

Entwurf eines ATM-basierten Funknetzes und Software-Entwurfsmethodik zur Implementierung

Stefan Böhmer

Entwurf eines ATM-basierten Funknetzes und Software-Entwurfsmethodik zur Implementierung

Entwurf eines ATM-basierten Funknetzes

Entwurf eines ATM-basierten Funknetzes

Entwurf eines ATM-basierten Funknetzes

Entwurf eines ATM-basierten Funknetzes

Entwurf eines ATM-basierten Funknetzes

Entwurf eines ATM-basierten Funknetzes

Entwurf eines ATM-basierten Funknetzes

Entwurf eines ATM-basierten Funknetzes

AACHENER BEITRÄGE ZUR MOBIL- UND TELEKOMMUNIKATION

Herausgeber:

Universitätsprofessor Dr.-Ing. Bernhard Walke

Böhmer, Stefan:

Entwurf eines ATM-basierten Funknetzes und Software-Entwurfsmethodik zur Implementierung

1. Auflage Aachen:

Verlag der Augustinus Buchhandlung, 1996

(Aachener Beiträge zur Mobil- und Telekommunikation, Band 5)

ISBN 3-86073-384-2

© 1996 Stefan Böhmer

Verlag der Augustinus Buchhandlung

Pontstraße 96

52062 Aachen

Tel. & Fax: 0241-23948

Druck: Druckerei Paletti, Aachen

Gedruckt auf chlorfrei gebleichtem Papier

"D 82 (Diss. RWTH Aachen)"

ZUSAMMENFASSUNG

ATM als paketvermittelnde Übertragungstechnik digitaler Breitbandnetze wird gegenwärtig im Festnetz eingeführt und motiviert zu Überlegungen, wie Benutzer schnurlos angeschlossen werden können. Dabei geht es z.B. um Endgeräte eines drahtlosen LANs in Gebäuden oder im Zugangsnetz eines Telekommunikationsnetzes. Wegen der geringen Funkübertragungsraten in Zugangsnetzen ist es zweckmäßig, Sprachdienste durch synchrone Kanäle und die übrigen Dienste paketvermittelt bzw. ATM-basiert zu unterstützen. Dementsprechend werden ein dezentral organisiertes Funknetz und seine Funkschnittstelle entwickelt, als Modell zur verkehrstheoretischen Leistungsanalyse formuliert und durch stochastische Simulation bewertet. Es basiert auf einem Pulsrahmen, überträgt verbindungsorientiert und macht Übertragungskapazität im Vielfachzugriff verfügbar. Die Stationen des Netzes arbeiten als Vermittlungsrelais, um Routen auch über mehrere Teilstrecken zum gewünschten Ziel (gerufene Station oder Festnetz-zugang) zu leiten. Mit aufwendigen Maßnahmen werden sog. Multi-Hop Umgebungen mit versteckten Stationen unterstützt, um frequenzökonomisch übertragen zu können. Ein Leistungsvergleich mit dem HIPERLAN-System, einem bei ETSI standardisierten drahtlosen LAN, ergibt, daß das in dieser Arbeit entwickelte System für ATM-Dienste den fünffachen Durchsatz erreicht. Eine Vorversion wird gegenwärtig industriell realisiert. In einem zweiten Teil der Arbeit wird eine neue, auf der formalen Spezifikations-sprache SDL basierende Softwareentwicklungsmethodik für Kommunikationsprotokolle entwickelt, die den Stand der Technik um zwei wesentliche Komponenten erweitert: In der Entwicklungsumgebung für formal spezifizierte Protokolle, üblicherweise einem rechnergestützten Werkzeug, werden sowohl durch stochastische Simulation das Leistungsverhalten der entwickelten Software in einer ebenfalls formal spezifizierten Umgebung, als auch die erforderliche Hardware-Verarbeitungsleistung des Zielsystems untersuchbar. Damit können die Implementierung von Softwaremodulen (z.B. Protokollen) im Werkzeug optimiert und auf die vorgegebene Rechnerplattform abgestimmt, bzw. die Leistung der Plattform definiert werden. Diese neuen Konzepte werden in dieser Arbeit validiert und z.Zt. zusammen mit einem in Europa führenden Hersteller von Softwareentwicklungswerkzeugen für den praktischen Gebrauch umgesetzt.

ABSTRACT

ATM, a packet switching transmission technique, is currently introduced in fixed broadband networks and motivates considerations on how wireless access can be made available to users. Terminals for wireless local area networks and wireless access networks are taken into account. Due to the low transmission rates of radio networks it is useful to support speech transmission by means of synchronous channels and other services on a packet switched (ATM) basis. In accordance to this, a decentrally organized radio network and its appropriate radio interface is developed, formulated as a model for traffic analysis and evaluated by means of stochastic simulation. It bases on a pulse frame, transmits connection-orientedly and provides transmission capacity for multiple access. Every station works as a relay, so that routing is supported over several radio links to the desired target (called station or fixed network access). By means of sophisticated measures, so called multi-hop networks with hidden stations are operated in order to facilitate a frequency efficient transmission. A performance comparison with the HIPERLAN system, which is a wireless local area network standardized by ETSI, shows that the system developed in this thesis reaches a fivefold throughput for ATM services. A preliminary version is currently being realized for industrial purposes. The second part of the thesis introduces a new software development technique for communication systems based on the formal specification language SDL, which enhances the state of the art by two essential components: in a development workbench for formally specified protocols the performance of the developed software can be evaluated by stochastic simulation as well as the necessary hardware performance of the target system. Thus, the implementation of software modules (e.g. protocols) can be optimized inside the tool, be suited to a given hardware platform and the performance of the platform can be defined respectively. These new concepts are validated in this thesis and are currently realized for practical use in cooperation with a leading supplier of software development tools in Europe.

INHALTSVERZEICHNIS

1	Einleitung	1
2	Mobilfunknetze	4
2.1	Einleitung	4
2.2	Zentral organisierte Mobilfunknetze	5
2.3	Dezentral organisierte Mobilfunknetze	6
2.4	HIPERLAN	8
2.4.1	Kenngrößen von HIPERLAN-Netzen	8
2.4.2	HIPERLAN-Referenzmodell	9
2.4.3	HIPERLAN-Bitübertragungsschicht	11
2.5	DECT	12
2.5.1	Eigenschaften und Kenngrößen von DECT-Systemen	12
2.5.2	DECT-Referenzmodell	15
2.5.3	DECT-Relais	17
2.6	Dezentral organisiertes Multi-Hop Funknetz für innen und außen	18
2.6.1	Eigenschaften des Multi-Hop Funknetzes	18
2.6.2	Referenzmodell des dezentral organisierten Multi-Hop Funknetzes	19
2.6.3	Verfahren zur Ermittlung und Aktualisierung von Routing Informationen	20
2.6.4	Verfahren für die An-/Abmeldung von Netzstationen	21
2.6.5	Bitübertragungsschicht des Multi-Hop Funknetzes	22
3	Formale Spezifikation von Kommunikationsprotokollen	25
3.1	Formale Methoden und Sprachen	25
3.2	Einsatzbereiche von SDL	26
3.3	Standardisierung	26
3.4	SDL-92	29
3.4.1	Struktur einer SDL-Spezifikation	29
3.4.2	Kommunikation zwischen Instanzen	30
3.4.3	Die Spezifikation von SDL-Typen	31

3.4.4	Parametrisierung von Typen	32
3.4.5	Operationen auf Typen	34
3.4.6	Spezifikation des Systemverhaltens	37
3.4.7	Daten in SDL-92	39
4	RMAN: ein ATM-basiertes Mobilfunknetz	44
4.1	Entwurfskriterien und -ziele für das RMAN	44
4.2	Verfahren zur adaptiven, dienstspezifischen Bandbreitenzu- teilung	47
4.3	Störeinfluß versteckter Stationen in teilvermaschten Funknet- zen	51
4.4	Protokoll für den Betrieb kanalvermittelter Verbindungen . .	53
4.4.1	Protokoll für Einrichtung und Betrieb von Teil- streckenverbindungen	54
4.4.2	Datenübertragungsphase und Verbindungsauslösung .	56
4.5	Protokolle zur Übertragung von ATM-Zellen	56
4.5.1	DPRMA-Protokoll für One-Hop Teilstrecken	57
4.5.2	DPRMA-Protokoll für Einrichtung, Betrieb und Auslösung virtueller ATM-Verbindungen	64
4.5.3	Zusammenfassung	70
5	Der RMAN-Simulator zur Leistungsbewertung	71
5.1	Entwurfskriterien und Realisierung	71
5.2	Software-Architektur des RMAN-Simulators	72
5.2.1	Stationen	74
5.2.2	Protokolle	74
5.2.3	Netzkonfiguration	75
5.2.4	Funkausbreitung und Kanalmodell	75
5.2.5	Konfigurationsdatei und Parser	76
5.2.6	Simulatorbibliothek	77
5.2.7	Graphische Benutzeroberfläche	78
6	RMAN: Simulationsszenarien und Leistungsbewertung	80
6.1	Kenngrößen zur Charakterisierung von Multi-Hop Funknetzen	80
6.2	Simulationsszenarien und -parameter	82
6.2.1	Szenarien	82
6.2.2	Simulationsparameter	84
6.3	Simulationsergebnisse und Bewertung	86

6.3.1	Szenario 'Vollvermaschtes Netz' – Vergleich von RMAN und HIPERLAN	86
6.3.2	Szenarien 'Lokales Netz' und 'Teilnetz im Innenstadtbereich'	90
7	Softwareentwicklungsmethodik für Kommunikationssysteme	95
7.1	Motivation	95
7.2	Konzept des erweiterten System-Entwicklungszyklus	96
7.3	Basismodell und erste Realisierung des Simulationswerkzeuges	99
7.3.1	Basis-Simulationsmodell	100
7.3.2	Der multimediale Lastgenerator MultiGen des SPEET-Simulators	102
7.3.3	Die Simulationssteuerung des SPEET-Simulators	108
7.3.4	Statistische Auswertung von Simulationsmeßdaten	109
7.3.5	Kommunikations- und Steuerkanäle	110
7.4	Leistungsbewertung der GSM Signalisierungsprotokolle	110
7.4.1	Überblick	110
7.4.2	Realisierung	113
7.4.3	Ergebnisse der Simulation formal spezifizierter GSM-Protokolle mit SPEET und Bewertung	116
8	Der SPEET Hardware-Emulator	122
8.1	Kenngrößen zur Leistungsbewertung von Prozessorsystemen	122
8.2	Konzept und Realisierung des SPEET HardwareEmulators	124
8.3	Ergebnisse für ein formal spezifiziertes Dateitransferprotokoll	128
8.4	Vergleichsmessungen	135
8.5	Weiterentwicklung des SPEET Hardware-Emulators	136
8.6	Meßumgebung zur Validierung des erweiterten Emulatorkonzepts	138
8.7	Meßergebnisse für den erweiterten Hardware-Emulator	142
9	Zusammenfassung und Ausblick	147
	Quellenverzeichnis	150
	Nachwort	161

KAPITEL 1

Einleitung

Der technologische Fortschritt der letzten Jahrzehnte hat eine Entwicklung zur Informationsgesellschaft bewirkt. Eine ständig wachsende Anzahl von Informationsangeboten und -diensten kennzeichnet veränderte Formen der Arbeit und Kommunikation im geschäftlichen und privaten Bereich. Telekommunikationsnetze werden in zunehmendem Maße zur Übermittlung und Verbreitung von Informationen, Durchführung von Transaktionen usw. eingesetzt.

Dieser Trend wird sich in Europa durch die Liberalisierung des Telekommunikationsmonopols zu Beginn des Jahres 1998 verstärkt fortsetzen. Private Anbieter werden dann in Konkurrenz zu existierenden Netzbetreibern mobile und multimediale Kommunikationsdienste verfügbar machen. Bestehende Netzinfrastrukturen dieser neuen Betreiber sollen vorwiegend durch (breitbandige) Mobilfunknetze ergänzt werden, um eine Flächendeckung z.B. im Ortsnetzbereich schnell und effizient zu erreichen. Entsprechend werden Entwurf und Entwicklung derartiger Mobilfunksysteme augenblicklich stark vorangetrieben.

Analog zu den meisten technisch hochwertigen Produkten ist Software ein wesentlicher Bestandteil von Mobilfunknetzen. Betrachtet man die digitale Vermittlungstechnik, so werden ca. 80 % der Entwicklungskosten für die Erstellung von Software aufgewendet. Aufgrund der immer kürzer werdenden Entwicklungszyklen für neue Produkte erweist sich die langwierige und kostenintensive Entwicklung von Softwaresystemen zunehmend als Hinderungsgrund für eine zeitgerechte Verfügbarkeit marktfähiger Produkte. Software, insbesondere mit hohem Komplexitätsgrad, kann nur dann effizient entwickelt werden, wenn die zugehörigen Methoden und Werkzeuge einen entsprechenden Entwicklungsstand aufweisen.

Diese Arbeit leistet zu den angesprochenen Themenbereichen Entwurf eines funkbasierten Zugangsnetzes und Entwicklungsmethodik zugehöriger Software einen Beitrag.

Als Basissystem wird ein völlig dezentral organisiertes Mobilfunknetz mit Teil- oder Vollvermaschung zugrunde gelegt; derartige Systeme werden auch

als Multi-Hop Funknetze bezeichnet.

Die entwickelten Verfahren und Protokolle sind im wesentlichen durch folgende Merkmale gekennzeichnet:

- Die Übertragungskapazität des Funknetzes wird adaptiv für verschiedene Dienstarten entsprechend den aktuellen Teilnehmeranforderungen verfügbar gemacht.
- Das Verfahren des Asynchronen Transfer Modus (ATM) zur Paketdatenübertragung wird für den Einsatz in dezentral organisierten Funknetzen um alle wesentlichen Bestandteile ergänzt. Die Erweiterungen betreffen unter anderem die Berücksichtigung spezieller Eigenschaften des Funkkanals (z.B. höhere Bitfehlerwahrscheinlichkeit) durch Integration geeigneter Schutz- und Korrekturmaßnahmen, die dienstgütespezifische, adaptive Zuteilung von Übertragungskapazität für parallel existierende virtuelle Verbindungen, die Einführung dienstspezifischer Übertragungsprioritäten und die Realisierung von Verkehrsflußsteuermechanismen.
- Kanalvermittelte und virtuelle Verbindungen können parallel zum verbindungslosen Kommunikationsmodus betrieben werden.
- Einrichtung, Betrieb und Auslösung kanalvermittelter und virtueller Verbindungen werden so realisiert, daß durch Schutzzonenausbildung der Störeinfluß versteckter Stationen in teilvermaschten Netzen deutlich reduziert wird.

Zur Leistungsbewertung wurde ein Simulationsprogramm mit der Bezeichnung *Radio Metropolitan Area Network Simulator* (RMAN-S) entwickelt und alle angeführten Verfahren und Protokolle vollständig implementiert. Für typische Szenarien und Anwendungsfälle wurde die Leistungsfähigkeit des Netzes anhand der interessierenden Kenngrößen bewertet und die Ergebnisse miteinander verglichen.

Die zugehörige Softwareentwicklungsmethodik zur Implementierung wurde ausgehend von einer werkzeuggestützten Systementwicklung auf Basis einer formalen Spezifikationssprache hergeleitet. Die grundlegende Idee besteht darin, formale Spezifikationen nicht nur zur Beschreibung von Systemen zu verwenden, sondern mittels stochastischer Simulation sowohl zur Leistungsbewertung als auch zur Prädiktion der erforderlichen Leistungsfähigkeit der Endsystemhardware heranzuziehen. Im Gegensatz zu üblichen Anwendungen der stochastischen Simulation werden hier die realen (Teil-) Systeme und nicht ihre Modelle für die simulative Leistungsbewertung verwendet. Die-

se Vorgehensweise erfordert ein flexibles Entwicklungswerkzeug zur Umgebungssimulation, das darüber hinaus in der Lage ist, Hardwareplattformen mit hoher Genauigkeit nachzubilden. Zu diesem Zweck wurde die Simulationsumgebung *SDT Performance Evaluation Toolkit* (SPEET) entwickelt. Die Validierung des SPEET-Simulators erfolgte durch Untersuchungen an formal spezifizierten Kommunikationsprotokollen und parallele Vergleichsmessungen an existierenden Kommunikationssystemen.

Gliederung der Arbeit

In **Kapitel 2** werden kurz die Charakteristika zentral und dezentral organisierter Mobilfunknetze besprochen, an die sich eine Darstellung technischer Merkmale verschiedener, derzeit in der Entwicklung befindlicher, dezentral organisierter Funknetze anschließt.

Kapitel 3 führt zunächst allgemein in den Bereich formaler Spezifikationssprachen ein. Der Schwerpunkt liegt auf der Sprache SDL (Specification and Description Language), die für die Spezifikation von Telekommunikationssystemen besonders geeignet ist. Alle relevanten Standards werden im Überblick dargestellt, es folgt eine detaillierte Beschreibung der aktuellen Sprachversion SDL-92.

Die neu entwickelten Verfahren und Protokolle für das ATM-basierte Mobilfunknetz RMAN sind umfassend in **Kapitel 4** dargestellt.

Kapitel 5 beschreibt detailliert Struktur und Komponenten des RMAN-Simulators zur Leistungsbewertung.

In **Kapitel 6** werden zunächst Kenngrößen zur Charakterisierung von Multi-Hop Mobilfunknetzen definiert, sowie die verwendeten Simulationsszenarien und -parameter vorgestellt. Die ermittelten Simulationsergebnisse werden anschließend einer vergleichenden Bewertung unterzogen.

Die Softwareentwicklungsmethodik zur Realisierung des RMAN-Mobilfunknetzes basierend auf formalen Spezifikationen wird in **Kapitel 7** hergeleitet, es folgt eine ausführliche Beschreibung der ersten Entwicklungsstufe der SPEET-Simulationsumgebung.

Kapitel 8 enthält eine detaillierte Darstellung der SPEET-Komponenten zur Prädiktion der Verarbeitungsleistung eines Zielsystems sowie eine Validierung des Simulators anhand von Vergleichsmessungen an (formal spezifizierten) Protokollen realer Kommunikationssysteme.

Die Arbeit schließt mit einer Zusammenfassung und einem Ausblick in **Kapitel 9**.

Mobilfunknetze

2.1 Einleitung

Durch den EU-Ministerrat für Telekommunikation wurde 1993/94 eine Öffnung der EU-Telekommunikationsnetze für den Wettbewerb zum 1. Januar 1998 beschlossen, die den Wegfall des Dienstmonopols einschließt. Dadurch werden private Anbieter in die Lage versetzt, im Wettbewerb mit derzeitigen Netzbetreibern Telekommunikationsnetze und -dienste, auch Sprachdienste, (flächendeckend) verfügbar zu machen. Insbesondere für Unternehmen, die bereits eine grundlegende Netzinfrastruktur besitzen, z.B. Energieversorgungsunternehmen, Deutsche Bahn AG, eröffnet sich ein neues, interessantes Geschäftsfeld.

Nachteilig für diese potentiellen Telekommunikationsanbieter ist die nicht vorhandene Netzinfrastruktur für den Anschluß von Endteilnehmern (Kunden). Lösungsansätze basieren auf dem Einsatz von Funknetzen zur Überbrückung der Distanz zwischen örtlicher Zugangsstation (z.B. Bahnhof, Trafostation) und Endteilnehmer, vgl. Abb. 2.1. Derartige Anwendungen werden in den nächsten Jahren zu einem großen Bedarf an Funknetzen führen, wie sie in Abschn. 2.4-2.6 beschrieben und im Rahmen dieser Arbeit untersucht werden. Existierende und zukünftige (Mobil-) Funknetze lassen sich nach verschiedensten Gesichtspunkten kategorisieren, wie z.B.

- technische Netzmerkmale: analoge oder digitale Übertragungstechnik, verwendete Frequenzbereiche, verfügbare Übertragungsraten,
- geographische Ausdehnung (Reichweite) von Netzen: indoor, regional, national, kontinental, interkontinental,
- Dienstangebot im Netz: Sprachdienste, schmalbandige Datendienste (Fax, E-Mail), Breitbanddienste (Video, Tele-Education usw.).

Eine tabellarische Zusammenstellung von Funknetzen nach verschiedensten Gesichtspunkten enthält [104]. Im folgenden wird eine Unterscheidung nach zentral und dezentral organisierten Funknetzen vorgenommen.

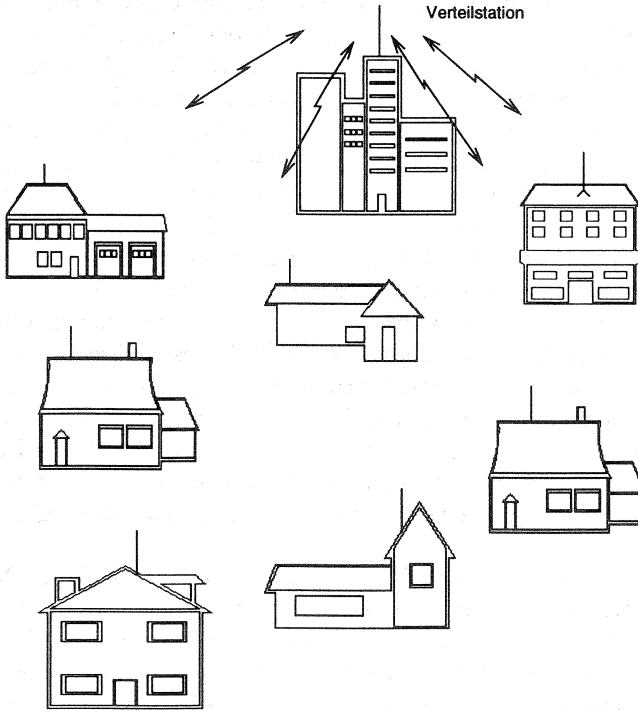


Abbildung 2.1: Funkbasiertes Ortsnetz (Radio Local Loop, RLL)

2.2 Zentral organisierte Mobilfunknetze

Charakteristisch für zentral organisierte Mobilfunknetze ist das Vorhandensein von Funkfeststationen, die jeweils eine Anzahl von mobilen Teilnehmern in einem räumlich begrenzten Gebiet versorgen und über Festnetze mit anderen Funkfeststationen, Stationen der nächsthöheren Hierarchieebene (z.B. Vermittlungsstationen) oder anderen Netzen verbunden sind. Bei Mobilfunknetzen der ersten Generation wurde eine hohe Sendeleistung zur Versorgung eines möglichst großen Gebietes verwendet. Dies führt bei hohen Teilnehmerdichten und Vermeidung von Interferenzen mit benachbarten Versor-

gungsgebieten zu einem erheblichen Bedarf an Funkressourcen, die aufgrund der geringen Vakanzen im gesamten Frequenzband meist nicht verfügbar gemacht werden können. Bei neueren, sog. zellularen Mobilfunknetzen wird ein gesamtes Netzversorgungsgebiet in Funkzellen eingeteilt, wobei jede Zelle von einer Funkfeststation mit geringerer Sendeleistung versorgt wird. Eine Funkfeststation erhält nur einen Teil der dem gesamten Netz zugewiesenen Frequenzkanäle, die übrigen werden in benachbarten Zellen verwendet. Eine Anordnung von benachbarten Zellen, die disjunkte Frequenzkanäle verwenden, wird als Cluster bezeichnet. Die Flächendeckung des zellularen Mobilfunknetzes wird durch Aneinanderfügen von Clustern erreicht, wodurch eine geplante, räumliche Wiederverwendung von Frequenzkanälen erzielt wird. Wird eine Funkzelle idealerweise als regelmäßiges Sechseck angenommen, so sind nur bestimmte Clusteranordnungen von z.B. 3, 4, 7, 12 oder 21 Funkzellen möglich. Die endgültige Dimensionierung eines Clusters (Anzahl der enthaltenen Funkzellen, Größe der Funkzellen usw.) ist von Randbedingungen wie z.B. dem verwendeten Modulationsverfahren, der technischen Ausführung des Empfängers, der Sendeleistung usw. abhängig, vgl. auch [15].

GSM¹-Netze (in Deutschland D1, D2 und E1-Netz), die NMT² Netze oder das zukünftige mobile Breitbandnetz MBS³ sind Beispiele für zellulare Mobilfunknetze. Ein wesentliches Merkmal der oben beschriebenen Netze ist, daß Teilnehmerfunkstationen nur Verbindungen mit einer Funkfeststation einrichten können, eine Einrichtung direkter Funkverbindungen zwischen Teilnehmerstationen oder die Funktion einer Teilnehmerstation als Vermittlungsstation ist nicht möglich. Diese Systeme werden auch als One-Hop Funknetze bezeichnet.

2.3 Dezentral organisierte Mobilfunknetze

Dezentral organisierte Mobilfunknetze werden durch (mobile) Funkstationen aufgespannt, die entweder Bestandteil des Teilnehmerendgerätes sind, oder an die Teilnehmerendgeräte direkt angeschlossen sind. Neben der Bereitstellung des Teilnehmernetzzugangs verfügen die Funkstationen über Protokolle/Funktionen zur dezentralen Abwicklung aller Organisationsauf-

¹Global System for Mobile Communication

²Nordic Mobile Telephone

³Mobile Broadband System

gaben wie Ermittlung, Verwaltung und Aktualisierung netz- und teilnehmerspezifischer Informationen. Besitzen die Funkstationen darüber hinaus Vermittlungsfunktionen, so können Verbindungen zwischen Teilnehmern, deren Stationen keinen direkten Funkkontakt besitzen, über als Relais arbeitende Stationen eingerichtet werden. Abb. 2.2 zeigt den Ausschnitt eines ortsfesten, dezentral organisierten Funknetzes, bei dem Verbindungen zwischen Teilnehmern an den Stationen A und E über Relaisstationen B, C und D geführt werden. Dezentral organisierte Funknetze, die Teilnehmerverbindungen

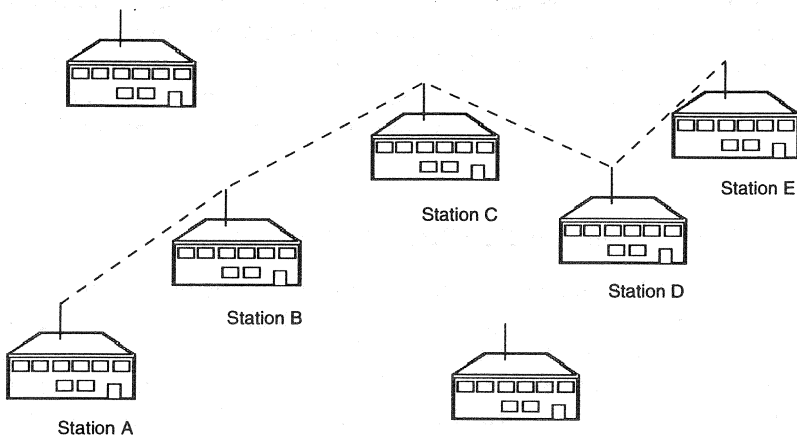


Abbildung 2.2: Multi-Hop Funknetz

ungen zwischen Stationen ohne direkten Funkkontakt durch Einrichten von N Teilstreckenfunkverbindungen ($N \geq 1$) realisieren können, werden als Multi-Hop Funknetze bezeichnet. Multi-Hop Funknetze eignen sich insbesondere für den Einsatz in Bereichen, in denen eine hohe Flexibilität der Netzkonfiguration erforderlich ist (z.B. temporäre lokale Netze bei Konferenzen, Arbeitstreffen), die Netztopologie aufgrund hoher Teilnehmermobilität ständigen Änderungen unterliegt (Fahrzeug-Fahrzeug Kommunikation), Funknetze anstatt terrestrischer Netze für den Anschluß von Endteilnehmern eingesetzt werden und Endteilnehmerstationen typischerweise

nicht direkt über Funkverbindungen mit einer Zugangsstation des Netzbetreibers verbunden werden können, vgl. Abschn. 2.1. Im folgenden werden derzeit in der Entwicklung befindliche dezentral organisierte Multi-Hop Funknetze beschrieben.

2.4 HIPERLAN

Unter der Bezeichnung HIPERLAN (High Performance Radio Local Area Network) wird von der ETSI⁴-Fachgruppe RES 10 (Radio Equipment and Systems) ein Nahbereichs-Funknetz standardisiert⁵, das auf die funkbasierte, lokale Vernetzung von Computersystemen als Alternative zu drahtgebundenen lokalen Netzen abzielt. HIPERLAN-Systeme sollen im Bereich Büro- und industrieller Automatisierung, Lehre und Ausbildung, Aufbau von Ad-Hoc Netzen (bei Konferenzen, Arbeitstreffen) usw. eingesetzt werden, vgl. [28].

2.4.1 Kenngrößen von HIPERLAN-Netzen

HIPERLAN-Funkstationen werden in Verbindung mit tragbaren Rechner-Systemen, insbesondere Laptops, eingesetzt, weshalb eine Realisierung, ohne Antenne, als PCMCIA (Personal Computer Memory Card Interface Association) Typ III Adapter mit einer Leistungsaufnahme von wenigen hundert Milliwatt angestrebt wird. Eine Mobilität der Teilnehmerstation von bis zu 36 km/h und einer achsialen Drehgeschwindigkeit von 360°/s wird unterstützt.

Für HIPERLAN-Netze ist ein Frequenzbereich von 5,15 GHz – 5,30 GHz (zukünftig auch 17,1 GHz -17,3 GHz) reserviert, der in fünf Frequenzkanäle mit Trägerfrequenzen von $5,17648 \text{ GHz} + n \cdot 235,294 \text{ MHz}$ ($0 \leq n \leq 4$) aufgeteilt ist. Verschiedene HIPERLANs können somit parallel zu einander existieren, jedes HIPERLAN-Teilnetz identifiziert sich durch einen eindeutigen HIPERLAN-Identifikator (HID), jede Station innerhalb eines Teilnetzes durch einen eindeutigen Knoten-Identifikator (Node Identifier, NID). Je Netz wird ein Funkkanal benutzt, auf den die Stationen im Zeitvielfach zugreifen.

⁴European Telecommunication Standards Institute

⁵Das gleiche Netz wird durch das Institute of Electrical and Electronics Engineers (IEEE) unter der Bezeichnung Wireless Local Area Network (WLAN) standardisiert.

Zwei HIPERLAN-Teilnetze A und B werden als unabhängig voneinander betrachtet, sofern keine Station des Teilnetzes A in der Übertragsreichweite einer Station des Teilnetzes B ist, und umgekehrt. Wird ein Kontakt zwischen zwei oder mehreren unabhängigen Teilnetzen hergestellt, z.B. aufgrund der Stationsmobilität, so können die Stationen aller Teilnetze mittels des Intra-HIPERLAN Forwarding Mechanismus miteinander kommunizieren. Auftretende Konflikte bezüglich der Eindeutigkeit der Netzidentifikatoren (HID) werden durch die HIPERLAN Sharing Function beseitigt.

In HIPERLAN-Netzen wird eine Gesamtdatenrate von 20 Mbit/s bzw. 1 Mbit/s für asynchrone Datenverkehre verfügbar sein, wobei zwei Funkstationen maximal 50 m bzw. 800 m voneinander entfernt sein dürfen. Der minimale Abstand zwischen zwei Netzstationen darf 0,5 m betragen. Verbindungsorientierte, synchrone Dienste mit einer Gesamtübertragungsrate von 64 kbit/s bis höchstens 2048 kbit/s sollen pro Trägerfrequenz unterstützt werden.

Die Latenzzeit des Kanalzugriffs soll für asynchrone Datendienste bei einer Netzbelastung bis zu 30 % weniger als 1 ms betragen, bei verbindungsorientierten Diensten soll sie kleiner als 3 s sein. Die Varianz der Übertragungsverzögerung ist bei verbindungsorientierten Diensten auf 9 (ms)^2 begrenzt.

Bei maximaler Reichweite und ohne Interferenzen wird eine entdeckte Verlust-/Fehlerrate für Medium Access Control (MAC) Protokolldateneinheiten der Sicherungsschicht von maximal 10^{-3} gefordert, die Restfehlerrate für MAC-Protokolldateneinheiten soll unterhalb von $8 \cdot 10^{-8}$ pro Oktett, bei MAC-Dienst dateneinheiten unterhalb von $5 \cdot 10^{-14}$ pro Oktett liegen.

2.4.2 HIPERLAN-Referenzmodell

Der HIPERLAN-Standard wird eine Spezifikation der ISO/OSI Schichten 1 (Bitübertragungsschicht) und 2 (Sicherungsschicht) enthalten, Festlegungen und Realisierungen bezüglich höherer OSI-Schichten verbleiben in der Verantwortlichkeit der Hersteller/Betreiber.

Für die Systembeschreibung wurde in Anlehnung an das ISO/OSI Referenzmodell ein HIPERLAN-Referenzmodell entworfen, das eine weitere Unterteilung der OSI MAC-Teilschicht (Schicht 2a) in eine HIPERLAN MAC-Teilschicht und eine Teilschicht für den Kanalzugriff (Channel Access Control, CAC) vornimmt, vgl. Abb. 2.3.

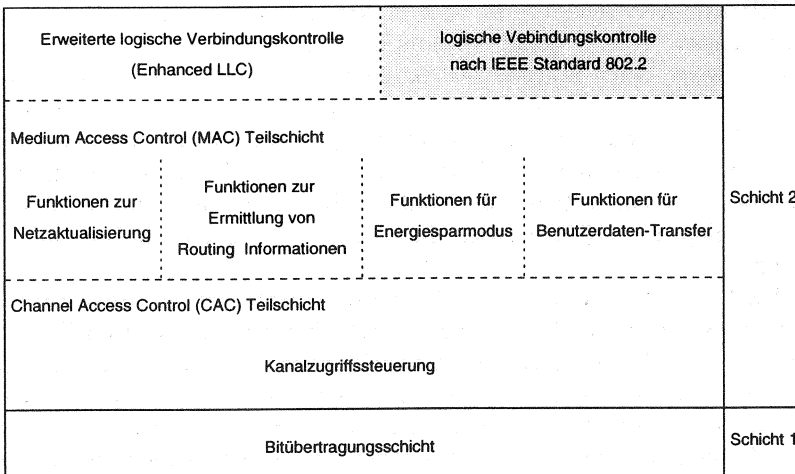


Abbildung 2.3: HIPERLAN-Referenzmodell

Im Rahmen der HIPERLAN-Standardisierung soll in Anlehnung an den IEEE 802.2 Standard [47] ein erweitertes Sicherungsschichtprotokoll für die logische Verbindungssteuerung (Logical Link Control, LLC) spezifiziert werden, das der Netzschicht verbindungsorientierte Dienste verfügbar macht. Für asynchrone Datendienste wird das LLC-Protokoll gemäß IEEE 802.2 eingesetzt.

Die HIPERLAN MAC-Teilschicht ist in vier Funktionsbereiche gegliedert: Die Lookup-Prozeduren ermöglichen die Festlegung einer HIPERLAN-Kommunikationsumgebung durch Funktionen für das Einrichten/Auslösen eines HIPERLANs, die An- und Abmeldung von Teilnehmerstationen, Modifikation der Verschlüsselungstechnik, Ermittlung der Knotenkonfiguration usw.

Für die Kommunikation zwischen Stationen über Funkbereichsgrenzen hinaus müssen stationslokal Wegelenkungsinformationen (Routing Information) verfügbar sein. Die Ermittlung/Aktualisierung von Routing Informationen erfolgt durch Senden/Empfang sog. Hello-Pakete, deren Informationen in einer stationsinternen Datenbank, der Routing Information Base (RIB),

gespeichert werden. Für die Weiterleitung von Datenpaketen werden je nach Kenntnisstand der Relaisstation unterschiedliche Verfahren angewendet: Ist die Adresse der nächsten Relaisstation in Richtung der Zielstation bekannt, wird das betreffende Datenpaket an diese Relaisstation gesendet (Unicast Relaying), im anderen Fall wird das Datenpaket durch Rundsendung an alle empfangenden Nachbarstationen weitergeleitet (Broadcast Relaying).

Zur Unterstützung batteriebetriebener HIPERLAN-Systeme ist ein optionaler Energiesparmodus (Power Conservation Function) definiert, der inaktive Endgeräte in festgelegten Zeitintervallen aktiviert.

Die MAC-Teilschicht stellt der überlagerten Verbindungssteuerungs-Teilschicht einen verbindungslosen, unbestätigten Datenübertragungsdienst zur Verfügung; eine Dienstanforderung enthält neben Quell-/Zieladresse und Daten, die Festlegung einer (binären) Priorität und die Angabe eines Zeitintervalls, innerhalb dessen diese Daten gesendet werden sollen.

Der Zugriff auf den Funkkanal wird durch die CAC-Teilschicht gesteuert. Ein Kanalzugriff kann ab einer festgelegten Zeitspanne, dem Channel Access Cycle, nach Beendigung einer laufenden Übertragung erfolgen, oder wenn der Funkkanal über einen längeren Zeitraum als frei detektiert wird. Ein Kanal- zugriffszyklus gliedert sich in drei Phasen:

1. innerhalb der Prioritätsphase (Priorization Phase) vergleichen alle sendewilligen Stationen ihre Kanalzugriffspriorität, nur Stationen mit der höchsten Priorität konkurrieren weiter um das Kanalzugriffsrecht.
2. durch eine Konkurrenzphase erhalten diejenigen Stationen das Datenübertragungsrecht, die die kürzeste (quasi-zufällige) Wartezeit für einen Kanalzugriff innerhalb dieser Phase ermittelt haben.
3. Datenübertragung und Bestätigung durch den Empfänger.

Eine detaillierte Spezifikation des CAC-Teilschichtprotokolls *Elimination Yield-Non Preemptive Priority Multiple Access (EY-NPMA)* enthält [30].

2.4.3 HIPERLAN-Bitübertragungsschicht

Datenpakete und Empfangsbestätigungen für Datenpakete werden von der Bitübertragungsschicht unterschiedlich moduliert und mit verschiedenen Datenraten übertragen. Für Datenpakete wird das Gaussian Minimum Shift Keying (GMSK) Modulationsverfahren verwendet, die Übertragungsrate beträgt 23,5294 Mbit/s mit einer maximalen Abweichung von ± 235 bit/s. Empfangsbestätigungen werden mit einer Rate von 1,40706 Mbit/s (± 15

bit/s) übertragen; jedes zu übertragende Bit wird mittels des Frequenzmodulationsverfahrens in ein Signal der Frequenz $f_c +$ bzw. $- 368$ kHz (Bitwert = 1 bzw. 0) transformiert, wobei f_c die Mittenfrequenz des Funkkanals ist, vgl. Abschn. 2.4.1.

In HIPERLAN-Netzen beträgt die maximal zulässige Sendeleistung 1 Watt. Die Empfängersensibilität beträgt je nach Empfängerklasse zwischen -50 dBm und -70 dBm, die Sendeleistung je nach Senderklasse zwischen 10 und 30 dBm, wobei nicht jede Kombination von Sender- und Empfängerklasse zulässig ist, vgl. [30].

2.5 DECT

Durch das European Telecommunication Standards Institute (ETSI) wurde unter der Bezeichnung *Digital Enhanced Cordless Telecommunication*, *DECT* (früher: Digital European Cordless Telecommunication) zunächst ein zentral organisiertes Mobilfunksystem für den Betrieb innerhalb von Gebäuden bzw. auf privaten Grundstücken standardisiert. Eine weitere Empfehlung der ETSI, vgl. [31], spezifiziert Funktionen sog. DECT-Relais (Wireless Base Station, WBS), wodurch der Einsatz von DECT-Systemen z.B. als funkbasiertes Zugangsnetz, vgl. Abschn. 2.1, mit dezentraler Organisation ermöglicht wird, vgl. Abschn. 2.5.3.

Ein lokales DECT-Netz besteht aus einer oder mehreren Feststationen und zugehörigen mobilen Teilnehmerendgeräten, wobei die Feststationen typischerweise an ein lokales Netz (z.B. ISDN⁶, X.25, GSM) angeschlossen sind. Das funktionale DECT-Referenzsystem, vgl. [22], sieht eine Verbindung der lokalen Netze durch überregionale (nationale oder internationale) Netze vor. Lokale DECT-Netze weisen eine mikrozellulare Struktur auf, direkte Verbindungen zwischen mobilen Endgeräten ohne Beteiligung einer Feststation können nicht eingerichtet werden.

2.5.1 Eigenschaften und Kenngrößen von DECT-Systemen

Für den Betrieb von DECT-Systemen ist das Frequenzband im Bereich von 1880–1900 MHz reserviert, das linear in zehn Teilbänder unterteilt wird. Die zugehörigen Trägerfrequenzen f_c im Abstand von 1782 kHz ergeben sich

⁶Integrated Services Digital Network

entsprechend folgender Gleichung:

$$f_c = f_0 - c \cdot 1782 \text{ kHz} \quad \text{mit} \quad c = 0, 1, 2, \dots, 9 \quad \text{und} \quad f_0 = 1897,344 \text{ MHz}$$

Die zulässige Abweichung von einer Trägerfrequenz während des Betriebs beträgt $\pm 50 \text{ kHz}$.

Als Modulationsverfahren wird entweder ein GFSK-Verfahren (Gaussian Frequency Shift Keying) mit einem Bandbreite-Zeitprodukt von $B \cdot T = 0,5$ oder das GMSK-Verfahren (Gaussian Minimum Shift Keying) verwendet. Die Übertragung einer binären Eins erfolgt mit einer um $\Delta f = 288 \text{ kHz}$ erhöhten Frequenz ($f_c + \Delta f$), eine binäre Null wird mit einer um $\Delta f = 288 \text{ kHz}$ erniedrigten Frequenz ($f_c - \Delta f$) gesendet.

Physikalische Kanäle werden weiterhin durch Zeitmultiplex (TDM⁷) je Trägerfrequenz erzeugt, der zugehörige TDM-Rahmen mit einer Dauer von 10 ms ist in 24 Zeitschlitzte eingeteilt, vgl. Abb. 2.4.

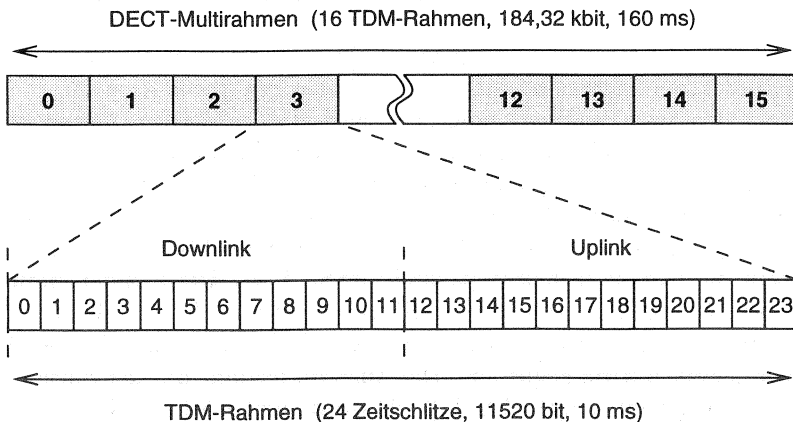


Abbildung 2.4: DECT-Multirahmen und DECT-TDM-Rahmen

16 TDM-Rahmen bilden eine weitere logische Einheit, den DECT-Multirahmen. Je Zeitschlitz können 480 bit übertragen werden, woraus sich eine Bruttoübertragungsrate von 1152 kbit/s pro Trägerfrequenz ergibt.

⁷Time Division Multiplex

Die Nutzung des TDM-Rahmens erfolgt nach dem Zeitgetrenntlageverfahren (Time Division Duplexing, TDD), für Übertragungen von der Feststation zu den Mobilstationen (Downlink) werden die Zeitschlitz 0–11, in Richtung der Mobilstationen zur Feststation (Uplink) die Zeitschlitz 12–23 verwendet, vgl. Abb. 2.4. Zwei Zeitschlitz eines TDM-Rahmens werden für eine Duplex-Verbindung zwischen Fest- und einer Mobilstation benötigt, der DECT-Standard schreibt hierfür die Nutzung eines Zeitschlitzpaares $(k, k+12)$ mit $0 \leq k \leq 11$ fest. Jeder Kanal hat eine Kapazität von 32 kbit/s; bei höherer erforderlicher Übertragungsrate können $n \leq 12$ Kanäle derselben Verbindung zugeordnet werden.

