gesetzt hat, der nur ein paar Fragen am Telefon klären wollte.

Zum Zweiten gilt mein Dank Herrn Prof. Buchholz. Trotz vieler anderer Verpflichtungen war er immer bereit, sich mit den mannigfaltigen Ideen eines Neulings zu beschäftigen. Mit seiner fachlichen Präzision ist er mir ein wichtiges Vorbild und ein guter und geduldiger Lehrer. Unter anderem lehrte ich von ihm, dass Formeln in jedem Sinne nicht kompliziert aussehen müssen.

Ferner möchte all jenen senioren Mitwirkenden am NetPlan-Projekt danken, die mit Witz und Charme das Projekt gestalteten und uns jungen Spunden den Freiraum und den Ansporn vermittelten, in der Welt der Wissenschaft Fuß zu fassen.

Den Jungspunden aus dem Projekt sei gedankt, dass sie tatkräftig die Basis für diese Arbeit geschaffen haben. Dem immerfort Jüngsten von ihnen, Herrn Mario Neugebauer, bin ich zum größten Dank verpflichtet. Er hat viele Aspekte der Arbeit mitkreiert, mitdiskutiert und miterschaffen. Seine ruhige Art war mir manches Mal eine wichtige Bremse und sein unbändiger Tatendrang ein wichtiger Antrieb.

Teile dieser Arbeit und meines Dankes gebühren den Studenten Alexander Fuchs, Mario Gürtler, Boris Kolman und Daniel Wendt. Sie verdeutlichten mir, kurz nach meiner eigenen Studentenzeit, wie zwiespältig die Aufgabe eines Betreuers ist.

Auch danke ich den Kollegen am Lehrstuhl für Technische Informationssysteme, dass sie den Regelkreis einer angenehmen Arbeitsatmosphäre immer stabil hielten. Insbesondere Frau Anja Brauny als wohlwollende Proportioniererin, Michael Knittel als hochangerechneter Integrator und Uwe Schumann als feinhumoriger

Differentiator.

Ich danke Claudia Kühn, Jens Naake und all jenen, die sich ungeachtet der störenden Formeln bemühten, die Fehler im Text der Arbeit zu finden. Ihnen sei gesagt: Wenn $p_F > 0$ die Wahrscheinlichkeit ist, dass ich einen Fehler mache und sie mit der Wahrscheinlichkeit $p_S > 0$ einen Fehler finden, so bleibt auch nach r Iterationen immer eine Restfehlerwahrscheinlichkeit $p_F[1 - p_S(1 - p_F)]^r$ bestehen.

Ich möchte all jenen Leuten danken, die immer wieder dezent nachgefragt haben, wie es um meine Arbeit steht. Es war für mich eine Quelle des Antriebs und an dieser Stelle sei ihnen geantwortet: Es steht gut.

Mein Dank gilt auch meinen Freunden dafür, dass sie es waren, wenn ich einer Ablenkung bedurfte und es geblieben sind, solange ich mehr Zeit mit dieser Arbeit verbracht habe als mit ihnen.

Ferner möchte ich dem Leser danken, dass er die Ruhe und Muße gefunden hat, bis hierher zu lesen. Insbesondere meinen Gutachtern möchte ich danken und ihnen zur Beruhigung erklären: Sie haben es fast geschafft.

Als Letztes möchte ich jenen danken, die immer wieder betonen, dass man ihnen nicht danken braucht — meiner Familie und meinen Eltern. Sie sind es mir wert, ihnen zu danken.

Jörn Plönnigs
Dresden, 31. Januar 2007

VERZEICHNIS ALLER ZIELE, ANFORDERUNGEN UND PRÄMISSEN

ZIEL (S. 21)

Zielstellung der Arbeit ist die Erhöhung der Qualität von Netzwerkentwürfen von Steuerungs- und Regelungsnetzwerken durch eine Entwurfsunterstützung zur Vermeidung, Identifikation und Behebung von Entwurfsfehlern mit Hilfe einer begleitenden, automatisierten Qualitäts- und Leistungsbewertung der Kommunikationsmedien.

TEILZIELE

TEILZIEL 1: MODELLIERUNG DES LEISTUNGSMODELLS (S. 28)

Es wird ein Konzept zur automatischen Modellierung von Leistungsmodellen aus Entwurfsdatenbasen für ausgewählte Netzwerke der Gebäudeautomation entwickelt.

TEILZIEL 2: BEWERTUNG DER QUALITÄT DES KOMMUNIKATIONSMEDIUMS (S. 29)

Die Qualität des Kommunikationsmediums wird anhand der in den Qualitätsmerkmalen 2 bis 5 festgelegten Kriterien bewertet.

TEILZIEL 3: IDENTIFIKATION VON ENTWURFSFEHLERN (S. 29)

Werden Anforderungen an die Qualität des Kommunikationsmediums nicht erfüllt, so werden diese als Entwurfsfehler identifiziert.

TEILZIEL 4: BERATUNG ZUR FEHLERBEHEBUNG (S. 29)

Werden Entwurfsfehler identifiziert, so werden die Ursachen analysiert und der Anwender wird zum Zwecke der Fehlerbehebung beraten.

ANFORDERUNGEN

ANFORDERUNG 1: EFFIZIENZ (S. 21)

Die Bewertung der Qualität von Netzwerkentwürfen und die Identifikation und Behebung von Entwurfsfehlern darf die Konfigurierung nicht wesentlich verlangsamen.

ANFORDERUNG 2: VERSTÄNDLICHKEIT (S. 22)

Die Bewertung der Qualität von Netzwerkentwürfen und die Identifikation und Behebung von Entwurfsfehlern sollte vom Entwerfer kein umfangreiches Zusatzwissen erfordern.

ANFORDERUNG 3: GENAUIGKEIT (S. 22)

Die resultierenden Qualitätsabschätzungen müssen in Bezug auf die Qualitätsmerkmale 2 bis 6 eine hinreichend genaue Aussage erlauben, um die wesentlichen Fehler korrekt zu identifizieren.

ANFORDERUNG 4: DIFFERENZIIERTHEIT (S. 22)

Die Methoden und Modelle sollten in ihrem Detaillierungsgrad an die vorhandenen Systeminformationen und die Anforderungen an die Ergebnisse anpassbar sein.

ANFORDERUNG 5: AUTOMATISIERUNG (S. 22)

Die Lösung muss weitestgehend automatisiert sein, so dass sie nur geringe Eingriffe des Anwenders benötigt.

ANFORDERUNG 6: INTEGRATION (S. 22)

Der Lösung muss sich in den vorhandenen Entwurfsprozess nahtlos integrieren.

QUALITÄTSMERKMALE

QUALITÄTSMERKMAL 1: KORREKTE FUNKTIONSIMPLEMENTATION (S. 13)

Während der Konfigurierung muss jede Funktion in der Form umgesetzt werden, dass die Geräte die geforderte Funktionalität in Abwesenheit von Kommunikationsfehlern korrekt erfüllen können.

QUALITÄTSMERKMAL 2: GARANTIERBARKEIT (S. 14)

Die Einhaltung geforderter Prozessparameter muss in Bezug auf das Kommunikationsmedium innerhalb vorgegebener Toleranzen garantiert und nachgewiesen werden.

QUALITÄTSMERKMAL 3: BETRIEBSSICHERHEIT (S. 14)

Ein sicherer und gefahrloser Betrieb muss in Bezug auf das Kommunikationsmedium im Rahmen von Qualitätsmerkmal 2 garantiert werden.

QUALITÄTSMERKMAL 4: ECHTZEITVERHALTEN (S. 14)

Eine rechtzeitige Reaktion des Systems bei Störungen muss in Bezug auf das Kommunikationsmedium im Rahmen von Qualitätsmerkmal 2 garantiert werden.

QUALITÄTSMERKMAL 5: VERFÜGBARKEIT (S. 14)

Die Verfügbarkeit des Systems soll in Bezug auf das Kommunikationsmedium im Rahmen von Qualitätsmerkmal 2 abgeschätzt werden.

QUALITÄTSMERKMAL 6: ANGEMESSENE KOSTEN (S. 14)

Das Kommunikationsmedium soll den technischen und ökonomischen Anforderungen nach angemessen dimensioniert werden, wobei die technischen Mindestanforderungen erfüllt sein müssen und eine Minimierung der Kosten angestrebt wird.

PRÄMISSEN

PRÄMISSE 1: KORREKTE FUNKTIONSIMPLEMENTATION (S. 17)

In der Arbeit wird Qualitätsmerkmal 1 vorausgesetzt. Dementsprechend ist davon auszugehen, dass die Funktionen in den Geräten korrekt implementiert sind und in der Konfigurierung eine jede Funktion so eingebunden wird, dass die Geräte

die geforderte Funktionalität in Abwesenheit von Kommunikationsfehlern korrekt erfüllen können.

PRÄMISSE 2: KEIN REALES NETZWERK VORHANDEN (S. 24)

Bei der Konfigurierung ist kein reales Netzwerk vorhanden oder aus anderen Gründen ist eine Messung zur direkten Leistungsbewertung nicht sinnvoll.

PRÄMISSE 3: ELEKTRONISCH LESBARER NETZWERKENTWURF (S. 28)

Es gibt eine elektronisch lesbare Repräsentation des Netzwerkentwurfs, die das zu untersuchende Netzwerk in seinen wesentlichen strukturellen und parametrischen Eigenschaften widerspiegelt.

PRÄMISSE 4: KABELGEBUNDEN (S. 44)

Das zu untersuchende Netzwerk nutzt kabelgebundene Übertragungsmedien, beziehungsweise setzt kurze Funk- oder Infrarotstrecken ein, die sich im Verhalten durch ein Kabel approximieren lassen.

PRÄMISSE 5: UNVERÄNDERLICHE KOMMUNIKATIONSBEZIEHUNGEN (S. 49)

Es gibt im untersuchten Netzwerk a priori bekannte unveränderliche, gerichtete Kommunikationsbeziehungen.

PRÄMISSE 6: UNVERÄNDERLICHKEIT DER NACHRICHTENGRÖSSE (S. 52)

Der Variablentyp, die Adressierung und die Übertragungsdienste der Kommunikationsbeziehungen aus Prämisse 5 sind im Betrieb unveränderlich und im Netzwerkentwurf bekannt.

PRÄMISSE 7: EINDEUTIGKEIT DER WEGE (S. 53)

Zwischen zwei Geräten kann eine Nachricht maximal einen eindeutigen Weg nehmen.

PRÄMISSE 8: FUNKTIONSBLOCKBASIERT (S. 58)

Der Verkehr wird vorwiegend von Funktionsblöcken und ihren Datenpunkten erzeugt, die in dieser Form im Modell modelliert sind.

PRÄMISSE 9: ENTKOPPLUNG DER BEARBEITUNG (S. 65)

Der Zusammenhang zwischen dem Sendeereignis eines Ausgangsdatenpunktes σ_{δ} und dem Ankunftsereignis eines Eingangsdatenpunktes ι_{δ} kann isoliert von internen Zuständen und anderer Eingängen betrachtet werden.

PRÄMISSE 10: STATISTISCHE UNABHÄNGIGKEIT (S. 66)

Auf der Anwendungsschicht sind die Zwischenankunftszeiten an den Datenpunkten und die Bearbeitungszeiten statistisch unabhängig und identisch verteilt oder liegen außerhalb für das Netzwerkverhalten relevanten Grenzen (bei LON etwa 100–200 ms).

PRÄMISSE 11: EINGESCHRÄNKTE ZYKLEN IM VERKEHRSMODELL (S. 67)

Existiert ein Zyklus im Systemmodell der Anwendungsschicht und dem Verkehrsmodell, so muss die durch die Übergangswahrscheinlichkeiten und Fehlerwahrscheinlichkeiten von Bindings im Zyklus abgehende Rate an Nachrichten größer sein, als die von Quellprozessen eingetragene Ankunftsrate.

PRÄMISSE 12: BEARBEITUNGSZEITEN (S. 70)

Die Bearbeitungszeiten der einzelnen Schichten $\tau_{*,\delta}$ eines Gerätes sind, ausgenommen vom Medienzugriff, in Erwartungswert und Varianz deutlich kleiner als die

Zwischenankunftszeiten aller ankommenden Nachrichten, womit gilt $\mathbb{E}[\tau_{*,\delta}] \ll \mathbb{E}[T_{Z,\delta}] \wedge \text{Var}[\tau_{*,\delta}] \ll \text{Var}[T_{Z,\delta}]$.

PRÄMISSE 13: ZYKLUSZEIT ERFÜLLT ABTASTTHEOREM (S. 75)

Die Zykluszeit ist ausreichend klein, um ein rekonstruierbares Abbild des Signals zu liefern. Dies bedeutet bei kontinuierlichen Signalen, dass die Zykluszeit das Abtasttheorem erfüllt [184, 204] und bei Zustandssignalen, dass die Zyklusfrequenz mehr als doppelt so groß ist wie die maximale Änderungsfrequenz der Zustände.