Bei DECT-Systemen wird ein dynamisches Kanalwahlverfahren (Dynamic Channel Selection, DCS) verwendet, wodurch in jeder Mikrozelle grundsätzlich das gesamte Kanalspektrum zur Verfügung steht. Eine sorgfältige Zellplanung und Kanaluweisung anhand eines festen Schemas, wie dies bei anderen zellularen Mobilfunksystemen (z.B. GSM) erforderlich ist, entfällt, die Netzplanung betrifft im wesentlichen die Standortplanung für Feststationen.

Eine nicht synchronisierte Mobilstation (bei Neuzugang oder Verlust des Kontaktes zur Feststation) ordnet sich derjenigen Feststation zu, die sie mit dem stärksten Pegel empfängt; erst nach drei erfolglosen Verbindungseinrichtungen wechselt sie zur nächst stärkeren Feststation.

Kanäle für die Verbindungseinrichtung wählt die Mobilstation anhand einer von ihr geführten Liste, die auf Pegelmessungen basiert. Die Kanalliste vermerkt für jeden Kanal den RSSI-Pegel⁸ (Radio Signal Strength Indicator) und teilt die Kanäle in belegt (typisch: $\text{RSSI} \geq -33 \text{ dBm}$), möglicherweise belegbar ($-33 \text{ dBm} > \text{RSSI} > -93 \text{ dBm}$) und frei ($\text{RSSI} \leq -93 \text{ dBm}$) ein. Der Bereich möglicherweise belegbarer Kanäle ist weiterhin in Abstufungen von jeweils 6 dBm gegliedert. Die Auswahl beginnt mit der Suche nach einem freien Kanal (zugeh. $\text{RSSI} \leq -93 \text{ dBm}$) und wird bei Nichterfolg in den Stufen mit höheren Signalpegeln fortgesetzt, bis ein Kanal für die Verbindungseinrichtung gefunden oder die Feststation als ausgelastet (busy) markiert wird.

Für eine bestehende Verbindung kann bei geringer Qualität ein Handover, d.h. Kanalwechsel (sog. Intra-Cell Handover) oder Wechsel zu einer anderen Feststation (Inter-Cell Handover), durchgeführt werden. Bei DECT-

⁸Entspricht der auf 1 mW bezogenen Signalleistung, die auf einem Kanal gemessen wurde.

Systemen wird der Handover dezentral durch die Mobilstation (Mobile Controlled Handover, MCHO) gesteuert. Bezüglich der logischen Handover-Merkmale wird zwischen einem Handover innerhalb eines geschlossenen DECT-Systems (Internal Handover) und einem Wechsel zwischen zwei unabhängigen DECT-Systemen (External Handover) unterschieden. Ist ein neuer Kanal zum Zeitpunkt eines Handovers bereits allokiert, so bezeichnet man diesen Wechsel als *Seamless Handover*, sonst als *Non-Seamless Handover*, bei dem es zum kurzzeitigen Dienstverlust kommen kann.

2.5.2 DECT-Referenzmodell

Abb. 2.5 zeigt das DECT-Referenzmodell und seinen Bezug zur Schichtenstruktur des ISO/OSI Referenzmodells. Die DECT-Standardisierung

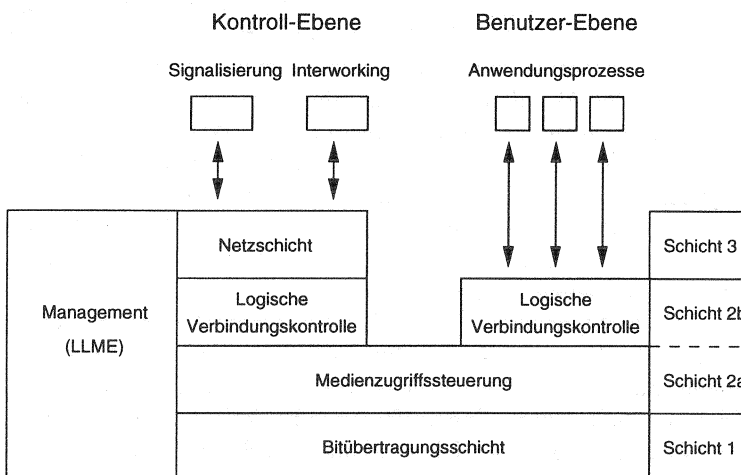


Abbildung 2.5: DECT-Referenzmodell

umfaßt die Funktionen und Protokolle der unteren drei OSI-Schichten. Oberhalb der Medienzugriffsschicht (Schicht 2a) gliedert sich das DECT-Referenzmodell, wie bei diensteintegrierenden digitalen Netzen (z.B. ISDN, GSM) üblich, in eine Kontrollebene (Signalisierung und Interworking) und

eine Anwendungsebene (Benutzer–Applikationen). In Übereinstimmung mit dem OSI-Modell ist dem DECT–Protokollstapel ein schichtenübergreifendes Managementsystem (Lower Layer Management Entity, LLME) zugeordnet. Die *Bitübertragungsschicht* ist für die Belegung des Funkmediums mit Übertragungskanälen entsprechend dem in Abschn. 2.5.1 dargestellten Frequenz– und Zeitvielfach verantwortlich. Die Beschränkung auf eine max. Sendeleistung von 250 mW unterstützt darüber hinaus die räumliche Wiederverwendung des DECT–Frequenzbandes (Space Division Multiplex, SDM). Aufgrund der geforderten Bitfehlerwahrscheinlichkeit von 0,001 besitzt ein DECT–Empfänger eine Empfindlichkeit von -83 dBm ($60 \text{ dB}\mu\text{V/m}$), die derzeit auf -86 dBm umdefiniert wird.

Das *Medienzugriffsprotokoll* (MAC⁹–Protokoll) ist für Einrichtung, Betrieb und Auslösung von Verkehrskanälen verantwortlich. Zur Fehlererkennung und –korrektur werden die MAC–Protokolldateneinheiten mit zyklischen Codes geschützt. Der überlagerten Teilschicht (Schicht 2b) werden folgende Dienste zur Verfügung gestellt: Ein Nachrichtenverteildienst, der beispielsweise von einer Feststation zur Übertragung systembezogener Daten und entsprechend von den Mobilstationen zur Identifikation der Feststation sowie zur Messung der Kanalqualität verwendet wird. Weiterhin wird ein verbindungsloser Dienst für die Punkt–zu–Punkt und Punkt–zu–Mehrpunkt Kommunikation, sowie ein verbindungsorientierter Dienst ausschließlich für die Punkt–zu–Punkt Kommunikation angeboten. Das DECT–MAC–Protokoll beinhaltet ebenfalls Funktionen für die Steuerung von Zellen und Gruppierungen von Zellen (Cluster).

Das *LLC–Teilschichtprotokoll* (Logical Link Control) der Kontrollebene macht der überlagerten Vermittlungsschicht neben Punkt–zu–Punkt Verbindungen auch einen Verteildienst verfügbar. Bezüglich der LLC–Protokolldateneinheiten wird eine umfassende Fehlerkontrolle durchgeführt, wodurch die Zuverlässigkeit der Übertragung von Signalisierungsdaten weiter verbessert wird.

Das LLC–Teilschichtprotokoll der Benutzerebene bietet einen Ende–zu–Ende Dienst zur Übertragung von Anwenderdaten an, das zugehörige Dienstgutespektrum reicht von der ungeschützten Datenübertragung mit geringer Verzögerung bis zu geschützten Diensten mit variabler Verzögerung. Einrichtung, Betrieb und Auslösung von Verbindungen zwischen Teilnehmer und Netz werden durch das *Vermittlungsschichtprotokoll* abgewickelt.

⁹Medium Access Control

Es übernimmt weiterhin die Kontrolle des Datenaustauschs und die Übermittlung von Authentifizierungs- und Verschlüsselungsdaten. Das Mobilitätsmanagement des Netzschnittprotokolls enthält alle Funktionen, die zur Unterstützung der Mobilität von Teilnehmerendgeräten benötigt werden. Das schichtübergreifende *Managementsystem (LLME)* steuert Einrichtung, Belegung und Auslösung von Verkehrskanälen, zu seinen Funktionen zählt weiterhin z.B. die Auswahl freier Kanäle, Führung der Kanalliste, die Qualitätsbewertung des Empfangssignals, vgl. Abschn. 2.5.1.

2.5.3 DECT-Relais

Der WBS-Standardentwurf, vgl. [31], führt Relaisysteme als weitere Komponente für DECT-Netze ein, die inner- und außerhalb von Gebäuden eingesetzt werden können. Im Außenbereich können sie z.B. als Komponenten drahtloser lokaler Netze (Radio Local Loop, RLL) verwendet werden, um Teilnehmer (Kunden) mit einer Verteilstation (Ortsvermittlungsstelle) zu verbinden, vgl. Abschn. 2.1. Es wird angestrebt, mobilen Teilnehmern einen öffentlichen Zugang zu solchen Netzen verfügbar zu machen, entsprechende herstellerunabhängige Zugangsschnittstellen für DECT-Systeme sind bereits standardisiert (Generic Access Profile, GAP und Public Access Profile, PAP).

Innerhalb von Gebäuden werden DECT-Relais als integrale Bestandteile des gesamten Systems eingesetzt, sie dienen beispielsweise zur Vernetzung von Montagehallen, großflächigen Firmengeländen oder zum Aufbau temporärer Netze. Die Vorteile des Einsatzes von funkbasierten Relais liegen vor allem in der möglichen Kosteneinsparung gegenüber einer zusätzlich benötigten Verkabelung, der flexiblen Rekonfigurierbarkeit des Netzes, den Möglichkeiten bestimmte Bereiche besser auszuleuchten usw. Konzeptionell wird zwischen ortsfesten (FRS)¹⁰ und mobilen (MRS)¹¹ DECT-Relais unterschieden. Bei FRS-Systemen handelt es sich um dedizierte Stationen, für die eine ortsfeste Platzierung angestrebt wird, ihre Funktion ist für ein mobiles Teilnehmerendgerät völlig transparent, d.h. sie übernehmen ausschließlich die Weiterleitung von Verbindungen zu einer netzgekoppelten Feststation. FRS-Stationen arbeiten als Schicht 2a Relais (Brücke) oder Repeater (Schicht 1 Relais). Verbindungen zwischen mobilen Teilnehmerendgeräten können prinzipiell über mehrere (sequentiell arbeitende) Relais

¹⁰Fixed Relais Station

¹¹Mobile Relais Station

geführt werden, die Dienstgüte derartiger Multi-Hop Verbindungen wird durch DECT-Systemparameter wie z.B. die TDM-Rahmendauer (10 ms), Verwendung des Zeitgetrenntlageverfahrens (feste Zuordnung von Up- und Downlink), mögliche Handover, stark begrenzt.

Mobile Relaisstationen können z.B. durch Modifikation von DECT-Teilnehmerendgeräten realisiert werden, direkte Verbindungen zwischen Teilnehmern ohne Einbeziehung einer DECT-Feststation sind auch hier nicht vorgesehen. Der Aufbau mobiler Relais erfordert wesentliche Erweiterungen des DECT-Protokollstapels, z.B. werden Funktionen zur Aktualisierung von Routing Tabellen, Begrenzung der max. Hopzahl einer Verbindung usw. benötigt.

2.6 Ein dezentral organisiertes Multi-Hop Funknetz für innen und außen

2.6.1 Eigenschaften des Multi-Hop Funknetzes

Das dezentral organisierte Multi-Hop Funknetz, welches in [71] und [70] konzeptionell beschrieben ist, wird im Rahmen der vorliegenden Arbeit bezüglich seiner Kommunikationsprotokolle entwickelt, untersucht und bewertet. Es ist für den Betrieb im freien Gelände, auf dem Wasser, in der Nähe oder innerhalb von Gebäuden vorgesehen. Die Gesamtausdehnung eines durch (mobile) Funkstationen aufgespannten Netzes beträgt typischerweise zwischen 0,5 und 20 km.

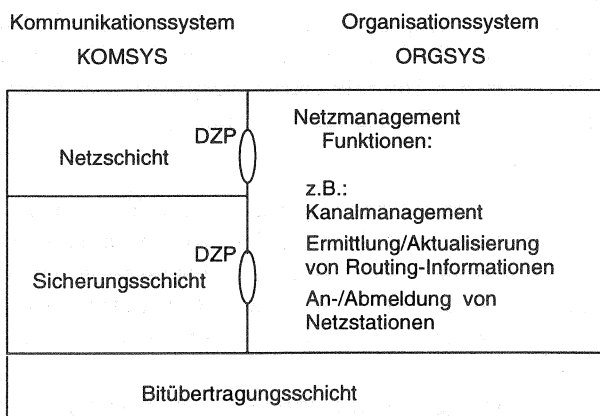
Die Funkstationen sind mit Rechenleistung (Mikroprozessorsystemen) ausgestattet, so daß alle für Organisation und Kommunikation des Netzes benötigten Protokolle völlig dezentral abgewickelt werden können. Je nach Einsatzbereich des Netzes werden Funkstationen mit rundstrahlenden oder bündelnden Sende- bzw. Empfangsantennen oder Kombinationen von beidem ausgerüstet sein.

Teilnehmerendgeräte sind entweder direkt oder über drahtgebundene (lokale) Netze mit den Funkstationen verbunden. Der Zugang zu bestehenden (Tele-) Kommunikationsnetzen (ISDN, GSM, Datex-P, Telefonnetz usw.) wird durch sog. Gateway-Funkstationen realisiert. In diesen Funknetzen werden alle Sprach- und Datendienste, eventuell mit reduzierter Dienstgüte, angeboten wie sie in drahtgebundenen Netzen verfügbar sind.

Wie die folgende Beschreibung zeigen wird, bestehen Ähnlichkeiten zum DECT-System, wenn Relaisstationen eingesetzt werden.

2.6.2 Referenzmodell des dezentral organisierten Multi-Hop Funknetzes

Wie bereits angesprochen, müssen alle Funkstationen eines dezentral organisierten Funknetzes über Netzmanagementfunktionen verfügen, um teilnehmer- und netzbezogene Informationen eigenständig ermitteln, verwalten und aktualisieren zu können. Für die Beschreibung des Funknetzes wurde eine logische Trennung der für Benutzerkommunikation und Organisation benötigten Protokollstapel vorgenommen. Abbildung 2.6 zeigt das zugehörige Referenzmodell, das eine Schichtung der benötigten Kommunikationsprotokolle entsprechend dem ISO/OSI Referenzmodell und eine logische Aufteilung der Sicherungs- und Netzschicht in ein Kommunikationssystem (KOMSYS) und ein Organisationssystem (ORGSYS) vornimmt.



DZP: Dienstzugangspunkt

Abbildung 2.6: Referenzmodell des dezentral organisierten Multi-Hop Funknetzes

Das Kommunikationssystem enthält alle Netzprotokolle für die Teilnehmer-zu-Teilnehmer Kommunikation (Sprach- und Datendienste). Bei ihrem Entwurf sind die speziellen Eigenschaften und Probleme von Multi-Hop Funknetzen zu berücksichtigen, vgl. Abschn. 4.1. Organisations- und Kommuni-

kationssystem tauschen Informationen über schichtinterne Dienstzugangspunkte (Service Access Point, SAP) aus.

Die Protokolle des Organisationssystems wickeln Aufgaben wie Ermittlung und Aktualisierung von Routing Informationen, An- und Abmeldung von Funkstationen beim/vom Netz, Verwaltung von Kanalbelegungstabellen und Teilnehmerverzeichnissen, Empfangspegelmessung und -auswertung usw. ab. Im folgenden werden beispielhaft zwei zugehörige Verfahren vorgestellt.

2.6.3 Verfahren zur Ermittlung und Aktualisierung von Routing Informationen

Funkstationen, die nicht direkt, sondern nur über Relaisfunkstationen miteinander kommunizieren können, benötigen aktuelle Routing Informationen zur Festlegung des Weges zwischen Quell- und Zielstation während der Signalisierungsphase (bei kanalvermittelter und verbindungsorientierter Kommunikation) oder der Datenübertragungsphase (bei verbindungsloser Kommunikation). Für die Festlegung der Route gibt es grundsätzlich zwei Möglichkeiten:

1. Die Quellstation sendet das Signalisier- bzw. Datenpaket unter Nutzung der Zieladresse und ihrer lokalen Routingtabelle an die nächste auf der Route zur Zielstation liegende Relaisstation. Diese sendet das Paket unter Zuhilfenahme ihrer lokalen Routingtabelle an die nächste Relaisstation in Zielrichtung usw. bis es schließlich von der Zielstation empfangen wird.
2. Die Quellstation legt anhand ihrer lokalen Routingtabelle die Route zur Zielstation durch Angabe aller Relaisstationen im Signalisier-/Datenpaket fest.

Beide Verfahren setzen voraus, daß die in den Routingtabellen enthaltenen Informationen aktualisiert werden, sofern netztopologische Veränderungen, z.B. durch Stationsmobilität oder An- und Abmeldung von Stationen, auftreten. Zur Aktualisierung stationslokaler Routingdaten können verschiedene Verfahren eingesetzt werden:

Bei *ereignisorientierten* Verfahren wird von einer Funkstation ein Aktualisierungspaket mit Routinginformationen per Rundsendung netzweit übertragen, sobald der Verlust/Gewinn eines Funkkontaktes festgestellt wird.

Bei *periodischen* Aktualisierungsverfahren überträgt jede Funkstation des Netzes in festgelegten Zeitintervallen per Rundsendung ihre lokal ermittelte Konnektivitätsmatrix, die Informationen über alle von ihr erreichbaren Nachbarstationen enthält. Die Länge des Zeitintervalls muß so gewählt werden, daß ein optimales Verhältnis zwischen erzeugtem Verkehrsaufkommen durch Übertragung von Aktualisierungsinformationen und benötigter Aktualität der stationslokalen Routing-Informationen erreicht wird.

Bei *asynchronen* Verfahren werden Informationen zur Aktualisierung von Routingtabellen mittels Huckepack- (piggy backed) Technik gemeinsam mit Benutzerdaten übertragen. Im Gegensatz zu ereignisorientierten Verfahren werden Aktualisierungsinformationen nicht direkt nach Feststellung einer topologischen Veränderung ausgetauscht, sondern wenn Benutzerdaten zur Übertragung anstehen.

Eine Beschreibung und Bewertung von Verfahren zur Aktualisierung stationslokaler Daten enthält z.B. [12, 13].

2.6.4 Verfahren für die An-/Abmeldung von Netzstationen

Die An-/Abmeldung von Netzstationen wird über Funkkanäle abgewickelt, die dem Organisationssystem exklusiv für Netzmanagementaufgaben zur Verfügung stehen. Bezüglich des Anmeldeverfahrens ist grundsätzlich zwischen zwei Fällen zu unterscheiden:

Einrichtung eines Netzes mit zwei Funkstationen

Es wird angenommen, daß eine Station A sich im Sende-/Empfangsbereich einer Station B befindet und sich um Aufnahme in ein Netz bemüht. Station A hört zunächst die für das Organisationssystem reservierten Kanäle ab, um Zugangsforderungen anderer Funkstationen zu empfangen. Stellt die Station A keine Sendungen anderer Funkstationen fest, so sendet sie ein Datenpaket mit Zutrittsforderung, welches die eigene Adresse und weitere stationsspezifische Informationen enthält, an alle Funkstationen (Rundsende-Paket). Nachdem Station B die Zutrittsforderung empfangen hat, sendet sie ein Datenpaket vom Typ 'Anmeldebestätigung', welches die eigene Adresse und weitere stationsspezifische Informationen enthält, an Station A, die den Erhalt der Anmeldebestätigung wiederum durch Senden eines Datenpakets vom Typ 'Anmeldebestätigung erhalten' bestätigt. Das Netz mit den Teilnehmerstationen A und B ist somit eingerichtet und betriebsbereit. Fehler-

situationen während der Anmeldungsphase, z. B. durch ausbleibende oder zu stark gestörte Datenpakete, können durch eine begrenzte Anzahl von Wiederholungssendungen behoben werden. Gelingt es einer Station trotz mehrfacher Versuche nicht, ein Netz mit einer anderen Station einzurichten oder Zugang zu einem Netz zu erhalten (s.u.), so nimmt sie an, daß sie entweder die bisher einzige Station eines Netzes ist oder von ihrem jetzigen Standort aus keinen Kontakt zu einem evtl. vorhandenen Netz erhalten kann.

Aufnahme einer Station in ein Netz mit mehreren Stationen

Die Aufnahme einer Station A in ein Netz mit zwei oder mehreren Stationen erfolgt grundsätzlich entsprechend dem oben beschriebenen Verfahren. Da die Zugangsforderung der Station A evtl. von mehreren Stationen empfangen wird, wählen alle Stationen eine stationsbezogene, zufällige Wartezeit bis zur Übertragung einer Anmeldebestätigung zur Station A. Dadurch wird die Anzahl möglicher Kollisionen auf einem von den Netzstationen gemeinsam genutzten Funkkanal des Organisationssystems reduziert. Erhält Station A durch erfolgreiche Abwicklung des Zugangsverfahrens mit mindestens einer Netzstation den Netzzugang, so vervollständigt sie ihre Netzintegration durch gezielte Kontaktaufnahme zu potentiellen Nachbarstationen, Empfang/Versenden von aktualisierten Routing-Informationen usw.

Abmeldung einer Netzstation

Funkstationen, die ein Netz verlassen wollen, informieren ihre Nachbarstationen zunächst per Rundsende-Paket über die beabsichtigte Abmeldung. Diese Vorgehensweise dient dazu, Routing-Tabellen und Teilnehmerverzeichnisse innerhalb des Netzes rechtzeitig zu aktualisieren. Ist eine abgehende Station als Relaisstation für Multi-Hop Verbindungen tätig, so werden, wenn möglich, die bestehenden Verbindungen in Absprache mit den beteiligten Nachbarrelaisstationen über alternative Routen umgeleitet.

2.6.5 Bitübertragungsschicht des Multi-Hop Funknetzes

Für die Funkübertragung werden z.B. Frequenzen im 60 GHz Bereich verwendet. Elektromagnetische Wellen dieses Frequenzbereichs unterliegen u.a. einer starken Sauerstoffdämpfung von ca. 15 dB/km, vgl. [93], wodurch ei-

ne hohe räumliche Wiederverwendbarkeit der Frequenzkanäle erzielt werden kann (Space Division Multiplex, SDM).

Übertragungskanäle werden in diesem dezentral organisierten Funknetz durch eine Kombination von Zeit- und Frequenzmultiplexverfahren erzeugt; durch Frequenzmultiplex wird das verfügbare Frequenzband in einzelne Teilbänder, durch das Zeitmultiplex jedes Teilband in eine Folge von Zeitschlitzten (Timeslots) aufgeteilt. Eine festgelegte Anzahl von Zeitschlitzten wird zu einem TDM- (Time Division Multiplex) Rahmen zusammengefaßt, der sich periodisch wiederholt. Ein Übertragungskanal ist durch die verwendete Trägerfrequenz und die Zeitschlitznummer innerhalb des TDM-Rahmens bestimmt. Für verschiedene Anwendungsgebiete (Backbone-Netz, Verteilnetz usw.) werden, bezüglich der Bitübertragungsschicht unterschiedliche Funkstationen entwickelt, die der Sicherungsschicht Nettoübertragungsraten von 2 Mbit/s bis zu 34 Mbit/s pro Trägerfrequenz verfügbar machen. Abb. 2.7 zeigt den Aufbau des TDM-Rahmens für eine Nettoübertragungsrate von 2 Mbit/s. Ein TDM-Rahmen

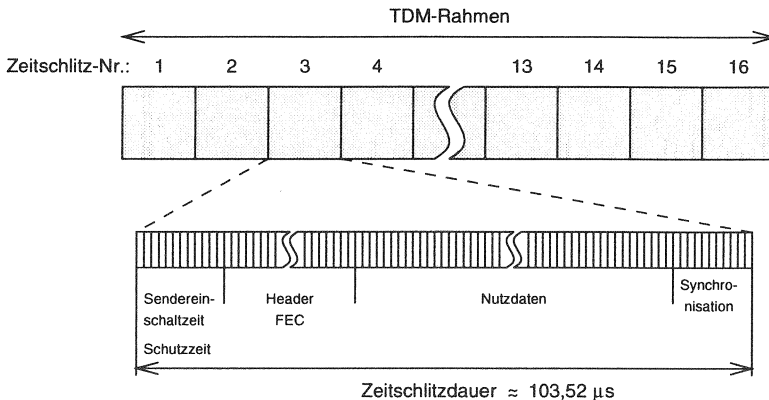


Abbildung 2.7: Zeitmultiplexrahmen für eine Nettoübertragungsrate von 2 Mbit/s pro Trägerfrequenz

besteht aus 16 Zeitschlitzten, je Zeitschlitz kann (aus Sicht der Sicherungsschicht) eine Nettonutzinformation von 212 bit übertragen werden, woraus bei 2 Mbit/s eine Rahmendauer bzw. Slotwiederkehrzeit von 1,65625 ms resultiert. Die in einem Zeitschlitz gesendete Datenmenge wird als Burst

bezeichnet. Ein Burst enthält neben den Nutzinformationen der Sicherungsschicht ein Schutz-/Synchronisationsintervall, Steuerinformationen der Bitübertragungsschicht und Informationen für die Vorwärtsfehlerkorrektur (Forward Error Correction, FEC). Zu den wesentlichen Aufgaben der Bitübertragungsschicht zählt die Herstellung der Rahmen- und Zeitschlitzsynchronisation im Verbund mit den Partnerinstanzen benachbarter Funkstationen. Entsprechende Verfahren für zentral und dezentral organisierte Netze wurden in der Arbeit von [113] entwickelt bzw. untersucht, miteinander verglichen und bewertet.

Formale Spezifikation von Kommunikationsprotokollen

3.1 Formale Methoden und Sprachen

Eine Methode wird (im technischen Sinne) als formal bezeichnet, wenn sie auf mathematischen Grundlagen, z.B. einem mathematischen Modell, basiert. Formale Sprachen machen die mathematische Basis einer formalen Methode verfügbar und dienen als Werkzeug zur systematischen Spezifikation, Entwicklung und Verifikation technischer Systeme. Sie werden von Systementwicklern verwendet, um das gewünschte Verhalten und die strukturellen Eigenschaften eines Systems zu beschreiben. Formale Sprachen eignen sich für jede Phase einer Systementwicklung:

- Anforderungsspezifikation,
- Systemdesign,
- Implementierung,
- Test und Fehlerbehebung,
- Maintenance,
- Verifikation.

Aufgrund der mathematischen Basis sind formale Spezifikationen eindeutiger und genauer als informale Beschreibungen. Die Vorteile formaler Sprachen für die Beschreibung technischer Systeme wurden auch von den internationalen Standardisierungsgremien erkannt, was zur Standardisierung der folgenden Sprachen geführt hat:

- SDL (Specification and Description Language, ITU-T Standard Z.100 ff),
- LOTOS (Language of Temporal Ordering Specification, ISO/IS 8807),
- ESTELLE (Extended State Transition Language, ISO/IS 9074).

Die Spezifikationssprache SDL zielt insbesondere auf die Entwicklung von Telekommunikationssystemen ab und wird zunehmend bei der Standardisierung sowie im industriellen Entwicklungsumfeld eingesetzt.

3.2 Einsatzbereiche von SDL

SDL wurde primär für die Beschreibung reaktiver und zeitdiskreter Systeme konzipiert. Als reaktive werden solche bezeichnet, deren Eigenschaften wesentlich durch Reaktionen auf Stimuli der jeweiligen Systemumgebung geprägt sind. Die Einordnung eines Systems als reaktiv/nicht-reaktiv ist nicht eindeutig, da das Verhältnis von Interaktion mit der Systemumgebung zu interner Verarbeitungsleistung des Systems kein absolutes Maß für die Beurteilung bietet. Beispiel für ein reaktives System ist ein Geldausgabeautomat, dessen Systemverhalten durch den Dialog mit dem Benutzer charakterisiert werden kann. Datenbanksysteme einer Client-Server Architektur, vgl. [106], sind nicht-reaktive Systeme, da ihre Eigenschaften hauptsächlich durch interne Verarbeitungsleistung gekennzeichnet sind.

Die Prädestination von SDL für reaktive Systeme kann deshalb nur als Entscheidungshilfe dienen, ob eine Spezifikationsaufgabe mit SDL befriedigend gelöst werden kann oder nicht.

Als zeitdiskret in obigem Sinne werden Systeme bezeichnet, deren Verhalten durch Reaktion auf einzelne, zeitlich voneinander trennbare Ereignisse beschrieben werden kann.

SDL ist somit auch nicht uneingeschränkt zur Spezifikation beliebiger Telekommunikationssysteme geeignet (z.B. Datenbanken von Vermittlungssystemen), viele (Teil-) Systeme wie Protokolle, Schnittstellen usw. sind reaktive zeitdiskrete Systeme in obigem Sinn.

3.3 Standardisierung

Tab. 3.1 enthält eine Übersicht über alle relevanten SDL-Standards der ITU-T und führt weitere Standards an, die von Bedeutung für den Einsatz von SDL sind.

Der Standard Z.100 (ohne Anhänge) beinhaltet eine vollständige SDL-Sprachbeschreibung in Form einer abstrakten Grammatik, sowie die Definition der Relationen (z.B. Transformation) von konkreter graphischer

Tabelle 3.1: SDL-Standards

ITU-T Standard	Titel
Z.100	CCITT Specification and Description Language (SDL)
Z.100 Annex A	Index of non-terminals and keywords
Z.100 Annex B	Glossary
Z.100 Annex C	Initial algebra model
Z.100 Annex D	SDL predefined data
Z.100 Annex E	[reserved for future use]
Z.100 F1	Specification and Description Language (SDL) - SDL formal definition: Introduction
Z.100 F2	Specification and Description Language (SDL) - SDL formal definition: Static semantics
Z.100 F3	Specification and Description Language (SDL) - SDL formal definition: Dynamic semantics
Z.100 Appendix I	SDL methodology guidelines
Z.100 Appendix II	SDL bibliography
Z.105	SDL Combined with ASN.1 (SDL/ASN.1)
Z.110	Criteria for the use and applicability of Formal Description Techniques
Z.120	Messages Sequence Chart (MSC)

bzw. (gemeinsamer) textueller Grammatik zur abstrakten Grammatik, vgl. Abschn. 3.4.

Die Anhänge A–D umfassen ergänzende Erläuterungen, Tabellen, Darstellungen und Definitionen des Z.100 Standards.

Der Anhang F1–3 enthält eine ergänzende, formale SDL–Sprachdefinition,

die eine veränderte Definitionsstruktur aufweist. Ausgangspunkt ist ein kompletter Satz von Konstruktionsregeln (AS_0), der die SDL-Syntax in abstrakter Form ohne semantische Elemente definiert. Der Anhang F2 beschreibt neben den semantischen Grundregeln (z.B. Gültigkeitsregeln für Bezeichner, Wertebereiche für Variablen) Transformationsregeln für die Überführung einer AS_0 -Repräsentation in eine sog. AS_1 -Repräsentation, die im Standard Z.100 zur syntaktischen Sprachdefinition verwendet wird. Die dynamische Semantik (Anhang F3) definiert die dynamischen Eigenschaften von SDL auf Basis von AS_1 -Repräsentationen. Als Modell wird eine abstrakte SDL-Maschine mit sechs Prozeß-Typen zugrunde gelegt.

Diese zusätzliche formale Sprachdefinition ist aufgrund ihrer strengen Trennung von syntaktischer, statisch- und dynamisch-semantischer Sprachdefinition sowohl für die Entwicklung von SDL-Werkzeugen als auch für ein tiefgehendes Verständnis der SDL-Sprachkonzepte geeignet.

Das Zusatzdokument I des Z.100 Standards macht Richtlinien für den effektiven Einsatz von SDL im Rahmen einer Gesamtmethodik verfügbar. Behandelt werden beispielsweise Themenbereiche wie die stufenweise Entwicklung von SDL-Spezifikationen, Einsatz von Message Sequence Charts (MSC), formale Ansätze zur Verifikation und Validierung, Einsatz von SDL in verschiedensten Anwendungsbereichen.

Die Verwendung von ASN.1¹ in SDL-Spezifikationen ist in der Empfehlung Z.105 festgelegt. ASN.1 ist eine abstrakte Syntaxnotation zur zielsystemunabhängigen Beschreibung und Darstellung von Daten. Die konkrete Datenrepräsentation in einem Zielsystem sowie eine Transfersyntax für den Datenaustausch zwischen beliebigen Endsystemen ist in den 'Basic Encoding Rules for ASN.1' beschrieben. Durch die (ergänzende) Verwendung von ASN.1 für Spezifikationen sollen bisherige Schwächen von SDL bzgl. der Datentyp-Definition und Datendarstellung beseitigt werden.

Der Standard Z.110 gibt einen kurzen Überblick über die Ziele und Vorteile des Einsatzes formaler Sprachen. Es werden Empfehlungen für die stufenweise Einführung formaler Beschreibungssprachen zur Spezifikation von ITU-T Standards gegeben.

Message Sequence Charts (Recommendation Z.120) visualisieren Sequenzen von Nachrichten, die zwischen Systemen ausgetauscht werden. In Verbindung mit SDL-Systemen (Spezifikationen) werden MSCs zur Anforderungsspezifikation, Schnittstellenbeschreibung, Spezifikation von Testfällen, Vali-

¹Abstract Syntax Notation One

dierung usw. verwendet. Der Standard Z.120 enthält Syntax-Definitionen für MSC-Dokumente, MSCs und alle Elemente eines MSC, sowie Empfehlungen für die Strukturierung und Modularisierung von MSC-Dokumenten.

3.4 SDL-92

Die Sprache SDL bietet zwei syntaktische Beschreibungsformen an: eine textuelle (SDL-PR²) und eine graphische Variante (SDL-GR³). Anhand einer abstrakten Grammatik können beide Syntaxen semantisch äquivalent ineinander überführt werden. Da nicht alle textuellen Sprachelemente ein graphisches Analogon besitzen, existiert eine in beiden Beschreibungsformen gemeinsam verwendete Grammatik (Common Textual Grammar).

Innerhalb der ITU-T Studienperiode 1988-92 wurden wesentliche Veränderungen bezüglich der SDL-Sprachkonzeption vorgenommen und eine Vielzahl neuer Sprachelemente integriert.

Ziel der folgenden Abschnitte ist es, ausschließlich die wesentlichen Erweiterungen von SDL-92 gegenüber SDL-88 darzustellen. Der Sprachumfang von SDL-88 wird als hinreichend bekannt vorausgesetzt bzw. auf entsprechende Literatur [52, 1] verwiesen.

3.4.1 Struktur einer SDL-Spezifikation

Der wesentliche Unterschied zwischen SDL-88 und SDL-92 ergibt sich durch Einführung objektorientierter Methoden; bei den neuen Sprachelementen von SDL-92 wird grundlegend zwischen Typen und Instanzen unterschieden:

- Als *Instanzen* werden die realen Objekte bzw. Komponenten einer SDL Spezifikation bezeichnet.
- *Typen* beschreiben ausschließlich die Eigenschaften von Instanzen, sie stellen keine Objekte in einer Spezifikation dar.

Die Sprache SDL zeichnet sich durch eine strenge Hierarchie der Strukturierungskomponenten aus, wodurch jeweils verfeinernde Spezifikationsebenen entstehen. Die Komponente der höchsten Hierarchieebene ist die SDL-System-Instanz, die mit ihrer Umgebung (Environment) kommuniziert. Das Environment ist die kummulative Bezeichnung für alle Objekte außerhalb

²Phrase Representation

³Graphical Representation

einer Systemspezifikation. Eine Spezifikation läßt sich durch die folgenden Komponenten gliedern:

- Eine *System-Instanz* besteht aus einer oder mehreren Block-Instanzen.
- *Block-Instanzen* enthalten entweder Prozeß-Instanzen oder Sub-Block-Instanzen. Das Block-Konzept zielt auf die konzeptionelle Zusammenfassung logisch zusammenhängender Prozesse ab.
- *Prozeß-Instanzen* beinhalten entweder direkt einen endlichen Zustandsautomaten, oder sie bestehen aus einer beliebigen Anzahl divergenter Service-Instanzen.
- Innerhalb einer *Service-Instanz* wird ein endlicher Zustandsautomat beschrieben.

Prozeßspezifikationen können durch Verwendung von Prozeduren, die beispielsweise häufig verwendete Routinen beinhalten, weiter strukturiert werden. Das bisherige Konzept wurde in SDL-92 erweitert: Prozesse können ihre Prozeduren durch Kennzeichnung mit einer Export-Anweisung anderen Prozessen verfügbar machen. Die Bearbeitung eines Prozeduraufrufs übernimmt der exportierende Prozeß, der Austausch der Übergabeparameter bzw. des Rückgabewertes zwischen aufrufendem und bearbeitendem Prozeß erfolgt über implizite Signale. Die Bearbeitung von Prozeduraufrufen kann für einen Prozeßzustand explizit unterdrückt werden, indem ein entsprechendes Sicherungssymbol in die Liste potentieller Eingangssignale eingefügt wird.

3.4.2 Kommunikation zwischen Instanzen

Instanzen einer SDL-Spezifikation tauschen Informationen mit Hilfe von Signalen aus. Block-Instanzen werden über Kanäle miteinander oder dem Environment verbunden, während Prozeß- und Service-Instanzen über Signalwege kommunizieren.

Über Kanäle bzw. Signalwege werden Signal-Instanzen eines bestimmten Signal-Typs übertragen; die Kommunikation zwischen Instanzen in unterschiedlichen Strukturierungspfaden eines Systems erfolgt analog zur hierarchischen Struktur einer System-Spezifikation: Beispielsweise kommunizieren zwei Prozeß-Instanzen, die in unterschiedlichen Blöcken definiert sind, zunächst über Signalwege mit der zugehörigen Block-Instanz, während die Block-Instanzen über Kanäle miteinander verbunden sind.

Kanäle und Signalwege weisen folgende unterschiedliche Merkmale auf:

- Für Kanäle können sog. Kanal-Substrukturen definiert werden, die Eigenschaften des Übertragungsmediums spezifizieren und beispielsweise zur Modellierung (gestörter) Funkkanäle dienen können.
- Die Signalübertragung über Kanäle ist nicht deterministisch, d.h. es können zufällige Übertragungsverzögerungen entstehen. Durch Anwendung des Schlüsselwortes NODELAY bei der Kanalspezifikation können Verzögerungen unterbunden werden.
- Signalwege übertragen immer verzögerungsfrei; eine Liste zulässiger Signale wird, sofern nicht explizit definiert, von der Kanalspezifikation implizit abgeleitet.

Für die Spezifikation von SDL-Typen, vgl. Abschn. 3.4.3, wurde das Gate-Konzept eingeführt. Gates repräsentieren Verbindungsendpunkte für Kanäle und Signalwege in einer Typ-Deklaration; sie ermöglichen Verbindungen zwischen einer Instanz dieses Typs und anderen äquivalenten Instanzen bzw. der nächsthöheren Hierarchiekomponente. Durch strenge Definition eines Gates kann die Menge der ankoppelbaren Kanäle/Signalwege, ein- bzw. ausgehender Signale und Kommunikationspartner eng begrenzt werden.

3.4.3 Die Spezifikation von SDL-Typen

Die Spracherweiterungen in SDL-92 durch Einführung typ-basierter Instanzen bieten, in Verbindung mit dem Vererbungskonzept, neue Möglichkeiten der Strukturierung einer SDL-Spezifikation nach dem Prinzip der Generalisierung/Spezialisierung. Die zugehörigen Sprachelemente ermöglichen:

- Typ-Deklarationen innerhalb eines SDL-Systems oder eines Packages (s.u.),
- Definition typbasierter Instanzen oder Mengen von Instanzen,
- Deklaration parametrisierbarer Typen mit Hilfe formaler Kontext-Parameter,
- Spezialisierung von Typ-Deklarationen in Sub-Typen durch Hinzufügen von Eigenschaften und Redefinition virtueller Typen bzw. Zustandsübergänge.

Der SDL-92 Standard sieht die Typisierung von folgenden Sprachelementen vor: System, Block, Prozeß und Service.

Prozeduren, Signale, Timer und Sorten besitzen implizit aufgrund ihrer Merkmale den Typen-Charakter.

Die Spezifikation von Typen erfolgt in zugehörigen Typ-Diagrammen. Sofern mehrere Spezifikationen von derselben Typ-Deklaration Gebrauch machen wollen, muß das Typ-Diagramm in ein sog. Package bzw. Package-Diagramm integriert werden.

Package-Diagramme können beliebige Arten von Typ-Spezifikationen enthalten, System-Typen können ausschließlich innerhalb eines Package-Diagramms spezifiziert werden. Das Package-Konzept in SDL-92 ist vergleichbar mit dem der Bibliotheken in (informalen) Programmiersprachen. Abb. 3.1 zeigt Referenz-Symbole für SDL-92-Typen. Referenz-Symbole dienen einerseits zur Definition von Instanzen (Abb. 3.1 links), sie verweisen auf das Typ-Diagramm, in dem die Deklaration des Typen explizit vorgenommen wird. Andererseits werden durch Referenz-Symbole Bezüge auf die Elemente einer Typ-Deklaration bei der Spezifikation eines Sub-Typen genommen (Abb. 3.1 rechts), sie stellen ausschließlich die Funktion der referenzierten Elemente im aktuellen Kontext dar.

Die Instanziierung von Komponenten einer SDL-Spezifikation basiert nicht notwendigerweise auf deklarierten Typen. Sofern der Aufwand für eine Trennung von Typen-Deklaration und Instanzen für eine konkrete Anwendung nicht sinnvoll erscheint, z.B. bei einem System, das nur aus einem Block und einem Prozeß besteht, können Typ-Deklarationen und Instanz-Definition in einem Instanz-Diagramm vereint werden, wobei folgende Einschränkungen gelten:

- Innerhalb des Diagramms können keine direkten Typ-Deklarationen vorgenommen werden. Operationen auf Typen, vgl. Abschn. 3.4.5, wie Spezialisierung, Redefinition virtueller Typen und Zustandsübergänge sind ebenfalls nicht möglich.
- Das Schlüsselwort TYPE darf nicht verwendet werden.
- Kanäle und Signalrouten sind direkt mit Kanälen bzw. Signalrouten der nächsthöheren Hierarchieebene verbunden.
- Instanz-Symbole dienen als Referenz-Symbole für die zugehörigen Instanz-Diagramme.

3.4.4 Parametrisierung von Typen

Zur weiteren Flexibilisierung und Wiederverwendbarkeit von Typ-Deklarationen wurde das Konzept der Parametrisierung durch formale Kontext-Parameter eingeführt. Formale Kontext-Parameter werden, eingeschlossen in <> Klammern, nach dem Typ-Namen spezifiziert; System-, Block-,

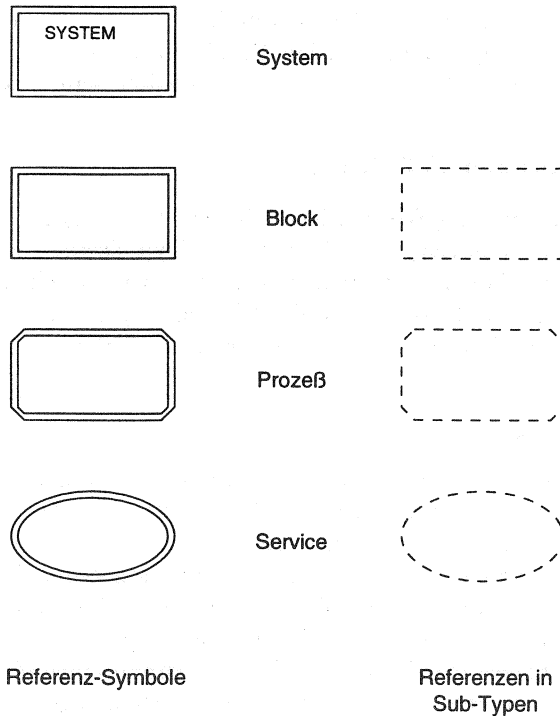


Abbildung 3.1: Typ Referenz-Symbole in SDL-92

Prozeß- und Service-Typen sowie Prozeduren, Signale und Sorten können formale Kontext-Parameter enthalten. Die Spezifikation der Parameter enthält neben dem Bezeichner Anforderungen bezüglich seiner Eigenschaften.

Hierbei wird wie folgt unterschieden:

- *Bezeichner-Anforderungen* spezifizieren direkt die Eigenschaften des Parameters,
- eine *Typ-Anforderung*, die durch das Schlüsselwort ATLEAST eingeleitet wird, legt fest, daß der Parameter bei der Instanzierung durch einen aktuellen Kontext-Parameter ersetzt wird, der dem Typ bzw. einem abgeleiteten Typ der atleast-Anforderung entspricht.

Formale Kontext-Parameter können vom Typ Prozeß, Prozedur, Variable, Timer, Sorte oder Synonym sein. Die Spezifikation einer Bezeichner- oder Typ-Anforderung dient im wesentlichen der statisch semantischen Analyse.

3.4.5 Operationen auf Typen

Mit der Einführung objektorientierter Konzepte in SDL-92 werden auch Methoden zur semantischen Bearbeitung von Typ-Deklarationen verfügbar gemacht. Hierzu zählen die folgenden Verfahren zur Verfeinerung und kontextspezifischen Modifikation von abgeleiteten Typen, die ihre Basiseigenschaften entsprechend dem Prinzip der Vererbung erhalten.

Spezialisierung

Eine (Sub-) Typ-Deklaration kann durch Spezialisierung, d.h. typischerweise funktionale Erweiterung, eines Basistypen entstehen. Durch fortlaufende Spezialisierung entsteht eine baumartige Hierarchie von Typ-Deklarationen, wobei jeweils der zugehörige Knoten in der nächsthöheren Hierarchieebene den Basistyp des spezialisierten Typen bezeichnet. Das Verfahren der Spezialisierung soll anhand eines vereinfachten Beispiels in Abb. 3.2 verdeutlicht werden. Der Prozeß-Typ *Drucker_Steuerung* enthält einen stark vereinfachten Zustandsautomaten für eine Nadeldrucker-Steuerung.

Die eintreffenden ASCII Zeichen werden mittels eines Prozeduraufrufs (Prozedur *Bestimme_Pixelmatrix*) in eine Matrix überführt und zwischengespeichert. Die angeschlossene Druckeinheit zeigt ihre Bereitschaft über das Signal 'Fertig' an, und erhält durch die Prozedur *Drucke_Pixelmatrix* einen weiteren Auftrag.

Soll die Druckersteuerung zur Bedienung eines grafikfähigen Druckers weiterverwendet werden, so kann eine Spezialisierung in einem weiteren Prozeß-Typ erfolgen (hier: *Erweiterte_Drucker_Steuerung*), die durch das Schlüsselwort *INHERITS* angezeigt wird.

Die Spezialisierung deklariert:

- ein neues Eingangssignal (*Graphik_Zeichen*) am Gate mit der Bezeichnung 'VonSchnittstelle'.
- einen weiteren Ast des Zustandsautomaten, der, nach Eingang des Signals 'Graphik_Zeichen', eine Prozedur zur Bestimmung und Zwischenspeicherung der Pixelmatrix (für das Graphikzeichen) ausführt.

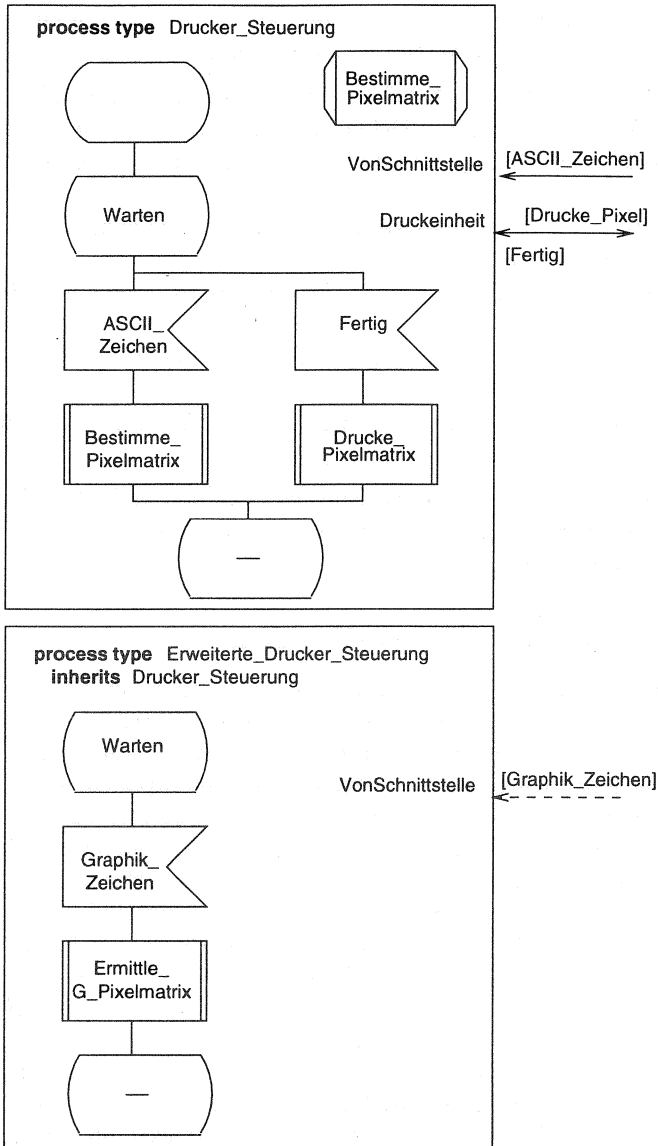


Abbildung 3.2: Beispiel: Vereinfachte Druckersteuerung

Generell können bei einer Spezialisierung alle Eigenschaften eingefügt werden, die auch bei der Deklaration eines Typen erlaubt sind:

- Block-, Prozeß- und Service-Typen können existierende Gates um Signale und die gesamte Deklaration um neue Gates erweitern.
- Prozeß-, Service-Typen und Prozeduren können weitere Zustände und Zustandsübergänge hinzufügen.
- Die Parameterliste eines spezialisierten Signals kann zusätzliche Datentypen enthalten.
- Die Anzahl formaler Parameter ist bei Prozeß-Typen und Prozeduren erweiterbar.
- Für Datentypen können weitere Literale und Operatoren definiert werden.
- Formale Kontext-Parameter, vgl. Abschn. 3.4.4, können hinzugefügt werden.

Voraussetzung für jede Spezialisierung ist, daß eine Sub-Typ-Deklaration keinen Bezeichnernamen enthalten darf, der bereits in einem zugehörigen Basistyp verwendet wurde.

Virtuelle Typen und Zustandsübergänge

Die Merkmale einer Typ-Deklaration können entsprechend der oben angeführten Spezialisierungsregeln beliebig *erweitert* werden.

Das Konzept der virtuellen Typen und Transitionen unterstützt eine effiziente und gezielte *Redefinition* von Spezifikationskomponenten: der Basistyp bestimmt diejenigen Typ-Deklarationen, welche durch einen Sub-Typen redefiniert werden dürfen, und kennzeichnet diese mit dem Schlüsselwort **VIRTUAL**. Dieselbe Methode kann für Transitionen angewendet werden, die zu einem Zustand oder einem Start-Symbol gehören (virtual transition).

Sofern ein Sub-Typ die Redefinition eines virtuellen Typen bzw. einer virtuellen Transition durch weitere Sub-Typen gestatten möchte, kennzeichnet er die entsprechende Deklaration mit dem Schlüsselwort **REDEFINED**, sonst mit dem Schlüsselwort **FINALIZED**.

Die Verwendung virtueller Typen und Transitionen soll anhand des Beispiels einer Nadeldrucker-Steuerung verdeutlicht werden. Abb. 3.3 zeigt eine modifizierte Deklaration des Prozeß-Typs `Drucker_Steuerung`. Der Zustandsübergang bei Empfang eines Signals 'ASCII-Zeichen' sowie die Prozedur 'Bestimme.Pixelmatrix' sind als virtuell gekennzeichnet. Die virtuel-

len Komponenten können innerhalb einer Sub-Typ-Deklaration (Erweiterte_Drucker_Steuerung) durch eine geeignete Prozedur bzw. Transition ersetzt werden.

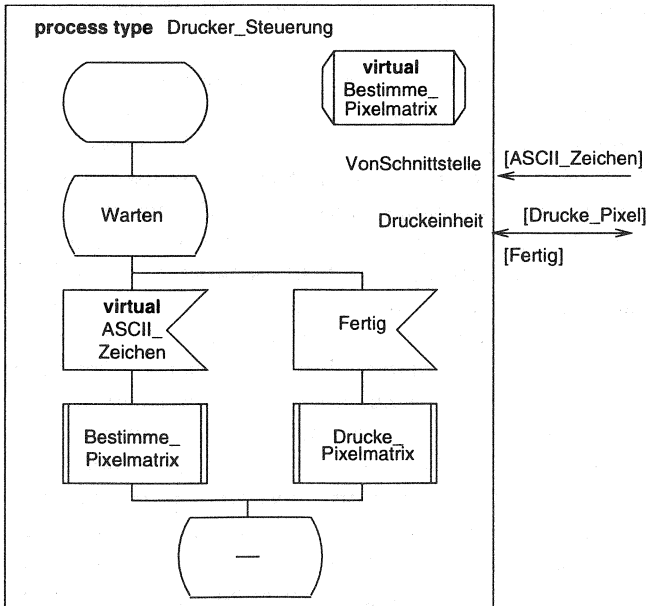


Abbildung 3.3: Beispiel: Vereinfachte Druckersteuerung mit virtuellen Typ-Deklarationen

3.4.6 Spezifikation des Systemverhaltens

Im folgenden Abschnitt werden einige interessante Spracherweiterungen zur Spezifikation von Zustandsautomaten (EFSM) vorgestellt.

Stimuli für Transitionen

Jede Prozeß-Instanz eines SDL-Systems besitzt eine Eingabewarteschlange, in der eingehende Signale bis zur Verarbeitung zwischengespeichert werden.

Für jeden Zustand des Prozeßautomaten wird eine Menge erwarteter Eingangssignale definiert, die einen Zustandsübergang auslösen können. Der SDL-88 Standard sieht vor, daß ein Signal verworfen wird, das der Eingabewarteschlange entnommen wurde und nicht zur Liste erwarteter Eingangssignale des aktuellen Zustands gehört. SDL-92 bietet weitere Sprachkonstrukte für Signale an:

- ein Symbol für eingehende Signale kann mit einem * (Wildcard) versehen werden. Der zugehörige Transitions-Ast wird für alle Signale durchlaufen, die nicht zur Liste erwarteter Eingangssignale gehören. Der * Bezeichner kann ein SAVE-Symbol kennzeichnen, jedoch nicht parallel zu einem Eingangssymbol mit * Bezeichner.
- spontane Zustandsübergänge werden durch Kennzeichnung eines Eingangssignal-Symbols mit dem Schlüsselwort NONE erlaubt. Der Zustandsübergang erfolgt zufällig durch den zugehörigen Prozeß (kein externer Stimulus) und ist unabhängig von der Existenz eines Signals in der Eingabewarteschlange. Eine Priorisierung der Signalverarbeitung oder des spontanen Zustandsübergangs ist nicht vorgesehen. Diese Sprachkomponente wurde insbesondere zur Modellierung unzuverlässiger Systeme (z.B. gestörter Übertragungskanäle) eingeführt.
- ein Zustandsübergang kann ausgelöst werden, sofern eine vorgegebene Bedingung erfüllt ist. Diese Bedingung wird innerhalb eines <> Symbols und parallel zu den erwarteten Eingangssignalen eines Prozesses definiert. Ist die Eingabewarteschlange des Prozesses leer und die spezifizierte Bedingung erfüllt, so wird die zugehörige Transition ausgelöst. Existieren mehrere konditionale Transitionen für einen Zustand, so können Bearbeitungsprioritäten mittels des Schlüsselworts PRIORITY vergeben werden.

Nicht-deterministische Entscheidungen

Mit Hilfe von Verzweigungsanweisungen kann, wie aus (informalen) Programmiersprachen bekannt, der Ablauf eines Zustandsübergangs abhängig von der Auswertung einer Bedingung entlang unterschiedlicher Wege geführt werden. In Anlehnung an das Konzept spontaner Zustandsübergänge, vgl. vorherigen Abschn., werden nicht deterministische Entscheidungen bei Verzweigungsanweisungen unterstützt.

Erreicht ein Zustandsübergang ein Verzweigungssymbol, das mit dem Schlüsselwort ANY gekennzeichnet ist, dann wird zufällig ein Pfad aus

der Menge verfügbarer Verzweigungen ausgewählt.

Nicht-deterministische Entscheidungen wurden ebenso wie spontane Zustandsübergänge als Sprachelemente zur Modellierung zufälliger Vorgänge in Systemen eingeführt.

3.4.7 Daten in SDL-92

Die Sprache SDL basiert auf dem Modell des endlichen Automaten, das um ein Variablenkonzept erweitert wurde. Variablenwerte werden zur aktiven Steuerung des Verhaltens einer Prozeß-Instanz verwendet, wie z.B.:

- bei Verzweigungssymbolen wird anhand eines Variablenwertes der folgend zu durchlaufende Verzweigungsast bestimmt,
- Variablenwerte können beim Aufruf einer Prozedur oder der dynamischen Instanzierung eines Prozesses übergeben werden,
- durch parametrisierte Ein- bzw. Ausgabesignale können Daten zwischen Instanzen ausgetauscht werden,
- Variablenwerte bestimmen den Ablaufzeitpunkt eines Timers.

Die Verwendung bzw. Verwaltung von Daten-Typen entspricht im wesentlichen den bekannten Verfahren von (informalen) Programmiersprachen:

- SDL bietet vordefinierte Datentypen und Sprachelemente zur Definition neuer Datentypen.
- Jeder Datentyp spezifiziert Namen für jeden möglichen Variablenwert, die Namen werden als *Literale* des Datentypen bezeichnet.
- Es existieren Anweisungen, mit denen der Wert einer Variable ermittelt werden kann.
- Eine Variable kann nur Werte eines Datentyps enthalten, dem sie zugeordnet wurde.

Das folgende Beispiel zeigt eine (nicht vollständige) Spezifikation des vordefinierten Datentyps REAL:

```
newtype Real
literals 0.0,0.1, ... 4
operators
```

⁴Zur Beschreibung einer unendlichen Menge von Literalen wird ein spezieller Konstrukt mit der Bezeichnung `nameclass` verwendet, vgl. [32].

"−"	: Real	→ Real
"+"	: Real, Real	→ Real
"−"	: Real, Real	→ Real
"*"	: Real, Real	→ Real
"/"	: Real, Real	→ Real
">"	: Real, Real	→ Boolean
"<"	: Real, Real	→ Boolean
">="	: Real, Real	→ Boolean
"<="	: Real, Real	→ Boolean

/* Definition der Operatoren */
endnewtype

Die Typdefinition beginnt mit einer Liste bzw. Spezifikation der Literale, gefolgt von einer Aufzählung der Operatoren. Für jeden Operator (begrenzt durch Anführungszeichen) werden die Datentypen der Argumente und des Ergebnisses der Operation angegeben. Beispielsweise werden dem '<' Operator zwei Argumente vom Typ Real übergeben, das Ergebnis ist vom Typ Boolean.

Bei Anwendung eines Operators wird entsprechend die Infix-Notation verwendet (z.B. $2.0 < 3.0$), bei arithmetischen oder relationalen Operatoren ist auch die Präfix-Notation, z.B. "<" (2.0,3.0), anwendbar. Die Vergleichsoperatoren ($=$, $\neq$ (nicht identisch)) sind für jeden Datentyp implizit definiert. Der Auflistung der Operatoren schließt sich ihre eigentliche Definition an. SDL-92 bietet hierfür vier verschiedene Methoden an, die miteinander verknüpft werden können:

- Definition durch informalen Text,
- Axiomatische Definition,
- Algorithmische Beschreibung,
- Definition mittels (externer) Beschreibungssprachen.

Informale Methode

Informale Beschreibungsformen sollen insbesondere zur Operator-Definition in der Startphase einer Systemspezifikation dienen, zur Konkretisierung kann eine der nachfolgenden Methoden verwendet werden.

Das folgende Beispiel zeigt eine Definition mit informalem Text, die durch das Schlüsselwort **AXIOMS** eingeleitet und durch Apostroph-Zeichen begrenzt wird.

newtype SLib
operators

last : Charstring → Character

axioms

'last(string) = ermittle den letzten Buchstaben der Zeichenkette *string*'

endnewtype SLib

Axiomatische Definition

Diese Beschreibungsform entspricht dem von SDL-88 bekannten Verfahren zur Definition abstrakter Datentypen (ADT). Die Spezifikation erfolgt anhand quantifizierter Äquivalenzen, die aus einem links- und einem rechtsseitigen Ausdruck verbunden durch das Symbol **==** bestehen. Die Quantifizierung definiert die Gültigkeit der Äquivalenzen für jeden Wert eines Typs (for in ... all Anweisung).

Beispiel:

newtype SLib
operators

last : Charstring → Character

axioms

for all last_element in Character
(for all rest_of_string in Charstring
(last("") == error!;
last(rest_of_string // Mkstring(last_element)) == last_element))

endnewtype SLib

Die Operator-Definition basiert auf vorgegebenen Sprachelementen in SDL: die Anwendung des last-Operators auf eine leere Zeichenkette (") erzeugt einen dynamischen Fehler. Eine weitere Äquivalenz beschreibt ein Zeichen

als Ergebnis der Anwendung des last-Operators auf eine Verknüpfung von Zeichenketten (rest_of_string) und Mkstring(last_element). Die einschließende Quantifizierung definiert die Gültigkeit der Äquivalenzen für alle Literale des Datentypen Character bzw. Charstring.

Algorithmische Beschreibung

Diese Methode stützt sich auf einen in SDL-92 eingeführten Typ mit der Bezeichnung OPERATOR, der eine prozedurale Operator-Definition erlaubt. Diese Beschreibungsform setzt häufig das Vorhandensein grundlegender Datentypen und Operatoren voraus.

Das nachfolgende Beispiel zeigt die Spezifikation des last-Operators in der textuellen Form kombiniert mit einem Operator-Diagramm (Abb. 3.4).

newtype SLib

operators

last : Charstring → Character

operator last **fpar** string Charstring;

returns Character; **referenced**

endnewtype SLib

Definition mittels (externer) Beschreibungssprachen

Dieses Verfahren zielt auf die Integration spezieller Notationssprachen wie ASN.1, vgl Abschn. 3.3, ab und erlaubt auch die Einbindung von Routinen und Bibliotheken (informaler) Programmiersprachen. Die Anbindung bzw. Referenzierung erfolgt innerhalb einer Sektion, die durch die Schlüsselwörter ALTERNATIVE und ENDALTERNATIVE gekennzeichnet ist.

Beispiel:

newtype SLib

operators

last : Charstring → Character

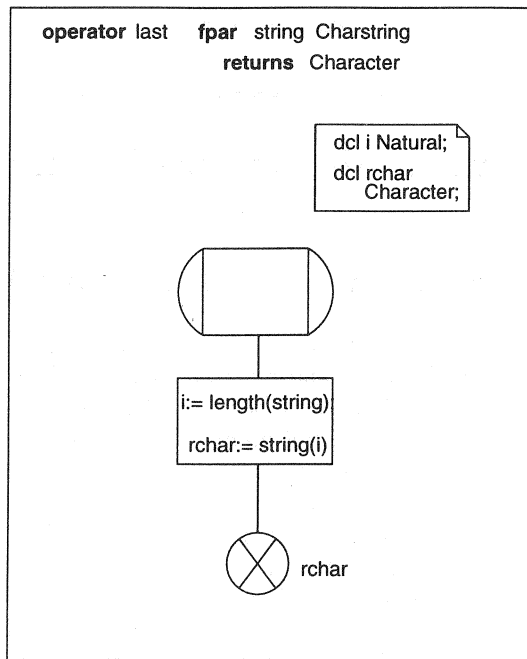


Abbildung 3.4: Prozedurdiagramm für den last-Operator

```

alternative C
#include "slib.h"
#include "slib.c"
endalternative C
endnewtype SLib

```

RMAN: ein ATM-basiertes Mobilfunknetz

4.1 Entwurfskriterien und -ziele für das RMAN

In Abschn. 2.6 wurden die grundlegenden Spezifikationen für ein dezentral organisiertes Mobilfunknetz und Festlegungen bezüglich der Bitübertragungsschicht (OSI-Schicht 1) beschrieben. Ein Ziel der vorliegenden Arbeit besteht darin, das Multi-Hop Mobilfunknetz hinsichtlich seiner Architektur weiterzuentwickeln, insbesondere geeignete Kommunikationsprotokolle für Sprach- und Datendienste zu entwerfen und ihre Leistungsfähigkeit zu bewerten. Das resultierende Netz erhält im folgenden die Bezeichnung *Radio Metropolitan Area Network (RMAN)*.

Die hohen Entwicklungskosten für Mobilfunknetze und die ständig wachsenden Angebote neuer, zukünftig personalisierter Dienste erfordern zunehmend eine hohe Flexibilität der Netzkomponenten, d.h. das Netz muß sich verschiedenen oder veränderten Anforderungen des Betreibers anpassen können. Das RMAN soll deshalb neben *kanalvermittelten* auch verschiedenste *paketvermittelnde* Dienste verfügbar machen.

Die Entwicklung zukünftiger, digitaler (Multi-Hop) Mobilfunknetze, z.B. für den Einsatz als Zugangsnetz (Abschn. 2.1), verläuft parallel zur Einführung breitbandiger, drahtgebundener Weitverkehrsnetze und echtzeitfähiger lokaler Netze auf der Basis des Asynchronen Übertragungsmodus (Asynchronous Transfer Mode, ATM). ATM wurde zunächst als Übertragungsverfahren für das diensteintegrierende, digitale Breitbandnetz B-ISDN, vgl. [57], mit Übertragungsraten jenseits von 100 Mbit/s von der ITU-T standardisiert. Mittlerweile existieren Empfehlungen der ITU-T, vgl. [56], die den Einsatz von ATM in existierenden Festnetzen mit deutlich geringeren Übertragungsraten, z.B. bei 2,048 Mbit/s (N-ISDN), spezifizieren.

Das ATM-Verfahren ist als schnelles Paketvermittlungsverfahren konzipiert. Die verwendeten Pakete, die als Zellen bezeichnet werden, haben eine konstante Länge von 53 byte, wobei 48 byte für Nutzinformationen und 5 byte für Steuerinformationen verfügbar sind. Die Übertragung von ATM-

Zellen erfolgt verbindungsorientiert über virtuelle Kanäle und Gruppierungen solcher Kanäle (virtuelle Pfade). Der ATM-Zellkopf (5 byte) hat gegenüber bekannten Paketvermittlungsnetzen stark reduzierte Aufgaben. Er enthält Informationen für die Wegelenkung (Virtual Path/Channel Identifier, VPI/VCI), den Typ der Zelle, Priorität, Dienstgüteparameter und eine zellkopfbezogene Prüfsumme zur Fehlerkorrektur.

(Breitband-) Netze auf ATM-Basis werden ein breites Angebot von Telekommunikationsdiensten mit konstanter oder variabler Bitrate wie Sprache, Fax, Datenübertragung, Video, Multimedia-Dienste usw. verfügbar machen. In Verbindung mit derzeit noch in der Standardisierung befindlichen, erweiterten Netzfunktionen wie Verbindungsannahmesteuerung, Staukontrolle, Dienstgüteüberwachung usw. sollen ATM-Netze teilweise oder insgesamt z.B. folgende Vorteile gegenüber existierenden Netzen aufweisen:

- effiziente Bandbreitenvergabe und Bandbreitennutzung: Jeder Teilnehmerverbindung wird nur der jeweils benötigte Anteil der Netz-Übertragungskapazität verfügbar gemacht. Durch Verwendung von Zellen konstanter Länge wird eine optimierte Kapazitätsausnutzung der Netzteilstrecken (zwischen ATM-Vermittlungsknoten) ermöglicht.
- zeitliche und semantische Transparenz: Als wesentliches Dienstgütemerkmal für Echtzeitanwendungen wird seitens des Netzes eine maximale Verzögerung bzw. eine maximale Varianz der Verzögerung für die Zellübertragung garantiert. ATM-Zellen werden sequentiell entlang eines virtuellen Pfades durch das Netz zwischen Datenquelle und -ziel übertragen, wodurch die semantische Reihenfolge der Zellen nicht verändert wird.
- Granularität: Die Entwicklung neuer Dienste und Applikationen muß sich, im Gegensatz zu kanalvermittelnden Netzen, nicht zwangsläufig an einer fest vorgegebenen Übertragungsrate eines Kanals orientieren.

Die angesprochene Entwicklung im Bereich der diensteintegrierenden, digitalen Weitverkehrsnetze und lokalen Netze ist beim Entwurf zukünftiger Mobilfunknetze zu berücksichtigen. Es ist naheliegend, Konzepte und Protokolle ATM-basierter Festnetze auf den Bereich der Mobilfunknetze zu übertragen. Zwar werden Mobilfunknetze aufgrund funktechnischer Begrenzungen zunächst nur eine Teilmenge von Festnetzdiensten mit evtl. reduzierten Dienstgütemerkmalen verfügbar machen, diese grundsätzliche Vorgehensweise erweist sich aber aus Sicht von Netzbetreibern, Dienst Anbietern, Endgeräteherstellern und Anwendern als sinnvoll. Vorteile liegen

beispielsweise in der netzübergreifenden Verwendung von ATM-Zellen als Basis-Protokolldateneinheit für Endsystem- und Vermittlungsknotenprotokolle, was die Entwicklung von Endgeräten und Netzkoppelementen (Router) wesentlich vereinfacht.

Beim Einsatz des ATM-Übertragungsverfahrens in Multi-Hop Mobilfunknetzen entstehen eine Vielzahl von z.T. ungelösten Problemstellungen, für die eine konsistente und integrierte Gesamtlösung entwickelt werden muß:

- Untersuchungen für Multi-Hop Paketfunknetze zeigen, daß beim Einsatz asynchroner Zugriffsverfahren in *teilvermaschten* Mobilfunknetzen deutliche Durchsatzverluste bei höherem Verkehrsangebot zu verzeichnen sind. Diese Verluste sind auf den Einfluß sog. versteckter Stationen in der Umgebung einer aktuell empfangenden Station zurückzuführen, die aufgrund der begrenzten Funkreichweite den Funkkanal als frei detektieren, deshalb zeitgleich zur bestehenden Übertragung senden, was potentiell zur Zerstörung des Datenpakets beim Empfänger führt, vgl. [100] und Abschn. 4.3.
- das ATM-Verfahren wurde für den Einsatz in Netzen mit hochwertigen Übertragungsmedien (z.B. Lichtwellenleiter) entworfen, die eine sehr niedrige Bitfehlerrate (Bereich $< 10^{-8}$) aufweisen. Der ATM-Protokollstapel ist zwar weitgehend unabhängig vom verwendeten Übertragungsmedium einsetzbar, vgl. [55], für Multi-Hop Funknetze wird jedoch ein Medien-Zugriffsprotokoll benötigt, das spezielle Eigenschaften des Funkkanals (z.B. höhere Bitfehlerwahrscheinlichkeit) berücksichtigt, dezentrale Verwaltung und Zugriff auf den Funkkanal usw. regelt.
- die existierenden ATM-Protokolle unterstützen keine vom Netz getragenen Verkehrsflußsteuerungsmechanismen in der Datenübertragungsphase einer (virtuellen) Verbindung wie wiederholte Übertragung von ATM-Zellen auf Verbindungsteilstrecken, dienstspezifische Übertragungsprioritäten, Schutz gegen Pufferüberlauf in den Vermittlungsknoten usw., die für eine optimierte Nutzung der typischerweise knappen Funkressourcen benötigt werden.
- ein wesentliches Leistungsmerkmal von ATM-basierten Netzen ist die adaptive, dienstspezifische Zuteilung von Übertragungskapazität. Mobilfunknetze auf Basis bekannter Multiplexverfahren wie TDM, FDM, CDM bieten solche Managementfunktionen nicht grundsätzlich an.

In Abschn. 4.3 wird verdeutlicht, daß der oben angesprochene Einfluß versteckter Stationen eine prinzipielle Problematik für teilvermaschte, dezentral organisierte Mobilfunknetze darstellt, die unabhängig vom verwendeten Übertragungsverfahren (synchron oder asynchron) auftritt und somit auch den effektiven Durchsatz bei kanalvermittelten Verbindungen reduziert. Im folgenden werden neue Verfahren und Protokolle beschrieben, vgl. [11], und anhand von Leistungsmessungen bewertet, die schwerpunktmäßig folgende Entwicklungsziele behandeln:

- flexible und dienstspezifische Zuteilung von Übertragungskapazität in FDM/TDM Multi-Hop Funknetzen,
- Einrichtung und Betrieb kanalvermittelter Verbindungen und zugehörige Managementfunktionen,
- Einsatz des ATM-Übertragungsverfahrens in Multi-Hop Funknetzen.

4.2 Verfahren zur adaptiven, dienstspezifischen Bandbreitenzuteilung

Physikalische Übertragungskanäle des RMAN werden durch eine Kombination von Zeit- und Frequenzmultiplexverfahren erzeugt: Das verfügbare Frequenzband wird durch Frequenzmultiplex in einzelne Teilbänder unterteilt, jedes Teilband durch Zeitmultiplex in eine Folge von Zeitschlitzten (Time-slots) aufgeteilt. Eine definierte Anzahl von Zeitschlitzten wird zu einem TDM-Rahmen zusammengefaßt, der sich periodisch wiederholt. Ein Übertragungskanal ist durch die verwendete Trägerfrequenz des Teilbandes und die Zeitschlitznummer innerhalb des zugehörigen TDM- Rahmens charakterisiert. Der TDM-Rahmen wird durch folgende Kenngrößen beschrieben: die Rahmendauer T_R , die Anzahl der Zeitschlitzte N_R und die Anzahl zu übertragender Bits pro Zeitschlitz B_{ZS} .

Die Gesamtübertragungsrate (Durchsatz) D pro Trägerfrequenz läßt sich wie folgt berechnen:

$$D = \frac{N_R B_{ZS}}{T_R} \quad [\text{bit/s}]$$

Den Durchsatz D_K eines Übertragungskanals, der durch einen Zeitschlitz des TDM-Rahmens realisiert wird, erhält man durch:

$$D_K = \frac{B_{ZS}}{T_R} \quad [\text{bit/s}]$$

Aus diesen Netzkenngößen ergibt sich aus Sicht der Sicherungsschicht (OSI-Schicht 2) zunächst eine feste Zuordnung von Übertragungskapazität zu den verfügbaren, physikalischen Kanälen. Um nun die Übertragungskapazität dynamisch für verschiedenste Dienste, Protokollstapel usw. verfügbar zu machen, wird ein logisches Kanalmuster auf das vorhandene physikalische Kanalraster aufgesetzt. Ein logisches Kanalmuster wird durch eine adaptiv veränderbare Anzahl logischer Kanäle gebildet. Für die Zuordnung von Übertragungskapazität zu logischen Kanälen existieren grundsätzlich drei Verfahren:

1. Die Übertragungskapazität eines TDM-Zeitschlitzes wird auf mehrere logische Kanäle aufgeteilt (Slotsplitting).
2. Ein logischer Kanal verwendet einen Zeitschlitz in jedem n -ten TDM-Rahmen ($n \geq 1$).
3. m Zeitschlitze ($m \geq 1$) eines TDM-Rahmens werden zu einem logischen Kanal zusammengefaßt.

Durch Anwendung der Verfahren 1 bzw. 2 entstehen logische Kanäle mit einem Durchsatz $D_{LK} \leq D_K$, wobei D_{LK} der Durchsatz eines logischen Kanals ist. Verfahren 3 definiert logische Kanäle mit einem Durchsatz $D_{LK} \geq D_K$.

Auf dieser Basis können anwendungsspezifisch verschiedene Kanaltypen mit jeweils unterschiedlichen Übertragungsraten definiert werden. Tab. 4.1 zeigt eine Auswahl logischer Kanaltypen, getrennt nach funktionalen Gruppen, deren Funktion im Rahmen der noch folgenden Protokollbeschreibungen erklärt wird.

Die vollständige Beschreibung eines logischen Kanals im RMAN besteht aus dem Kanaltyp, der Anzahl der verwendeten Zeitschlitze pro TDM-Rahmen und der Kennzeichnung seiner periodischen Wiederholung:

TCH1/2 definiert einen Verkehrskanal, der *einen* Zeitschlitz in jedem *zweiten* TDM-Rahmen belegt.

CPCH5/1 spezifiziert einen kontinuierlichen Paketdatenkanal, der *fünf* Zeitschlitze in *jedem* TDM-Rahmen belegt.

Tabelle 4.1: Logische Kanäle des RMAN

Gruppe	Kanaltyp	Abk.	Zugriff
Kanal- vermittlung	Signalisierungskanal	SCH	gemeinsam oder dediziert
	Verkehrskanal	TCH	dediziert
Paket- vermittlung	Zugriffskanal	ACH	gemeinsam oder dediziert
	Dedizierter Paketkanal	DPCH	dediziert
	Gemeinsamer Paketkanal	CPCH	gemeinsam
Netz- management	Organisationskanal	OCH	gemeinsam oder dediziert

Pro FDM-Teilband des Netzes kann somit eine flexible Kanalstruktur festgelegt werden, die aus einer Kombination verschiedenster logischer Kanäle besteht und ggf. während des Netzbetriebs verändert werden kann.

Abb. 4.1 zeigt eine mögliche Kanalkonfiguration für eine Trägerfrequenz des RMAN.

Die zyklische Wiederholung des Kanalmusters wird durch den logischen Kanal mit der geringsten Wiederholungsrate bestimmt: im gezeigten Beispiel ist dies ein Verkehrskanal (TCH1/3), der in jedem dritten TDM-Rahmen auftritt. Die gesamte Kanalkombination wiederholt sich im vorliegenden Fall periodisch nach drei TDM-Rahmen, die eine (logisch) übergeordnete Rahmenstruktur, den RMAN-Multirahmen, bilden. Die benötigte Funktionalität zur Abbildung des logischen Kanalmusters auf die physikalische Zeitschlitzstruktur wird als zusätzlicher Bestandteil der RMAN-Bitübertragungsschicht definiert. Medienzugriffsprotokolle der Sicherungsschicht können prinzipiell unabhängig von der TDM-Struktur des Netzes entworfen/eingesetzt werden, sie greifen ausschließlich auf logische Kanäle zu.

Wesentliches Merkmal der logischen Kanalstruktur ist, daß logische Kanäle für den synchronen und asynchronen Zugriff festgelegt und somit verschiede-

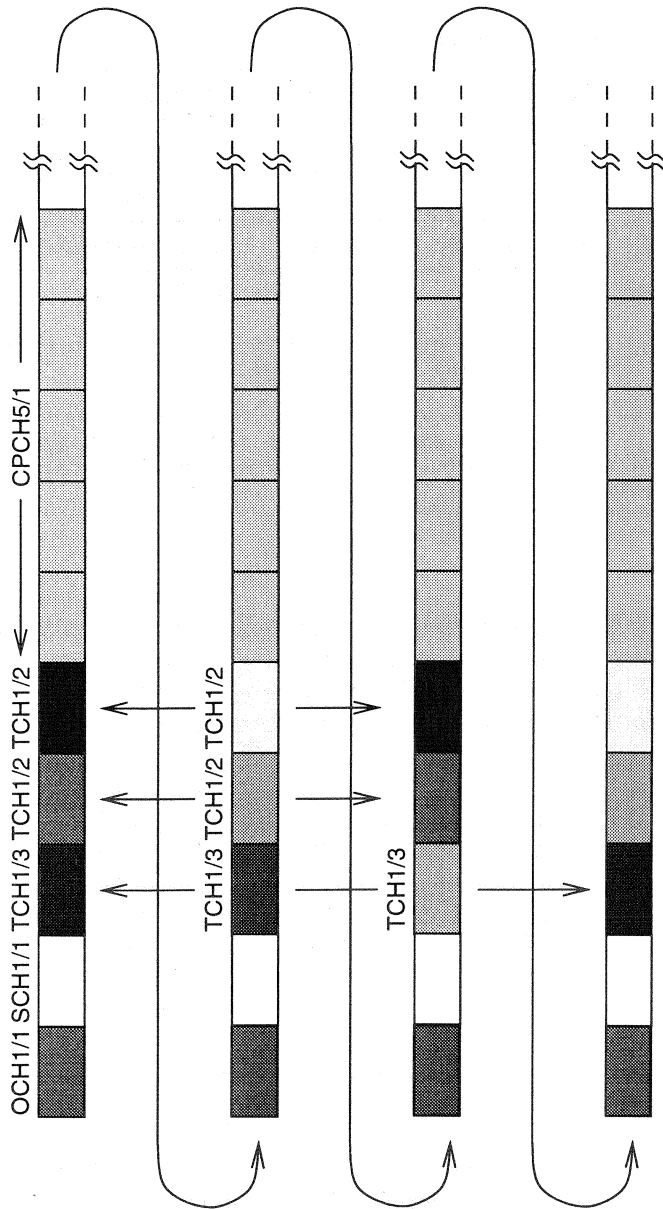


Abbildung 4.1: Beispiel: logisches Kanalmuster eines RMAN

ne Schicht 2a Protokolle wie S-Aloha (synchron) oder CSMA¹ (asynchron) eingesetzt werden können. Verwaltungsaufgaben bezüglich der logischen Kanalstruktur werden vom stationsinternen Managementsystem übernommen, vgl. Abschn. 2.6. Jede Netzstation führt pro Teilband eine *Kanalbelegungsliste*, die pro logischem Kanal u.a. folgende Einträge enthält:

- Kanaltyp und zugehörige Merkmale wie Durchsatz, relative Lage innerhalb des TDM-Rahmens, Blockwiederkehrzeit usw.,
- Kanalstatus, z.B. belegt, temporär gesperrt, akt. Signalpegel,
- Systemzeit des letzten Zugriffs,
- relevante Parameter des Medienzugriffsprotokolls.

Alle zeitlich veränderlichen Kenngrößen der Kanalbelegungsliste werden anhand von Meßwertreporten des Managementsystems oder Managementnachrichten anderer Netzstationen zyklisch auf den aktuellen Stand gebracht.

4.3 Störeinfluß versteckter Stationen in teilvermaschten Mobilfunknetzen

Abb. 4.2 zeigt eine typische Teiltopologie eines Multi-HopFunknetzes.

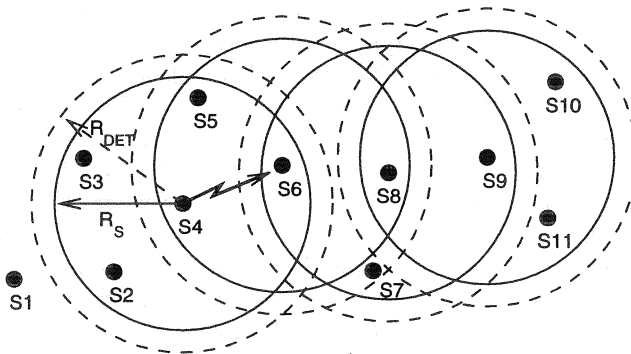


Abbildung 4.2: Sende- und Detektionsradius R_s , R_{DET} einer Station

¹Carrier Sense Multiple Access Protocol, vgl. [46].

Es wird angenommen, daß alle Stationen des Netzes mit rundstrahlenden Sende- bzw. Empfangsantennen ausgestattet sind. Eine Station S_i akzeptiert diejenigen Stationen als Nachbarstationen, deren Sendungen sie mit einem ausreichend hohen Signalpegel empfängt, was gleichbedeutend mit Signal-/Rauschleistungsverhältnis S/N ist, das einen vorgegebenen Schwellwert S/N_{Nachbar} bei der Festlegung der Konnektivität überschreitet. Die Aktualisierung konnektivitätsbezogener Größen und Verzeichnisse, wie Anzahl bzw. Adressen der Nachbarstationen, Teilnehmerverzeichnis, Wegelenkungsinformationen usw., erfolgt periodisch durch das stationsinterne Managementsystem; sie basiert auf der Auswertung lokaler Pegelmessungen einerseits und Managementinformationen andererseits, die zwischen Stationen (im RMAN über Organisationskanäle (OCH)) ausgetauscht werden.

Betrachtet man das Netz in der Ebene, so bildet sich idealerweise ein kreisförmiges Gebiet um jede Station S_i aus, innerhalb dessen die übrigen Stationen als Nachbarn erkannt werden. In der Praxis ist dieses Gebiet aufgrund von Störungen der Funkübertragung, z.B. durch Abschattungen, Mehrwegeausbreitung, Gleich- und Nachbarkanalstörer unregelmäßig und nur näherungsweise kreisförmig. Analoge kreisartige Gebiete entstehen pro Station für den Senderadius R_S und den Detektionsradius R_{DET} . Der Senderadius R_S gibt den maximalen Abstand zu einer sendenden Station an, bis zu dem eine gesendete Nachricht mit ausreichend kleiner Bitfehlerrate empfangen werden kann. Der Detektionsradius R_{DET} gibt die maximale Entfernung zu einem Sender an, bis zu dem seine Übertragung auf dem Funkkanal noch detektiert werden kann. Bedingt durch die dezentrale Kanalverwaltung/das Raummultiplex wird in bestimmter Entfernung zu einer sendenden Station derselbe Kanal/Zeitschlitz von anderen Stationen wiederverwendet. Abb. 4.2 zeigt eine Station S_4 , die ein Datenpaket zu einem Empfänger S_6 überträgt. Der Sendevorgang wird von allen Stationen innerhalb des Detektionsradius von S_4 registriert. Station S_8 befindet sich beispielsweise außerhalb der Detektionsfläche des Senders S_4 und kann den Kanal/Zeitschlitz für eine gleichzeitige Übertragung, z.B. zu Station S_9 , verwenden, was zu einem erhöhten Störpegel beim Empfänger S_6 führt. Stationen wie S_8 werden als *versteckte Stationen* bezeichnet.

Senden eine oder mehrere versteckte Stationen in der Umgebung eines Empfängers S_6 gleichzeitig zur Übertragung von S_4 nach S_6 , so ist das von S_6 empfangene Paket mit hoher Wahrscheinlichkeit zerstört. Dieser Effekt ist umso stärker, je weiter die empfangende Station (hier S_6) von der sendenden Station (S_4) entfernt ist. Bisher bekannte Medienzugriffsprotokolle wie

S-Aloha, CSMA oder das in der Standardisierung befindliche HIPERLAN-Protokoll EY-NPMA, vgl. [30], bieten keinen Schutzmechanismus gegen den Störeinfluß versteckter Stationen an.

4.4 Protokoll für den Betrieb kanalvermittelter Verbindungen

Bei kanalvermittelten Verbindungen belegen die beteiligten Stationen einen Kanal während der gesamten Dauer der Kommunikationsbeziehung. Verbindungen zwischen zwei nicht direkt, sondern nur über Relais verbindbaren Stationen, werden schrittweise (hopweise) aus Sicht des Verbindungsinitiators zum Adressaten aufgebaut. Die Verbindungseinrichtung zwischen zwei Teilnehmern umfaßt somit die Einrichtung von N Teilstreckenverbindungen, wobei N gleich der Anzahl der Teilstreckenverbindungen zwischen Initiator und Adressat ($N \geq 1$) ist. Das hier beschriebene Protokoll verwendet einen Signalisierungskanal (SCH) und Verkehrskanäle (TCH) entsprechend Tab. 4.1. Wesentliches Merkmal des entwickelten Protokolls ist, daß die dedizierten Verkehrskanäle von beiden an einer Teilstreckenverbindung beteiligten Stationen im Zeitgetrenntlageverfahren genutzt werden, d.h. sie senden abwechselnd auf demselben Verkehrskanal. Dies führt in Verbindung mit dem Signalisierungs- und Kanalmanagement zur Ausbildung einer Schutzzone um beide kommunizierenden Stationen, wodurch eine deutliche Reduktion des Störeinflusses versteckter Stationen, vgl. Abschn. 4.3, erreicht werden kann. Die Einrichtung einer kanalvermittelten Verbindung wird beispielhaft anhand eines teilvermaschten RMAN beschrieben, das in Abb. 4.3 dargestellt ist. Es wird angenommen, daß Teilnehmer A (Quelle) an Station S1 eine Verbindung zu Teilnehmer B (Ziel) an Station S8 einrichten möchte. Die Wegfindung durch das Netz vom Quell- zum Zielteilnehmer erfolgt pro Teilstreckenverbindung anhand der stationslokalen Routingtabellen, vgl. Abschn. 2.6.3. Für das vorliegende Beispiel sei angenommen, daß die Routingtabellen auf Basis des Min-Hop Verfahrens, vgl. [96], ermittelt werden. Für eine kanalvermittelte Verbindung zwischen Teilnehmer A und Teilnehmer B sind demnach sequentiell Teilstreckenverbindungen von S1 nach S2, S2 nach S4, S4 nach S6 und S6 nach S8 einzurichten.