PRÄMISSE 14: DÄMPFUNG DER UMGEBUNG (S. 89)

Die Einflüsse der Totzeiten und Dämpfungseigenschaften der Umgebungsprozesse auf die Ankunftsrate sind bedeutend größer, als die Einflüsse des Netzwerkverhaltens.

PRÄMISSE 15: LEISTUNGSKENNGRÖSSEN (S. 111)

Die Leistungsbewertung muss die in Abschnitt 7.2 definierten Leistungskenngrößen in ausreichender Genauigkeit nach Anforderung 3 bereitstellen.

PRÄMISSE 16: QUANTIFIZIERBARKEIT DER QUALITÄTSMERKMALE (S. 123)

Die geforderten Qualitätsmerkmale werden vollständig als Anforderungen an die Dienstqualität für einzelne Verbindungen spezifiziert.

ABBILDUNGSVERZEICHNIS

2.1. Funktionen der Gebäudeautomation.	8
2.2. Entwurfsprozess von Systemen der Gebäudeautomation.	12
3.1. Methoden zur Leistungsbewertung.	24
3.2. Integration des Control Network Performance Engineering in die Konfiguration.	29
4.1. Strukturmodell der Gebäudeautomation.	32
4.2. Vergleich verschiedener Protokolle mit dem OSI-Referenzmodell.	34
5.1. Schematische Aufteilung des Systemmodells in seine Teilmodelle.	40
5.2. Detaillierte Struktur der automatischen Modellierung.	41
5.3. Installationsplan des verwendeten Beispiels.	43
5.4. Topologie am Beispiel des Einzelraums.	44
5.5. UML-Darstellung des Systemmodells der Bitübertragungsschicht.	45
5.6. Adressierung am Beispiel des Einzelraums.	47
5.7. UML-Darstellung des Systemmodells der Vermittlungsschicht.	47
5.8. Systemmodell der Anwendungsschicht am Beispiel des Einzelraums.	49
5.9. UML-Darstellung des Systemmodells der Anwendungsschicht.	49
5.10. Beispiel einer Nachrichtenklasse.	56
5.11. UML-Modell des zusammengeführten Systemmodells.	57
6.1. Gerätemodell.	65
6.2. Zeitverhalten bei der Gerätekommunikation.	69
6.3. Zeitverhalten eines Gerätes.	70
6.4. Entwicklung Zwischenankunftszeiten zweier gekoppelter Ausgänge.	74
6.5. Klassifikation der Ankunftsprozesse.	76
6.6. Histogramme der Zwischenankunftszeit bei zyklischer Nachrichtengenerierung mit wiederholtem Timer.	77
6.7. Histogramme der Zwischenankunftszeit bei zyklischer Nachrichtengenerierung mit einmaligem Timer.	78
6.8. Signaldigitalisierung und Send-On-Delta an einem Beispiel.	81
6.9. Histogramme der Zwischenankunftszeiten für eine Temperaturspur.	84
6.10. Histogramme der Zwischenankunftszeiten der Beleuchtungsstärke.	85
6.11. Histogramme der Anstiege verschiedener Wettergrößen	87
6.12. Jahresverläufe der Außentemperatur und Beleuchtungsstärke.	88
6.13. Zwischenankunftszeiten vom 29.07.84 bis zum 31.07.84	89

6.14. Korrelogramme der Zwischenankunftszeiten.	90
6.15. Jahresverläufe der Raumtemperatur und Raumbeleuchtungsstärke.	91
6.16. Histogramme der Grenzüberschreitung am Signal der Temperatur.	92
6.17. Histogramme der Grenzüberschreitung am Signal der Beleuchtungsstärke.	93
6.18. UML-Darstellung des Gerätemodells.	99
6.19. Zwischenankunftszeiten einer Messung am Musterraum.	101
6.20. Verkehrsstatistiken ausgewählter Geräte aus dem Musterraum.	102
7.1. Detaillierte Struktur der Leistungsbewertung.	110
8.1. Detaillierte Struktur der Fehleranalyse und Nutzerberatung.	120
8.2. Typische Verzögerungszeiten unterschiedlicher Übertragungsdienste.	125
8.3. Markov-Kette der Wahrscheinlichkeit eines (m, k) -Fehlers.	126
8.4. Fehlerbaum der Kanalauslastung (Ergänzt in Abb. 8.5).	133
8.5. Fehlerbaum des Protokolloverheads ergänzend zu Abb. 8.4.	135
8.6. Fehlerbaum für die Verzögerungszeit.	136
8.7. Fehlerbaum für die Übertragungsfehler.	139
8.8. Prototyp der Diagnoseansicht des NetPlan-Tools aus [124].	140
8.9. Entscheidungsbaum zur Auswahl des richtigen Diagnosewerkzeuges.	153
9.1. Vergleich der einzelnen Analyseschritte.	158

TABELLENVERZEICHNIS

I.	Vergleich verschiedener Branchen der Automatisierungstechnik und Bürokommunikation.	11
II.	Bewertung verschiedener Branchen in der Eignung für eine genaue Dimensionierung und Qualitätsbewertung.	18
III.	Zusammensetzung der Nachrichtengröße in LON.	51
IV.	Nachrichtengröße in LON für das Beispiel.	52
V.	Vergleich der Eignung des Systemmodells für LON, Konnex und BACnet.	60
VI.	Verschiedene Messergebnisse zur Verzögerungszeit eines Neuron-Chips.	71
VII.	Parameter der Bearbeitungszeiten im Neuron-Chip.	72
VIII.	Charakteristika des Anstiegs unterschiedlicher physikalischer Größen.	86
IX.	Vergleich der Abschätzung der Zwischenankunftszeit an der Simulation.	97
X.	Parameter der Geräte aus der Messung.	103
XI.	Vergleich der Abschätzung der Zwischenankunftszeit bei Send-On-Delta zur Messung.	104
XII.	Vergleich der Abschätzung der Ankunftsrate zweier λ -Filter zur Messung.	105
XIII.	Ankunftsrate für das Beispiel.	108
XIV.	Ergebnisse der Leistungsbewertung für das Beispiel.	117
XV.	QoS-Charakteristika nach ISO/IEC 13236 [101].	121
XVI.	Qualität der QoS-Anforderungen im Beispiel.	128
XVII.	Empfohlene Übertragungsdienste nach den geforderten QoS-Charakteristika.	142

FORMELVERZEICHNIS

$\ \mathcal{D}\ $	Mächtigkeit der Menge \mathcal{D} (Anzahl der Elemente), S. 40	$\vec{\lambda}_{\mathcal{J}_\delta}$	Vektor der Erwartungswerte der Ankunftsrate aller Eingangsdatenpunkte \mathcal{J}_δ , S. 68
$\mathbf{A}_{\mathcal{J} \cup \mathcal{D}}$	Adjazenzmatrix aller Datenpunkte $\mathcal{J} \cup \mathcal{D}$, S. 149	$\bar{\lambda}_{\mathcal{I};p}$	Mittlere Ankunftsrate die am Port p ankommt, S. 114
\mathfrak{B}	Menge aller Bindings in einem Netzwerk, S. 49	$\vec{\lambda}_\delta$	Vektor der Erwartungswerte der Ankunftsrate aller Ausgangsdatenpunkte \mathcal{D}_δ eines Gerätes δ , S. 68
b_δ	Buffergröße des Geräts δ , S. 110	$\bar{\lambda}_{\mathcal{O};p}$	Mittlere Ankunftsrate die den Port p verlässt, S. 114
$b_{o,i}$	Binding zwischen o und i , S. 67	$\lambda_{p\text{phy}}$	Ankunftsrate auf Bitübertragungsschicht, S. 57
\mathcal{C}	Menge aller Kanäle in einem Netzwerk, S. 44	$\vec{\lambda}_{\mathcal{Q};\mathcal{D}_\delta}$	Vektor der Erwartungswerte der Ankunftsrate der Quellprozesse aller Ausgangsdatenpunkte eines Gerätes δ , S. 68
$C_{10;\delta}$	Prozessortakt zum Normtakt von 10 MHz, S. 71	$\vec{\lambda}_{\mathcal{Q};\mathcal{J} \cup \mathcal{D}}$	Vektor aller Quellprozesse aller Datenpunkte $\mathcal{J} \cup \mathcal{D}$ im Netzwerk, S. 149
C_δ	Prozessortakt eines Geräts, S. 71	$\bar{\lambda}_w$	Mittlere Ankunftsrate aller Nachrichtenklassen, die das Wegelement w passieren, S. 114
\mathcal{D}	Menge aller Geräte in einem Netzwerk, S. 4044	\mathfrak{M}_b	Menge der Nachrichtenklassen des Bindings $b \in \mathfrak{B}$, S. 54
$\delta \in \mathcal{D}$	Ein beliebiges Gerät aus der Menge aller Geräte in einem Netzwerk, S. 40	μ_ϵ	Bedienrate des Element ϵ , S. 112
δ_o	Delta-Parameter beim Send-On-Delta, S. 81	\mathfrak{P}	Menge aller Ports in einem Netzwerk, S. 45
d_o	Anstiegsbetrag $ f'_o $ des kontinuierlichen Signals f_o , S. 83	\mathbf{P}^+	Matrix der Erfolgswahrscheinlichkeit der Bindings im Netzwerk von \mathcal{D} nach \mathcal{J} , S. 149
\bar{e}	Mittlere Extremwertdichte eines Signals, S. 82	$p_{b_{o,i}}^-$	Fehlerwahrscheinlichkeit für den Verlust der Information beim Binding $b_{o,i}$, S. 116
G_{ξ^x}	QoS-Gewicht der QoS-Anforderung ξ^x , S. 131	f_C	Dichtefunktion des Zufallsprozesses C , S. 66
\mathcal{J}	Menge der Eingangsdatenpunkte (Empfänger), S. 48	p_i	Einer der Zielports eines Binding zwischen o und $i \in \mathcal{O}$, S. 113
\mathcal{D}	Menge der Ausgangsdatenpunkte (Sender), S. 48	p_o	Der Ursprungsport eines Binding mit dem Ursprungsdatenpunkt o , S. 113
K_b	Gewicht der freien Kapazitäten in den QoS-Anforderungen des Bindings b , S. 150		
k_i	Index der Zwischenankunftszeiten des Eingangsdatenpunktes i , S. 64		
k_o	Index der Zwischenankunftszeiten des Ausgangsdatenpunktes o , S. 64		
λ_{app}	Ankunftsrate auf Anwendungsschicht, S. 57		
$\lambda_{b_{o,i}}^*$	Ankunftsrate am Eingang i nach der verlustbehafteten Übertragung über das Binding $b_{o,i}$, S. 67		