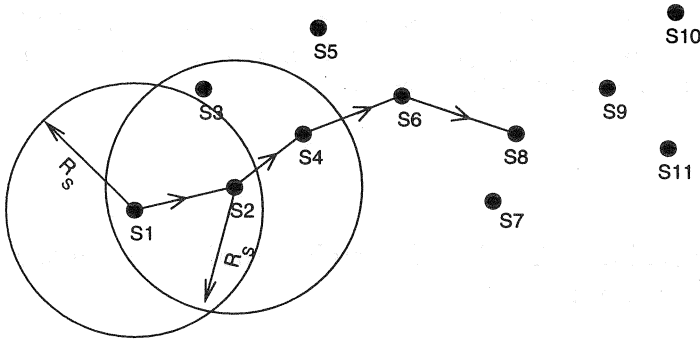


Abbildung 4.3: Beispielnetz für Einrichtung/Betrieb kanalvermittelter Verbindungen

4.4.1 Protokoll für Einrichtung und Betrieb von Teilstreckenverbindungen

In der ersten Signalisierungsphase wird zur Nachrichtenübertragung ein Signalisierungskanal (SCH) verwendet, auf den gemäß S-Aloha Protokoll, vgl. [96], oder einem äquivalenten, synchronen Medienzugriffsprotokoll zugegriffen wird. Die Station S1 sendet ein Signalisierungspaket vom Typ 'Verbindungswunsch-Teilstrecke' an Station S2, dessen Elemente u.a. die folgenden Informationen enthalten:

- Identifikation des Pakets,
- Adresse des Quell- und Zielteilnehmers,
- Adresse des Senders (S1) und der adressierten Relaisstation (S2),
- Liste von Verkehrskanälen, die aus Sicht des Senders für die Teilstreckenverbindung verwendet werden können,
- Signalisierungsinformationen der Netzschicht, wie geforderter Dienst, Dienstgüteparameter usw.

Die Liste freier Verkehrskanäle für eine Teilstreckenverbindung stellt die Station S1 anhand ihrer Kanalbelegungsliste zusammen, die durch das stationslokale Netzmanagement anhand von Pegelmessungen ständig auf den aktuellen Stand gebracht wird. Nach Absenden des Signalisierungspakets markieren alle empfangenden Stationen im Sendebereich von S1, vgl. Abb. 4.3, alle Verkehrskanäle der Liste für einen Zeitraum T1 als belegt. Station S1

hört diese Kanäle während des Zeitintervalls T_1 ab, um eine Antwort von Station S2 zu erhalten. T_1 ist als protokollspezifischer Parameter allen Stationen bekannt und wird typischerweise als Protokolltimer realisiert. Die Relaisstation S2 vergleicht die im empfangenen Signalisierungspaket enthaltene Liste der Kanalvorschläge mit ihrer lokalen Kanalbelegungsliste und prüft, ob einer der vorgeschlagenen Kanäle aus ihrer Sicht für die Teilstreckenverbindung verwendet werden kann. Im Falle einer positiven Prüfung, d.h. mindestens ein vorgeschlagener Kanal ist auch aus Sicht der Station S2 für die Verbindung verwendbar, werden zwei Fälle unterschieden:

1. Der (Teilstrecken-) Verbindungswunsch wird von Station S2 akzeptiert. Sie sendet auf dem ausgewählten Verkehrskanal ein Signalisierungspaket mit positiver Bestätigung zurück (Inband-Signalisierung).
2. Der (Teilstrecken-) Verbindungswunsch kann nicht akzeptiert werden, z.B. weil die geforderte Dienstgüte aufgrund von Interferenzstörungen nicht garantiert werden kann. Station S2 sendet eine negative Bestätigung auf dem ausgewählten Verkehrskanal.

Bei positiver Bestätigung nach 1. markieren alle Stationen im Sendebereich der Station S2 den von S1 und S2 benutzten Verkehrskanal als belegt. Danach beginnt Station S2 sofort mit der Einrichtung der nächsten Teilstreckenverbindung (hier nach S4) entsprechend dem zuvor beschriebenen Verfahren. Die Einrichtung von Teilstrecken-Verbindungen wird sequentiell fortgesetzt, bis eine N-Hop Verbindung (hier: $N=4$) zwischen Teilnehmer A und B verfügbar ist.

Nachdem eine Teilstreckenverbindung eingerichtet worden ist, werden, solange die Verbindung zwischen Quell- und Zielteilnehmer noch nicht vollständig eingerichtet ist, ständig Daten (z.B. Signalisierungsnachrichten, Managementnachrichten) über den Verkehrskanal zwischen den an einer Teilstreckenverbindung beteiligten Stationen ausgetauscht. Dieses Verfahren schützt den für die Teilstreckenverbindung verwendeten Verkehrskanal gegen Verwendung durch andere Stationen: durch die kontinuierliche und bi-direktionale Nutzung des Verkehrskanals entsteht eine Schutzzone um die an der Teilstreckenverbindung beteiligten Stationen S1 und S2, deren Fläche durch die Vereinigungsmenge der Kreisflächen mit dem Senderadius R_s gegeben ist, vgl. Abb. 4.3. Die Wirksamkeit der Schutzzone kann durch Erweiterung des stationsinternen Netzmanagement-Systems auf die Vereinigungsmenge der Detektionsflächen ausgedehnt werden: ein Verkehrskanal wird in der Kanalbelegungsliste erst dann als frei gekennzeichnet, wenn

Pegelmessungen für zwei aufeinanderfolgende (potentielle) Kanalzugriffe jeweils ein Signal-/Rauschleistungsverhältnis S/N_{Messung} aufweisen, das kleiner als der Detektionsschwellwert für Signale S/N_{DET} ist. Somit wird der Zugriff auf einen belegten Verkehrskanal durch solche Stationen verhindert, die sich außerhalb des Senderradius, aber innerhalb des Detektionsradius einer der beiden kommunizierenden Stationen befinden. Der Einsatz von Inkanal-Signalisierung vermindert zusätzlich das Kollisionsrisiko auf dem gemeinsam genutzten Signalisierkanal (SCH) und unterstützt die Aufrechterhaltung der Schutzzone durch Übertragungen auf den Verkehrskanälen (TCH).

4.4.2 Datenübertragungsphase und Verbindungsauslösung

Nach der Einrichtung aller für eine N-Hop Verbindung benötigten Teilstreckenverbindungen können Daten zwischen Quell- und Zielstation ausgetauscht werden. Sofern innerhalb der Datenübertragungsphase Pausen entstehen, werden die Teilstreckenverbindungen durch Übertragung von Informationsdaten entsprechend dem oben für die Verbindungseinrichtungsphase angegebenen Verfahren geschützt. Die Ende-zu-Ende Verbindung über N-Teilstrecken kann durch eine der beteiligten Stationen durch Senden eines Signalisierpakets vom Typ 'Verbindungsende' auf dem Verkehrskanal ausgelöst werden. Dies führt zu einer sukzessiven Auslösung aller verwendeten Teilstreckenverbindungen. Ein Kanal, der von einer Station während des Zeitintervalls T_U als unbenutzt beobachtet wird, in ihrer Kanalbelegungsliste jedoch als belegt markiert ist, wird nach Ablauf des Zeitintervalls T_U in der Kanalbelegungsliste als frei markiert. Diese Funktion erlaubt die dezentrale Freigabe von Kanälen bzw. Auslösung einer Teilstreckenverbindung durch Stationen, die an der Verbindung nicht beteiligt sind; es werden Blockierungen von Verkehrskanälen vermieden, die dadurch entstehen, daß zwei an einer Teilstreckenverbindung beteiligte Stationen diese nicht regulär, z.B. durch Verlust des Funkkontakts, auslösen können.

4.5 Protokolle zur Übertragung von ATM-Zellen

Vorteile und Probleme der Verwendung des ATM-Übertragungsverfahrens in dezentral organisierten Multi-Hop Funknetzen wurden in Abschn. 4.1 diskutiert. Diese Überlegungen haben im Rahmen dieser Arbeit zur Entwicklung des Differential Packet Reservation Multiple Access (DPRMA)

Protokolls geführt, das entsprechende Leistungsmerkmale zur Beseitigung dieser Problemstellungen aufweist. Ein Grundgedanke besteht darin, ATM-Zellen, als Dienstdateneinheiten der DPRMA-Protokollschicht, gekapselt in DPRMA-Protokolldateneinheiten (DPRMA-PDEs) paketvermittelt, entlang virtueller Verbindungen zwischen den Stationen des RMAN zu übertragen. Der Funktionsumfang des DPRMA-Protokolls läßt sich in zwei (gekoppelte) Protokoll-Bereiche untergliedern:

1. Protokoll zur Übertragung von DPRMA-Protokolldateneinheiten über One-Hop Teilstrecken zwischen benachbarten RMAN-Stationen.
2. Protokoll für Einrichtung, Betrieb und Auslösung virtueller ATM-Verbindungen zwischen beliebigen Stationen des RMAN.

4.5.1 DPRMA-Protokoll für One-Hop Teilstrecken

Die Entwicklung des DPRMA-Protokolls für One-Hop Teilstrecken basiert auf der Idee, dienstabhängig Kanalkapazität für die Übertragung von Protokolldateneinheiten zu reservieren und diese Übertragungskapazität durch statistisches Multiplexen mehrerer Quellströme optimal zu nutzen. Die Funktionsweise des Protokolls wird anhand eines typischen RMAN-Teilnetzes, vgl. Abb. 4.4, erklärt.

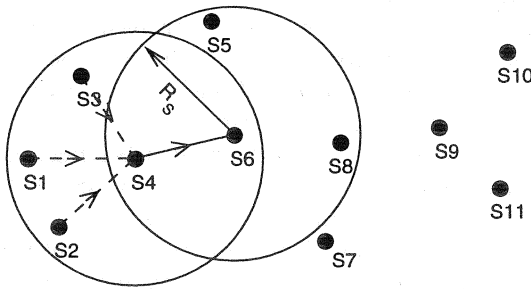


Abbildung 4.4: Beispielnetz für Einrichtung/Betrieb paketvermittelter Verbindungen

Es wird angenommen, daß Station S4 Paketdaten (DPRMA-Daten-PDEs) derselben Dienstklasse, die sie beispielsweise von den Stationen S1, S2, S3 empfangen hat, zu Station S6 übertragen will. Station S4 verwendet hierfür

reservierte Kanalkapazität. Der zugehörige Ablauf läßt sich in drei Phasen einteilen:

1. Reservierung von Übertragungskapazität für die One-Hop Teilstrecke,
2. Datenübertragung,
3. Freigabe der reservierten Kanalkapazität, sofern keine weiteren Daten zur Übertragung anstehen.

Das DPRMA-Teilstreckenprotokoll verwendet hierfür einen gemeinsam genutzten Zugriffskanal (Access Channel, ACH) und ein oder mehrere dedizierte Paketkanäle (Dedicated Packet Channel, DPC), vgl. Tab. 4.1.

Reservierung von Übertragungskapazität für One-Hop Teilstrecken

Station S4 beginnt mit der Übertragung eines Reservierungspakets (ACC-PDE), dessen struktureller Aufbau in Tab. 4.2 dargestellt ist, auf dem ACH-Kanal.

Der Zufallszugriff auf den von Netzstationen gemeinsam genutzten ACH-Kanal impliziert jedoch, daß ein Reservierungspaket (der Station S4) mit zeitgleichen Übertragungen anderer Stationen kollidieren kann. Aus diesem Grunde wurde in die ACC-Protokolldateneinheit ein *Selektionsbitmuster* eingefügt, vgl. Tab. 4.2, das eine dezentrale Zuteilung der Übertragungsbeziehung auf dem ACH-Kanal erlaubt und das Kollisionsrisiko erheblich reduziert: während der Sendung des Selektionsbitmusters konkurriert eine sendewillige Station mit den übrigen sendewilligen Stationen innerhalb ihres Senderadius R_s um den weiteren Zugriff auf den ACH-Kanal.

Für die dezentrale Auswahl einer (weiterhin) sendeberechtigten Station gilt folgende Regel: eine Station, die innerhalb der Übertragung des Selektionsbitmusters einen Bitwert 0 'senden' muß, hört während der 'Sendedauer' den ACH-Kanal ab. Sendet währenddessen eine andere Station (Bitwert = 1), so zieht sie ihren Sendewunsch bis zur nächsten Auswahlphase zurück. Die Sendeberechtigung wird also automatisch an diejenige Station eines Teilnetzbereichs vergeben, die während der Übertragung ihres Selektionsbitmusters keine 'Zurückweisung' ihres Sendewunsches entsprechend obiger Regel festgestellt hat.

Das Selektionsbitmuster besteht aus einer gleichverteilten N-Bit Zufallszahl, die jede Station pro Sendewunsch/Auswahlphase neu ermittelt. Eine Restkollisionswahrscheinlichkeit verbleibt, da unterschiedliche Stationen zufällig das gleiche Selektionsbitmuster ($\equiv$ Würfeln der gleichen Zufallszahl) ermit-

Tabelle 4.2: Aufbau der DPRMA-Protokolldateneinheiten

PDE- Bezeichnung	Kürzel	Parameter
Reservierung	ACC	Selektionsbitmuster, Paket-Identifikaton, Adresse Sender/Empfänger, Referenznummer der Verbindung, Liste der Kanalvorschläge, Dienstgüte
Bestätigung	ACK	Selektionsbitmuster, Paket-Identifikaton, Adresse Sender/Empfänger, Referenznummer der Verbindung,
Auslösung	DISC	Paket-Identifikaton, Adresse Sender/Empfänger, Referenznummer der Verbindung
Wiederholung	REP	Selektionsbitmuster, Paket-Identifikaton, Adresse Sender/Empfänger, Referenznummer der Verbindung, Referenznummern der Wiederholungs- zellen
Daten	DT	Paket-Identifikaton, Dienst(güte)parameter, Aktuelle Laufnummer, Laufzeit-Kontrolle, Daten (ATM-Zelle), Fehlererkennung

tern können. Die Auftrittswahrscheinlichkeit einer gleichverteilten Zufallszahl X der Länge N bit ist

$$p_A(X) = \frac{1}{2^N}$$

und die Wahrscheinlichkeit für das K -fache Auftreten von X bei M un-

abhängigen Versuchen

$$p^{(M)}(X) = \binom{M}{K} p_A^K q_A^{M-K}$$

Die Wahrscheinlichkeit für ein mindestens L-faches Auftreten der Zufallszahl X ergibt sich aus:

$$1 = P(X \leq L-1) + P(L-1 < X < L) + P(X \geq L)$$

Aufgrund der diskreten Verteilung ist

$$P(L-1 < X < L) = 0$$

woraus folgt

$$P(X \geq L) = 1 - P(X \leq L-1)$$

Die Restkollisionswahrscheinlichkeit $P(X \geq 2)$ in Abhängigkeit von der Stationszahl M innerhalb einer Kreisfläche mit dem Senderradius R_S und verschiedene Wertebereiche der Zufallszahl X zeigt Abb. 4.5.

Da die Stationszahl innerhalb eines Sendebereichs typischerweise weit unterhalb von 60 Stationen liegt, weist ein Selektionsbitmuster basierend auf einer 8 bit Zufallszahl eine genügend geringe Restkollisionswahrscheinlichkeit auf. Bei der Dimensionierung des Selektionsbitmusters ist jedoch zu beachten, daß die von einer Station ausgesendete Information aufgrund der Ausbreitungsverzögerung nicht an jedem Ort innerhalb ihres Senderradius R_S gleichzeitig empfangen und ausgewertet werden kann. Die Ausbreitungsverzögerung für einen Empfänger im Abstand R_S zum Sender beträgt

$$\Delta t_V = \frac{R_S}{c} \quad \text{mit} \quad c \approx 3 \cdot 10^8 \text{ m/s}$$

In Abhängigkeit von der Sendedauer pro Bit im Netz muß jeder Bitwert der Zufallszahl so häufig gesendet werden, daß alle Stationen innerhalb des Senderradius über eine gleichzeitige und ausreichend lange Detektionsphase verfügen. Ist beispielsweise die Bitübertragungsdauer t_B gleich der Ausbreitungsverzögerung Δt_V , so erhält man bei einem Verhältnis 1:3 der Zufallsbits zu Selektionsmusterbits ein sicheres Meßzeitintervall für alle empfangenden Stationen, vgl. Abb. 4.6.

Das ACC-Reservierungspaket enthält weiterhin die Adresse der Sender- und Empfängerstation, den Paketidentifikator (ACC-Paket), eine Liste von

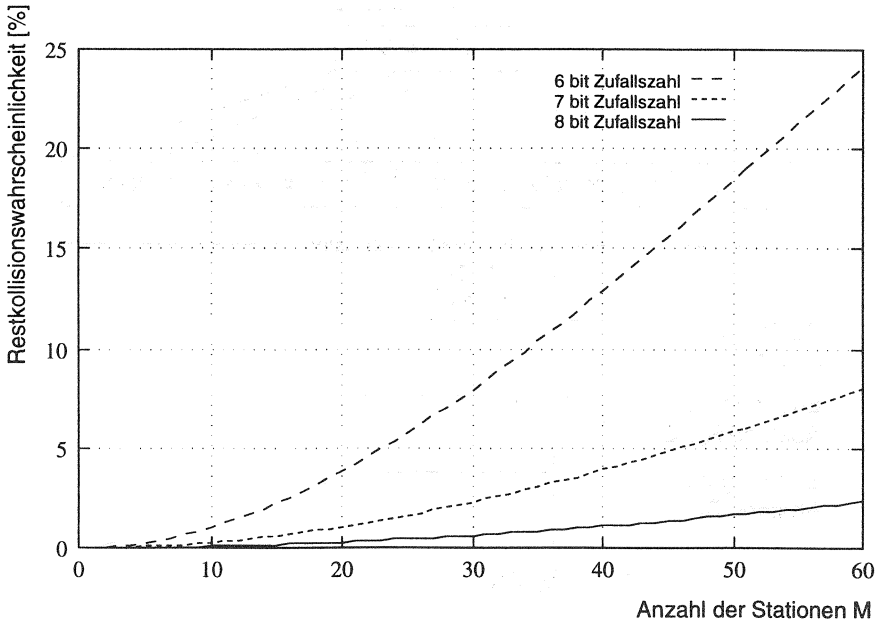


Abbildung 4.5: Kollisionswahrscheinlichkeit in Abhängigkeit von der Stationszahl M innerhalb des Sendebereichs

dedizierten Paketkanälen, die aus ihrer Sicht der Station S_4 für die Datenübertragung verwendet werden können, sowie Informationen über die geforderte Dienstgüte der Verbindung wie z.B. Anzahl benötigter Paketkanäle, Dienstart, Durchsatz, maximale Verzögerung. Erhält Station S_4 innerhalb eines Zeitintervalls T_A keine Antwort von Station S_6 , so wird die Übertragung des Reservierungspakets wiederholt; der nächste Zugriffszeitpunkt auf den ACH-Kanal kann durch Algorithmen des Netzmanagementsystems gesteuert werden, die eine Kollision mit Reservierungspaketen ebenfalls wiederholender Stationen vermeidet.

Es wird angenommen, daß Station S_4 , vgl. Abb. 4.4, die Sendeberechtigung für den ACH-Kanal erhalten hat und das Reservierungspaket erfolgreich von Station S_6 empfangen worden ist. Station S_6 vergleicht nach Empfang des Reservierungspakets die enthaltene Liste der Kanalvorschläge mit ihrer lokal geführten Kanalbelegungsliste. Ergibt sich aus dem Vergleich eine aus-

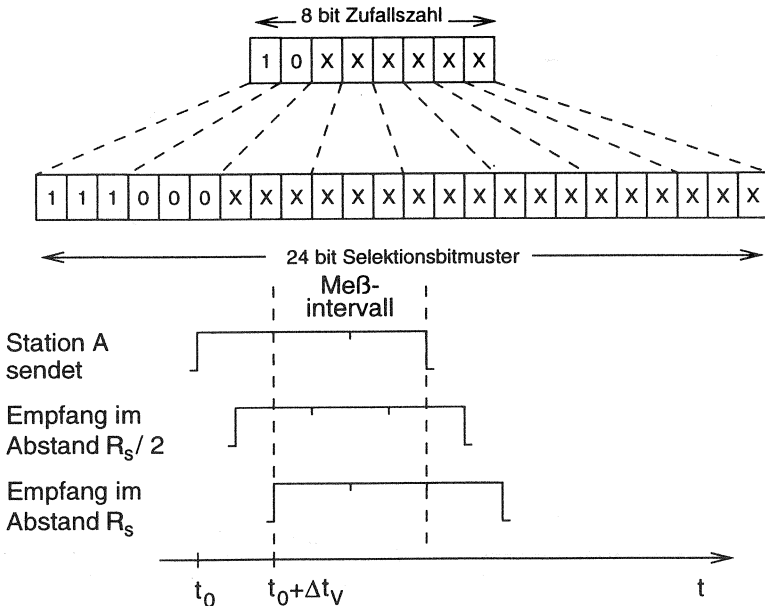


Abbildung 4.6: Zusammenhang zwischen Breite des Selektionsbitmusters und Meßzeitintervall der empfangenden Stationen (Annahme: $\Delta t_v = t_B$)

reichende Anzahl von Kanälen mit den geforderten Dienstgütemerkmalen, so sendet S4 auf den entsprechenden dedizierten Paketkanälen (DPCHs) ein Bestätigungspaket (ACK-Paket), vgl. Tab. 4.2. Das ACK-Paket enthält Prioritätsbits und ein Selektionsbitmuster zur dezentralen Vergabe des Belegungsrechts für dedizierte Paketkanäle (DPCHs) unter konkurrierenden Stationen entsprechend dem oben für den ACH-Kanal beschriebenen Verfahren.

Aktualisieren der Kanalbelegungslisten

Sendet eine Station, z.B. S4, erfolgreich ein Reservierungspaket auf dem gemeinsamen Zugriffskanal (ACH), markieren alle übrigen Stationen im Sendebereich von S4 die im ACC-Reservierungspaket angegebenen Paketkanäle

in ihrer stationslokalen Kanalbelegungsliste als *temporär* reserviert. Sofern auf einen oder mehrere dieser Kanäle innerhalb eines Zeitintervalls T_2 durch Station S4, S6 oder eine andere Station zugegriffen wird, markieren alle Stationen die entsprechenden Kanäle als belegt, andernfalls werden sie als frei in der Kanalbelegungsliste gekennzeichnet. Der Ablauf des Zeitintervalls T_2 wird durch einen Timer des DPRMA-Protokolls überwacht, sein Wert kann dynamisch durch das stationslokale Netzmanagementsystem, z.B. in Abhängigkeit der Netzauslastung, modifiziert werden. Entsprechend kennzeichnen alle Stationen im Sendebereich der Station S6 (adressierte Station) nach Mithören eines Bestätigungspakets den zugehörigen Kanal als belegt. Durch das Reservierungsverfahren bzw. Markieren belegter Kanäle in der stationslokalen Kanalbelegungsliste entsteht eine Schutzzone für die reservierten Paketkanäle (DPCH) um sendende und empfangende Station (S4 und S6), deren Fläche ungefähr durch die Vereinigungsmenge der Kreisflächen mit dem Detektionsradius R_{DET} (Sender) und Senderradius R_S (Empfänger) gegeben ist, vgl. Abb. 4.4. Die Schutzzonenausbildung, insbesondere im Rückraum des Empfängers, führt zu einer deutlichen Reduktion des Störeinflusses versteckter Stationen, vgl. Abschn. 4.3.

Datenübertragungsphase

Nachdem Station S4 ein oder mehrere Bestätigungspakete empfangen hat, werden Paketdaten (DPRMA-Daten-PDEs) auf den ausgewählten dedizierten Paketkanälen von Station S4 zu Station S6 übertragen. Wesentliches Merkmal des DPRMA-Teilstreckenprotokolls ist, daß die Anzahl reservierter DPCH-Kanäle für eine Teilstreckenverbindung adaptiv erhöht bzw. verringert werden kann. In Verbindung mit stationslokalen, dienstspezifischen Warteschlangen, vgl. Abschn. 4.5.2, können basierend auf Dienstgüteparametern, z.B. Nutzdatendurchsatz, Paket-/Zellverlustwahrscheinlichkeit, maximale Verzögerung, maximale Verzögerungsschwankung, ATM-Netzmanagementfunktionen wie Verbindungsannahmesteuerung (Connection Admission Control, CAC), Nutzungsparameterüberwachung (Usage Parameter Control, UPC) usw. realisiert werden.

Quittierung empfangener Datenpakete

Die Quittierung empfangener Datenpakete (DPRMA-Daten-PDEs) pro Hop übernimmt die Sequenzkontrolleinheit des Protokolls unter Zuhil-

fenahme des Informationsfeldes 'Aktuelle Laufnummer des Pakets', vgl. Abschn. 4.5.2, und der Fehlererkennungs-/Fehlerkorrektur-Funktionen des stationslokalen Netzmanagements. Der korrekte Empfang von Datenpaketen durch eine Empfängerstation, z.B. Station S6, wird von der sendenden Station, z.B. S4, vorausgesetzt solange sie keine explizite Wiederholungsanforderung von Station S6 empfängt: erhält die Empfängerstation (S6) ein fehlerbehaftetes, nicht korrigierbares Datenpaket oder stellt sie den Verlust eines Datenpaketes fest, so sendet sie ein Wiederholungspaket (REP-Paket), vgl. Tab. 4.2, zur sendenden Station S4.

Auslösen einer Verbindung

Eine bestehende One-Hop Teilstreckenverbindung ist aus Sicht der empfangenden Station (hier: S6) auszulösen, sofern sie innerhalb eines Zeitintervalls T_{DISC} keine Daten von der sendenden Station (S4) erhält. Allen übrigen Stationen innerhalb ihres Sendebereichs teilt Station S6 dies durch Senden eines DISC-Pakets, vgl. Tab. 4.2, auf allen für die Teilstreckenverbindung zwischen S4 und S6 reservierten Paketkanälen mit. Die Funkstationen im Sendebereich der Station S4, die das DISC-Paket der Station S6 nicht empfangen, aktualisieren ihre Kanalbelegungslisten nach Ablauf eines Zeitintervalls T_4 gemessen vom letzten Zugriff der Station S4 auf den jeweiligen Paketübertragungskanal. T_{DISC} und T_4 sind protokollspezifische Parameter und werden als Timer realisiert, ihre Werte können dynamisch unter Kontrolle des stationslokalen Netzmanagements, z.B. in Abhängigkeit von der Netzauslastung, verändert werden.

4.5.2 DPRMA-Protokoll für Einrichtung, Betrieb und Auslösung virtueller ATM-Verbindungen

In Abschn. 4.1 wurde bereits beschrieben, daß die existierenden ATM-Protokolle primär auf den Einsatz in Festnetzen mit hochwertigen Übertragungsmedien abgestimmt sind und vom Netz getragene Verkehrssteuerungsmechanismen für die Datenübertragungsphase nur auf Basis von Verkehrsmessungen realisiert werden können. Der transportierte Zellenstrom einer Verbindung unterliegt somit nur einer indirekten Kontrolle des ATM-Netzmanagements bezüglich der Einhaltung seiner verbindungsbezogen ausgehandelten Dienstgüteparameter. Werden die für eine Verbindung während der Einrichtungsphase festgelegten Kenngrößen (kurzfristig) über-

schritten, so kann die Dienstgüte für parallel existierende gegebenenfalls nicht mehr garantiert werden. Das DPRMA-Protokoll trägt dieser Problematik dadurch Rechnung, indem es ATM-Zellen gekapselt in DPRMA-Daten-PDEs überträgt, die zusätzliche Informationen für die Verkehrssteuerung auf (Relais-) Stationsebene enthalten. Eine virtuelle Ende-zu-Ende Verbindung zwischen Quell- und Zielteilnehmer wird über N One-Hop Teilstrecken ($N \geq 1$) eines RMAN geführt. Die beteiligten (Relais-) Stationen ermitteln während der Einrichtung von One-Hop Teilstrecken dezentral eine günstige Route zwischen Quell- und Zielteilnehmer anhand ihrer stationslokalen Routingtabellen. Der Aufbau einer DPRMA-Daten-PDE ist in Tab. 4.2 dargestellt, das neben der eigentlichen ATM-Zelle folgende Informations-elemente enthält:

1. Das Informationselement 'Paketidentifikator' kennzeichnet den Typ des Datenpakets. Es wird zwischen Paketen für Einrichtung (Connection Request, CR), Einrichtungsbestätigung (Connection Confirm, CC) und Auslösung (Disconnect Request, DR) virtueller Ende-zu-Ende Verbindungen sowie für den Datentransfer (Data, DT) unterschieden.
2. Angabe des Dienstes, der Dienstcharakteristika, z.B. kontinuierliches oder büschelartiges Verkehrsaufkommen, und der geforderten Dienstgüte wie Durchsatz, maximale Verzögerung, Priorität usw., zur Realisierung einer optimalen Datenflußsteuerung auf One-Hop Teilstrecken z.B. durch Einführung dienstspezifischer Übertragungsprioritäten, intelligenter Verbindungsannahmesteuerung, Verkehrsflußkontrolle usw.
3. Das Informationsfeld 'Aktuelle Laufnummer des Pakets' dient den Stationen zur Reihenfolgekontrolle empfangener Pakete und wird für (dienstspezifische) Wiederholungsanforderungen, Ausgleich unterschiedlicher Paketlaufzeiten bei verkehrsabhängiger alternativer Routenwahl zwischen Sender und Empfänger usw. verwendet.
4. Das Informationsfeld 'Laufzeit-Kontrolle' enthält die bisher benötigte Übertragungszeit (in Einheiten von z.B. Hops, TDMA-Rahmen usw.) vom Sender in Richtung des Empfängers. Es wird von den an einer Multi-Hop Verbindung beteiligten Stationen aktualisiert und berücksichtigt auch eventuelle Sendeverzögerungen durch Pufferung, Priorisierung anderer Datenpakete etc. in den Relaisstationen. Die Relaisstationen einer Multi-Hop Verbindung erhalten somit die Möglichkeit,

zu lange verzögerte Datenpakete vor der weiteren Übertragung nach folgendem Verfahren zu erkennen und ggf. zu verwerfen: Während der Verbindungseinrichtungsphase zwischen Quell- und Zielstation der Multi-Hop Verbindung werden von jeder der beteiligten Relaisstationen verbindungspezifisch die folgenden Kennwerte gespeichert:

- benötigte Übertragungszeit TTI für Datenpakete von dieser Relaisstation bis zur Verbindungsquelle,
- benötigte Übertragungszeit TTA für Datenpakete von dieser Relaisstation bis zur Verbindungssenke,
- maximal zulässige Verzögerungszeit für Datenpakete zwischen Quelle und Senke $TDA_{\max}$ und umgekehrt $TDI_{\max}$.

TTI und TTA sind Mittelwerte von Zufallsvariablen der entsprechenden Dienstarten, deren Varianz und höhere Momente aufgrund periodischer Messungen des Netzmanagementsystems ermittelt werden. Anhand der Momente werden korrespondierende Kenngrößen $TTI_{\%}$ und $TTA_{\%}$ definiert, wobei $TTI_{\%} > TTI$ und $TTA_{\%} > TTA$ Quantile der zugehörigen Verteilungsfunktionen sind. Zu stark verzögerte Datenpakete können von einer Relaisstation durch positive Prüfung einer der folgenden Gleichungen erkannt werden (TRT sei der im Informationsfeld für die Laufzeitmessung eines zu versendenden Datenpakets enthaltene Wert):

In Richtung vom Quell- zum Zielteilnehmer:

$$TRT + TTA > TDA_{\max}$$

bzw.

$$TRT + TTA_{\%} > TDA_{\max}$$

In Richtung vom Ziel- zum Quellteilnehmer:

$$TRT + TTI > TDI_{\max}$$

bzw.

$$TRT + TTI_{\%} > TDI_{\max}$$

Pakete werden solange über One-Hop Teilstrecken übertragen, bis die für die zugehörige virtuelle Verbindung festgelegte maximale Verzögerungszeit ($TDI_{\max}$ bzw. $TDA_{\max}$) überschritten wurde. Ebenso wird bezüglich der Wiederholung fehlerbehafteter Datenpakete verfahren.

Pakete, die zu Verlust gegangen sind bzw. deren maximale Verzögerungszeit überschritten wurde, werden separat gezählt, die Meßwerte periodisch an das Netzmanagementsystem übergeben.

5. Das Informationsfeld 'Fehlererkennung' wird verwendet, um Paketfehler, die aufgrund von Funkkanalstörungen entstehen, zu erkennen bzw. korrigieren; es können ARQ- (Automatic Repeat Request), FEC- (Forward Error Correction) oder Kombinationen beider Verfahren eingesetzt werden. Zur Realisierung einer hohen Dienstgüte wird die Länge des Informationsfeldes 'Fehlererkennung' adaptiv unter Kontrolle des Netzmanagementsystems, z.B. in Abhängigkeit von der Netzauslastung, verändert. Die Messung der Netzauslastung bzw. Bestimmung der Restübertragungskapazität für die Fehlererkennung/-korrektur wird periodisch von Netzstationen durchgeführt. Der adaptive Codierungstyp für das aktuelle Datenpaket ist im Informationsfeld 'Fehlererkennung' angegeben.

Die Informationselemente der DPRMA-Daten-PDEs werden vom stationslokalen Netzmanagementsystem zur Verkehrs- und Überlastkontrolle herangezogen. Daneben werden von jeder Station pro virtueller Verbindung, die über sie geführt wird, zusätzlich folgende Daten gespeichert bzw. verwaltet:

- Diensttyp, Dienstgüte und Dienstcharakteristika, die in den entsprechenden Informationselementen der DPRMA-Daten-PDE für die Einrichtung einer virtuellen Ende-zu-Ende Verbindung enthalten sind.
- Sende- und Empfangszähler werden für beide Richtungen (Quell- $\Leftrightarrow$ Zielstation) getrennt verwaltet. Der aktuelle Zählerstand gibt die Nummer der nächsten zu sendenden bzw. empfangenden DPRMA-Daten-PDE in der zugehörigen Richtung an. In Verbindung mit dem Informationsfeld 'Aktuelle Laufnummer des Pakets' wird der Verlust von Datenpaketen festgestellt und eine Sequenzkontrolle durchgeführt.
- Die maximale Sperrzeit spezifiziert den Zeitraum, für den der Sperrvermerk einer Daten-PDE in der Senderwarteschlange gültig ist: Wird der Verlust eines Datenpakets registriert, so werden alle bereits empfangenen Datenpakete dieser Verbindung mit höherer Laufnummer für die weitere Übertragung gesperrt. Der Sperrvermerk wird gelöscht, sofern die zu Verlust gegangene Daten-PDE nach einer Wiederholungsanforderung erfolgreich empfangen wurde oder das Sperrzeit-Intervall abgelaufen ist. Im letzteren Fall entscheidet das stationslokale Netzmanagement in Abhängigkeit von den Dienstgüteparametern bezüglich

der Reaktion auf diesen Fehlerfall, z.B. ob der Datenverlust toleriert, eine weitere Wiederholungsanforderung gesendet oder die Verbindung ausgelöst wird.

- Die Kenngrößen der Laufzeitkontrolle TTI , $TTI_{\max}$ bzw. TTA , $TTA_{\max}$, vgl. Informationselemente der DPRMA-Daten-PDE.
- Die Adresse der nächsten (Relais-) Station in Richtung der Quelle wird benötigt, um die Ende-zu-Ende Signalisierungsinformationen der Zielstation auf derselben Route durch das Netz zu übertragen wie die Signalisierungsinformationen von der Quell- zur Zielstation. Dies ist erforderlich, um den beteiligten Relaisstationen eine Aktualisierung der verbindungsbezogenen Daten, z.B. bei Bestätigung der Verbindungseinrichtung oder Verbindungsauslösung, zu ermöglichen.

Jede Station verwaltet pro Nachbarstation einen Satz von Warteschlangen, in der empfangene Daten-PDEs, die in die entsprechende Richtung weiterzuvermitteln sind, zwischengespeichert werden, vgl. Abb. 4.7. Ein richtungsbezogenes Warteschlangenbündel besteht aus Warteschlangen, die sich bezüglich des Diensttyps der darin gespeicherten Datenpakete unterscheiden. Unter einem Diensttyp werden Dienste mit gleichartigen Anforderungen bezüglich der Dienstgüte und -charakteristika wie Durchsatz, Verlustwahrscheinlichkeit, maximale Verzögerung usw. zusammengefaßt. Jeder Warteschlange ist ein Sicherungspuffer zugeordnet, in dem Datenpakete für einen bestimmten Zeitraum nach der Übertragung zur nächsten Station gespeichert werden, um auf Wiederholungsanforderungen reagieren zu können. Eine Warteschlange wird durch einen Satz von Parametern beschrieben, die eine dienstspezifische Verkehrsflußsteuerung durch das stationslokale Netzmanagement unterstützen:

- **Kanaltyp:** Dieser Parameter spezifiziert den für den jeweiligen Diensttyp zu verwendenden dedizierten Paketkanaltyp. Die verfügbaren Kanalressourcen können effektiv und adaptiv auf die vom Netz angebotenen Dienste mit typischerweise unterschiedlichen Bandbreitenanforderungen aufgeteilt werden. Weiterhin wird ausgeschlossen, daß ein Dienst dedizierte Paketkanäle eines für ihn nicht spezifizierten Typs reserviert und die Verfügbarkeit anderer Dienste beeinträchtigt bzw. blockiert.
- **Maximale Kanalzahl pro Teilstrecke:** Diese Kenngröße legt fest, wieviele dedizierte Paketkanäle des vorgegebenen Kanaltyps für eine One-Hop Teilstrecke reserviert werden können. Die Begrenzung der

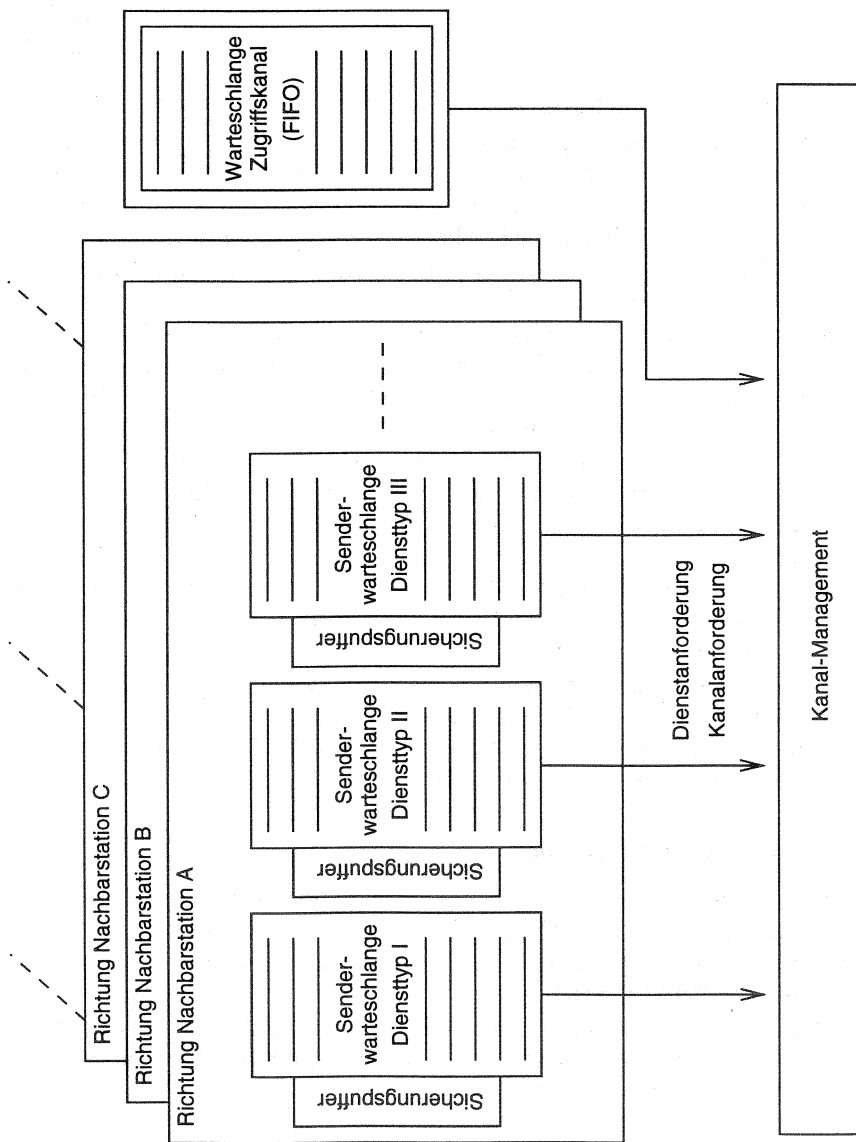


Abbildung 4.7: Warteschlangenstruktur auf Stationsebene

maximalen Kanalzahl verhindert, daß die verfügbaren Paketkanäle innerhalb eines räumlich begrenzten Gebietes (Gebiet ohne Raummultiplex) ausschließlich von einer bzw. zwei Netzstation reserviert werden, und beugt somit Blockierungen vor.

- **Kanal Anforderungsschwellwert:** Enthält die Warteschlange eine Anzahl von Datenpaketen, die den Kanal Anforderungsschwellwert übersteigt, so wird ein weiterer Kanal mit Hilfe einer ACC-Protokolldateneinheit, vgl. Abschn. 4.5.1, angefordert. Die Kanal Anforderung wird akzeptiert, wenn die maximale Kanalzahl noch nicht erreicht ist, andernfalls abgewiesen.
- **Kanalfreigabeschwellwert:** Bei Unterschreiten dieses Schwellwertes durch die Anzahl der Warteschlangenelemente wird, sofern die Anzahl reservierter Kanäle größer 1 ist, ein Kanal senderseitig freigegeben. Durch die Differenz zwischen Anforderungs- und Freigabeschwellwert wird ein Übergangsbereich definiert, bei dem nach Unterschreiten der Kanal Anforderungsschwelle ein Kanal noch nicht freigegeben wird, wodurch ein permanentes Anfordern und Freigeben von Kanälen vermieden wird.
- **Sicherungspuffergröße:** Die Größe des benötigten Sicherungspuffers ist dienstabhängig und wird durch Kenngrößen wie maximale Kanalzahl, Kanaltyp, Dienstgüteparameter usw. bestimmt. Unter Kontrolle des Netzmanagementsystems kann somit die Länge des Sicherungspuffers in Abhängigkeit von der Netzauslastung modifiziert und Kenngrößen wie z.B. die Verlustwahrscheinlichkeit auf One-Hop Teilstrecken adaptiv verändert werden.

4.5.3 Zusammenfassung

In den vorangegangenen Abschnitten wurden neue Verfahren und Protokolle vorgestellt, die eine dem RMAN gegebene Übertragungskapazität adaptiv für verschiedene Dienstarten entsprechend den aktuellen Teilnehmeranforderungen verfügbar machen.

Die beiden folgenden Kapitel enthalten Beschreibungen des RMAN-Simulators zur Leistungsbewertung sowie eine Darstellung von Simulationsergebnissen für verschiedene Szenarien.

Der RMAN-Simulator zur Leistungsbewertung

5.1 Entwurfskriterien und Realisierung

Die Entwicklung des RMAN-Simulators zur Leistungsbewertung dezentral organisierter Multi-Hop Funknetze erfolgte innerhalb der vorliegenden Arbeit. Die resultierende Systemarchitektur berücksichtigt neben den technischen Anforderungsmerkmalen eine Vielzahl softwaretechnischer Entwurfskriterien und Entwicklungsziele wie z.B.

- Modularität und Erweiterbarkeit,
- Portabilität,
- Speichereffizienz,
- Laufzeiteffizienz.

Modularität und *Erweiterbarkeit* wurden durch ein streng definiertes Schnittstellenkonzept realisiert, d.h. logische Einheiten bzw. Komponenten des Simulators kommunizieren nur über flexibel ausgelegte Schnittstellenprotokolle miteinander. Dies erhöht die Transparenz des komplexen Simulationswerkzeugs und erlaubt eine weitgehend parallele Entwicklung von Einzelkomponenten.

Der Forderung nach *Portabilität* wurde dadurch Rechnung getragen, daß alle Simulatorkomponenten in der Programmiersprache C nach ANSI¹-C Standard implementiert und die Anzahl verwendeter Bibliotheksfunktionen auf ein Minimum sowie Standardbibliotheken beschränkt wurde. Der Entwurf eines Konzepts zur ausschließlichen Verwendung virtueller Datentypen und seine konsequente Anwendung innerhalb des gesamten Quelltextes vermeidet typische Portierungsprobleme, die aufgrund unterschiedlicher Datentyp-Darstellungen bei verschiedenen Hardwareplattformen entstehen.

¹American National Standards Institute

Anforderungen bezüglich *Speicher-* und *Laufzeiteffizienz* lassen sich nicht gleichermaßen erfüllen, Verbesserungen der einen gehen zu Lasten der anderen Kenngröße: eine hinsichtlich der Speicherbelegung hochdynamische und effiziente Anwendung weist einen nicht unbeträchtlichen Rechenzeitbedarf für die Allokierung bzw. Deallokierung von Hauptspeicher sowie Speicherbedarf für die Verwaltung von Referenzzeigern (Zeigerlisten, Verkettungen usw.) auf. Die Verwendung statischer Datenstrukturen ermöglicht einen effizienten Datenzugriff, die Flexibilität bezüglich der Konfigurierbarkeit wird jedoch stark eingeschränkt, so daß bei Konfigurationsänderungen häufig eine Re-Kompilation des Quelltextes erforderlich ist. Ein hoher statischer Hauptspeicherbedarf mit geringem Nutzungsgrad wirkt sich bei Multiprozess-Maschinen ebenfalls negativ auf das Laufzeitverhalten aus, sofern das Betriebssystem aufgrund hoher Speicherauslastung, auch bedingt durch andere Prozesse, häufig Daten in und aus dem virtuellen Systemspeicher (Swap Space) transferieren muß.

Bei der Realisierung des RMAN-Simulators wurde jeweils anhand der konkreten Problemstellung und unter Berücksichtigung der oben genannten Kriterien zugunsten statischer oder dynamischer Datenstrukturen entschieden. Weiterhin wurde eine ereignisorientierte Simulationssteuerung realisiert, vgl. Abschn. 5.2, was sich insbesondere in der Einschwingphase nach dem Start einer Simulation positiv auf das Laufzeitverhalten auswirkt.

5.2 Software-Architektur des RMAN-Simulators

Ziel der Simulation ist es, verlässliche Aussagen über Verhalten und Leistungsfähigkeit der entworfenen Protokolle zu gewinnen. Bei der Realisierung des RMAN-Simulators wurde nur soweit im simulationstechnischen Sinne notwendig Gebrauch von Modellen (Lastmodelle, Kanalmodell) gemacht, die Kommunikationsprotokolle wurden beispielsweise vollständig implementiert. Abb. 5.1 zeigt die Architektur des RMAN-Simulators im Überblick. Der Simulator besteht aus unabhängigen, miteinander kommunizierenden und nach funktionalen Gesichtspunkten gegliederten Modulen, die aus Blöcken zusammengesetzt sind.

Die ereignisorientierte Simulationssteuerung (Scheduler) koordiniert als zentrales Modul den gesamten Simulationslauf:

- die Simulationszeit schreitet in Vielfachen eines TDM-Rahmens fort, vgl. Abschn. 2.6.5.

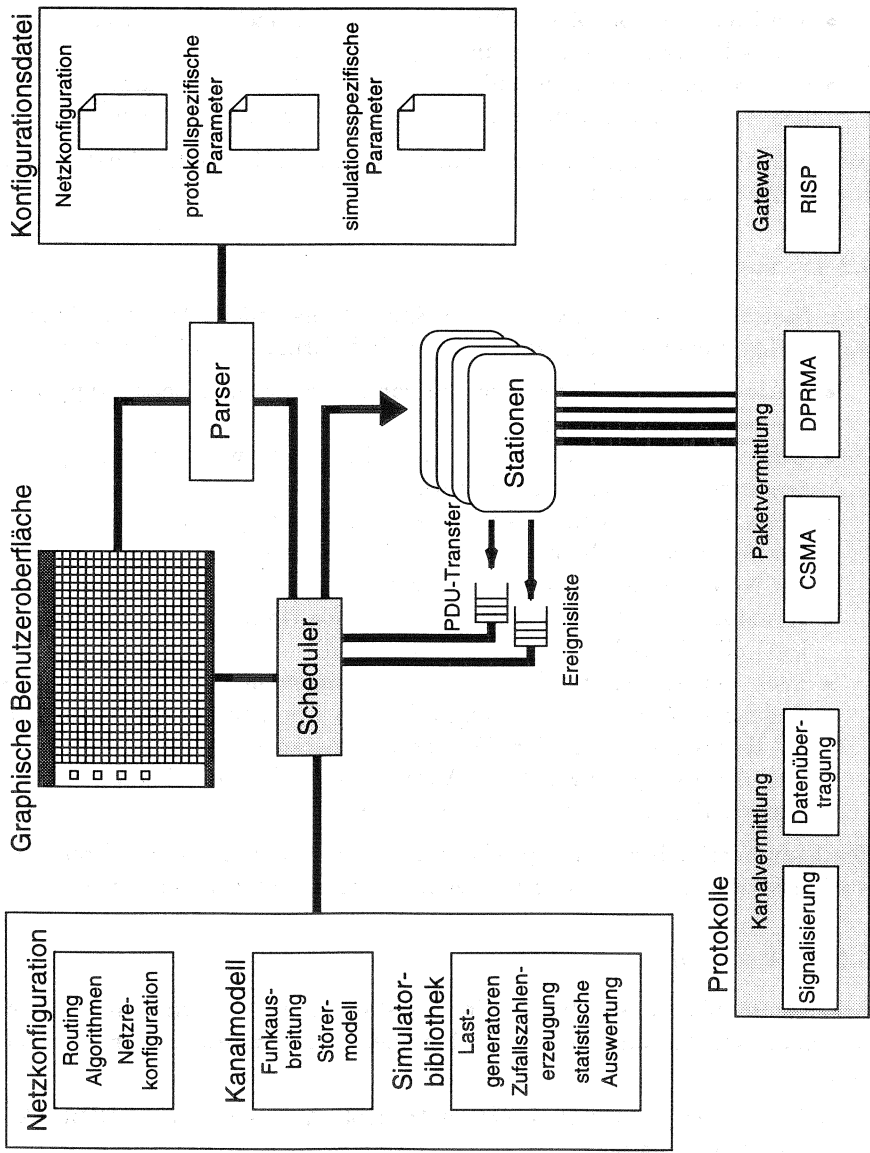


Abbildung 5.1: Architekturdiagramm des RMAN-Simulators

- alle Ereignisse pro Zeittakt werden getrennt nach Zeitschlitten (physikalische Kanäle) sortiert und abgearbeitet.
- die Ereignisnachrichten und zugehörige Protokolldateneinheiten werden den Stationen bzw. Protokollen zur Bearbeitung übergeben.

Die Funktion der übrigen Komponenten wird in den folgenden Abschnitten überblicksartig dargestellt:

5.2.1 Stationen

Jede Station eines RMAN wird im Simulationsmodell als eigenständige Strukturkomponente realisiert, die folgende Informationen beinhaltet:

- stationsbezogene Daten wie geographische Lage, Antennentyp, Typ der Festnetzanbindung usw.,
- Daten der angeschlossenen Benutzer-Endgeräte wie Terminaltyp, unterstützte Dienste, Terminalstatus usw.,
- Tabelle mit Wegelenkungsinformationen (Routing-Tabelle), die Adressen der nächsten Relaisstation in Richtung zu jeder anderen Station des Netzes enthält, vgl. Abschn. 2.6.3,
- Kanalbelegungstabelle, in der die Zustände (frei, temporär belegt, belegt) aller logischen Kanäle markiert sind,
- Liste der aktuell betriebenen Verbindungen in der Eigenschaft als Zugangsstation für direkt angeschaltete Endgeräte oder vermittelnde Relaisstation, in der Verbindungszustände (initiiert, eingerichtet, Auslösung eingeleitet), Adressen der Kommunikationspartner, Dienstgüteparameter usw. vermerkt sind.

Die komponentenbezogene Speicherung dieser Daten entspricht direkt der stationslokalen Datenhaltung in dezentral organisierten Funknetzen.

5.2.2 Protokolle

Alle Kommunikationsprotokolle sind vollständig und in Form von erweiternden endlichen Automaten (EFSM²) implementiert worden. Hierzu zählen derzeit:

- Differential Packet Reservation Multiple Access Protocol (DPRMA), vgl. Abschn. 4.5,

²Extended Finite State Maschine

- Carrier Sense Multiple Access Protocol (CSMA) nach IEEE 802.3, vgl. [46], das um ein Mehrfachquittierungsverfahren nach Brass [12] erweitert wurde.
- Signalisierungsprotokolle für kanalvermittelte Verbindungen, vgl. Abschn. 4.4,
- RMAN Interworking and Service Protocol (RISP) [10] für die Kopplung von RMAN- mit anderen (Mobilfunk-) Netzen.

5.2.3 Netzkonfiguration

Das Modul Netzkonfiguration umfaßt Routinen und Funktionen des RMAN-Netzmanagements. Da die Entwicklung des Simulators bisher nicht auf die Erprobung und Bewertung zugehöriger Protokolle zielt, werden diese Funktionen simulationstechnisch aus netzweiter Sicht bearbeitet.

Hierzu zählen:

- die Bereitstellung von Konnektivitätsmatrizen entsprechend eines Pegelmeßverfahrens,
- die Erstellung von Routingtabellen nach Verfahren wie Min-Hop, Netzrand-Routing, vgl. [44], usw.,
- die Netzinitialisierung bzw. -rekonfiguration nach Zugang/Abgang von Komponenten (Stationen, Terminals usw.).

5.2.4 Funkausbreitung und Kanalmodell

Das Kanalmodell des RMAN-Simulators bildet die Eigenschaften des Funkkanals sowie des digitalen Übertragungssystems nach und stützt sich auf eine empirische Untersuchung der Funkausbreitung im 60 GHz Bereich, vgl. [4].

Basis des Modells ist die Berechnung der Empfangsleistung bei Freiraumausbreitung in Abhängigkeit der Parameter Sendeleistung, Frequenz, Gewinn der Sende- bzw. Empfangsantenne und Abstand Sender-Empfänger. Bei der Verwendung von Sendefrequenzen im 60 GHz Bereich wird der Empfangspegel stark durch atmosphärische Störeinflüsse geschwächt, was durch einen linearen Dämpfungsterm mit Absorptionskonstanten für Sauerstoff und Wasserdampf (in Abhängigkeit einer vorgebbaren Niederschlagsmenge) berücksichtigt wird.

Für die Berechnung eines normierten Signal- zu Rauschleistungsverhältnisses E_b/N_0 am Empfänger wird die Störleistung durch Rauschen sowie durch gleichzeitig übertragende Stationen (co-channel interferer) einbezogen.

Thermisches Rauschen kann durch einen mittelwertfreien Gauß-Prozeß modelliert werden; das Kanalmodell berücksichtigt additives weißes Rauschen (AWGN)³ in Abhängigkeit der Umgebungstemperatur und Systembandbreite. Bezüglich der Gleichkanalstörer wird eine kumulierte Störleistung am Empfänger entsprechend der oben beschriebenen Funkausbreitung berechnet.

Übertragungsverluste zwischen Sender und Empfänger, die aufgrund des Fadings (Abschattung, Mehrwegeausbreitung usw.) entstehen, werden durch eine additive Dämpfungskonstante (Fade Margin), vgl. [93], ebenfalls berücksichtigt.

Zur abschließenden Berechnung der Paketfehler in Abhängigkeit des E_b/N_0 Verhältnisses wird angenommen, daß RMAN-Stationen eine 32 FSK⁴ Modulation sowie einen (32,23) Reed Solomon Code zur Kanalkodierung verwenden, Abb. 5.2 zeigt die Bitfehlerwahrscheinlichkeit in Abhängigkeit von E_b/N_0 , vgl. [93].

5.2.5 Konfigurationsdatei und Parser

Zu Beginn eines Simulationslaufs wird eine Konfigurationsdatei durch das Parser-Modul gelesen und ausgewertet. Die Konfigurationsdatei enthält im wesentlichen drei Sektionen, die das Simulationsszenario und die -parameter beschreiben:

- **Netzkonfiguration**

Hier werden alle Kenngrößen des zu untersuchenden Netzes festgelegt wie Anzahl der Stationen, geographische Lage jeder Station in der Ebene, von der Station verwendeter Antennentyp, Anzahl der aktiven Benutzer für jede Station, verfügbare Dienste an den Terminals usw.

- **Protokollspezifische Parameter**

Für alle verfügbaren Kommunikationsprotokolle können Initialisierungswerte der Zeitüberwachung (Timer), Übertragungswahrscheinlichkeit, Anzahl der Kanalvorschläge bei Verbindungseinrichtung usw. vorgegeben werden.

³Additive White Gaussian Noise

⁴Frequency Shift Keying

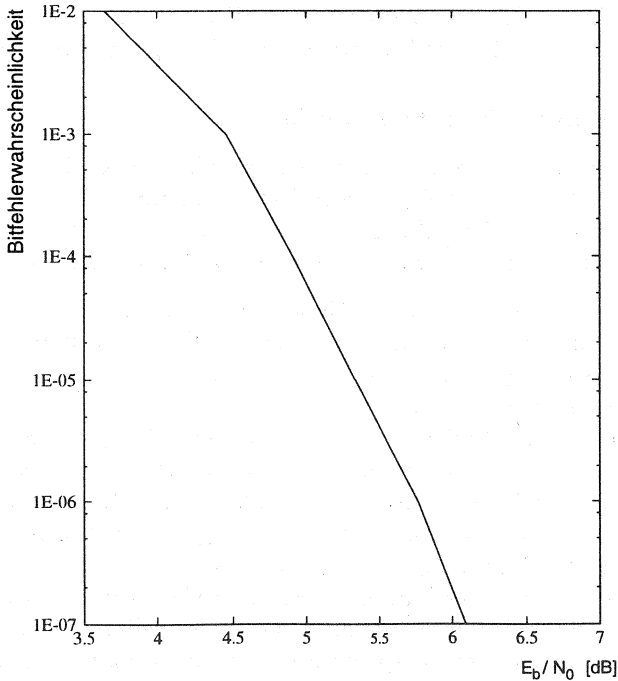


Abbildung 5.2: Bitfehlerwahrscheinlichkeit in Abhängigkeit von E_b/N_0

- **Simulationsspezifische Parameter**

Die Vorgaben dieser Sektion betreffen Fehlerschranken der statistischen Auswertung, die Parametrisierung von Lastmodellen, die Auswahl der statistischen Auswertungsmethode usw.

5.2.6 Simulatorbibliothek

Dieses Modul beinhaltet Komponenten, die universell für verschiedene Simulatoren einsetzbar sind, wie z.B.:

- einen synthetischen, quasi-idealen Zufallsgenerator (Lagged Fibonacci Generator), vgl. [86],
- Lastgeneratoren für verschiedene Daten- und Verkehrsarten, vgl. [2],

- Algorithmen für die Auswertung statischer Daten, wie das Batch-Means bzw. die Limited Relative Error (LRE) Verfahren, vgl. [91, 92].

5.2.7 Graphische Benutzeroberfläche

Der RMAN-Simulator verfügt über eine graphische X-Motif Oberfläche, die in Abb. 5.3 für eine beispielhafte Netzkonfiguration zu sehen ist. Die Oberfläche dient zur Erstellung von Simulationsszenarien, Visualisierung von Simulationsdaten und erlaubt die Steuerung des Simulationsablaufs:

- Das Hauptfenster stellt ein Gebiet von 2x2 km dar, das mit einem Raster zur einfachen Positionierung von Stationen unterlegt ist. Die Positionsanzeige in kartesischen Koordinaten erfolgt in der linken unteren Fensterecke.
- Die Menüleiste (links) ermöglicht die Erstellung von Szenarien. Es wird zwischen Stationen ohne/mit Zugang zu anderen Netzen unterschieden. Die Menüfunktionen ermöglichen den Zugang/Abgang bzw. das Verschieben von Stationen, die Festlegung der Anzahl aktiver Teilnehmer, die Zuordnung von Diensten usw.
- Aktuelle Werte der Simulationskenngrößen werden getrennt nach kanal- und paketvermitteltem Verkehr in weiteren Fenstern dargestellt. Hierzu zählen beispielsweise die Simulationszeit, die Anzahl der eingerichteten Verbindungen, der Durchsatz, die Anzahl der Blockierungen usw.
- Die Netzkonnektivität wird mittels durchgezogener Linien zwischen benachbarten Stationen dargestellt. Je Station kann ein Zusatzfenster angefordert werden, das die Anzahl der derzeit vermittelten Verbindungen und die belegten Funkkanäle anzeigt (in Abb. 5.3 nicht dargestellt).

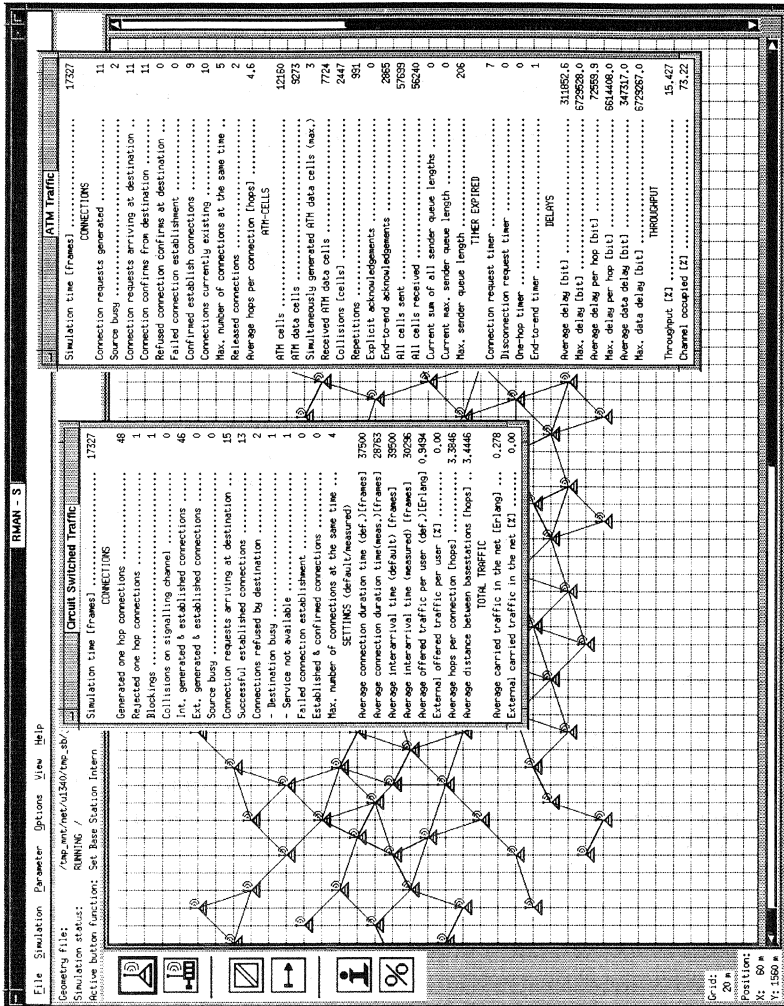


Abbildung 5.3: Graphische Benutzeroberfläche des RMAN-Simulators

RMAN: Simulationsszenarien und Leistungsbewertung

6.1 Kenngrößen zur Charakterisierung von Multi-Hop Mobilfunknetzen

Die Leistungsfähigkeit dezentral organisierter Mobilfunknetze bzw. zugehöriger Protokolle wird durch die geometrische Anordnung der Stationen (Netzgeometrie) stark beeinflusst. Bisher bekannte Leistungsuntersuchungen, vgl. z.B. [12, 100], bringen eine bestimmte Anordnung der Stationen entsprechend einer mathematischen Verteilung (z.B. Poisson Verteilung) in Ansatz. Ein Vergleich verschiedener Untersuchungen erweist sich häufig aufgrund unterschiedlicher Modellannahmen als schwierig. Deshalb sollen charakteristische Kenngrößen eingeführt werden, die geometrische Eigenschaften (dezentral organisierter) Mobilfunknetze beschreiben und einen Leistungsvergleich unterschiedlicher Protokolle gestatten.

Der Vermaschungsgrad eines Netzes soll durch einen Konnektivitätsfaktor beschrieben werden, der eine Aussage über den Anteil der Netzstationen liefert, die im Mittel von jeder Station erreicht werden können:

Der **Konnektivitätsfaktor KF** wird durch das Verhältnis der mittleren Anzahl von Nachbarn pro Station zur maximal möglichen Anzahl von Nachbarn definiert.

$$KF = \frac{1}{N(N-1)} \sum_{i=1}^N n_i$$

N bezeichnet die Anzahl der Stationen im Netz, n_i die Anzahl der Nachbarn der i -ten Station. Zwei Stationen heißen benachbart, sofern sie direkten Funkkontakt besitzen, vgl. Abschn. 4.3. Der Wertebereich des Konnektivitätsfaktors liegt zwischen Null und Eins (vollvermaschtes Netz).

Als weitere Kenngröße wird die mittlere Entfernung pro Verbindung eingeführt, sie beschreibt die mittlere Anzahl benötigter Teilstreckenverbindungen für eine Ende-zu-Ende Verbindung zwischen beliebigen Stationen eines Netzes.

Die **mittlere Entfernung pro Verbindung** $\bar{h}$ wird durch das Verhältnis der Entfernungssumme aller möglichen Quelle-/Ziel-Verbindungen zur Anzahl möglicher Quelle-/Ziel-Kombinationen gebildet.

$$\bar{h} = \binom{N}{2}^{-1} \sum_{i=1}^{\binom{N}{2}} h_i$$

Die mittlere Entfernung $\bar{h}$ bzw. die Entfernung h_i der i -ten Verbindung wird in Hops angegeben, wobei unter einem Hop die Teilstrecke zwischen zwei benachbarten Stationen zu verstehen ist.

Abb. 6.1 zeigt beispielhaft Anordnungen von Stationen in Form geometrischer Figuren und die zugehörigen Kenngrößen Konnektivitätsfaktor bzw. mittlere Entfernung pro Verbindung. Die schwarz ausgefüllten Kreise markieren eine Station, eine Verbindungslinie zwischen Stationen zeigt den Funkkontakt ($\equiv$ benachbarte Stationen) an.

Die Netzgeometrie 'Diamant' besitzt im gezeigten Beispiel den höchsten Vermaschungsgrad; aufgrund der hohen Symmetrie erkennt man, daß 1/3 aller Stationen 7 Nachbarstationen, 2/3 aller Stationen 4 Nachbarstationen haben, woraus sich ein Konnektivitätsfaktor von ungefähr 0,46 ergibt. Die mittlere Verbindungslänge beträgt 1,67 Hops.

Den schlechtesten Fall einer Netzvermaschung weist die linienförmige Anordnung von Stationen auf. Die Stationen am Netzrand haben eine Nachbarstation, alle übrigen zwei Nachbarstationen. Daraus ergibt sich ein Konnektivitätsfaktor, der umgekehrt proportional zur Stationszahl ist ($KF=2/N$), für ein Netz mit $N=12$ Stationen ist $KF \approx 0,17$. Entsprechend hoch ist die mittlere Anzahl benötigter Teilstreckenverbindungen für eine Ende-zu-Ende Verbindung zwischen zwei Stationen ($\bar{h}=4,33$ Hops).

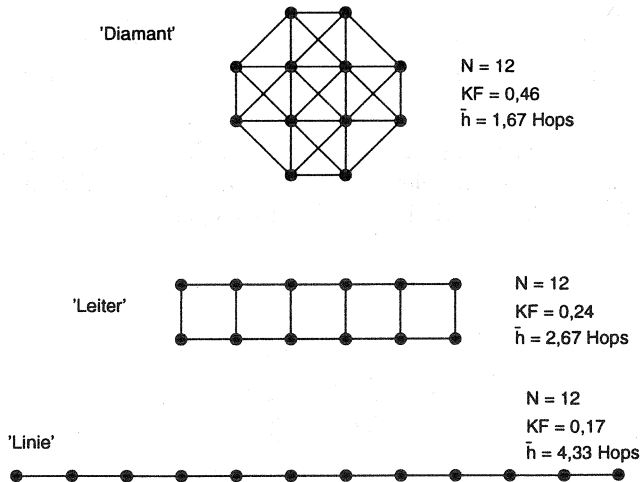


Abbildung 6.1: Netzgeometrien mit ihren Kenngrößen

6.2 Simulationsszenarien und -parameter

6.2.1 Szenarien

Zur Leistungsbewertung des RMAN wurden verschiedene Szenarien ausgewählt, dabei wurde von typischen Anwendungsfällen ausgegangen, wie sie in Abschn. 2.1 beschrieben wurden.

Die Erstellung der Szenarien erfolgte mit Hilfe der graphischen Benutzeroberfläche des RMAN-Simulators, vgl. Abschn. 5.2.7, die eine entsprechende Parameterdatei für das Parser-Modul des Simulators erzeugt.

Um eine weitgehende Vergleichbarkeit der Ergebnisse zur gewährleisten, wurden alle wesentlichen Simulationsparameter wie z.B. Anzahl der Stationen, Sendeleistung, protokollbezogene Kenngrößen usw. konstant gehalten, variiert wurde jeweils der im Netz angebotene Verkehr.

Die geographische Größe der betrachteten Gebiete ist für die untersuchten Szenarien nicht von Bedeutung, die Funknetze werden durch den Konnektivitätsfaktor, die mittlere Verbindungslänge und die geometrische Anordnung der Stationen charakterisiert.

Vollvermaschtes Mobilfunknetz

Bisher ist über die Leistungsfähigkeit dezentral organisierter Funknetze wenig bekannt.

In [82] wurden Leistungskenngrößen des EY-NPMA-Zugriffsprotokolls, vgl. Abschn. 2.4, für ein vollvermaschtes HIPERLAN bei einer Paketgröße von 53 byte ($\equiv$ einer ATM-Zelle) simulativ ermittelt, zu Vergleichszwecken wurden Untersuchungen für ein entsprechendes RMAN (DPRMA-Protokoll) durchgeführt. Vollvermaschte Funknetze ermöglichen eine weitgehend optimale Ausnutzung der verfügbaren Funkressourcen, da keine Kapazität für die Übertragung von Datenpaketen über den zweiten bis n -ten Hop einer Ende-zu-Ende Verbindung benötigt wird. Bei vollvermaschten Netzen ist die geometrische Anordnung und Anzahl der Stationen nicht von Bedeutung, sofern die (Daten-) Quellen gleichartig und homogen verteilt sind. Abb. 6.2 zeigt das vollvermaschte RMAN mit dem entsprechenden Konnektivitätsfaktor $KF=1$ bzw. der mittleren Verbindungslänge $\bar{h}=1$. Aus Übersichtsgründen sind in Abb. 6.2 Funkverbindungen (Links) zwischen Nachbarstationen nicht dargestellt, anhand des Radius R_L (für die Stationen S_1, S_2 eingezeichnet) kann die Anzahl der Nachbarn pro Station ermittelt werden, vgl. Abb. 6.3 und 6.4.

Lokales Netz

Das in Abb. 6.3 dargestellte Szenarium modelliert den Einsatz des RMAN z.B. als

- lokales Netz innerhalb eines Großraumbüros,
- Backbone Netz zur Verbindung drahtgebundener lokaler Netze aneinandergrenzender Bürogebäude.

Aufgrund des typischerweise symmetrischen Aufbaus von Bürogebäude-Etagen ergibt sich meist eine entsprechende Anordnung der Netzstationen. Der Konnektivitätsfaktor des Netzes beträgt 0,34, es weist einen höheren Vermaschungsgrad als die Netzgeometrie 'Leiter', vgl. Abb. 6.1, auf, was auf den geringeren Abstand zwischen den Stationen zurückzuführen ist. Die mittlere Entfernung beträgt 2,1 Hops. Für die Stationen S_1, S_2 ist beispielhaft das Kreisgebiet mit dem Radius R_L eingezeichnet, innerhalb dessen Stationen als Nachbarn erkannt werden.

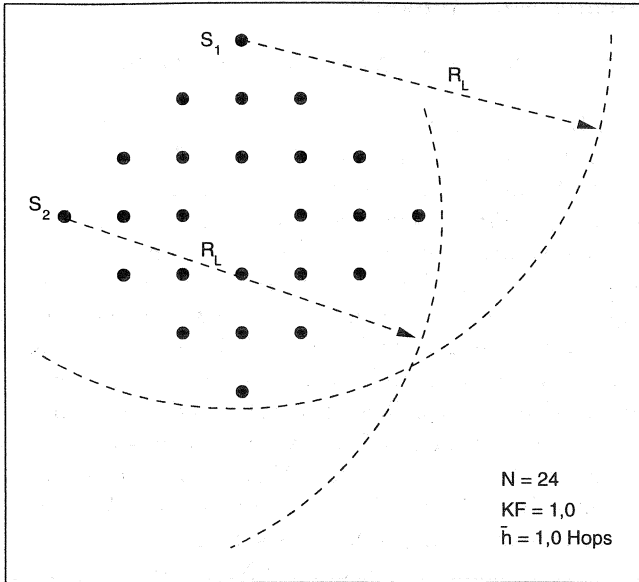


Abbildung 6.2: Vollvermaschtes Mobilfunknetz

Teilnetz im Innenstadtbereich

Dieses Szenarium modelliert ein Teilversorgungsgebiet im Ortsnetzbereich, vgl. Abb. 6.4. Das Anwendungsbeispiel steht exemplarisch für funkgestützte Zugangsnetze in Stadtgebieten zur Umgehung eines drahtgebundenen Ortsnetzes, wie in Abschn. 2.1 beschrieben, vgl. auch Abb. 2.1. Der Konnektivitätsfaktor ist 0,39, die mittlere Entfernung für eine Ende-zu-Ende Verbindung beträgt 1,79 Hops.

6.2.2 Simulationsparameter

Simulationen mit den oben beschriebenen Szenarien wurden für unterschiedliche Werte des angebotenen Verkehrs durchgeführt, der sich aus einem Mischangebot von zwei Quelltypen zusammensetzt. Tab. 6.1 führt die wesentlichen Simulationsparameter mit ihren Werten an.

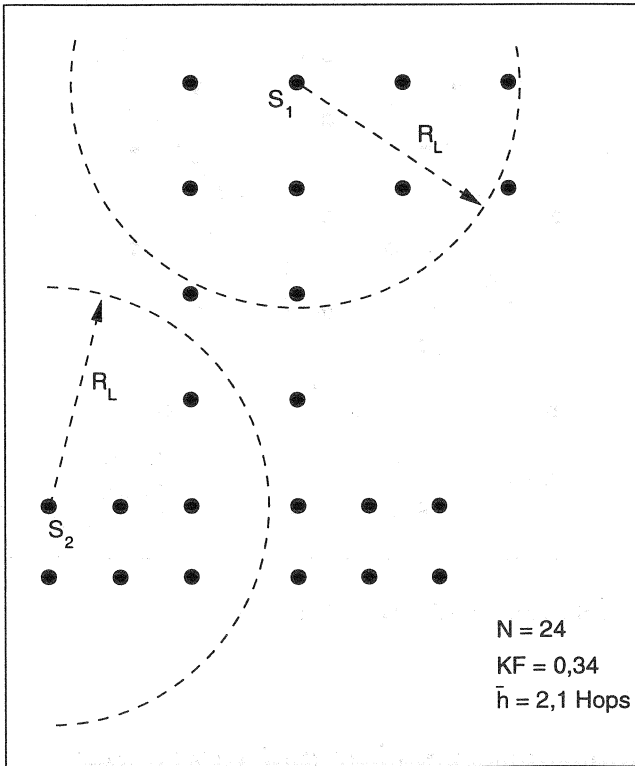


Abbildung 6.3: Szenarium: Lokales Netz

Quellentyp A modelliert Videoanwendungen mit niedriger Qualität, wie z.B. Bildfernsprechen, durch einen autoregressiven (AR) Prozeß. Der Anteil dieser Quellen bezüglich des angebotenen Gesamtverkehrs beträgt 20 %. 80 % des Angebots wird von Quellen (Typ B) erzeugt, die Datenverkehr nachbilden, wie er von Client-Server-Anwendungen oder Dateitransfer-Anwendungen hervorgerufen wird. Die Länge zugehöriger Datenpakete ist konstant, ihre Zwischenankunftszeit negativ exponentiell verteilt.

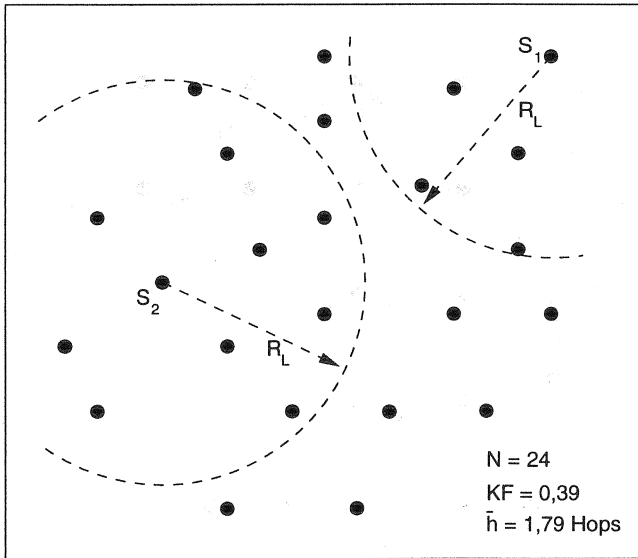


Abbildung 6.4: Szenarium: Teilnetz im Innenstadtbereich

6.3 Simulationsergebnisse und Bewertung

6.3.1 Szenario 'Vollvermaschtes Netz' – Vergleich von RMAN und HIPERLAN

In Abb. 6.5–6.7 sind korrespondierende Simulationsergebnisse für das vollvermaschte RMAN, vgl. Abb. 6.2, mit DPRMA-Kanalzugriffsprotokoll und ein äquivalentes HIPERLAN-Netz (EY-NPMA-Kanalzugriffsprotokoll) dargestellt.

Es zeigt sich, daß beim RMAN ein wesentlich höherer Nettodatendurchsatz, bis zu 500 % des Durchsatzes des HIPERLAN-Netzes erreicht wird. Die Durchsatzkurve für das RMAN steigt zunächst proportional zum angebotenen Verkehr an, bei höheren Verkehrsangeboten, bis zum maximalen Durchsatz von ca. 40 %, treten Sättigungseffekte auf, danach befindet sich das Netz im Überlastzustand (Angebot $> 0,65$). Die Differenz zwischen dem

Tabelle 6.1: Simulationsparameter (Auszug)

Sendeleistung	150 mW
Antennentyp	omnidirektional
Medienzugriffsprotokoll	DPRMA
Routing-Verfahren	Min-Hop
Zielteilnehmerwahl	freie Teilnehmer
mittlere Verbindungsdauer	$125,79 \cdot 10^6 \text{ bit}^1$
mittlere Zwischenankunftszeit von Rufen	$251,58 \cdot 10^6 \text{ bit}^1$
Quellentyp A	
mittlere Paketlänge	$1,6 \cdot 10^3 \text{ bit}$
mittlere Zwischenankunftszeit der Pakete	$140,17 \cdot 10^3 \text{ bit}^1$
Quellentyp B	
mittlere Paketlänge	$1,152 \cdot 10^3 \text{ bit}$
mittlere Zwischenankunftszeit der Pakete	$18,83 \cdot 10^3 \text{ bit}^1$
Kanalanforderungsschwellwert	$k > n \cdot 30$ ATM-Zellen ²
Kanalfreigabeschwellwert	$k < (n - 1) \cdot 30 + 10$ ATM-Zellen

für das RMAN erreichbaren Durchsatz und der Gesamtübertragungsrate hat folgende Ursachen:

- Protokolloverhead des DPRMA-Protokolls und der ATM-Zellen (5 byte je Zelle),
- benötigte Übertragungskapazität für den gemeinsamen Zugriffskanal (ACH), vgl. Abschn. 4.5,
- auf dedizierten Kanälen (DPCH) übertragene Signalisierungspakete, wie Bestätigungen für die Einrichtung von One-Hop Teilstrecken und die Ende-zu-Ende Signalisierung,
- Ende-zu-Ende Quittungen für erfolgreich empfangene Datenblöcke,

¹Normierung auf die Übertragungszeit pro Bit, vgl. Abschn. 6.3.1

² $k \equiv$ Anzahl der Zellen in einer Warteschlange, $n \equiv$ Anzahl der reservierten Kanäle

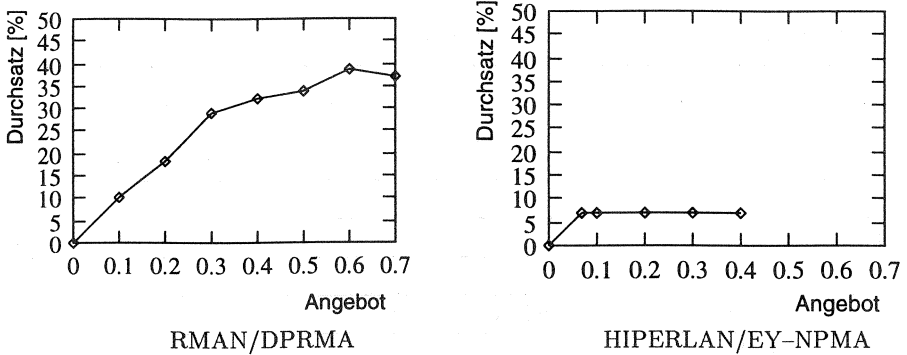


Abbildung 6.5: Szenario 'Vollvermaschtes Netz': Durchsatz in Abhängigkeit von dem angebotenen Verkehr

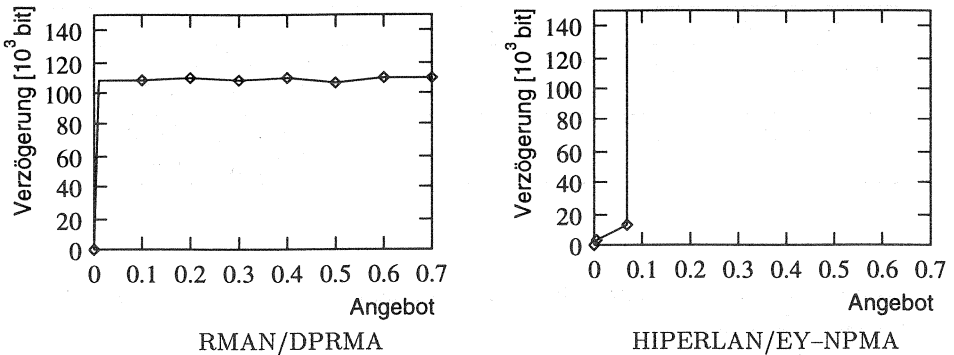


Abbildung 6.6: Szenario 'Vollvermaschtes Netz': mittlere Ende-zu-Ende Verzögerung in Abhängigkeit von dem angebotenen Verkehr

- Wiederholungen aufgrund fehlerhafter oder zerstörter Datenpakete, vgl. Abschn. 5.2.4.

Der Nettodatendurchsatz des HIPERLAN EY-NPMA-Protokolls erreicht schon bei geringem Verkehrsangebot sein Maximum von ca. 7 %. Dieser niedrige Wert ist wesentlich durch die komplexe Ausscheidungsphase für sendeberechtigte Stationen (Priority Phase und Contention Phase), vgl. [82],

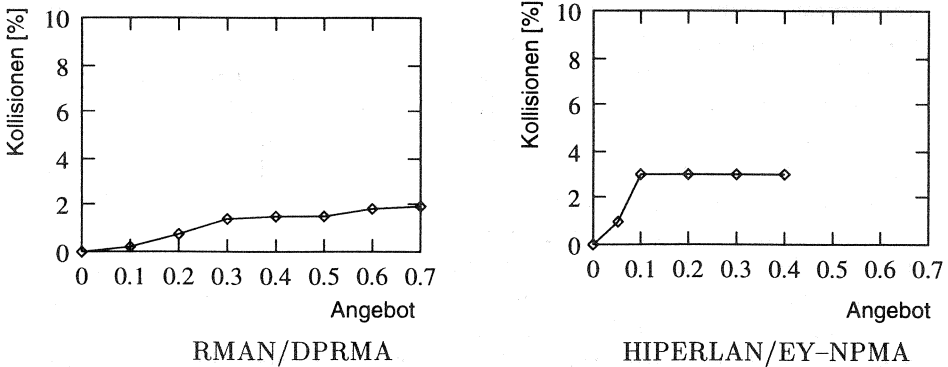


Abbildung 6.7: Szenario 'Vollvermaschtes Netz': Kollisionen in Abhängigkeit von dem angebotenen Verkehr

bedingt, wofür ein erheblicher Anteil der Gesamtübertragungsrate benötigt wird. Weiterhin wird der Durchsatz durch die explizite Bestätigung jeder erfolgreich gesendeten EY-NPMA-Protokolldateneinheit reduziert. Bei höheren Verkehrsangeboten ($\geq 0,1$) ist der Datendurchsatz konstant, vgl. Abb. 6.5 (rechts), da die Differenz der Pakete (Überlastangebot - max. Anzahl übertragbarer Pakete) an die überlagerte LLC-Schicht zurückgegeben wird und eine gesicherte Datenübertragung dann nicht mehr möglich ist. Simulationsergebnisse bezüglich der mittleren Ende-zu-Ende Verzögerung von Protokolldateneinheiten (PDE) sind in Abb. 6.6 dargestellt, die Meßgrößen sind auf die Übertragungszeit pro Bit bezogen, um eine Vergleichbarkeit trotz unterschiedlicher Gesamtübertragungsraten zu ermöglichen. Die mittlere Verzögerung ($\bar{V}$) in Sekunden erhält man durch Multiplikation mit der inversen Gesamtübertragungsrate, z.B. für eine mittlere Verzögerung $\bar{V} = 110 \cdot 10^3 \text{bit/s}$ ergeben sich folgende Werte in Abhängigkeit von der Gesamtübertragungsrate (D_{GES}):

$$D_{\text{GES}} = 2 \text{Mbit/s} \Rightarrow \bar{V}_{2 \text{Mbit/s}} = 55 \text{ms}$$

und

$$D_{\text{GES}} = 34 \text{Mbit/s} \Rightarrow \bar{V}_{34 \text{Mbit/s}} = 3,4 \text{ms}$$

Für das RMAN ist die mittlere PDE-Verzögerung über den gesamten dargestellten Bereich des Angebots nahezu konstant, was auf die Verbindungs-

annahmesteuerung des DPRMA-Protokolls zurückzuführen ist und für die Szenarien 'lokales Netz' und 'Teilnetz im Innenstadtbereich', vgl. Abschn. 6.3.2, detaillierter besprochen wird. Beeinflussende Größen sind im wesentlichen die Einrichtungszeit für One-Hop Teilstrecken und die Blockwiederkehrzeit des RMAN TDM-Rahmens, vgl. Abschn. 2.6.5.