$p_{I;w}^+$	Wahrscheinlichkeit das Wegelement w erfolgreich zu erreichen, S. 113	$\Theta_{\Pi;b}$	Relativer Datendurchsatz, den alle Nachrichtenklassen eines Bindings im Netzwerk pro Nachricht erzeugen, S. 143
$p_{O;w}^+$	Wahrscheinlichkeit das Wegelement w erfolgreich zu verlassen, S. 113	$\Theta_{\Sigma;b}$	Akkumulierter mittlerer Datendurchsatz, den alle Nachrichtenklassen eines Bindings im Netzwerk erzeugen, S. 143
p_{ϵ}^+	Erfolgswahrscheinlichkeit des Elements ϵ , S. 112	t_{k_i}	Sendezeitpunkt am Eingangsdatenpunkt i , S. 64
p_{ϵ}^-	Fehlerwahrscheinlichkeit des Elements ϵ , S. 112	t_{k_o}	Sendezeitpunkt am Ausgangsdatenpunkt o , S. 64
p_{ξ^x}	Wahrscheinlichkeit eines QoS-Defekts, S. 126	$T_{L;o}$	Parameter der Min-Send-Time beim Send-On-Delta, S. 7692
\bar{Q}	Qualitätsmaß eines Netzwerkentwurfs (Mittleres Qualitätsmaß aller Anforderungen), S. 127	$\mathfrak{T}_{Q;o}$	Menge der Sendezeitpunkte des Quellprozesses am Ausgangsdatenpunkt o , S. 65
Q_{\min}	Qualitätsmaß eines Netzwerkentwurfs (Minimales Qualitätsmaß aller Anforderungen), S. 127	$T_{Q;o}^*$	Ein beliebiger Quellprozess der ereignisorientierten Nachrichtengenerierung ohne Min- und Max-Send-Time, S. 92
Q_{ξ^x}	Qualitätsmaß einer QoS-Anforderung ξ^x , S. 127	$T_{RT;b}$	Repeat-Interval-Timer eines Bindings, der den Abstand wiederholter Nachrichten vorgibt, S. 136
\mathfrak{R}	Menge aller AbstractRouter in einem Netzwerk, S. 44	$T_{TT;b}$	Transmission-Timer eines Bindings nach dem, bei ausbleiben der Bestätigung, eine Erinnerung generiert wird, S. 137
r_b^+	Anzahl der Nachrichten beim wiederholten Übertragungsdienst, S. 114	$T_{U;o}$	Parameter der Max-Send-Time beim Send-On-Delta, S. 7692
r_b^-	Failure-Retrieys eines Bindings: die maximale Anzahl an Wiederholungen, S. 114	$T_{Z;b_o,i}^*$	Zwischenankunftszeit am Eingang i nach der verlustbehafteten Übertragung über das Binding $b_{o,i}$, S. 67
ρ_{ϵ}	Auslastung des Elements ϵ , S. 112	$T_{Z;i,o}^*$	Zwischenankunftszeit am Ausgang o , die auf einige Eingangsereignisse an i zurückzuführen sind, S. 66
$r_{\xi}(j)$	Koeffizient der Autokorrelation eines QoS-Charakteristikum ξ , S. 126	$T_{Z;k_i}$	Zwischenankunftszeiten des Eingangsdatenpunktes i zweier aufeinander folgender Ereignisse, S. 64
R_{ξ^x}	Risikoklasse der QoS-Anforderung ξ^x , S. 131	$T_{Z;k_o}$	Zwischenankunftszeiten des Ausgangsdatenpunktes o zweier aufeinander folgender Ereignisse, S. 64
s	Sicherheitsfaktor bei der Auslegung, S. 144	$\mathfrak{T}_{Z;o}$	Menge der Sendezeitpunkte des Ausgangsdatenpunktes o , S. 64
S_{phy}	Nachrichtengröße auf dem Kanal in Bits, S. 51	u_m	Ankunftsdatenmodifier, S. 113
$\bar{S}_{\text{phy};w}$	Mittlere Nachrichtengröße aller Nachrichtenklassen, die das Wegelement w passieren, S. 114	\mathbf{V}	Matrix der Übertragungswahrscheinlichkeiten im Netzwerk von \mathcal{J} nach \mathcal{D} , S. 149
$\tau_{B;i,o}$	Bearbeitungszeit zwischen i und o , S. 64	$\mathbf{V}_{\mathcal{D}}$	Matrix der Übertragungswahrscheinlichkeiten eines Gerätes \mathcal{D} , S. 68
$\tau_{i,o}(k_i)$	Verzögerungszeit zwischen i und o , S. 64		
τ_S	Sendezeit einer Nachricht, S. 138		
τ_W	Wartezeit einer Nachricht, S. 138		
$\tau_{b_{o,i}}$	Verzögerungszeit des Bindings $b_{o,i}$, S. 67		
$\mathfrak{T}_{F;i,o}$	Menge der Sendezeitpunkte des Filterprozesses am Ausgangsdatenpunkt o , S. 65		
Θ	Datendurchsatz (Last), S. 111		
ϑ_{ϵ}	Durchsatz (Abgangsrate), S. 111		

$v_{i,o}$	Übertragungswahrscheinlichkeit zwischen i und o , S. 66		Verteilung der Zwischenankunftszeit, S. 94
w_δ	Anteil der ursprünglichen Zwischenankunftszeit ohne Min- / Max-Send-Time, S. 94	w_U	Anteil der Max-Send-Time an der Verteilung der Zwischenankunftszeit, S. 94
$\mathbf{w}_{p_o \rightarrow p_i}$	Weg der die Ports p_o und p_i zweier Geräte mit den Datenpunkten o und i miteinander verbindet, S. 55	ξ	Ein beliebiges QoS-Charakteristikum, S. 123
w_L	Anteil der Min-Send-Time an der	\mathbf{Z}	Sensibilitätsmatrix, S. 149
		\bar{z}	Sensibilitätsvektor, S. 150

ABKÜRZUNGSVERZEICHNIS

Ack	Bestätigung (en. Acknowledgment), S. 53	EIB	Europäischer Installations-Bus (en. European Installation Bus), S. 33
AckD	Nachricht mit der Aufforderung zur Bestätigung (en. Acknowledged-Request), S. 53	EN	Europäische Norm, S. 179
APDU	en. Application PDU (Anwendungsschicht), S. 51	ETS	en. EIB Tool Software, S. 41
AR	Autoregressiv, S. 125	FIFO	First-In-First-Out, S. 70
ASHRAE	en. American Society of Heating, Refrigerating and Air-Conditioning Engineers, S. 33	GAN	en. Global Area Network (Globale Netzwerke), S. 10
AuthPDU	en. Authentication PDU (Sitzungsschicht), S. 51	HIL	en. hardware-in-the-loop, S. 23
BACnet	en. Building Automation and Control Network, S. 33	ISO	Internationale Organisation für Normung (en. International Organization for Standardization), S. 179
CA	Kollisionsvermeidung (en. Collision Avoidance), S. 33	ITU-T	en. International Telecommunication Union - Telecommunication Standardization Sector, S. 179
CD	Kollisionserkennung (en. Collision Detection), S. 33	KNX	Konnex, S. 33
CDF	Kumulative Verteilungsfunktion (en. Cumulative Distribution Function), S. 83	LAN	Lokales Netzwerk (en. Local Area Network), S. 10
CEN	Europäisches Komitee für Normung (frz. Comité Européen de Normalisation; en. European Committee for Standardization), S. 179	LNS	en. LonWorks Network Operating System, S. 41
Chall	Aufforderung zur Authentifizierung (en. Challenge), S. 53	LON	en. Local Operating Network, S. 33
CNPE	Control Network Performance Engineering, S. 30	LPDU	en. Link PDU (Sicherheitsschicht), S. 51
COM	en. Component Object Model, S. 41	MAN	en. Metropolitan Area Network (Großstadtnetzwerk), S. 10
CPU	Hauptprozessor (en. Central Processing Unit), S. 68	MPDU	en. MAC PDU (Sicherheitsschicht), S. 51
CSMA	en. Carrier Sense Media Access, S. 34	NPDU	en. Network PDU (Vermittlungsschicht), S. 51
DCOM	en. Distributed Component Object Model, S. 41	OSI	Kommunikation offener Systeme (en. Open System Interconnection), S. 31
DIN	Deutsches Institut für Normung e.V., S. 179	PDF	Dichtefunktion (en. Probability Distribution Function), S. 66
EHS	en. European Home System, S. 33	PDU	en. Protocol Data Unit, S. 51
		Rem	Erinnerung nach Ausbleiben einer Antwort (en. Reminder), S. 53
		RemMsg	Erinnerung inklusive Nachricht (en. Reminder-Message), S. 53

ABKÜRZUNGSVERZEICHNIS

Req	Aufforderung zur Antwort (en. Request), S. 53	TPDU	en. Transport PDU (Transportschicht), S. 51
Res	Antwort auf ein Request (en. Response), S. 53	UML	en. Unified Modelling Language, S. 40
SNVT	Standard Netzwerk Variablen Typ (en. Standard Network Variable Type), S. 50	VDE	Verband der Elektrotechnik, Elektronik und Informationstechnik e.V., S. 179
SPDU	en. Session PDU (Sitzungsschicht), S. 51	VDI	Verein Deutscher Ingenieure e.V., S. 179
SPS	Speicherprogrammierbare Steuerung, S. 62	WAN	en. Wide Area Network (Weitverkehrsnetzwerk), S. 10
TGA	Technische Gebäudeausrüstung, S. 8		

LITERATURVERZEICHNIS

- [1] ADAS, Abdelnaser: Traffic Models in Broadband Networks. In: *IEEE Communications Magazine* 35 (1997), Juli, Nr. 7, S. 82–89
- [2] ALEX 3. http://www.spega.de/de/html/products/p_alex3.html. Version: 2006
- [3] ASSAF, Tariq; DUGAN, Joanne B.: Diagnostic expert systems from dynamic fault trees. In: *Reliability and Maintainability, Annual Symposium*, 2004, S. 444–450
- [4] BALL, P.; LOVE, D.: The Key to Object-Oriented Simulation: Separating the User and the Developer. In: *Winter Simulation Conference*. San Francisco, CA, USA, Januar 1995, S. 768–774
- [5] BAUMGARTH, S.; BOLLIN, E.; BÜCHEL, M.; FROMM, B.; KARBACH, A.; OTTO, D.; PAESCHKE, H.; RITZENHOFF, P.; SCHERNUS, G.-P.; SOKOLLIK, F.; TIERSCH, F.; TREUSCH, W.: *Digitale Gebäudeautomation*. London; Berlin; Heidelberg: Springer, 2004. – ISBN 3–540–00469–6
- [6] BERNAT, Guillem; BURNS, Alan; LLAMOSI, Albert: Weakly Hard Real-Time Systems. In: *IEEE Transactions on Computers* 50 (2001), Nr. 4, S. 308–321. – ISSN 0018–9340
- [7] BIELER, P. (Hrsg.): *Veranstaltungunterlagen Gebäudeautomation*. Haus der Technik e. V., 27. September 1995 (E-10-739-103-5)
- [8] BLAKE, Ian F.; LINDSEY, William C.: Level-crossing problems for random processes. In: *IEEE Transactions on Information Theory* 19 (1973), Mai, Nr. 3, S. 295–315. – ISSN 0018–9448
- [9] BOLCH, Gunter: *Leistungsbewertung von Rechensystemen mittels analytischer Warteschlangenmodelle*. Stuttgart: Teubner, 1989. – ISBN 3–519–02279–6
- [10] BOX, George E. P.; JENKINS, Gwilym M.; REINSEL, Gregory C.: *Time series analysis: Forecasting and Control*. 3. Englewood Cliffs, NJ: Prentice Hall, 1994. – ISBN 0–13–060774–6
- [11] BRASSARD, Gilles; BRATLEY, Paul: *Algorithmic - Theory and Practice*. Englewood Cliffs, NJ: Prentice Hall, 1988. – ISBN 0–13–023243–2
- [12] BRETTHAUER, G.; DIETZE, S.; HÄFELE, K.-J.; ISELE, J.; JÄKEL, J.: *Nachhaltiges Planen, Bauen und Wohnen im Informationszeitalter / Forschungszentrum Karlsruhe GmbH*. 2001. – Wissenschaftlicher Bericht
- [13] BROSTER, Ian: *Flexibility in Dependable Real-time Communication*, University of York, Dissertation, August 2003
- [14] BROY, Manfred; HEGERING, Heinz-Gerd; PICOT, Arnold; BUTTERMANN, Anne; GARSCHHAMMER, Markus; VOGEL, Sascha; GEHRING, Frank; KELTER, Harald; ULLMANN, Markus: *Integrierte Gebäudesysteme - Technologien, Sicherheit und Märkte / Bundesamt für Sicherheit in der Informationstechnik*. 2002. – Studie. – ISBN 3–922746–35–7
- [15] BUCHHOLZ, Peter; PANCHENKO, Andry: The Influence of Priorities on the Performance of CSMA-Based Access Schemes for Control Nets. In: *WFCS 2006 - 6th IEEE International Workshop on Factory Communication Systems*. Torino, Italy, 28.–30. Juni 2006, S. 333–342

- [16] BUCHHOLZ, Peter; PLÖNNIGS, Jörn: Analytical analysis of access-schemes of the CSMA-type. In: *WFCS 2004 - 5th IEEE International Workshop on Factory Communication Systems*. Vienna, Austria, 22.–24. September 2004, S. 127–136
- [17] BUCKLEW, James A.: *Introduction to Rare Event Simulation*. London; Berlin; Heidelberg; Springer, 2004 (Springer Series in Statistics). – ISBN 0–387–20078–9
- [18] CAMPBELL, Andrew; COULSON, Geoff; HUTCHISON, David: A quality of service architecture. In: *SIGCOMM Computer Communication Review* 24 (1994), Nr. 2, S. 6–27. – ISSN 0146–4833
- [19] CAPEK, J.; HANZALEK, Z.: STPN model of physical and MAC layer of LonWorks. In: *CSD'2000 - IFAC Conference on Control Systems Design*. Bratislava, Slovak Republic, 18.–20. Juni 2000, S. 335–340
- [20] CASTELPIETRA, P.; SONG, YQ.; SIMONOT-LION, F.; ATTIA, M.: Analysis and simulation methods for performance evaluation of a multiple networked embedded architecture. In: *IEEE Transactions on Industrial Electronics* 49 (2002), Nr. 6, S. 1251–1264. – ISSN 0278–0046
- [21] CENELEC - European Committee for Electrotechnical Standardization: *SmartHouse Code of Practice*. 2005. – Workshop Agreement
- [22] CHARATSIS, K.; KALOGERAS, A.P.; GEORGIOUDAKIS, M.; GIALELIS, J.; PAPADOPOULOS, G.: Home/Building Automation Environment Architecture Enabling Interoperability, Flexibility and Reusability. In: *Industrial Symposium on Industrial Electronics*. Montreal, Canada, 2005, S. 1441–1446
- [23] CHRISTEN, Markus; FRIEDRICHS, Kay; JUND, Eduard; KRANZ, Hans R.; MEYER, Heinz; MÜLLER, Rony; SCHEITLIN, Hans; STAUB, Richard; STÖRK, Harald; TRES, Diego; WILLERS, Jobst; STAUB, Reichard (Hrsg.); TRÖSCH, Bernhard (Hrsg.): *GNI-Handbuch der Raumautomation - Gebäudetechnik mit Standard-systemen*. Aarau, Switzerland: AZ Fachberlag, VDE-Verlag, 1999. – ISBN 3–905214–33–4
- [24] Cisco Systems: *Quality of Service Networking*. http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/qos.htm#xtocid3. Version: 2006
- [25] CLARKE, Joseph A.; 2 (Hrsg.): *Energy Simulation in Building Design*. Butterworth-Heinemann, 2001. – ISBN 0–7506–5082–6
- [26] COLE, Robert G.; RAMASWAMY, Ravi: *Wide Area Data Network Performance Engineering*. Artech House Books, 1999. – ISBN 0–89006–569–1
- [27] COOLING, Jim E.: *Software Engineering for Real-Time Systems*. Addison Wesley, 2003. – ISBN 0–201–59620–2
- [28] COOMBS, Clyde F.: *Communication network test and measurement handbook*. 1. McGraw-Hill, 1998 (Digital communications). – ISBN 0–07–012617
- [29] CORMEN, Thomas H.; LEISERSON, Charles E.; RIVEST, Ronald L.; STEIN, Clifford: *Introduction to Algorithms*. 2. Cambridge, Mass.: MIT Press, 2001. – ISBN 0–262–03293–7
- [30] CRAWLEY, Drury B.; LAWRIE, Linda K.; WINKELMANN, Frederick C.; BUHL, W.F.; HUANG, Y. J.; PEDERSEN, Curtis O.; STRAND, Richard K.; LIESEN, Richard J.; FISHER, Daniel E.; WITTE, Michael J.; GLAZER, Jason: EnergyPlus: creating a new-generation building energy simulation program. In: *Energy and Buildings* 33 (2001), Nr. 4, S. 443–457. – ISSN 0378–7788
- [31] DANIELS, Klaus: *Advanced Building systems: A Technical Guide for Architects and Engineers*. Basel: Birkhäuser, 2003. – ISBN 3–7643–6723–7
- [32] DAVIDSSON, Paul; BOMAN, Magnus: Saving Energy and Providing Value Added Services in Intelligent Buildings: A MAS Approach. In: *2nd international Symposium on Agent Systems and Applications and 4th international Symposium on Mobile Agents*. Zurich, Switzerland: Springer, 2000 (Lecture Notes in Computer Science 1882). – ISSN 0302–9743, S. 166–177

- [33] DEMENTJEV, Alexander; KABITZSCH, Klaus: A Consulting Module in Room Automation. In: *IFAC Symposium on Telematics Applications In Automation and Robotics*. Espoo, Finland, 2004, S. 37–42
- [34] DEVROYE, Luc: *Non-Uniform Random Variate Generation*. London; Berlin; Heidelberg; Springer, 1986. – ISBN 3–540–40652–5
- [35] DEWSBURY, Guy: *CATCH - A Compendium of Assistive Technology Checklists for the Home: For Professionals to determine Technology Requirements for the home*. <http://www.smartthinking.ukideas.com/CATCH.html>. Version: 2006
- [36] DEWSBURY, Guy; TAYLOR, Bruce; EDGE, Martin: A Dependability Model For Domestic Systems. In: *Safecom 2003 - 22nd International conference on Computer Safety, Reliability and Security*. Edinburgh, UK: Springer, 23–26. September 2003 (Lecture notes in computer science 2788). – ISBN 3–540–20126–2
- [37] DIBOWSKI, Henrik; OEZLUEK, Cemal; PLOENNIGS, Joern; KABITZSCH, Klaus: Realizing the Automated Design of Building Automation Systems. In: *INDIN 2006 - 4th International IEEE Conference on Industrial Informatics*. Singapore, 16.–18. August 2006, S. 251–256
- [38] DIETRICH, Dietmar; FISCHER, Peter: *LonWorks-Planerhandbuch*. Berlin: VDE-Verlag, 2001. – ISBN 3–8007–2599–1
- [39] DIETRICH, Dietmar (Hrsg.); KASTNER, Wolfgang (Hrsg.); SAUTER, Thilo (Hrsg.): *EIB - Gebäudebussystem*. Heidelberg: Hüthig Verlag, 2000
- [40] DIETRICH, Dietmar; LOY, Dietmar; SCHWEINZER, Hans-Jörg: *Open Control Networks*. Boston, Dordrecht, London: Kluwer Academic Publishers Boston, 2001. – ISBN 0–7923–7406–1
- [41] *DIN 18599 - Energetische Bewertung von Gebäuden - Berechnung des Nutz-, End- und Primärenergiebedarfs für Heizung, Kühlung, Lüftung, Trinkwasser und Beleuchtung*. 2005
- [42] *DIN 25424 - Fehlerbaumanalyse*. 1990
- [43] *DIN 66271 - Software-Fehler und ihre Beurteilung durch Lieferanten und Kunden*. 1995
- [44] DONATH, Ulrich; ALTMANN, Sven; SCHWARZ, Peter: Simulation von Feldbusprotokollen in Verbindung mit Gebäudemodellen. In: [110]
- [45] DUATO, José; YALAMANHILI, Sudhakar; NI, Lionel: *Interconnection networks: An engineering Approach*. Los Alamitos, Calif.: IEEE Computer Society, 1997. – ISBN 0–8186–7800–3
- [46] EBERT, Jean-Pierre; WILLIG, Andreas; WOLISZ, Adam (Hrsg.): A Gilbert-Elliot Bit Error Model and the Efficient Use in Packet Level Simulation / Technical University Berlin - Telecommunication Networks Group. 1999. – Technical Report
- [47] ECHELON CORPORATION: *LonMaker Integration Tool*. <http://www.echelon.com/products/integration/lonmaker/>. Version: 2005
- [48] ECHELON CORPORATION: *LNS Network Operating System*. <http://www.echelon.com/lns>. Version: 2006
- [49] ECHELON CORPORATION (Hrsg.): *Neuron C Programmer's Guide*. v7. Echelon Corporation, 2006
- [50] EHRGOTT, Matthias: *Multicriteria Optimization*. London; Berlin; Heidelberg: Springer, 2000 (Lecture Notes in Economics and Mathematical Systems 491). – ISBN 3–540–67869–7
- [51] *EIA/CEA-709.1-B-2002: Control Network Protocol Specification*. 2002
- [52] ELLIOT, E. O.: Estimates of error rates for codes on burstnoise channels. In: *Bell Systems Technical Journal* 42 (1963), September, S. 1977–1997
- [53] EMMRICH, Frank; PENETTI, Enrico; LOST, Thomas; NITZ, Silvia; PLESSOW, Matthias; VIGERSKE, Werner: Projekt Network-IFC - Industrial Foundation Classes (IFC) für Netzwerke in Gebäuden / IIEF GmbH and GFal e.V. and BAUAKADEMIE GmbH. 2001. – Abschlußbericht
- [54] *EN 14908 - Open Data Communication in Building Automation, Controls and Building Management*. 2005

- [55] EN 50090 - Home and Building Electronic System. 2004. – Under Approval
- [56] EN 50090 - Home and Building Electronic Systems (HBES) - Part 3-2: Aspects of application - User process. 2004. – Under Approval
- [57] EN 61499-1 - Function blocks - Part 1: Architecture. 2006
- [58] EN 61804 - Function Blocks (FB) for process control - Part 2: Specification of FB concept and Electronic Device Description Language (EDDL). 2004
- [59] ETS 3. <http://www.konnex.org/tools/ets/>. Version: 2006
- [60] FAHMY, Hany I.; DOULIGERIS, Christos: NAMS: Network Automated Modeler and Simulator. In: *SS '96 - 29th Annual Simulation Symposium*. Los Alamitos, CA, USA, 1996. – ISSN 1080–241X, S. 65–70
- [61] FAHMY, Hany I.; DOULIGERIS, Christos: An Integrated AI approach for Automating Networks Design, Modeling and Simulation. In: *2nd IEEE Symposium on Computers and Communications*. Los Alamitos, CA, USA, 1997. – ISBN 0–8186–7852–6, S. 339–343
- [62] FAHMY, H.I.; DEVELEKOS, G.; DOULIGERIS, C.: Application of neural networks and machine learning in network design. In: *Selected Areas in Communications, IEEE Journal on* 15 (1997), Februar, Nr. 2, S. 226–237. – ISSN 0733–8716
- [63] FERRARI, D.: Delay Jitter Control Scheme for Packet-Switching Internetworks. In: *Computer Communications* 15 (1992), August, Nr. 6, S. 367–373. – ISSN 0140–3664
- [64] FISCHER, Peter: *Analyse und Bewertung von Kommunikationssystemen in der Gebäudeautomation*, Technische Universität Wien, Dissertation, September 2002
- [65] FLORSTEDT, Thomas: *Erstellung und Implementation von Algorithmen zur Vorhersage der Bandbreitenauslastung innerhalb von verteilten Rechnersystemen*, Technische Universität Dresden, Diplomarbeit, 1999
- [66] FORTAK, H.; JAHN, A.: Entwicklung von Testreferenzjahren (TRY) für Klimaregionen der Bundesrepublik Deutschland / Institut für Geophysikalische Wissenschaften, FU Berlin. 1986 (BMFT FB T 86 051). – Forschungsbericht
- [67] FROLUND, Svend; KOISTINEN, Jari: Quality-of-service specification in distributed object systems. In: *Distributed Systems Engineering* 5 (1998), Dezember, Nr. 4, S. 179–202. – ISSN 0967–1846
- [68] FROST, Victor S.; MELAMED, Benjamin: Traffic Modeling For Telecommunications Networks. In: *IEEE Communications Magazine* 32 (1994), März, Nr. 3, S. 70–81. – ISSN 0163–6804
- [69] FUCHS, Alexander: *Messung von Verzögerungszeiten von Routern und Geräten in einem LON-Netzwerk*, Technische Universität Dresden, Bakkalaureatsarbeit, September 2005
- [70] FURRER, Frank J.: *Ethernet-TCP/IP für die Industrieautomation - Grundlagen und Praxis*. Heidelberg: Hüthig, 1998. – ISBN 3–7785–2641–3
- [71] GARCIA, Alberto E.; HACKBARTH, Klaus D.: Survey Describing Current Internet Design Tools / University of Cantabria. 2005 (D. JRA.3.4.1). – EU Deliveral
- [72] GILBERT, E. N.: Capacity of a Burst Noise Channel. In: *Bell Systems Technical Journal* 39 (1960), September, S. 1253–1266
- [73] GRÖGER, A.: Gebäudebetrieb - Wandel zur vernetzten Technik. In: [224], Kapitel 5, S. 49–57
- [74] GROSSGLAUSER, M.; BOLOT, J.-C.: On the relevance of long-range dependence in network traffic. In: *IEEE/ACM Transactions on Networking* 7 (1999), Oct., Nr. 5, S. 629–640. – ISSN 1063–6692
- [75] GÜRTLER, Mario: *Modellierung des Geräteverhaltens in Netzwerken der Gebäudeautomation*, Technische Universität Dresden, Diplomarbeit, September 2006
- [76] GRZEMBA, A.: Raumautomation - Ein Überblick. In: [223], Kapitel 4, S. 39–50