Die mittlere Übertragungsverzögerung für das HIPERLAN weist bei geringen Verkehrsangeboten sehr niedrige Werte auf, was typisch für Netze mit einem von allen Stationen genutzten gemeinsam Übertragungsmedium ist. Bei höheren Angeboten steigt die mittlere Verzögerungszeit sprunghaft bis auf den Wert der max. Lebensdauer von Paketen an, d.h. die meisten Datenpakete werden vor der Übertragung verworfen.

Abb. 6.7 zeigt den prozentualen Anteil der Kollisionen, für das RMAN bzgl. des gemeinsam genutzten Zugriffskanals (ACH), beim HIPERLAN bezüglich der insgesamt übertragenen Datenpakete (für das von allen Stationen gemeinsam genutzte Übertragungsmedium). Es zeigt sich, daß die komplexe Auswahlphase des EY-NPMA-Protokolls zur Ermittlung sendeberechtigter Stationen insgesamt keine Vorteile gegenüber dem Reservierungsverfahren des DPRMA-Protokolls aufweist.

Insgesamt zeigen die vorliegenden Ergebnisse, daß die Konzeption des RMAN bezüglich der Protokolle und der Einteilung der Übertragungskapazität deutliche Vorteile im Hinblick auf den erzielbaren Durchsatz und die Stabilität des Gesamtsystems gegenüber dem HIPERLAN aufweist.

6.3.2 Szenarien 'Lokales Netz' und 'Teilnetz im Innenstadtbereich'

Simulationsergebnisse für die Szenarien lokales Netz (im folgenden als LAN-Szenario bezeichnet), vgl. Abb. 6.3, und Teilnetz im Innenstadtbereich (im folgenden RLL-Szenario), vgl. Abb. 6.4, zeigen die Abb. 6.8–6.10 und Abb. 6.11–6.13.

Die Durchsatzkurve für das LAN-Szenario steigt zunächst bis zum Maximalwert von 18 % (Angebot $\equiv 0,25$) linear an, bei steigendem Angebot ist der Durchsatz unter Berücksichtigung der statistischen Ungenauigkeit nahezu konstant. Der geringere Durchsatz im Vergleich zum vollvermaschten RMAN ergibt sich aufgrund der mittleren Verbindungslänge (2,1 Hops vs. 1 Hop). Ein äquivalenter Verlauf zeigt sich für das RLL-Szenario, der Maximalwert des Durchsatzes (ca. 23 %) wird entsprechend der mittleren Verbindungslänge von 1,8 Hops erst bei einem Angebot von 0,35 erreicht.

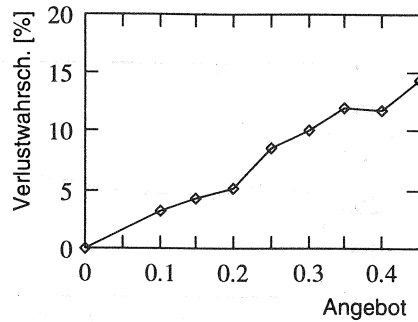
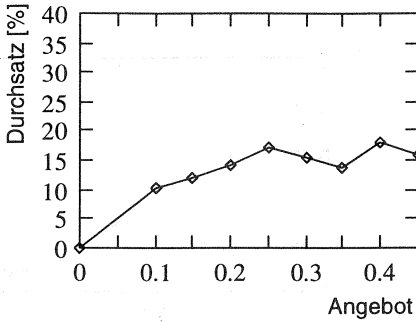


Abbildung 6.8: Szenario 'Lokales Netz': Durchsatz und Verlustwahrscheinlichkeit in Abhängigkeit von dem angebotenen Verkehr

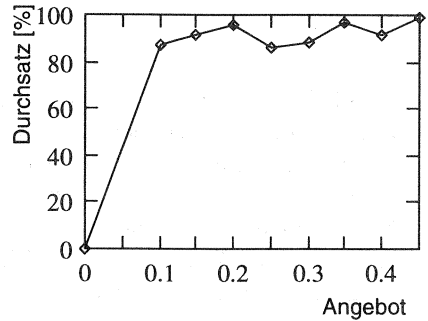
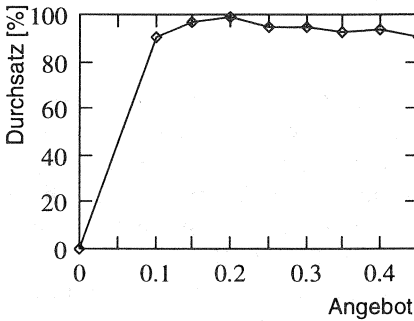


Abbildung 6.9: Szenario 'Lokales Netz': dienstbezogener Durchsatz für Dateitransfer-Quellen und Video-Quellen in Abhängigkeit von dem angebotenen Verkehr

Die Verlustwahrscheinlichkeit, Abb. 6.8 und Abb. 6.11 (rechts), gibt den prozentualen Anteil der zurückgewiesenen Verbindungswünsche an, diese Meßgröße muß im Kontext mit den dienstbezogenen Durchsätzen bzw. der mittleren Ende-zu-Ende Verzögerung betrachtet werden. Für beide Szenarien steigt die Verlustwahrscheinlichkeit mit wachsendem Angebot kontinuierlich an, was durch die Verbindungsannahmesteuerung (Connection Admission Control, CAC) des DPRMA-Protokolls bedingt ist: Existierende Verbindungen werden vor einer Verringerung der Dienstgüte durch neu hinzukom-

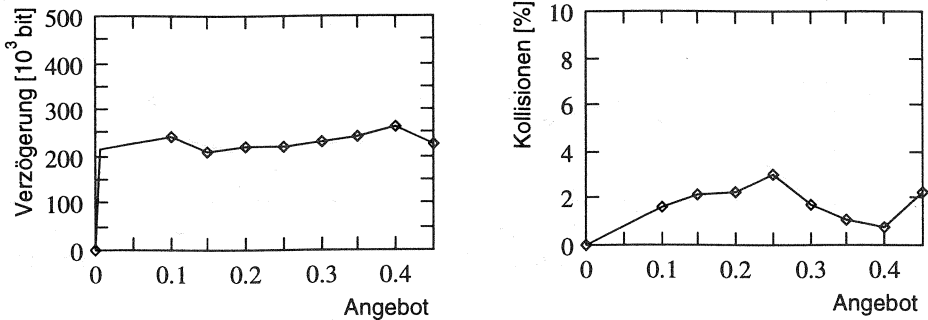


Abbildung 6.10: Szenario 'Lokales Netz': mittlere Ende-zu-Ende Verzögerung und Kollisionen in Abhängigkeit von dem angebotenen Verkehr

mende Verbindungen geschützt, die derzeit im RMAN-Simulator realisierte CAC-Funktionalität berücksichtigt die aktuelle Warteschlangenlänge, vgl. Abschn. 4.5, und die Anzahl freier Kanäle entsprechend der Kanalbelegungsliste bei der Entscheidung über die Zulassung neuer Verbindungen. Diese Vorgehensweise entspricht den Anforderungen an ATM-basierte Netze und zeigt deutlich Vorteile für die Einhaltung von Dienstgüteparametern: Die dienstbezogenen Durchsätze, vgl. Abb. 6.9 und 6.12, geben das Verhältnis von realisiertem zum geforderten Durchsatz eines Dienstes an, die zugehörigen Werte sind über den gesamten dargestellten Bereich des Angebots auf hohem Niveau (ca. 90–99 %) nahezu konstant. Schwankungen in Abhängigkeit von dem Angebot ergeben sich für das LAN-Szenario bezüglich der (burstartigen) Videoquelle, was auf ein hohes Verkehrsaufkommen in den Relaisstationen und den hohen Kanalbedarf insbesondere bei Verbindungen mit mehr als zwei Teilstrecken zurückzuführen ist.

Einen nahezu konstanten Wert weist ebenso die mittlere Verzögerung der Datenpakete in Abhängigkeit von dem angebotenen Verkehr bei beiden Szenarien auf. Der Anstieg der Verzögerung im Vergleich zum vollvermaschten RMAN entspricht ungefähr der höheren mittleren Verbindungslänge von 2,1 Hops (LAN) bzw. 1,8 Hops (RLL). Die Anzahl der Kollisionen, vgl. Abb. 6.10 und Abb. 6.13 (rechts), erreichen ihren Maximalwert entsprechend dem Höchstwert des Durchsatzes, bei weiter steigendem Angebot sinkt der Wert

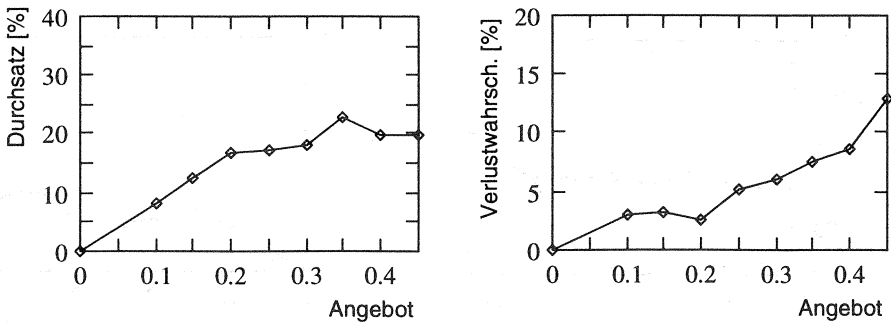


Abbildung 6.11: Szenario 'Teilnetz im Innenstadtbereich': Durchsatz und Verlustwahrscheinlichkeit in Abhängigkeit von dem angebotenen Verkehr

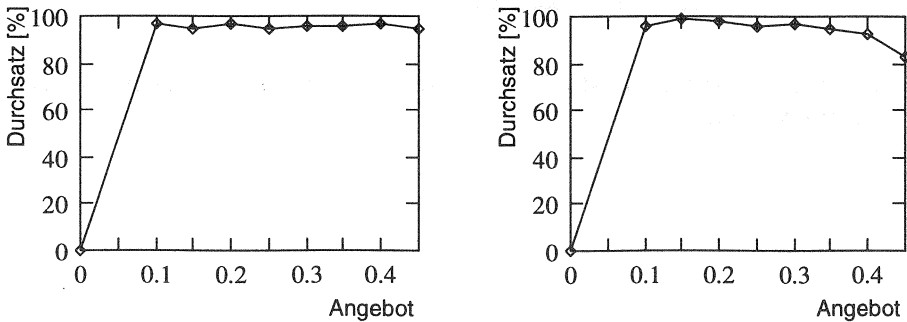


Abbildung 6.12: Szenario 'Teilnetz im Innenstadtbereich': dienstbezogener Durchsatz für Dateitransfer-Quellen und Video-Quellen in Abhängigkeit von dem angebotenen Verkehr

aufgrund der höheren Zurückweisungsrate von Verbindungen. Verglichen mit dem vollvermaschten RMAN ergibt sich wegen der Teilvermaschung der Netze ein höherer Maximalwert. Die Simulationsergebnisse zeigen insgesamt die Wirksamkeit der in das DPRMA-Protokoll integrierten Verfahren zum Schutz gegen den Störeinfluß versteckter Stationen in teilvermaschten Netzen: alle wesentlichen Leistungskenngrößen des Netzes blieben unter Berücksichtigung der bei Teilvermaschung benötigten Ressourcen bzw.

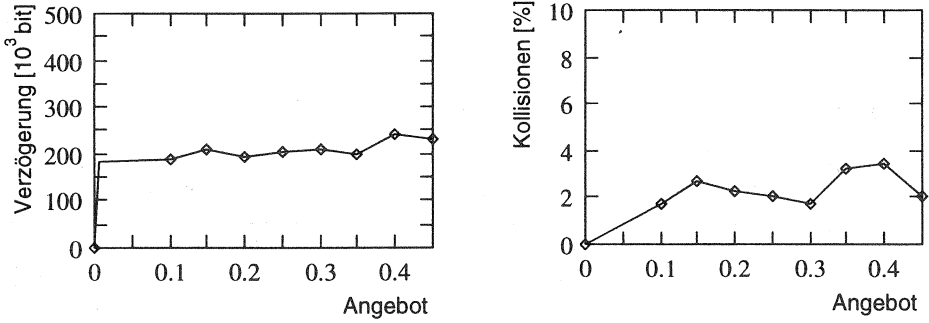


Abbildung 6.13: Szenario 'Teilnetz im Innenstadtbereich': mittlere Ende-zu-Ende Verzögerung und Kollisionen in Abhängigkeit von dem angebotenen Verkehr

Übertragungszeiten auf stabilem und zum vollvermaschten RMAN äquivalentem Niveau.

Softwareentwicklungsmethodik für Kommunikationssysteme

7.1 Motivation

Nachfolgend wird eine Softwareentwicklungsmethodik entworfen, deren zugehörige Verfahren und Werkzeuge zur Realisierung der RMAN-Netzkomponenten, darüber hinaus auch zur Entwicklung von Kommunikationssystemen generell, verwendet werden können. Zur Realisierung von Kommunikationssoftware werden heute im wesentlichen folgende Methoden eingesetzt:

1. Die geforderte Softwarefunktionalität wird einer Anforderungsspezifikation entnommen und durch einen Entwickler manuell mit Hilfe einer informalen Programmiersprache (C, Pascal, Assembler usw.) in ein auf der Endsystemhardware ablauffähiges Programm umgesetzt.
2. Die Systemfunktionalität wird eindeutig mit einer formalen Beschreibungssprache spezifiziert, aus der Spezifikation wird mittels automatischer Codegenerierung ein ablauffähiges Programm erzeugt.

Beide Methoden weisen im Sinne einer effizienten Softwareentwicklungstechnik erhebliche Schwächen auf: Software, die nach Methode 1 entwickelt wird, weist nach der Erstellungsphase häufig funktionale Unterschiede zur Spezifikation auf, die auf (menschliche) Transformationsfehler zurückzuführen sind und in aufwendigen Testreihen beseitigt werden müssen. Weiterhin erweist sich typischerweise die mangelnde Aktualität der Quelltextdokumentation als wesentliches Hindernis für die Weiter- bzw. Wiederverwendung der Software durch andere Entwickler.

Der Einsatz formaler Spezifikationssprachen entsprechend Methode 2 vermeidet die zuvor genannten Probleme dadurch, daß die Spezifikation zugleich Programmquelltext und -dokumentation ist. Formale Sprachen sind jedoch primär für die eindeutige Systembeschreibung ausgelegt, die Effizienz der aus einer Spezifikation resultierenden, automatisch erzeugten Implementierung und der Einfluß der entwicklerspezifischen Spezifikationsmethodik

auf die Softwareperformance kann mit den bisher verfügbaren Werkzeugen nicht beurteilt werden.

Beiden Methoden gemeinsam ist, daß die Entwicklung neuer Anwendungssoftware und -systeme derzeit im wesentlichen auf Erfahrungen der Entwickler mit existierenden Systemen und Softwarepaketen basiert. Nach wie vor fehlen Methoden und somit auch Werkzeuge, die eine ingenieurmäßige Konstruktion von Kommunikationssoftware und -systemen unterstützen und eine objektive Bewertung der Implementierung gestatten.

7.2 Konzept des erweiterten System-Entwicklungszyklus

Ein Ziel dieser Arbeit ist es, eine Softwareentwicklungsmethodik zu entwerfen und Werkzeuge zu realisieren, mit deren Hilfe sich formale Spezifikationen über eine Systemspezifikationsphase bzw. -definitionsphase hinaus als wesentliche Softwarekomponenten eines Softwarelebenszyklus verwenden lassen.

Die Entwicklungen bzw. Untersuchungen beschränken sich auf formale Systemspezifikationen, da sie im Sinne einer effizienten Softwareentwicklungsmethodik deutliche Vorteile gegenüber informalen aufweisen, vgl. Kap. 3. Die grundlegende Idee der erweiterten Softwareentwicklungsmethodik besteht darin, formale Spezifikationen sowohl zur simulativen Leistungsbeurteilung als auch zur Prädiktion der erforderlichen Leistungsfähigkeit einer Endsystemhardware heranzuziehen. Die Einordnung der neuen Verfahren in einen Softwarelebenszyklus zeigt ein vereinfachtes Phasenmodell in Abb. 7.1.

Diese neue Methodik zielt zunächst auf die Entwicklung von Kommunikationsprotokollen und -systemen ab. Der konzeptionelle Ansatz und die Realisierung erfolgte dabei so, daß eine Anwendung auf anderen Gebieten der Softwareentwicklung möglich und sinnvoll ist.

Zur Bewertung der Leistungsfähigkeit von Kommunikationssoftware bzw. -systemen bieten sich grundsätzlich folgende Verfahren an:

- Messungen am realen System,
- Modellierung und analytische Berechnung,
- Modellierung und stochastische Simulation.

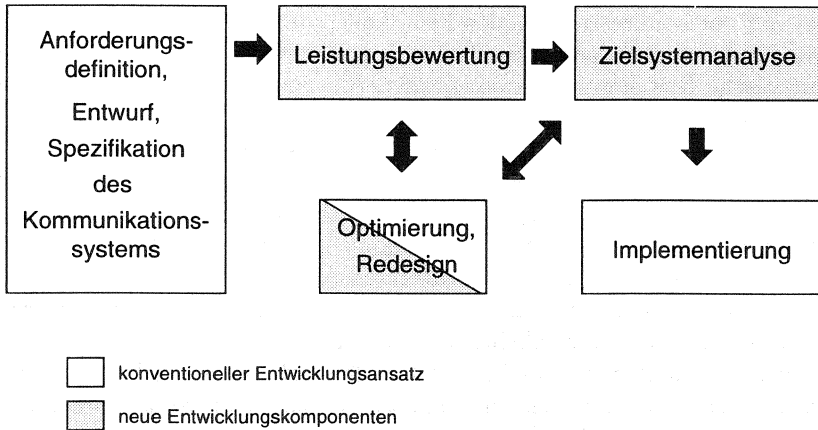


Abbildung 7.1: Vereinfachtes Phasenmodell eines Softwarelebenszyklus

Messungen an realen Systemen sind als Verfahren zur Leistungsbewertung während der Systementwicklungsphase nicht geeignet, da sie die Existenz des Gesamtsystems und seinen Betrieb unter realen Belastungsbedingungen voraussetzen.

Modelle zur analytischen Berechnung erfordern typischerweise eine starke Abstraktion bzw. Vereinfachung des realen Systems, um mathematisch lösbar zu sein. Die Untersuchungen müssen meist auf kleine Systeme beschränkt werden, eine detaillierte Bewertung komplexer (Kommunikations-) Systeme ist schwierig, teilweise unmöglich.

Die stochastische Simulationstechnik hat sich seit Jahrzehnten als geeignetes Verfahren zur Leistungsbewertung von Systemen bewährt. Ein zufallsgesteuertes System wird durch Modellierung auf ein Computerprogramm abgebildet und die interessierenden Systemzufallsgrößen durch Messung und statistische Auswertung während des Simulationslaufes bestimmt, vgl. [14]. Vernetzte Systeme der Kommunikations- und Datentechnik sind typischerweise zufallsgesteuerte Systeme.

In einem ersten Erweiterungsschritt sollen deshalb die Techniken der stochastischen Simulation und formalen Spezifikation integriert werden. Im Gegensatz zur konventionellen Vorgehensweise bei der Erstellung stochastischer Simulationsprogramme soll kein Modell des zu untersuchenden Kommunikationsprotokolls verwendet werden; simulative Untersuchungen werden

direkt mit der konkreten formalen Protokollspezifikation durchgeführt. Dazu wird eine stochastische Simulationsumgebung benötigt, die eine semi-automatische Einbettung formal spezifizierter Protokolle zur Leistungsbeurteilung ermöglicht. Die Vorteile dieser Vorgehensweise liegen einerseits darin, daß ein Systementwickler nicht über Kenntnisse bezüglich der Modellierung von Kommunikationssystemen verfügen muß, andererseits die Erstellung eines eigenständigen Simulationsprogramms parallel zur Systementwicklung entfällt. Mit diesem System können sowohl einzelne Kommunikationsprotokolle als auch ganze Protokollstapel zunächst unabhängig von der Existenz eines Kommunikationsnetzes oder einer Endsystemhardware analysiert, miteinander verglichen und bewertet werden.

Eine zweite Erweiterung des konventionellen Softwareentwicklungsprozesses zielt darauf ab, die meist unabhängig voneinander verlaufenden Entwicklungsprozesse von Hard- und Software eines Kommunikationssystems stärker zu korrelieren.

Die grundlegende Idee besteht darin, Werkzeuge für simulative Untersuchungen an formal spezifizierten Kommunikationsprotokollen zu entwickeln, um z.B.

- Aussagen über detaillierte Anforderungen an Leistungsfähigkeit und Aufbau der benötigten Systemhardware zu treffen,
- Protokollspezifikationen im Hinblick auf eine gleichmäßige und effiziente Ausnutzung der Systemhardware zu analysieren und zu bewerten,
- Implementierungen zu optimieren und eine Über-/Unterdimensionierung der Systemhardware bereits in einer frühen Entwicklungsphase zu vermeiden,
- Protokolloptionen und zusätzliche Dienste im Bezug auf die benötigte Rechenleistung zu analysieren und eine Aufwand-/Nutzen-Analyse durchzuführen,
- Verschiedene Entwicklungswerkzeuge bezüglich ihrer automatischen Codegenerierung vergleichbar zu machen.

Da ein derartiges Verfahren bisher nicht eingesetzt wurde und somit keine Werkzeuge verfügbar sind, soll diese Aufgabenstellung durch Konzeptionierung und Implementierung eines universellen, modularen Hardware-Emulators gelöst werden, mit dem aussagekräftige Leistungskenngrößen der benötigten Hardware ermittelt werden. Dafür müssen ebenfalls Kennwerte definiert werden, die einerseits eine Quantifizierung der individuellen Hard-

wareleistung und andererseits einen Vergleich dieser Werte auch über Architekturgrenzen hinweg ermöglichen.

Insgesamt gesehen wurde eine Softwareentwicklungsmethodik entworfen, die es ermöglicht, bereits in einer frühen Phase des Entwurfs und der Implementierung eines Systems einzelne Komponenten zu testen, zu optimieren und somit den Entwicklungszeitraum entscheidend zu verkürzen. In den folgenden Abschnitten wird die stufenweise Realisierung des Konzepts eines erweiterten System-Entwicklungszyklus beschrieben, sowie Untersuchungsergebnisse für verschiedene Kommunikationsprotokolle vorgestellt und bewertet.

7.3 Basismodell und erste Realisierungsstufe des Simulationswerkzeuges

In einer ersten Realisierungsstufe wurde, gemäß des in Abschn. 7.2 vorgestellten Konzeptes, ein Simulationswerkzeug zur Integration der Techniken der formalen Spezifikation von Kommunikationsprotokollen und der stochastischen Simulation entwickelt.

Für Untersuchungen mit dem neu entwickelten Simulationswerkzeug wurden Kommunikationsprotokolle formal mit der Sprache SDL (Specification and Description Language) spezifiziert, vgl. Abschn. 3.4. SDL basiert auf dem Modell des erweiterten endlichen Automaten und ist somit für die Beschreibung von Protokollen gut geeignet.

Das Simulationswerkzeug versteht sich nicht als parallele Entwicklung zu derzeit verfügbaren Spezifikationswerkzeugen, sondern als neue, additive Komponente im Sinne der beschriebenen erweiterten Softwareentwicklungsmethodik. Deshalb wurde auf die Realisierung eines eigenen SDL-Spezifikationswerkzeuges mit Editor, Syntaxparser, Codegenerator usw. verzichtet und statt dessen auf die Funktionalität des SDL Development Tool (SDT), vgl. [98], zurückgegriffen. Das neu entwickelte Simulationswerkzeug erhielt, in Anlehnung an das SDT-Werkzeug, die Bezeichnung SDT Performance Evaluation Toolkit (SPEET). Die Kombination beider Werkzeuge (SDT + SPEET) ermöglicht die Durchführung eines erweiterten, optimierten Softwareentwicklungszyklus wie er in Abschn. 7.2 beschrieben wurde.

7.3.1 Basis-Simulationsmodell

Realitätsnahe, simulative Untersuchungen an (formal spezifizierten) Kommunikationsprotokollen erfordern einen Betrieb, der dem späteren tatsächlichen Einsatz weitgehend entspricht. Für die Realisierung des SPEET-Simulationswerkzeugs wurde anhand einer Anforderungsspezifikation ein Basis-Simulationsmodell in Anlehnung an das ISO/OSI Referenzmodell entwickelt, vgl. Abb. 7.2. Durch die dunkler gezeichneten Symbole sind die Hauptkomponenten des SPEET-Simulators (Realisierungsstufe I) markiert, die heller gezeichneten Symbole stellen beispielhaft die Konfiguration eines zu untersuchenden Kommunikationssystems dar und sind nicht fester Bestandteil des Simulators.

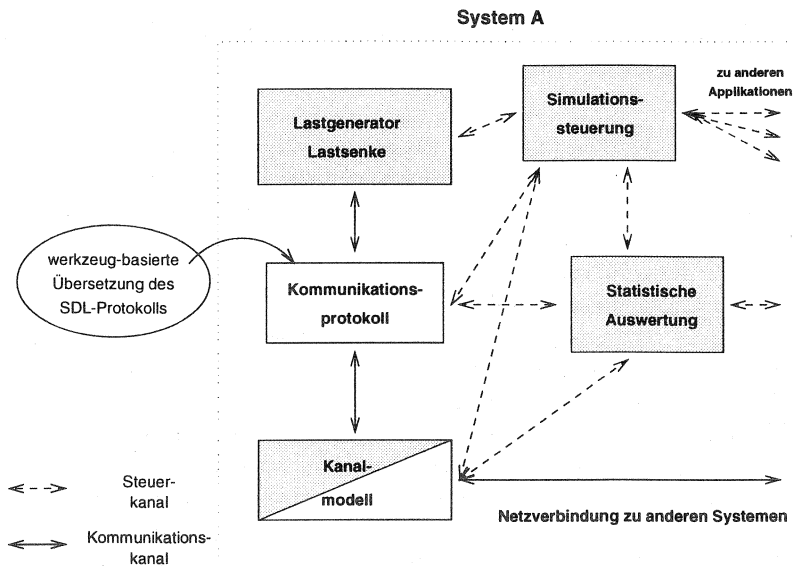


Abbildung 7.2: Basis-Simulationsmodell

Das Basismodell umfaßt einerseits simulations- und bewertungsspezifische Komponenten, andererseits grundlegende Komponenten einer realen Protokollumgebung.

Ein Protokoll oder Protokollstapel ist Bestandteil eines Kommunikationssystems (hier System A), eines Endgerätes, eines Vermittlungsknotens etc., das mit anderen Systemen kommuniziert. Jede Protokollschicht stellt der überlagerten Schicht oder einem Anwendungsprozeß Dienste unter Nutzung der Dienste einer unterlagerten Schicht zur Verfügung, vgl. [51]. Von zentraler Bedeutung für die Systemanalyse ist der Protokollbetrieb unter exakt definierten, realistischen und reproduzierbaren Lastbedingungen, der dem späteren tatsächlichen Einsatz weitgehend entspricht. Die zu erwartenden Systembelastungen müssen möglichst exakt bereits in der Test- und Validierungsphase durch geeignete Lastmodellierung substituiert sein. Diese Aufgabe wird von einem multimedialen Lastgenerator des SPEET-Simulators übernommen, der gleichzeitig als Verkehrssenke dient.

Für die automatische Einbettung eines Protokolls oder Protokollstapels in die SPEET-Simulationsumgebung wird der Codegenerator eines (SDL-) Spezifikationswerkzeuges, z.B. des SDT, verwendet, der aus einer formalen Spezifikation ein in der Simulationsumgebung ablauffähiges Programm erzeugt.

Die Simulationssteuerung koordiniert alle an einem Simulationslauf beteiligten Prozesse bzw. Anwendungskomponenten. Bei unabhängig voneinander ablaufenden Prozessen einer Simulation ist eine zentrale Steuerung und Verwaltung der Simulationszeit notwendig, um den Einfluß einer wechselnden Belastung des Rechnersystems während eines Simulationslaufs zu eliminieren. Aufgabe der statistischen Auswertung ist es, die interessierenden Meßdaten der verschiedenen Systemkomponenten während des Simulationslaufs aufzuzeichnen und nach vorgegebenen statistischen Verfahren auszuwerten. Die Verbindung eines Protokolls bzw. Protokollstapels mit anderen Kommunikationssystemen wird durch eine anwendungsspezifische Substitution unterlagerter Protokollschichten bzw. eines Kommunikationsnetzes realisiert. Diese Komponente ist als Kanalmodell des Basis-Simulationsmodells bezeichnet.

Die Kommunikation zwischen allen an einer Simulation beteiligten Komponenten erfolgt über bidirektionale Kanäle; bezüglich der Kanäle wird logisch zwischen (simulationsspezifischen) Steuerkanälen und Kommunikationskanälen, die den Kommunikationsfluß des realen Systems nachbilden, unterschieden.

Die Aufteilung der Simulationsumgebung in die beschriebenen Einzelkomponenten erfolgte unter dem Gesichtspunkt, funktionale Einheiten durch eigenständige Anwendungen zu repräsentieren und verteilte Simulationsläufe

zu unterstützen. Im folgenden wird der Aufbau der einzelnen Komponenten mit ihren Leistungsmerkmalen detailliert beschrieben, sie wurden in der Programmiersprache C unter UNIX realisiert.

7.3.2 Der multimediale Lastgenerator MultiGen des SPEET-Simulators

Überblick

Der Lastgenerator MultiGen basiert auf der prototypischen Realisierung eines Lastgenerators für Untersuchungen bezüglich der Parallelisierungsmöglichkeiten formal spezifizierter Kommunikationsprotokolle, vgl. [111]. Für den Einsatz in der SPEET-Simulationsumgebung wurde der in [111] entwickelte Lastgenerator völlig neu strukturiert und bezüglich seines Funktionsumfangs, insbesondere der Lastmodelle, erheblich erweitert.

Zur einfachen Handhabung verfügt der Lastgenerator MultiGen über eine graphische OSF/Motif Benutzeroberfläche, die den Anwender über alle wesentlichen Aktionen und Zustände des Lastgenerators informiert und ihm die Modifikation aller verfügbaren Parameter gestattet. Jeder MultiGen Lastgenerator identifiziert sich gegenüber anderen Anwendungen durch eine vom Benutzer festlegbare Systemadresse, die synonym zu einer Endgeräteidentifikation, Vermittlungsknotenadresse usw., ist.

In der Funktion als Lastquelle können

- virtuelle Verbindungen zu allen an einer Simulation beteiligten MultiGen Lastgeneratoren oder äquivalenten Anwendungen eingerichtet werden. In der aktuellen Version können von einem MultiGen Lastgenerator bis zu zehn parallele Verbindungen unterhalten werden, die anhand einer Verbindungsidentifikation (Connection Endpoint Identifier, CEI) unterschieden werden.
- verbindungsbezogenen Protokolldateneinheiten nach vorgegebenen Verteilungen generiert, mit einer CRC-16 Fehlerprüfsumme nach ITU-T V.41 versehen und versendet werden.
- eingerichtete, virtuelle Verbindungen ausgelöst werden.

In seiner Eigenschaft als Lastsenke

- werden eintreffende Verbindungswünsche entgegengenommen und anhand der vom Anwender spezifizierten Parameter akzeptiert oder abgewiesen,

- ankommende Protokolldateneinheiten empfangen und optional auf Fehler (CRC-16) überprüft.

Der Lastgenerator MultiGen besteht aus fünf voneinander unabhängigen Modulen, die über definierte Schnittstellen miteinander kommunizieren. Der strukturelle Aufbau ist in Abb. 7.3 dargestellt.

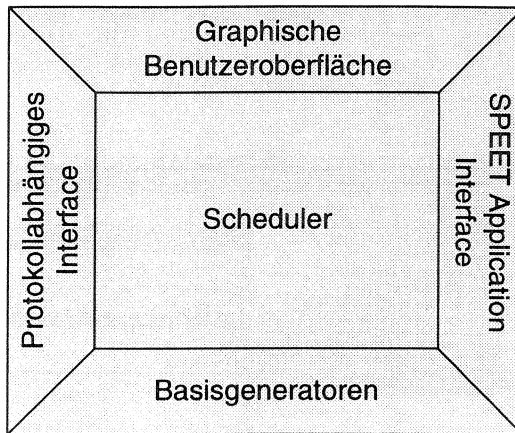


Abbildung 7.3: Komponenten des SPEET-Lastgenerators MultiGen

Graphische Benutzeroberfläche

Die graphische Benutzeroberfläche, vgl. Abb. 7.4 (oben), dient zur Steuerung des MultiGen Lastgenerators und Visualisierung seiner Zustandsdaten:

- Darstellung verbindungsbezogener Daten:
Ein Statusfenster zeigt für alle parallel einrichtbaren Verbindungen den Zustand (keine, uni- oder bidirektionale Verbindung), die protokollbezogene Verbindungsidentifikation (CEI), die Adresse des Verbindungsinitiators und -adressaten, die Anzahl der gesendeten/empfangenden Datenpakete und Bytes, sowie die Anzahl detektierter Übertragungsfehler an.

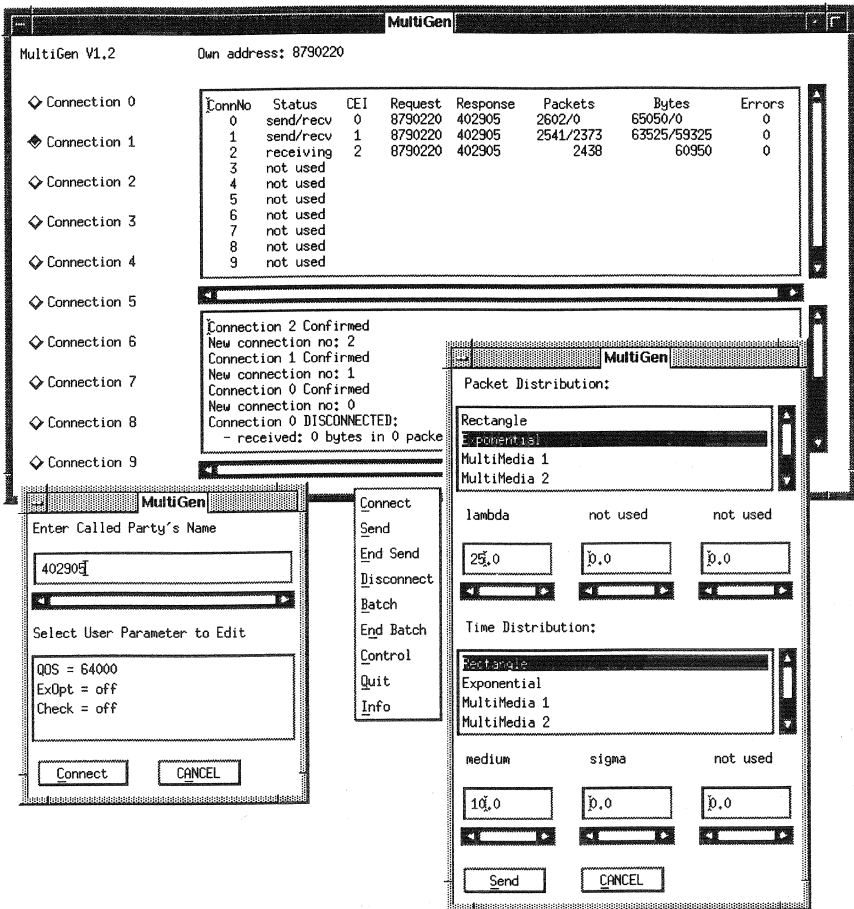


Abbildung 7.4: Graphische Benutzeroberfläche des SPEET-Lastgenerators

- Anzeige von Systemmeldungen:
Status-/Fehlermeldungen der anderen Lastgenerator-Module werden in einem Nachrichtenfenster angezeigt.
- Verbindungssteuerung:
Über ein Menü, vgl. Abb. 7.4 Mitte, können verbindungsbezogen folgende Aktionen initiiert werden:
 - Einrichten von Verbindungen:
In einer Dialogbox, vgl. Abb. 7.4 (l.u.) können Parameter für die Verbindungseinrichtung wie die Adresse des Zielsystems und benutzerdefinierte Dienstgüterkmale (Diensttyp, Durchsatz, maximale Verzögerung etc.) spezifiziert werden.
 - Sendeauftrag:
Die Datenübertragungsphase einer bestehenden Verbindung wird durch Parametrisierung eines Sendeauftrags definiert. Über eine Dialogbox, vgl. Abb. 7.4 (r.u.), können verschiedene Verteilungen für die Paketlänge und die Zwischenankunftszeit der Datenpakete ausgewählt und parametrisiert werden.
 - Beenden eines Sendeauftrags,
 - Auslösen einer Verbindung,
 - Batch-Betrieb:
Der Batch-Betriebsmodus ermöglicht die Ausführung aller oben aufgeführten Verbindungssteuerungsaktionen nach dem Stapelverarbeitungsprinzip, somit können komplette Simulationsszenarien vordefiniert werden. In einer Batch-Datei werden mit Hilfe einer Kommandosprache Aktionen und ihre Ausführungszeit relativ zum Verarbeitungsbeginn oder zu einer definierten Simulationszeit festgelegt. Die Ausführung der Batch-Kommandos und die Ausführungskontrolle übernimmt der Scheduler des Lastgenerators.
 - Beenden des Batch-Betriebs.

Basisgeneratoren

Die verfügbaren Basisgeneratoren zur Lastmodellierung multimedialer Anwendungen basieren auf einer umfangreichen Sichtung und Auswertung von Literatur, vgl. [2]. Darauf aufbauend wurden zunächst die in derzeitigen Kommunikationsnetzen auftretenden Daten- und Verkehrsarten zusammengestellt:

Datenarten:

- Audio (hohe und niedrige Qualität),
- Video (hohe, mittlere, und niedrige Qualität),
- Bilder,
- Graphiken,
- Texte,
- numerische Daten.

Verkehrsarten:

- Kommunikation, Dialog in Echtzeit,
- Broadcast, Verteilkommunikation,
- Station-Host Anfragen,
- Datenverkehr (Dateitransfer, E-Mail usw.),
- Verteilte Verarbeitung, Gemeinsames Arbeiten,
- Fernmessen, Fernwirken.

Aus der Kombination von Daten- und Verkehrsarten ergeben sich theoretisch 54 verschiedene Möglichkeiten für Datenverkehr zwischen Kommunikationssystemen. Weitere Untersuchungen haben gezeigt, daß in Kommunikationssystemen hauptsächlich 18 Kombinationen von Daten- und Verkehrsarten auftreten, wobei es möglich ist, die übrigen Kombinationen durch Parametrisierung der Lastmodelle größtenteils abzudecken. Die Anzahl benötigter Verkehrsgeneratoren kann durch geeignete Gruppierung von Daten- und Verkehrsarten auf vier Basisgeneratoren reduziert werden. Tabelle 7.1 zeigt die relevanten Daten- und Verkehrsarten und die jeweilige Zuordnung zu einer der vier Basisgeneratoren A, B, C, D.

Die Basisgeneratoren unterscheiden sich im wesentlichen bezüglich der Längen der generierten Protokolldateneinheiten und ihrer Zwischenankunftszeit sowie der Parameter der zugrundeliegenden Verteilungsfunktionen. Die Reduktion auf vier Basisgeneratoren erfolgte unter dem Gesichtspunkt einer handhabbaren Anzahl von Generatoren einerseits und einer begrenzten Anzahl von Lastmodellparametern andererseits.

Scheduler

Der Scheduler steuert bzw. initiiert als zentrales Modul alle Aktionen des MultiGen Lastgenerators und koordiniert den Datenaustausch zwischen den übrigen Modulen.

Tabelle 7.1: Gruppierung der Daten- und Verkehrsarten

Verkehrsart	Datenarten							
	Audio HQ NQ	Video HQ MQ NQ			Bd	Gk	Tx	ND
Kommunikation, Dialog in Echtzeit	D	B B						
Broadcast, Verteilung	C	C						
Station-Host Anfragen	C	C			C	A	C	A
Herkömmlicher Datenverkehr					C		C	
Verteilte Verarbeitung, Gemeinsames Arbeiten					C	A	A	A
Fernmessen und Fernwirken								A

HQ: hohe Qualität

NQ: niedrige Qualität

ND: numerische Daten

MQ: mittlere Qualität

Bd: Bilder

Gk: Graphik

Tx: Text

Protokollabhängiges Interface

Das protokollabhängige Interface Modul realisiert die Schnittstelle zwischen dem MultiGen Lastgenerator und einem Protokoll bzw. Protokollstapel, vgl. Abb. 7.2. Da sich zu untersuchende Protokollschichten typischerweise bezüglich der Dienstprimitive, über die sie mit der nächsthöheren Schicht kommunizieren (hier dem Lastgenerator der Simulationsumgebung), voneinander unterscheiden, muß dieses Interface Modul anwendungsspezifisch vom Benutzer angepaßt werden. Um eine benutzerfreundliche Handhabung zu gewährleisten, ist ein Gerüst generischer C-Funktionen vorgegeben, das keine Kenntnisse des Anwenders bezüglich der Implementierungsdetails des Lastgenerators erfordert. Eine vordefinierte Funktion wird jeweils bei einer der folgenden Aktionen des Schedulers aufgerufen:

- Initialisierung des Lastgenerators,
- Initiierung einer Verbindung,

- Bestätigung einer Verbindungseinrichtung,
- Datenübertragung,
- Auslösen einer Verbindung,
- Austausch von Management-Nachrichten,
- Lastgenerator-Programm beenden.

Dienstanforderungen, die durch das Protokoll an den Lastgenerator weitergeleitet werden sollen, werden vom Scheduler durch zyklischen Aufruf einer C-Funktion verarbeitet.

Weiterhin besitzt das Interface Modul eine Parameter-Sektion, in der Dienstgüteparameter mit Vorgabewerten vom Benutzer spezifiziert werden können. Diese werden während der Initialisierungsphase des Lastgenerators durch den Scheduler eingelesen und können als Dienstgütemerkmale für die Verbindungseinrichtung, Datenübertragung usw. in einer Dialogbox der graphischen Benutzeroberfläche ausgewählt werden, vgl. Abs. Graphische Benutzeroberfläche.

SPEET Application Interface

Das Modul SPEET Application Interface verbindet den Lastgenerator MultiGen mit den übrigen Komponenten des SPEET-Simulators wie der Simulationssteuerung SimCtrl, der Meßwerterfassung SimEval usw.

7.3.3 Die Simulationssteuerung des SPEET-Simulators

Die Simulationssteuerung SimCtrl übernimmt die zeitliche Synchronisation aller an einem Simulationslauf beteiligten Prozesse. SimCtrl verwaltet als eigenständiges Programm mit graphischer OSF/Motif Benutzeroberfläche zentral die Simulationszeit, die entweder in gleichgroßen Zeitintervallen (bei zeitdiskreten oder quasi-zeitkontinuierlichen Simulationen) oder Zeitintervallen beliebiger Größe (bei ereignisorientierten Simulationen) fortschreitet. Die Kommunikation mit den übrigen Simulationsanwendungen erfolgt verbindungsorientiert über Steuerkanäle, vgl. Abb. 7.2.

Ein Fortschreiten der Simulationszeit erfolgt, sofern alle Simulationsprozesse die vollständige Bearbeitung ihrer Aufgaben für den aktuellen Simulationszeitpunkt, bei ereignisorientierten Simulationen unter Angabe der nächsten gewünschten Simulationszeit, quittiert haben. Alle Funktionen, die von Simulationsanwendungen zur Kommunikation mit der Simulationssteuerung benötigt werden, wie Registrierung bei der Simulationssteuerung, Abfrage

der aktuellen Simulationszeit, Quittierung usw., sind in einem bindefähigen Bibliotheksmodul zusammengefaßt.

7.3.4 Statistische Auswertung von Simulationsmeßdaten

Das Programm SimEval des SPEET-Simulators wurde für die statistische Auswertung gemessener Simulationsdaten entwickelt.