- [77] HADRÉ, E.: Normgerechte Planung und Ausschreibung der Gebäudeautomation. In: [224], Kapitel 6, S. 67–97
- [78] HAMD AOUI, M.; RAMANATHAN, P.: A dynamic priority assignment technique for streams with (m,k)-firm deadlines. In: *IEEE Transactions on Computers* 44 (1995), Dezember, Nr. 12, S. 1443–1451. – ISSN 0018–9340
- [79] HAMD AOUI, Moncef; RAMANATHAN, Parameswaran: Evaluating Dynamic Failure Probability for Streams with (m, k)-Firm Deadlines. In: *IEEE Transactions on Computers* 46 (1997), Nr. 12, S. 1325–1337. – ISSN 0018–9340
- [80] HAMSCHER, Walter (Hrsg.); CONSOLE, Luca (Hrsg.); KLEER, Johan de (Hrsg.): *Readings in model-based diagnosis*. San Francisco: Morgan Kaufmann Publishers Inc., 1992. – ISBN 1–55860–249–6
- [81] HANZÁLEK, Zdeněk; ČAPEK, Josef: Channel Backlog Estimation in Lon-Works. In: MAHALIK, N.P. (Hrsg.): *Fieldbus Technology - Industrial Network Standards for Real-Time Distributed Control*. London; Berlin; Heidelberg: Springer, 2003. – ISBN 3–540–40183–0, S. 487–500
- [82] HARTEBRODT, Martin; KABITZSCH, Klaus: Fault Detection in Fieldbuses with Time Domain Reflectometry. In: *AFRICON 2004 - 7th IEEE Africon Conference* Bd. 1. Gaborone, Botswana, September 2004, S. 391–396
- [83] HASSEL, Martin v.: *Auswahl, Konfiguration und Einsatz lokaler Netze in Fertigungssystemen*, Technische Universität Hamburg-Harburg, Dissertation, 1991
- [84] HAVES, Philip; SALSBURY, Tim; CLARIDGE, David; LIU, Mingsheng: Use of whole building simulation in on-line performance assessment: modeling and implementation issues. In: *7th International IBPSA Conference Building Simulation*. Rio de Janeiro, Brazilia, 13.–15. August 2001
- [85] HINTZE, Elke; KUCERA, Pavel: Simulation of RFieldbus Networks. In: *FeT 2003 - 5th IFAC International Conference on Fieldbus Systems and their Applications*. Aveiro, Portugal, Juli 2003, S. 115–122
- [86] HOFFMANN, C.; VOSS, Karsten: SolarBau : MONITOR - Tageslichtnutzung und Beleuchtung in Bürogebäuden / Bergische Universität Wuppertal, b+tgA. 2005. – Forschungsbericht
- [87] HONEYWELL: Umfassendes Gebäudemanagement beim Flughafen München. In: *LNO - Brief* (2003), Oktober, Nr. 29
- [88] HONG, Tianzhen; CHOU, S. K.; BONG, T. Y.: Building simulation: An overview of developments and information sources. In: *Building and Environment* 35 (2000), Nr. 4, S. 347–361. – ISSN 0360–1323
- [89] HUNSTOCK, R.; RÜPING, S.; RÜCKERT, U.: A distributed simulator for large networks of building automation systems. In: *WFCS 2000 - 3rd IEEE International Workshop on Factory Communication Systems*. Porto, Portugal, September 2000, S. 203–210
- [90] *IEC 61508 - Functional safety of E/E/PE safety-related systems*. 2005
- [91] *IEC 61511 - Functional safety - Safety instrumented systems for the process industry sector*. 2004
- [92] INTERNATIONAL ALLIANCE FOR INTEROPERABILITY - MODELING SUPPORT GROUP (Hrsg.): *IFC 2x Edition 3 - Model Implementation Guide*. 1.7. International Alliance for Interoperability - Modeling Support Group, März 2004
- [93] INTERNATIONAL ALLIANCE FOR INTEROPERABILITY - MODELING SUPPORT GROUP (Hrsg.): *ifcXML implementation guide*. 1.0. International Alliance for Interoperability - Modeling Support Group, Februar 2005
- [94] ISERMANN, Rolf: Model base fault detection and diagnosis methods. In: *American Control Conference* Bd. 3. Seattle, USA, 21.–23. Juni 1995, S. 1605–1509
- [95] ISERMANN, Rolf; BALLÉ, P.: Trends in the application of model-based fault detection and diagnosis of technical processes. In: *Control Engineering Practice* 5 (1997), Nr. 5, S. 709–719. – ISSN 0967–0661
- [96] *ISO 16484-1 - Building automation and control systems (BACS) - Part 1: Overview and definitions*. 1.1. 2005. – Draft

- [97] ISO 16484-2 -Building automation and control systems (BACS) - Part 2: Hardware. 2004
- [98] ISO 16484-3 -Building automation and control systems (BACS) - Part 3: Functions. 2005
- [99] ISO 16484-5 - Building automation and control systems (BACS) – Part 5: Data communication protocol. 2003
- [100] ISO 9000 - Quality management systems - Fundamentals and vocabulary. 2000
- [101] ISO/IEC 13236 - Quality of Service framework. 1995
- [102] ISO/IEC 7498 - Information technology - Open Systems Interconnection - Basic Reference Model. 2. 1994
- [103] ITU-T Recommendation E.800 - Quality of service and dependability vocabulary. 1994
- [104] JASPERNEITE, J.; NEUMANN, P.; THEIS, M.; WATSON, K.: Deterministic real-time communication with switched Ethernet. In: *WFCS 2002 - 4th IEEE International Workshop on Factory Communication Systems*. Vasteras, Sweden, 27.-30. August 2002, S. 11–18
- [105] JASPERNEITE, Jürgen: *Leistungsbewertung eines lokalen Netzwerkes mit Class-of-Service Unterstützung für die prozessnahe Echtzeitkommunikation*. Shaker Verlag, 2002 (Berichte aus der Kommunikationstechnik). – ISBN 3–8322–0832–1. – Dissertation
- [106] JASPERNEITE, Jürgen; NEUMANN, P.: Measurement, Analysis and Modeling of Real-Time Source Data Traffic in Factory Communication Systems. In: *WFCS 2000 - 3rd IEEE International Workshop on Factory Communication Systems*. Porto, Portugal, 6.–8. September 2000, S. 327–334
- [107] JUNEJA, S.; SHAHABUDDIN, P.: Rare Event Simulation Techniques: An Introduction and Recent Advances. In: HENDERSON, Shane G. (Hrsg.); NELSON, Barry L. (Hrsg.): *Handbooks in Operations Research And Management Science: Simulation*. Elsevier, 2006 (Handbooks in Operations Research and Management Science 13). – ISBN 0–444–50504–0, S. 291–350
- [108] KABITZSCH, Klaus (Hrsg.); BUCHHOLZ, Peter (Hrsg.); PLÖNNIGS, Jörn (Hrsg.): *Netze der Automation - Analyse und Planung von CSMA basierten Systemen*. VDE-Verlag, 2007. – In Vorbereitung
- [109] KABITZSCH, Klaus; NAAKE, Jens; VOLODYMYR VASYUTYNSKYI, Sebastian T.: Untersuchung zum Fernzugriff auf Automatisierungstechnik. In: *atp - Automatisierungstechnische Praxis* 7 (2006), August, Nr. 48. – ISSN 0178–2320
- [110] KABITZSCH, Klaus (Hrsg.); NEUGEBAUER, Mario (Hrsg.): *Netzwerke in der Gebäudeautomation - Modellierung, Voraussage, Planung*. Dresden, Germany, 27. November 2003. – ISBN 3–86005–409–0
- [111] KABITZSCH, Klaus; STEIN, Gunnar; VASYUTYNSKYI, Volodymyr: A Toolset for Stress Test in Field Bus Systems. In: *45th International Scientific Colloquium*. Ilmenau, Germany, Oktober 2000, S. 685–690
- [112] KANT, Krishna: *Introduction to computer system performance evaluation*. New York: McGraw-Hill, 1992. – ISBN 0–07–033586–9
- [113] KARAVAN, Alexander; NEUGEBAUER, Mario; KABITZSCH, Klaus: Merging Building Automation Network Design and IFC 2x Construction Projects. In: *22nd Conference on Information Technology in Construction*. Dresden, Germany, 19.–21. Juli 2005, S. 575–581
- [114] KASTNER, Wolfgang; NEUGSCHWANDTNER, Georg; KÖGLER, Martin: An open approach to EIB/KNX software development. In: *FeT 2005 - 6th IFAC International Conference on Fieldbus Systems and their Applications*. Puebla, Mexico, 14.–15. November 2005, S. 255–262
- [115] KASTNER, Wolfgang; NEUGSCHWANDTNER, Georg; SOUCEK, Stefan; NEWMAN, H. M.: Communication Systems for Building Automation and Control. In: *Proceedings of the IEEE* 93 (2005), Nr. 6, S. 1178–1203. – ISSN 0018–9219

- [116] KATZELA, Irene; SCHWARTZ, Mischa: Schemes for Fault Identification in Communication Networks. In: *IEEE/ACM Transactions on Networking* 3 (1995), Nr. 6, S. 753–764. – ISSN 1063–6692
- [117] KELL, Alan; COLEBROOK, Peter: *Offene Systeme für die Gebäudeautomation: LonWorks und KNX im Vergleich*. www.iandi.ltd.uk/pages/study_deutsch.pdf. Version: 2004. – white paper
- [118] KESHAV, S.; SHARMA, R.: Achieving Quality of Service through Network Performance Management. In: *NOSS-DAV'98 - 8th International Workshop on Network and Operating Systems Support for Digital Audio and Video*. Cambridge, GB, Juli 1998
- [119] KLEIN, S. A.; DUFFIE, J. A.; BECKMAN, W. A.: TRNSYS - A Transient Simulation Program. In: *ASHRAE Transactions* 82 (1976), Nr. 1, S. 623–633. – ISSN 0001–2505
- [120] KLEINROCK, Leonard: *Queueing Systems, vol I & II*. New York: Wiley, 1975 (A Wiley-Interscience publication). – ISBN 0–471–49110–1, 0–471–49111–X
- [121] KNABE, Gottfried: *Gebäudeautomation*. 1. Verlag für Bauwesen, 1992. – ISBN 3–345–00542–8
- [122] KNISSEL, Jens: *Energieeffiziente Bürogebäude mit reduzierten internen Wärmequellen und Wärmeschutz auf Passivhausniveau*. Berlin, Technische Universität Berlin, Dissertation, Oktober 2002
- [123] KNUTH, Donald E.: *The Art of Computer Programming, Volume 3, Sorting and Searching*. 2. Boston: Addison-Wesley, 1998. – ISBN 0–201–89685–0
- [124] KOLMAN, Boris: *Entwicklung einer Diagnosekomponente zur Fehleridentifikation und Ursachenanalyse von Netzwerk-Fehlern im Entwurf*, Technische Universität Dresden, Diplomarbeit, Juli 2006
- [125] KONNEX ASSOCIATION: *Falcon (API) :: Description*. <http://www.konnex.org/tools/falcon/>. Version: 2006
- [126] KOOPMAN, Philip: Critical embedded automotive networks. In: *IEEE Micro* 22 (2002), Juli, Nr. 4, S. 14–18. – ISSN 0272–1732
- [127] KOOPMAN, Philip J.: Tracking down lost messages and system failures. In: *Embedded Systems Programming* 9 (1996), Oktober, Nr. 11, S. 38–52
- [128] KOUBAA, Anis; SONG, Ye-Qiong; THOMESSE, Jean-Pierre: Integrating (m, k)-Firm Real-Time Guarantees into the Internet QoS Model. In: *NETWORKING 2004 - 3rd international IFIP-TC6 Networking Conference*, Springer, 2004 (Lecture Notes in Computer Science 3042). – ISBN 3–540–21959–5, S. 1378–1383
- [129] KRANZ, Hans R.: Integrierte Automation. In: [223], Kapitel 3, S. 15–388
- [130] KRANZ, Hans R.: *BACnet Gebäudeautomation 1.4*. Karlsruhe: Promotor Verlags- und Förderungsgesellschaft mbH, 2005. – ISBN 3–922420–02–8
- [131] KRIESEL, Werner; HEIMBOLD, Tilo; TELSCHOW, Dietmar: *Bustechnologien für die Automation - Vernetzung, Auswahl und Anwendung von Kommunikationssystemen*. 2. Heidelberg: Hüthig, 2000. – ISBN 3–7785–2778–9
- [132] KRONFELDT, Simone: LonWorks im Vergleich mit EIB. In: *de - Der Elektro- und Gebäudetechniker* 15 (2000), Nr. 16, S. 35–41. – ISSN 1617–1160
- [133] KUROSE, James F.; ROSS, Keith W.: *Computer networking - a top-down approach featuring the Internet*. 3. Boston: Addison-Wesley, 2005. – ISBN 0–321–22735–2
- [134] LANGENDÖRFER, Horst: *Leistungsanalyse von Rechensystemen: Messen, Modellieren, Simulation*. München and Wien: Hanser, 1992. – ISBN 3–446–15646–1
- [135] LAUBER, R.: *Prozeßautomatisierung*. Bd. 1. 2. London; Berlin; Heidelberg: Springer, 1989. – ISBN 3–540–50195–9
- [136] LEEN, Gabriel; HEFFERNAN, Donal: Expanding automotive electronic systems. In: *IEEE Computer* 35 (2002), Januar, Nr. 1, S. 88–93. – ISSN 0018–9162
- [137] LELAND, Will E.; TAQQ, Murad S.; WILLINGER, Walter; WILSON, Daniel V.: On the self-similar nature of Ethernet traffic (Extended Version). In: *IEEE/ACM Transactions on networking* 2 (1994),

- Februar, Nr. 1, S. 1–15. – ISSN 1063–6692
- [138] LEWIS, Robert: *Modelling control systems using IEC 61499*. London, UK: IEE - The Institution of Electrical Engineers, 2001 (Control Engineering Series 59). – ISBN 0–85296–796–9
- [139] LI, Yue: *Bewertung der Echtzeitfähigkeit von Feldbus-Systemen*. Düsseldorf, Eggstein-Leopoldshafen, Dissertation, 1993
- [140] LIAN, F. L.; MOYNE, J. R.; TILBURY, D. M.: Performance evaluation of control networks: Ethernet, ControlNet and DeviceNet. In: *IEEE Control Systems Magazine* 21 (2001), Februar, Nr. 1, S. 66–83. – ISSN 0272–1708
- [141] LIAN, Feng-Li: *Analysis, Design, Modeling, and Control of Networked Control Systems*, University of Michigan, Dissertation, 2001
- [142] LIU, Benyuan; FIGUEIREDO, Daniel R.; GUO, Yang; KUROSE, Jim; TOWSLEY, Don: A study of networks simulation efficiency: Fluid simulation vs. packet-level simulation. In: *INFOCOM 2001 - 20th Annual Joint Conference of the IEEE Computer and Communications Societies*, 2001, S. 1244–1253
- [143] LONMARK INTEROPERABILITY ASSOCIATION (Hrsg.): *Layers 1-6 Interoperability Guidelines*. v3.3. LonMark Interoperability Association, 2002
- [144] LONMARK INTEROPERABILITY ASSOCIATION (Hrsg.): *SNVT Master List*. v12. LonMark Interoperability Association, 2003
- [145] LONMARK INTEROPERABILITY ASSOCIATION: *Functional Profiles*. <http://www.lonmark.org/products/fprofile.htm>. Version: 2004
- [146] LONMARK INTEROPERABILITY ASSOCIATION (Hrsg.): *Enumeration Master List*. v13. LonMark Interoperability Association, 2006
- [147] LONMARK INTEROPERABILITY ASSOCIATION (Hrsg.): *Functional Profile: Open-Loop Sensor*. LonMark Interoperability Association, 2006
- [148] LOY, D.; PRINZ, J.: *3150 Neuron Chip*. 1994
- [149] LUKE, N. K.: *Global Protocol Review & End-User Needs Analysis / The Intelligent Buildings Institute Foundation*. Washington, 1993. – Studie
- [150] LUNTOVSKYY, Andriy; GUTTER, Dietbert; VASYUTYNSKY, Volodymyr; KARAVAN, Alexander; KABITZSCH, Klaus: Integrated design of office communication and building automation networks. In: *WFCS 2006 - 6th IEEE International Workshop on Factory Communication Systems*, 2006, S. 235–238
- [151] METZGER, Andreas; QUEINS, Stefan: Specifying Building Automation Systems with PROBanD, a Method Based on Prototyping, Reuse, and Object-orientation. In: *GI-Workshop OMER-1 & OMER-2 on Object-Oriented Modeling of Embedded Real-Time Systems*. Herrsching am Amerssee, Germany: GI, 28.–29. Mai 1999, 10.–12. Mai 2001 2001 (Lecture Notes in Informatics 5), S. 135–140
- [152] MICROSOFT CORPORATION: *Component Object Model Specification*. 0.9, Oktober 1995
- [153] MISKOWICZ, Marek: Analytical approximation of the uniform magnitude-driven sampling effectiveness. In: *ISIE 2004 - IEEE International Symposium on Industrial Electronics* Bd. 1, 2004, S. 407–410
- [154] MISKOWICZ, Marek: Bandwidth requirements for event-driven observations of continuous-time variable. In: *WODES 2004 - 7th IFAC Workshop on Discrete Event Systems*. Reims, France, 22.–24. September 2004, S. 475–480
- [155] MISKOWICZ, Marek: Improving the performance of the networked control system using event-triggered observations. In: *PDS 2004 - IFAC workshop on Programmable Devices and Systems*. Cracow, Poland, 18.–19. November 2004, S. 53–58
- [156] MISKOWICZ, Marek: Sampling of Signals in Energy Domain. In: *ETFA 2005 - 10th IEEE International Conference on Emerging Technologies and Factory Automation* Bd. 1. Catana, Italy, 19.–20. September 2005, S. 263–266

- [157] MISKOWICZ, Marek: Saturation Performance of Predictive p-CSMA without Collision Detection. In: *ETFA 2005 - 10th IEEE International Conference on Emerging Technologies and Factory Automation* Bd. 1. Catania, Italy, 19.–20. September 2005, S. 93–100
- [158] MISKOWICZ, Marek: Send-On-Delta Concept: An Event-Based Data Reporting Strategy. In: *sensors* 6 (2006), Januar, Nr. 1, 49–63. <http://www.mdpi.org/sensors>. – online journal
- [159] MISKOWICZ, Marek; SAPOR, Maria; ZYCH, Marcin; LATAWIEC, Wojciech: Performance analysis of predictive p-persistent CSMA protocol for control networks. In: *WFCS 2002 - 4th IEEE International Workshop on Factory Communication Systems*. Västerås, Sweden, 2002, S. 249–256
- [160] LonWorks Engineering Bulletin: Lon-Talk Response Time Measurements. In: [161]
- [161] MOTOROLA INC. (Hrsg.): *LonWorks Technology Device Data*. Motorola Inc., 1995
- [162] NAGARAJAN, R.; KUROSE, J.: On Defining, Computing, and Guaranteeing Quality-of-Service in High-Speed Networks. In: *INFOCOM 1992 - 11th Annual Joint Conference of the IEEE Computer and Communications Societies*. Florenz, Italien, 4.–8. Mai 1992, S. 2016–2025
- [163] NEUGEBAUER, Mario; PLOENNIGS, Joern; KABITZSCH, Klaus: Analytic Analysis of Duty Cycle Adaptation in Wireless Sensor Networks. In: *ETFA 2006 - 11th IEEE International Conference on Emerging Technologies in Factory Automation*. Prague, Czech Republic, 20.–22. September 2006, S. 113–118
- [164] NEUGEBAUER, Mario; PLÖNNIGS, Jörn; KABITZSCH, Klaus: Prediction of Network Load in Building Automation. In: *FeT 2003 - 5th IFAC International Conference on Fieldbus Systems and their Applications*. Aveiro, Portugal, 7.–8. Juli 2003, S. 269–274
- [165] NEUGEBAUER, Mario; PLÖNNIGS, Jörn; KABITZSCH, Klaus: A New Beacon Order Adaptation Algorithm for IEEE 802.15.4 Networks. In: *EWSN 2005 - 2nd European Workshop on Wireless Sensor Networks*. Istanbul, Turkey, Januar 2005, S. 302–311
- [166] NEUGEBAUER, Mario; PLÖNNIGS, Jörn; KABITZSCH, Klaus; BUCHHOLZ, Peter: Automated Modeling of LonWorks Building Automation Networks. In: *WFCS 2004 - 5th IEEE International Workshop on Factory Communication Systems*. Vienna, Austria, 22.–24. September 2004, S. 113–118
- [167] NEUGEBAUER, Mario; STEIN, Gunnar; KABITZSCH, Klaus: A New Protocol for a Low Power Sensor Network. In: *23rd IEEE International Performance Computing and Communications Conference*. Phoenix, AZ, USA, 15.–17. April 2004, S. 393–399
- [168] NEUGSCHWANDTNER, Georg: Towards Plug and Play in Home and Building Automation Networks. In: *ETFA 2006 - 11th IEEE International Conference on Emerging Technologies and Factory Automation*, 2006, S. 261–264
- [169] NEWRON-SYSTEM: *NLSuite*. <http://www.newron-system.com/>. Version: 2006
- [170] NORD, Lena: *Smart residences - A study of services, network systems and future trends from an energy perspective*. Sweden, Kungl Tekniska Högskolan - The Royal Institute of Technology, Master Thesis, 2001. – EU-Project: On-Line Energy Services for Smart Homes
- [171] OBJECT MANAGEMENT GROUP (Hrsg.): *Unified Modeling Language Specification*. 2.0. Object Management Group, 2004
- [172] PARK, Kihong; SITHARAM, Meera; CHEN, Shaogang: Quality of service provision in noncooperative networks: heterogeneous preferences, multi-dimensional QoS vectors, and burstiness. In: *1st International Conference on Information and Computation Economics*. Charleston, SC, USA, 1998. – ISBN 1–58113–076–7, S. 111–127

- [173] PARK, Kihong; WILLINGER, Walter: Self-similar network traffic: An overview. In: PARK, K. (Hrsg.); WILLINGER, W. (Hrsg.): *Self-Similar Network Traffic and Performance Evaluation*. Wiley Interscience, 2000. – ISBN 0-471-31974-0
- [174] PAXSON, Vern; FLOYD, Sally: Wide Area Traffic: The Failure of Poisson Modeling. In: *IEEE/ACM Transactions on Networking* 3 (1995), Nr. 3, S. 226–244. – ISSN 1063-6692
- [175] PLEINEVAUX, P.; DECOTIGNIE, J.-D.: Time critical communication networks: Field buses. In: *IEEE Network* 2 (1988), Mai, S. 55–63. – ISSN 0890-8044
- [176] PLOENNIGS, Joern; BUCHHOLZ, Peter; NEUGEBAUER, Mario; KABITZSCH, Klaus: Automated Modeling and Analysis of CSMA Type Access-Schemes for Building Automation Networks. In: *IEEE Transactions on Industrial Informatics* 2 (2006), Mai, Nr. 2, S. 103–111. – ISSN 1551-3203
- [177] PLOENNIGS, Joern; NEUGEBAUER, Mario; KABITZSCH, Klaus: Fault Analysis of Control Networks Designs. In: *ETFA 2005 - 10th IEEE International Conference on Emerging Technologies and Factory Automation* Bd. 2. Catania, Italy, 19.–20. September 2005, S. 477–484
- [178] PLOENNIGS, Joern; NEUGEBAUER, Mario; KABITZSCH, Klaus: Automated model generation for performance engineering of building automation networks. In: *International Journal on Software Tools Technology Transfer (STTT)* 8 (2006), November, Nr. 6, S. 607–620. – ISSN 1433-2779
- [179] PLÖNNIGS, Jörn; NEUGEBAUER, Mario; KABITZSCH, Klaus: Automated Model Generation from Design Databases at the Example of Building Automation Networks. In: *ISoLA - 1st International Symposium on Leveraging Applications*. Paphos, Cyprus, 30. Oktober – 2. November 2004, S. 320–327
- [180] PLÖNNIGS, Jörn; NEUGEBAUER, Mario; KABITZSCH, Klaus: A Traffic Model for Networked Devices in the Building Automation. In: *WFCS 2004 - 5th IEEE International Workshop on Factory Communication Systems*. Vienna, Austria, 22.–24. September 2004, S. 137–145
- [181] PLÖNNIGS, Jörn; VASYUTYNSKYI, Volodymyr: AMES-Umfrage in der Zielgruppe „Gebäudeautomation“ / Institut für Angewandte Informatik, Technische Universität Dresden. 2005. – Forschungsbericht
- [182] PRATL, Georg: A Bionic Approach to Artificial Perception and Representation in Factory Automation. In: *ETFA 2005 - 10th IEEE International Conference on Emerging Technologies and Factory Automation* Bd. 1. Catania, Italy, 19.–20. September 2005, S. 251–254
- [183] PRICE, Chris J.: *Computer-based diagnostic systems*. London; Berlin; Heidelberg; Springer, 1999. – ISBN 3-540-76198-5
- [184] RAABE, Herbert P.: Untersuchungen an der wechselseitigen Multiplexübertragung. In: *Elektrische Nachrichtentechnik* 16 (1939), S. 213–228
- [185] RAJI, Reza S.: Smart networks for control. In: *IEEE Spectr.* 31 (1994), Nr. 6, S. 49–55. – ISSN 0018-9235
- [186] RAUSCHMAYER, Axel; KNAPP, Alexander; WIRSING, Martin: Consistency Checking in an Infrastructure for Large-Scale Generative Programming. In: *19th International Conference on Automated Software Engineering*, 2004, S. 238–247
- [187] RIEGEL, Jan P.; SCHÜTZE, Martin; ZIMMERMANN, Gerhard: Pattern Based Generation of Customized, Flexible Building Simulators. In: *CAAD Futures 97*. Dordrecht, Netherlands: Kluwer Academic Publishers, August 1997. – ISBN 0-7923-4726-9, S. 285–298
- [188] RÖHL, V.: Systemintegration - die Herausforderung. In: [224], Kapitel 5, S. 59–66
- [189] RÖSSLER, Peter: *Protokollimplementierungen mit intensiver Hardwareunterstützung am Beispiel des Standards ANSI/EIA 709.1*, Technische Universität Wien, Dissertation, April 2001