Alle an einem Simulationslauf beteiligten Anwendungen können unabhängig voneinander Meßwerte mit Hilfe einer Registrierungszprozedur zur Auswertung anmelden. Die Kommunikation mit dem statistischen Auswertungsprogramm erfolgt äquivalent zur Kommunikation mit der Simulationssteuerung über Steuerkanäle, vgl. Abb. 7.2. Meßgrößen werden unter Angabe der Meßgrößenbezeichnung und Spezifikation des gewünschten Auswertungsverfahrens mit seinen Initialisierungsparametern, wie z.B. maximaler relativer Fehler, Größe des Konfidenzintervalls, minimaler Wert der Verteilungsfunktion usw., registriert.

Derzeit kann zwischen folgenden statistischen Auswertungsverfahren gewählt werden:

- LRE1 (Limited Relative Error) Verfahren entsprechend [91] zur Bestimmung der Verteilungsfunktion bzw. inversen Verteilungsfunktion nicht korrelierter Zufallssequenzen,
- LRE2 Verfahren entsprechend [92] zur Bestimmung der Verteilungsfunktion bzw. inversen Verteilungsfunktion beliebiger Zufallssequenzen,
- Batch-Means Verfahren.

Meßdaten werden von den Anwendungen während des Simulationslaufs unter Angabe einer eindeutigen Referenznummer, die bei der Registrierung je Meßgröße durch SimEval zugewiesen wird, an das Auswertungsprogramm übergeben.

Bei Beendigung des Simulationslaufs werden alle Meßwerte und die durch Auswertungsverfahren bestimmten Kenngrößen der Zufallssequenzen (Wertepaare der Verteilungsfunktion, Mittelwert, Streuung usw.) in Dateien abgespeichert.

Die statistisch ermittelten Kennwerte eines Simulationslaufs können direkt auf dem Bildschirm des Simulationsrechners graphisch dargestellt werden. Hierzu wurden die existierenden graphischen Darstellungsprogramme drawlre (für die LRE-Verfahren) und drawbtm (für das Batch-Means Verfahren)

der lehrstuhleigenen CNCL-Bibliothek [66] in das Auswertungsprogramm SimEval integriert.

7.3.5 Kommunikations- und Steuerkanäle

Die einzelnen Prozesse des SPEET-Simulators kommunizieren über Kommunikations- und Steuerkanäle miteinander, vgl. Abb. 7.2; für die Realisierung des Simulators unter dem Betriebssystem UNIX wurden, je nach benötigter Funktionalität, die Interprozeß-Kommunikationsfunktionen Stream Socket oder Shared Memory in Kombination mit Semaphoren verwendet, mit denen gesicherte Verbindungen zwischen Prozessen betrieben werden können. Derartige Interprozeß-Kommunikationsfunktionen werden von allen verbreiteten Multitasking-Betriebssystemen verfügbar gemacht, vgl. [5], so daß eine äquivalente Implementierung auf verschiedensten Hardwareplattformen möglich ist.

7.4 Anwendungsbeispiel: Leistungsbewertung der GSM Signalisierungsprotokolle

Die Protokolle des eingangs vorgestellten und durch Simulation untersuchten ATM-basierten Funknetzes (RMAN) liegen bisher nicht in formal spezifizierter Form vor. Deshalb wird zur Demonstration bzw. zur Validierung von SPEET auf existierende formal spezifizierte Protokolle (GSM, Dateitransfer, ISDN) zurückgegriffen.

7.4.1 Überblick

In diesem Abschnitt wird exemplarisch der Einsatz des SPEET Simulators für die Leistungsbewertung (formal spezifizierter) Kommunikationsprotokolle beschrieben.

Mit dem paneuropäischen, digitalen Mobilfunknetz GSM¹ wurde der Gedanke diensteintegrierender, digitaler Netze (ISDN) auf den Bereich der Mobilfunknetze ausgedehnt. Die Dienstintegration, Möglichkeit des Dienstwechsels bei bestehenden Verbindungen, Verfügbarkeit zusätzlicher Dienste (Supplementary Services) usw. stellen hohe Anforderungen an die Leistungsfähigkeit der Signalisierungsprotokolle. Besonders im Bereich

¹Global System for Mobile Communication

der Luftschnittstelle zwischen Mobilstation und Funkfeststation wird die Komplexität der Signalisierungsprotokolle durch Aufgaben wie Handover, Verwaltung von Funkkanälen etc. weiter erhöht. Die Leistungsfähigkeit des gesamten Netzes wird entscheidend durch die Performance der Signalisierungsprotokolle, insbesondere bei stark gestörten Funkkanälen, beeinflusst. Die Aufgabe bestand darin, die Leistungsfähigkeit der Signalisierungsprotokolle an der GSM-Luftschnittstelle unter typischen Funkverhältnissen aus Sicht der Mobilstation zu untersuchen, bewerten und evtl. zu verbessern, vgl. [8].

Abb. 7.5 zeigt die Anordnung der Signalisierungsprotokolle für Mobil- (MS), Basis- (BS) und Netzvermittlungsstation (MSC). Die Basisstation ist über Festnetzverbindungen mit der Netzvermittlungsstation verbunden, die Verbindungen zu anderen Vermittlungsstationen oder Kommunikationsnetzen (analoges Telefonnetz, ISDN, X.25 Netz usw.) verfügbar macht. Mobilstation (MS) und Basisstation (BS) kommunizieren miteinander über Funkkanäle, die durch eine Kombination von Frequenz- und Zeitmultiplex im 900 MHz Bereich realisiert sind, vgl. [26]. Zur Übertragung von Signalisierungsnachrichten zwischen MS und BS werden, nach einer kurzen Anforderungs- und Zuweisungsphase, dedizierte Steuerkanäle (Dedicated Control Channels) verwendet, die abhängig von der Kanalzuweisungsstrategie des Netzes über unterschiedliche Durchsätze verfügen. Entsprechend GSM-Standard [24] wurden folgende Kanalzuweisungsstrategien untersucht:

Very Early Assignment (VEA):

Zur Übertragung von Signalisierungsinformationen wird ein Vollratensteuerkanal mit der Bezeichnung FACCH (Fast Associated Control Channel) und einer Übertragungsrate von 9,2 kbit/s (aus Sicht der OSI-Schicht 2) verwendet; die Sicherungsschicht kann Datenblöcke der Größe 184 bit in regelmäßigen Intervallen von 20 ms (Blockwiederkehrzeit des FACCH) übertragen.

Non-Off Air Call SetUp (nonOACSU):

Bei dieser Strategie steht der Sicherungsschicht für die erste Signalisierungsphase ein SDCCH- (Slow Dedicated Control Channel) Steuerkanal mit einer Übertragungsrate von 589/765 kbits/s ($\approx 0,782$ kbit/s), die sich aus der Schicht 2 Datenblockgröße von 184 bit dividiert durch die SDCCH-Blockwiederkehrzeit von 3060/13 ms errechnen läßt, zur Verfügung. Nachdem alle relevanten Informationen der Mobilstation für die Weiterleitung des Rufs durch die Basisstation/Netzvermittlungsstelle übertragen wurden, wird ein FACCH-Steuerkanal für die Übertragung weiterer Signalisierungs-

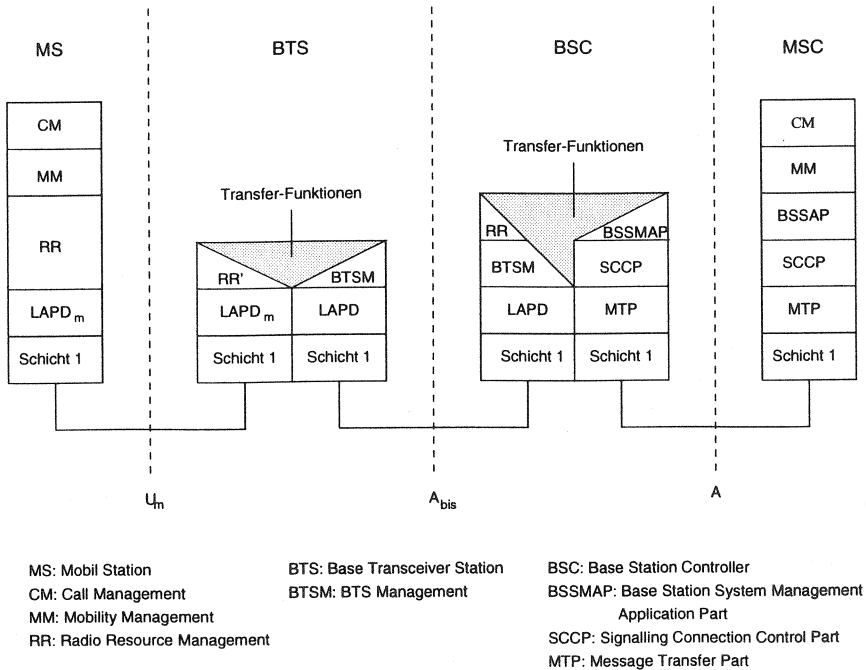


Abbildung 7.5: Signalisierungsprotokolle des GSM und Aufteilung auf die Netzkomponenten

informationen zwischen Mobil- und Basisstation benutzt.

Beide Verfahren tragen unterschiedlichen Gesichtspunkten der Netzplanung Rechnung, bei der einerseits ein hoher Komfort für den Teilnehmer, z.B. durch kurze Verbindungseinrichtungszeiten (VEA-Verfahren), andererseits ein sparsamer Umgang mit den verfügbaren Funkressourcen (beim non-OACSU) gefordert wird.

Das Sicherungsschichtprotokoll LAPD_m der Mobil- und Basisstation entsprechend GSM-Empfehlung 04.06 [24] basiert auf dem Schicht 2 Protokoll des ISDN, vgl. [54], und berücksichtigt die speziellen Randbedingungen des GSM-Funkkanals. Signalisierungsnachrichten der Schicht 3 werden über gesicherte Schicht 2 Verbindungen übertragen, das Eintreffen von Quittungen, als empfängerseitige Bestätigung für korrekt übertragene Schicht 2 Proto-

kolldateneinheiten, wird senderseitig durch einen Timer (T200) überwacht. Das LAPD_m-Protokoll arbeitet entsprechend GSM-Standard als Schiebefensterprotokoll mit Fenstergröße 1 (Stop-and-Wait Protokoll).

Die Netzschicht der Mobilstation gliedert sich im wesentlichen in drei Teilschichten auf, die Dienste wie

- Einrichtung, Unterhaltung und Auslösen kanalvermittelter Verbindungen, Kurznachrichtendienst, Rufweiterleitung, Gebührenabrechnung usw. (CM-Teilschicht),
- Verwaltung von Aufenthaltsdaten, Teilnehmeridentifikation, Prüfung der Zugangsberechtigung usw. (MM-Teilschicht),
- Verwaltung und Zuteilung von Funkkanälen, Zuordnung zu Zellen, Zellwechsel usw. (RR-Teilschicht)

realisieren. Die Schicht 3 Partnerinstanzen der Mobilstation sind auf Basisstation und Mobilvermittlungsstelle verteilt.

Die Spezifikationen der oben genannten Protokolle sind in den Serien 04.xx und 08.xx des GSM-Standards zu finden, die Serie 04 beschreibt Dienste und Funktionen der Protokolle, Serie 08 ihre Einordnung in die GSM-Teilsysteme MS, BS und MSC, vgl. Abb. 7.5.

7.4.2 Realisierung

Abb. 7.6 zeigt das Simulationsszenario für eine Mobil- und Basisstation. Netzseitig sind die benötigten GSM-Teilsysteme zu einem Gesamtsystem 'GSM-Netz' zusammengefaßt. Das Teilsystem 'Mobilstation' besteht aus dem SPEET-Lastgenerator MultiGen, der für Einrichtung und Auslösung von Verbindungen verwendet wird, und den GSM-Protokollen der Schichten 1-3, die formal in der Sprache SDL spezifiziert sind. Alle Protokollschichten der Mobilstation sind zu einem SDL-System, vgl. Abschn. 3.4, zusammengefaßt, jede Schicht wird durch einen SDL-Block repräsentiert.

Der SDL-Block für die GSM-Netzschicht einer Mobilstation enthält folgende Prozesse:

Interface_MS: Anpassung an die nächsthöhere Protokollschicht, hier SPEET-Lastgenerator MultiGen.

L3_Management: erzeugt, verwaltet, koordiniert und beendet die Prozesse Call_Control, Mobility_Management und Resource_Management.

Call_Control: dieser Prozeß enthält alle Funktionen für die Verbindungsverwaltung und repräsentiert den für die Simulation benötigten Funktionsumfang der Call Management Teilschicht, vgl. Abschn. 7.4.1.

Mobility_Management: übernimmt Einrichtung, Betrieb und Auslösung von MM-Teilschichtverbindungen für den Call_Control Prozeß, Teilnehmeridentifikation, Prüfung der Zugangsberechtigung.

Resource_Management: enthält die benötigten Funktionen zur Verwaltung und Zuteilung von Funkkanälen.

Der Schicht 2 Block der Mobilstation umfaßt folgende Prozesse:

LAPDmKernel_MS: enthält die vollständige Spezifikation des GSM-Sicherungsschichtprotokolls LAPD_m.

Sender_MS: versendet Schicht 2 Protokolldateneinheiten (PDE) auf den verschiedenen GSM-Kanälen, vgl. Abschn. 7.4.1. Je Kanal existiert ein Wartepuffer, in dem die PDEs zwischengespeichert werden; eine Protokolldateneinheit wird erst dann an die Schicht 1 weitergegeben, wenn der Prozeß Manager_MS (s.u.) die Sendebereitschaft (jeweils nach einem Zeitintervall der Größe einer Blockwiederkehrzeit) anzeigt.

Receiver_MS: empfängt die Dienstdateneinheiten der Schicht 1 und leitet sie nach einer Prüfung der zugehörigen Steuerinformation an den Prozeß LAPDmKernel_MS (Schicht 2) oder Resource_Management (Schicht 3) weiter.

L2Management_MS: erzeugt, verwaltet, koordiniert und beendet alle übrigen Prozesse des Schicht 2 Blocks.

Der Prozeß L1Manager_MS realisiert alle benötigten Funktionen der GSM-Schicht 1, wie z.B. Meldung der Sendebereitschaft für die verschiedenen GSM-Kanäle an die Sicherungsschicht, Empfang, Auswertung und Weitergabe von Schicht 1 Protokolldateneinheiten.

Für das System 'GSM-Netz', vgl. Abb. 7.6, werden dieselben Komponenten/-Protokollspezifikationen wie für das System 'Mobilstation' verwendet; die Protokollspezifikationen wurden um zusätzliche netzseitige Funktionen für den Betrieb von Punkt-zu-Mehrpunktverbindungen (Basisstation mit mehreren Mobilstationen), Wechsel der dedizierten Steuerkanäle, Auflösung von Mehrfachbelegungen eines dedizierten Steuerkanals durch Mobilstationen nach der Anforderungs- und Zuweisungsphase (Contention Resolution) usw. ergänzt.

Die Teilsysteme 'Mobilstation' und 'GSM-Netz' sind über einen Block mit der Bezeichnung 'Kanal- und Störmodell' verbunden, der folgende Aufgaben übernimmt:

- Nachbildung und Synchronisation aller benötigten GSM-Funkkanäle mit ihrer zeitlichen Verzahnung untereinander in Richtung Mobil- zur Basisstation und umgekehrt,
- Nachbildung typischer Störungen auf den Funkkanälen.

Für die Simulation von Funkkanalstörungen wurde auf Fehlermuster zurückgegriffen, die das European Telecommunication Standards Institute (ETSI) für die Bewertung von GSM-Protokollen verfügbar gemacht hat und die folgende Störungsursachen einbeziehen:

- Rayleigh Fading,
- Abschattungen,
- Gleichkanal- und Nachbarkanalstörer.

Diese Fehlermuster berücksichtigen nur die Fehlerschutz- und Fehlerkorrekturmechanismen (Interleaving und Faltungskodierung) der Schicht 1 bei Datenübertragung über GSM-Verkehrskanäle. Für die Bewertung von Signalisierungsprotokollen wurden die bitbezogenen Fehlermuster unter zusätzlicher Anwendung der GSM Fire Codierung, vgl. [27], aufbereitet und zusammengefaßt als Folge von 0 und 1 Werten in ASCII-Dateien abgespeichert: eine 1 kennzeichnet eine gestörte Schicht 2 Protokolldateneinheit, deren Bitfehler nach der Übertragung von den oben genannten Verfahren nicht korrigiert werden konnten, eine 0 kennzeichnet entsprechend eine fehlerfreie Protokolldateneinheit.

Es sind Dateien mit Fehlermustern für Signal-/Rauschleistungsverhältnisse (Carrier to Interferer Ratio, CIR) beim Empfänger von 3 dB (sehr schlechte Funkverhältnisse), 5 dB, 7 dB, 9 dB und 11 dB (gute Funkverhältnisse) verfügbar.

7.4.3 Ergebnisse der Simulation formal spezifizierter GSM-Protokolle mit SPEET und Bewertung

Im folgenden werden exemplarisch Simulationsergebnisse bezüglich der

- Verbindungseinrichtungsdauer für den Mobiltelefondienst,
- Rate erfolgreicher Verbindungseinrichtungen,

für die in Abschn. 7.4.1 beschriebenen Kanaluweisungsstrategien VEA und nonOACSU bei verschiedensten Signal- zu Rauschleistungsverhältnissen auf den Funkkanälen vorgestellt und bewertet.

Die Ergebnisse für die Verbindungseinrichtungsdauer berücksichtigen keine Verzögerungen durch Datenbankabfragen der Mobilvermittlungsstelle (bei Identifizierung, Authentifizierung usw.), Weiterleiten des Rufs in ein Mobil- oder Festnetz sowie die Reaktionszeit des gerufenen Teilnehmers (Zeit zwischen dem ersten Läuten des gerufenen Endgerätes und Abnahme des Hörers).

Zusätzlich wurde noch eine im GSM-Standard 04.08 vorgesehene, aber nicht spezifizierte Variante des Sicherungsschichtprotokolls LAPD_m untersucht: das Sequenznummernfeld der Schicht 2 Rahmen ermöglicht den Betrieb des LAPD_m als Schiebefensterprotokoll mit einer Fenstergröße größer 1. Zur Behebung von Sequenzfehlern in der Reihenfolge korrekt empfangener Schicht 2 Rahmen wurde das Verfahren der selektiven Wiederholung von Datenrahmen (Selective Repeat) gewählt, vgl. [96]. Die formale Spezifikation des LAPD_m, vgl. Abschn. 7.4.2, wurde um die benötigte Protokollfunktionalität erweitert.

Abb. 7.7 zeigt Simulationsergebnisse für die Verbindungseinrichtungsdauer in Abhängigkeit von

- einem Signal- zu Rauschleistungsverhältnis (CIR) von 5, 7, 9, 11 dB, der Kanaluweisungsstrategie VEA oder nonOACSU,
- einer Fenstergröße des LAPD_m-Protokolls von 1 (GSM-Standard) und 4 (erweitertes LAPD_m).

Bei einem Signal- zu Rauschleistungsverhältnis von 11 dB werden, auch bedingt durch die Fehlerkorrekturverfahren der Schicht 1, aus Sicht der Sicherungsschicht nahezu alle gesendeten Protokolldateneinheiten fehlerfrei empfangen (quasi störungsfreier Kanal).

Für ein CIR-Verhältnis von 11 dB ermittelte der Simulator eine mittlere Verbindungseinrichtungszeit von 1,09 s (VEA) und 3,93 s (nonOACSU). Diese Ergebnisse entsprechen exakt den Werten für die Verbindungseinrichtungszeit bei störungsfreien Kanälen, die unter Berücksichtigung der zu übertragenden Signalisierungsinformationen, verwendeten Steuerkanäle, Blockwiederkehrzeiten usw. rechnerisch ermittelt werden können. Es zeigt sich, daß bei Verwendung der Kanaluweisungsstrategie VEA eine deutlich geringere Einrichtungszeit für Verbindungen benötigt wird, was auf die geringere

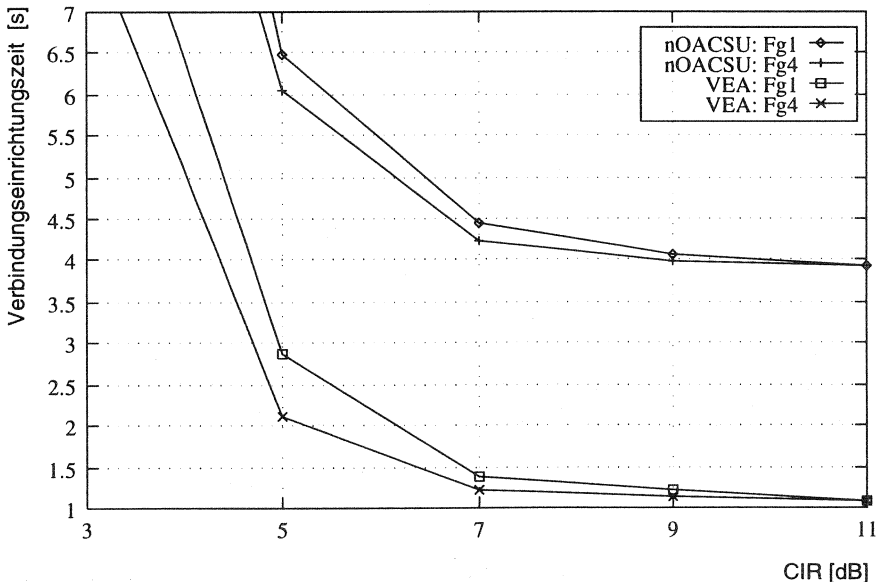


Abbildung 7.7: Verbindungseinrichtungszeit des LAPD_m in Abhängigkeit des Signal zu Rauschleistungsverhältnisses (CIR); Fg = Fenstergröße

Blockwiederkehrzeit des FACCH-Kanal im Gegensatz zum SDCCH- Kanal zurückzuführen ist, vgl. Abschn. 7.4.1.

Werden die Verbindungseinrichtungszeiten für VEA und nonOACSU bei schlechteren Funkverhältnissen bezogen auf die jeweiligen Werte bei 11dB miteinander verglichen, so weist das VEA Verfahren einen wesentlich höheren relativen Anstieg der Verbindungseinrichtungszeit im Gegensatz zur nonOACSU Strategie auf, z.B. bei CIR = 5 dB einen Faktor von ca. 2,61 (VEA) versus 1,58 (nonOACSU). Dieser Effekt läßt sich erst durch genauere Analyse der im GSM-Standard festgelegten Werte für den Schicht 2 Wiederholungstimer T200, vgl. Abschn. 7.4.1, erklären: für den FACCH-Steuerkanal ist der Wert für den Timer T200 auf 150 ms bei einer Blockwiederkehrzeit des FACCH von 20 ms, für den SDCCH auf 220 ms bei einer Blockwiederkehrzeit von 235 ms festgelegt. Erhält eine Schicht 2 Instanz innerhalb des Zeitintervalls, das durch den Timer T200 festgelegt ist, keine Emp-

fangsbestätigung für eine gesendete Protokolldateneinheit (PDE) durch ihre Partnerinstanz (z.B. durch Verlust der PDE), so sendet sie die PDE erneut. Im Falle des LAPD_m nach Standard 04.08 (Fenstergröße 1, stop and wait Protokoll) wartet die sendende Schicht 2 Instanz den Eingang der Empfangsbestätigung ab, bevor sie eine andere Protokolldateneinheit sendet. Bezogen auf die Kanaluweisungsstrategie VEA (verwendet nur FACCH) bzw. nonOACSU (verwendet SDCCH und FACCH) bedeutet das bei Ausbleiben einer Empfangsbestätigung:

- bei Datenübertragung auf dem FACCH-Kanal wartet die Schicht 2 Instanz mehrere Blockwiederkehrzeiten ab (nach Ablauf der Blockwiederkehrzeit kann gesendet werden), bevor sie die Protokolldateneinheit erneut überträgt.
- bei Datenübertragung auf dem SDCCH-Kanal kann die PDE nach Ablauf einer Blockwiederkehrzeit wiederholt werden.

Die erhöhte Wartezeit für Wiederholungsübertragungen auf dem FACCH-Kanal wirkt sich deshalb bei gestörten Funkkanälen negativ für das VEA Verfahren aus.

Der Vergleich der Verbindungseinrichtungszeiten bei einem CIR-Verhältnis von 11 dB zeigt für beide Kanaluweisungsstrategien keine Verbesserung durch Betrieb des LAPD_m als Schiebefensterprotokoll mit Fenstergröße 4 gegenüber 1. Bedingt durch die zeitliche Verzahnung von Sendekanälen (SDCCH und FACCH) und Übertragungskanälen für Empfangsbestätigungen im GSM-System, erhält eine sendende Schicht 2 Instanz bei quasi störungsfreien Funkverhältnissen bereits innerhalb eines Blockwiederkehrzeit-Intervalls eine Quittung für eine Protokolldateneinheit, so daß sie nach Ablauf der Blockwiederkehrzeit erneut senden kann und Möglichkeiten eines vergrößerten Sendefensters nicht zum tragen kommen. Anders stellt sich die Situation bei gestörten Funkkanälen dar, wenn Protokolldateneinheiten oder Quittungen bei der Übertragung zerstört werden: vor Ablauf des Wiederholungstimers T200 kann die Schicht 2 weitere PDEs übertragen. Wie in Abb. 7.7 zu sehen ist wirkt sich dies stärker für das VEA-Verfahren aus, was auf die oben beschriebene Relation der Blockwiederkehrzeit zum Wert des Timers T200 zurückzuführen ist. Bei Übertragungen auf dem SDCCH-Kanal wirkt sich die Vergrößerung des Sendefensters wegen des Ablaufs des Wiederholungstimers T200 innerhalb einer Blockwiederkehrzeit nicht aus. Die geringen Verbesserungen für das nonOACSU-Verfahren bei Fenstergröße 4 gegenüber 1 (ca. 3 - 7 %) sind auf

die Nutzung des FACCH-Kanals während der zweiten Signalisierungsphase zurückzuführen.

Tabelle 7.2 zeigt die gemessenen Raten erfolgreicher Verbindungseinrichtungen bei unterschiedlichen CIR-Verhältnissen. Für ein CIR-Verhältnis von 3 dB liegt keine gesicherte Aussage darüber vor, ob im realen Betrieb eine Synchronisation von Mobil- und Basisstation noch gewährleistet ist, weshalb die angegebenen Werte nur als Anhaltspunkt dienen können.

Tabelle 7.2: Rate erfolgreicher Verbindungseinrichtungen

	11 dB	9 dB	7 dB	5 dB	3 dB
VEA	100 %	100 %	100 %	69,7 %	~ 30 %
nOACSU	100 %	100 %	97,5 %	62,9 %	~ 14 %

Insgesamt sinkt die Rate erfolgreicher Verbindungseinrichtungen mit steigenden Störungen auf den Funkkanälen, was im wesentlichen auf das Fehlschlagen der Einrichtung einer gesicherten Schicht 2 Verbindung zurückzuführen ist. Ist eine gesicherte Schicht 2 Verbindung eingerichtet, so wird sie erst nach 34 aufeinanderfolgenden, erfolglosen Datenübertragungen auf dem FACCH-Kanal, 23 für den SDCCH-Kanal, vorzeitig ausgelöst. Die höhere Anzahl möglicher aufeinanderfolgender Wiederholungen auf dem FACCH-Kanal begründet die höhere Rate erfolgreicher Verbindungseinrichtungen für das VEA-Verfahren bei stark gestörten Funkkanälen (CIR = 5 dB). Die relative Verbesserung bei gestörten Funkkanälen durch eine LAPD_m-Fenstergröße von 4 ist auf die geringere Zahl möglicher Sequenzfehler von Schicht 2 Protokolladateneinheiten (ein durch Störungen erzeugter Sequenzfehler führt zur Auslösung der Schicht 2 Verbindung) zurückzuführen.

Zusammenfassend kann gesagt werden, daß sich das VEA-Verfahren als robuster bezüglich der Anfälligkeit gegen Funkkanalstörungen im Vergleich zum nonOACSU-Verfahren erweist und Verbindungseinrichtungszeiten in der gleichen Größenordnung wie beim terrestrischen ISDN (typisch: 0,8 bis 1 s) aufweist, was jedoch eine stärkere Nutzung der verfügbaren Funkkanäle im Gegensatz zum nonOACSU-Verfahren erfordert. Schwächen weist

das VEA-Verfahren entsprechend GSM-Standard in Verbindung mit den LAPD_m-Protokollparametern Wiederholungstimer bzw. Fenstergröße auf, die für nonOACSU-Verfahren optimal angepaßt sind.

Der SPEET Hardware-Emulator

Wie in Abschn. 7.2 beschrieben, umfaßt die zweite Realisierungsstufe des SPEET-Simulators die Entwicklung eines modularen Hardware-Emulators, der, entsprechend der integrierten Softwareentwicklungsmethodik, beispielsweise Leistungskenngrößen für die Hardwaredimensionierung eines Kommunikationssystems ermittelt, Protokollspezifikationen bezüglich einer effizienten Ausnutzung der Systemhardware analysiert oder Entwicklungswerkzeuge für Kommunikationsprotokolle miteinander vergleichbar macht.

8.1 Kenngrößen zur Leistungsbewertung von Prozessorsystemen

Eine Analyse und Bewertung existierender Verfahren zur Leistungsbewertung von Hardwaresystemen, z.B. Benchmarks, Kernels, Mix-Verfahren usw. ergab, vgl. [68], daß

- die von diesen Verfahren ermittelten Kenngrößen (MIPS¹-Zahl, Dhrystone-MIPS o.ä.) für einen Leistungsvergleich von Prozessorsystemen untereinander kaum geeignet sind,
- die ermittelten Kenngrößen signifikante Prozesseigenschaften wie den Prozesstyp (CISC oder RISC), Wortlänge (16 bit, 32 bit, usw.) nicht berücksichtigen,
- grundlegende Untersuchungen zur Entwicklung eines Benchmarks, Kernels etc. für Kommunikationsprotokolle bisher nicht durchgeführt wurden und bei Anwendung existierender Verfahren keine Gewähr für die Gültigkeit der resultierenden Aussagen übernommen werden kann.

Weiterhin liefern die genannten Verfahren mit ihren Meßwerten keine Aussage darüber, ob die Leistungsgrenze eines Prozessorsystems durch den Pro-

¹Million Instructions per Second

tokollbetrieb unter typischen Lastbedingungen bereits erreicht bzw. überschritten wird.

Zielsetzung für den SPEET Hardware-Emulators ist deshalb, neben Meßgrößen wie dem statischen und dynamischen Speicherbedarf eines Protokollstapels, Befehlsstatistiken usw. eine zeitbasierte Meßgröße verfügbar zu machen, die die Auslastung des Prozessorsystems unter gegebenen Lastbedingungen beschreibt. Der Zeitbezug der Prozessorauslastung erfordert die Definition von Grenzwerten, zu denen eine benötigte Rechenleistung in Relation gesetzt wird.

Die prinzipiellen Überlegungen zur Definition einer Auslastungskenngröße basieren auf der Tatsache, daß die terminorientierte Abfertigung von Aufträgen ein grundlegendes Merkmal von Kommunikationssystemen/-protokollen ist: Protokollspezifikationen enthalten klare Vorgaben bezüglich des Zeitraums, innerhalb dessen Aktionen, wie z.B. eine Verbindungseinrichtung, durchgeführt werden sollen. Häufig ergeben sich Zeitbedingungen auch dadurch, daß die verfügbaren Übertragungskapazitäten eines Kommunikationsnetzes optimal genutzt werden sollen.

Aus diesen Überlegungen ergibt sich der Ansatz, Protokolle zur Bewertung in kurze, logisch zusammenhängende Teilabschnitte zu unterteilen (z.B. ein bis zwei Zustandsübergänge), die für die Ausführung benötigte Zeit zu messen und das Verhältnis zur maximal für diesen Teilabschnitt vorgesehenen Zeitspanne zu bilden. Dieser Wert repräsentiert gleichzeitig die zeitliche Reserve eines betrachteten Prozessorsystems zur Durchführung anderer Protokollaktionen und wird deshalb als Kapazitätsausnutzung bezeichnet:

Definition:

Die *Kapazitätsausnutzung* $KA(i)$ eines Rechensystems bei der Bearbeitung eines Protokolls oder Protokollstapels berechnet sich aus der benötigten Ausführungszeit $AZ(i)$ für eine i -te Protokoll-Teilsequenz im Verhältnis zur maximal im Sinne der terminorientierten Abfertigung von Aufträgen verfügbaren Zeit $AZ_{max}(i)$ für die Bearbeitung der i -ten Teilsequenz.

Anhand der Meßgröße Kapazitätsausnutzung können verschiedene Implementierungen eines Protokollstapels bezüglich ihrer Effizienz und Performance oder unterschiedliche Prozessorsysteme hinsichtlich Bearbeitungseffizienz eines gegebenen Protokollstapels miteinander verglichen werden.

8.2 Konzept und Realisierung des SPEET Hardware-Emulators

Die grundlegende Entwurfsidee für den SPEET Hardware-Emulator besteht darin, Prozessorsysteme mit ihren wesentlichen Merkmalen durch ein Softwarepaket nachzubilden, um den Betrieb eines Protokollstapels im 'realen' Kommunikationssystem simulieren zu können.

Abb. 8.1 a) zeigt die Einordnung des Hardware-Emulators in die SPEET-Simulationsumgebung, vgl. Abb. 7.2, und Abb. 8.1 b) den strukturellen Aufbau.

Der Hardware-Emulator übernimmt während des Simulationslaufs die 'realen' Ausführung eines Protokolls bzw. Protokollstapels, bildet dabei das Zeitverhalten des Prozessorsystems nach und bestimmt die interessierenden Meßgrößen wie Kapazitätsausnutzung, Speicherbedarf usw. Die zu untersuchenden Protokolle müssen als Binärkode des emulierten Prozessortyps vorliegen, was durch Kompilation + Binden des Quelltextes erreicht wird, der ausgehend von einer formalen Spezifikation erzeugt wird, vgl. [98].

Das Prozessor-Modul bildet alle benötigten Hardwarekomponenten eines Mikroprozessors nach; in der ersten Implementierungsstufe des SPEET Hardware-Emulators wurde der gesamte Funktionsumfang des (verbreiteten) Intel 80386 Prozessors implementiert, vgl. [48]. Nachgebildet werden beispielsweise die allgemeinen Prozessorregister (AX, BX, SI usw.), die prozessor-internen Flags (Carry, Parity, Direction, Trap usw.), Befehlsdekodier- und Ausführungseinheit. Die beiden letztgenannten Einheiten werden zusammengefaßt durch separate C-Funktionen je Prozessorbefehl realisiert, die nach der Befehlsidentifikation durch den Scheduler (s.u.) die Operandendekodierung, Adreßberechnung, Berechnung und Ablage des Ergebnisses usw. übernehmen.

Weitere Bausteingruppen eines Prozessorsystems, wie z.B. der dynamische Systemspeicher, die Kontrolleinheit für den Systembus oder periphere Bausteine, werden durch das Modul 'Hardwarekomponenten' simuliert.

Die möglichst exakte Nachbildung des Zeitverhaltens (Timing) eines Prozessorsystems bei der Protokollausführung ist von zentraler Bedeutung für die Genauigkeit der resultierenden Meßergebnisse. Bezüglich des Systemspeichers werden beispielsweise folgende Kenndaten einer geplanten Realisierung durch Festlegung der Baustein-Typen (stat./dyn. RAM), Art der Speicherverschänkung, Prozessortakt usw. bestimmt, die als Simulationsparameter vorgegeben werden:

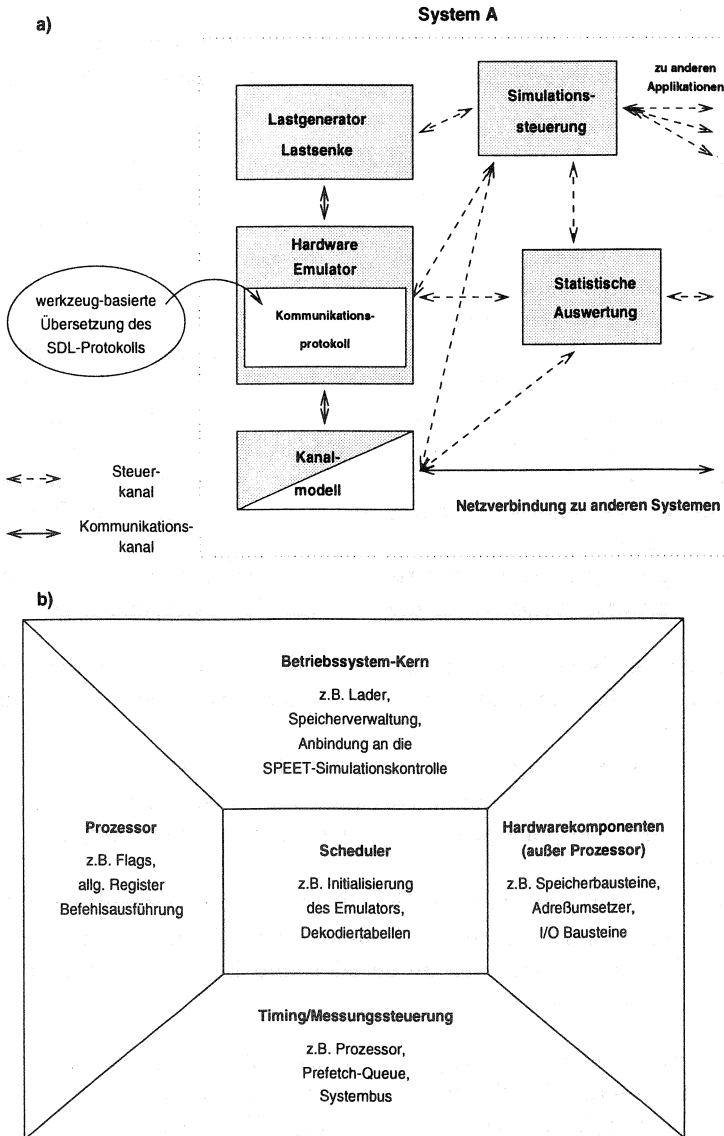


Abbildung 8.1: Einordnung des Hardware-Emulators in die SPEET-Simulationsumgebung und struktureller Aufbau des Hardware-Emulators

- Zugriffszeit, Zykluszeit mit evtl. Wartezyklen,
- Reduktionsfaktor der Zykluszeit durch Speicherverschränkung,
- Wartezyklen durch periodische Aufrischung der Bausteininhalte (Refresh).

Deutlich schwieriger ist es, die 'benötigte Rechenzeit' für den Simulationslauf bzw. definierte Befehlssequenzen genau zu bestimmen. Für den SPEET-Hardware-Emulator wurde zunächst folgendes Verfahren eingesetzt, das sich besonders im Hinblick auf die einfache Erweiterung des Emulators um weitere Prozessortypen auszeichnet: für jeden im Rahmen des Simulationslaufs ausgeführten Prozessorbefehl wird die benötigte Taktzyklenzahl anhand einer Timing-Tabelle bestimmt und zum bisher ermittelten Summenwert der Taktzyklen addiert. Die Timing Tabelle kann der Prozessor-Dokumentation des Herstellers entnommen werden, z.B. [48]. Tab. 8.1 zeigt beispielhaft einen Auszug aus der Timing-Tabelle für den Intel 80386 Prozessor. Die Gesamtrechenzeit ergibt sich durch Multiplikation des Gesamtwertes benötigter Taktzyklen mit dem invertierten Wert der Prozessortaktfrequenz.

Bei Verwendung der o.g. Tabellenwerte ist zu berücksichtigen, daß es sich hierbei um 'Idealwerte' handelt, für die folgende Voraussetzungen gelten:

- der Befehl ist bereits gelesen (d.h. er befindet sich in der Prefetch-Queue), dekodiert und liegt in der Befehlswarteschlange zur Ausführung bereit,
- Buszyklen bei Speicherzugriffen erfordern *keine* Wait-States;
- der Bus steht jederzeit zur Verfügung, d.h. es finden keine Zugriffe durch andere Komponenten statt, die einen sofortigen Buszyklus verhindern,
- während der Ausführung eines Befehls treten keine Exceptions auf,
- die Operanden im Speicher sind entsprechend ihrem Format ausgerichtet, d.h. Worte beginnen an geraden und Langworte an geraden, durch vier teilbaren Adressen.

Daher ist es unter Umständen nötig, entsprechende Zeitkorrekturfaktoren zu berücksichtigen, sofern z.B. die Prefetch-Queue aufgrund häufiger Systembuszugriffe nicht parallel zur Befehlsausführung gefüllt werden kann.

Das Modul 'Betriebssystemkern' enthält alle Funktionen eines rudimentären Betriebssystems und Systemprogramme, die bei realen Prozessorsystemen

Tabelle 8.1: Timing-Tabelle des Intel 80386 (Auszug)

Instruktion	Zyklen pro Instruk- tion	Beschreibung
AND AX, imm16	2	logische UND-Verknüpfung eines direkten, 16 Bit langen Datums mit dem Inhalt des AX-Registers
AND r/m32, imm32	2/7	logische UND-Verknüpfung eines direkten, 32 Bit langen Datums mit dem Inhalt eines Registers oder einer Speicherstelle
AND r8, r/m8	2/6	logische UND-Verknüpfung eines Register- oder Speicherinhalts mit dem Inhalt eines Registers
CLI	8	löscht das <i>Interrupt Enable</i> -Flag
CALL rel16	7+m	Prozeduraufruf, Einsprungadresse durch <i>Displacement</i> rel. zur nächsten Instruktion
CALL r/m32	7+m/ 10+m	indirekter Prozeduraufruf, Einsprungadresse als Register- oder Speicherinhalt gegeben
MUL AX, r/m16	9-22/ 12-25	Multiplikation des AX-Registerinhalts mit dem Inhalt eines Registers oder einer Speicherstelle

typischerweise in einem EPROM² abgelegt sind. Zu den wesentlichen Komponenten gehört ein Lader, der eine ausführbare Programmdatei in den simulierten Systemspeicher transferiert, anhand der Relozierungstabelle die Adressen im Codesegment aktualisiert und die 'Prozessorregister' mit Initialisierungswerten belegt. Weiterhin werden Betriebssystemfunktionen verfügbar gemacht, die einem auszuführenden Protokoll bzw. Protokollstapel den Zugriff auf Betriebsmittel des Prozessorsystems (z.B. Allokierung/Deallokierung von Systemspeicher, Abfrage der Systemzeit usw.) gestatten.

²Erasable Programmable Read Only Memory

Das zentrale Modul 'Scheduler' übernimmt die Ablaufsteuerung des gesamten Hardware-Emulators:

- beim Start des Simulationslaufs werden alle Teilkomponenten initialisiert und die ausführbare Protokolldatei in den 'Systemspeicher' geladen.
- die Programmausführung beginnt an der durch den Befehlszeiger (Instruction Pointer, IP) bezeichneten Speicherstelle im Codesegment.
- der binäre Befehlscode wird aus dem 'Systemspeicher' gelesen und die Instruktion über die prozessorspezifische Dekodiertabelle identifiziert.
- es folgt die Befehlsausführung wie oben beschrieben inkl. Aktualisierung des Befehlszeigers. Dieser Vorgang (Befehlscode lesen, Identifikation, Ausführung) wiederholt sich für jede weitere Instruktion.

8.3 Ergebnisse für ein formal spezifiziertes Dateitransferprotokoll

Zur Verifikation des in Abschn. 8.2 vorgestellten Konzepts wurden Untersuchungen zunächst an einem vereinfachten, formal spezifizierten Dateitransferprotokoll durchgeführt und die Ergebnisse mit Messungen an einem realen System verglichen. Aufgabe einer sendenden Protokollinstanz ist es, nach erfolgreicher Verbindungseinrichtung mit einer Partnerinstanz, Dienstdateneinheiten einer überlagerten Instanz an die Struktur des Übertragungskanals anzupassen (Segmentierung) und transparent zu versenden. Von der empfängerseitigen Protokollinstanz werden die Dienstdateneinheiten wiederhergestellt (Reassembling) und an die überlagerte Schicht übergeben. Diese einfache Protokollfunktionalität wurde bewußt gewählt, um die protokollinternen Abläufe mit Hilfe der verfügbaren Hardwaremeßwerkzeuge, vgl. Abschn. 8.4, und anhand der Meßergebnisse nachvollziehen zu können. Weiterhin wurde das formal spezifizierte Protokoll mit einem Lempel-Ziv-Welsch (LZW) Kompressionsalgorithmus (Kompressionsfaktor für Textdateien ca. 40 - 60 %) versehen, vgl. [109], der eine optionale Datenkompression vor Übertragung gestattet.