- [190] RÜPING, Stefan; HUNSTOCK, Ralf; GUNREBEN, Uwe: Simulation of LonWorks Systems. In: *LUI 97 - LonUsers International Fall 97 Conference and Exhibition*. Amsterdam, Netherlands, Oktober 1997. – on CD
- [191] RUMBAUGH, James; JACOBSON, Ivar; BOOCH, Grady: *The Unified Modeling Language Reference Manual*. 2. Boston: Addison-Wesley, 1999. – ISBN 0-321-24562-8
- [192] RYSSEL, Uwe; PLOENNIGS, Joern; KABITZSCH, Klaus: Generative Function Block Design and Composition. In: *WFCS 2006 - 6th IEEE International Workshop on Factory Communication Systems*. Torino, Italy, 28.–30. Juni 2006, S. 253–263
- [193] RYU, Bong K.; ELWALID, Anwar: The Importance of Long-Range Dependence of VBR Video Traffic in ATM Traffic Engineering: Myths and Realities. In: *SIGCOMM*. Stanford, CA, USA, 28.–30. August 1996, S. 3–14
- [194] SAVINO-VÁZQUEZ, Nunzio-Nicoló; PUIGJANER, Ramon: A UML-based Method to Specify the Structural Component of Simulation-based Queuing Network Performance Models. In: *32nd Annual Simulation Symposium*. San Diego, CA, USA, 11.–15. April 1999. – ISBN 0-7695-0128-1, S. 71–78
- [195] SCHMALEK, Richard: *Leistungsanalyse von Feldebussystemen*, Technische Universität Wien, Dissertation, Februar 1996
- [196] SCHNELL, Gerhard (Hrsg.): *Bussysteme in der Automatisierung und Prozesstechnik - Grundlagen, Systeme und Trends der industriellen Kommunikation*. 5. Vieweg Verlag, 2003. – ISBN 3-528-46569-7
- [197] SCHÖNING, Uwe: *Algorithmen - kurz gefasst*. Heidelberg, Berlin: Spektrum Akademischer Verlag, 1997 (Spektrum-Hochschultaschenbuch). – ISBN 3-8274-0232-8
- [198] SCHRICK, Dirk van: *Entepetives Management - Konstrukt, Konstruktion, Konzeption: Entwurf eines Begriffssystems zum Umgang mit Fehlern, Ausfällen und anderen nichterwünschten technischen Phänomenen*, Bergische Universität Wuppertal, Habilitation, 2002
- [199] SCHROM, Harald: *Realisierung eines optimierten Feldebussystems und Modellierung mit Petrinetzen*, Technischen Universität Carolo-Wilhelmina zu Braunschweig, Dissertation, 2003
- [200] SCHWAIGER, Christian; TREYTL, Albert: Smart card based security for fieldbus systems. In: *ETFA 2003 - 9th IEEE International Conference on Emerging Technologies and Factory Automation* Bd. 1, 2003, S. 398–406
- [201] SCHWARZ, K.: Industriestandards - Ergänzung oder Ersatz internationaler Normen. In: *Vernetzung durch industrielle Kommunikation*. Düsseldorf: VDI Verlag, 5. Mai 1994 (VDI Berichte 1123). – ISBN 3-18-091123-9, S. 33–44
- [202] SCHWARZ, Peter; DONATH, Ulrich: Simulation-based Performance Analysis of Distributed Systems. In: *International Workshop Parallel and Distributed Real-Time Systems*, 1997, S. 244–249
- [203] SEEMANN, Jochen; GUDENBERG, Jürgen W.: *Software-Entwurf mit UML 2*. 2. London; Berlin; Heidelberg: Springer, 2006. – ISBN 3-540-30949-7
- [204] SHANNON, Claude E.: Communications in the presence of noise. In: *Proceedings of the Institute of Radio Engineers* Bd. 37, 1949, S. 10–24
- [205] SO, Albert Ting-pat; CHAN, Wai L.: *Intelligent building systems*. Boston: Kluwer Academic Publ., 1999 (Kluwer international series on Asian studies in computer and information science 5). – ISBN 0-7923-8491-1
- [206] SOUCEK, Stefan; SAUTER, Thilo: Quality of Service Concerns in IP-based Control Systems. In: *IEEE Transactions on Industrial Electronics* 51 (2004), Nr. 9, S. 1249–1258. – ISSN 0278-0046
- [207] STEINDER, M.; SETHI, A.: The present and future of event correlation: A need for end-to-end service fault localization. In: *IIIS SCI - 5th World Multi-Conference Systemics Cybernetics Informatics*. Orlando, FL, USA, 22.–25. Juli 2001, S. 124–129

- [208] STILLER, Burkhard: *Quality-of-Service: Dienstgüte in Hochleistungsnetzen*. Bonn: Internat. Thomson Publ., 1996 (Thomson's aktuelle Tutorien 21). – ISBN 3-8266-0171-8
- [209] STÖCKLHUBER, Andreas: Eine Übersicht über die wichtigsten Lon-Programmiertools. In: *de-Special: Bussysteme für die Gebäudeinstallation* (1999), S. 143–146
- [210] STURM, Thomas F.: *Semantische Gerätevernetzung / Universität der Bundeswehr München*. 2005. – Skriptum
- [211] TANENBAUM, Andrew S.: *Computer networks*. 3. Upper Saddle River, NJ: Prentice-Hall, 1996 (Prentice-Hall international editions). – ISBN 0-13-349945-6
- [212] THOMESSE, Jean-Pierre: Fieldbuses and Quality of Service. In: *Controlo 2002-5th Portuguese Conference on Automatic Control*. Aveiro, Portugal, 2002, S. 10–14
- [213] TOMURA, Toyoaki; UEHIRO, Kiyoshi; KANAI, Satoshi; YAMAMOTO, Susumu: Developing Simulation Models of Open Distributed Control System by Using Object-Oriented Structural and Behavioral Patterns. In: *4th IEEE International Symposium on Object-Oriented Real-Time Distributed Computing*. Magdeburg, Germany, Mai 2001, S. 428–437
- [214] TRAN-GIA, Phuoc: *Analytische Leistungsbewertung verteilter Systeme - eine Einführung*. London; Berlin; Heidelberg; Springer, 1996. – ISBN 3-540-60666-1
- [215] TREYTL, Albert; SAUTER, Thilo; SCHWAIGER, C.: Security Measures in Automation Systems - a Practice-Oriented Approach. In: *ETFA 2005 - 10th IEEE International Conference on Emerging Technologies and Factory Automation* Bd. 2. Catania, Italy, September 2005, S. 847 – 855
- [216] ULIERU, Mihaela: Diagnosis by approximate reasoning on dynamic fuzzy fault trees. In: *3rd IEEE Conference on Fuzzy Systems and IEEE World Congress on Computational Intelligence* Bd. 3. Orlando, FL, USA, 26.–29. Juni 1994, S. 2051–2056
- [217] VASYUTYNSKY, Volodymyr; PLOENIGS, Joern; KABITZSCH, Klaus: Passive Monitoring of Control Loops in Building Automation. In: *5th IFAC International Conference on Fieldbus Systems and their Applications*. Puebla, Mexico, 14.–15. November 2005, S. 263–269
- [218] *VDI 3687 - Auswahl von Feldbussystemen durch Bewertung ihrer Leistungseigenschaften für industrielle Anwendungsbereiche*. November 1999
- [219] *VDI 3814- Gebäudeleittechnik*. 1983–96
- [220] *VDI 6010 - Sicherheitstechnische Einrichtungen*. 2005
- [221] *VDI 6015 - BUS-Systeme in der Gebäudeinstallation - Anwendungsbeispiele*. 2003
- [222] *VDI 6020 - Anforderungen an Rechenverfahren zur Gebäude- und Anlagensimulation - Gebäudesimulation*. Mai 2005
- [223] VEREIN DEUTSCHER INGENIEURE (Hrsg.): *Vernetzte Automation in der Gebäudetechnik*. Düsseldorf: VDI Verlag, 2002 (VDI-Berichte 1639). – ISBN 3-18-091639-7
- [224] VEREIN DEUTSCHER INGENIEURE (Hrsg.): *Gebäudeautomation - Voraussetzung für das Gebäudemanagement*. Düsseldorf: VDI Verlag, 2003 (VDI-Berichte 1740). – ISBN 3-18-091740-7
- [225] VEREIN DEUTSCHER INGENIEURE (Hrsg.): *Raumautomation*. Düsseldorf: VDI Verlag, 2004 (VDI-Berichte 1816). – ISBN 3-18-091816-0
- [226] VINYES; VAZQUEZ; MIGUEL: Throughput analysis of p-csma based lontalk protocols for building management systems. In: *MELECON 1996 - 8th Mediterranean Electrotechnical Conference* Bd. 3, 1996, S. 1741–1744
- [227] VYATKIN, V.; HANISCH, H.-M.: Formal modeling and verification in the software engineeringframework of IEC 61499: a way to self-verifying systems. In: *ETFA 2001 - 8th IEEE International Conference on Emerging Technologies and*

- Factory Automation* Bd. 2. Antibes-Juan les Pins, France, 15–18 Oktober 2001, S. 113–118
- [228] WANG, Q. P.; TAN, D. I.; WANG, Y. C.: The Control Oriented QoS: Analysis and Prediction. In: *IEEE International Conference on Robotics and Automation*. Seoul, Korea, Mai 2001, S. 1897–1902
- [229] WANG, Zhi; HUANG, Fang; SUN, Youxian; SONG, Yeqiong: Colored Petri net model of IEC function block and its application. In: *ETFA 2003 - 9th IEEE International Conference on Emerging Technologies and Factory Automation* Bd. 1, 2003, S. 648–651
- [230] WANG, Zhi; SONG, Ye Q.; POGGI, Enrico-Maria; SUN, Youxian: Survey of Weakly-Hard Real Time Schedule Theory and Its Application. In: *International Symposium on Distributed Computing and Applications to Business, Engineering and Science*, 2002, S. 429–437
- [231] Warema. <http://www.warema.de>. Version: 2006
- [232] WÄCHTLER, Reinhard: *Ein Beitrag zur Verkehrsgenerierung und Leistungsmessung in computergestützten Kommunikationssystemen*, Technische Universität Dresden, Dissertation, 1997
- [233] WEI, Wei; WANG, Bing; TOWSLEY, Don: Continuous-time hidden Markov models for network performance evaluation. In: *Performance Evaluation* 49 (2002), Nr. 1-4, S. 129–146. – ISSN 0166–5316
- [234] WENDT, Daniel: *Evaluierung eines komponentenbasierten Entwurfes unter unterschiedlichen funktionalen Aspekten am Beispiel NetPlan*, Technische Universität Dresden, Diplomarbeit, 2005
- [235] WERTHSCHULTE, Kay: *Integration von heterogenen Bussystemen in die Heimautomatisierung unter Verwendung von Middleware*, Technische Universität München, Dissertation, 2003
- [236] WESTERMEIER, Günter: *Diversitäre Zugangs- und Sicherheitsmechanismen angewendet in automatisierten Gebäuden*, Technische Universität München, Dissertation, 2004
- [237] WILLIG, Andreas: Scheduling Multiple Streams with (m,k)-firm Deadlines having Different Importance over Markovian Channels. In: *ETFA 2005 - 10th IEEE International Conference on Emerging Technologies and Factory Automation*. Catania, Italy, 19.–22.September 2005, S. 79–85
- [238] WILLIG, Andreas; KUBISCH, Martin; HOENE, Christian; WOLISZ, Adam: Measurements of a Wireless Link in an Industrial Environment using an IEEE 802.11-Compliant Physical Layer. In: *IEEE Transactions on Industrial Electronics* 43 (2002), Nr. 6, S. 1265–1282. – ISSN 0278–0046
- [239] WILLIG, Andreas; WOLISZ, Adam: Ring stability of the PROFIBUS token-passing protocol over error-prone links. In: *IEEE Transactions on Industrial Electronics* 48 (2001), Oktober, Nr. 5, S. 1025–1033. – ISSN 0278–0046
- [240] WILLINGER, Walter; TAQQU, Murad S.; SHERMAN, Robert; WILSON, Daniel V.: Ethernet LAN traffic at the source level. In: *IEEE/ACM Transactions on Networking* 5 (1997), Nr. 1, S. 71–86. – ISSN 1063–6692
- [241] WIRTH, U.: Gebäudeautomation - Geschäftstransparenz mit internationaler Normierung. In: [224], Kapitel 1, S. 1–12
- [242] WOLF, Dietrich: *Signaltheorie - Modelle und Strukturen*. London; Berlin; Heidelberg: Springer, 1999. – ISBN 3–540–65793–2
- [243] WOODSIDE, Murray; HRISCHUK, Curtis; SELIC, Bran; BAYAROV, Stefan: A wideband approach to integrating performance prediction into a software design environment. In: *1st International Workshop on Software and Performance*. New York, NY, USA: ACM Press, 1998, S. 31–41
- [244] WRATIL, Peter: *SafetyLON Prinzipdokument*. 2005
- [245] XIAOMING, Chen; GEOK-SOON, Hong: A simulation study of the predictive p-persistent CSMA protocol. In: *35th Annual Simulation Symposium*, 2002. – ISSN 1082–241X, S. 345–351

- [246] ZEIGLER, B. P.: *Object-Oriented Simulation with Hierarchical, Modular Models: Intelligent Agents and Endomorphic Systems*. Academic Press, 1990. – ISBN 0–12–778452–7
- [247] ZITZLER, Eckart; DEB, Kalyanmoy; THIELE, Lothar: Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. In: *Evolutionary Computation* 8 (2000), Nr. 2, S. 173–195. – ISSN 1063–6560
- [248] ZITZLER, Eckart; THIELE, Lothar: Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. In: *IEEE Transactions on Evolutionary Computation* 3 (1999), Nr. 4, S. 257–271. – ISSN 1089–778X

SCHLAGWORTVERZEICHNIS

— A —

Abgangsrate ◊ 111
Adjazenzmatrix ◊ 149
Adresse ◊ 46
Adressraum ◊ 40
Anforderungen ◊ 21
Ankunftsprozess ◊ 64
Ankunftsrate ◊ 57
Ankunftsratenmodifizier ◊ 57 ◊ 113
Ankunftsratenreduktion ◊ 148
Anwendungsschicht ◊ 48
AR-Modell ◊ 125
Aufzählungsdatentyp ◊ 50 ◊ 80
Auslastung ◊ 112 ◊ 123
Automatisierung ◊ 22
Automatisierungsnetzwerk ◊ 32

— B —

Bürokommunikation ◊ 10
Backbone ◊ 44
BACnet ◊ 33
Bearbeitungszeit ◊ 64
Bedienrate ◊ 112
Betriebssicherheit ◊ 7 ◊ 14
Binding ◊ 48
Bitfehler ◊ 139
Boolean ◊ 50
Broadcast ◊ 46
Building Automation and Control ◊ *siehe*
Gebäudeautomation
Burst ◊ 62

— C —

Channel ◊ *siehe* Kanal ◊ 44
Commissioning ◊ *siehe* Inbetriebnahme
Computer Integrated Buildings ◊ 10
Configuration ◊ *siehe* Konfigurierung
Control Network Performance Engineering
◊ 5 ◊ 30
CSMA ◊ 34

Cumulative Distribution Function ◊ *siehe*
Kumulative Verteilungsfunktion

— D —

Data Point ◊ *siehe* Datenpunkt
Datendurchsatz ◊ 111
akkumuliert ◊ 143
relativ ◊ 143 ◊ 150
Datenpunkt ◊ 48
Delay ◊ *siehe* Verzögerungszeit
Design Pattern ◊ *siehe* Entwurfsmuster
Device ◊ *siehe* Gerät ◊ 44 ◊ 46
Dichtefunktion ◊ 66
Dienstqualität ◊ 14 ◊ 27 ◊ 119 ◊ 121
Garantieklassen
bestmöglich ◊ 122
garantiert ◊ 122
QoS-Fehler ◊ 124
QoS-Warnung ◊ 124
Differenziertheit ◊ 22
Dimensionierung
Überdimensionierung ◊ 17
Unterdimensionierung ◊ 17
Domain ◊ 46
Durchsatz ◊ 111

— E —

Echtzeit ◊ 7
Echtzeitverhalten ◊ 14
Effizienz ◊ 21
Engineering ◊ 11
Entwurf ◊ 10
Entwurfsdatenbank ◊ 26
Entwurfsfehler ◊ 15
Entwurfsmethode ◊ 15
Entwurfsmuster ◊ 15
Entwurfsparadigma ◊ 15
Entwurfssprache ◊ 15
Entwurfstool ◊ 11 ◊ 26
Enumeration ◊ *siehe* Aufzählungsdatentyp

Ereignisstrom ◊ 64
 Erfolgswahrscheinlichkeit ◊ 112
 Erneuerungsprozess ◊ 66
 Europäischer Installations-Bus ◊ 33
 European Installation Bus ◊ *siehe*
 Europäischer Installations-Bus
 ExtendedAttribute ◊ 59

— F —

Fahrzeugtechnik ◊ 9
 Fehler ◊ *siehe* Entwurfsfehler
 Fehleranalyse ◊ 119
 Fehlerbaum ◊ 131
 Fehlerwahrscheinlichkeit ◊ 112 ◊ 123
 Feldbus ◊ 32
 Fertigungstechnik ◊ 9
 Filterprozess ◊ 64
 Fix-Punkt-Analyse ◊ 109
 Function Block ◊ *siehe* Funktionsblock
 Funktionsblock ◊ 48
 virtuell ◊ 58
 Funktionsimplementation
 korrekt ◊ 13

— G —

Ganzzahl ◊ 50
 Garantieklassen ◊ 122
 Garantierbarkeit ◊ 14
 Gebäudeautomation ◊ 8
 Genauigkeit ◊ 22
 Gerät ◊ 43
 Gerätemodell ◊ 42 ◊ 64
 Übertragbarkeit ◊ 100
 abstrakt ◊ 68
 detailliert ◊ 67
 Genauigkeit ◊ 100
 Geräteprofil ◊ 48
 Gewerk ◊ 9
 Gilbert-Elliot-Modell ◊ 139
 Gleitkommazahl ◊ 50
 Group Object ◊ *siehe* Gruppenobjekt
 Gruppe ◊ 48
 Gruppenobjekt ◊ 48

— H —

Heimautomation ◊ 9
 Home Automation ◊ *siehe* Heimautomation

— I —

Inbetriebnahme ◊ 11
 Informationsgröße ◊ 51

Informationstechnische Anlagen ◊ 8
 Integration ◊ 22
 Interworking-Datapoint-Type ◊ 50

— J —

Jitter ◊ 123

— K —

Kanal ◊ 43
 KNX ◊ *siehe* Konnex
 Kommunikation
 Engineering ◊ 58
 Management ◊ 58
 Prozess ◊ 58
 Komplexität ◊ 9
 Komposition ◊ 16 ◊ 48
 Konfiguration ◊ *siehe* Netzwerkentwurf
 Konfigurierung ◊ 11
 Konnex ◊ 33
 Korrelationshorizont ◊ 72
 Kosten
 angemessen ◊ 14
 Kumulative Verteilungsfunktion ◊ 83

— L —

λ -Filter ◊ 68
 λ -Quelle ◊ 68
 λ -Senke ◊ 68
 Langzeitabhängigkeit ◊ 62
 Last ◊ *siehe* Datendurchsatz
 Leistungsbewertung ◊ 23
 Analytisch ◊ 24
 Messung ◊ 23
 Simulation
 abstrakt ◊ 24
 deatailliert ◊ 23
 HIL ◊ 23
 Leistungskenngrößen ◊ 23 ◊ 27 ◊ 29
 Leistungsmodell ◊ 25
 Level-Crossing-Problem ◊ 91
 Linie ◊ 46
 Local Operating Network ◊ 33
 LonTalk ◊ 33
 LonWorks ◊ 33

— M —

Managementnetzwerk ◊ 32
 Max-Send-Time ◊ 76 ◊ 92
 Min-Send-Time ◊ 76 ◊ 92
 Multicast ◊ 46

— N —

Nachrichtengenerierung
 ereignisorientiert ◊ 75
 Schwellwert ◊ 90
 Send-On-Delta ◊ 75 ◊ 81
 zeitorientiert ◊ 75
 Nachrichtengröße ◊ 51
 Nachrichtenklasse ◊ 51
 Network Performance Engineering ◊ 25
 Netzwerk ◊ 11
 Netzwerk-Variablen ◊ 48
 Netzwerkentwurf ◊ 11
 Netzwerkmanagement ◊ 28 ◊ 48 ◊ 120
 Netzwerkdiagnose ◊ 28 ◊ 120
 Netzwerkprotokoll ◊ 31
 Node ◊ 46
 Nutzerberatung ◊ 120

— O —

OSI-Referenzmodell ◊ 31

— P —

Pareto-Menge ◊ 141
 Pareto-Optimum ◊ 141
 Performance Evaluation ◊ *siehe* Leistungs-
 bewertung
 Petrinetz ◊ 64
 Poisson-Prozess ◊ 67
 Port ◊ 45
 Prämisse ◊ 16
 Prioritätsberatung ◊ 146
 Prioritätslot ◊ 146
 Probability Distribution Function ◊ *siehe*
 Dichtefunktion
 Profil ◊ *siehe* Geräteprofil
 Protocol Data Unit ◊ 51
 Prozess
 stochastisch ◊ 65
 Prozesstechnik ◊ 9

— Q —

QoS Requirement ◊ *siehe* QoS-Anforderung
 QoS-Anforderung ◊ 122
 QoS-Charakteristikum ◊ 121
 Ableitung ◊ 121
 generisch ◊ 121
 Spezialisierung ◊ 121
 QoS-Defekt ◊ 122
 QoS-Gewicht ◊ 131
 QoS-Parameter ◊ *siehe* QoS-Anforderung
 QoS-Risikoklasse

Funktionsanforderung ◊ 131
 Komfortanforderung ◊ 131
 Sicherheitsanforderung ◊ 131
 QoS-Syntax ◊ *siehe* QoS-Anforderung
 Qualität ◊ 13
 Qualität des Netzwerkentwurfs ◊ 13
 Qualitätskenngrößen ◊ 29
 Qualitätsmerkmal ◊ 13 ◊ 122
 Quality of Control ◊ *siehe* Regelqualität
 Quality of Service ◊ *siehe* Dienstqualität
 Quellprozess ◊ 64 ◊ 74

— R —

Real Time ◊ *siehe* Echtzeit
 Referenzmessung ◊ 62
 Regelqualität ◊ 153
 Risiko ◊ 7 ◊ 130
 finanziell ◊ 17
 Risk ◊ *siehe* Risiko
 Router ◊ 43 f.

— S —

Safety ◊ *siehe* Betriebssicherheit
 Security ◊ *siehe* Sicherheit
 Selbstähnlichkeit ◊ 62
 Selektor ◊ 55
 Sensibilitätsmatrix ◊ 149
 Sensibilitätsvektor ◊ 150
 Sicherheit ◊ 53
 Signal
 kontinuierlich ◊ 74
 Zustandssignal ◊ 74
 Software Performance Engineering ◊ 26
 Standard Network Variable Type ◊ *siehe*
 Standard Netzwerk Variablen Typ
 Standard Netzwerk Variablen Typ ◊ 50
 Standard Object Property ◊ 50
 Strukturmodell Gebäudeautomation ◊ 32
 Subnetz ◊ 46
 Systemmodell ◊ 25 ◊ 40
 Anwendungsschicht ◊ 49
 Bitübertragungsschicht ◊ 43
 Nachrichtenklassen ◊ 51 ◊ 54 ◊ 111
 Vermittlungsschicht ◊ 46

— T —

Technische Gebäudeausrüstung ◊ 8
 Teilziel ◊ 28
 Throughput ◊ *siehe* Durchsatz
 Timer
 einmalig ◊ 78

wiederholt ◊ 77
Topologie ◊ 40 ◊ 43
Transaktion ◊ 53

— U —

Übertragungsdienst ◊ 53
 authentifiziert ◊ 53
 beantwortet ◊ 53
 bestätigt ◊ 53 ◊ 73
 unbestätigt ◊ 53
 unbestätigt wiederholt ◊ 53 ◊ 73
Übertragungsmedium ◊ 34
Übergangswahrscheinlichkeit ◊ 65
Übertragungsdienst ◊ 50
Übertragungsdienstberatung ◊ 143
Übertragungsdienstreduktion ◊ 147
Unicast ◊ 46
Unified Modelling Language ◊ 40
Utilisation ◊ *siehe* Auslastung

— V —

Verfügbarkeit ◊ 14
Verkehrsmodell ◊ 25 ◊ 40
Verständlichkeit ◊ 22
Verzögerungszeit ◊ 64 ◊ 112 ◊ 123

— W —

Wiederholungstimer ◊ 53

— Z —

Zielstellung ◊ 21
Zone ◊ 46