Simuliert wurde die Übertragung von Daten zwischen zwei Partnerinstanzen des Dateitransferprotokolls/SPEET-Lastgenerators entsprechend dem in Abb. 7.2 dargestellten Szenario, das spiegelbildlich um ein Protokollmodul und einen SPEET-Lastgenerator erweitert wurde. Jede Protokollinstanz verfügt über einen separaten Lastgenerator, der die zu übertragenden

Datenpakete generiert bzw. die empfangenen Datenpakete abspeichert. Der Lastgenerator einer sendenden Protokollinstanz liest zur Generierung von Datenpaketen sukzessive den Inhalt einer Textdatei im ASCII-Format, die Zwischenankunftszeit für Datenpakete kann beliebig vorgegeben werden. Die Simulation wurde wie folgt parametrisiert:

- die Simulationsuhr gibt einen konstanten Zeittakt von $\Delta t = 1$ ms vor. Diese wurde gewählt, um zeitbezogene Meßfehler bei Vergleichsmessungen an einem realen System weitgehend zu eliminieren, vgl. Abschn. 8.4.
- das Kanalmodell bildet einen störungsfreien TDM-Kanal mit einer Blockwiederkehrzeit von $t_B = 5$ ms nach. Pro Kanalzugriff wird ein Byte übertragen, was dem Zugriff auf einen seriellen PC-Kommunikationsbaustein entspricht.
- die Datenübertragung erfolgt unidirektional zwischen einem Sender und einem Empfänger, die Meßwerte werden sendeseitig ermittelt.
- der sendeseitige Lastgenerator übergibt in äquidistanten Abständen von 50 ms Datenpakete einer konstanten Größe von 10 byte an die zugehörige Instanz des Dateitransferprotokolls. Die Größen sind so gewählt, daß ständig Daten zur Übertragung bereitstehen (Vollastbetrieb).

Abb. 8.2 zeigt Meßergebnisse des SPEET Hardware-Emulators für ein Prozessorsystem mit Intel 80386 Prozessor (Taktfrequenz 16 MHz). Dargestellt ist für eine kurze Datenübertragungssequenz (inkl. Verbindungseinrichtung und -auslösung) der ermittelte Rechenzeitbedarf in Abhängigkeit von der Simulationszeit. Jeder Impuls des Histogramms entspricht dem Rechenzeitbedarf des Protokolls während des jeweiligen Simulationsintervalls von 1 ms. Die Werte des Rechenzeitbedarfs liefern keine direkte Aussage bezüglich der Auslastung des Prozessorsystems, sie werden später anhand definierter Randbedingungen in die Kapazitätsausnutzung, vgl. Abschn. 8.1, überführt. Abb. 8.2 bietet die Möglichkeit einer genauen Analyse der einzelnen Protokollbearbeitungsphasen:

- keine Daten stehen zur Übertragung auf dem Kanal an: diese Bearbeitungsphasen treten im dargestellten Bereich viermal auf:
 - vor der Verbindungseinrichtung, Bereich 10 - 57 ms,
 - nach der Verbindungseinrichtung bis zum Beginn der Datenübertragung (102 - 160 ms),

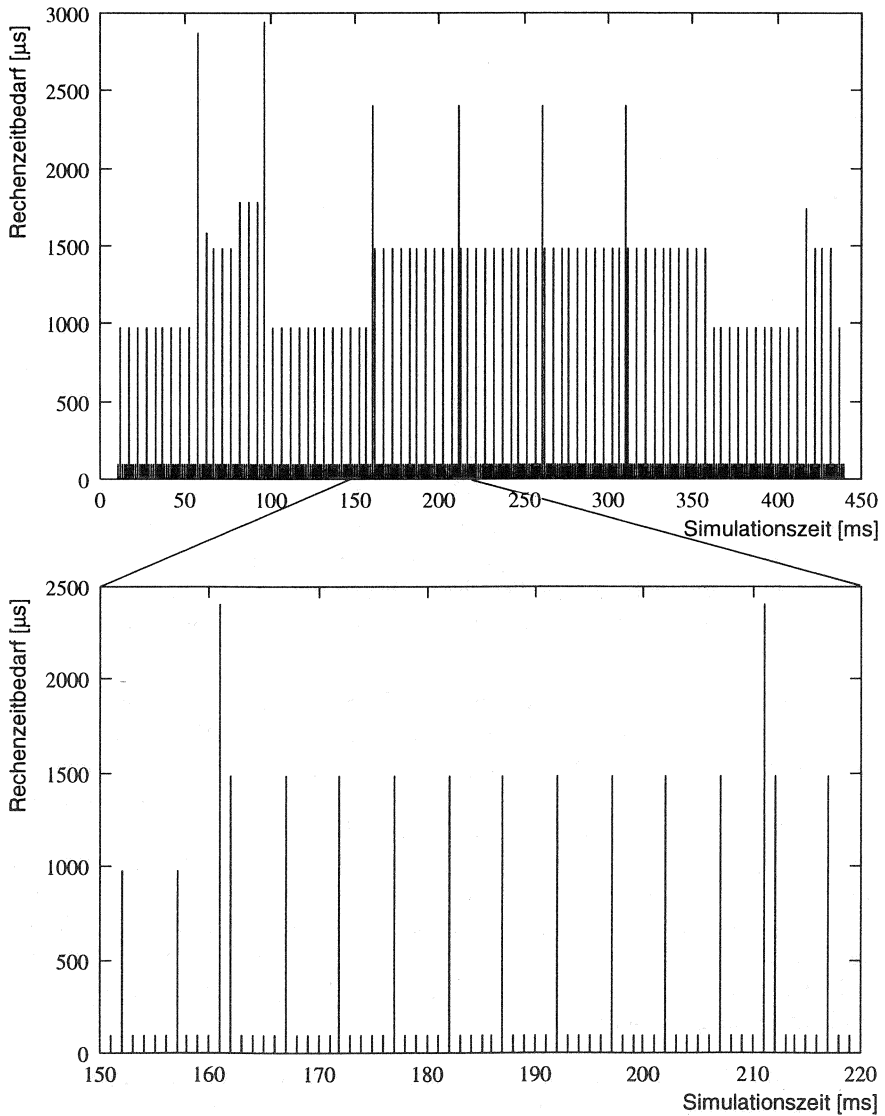


Abbildung 8.2: Rechenzeitbedarf für die Ausführung des Dateitransferprotokolls (Intel 80386 Prozessorsystem, 16 MHz Taktfrequenz)

- nach Beendigung der Datenübertragung bis zur Verbindungsauslösung (363 - 418 ms),
- nach Auslösung der Verbindung (> 438 ms).

Diese Phase weist ein sich ständig wiederholendes Profil auf, das in Abb. 8.2 b) zwischen 150 und 160 ms deutlich zu sehen ist: die Übertragungsbereitschaft des Kanals wird dem Protokoll durch Ablauf eines simulationsspezifischen Timers (alle 5 ms) signalisiert, für dessen Bearbeitung das Protokoll einen Rechenzeitbedarf von ca. 1000 μ s benötigt. Der Rechenzeitbedarf zwischen diesen Ereignissen entsteht durch den Protokollscheduler, der zyklisch alle Warteschlangen bezüglich abzuarbeitender Aufträge pollt.

- Verbindungseinrichtung im Bereich von 57 bis 97 ms der Simulationszeit.
- Datenübertragung zwischen 161 und 358 ms Simulationszeit: die Spitzen des Rechenzeitbedarfs von ca. 2400 μ s bei einer Simulationszeit von $161 \text{ ms} + n \cdot 50 \text{ ms}$ ($0 \leq n \leq 3$) resultieren aus der Bearbeitung von Dienstdateneinheiten der überlagerten Schicht (hier: Lastgenerator), die für die Übertragung auf dem Kanal segmentiert werden müssen. Ein Rechenzeitbedarf von 1500 μ s bei einer Simulationszeit von 162 ms ff. entsteht durch die Bearbeitung des Timers für die Übertragungsbereitschaft (s.o.) und das Versenden von Daten.
- Verbindungsauslösung: die entsprechende Dienstanforderung des Lastgenerators erhält das Protokoll zum Simulationszeitpunkt 418 ms, der Vorgang ist bei ca. 438 ms abgeschlossen.

Für die Berechnung der Kapazitätsausnutzung müssen zeitliche Grenzwerte definiert werden, vgl. Abschn. 8.1, auf die der Rechenzeitbedarf bezogen wird. Diese Grenzwerte sind protokollspezifisch und von technischen Kenngrößen des Kommunikationssystems (z.B. Kanalkapazität, Puffergröße der Schnittstellenbausteine) abhängig. Im Bezug auf das betrachtete Dateitransferprotokoll wird ein typisches Optimierungsziel der Kommunikationstechnik zur Grenzwertdefinition herangezogen: die optimale Nutzung der Übertragungskapazität des Kanals im Sinne einer maximalen Dienstgüte (hier: Durchsatz). Das Kommunikationssystem (Dateitransferprotokoll + Prozessorsystem) soll so ausgelegt sein, daß jeder verfügbare Zeitschlitz des (störungsfreien) TDM-Kanals für die Übertragung anstehender Dateneinheiten genutzt wird. Bezugsgröße zur Berechnung der Kapazitätsausnutzung ist die Blockwiederkehrzeit t_B . Der Wert der Kapazitätsausnutzung für ein

Zeitintervall t_B errechnet sich aus der Summe des zugehörigen Rechenzeitbedarfs, wobei der Rechenzeitbedarf des Protokollschedulers im 'Leerlauf' (Pollen der Warteschlangen) einmalig berücksichtigt wird, dividiert durch die Blockwiederkehrzeit t_B . Abb. 8.3 zeigt die Kapazitätsausnutzung des Prozessorsystems für den in Abb. 8.2 dargestellten Ausschnitt der Simulationszeit, der entstandene Rechenzeitbedarf zur Bearbeitung des simulationspezifischen Timers (Signalisierung der Kanal-Übertragungsbereitschaft) wurde ebenfalls eliminiert.

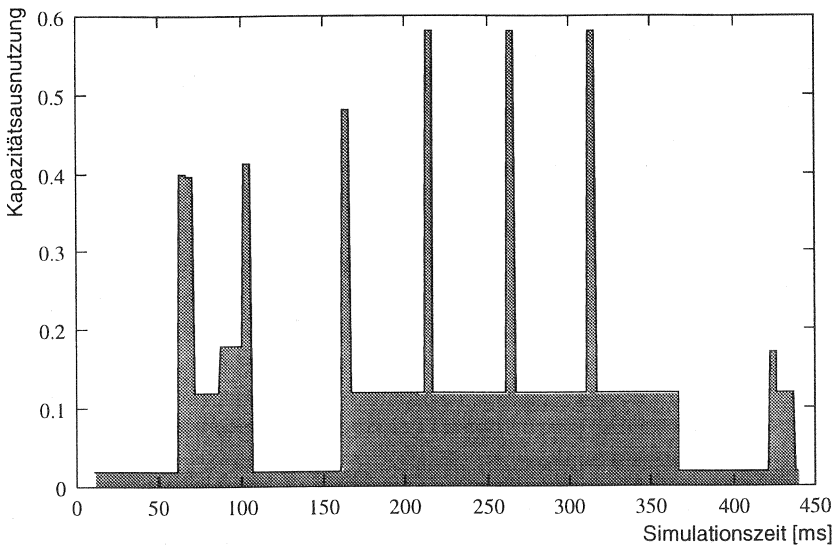


Abbildung 8.3: Kapazitätsausnutzung bei Ausführung des Dateitransferprotokolls (Intel 80386 Prozessorsystem, 16 MHz Taktfrequenz)

Die Ergebnisse zeigen, daß das untersuchte Kommunikationssystem den Übertragungskanal unter den gegebenen Bedingungen optimal nutzen kann, es verbleibt eine Kapazitätsreserve zwischen 42 % (min.) und 98 % (max.). Abb. 8.4 zeigt Ergebnisse für eine analoge Simulation, jedoch mit Datenkompression (LZW-Algorithmus) vor Übertragung. Die Kapazitätsausnutzung von mehr als 100 % für die Bearbeitung von Dienstdateneinheiten des

Lastgenerators zeigt, daß in einem realen System aufgrund der sequentiellen Abarbeitung aller Teilaufträge (segmentieren, codieren, übertragen) 10 % der Übertragungskapazität nicht genutzt und in Folge die protokollinternen Puffer für Daten bei längeren Datentransferphasen unter Vollast überlaufen würden. Die genannten Probleme können durch Modifikation der Protokollspezifikation oder der Hardwarekonfiguration (z.B. Taktfrequenz des Prozessorsystems) beseitigt werden; das Beispiel soll verdeutlichen, wie mit Hilfe des SPEET Hardware-Emulators Performance-Engpässe frühzeitig erkannt werden können.

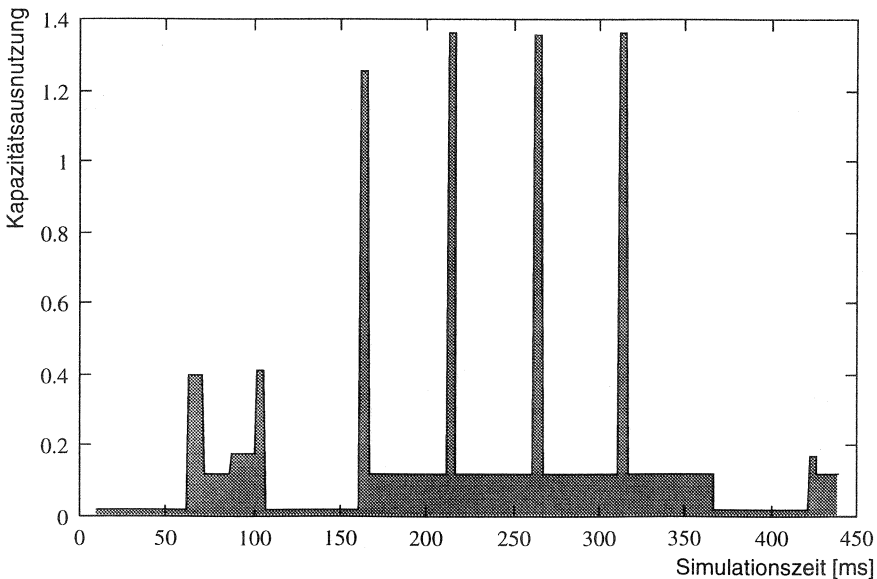


Abbildung 8.4: Kapazitätsausnutzung bei Ausführung des Dateitransferprotokolls mit LZW-Datenkompression (Intel 80386 Prozessorsystem, 16 MHz Taktfrequenz)

Abb. 8.5 faßt weitere interessante Meßergebnisse zusammen: Während des Simulationslaufs erstellt der SPEET Hardware-Emulator eine Statistik der Auftrittshäufigkeit bezüglich jeder ausgeführten Prozessorinstruktion. Um einen prozessorunabhängigen Vergleich verschiedener Untersuchungen zu

ermöglichen, werden die Ergebnisse nach folgenden Befehlsarten gruppiert, in die sich der Befehlssatz von Mikroprozessoren einteilen läßt:

- arithmetische Befehle (Addieren, Subtrahieren, usw.),
- logische Befehle (Und -, Oder-Verknüpfung, usw.),
- Transferbefehle (Register-Register Transfer, Register-Speicher Transfer, usw.),
- Verzweigungsbefehle (Sprunganweisung, Aufruf von Unterrouinen, usw.),
- Sonderbefehle (Interrupt, Exception, usw.).

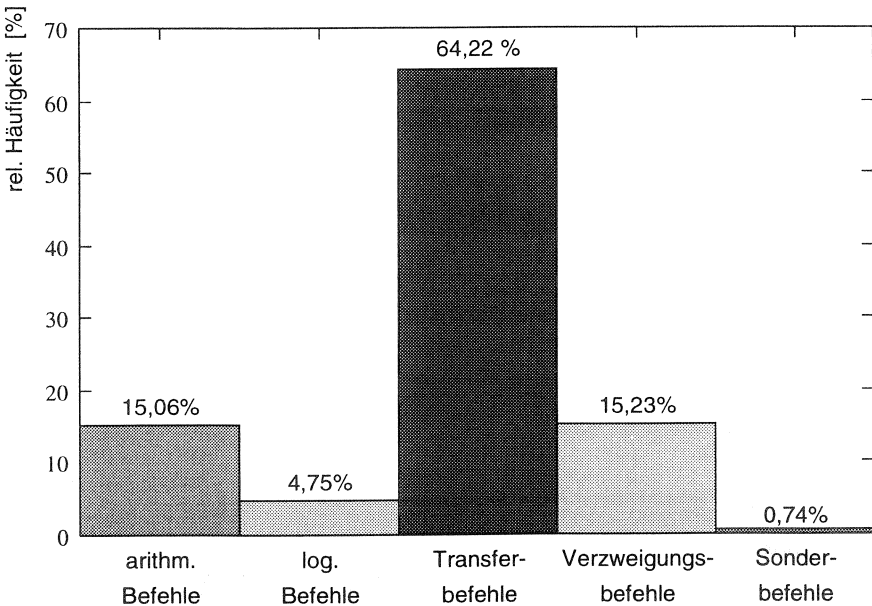


Abbildung 8.5: Befehlsstatistik bei der Ausführung des Dateitransferprotokolls

Folgende Teilaspekte sind für eine potentielle Systementwicklung von Bedeutung:

- die dargestellte Befehlsstatistik gilt gleichermaßen für den Protokollbetrieb mit/ohne Datenkompression. Die Abweichung der gemessenen Einzel-Verteilungen voneinander liegt im Promillebereich.
- für beide Protokollvarianten ist die Ausführungshäufigkeit von Datentransferbefehlen überproportional hoch im Vergleich zu den übrigen Befehlsarten. Diese Eigenschaft ist wesentlich auf die Umsetzung grundlegender Konzepte für Kommunikationsprotokolle entsprechend dem ISO/OSI Referenzmodell und der formalen Spezifikationsprache SDL zurückzuführen, die ansatzweise erläutert werden sollen: Zu den wesentlichen Aufgaben eines Protokolls (als Teil eines Protokollstapels) zählt die Bearbeitung von Dienstanforderungen bzw. Dienst- und Protokolldateneinheiten über-/unterlagerter Schichten, vgl. [51], die häufige Anforderungen bezüglich der Komposition/Extraktion von Dateneinheiten (Datentransfer) impliziert. Prozessspezifikationen in SDL basieren auf dem Modell des erweiterten endlichen Automaten, alle Systemreaktionen werden durch Signale erzeugt, die zuvor in prozeßinternen Warteschlangen abgespeichert werden. Die Bearbeitung dieser Warteschlangen (Entnahme, Abspeichern von Elementen), Abfrage von Zuständen und Zustandsvariablen usw. erfordern eine hohe Anzahl auszuführender Datentransferinstruktionen.

8.4 Vergleichsmessungen

Zur Verifikation des in Abschn. 8.2 beschriebenen Rechenzeitmodells (Timing) bzw. der Simulationsergebnisse (Rechenzeitbedarf) wurden Vergleichsmessungen mit folgender Systemkonfiguration durchgeführt:

- Verwendung handelsüblicher PC-Hardwarekomponenten, zwei Hauptplatinen (Mainboard) mit Intel 80386 Prozessor (16 MHz Taktfrequenz),
- beide Systeme kommunizieren miteinander über eine serielle Schnittstellenverbindung,
- Meßwerkzeuge: Logikanalyser, vgl. [69], Speicheroszilloskop, vgl. [43], protokollinterne Meßroutinen.

Die Systemzeit eines Personal Computers besitzt eine Auflösung im Sekundenbereich und ist für die Durchführung präziser Kurzzeitmessungen nicht geeignet, weshalb für Zeitmessungen direkt auf den Timer-Baustein Intel 8254 zugegriffen wurde, der einen Zeittakt von ca. 1 μ s besitzt. Meßfehler

durch Bearbeitung von Betriebssystem- bzw. Interrupt-Routinen wurden ausgeschlossen bzw. im Rahmen der Meßgenauigkeit eliminiert.

In verschiedenen Meßreihen wurde die benötigte Rechenzeit für Befehlssequenzen (z.B. 10^6 Addierbefehle), Befehlsmixes (Kombination verschiedener Instruktionen) und die Datenübertragung mit dem Dateitransferprotokoll (entsprechend des Simulationslaufs) ermittelt.

Die Meßergebnisse für das reale Prozessorsystem und den SPEET Hardware-Emulator differierten in einer Größenordnung von 8 - 20 %, in allen Fällen lag der durch Emulation ermittelte Wert der Rechenzeit unterhalb der realen Ausführungszeit. Eine detaillierte Analyse ergab, daß insbesondere bei Prozessorbefehlen, die auf den Systembus zugreifen (typisch Speicherzugriffe), das Zeitverhalten nicht ausreichend nachgebildet wird.

8.5 Weiterentwicklung des SPEET Hardware-Emulators

Bei der Weiterentwicklung des Hardware-Emulators wurden im wesentlichen zwei Ziele verfolgt:

- eine detailliertere (zeitliche) Nachbildung der prozessorinternen Abläufe bei der Befehlsausführung, insbesondere bezüglich der Mikroprozessor-Systembus Schnittstelle, um die Meßgenauigkeit des Emulators zu erhöhen.
- Erprobung der Anpassungsfähigkeit des Emulators im Hinblick auf die Implementierung weiterer 'Mikroprozessor-Emulationen'.

Zur Verbesserung der Meßgenauigkeit wurde eine Restrukturierung des 'Prozessor' Moduls, vgl. Abb. 8.1 b), in Verbindung mit dem 'Scheduler'-Modul vorgenommen; die Dekodiereinheit des Schedulers wurde in das 'Prozessor' Modul verlagert, der Scheduler steuert alle 'Hardware' Komponenten weiterhin durch einen zentralen Taktgeber. Abb. 8.6 zeigt die modifizierte Struktur des 'Prozessor' Moduls, das die typischen Teilkomponenten eines Mikroprozessors detaillierter nachbildet. Bei komplexeren Mikroprozessoren sind ggf. weitere Teilkomponenten hinzuzufügen.

Die Bearbeitung eines Befehls wird jetzt (realitätsnäher) in mehrere Teilschritte zerlegt; Grundlage für die Ermittlung des Rechenzeitbedarfs pro Befehl ist weiterhin die prozessorspezifische Timing-Tabelle, vgl. Abschn. 8.2. Die einzelnen Teilmodule berücksichtigen jedoch zusätzliche Verzögerungen während der Ausführung eines Befehls, die additiv zu den 'idealen'

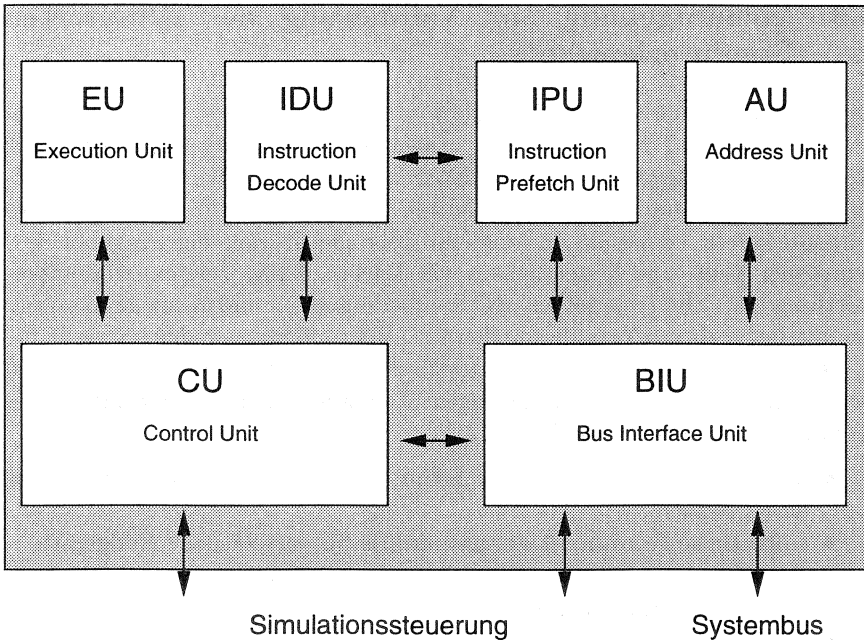


Abbildung 8.6: Aufbau des 'Prozessor' Moduls im SPEET-Simulator

Timing-Werten entstehen, wenn die Voraussetzungen, vgl. Abschn. 8.2, unter denen sie ermittelt wurden, nicht erfüllt sind.

Der SPEET Hardware-Emulator wurde weiterhin um einen Funktionssatz zur Emulation eines Intel 80186 Prozessors erweitert, einem Mikroprozessor, der sich besonders für den Aufbau integrierter Prozessorsysteme (Embedded Applications) eignet, vgl. [49]. Die Wahl des Intel 80186 steht in direktem Zusammenhang mit der Entwicklung einer Hardwareplattform, die für weitere Vergleichsmessungen bzgl. der Emulatorgenauigkeit verwendet wurde und im nächsten Abschnitt beschrieben wird.

8.6 Meßumgebung zur Validierung des erweiterten Emulatorkonzepts

Die Verfeinerungen des Emulatorkonzepts wurden anhand simulativer Untersuchungen und Messungen am realen System für ein standardisiertes Kommunikationsprotokoll eines existierenden Kommunikationsnetzes verifiziert: das diensteintegrierende, digitale Fernmeldenetz ISDN ist zwar als Fernsprechnet konzipiert, ermöglicht aber gleichzeitig das Angebot verschiedener Text- und Datendienste wie Telefax, BTX, Teletex, Datenübermittlung usw. Die technischen Rahmenbedingungen und Kommunikationsprotokolle für dieses Netz sind in den ITU-T Empfehlungen der I-Serie (I.110 ff.) spezifiziert, nationale oder europäische Standards beschreiben zusätzliche länderspezifische Realisierungen. Die wesentlichen Merkmale dieses Kommunikationsnetzes sind:

- international standardisierte Schnittstellen zum Teilnehmer und anderen Netzen,
- einheitliche Kommunikationssteckdose für verschiedene Endgeräte,
- Erreichbarkeit aller im ISDN angebotenen Dienste über eine einheitliche Rufnummernbasis,
- digitale Ende-zu-Ende Verbindungen,
- standardisierte Übertragungsraten von N·64 kbits/s.

Die Untersuchungen wurden bezüglich des Signalisierungsprotokolls LAPD, vgl. [54], dem Sicherungsschichtprotokoll an der Teilnehmer-Netz Schnittstelle, durchgeführt. Abb. 8.7 zeigt das ISDN-Referenzmodell und die Anordnung der Signalisierungsprotokolle für den ISDN-Basisanschluß.

Das LAPD-Protokoll wurde aus dem HDLC³-Protokoll abgeleitet und stellt im wesentlichen die folgenden Funktionen zur Verfügung:

- Einrichtung, Betrieb und Auslösung gesicherter Schicht 2 Verbindungen (zwischen Teilnehmerendgerät und Vermittlungsstelle),
- Bitorientierte und transparente Übertragung von Schicht 3 Daten in Schicht 2 Rahmen,
- Sicherung der zu übertragenden Daten durch Blockprüfung (Frame Check Sequence),
- Sequenzkontrolle der Schicht 2 Rahmen durch Numerierung,
- Flußkontrolle durch Fenstermechanismus.

³High-Level Data Link Control

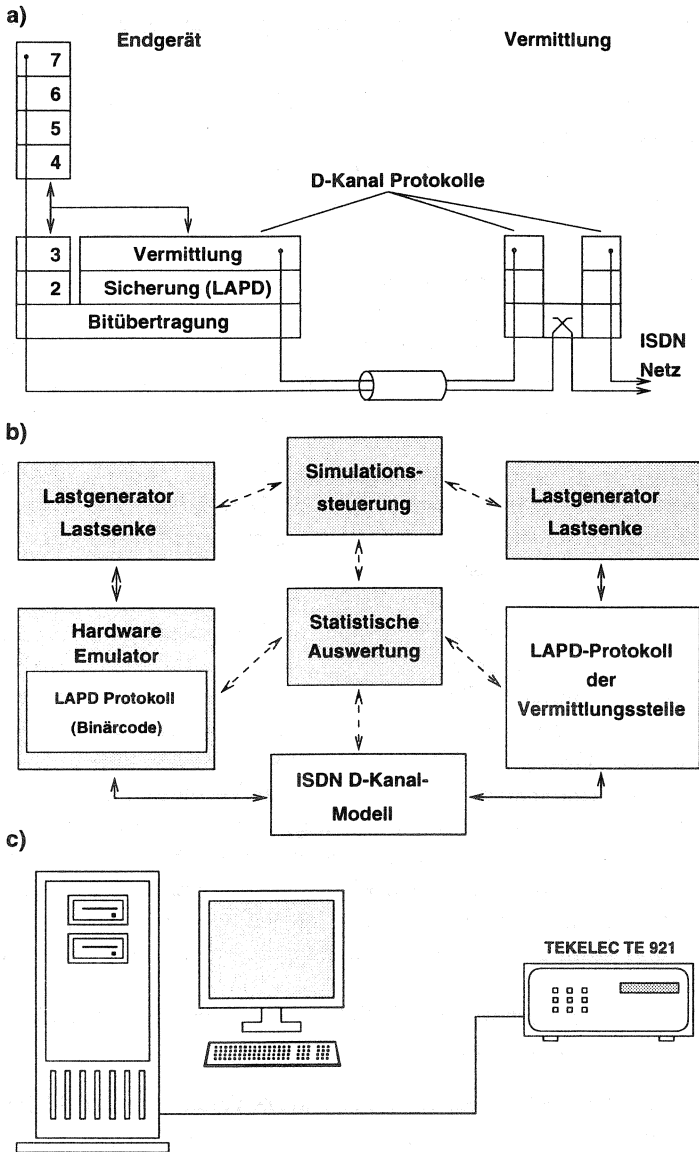


Abbildung 8.7: a) ISDN-Referenzmodell, b) Simulationsumgebung, c) reale Meßumgebung

Zur Durchführung von Simulationen/Messungen standen zwei Versionen des LAPD-Protokolls zur Verfügung:

- eine in SDL formal spezifizierte Protokollversion, die im Rahmen von [35] erstellt wurde,
- eine in der Programmiersprache C kodierte Version der Fa. Siemens für den ISDN-Primärratenanschluß, die funktional für den Basisanschluß angepaßt wurde.

Beide Protokolle beinhalten gleichermaßen nur die LAPD-Funktionen, die bei einem realen System nicht von Hardwarebausteinen (s.u.) bearbeitet werden. Durch vergleichende Messungen mit den unterschiedlich kodierten Protokollen soll ein Eindruck bezüglich der Effizienz/Protokollperformance bei verschiedenen Entwicklungsmethoden, vgl. Abschn. 7.1, vermittelt werden.

Abb. 8.7 b) zeigt das Simulationsszenario unter Einsatz des SPEET-Simulators in Übereinstimmung mit dem ISDN-Referenzmodell. Die benötigte Funktionalität der Netzschicht wird endgeräte- und netzseitig durch den SPEET-Lastgenerator substituiert, das Kanalmodell bildet die Bitübertragungsschicht und den ISDN-Signalisierungskanal (D-Kanal) nach. Messungen bezüglich der Protokollperformance erfolgen endgeräteseitig durch den SPEET Hardware-Emulator.

Die Gerätekonfiguration für Vergleichsmessungen an einem realen System zeigt Abb. 8.7 c). Alle netzseitigen Komponenten wie Netzabschluß, Protokolle der (Orts-) Vermittlungstelle usw. werden durch einen ISDN-Netzsimulator/Analysator der Fa. Tekelec zur Verfügung gestellt, der ebenfalls Monitorfunktionen für die netzseitigen Protokollabläufe besitzt, vgl. [97]. Das ISDN-Endgerät besteht aus einem PC-System, das mit einer aktiven ISDN-Adapterkarte bestückt ist, die im Rahmen der vorliegenden Arbeit entwickelt worden ist. Sie ist für den Betrieb am ISDN-Basisanschluß ausgelegt und kann zwei S_0 -Schnittstellen unabhängig voneinander betreiben.

Abb. 8.8 zeigt den Aufbau der CN2S0-Adapterkarte im Überblick mit den wesentlichen Komponenten:

- Mikroprozessor Intel 80186, einer CMOS Version des 80186-Prozessors, der mit einer Taktrate von 16 MHz betrieben wird,
- DRAM-Controller 82C08, vgl. [50], der den Zugriff auf die Speicherbausteine (RAM) organisiert und einen zyklischen Refresh der DRAM-

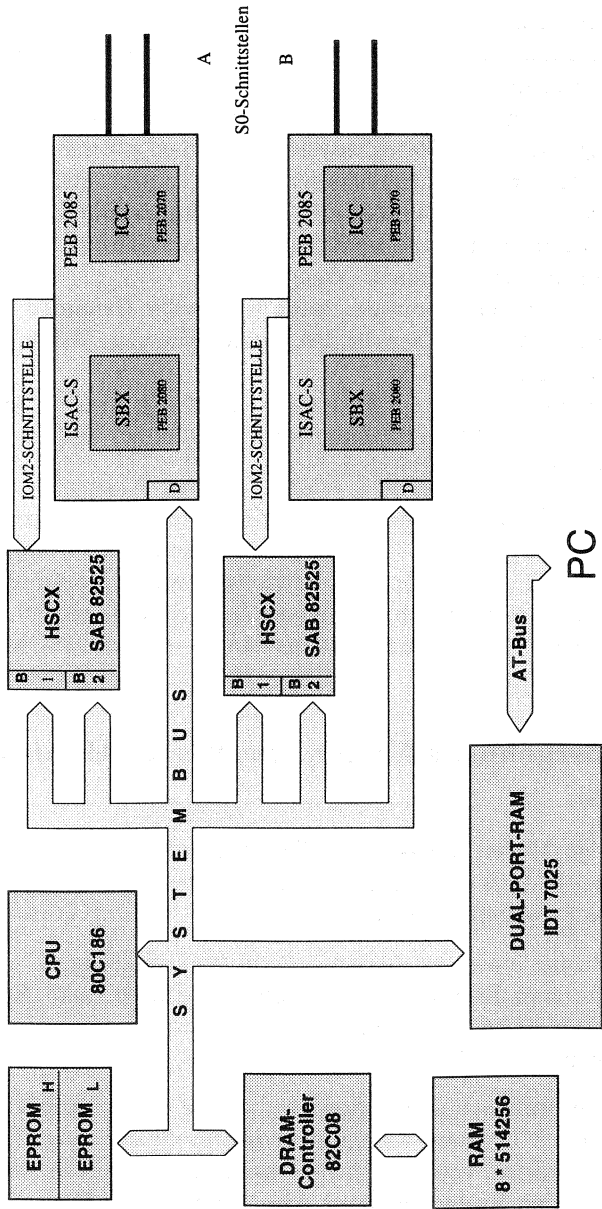


Abbildung 8.8: Struktureller Aufbau der ISDN-Adapterkarte CN2S0

Bausteine durchführt,

- DRAM-Bausteine vom Typ 514256, mit denen ein Speicher von 1 MByte realisiert wird,
- EPROMs vom Typ 27C256 (2 Bausteine = 64 kByte Festspeicher), in denen grundlegende Betriebssystem Routinen wie der System-Urlader, die Initialisierungsfunktionen der Bausteine usw. abgelegt sind,
- Dual-Port RAM (IDT 7025), das die Schnittstelle zwischen PC und ISDN-Adapterkarte realisiert und eine asynchrone Kommunikation zwischen beiden Systemen ermöglicht,
- High-Level Serial Communication Controller Extended (HSCX), ein OSI-Schicht 2 Controller für den Betrieb von zwei Vollduplex-HDLC-Verbindungen,
- ISDN Subscriber Access Controller (ISAC-S), dessen Funktionen die Protokolle der Bitübertragungsschicht (gesamt) und der Sicherungsschicht (teilweise) beinhalten.

Mikroprozessor, DRAM-Controller, EPROMs und DRAM-Bausteine bilden ein eigenständig lauffähiges Mikroprozessorsystem, das über das Dual-Port RAM flexibel mit verschiedenster Betriebssoftware (Kommunikationsprotokolle) versehen werden kann. Der periphere Systemteil, der den Zugang zum ISDN-Netz realisiert, setzt sich aus den Bausteinen HSCX und ISAC-S zusammen.

8.7 Meßergebnisse für den erweiterten Hardware-Emulator

Abb. 8.9 zeigt Meßergebnisse des SPEET Hardware-Emulators bezüglich des Rechenzeitbedarfs für beide Protokollversionen (in SDL und C codiert) und folgende Protokollaktionen:

- Einrichtung einer gesicherten Schicht 2 Verbindung,
- Transfer und Empfang von je 250 byte Daten, inkl. Empfang bzw. Versenden einer Quittung,
- Auslösung einer Schicht 2 Verbindung.

Die Meßwerte umfassen summatorisch die benötigten Rechenzeiten für alle Einzelaktionen (Zustandsübergänge) innerhalb der aufgeführten Protokollaktionen. Anhand von Vergleichsmessungen mit dem in Abschn. 8.6 beschriebenen realen System wurden Abweichungen im Bereich von 6 bis 8

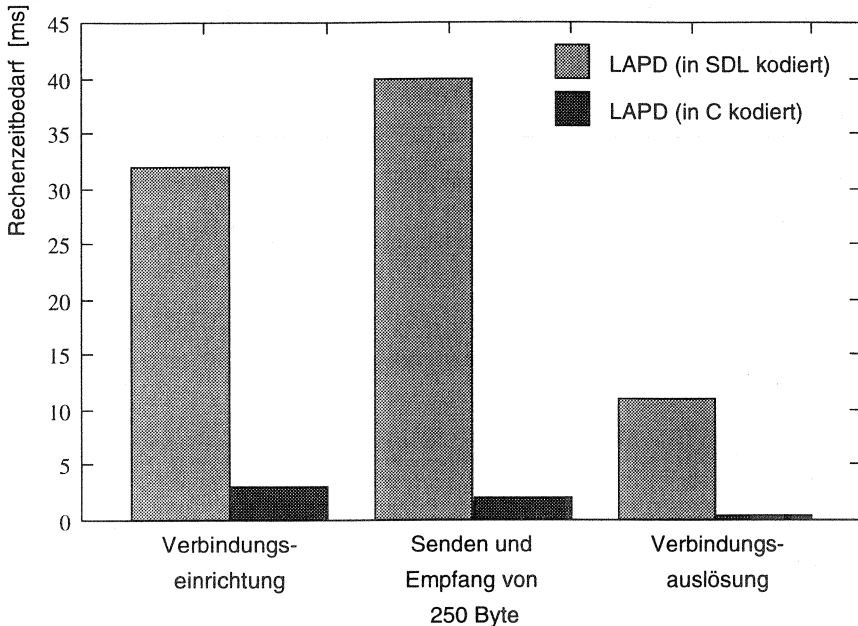


Abbildung 8.9: Rechenzeitbedarf für verschiedene Aktionen des LAPD-Protokolls

% ermittelt, die Meßwerte des Emulators lagen in allen Fällen unterhalb der gemessenen Werte für das reale System. Es zeigt sich, daß der SPEET Hardware-Emulator durch die konzeptionellen Verfeinerungen bzgl. der Ermittlung des Rechenzeitbedarfs deutlich verbessert wurde und eine für simulative Untersuchungen hohe Übereinstimmung im Vergleich zu Messungen am realen System erzielt wird.

Der Vergleich beider LAPD-Protokolle weist einen deutlich höheren Rechenzeitbedarf für das formal spezifizierte Protokoll auf. Analoge Ergebnisse zeigen sich für den Speicherbedarf beider Protokolle, der exakt durch den Hardware-Emulator ermittelt werden kann, vgl. Abb. 8.10: Der Speicher-

bedarf des formal spezifizierten Protokolls für das Codesegment (Größe des auszuführenden Protokolls) und das Heap-Segment (dyn. allozierter Speicher) ist ebenfalls deutlich höher, geringere Speichieranforderungen ergeben sich nur für das Datensegment (statische Daten), was die Gesamtrendenz nicht grundlegend verändert.

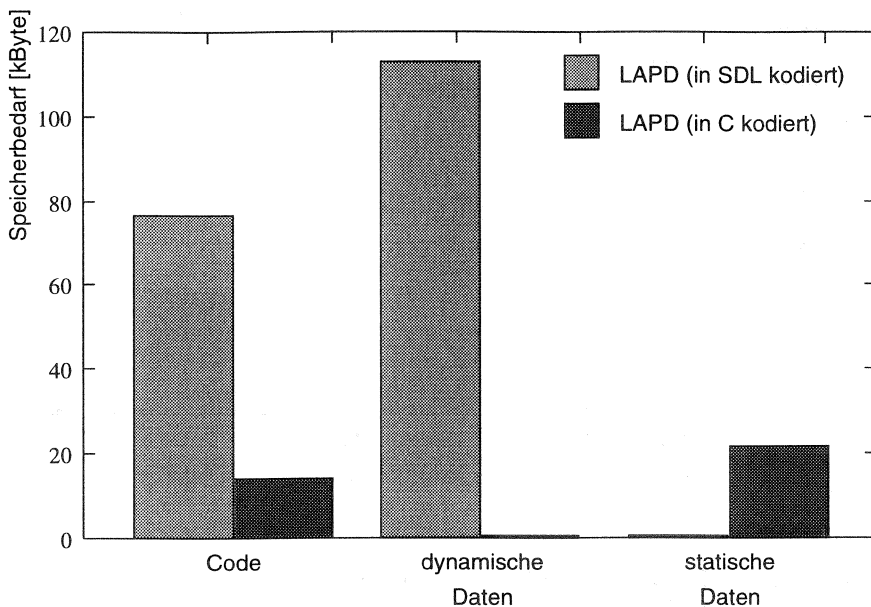


Abbildung 8.10: Speicherbedarf des LAPD-Protokolls für eine in SDL bzw. C kodierte Version

Diese Unterschiede sind u.a. dadurch zu begründen, daß

- die Beschreibungssprache SDL eine deutlich höhere Komplexität gegenüber der Programmiersprache C aufweist und dem Systementwickler ein hohes Maß an Freiheitsgraden (z.B. indirekte Adressierung von Prozessen) verfügbar macht. Die Umsetzung komplexer SDL-Sprachkonstrukte erfordert einen hohen Implementierungsaufwand.
- die C-Version des LAPD-Protokolls optimal auf das 80186 Mikroprozessorsystem angepaßt ist, währenddessen das formal spezifizierte

Protokoll ohne weitere Modifikationen auf verschiedenen Hardwareplattformen ablauffähig ist.

- der Codegenerator des SDT-Werkzeugs noch einige Schwächen hinsichtlich der effizienten Umsetzung von SDL-Sprachkonstrukten aufweist, z.B. bei Protokolltimern, vgl. Abschn. 8.3.
- Designrichtlinien zur Erstellung effizienter SDL-Spezifikationen bisher nicht verfügbar sind.

Der höhere Ressourcenbedarf eines formal spezifizierten Protokolls im Vergleich zu einer hardwarenahen Implementierung ist bei der Systementwicklung zu berücksichtigen bzw. kann teilweise durch gezielte Untersuchungen, z.B. mit dem SPEET-Simulator, und entsprechende Modifikation des Entwicklungswerkzeugs verringert werden. Die angesprochenen Nachteile kompensieren jedoch (subjektiv) keinesfalls die Vorteile des Einsatzes formaler Spezifikationssprachen, vgl. Abschn. 7.1.

Eine gemittelte Befehlsstatistik für die oben aufgeführten Protokollaktionen zeigt Abb. 8.11, wobei die drei Einzelstatistiken keine relevanten Abweichungen zur dargestellten Gesamtstatistik aufweisen. In Analogie zur Befehlsstatistik für das Dateitransferprotokoll (+ Protokollvariante mit Datenkompression) weisen beide Protokollversionen eine überproportionale Ausführungshäufigkeit von Datentransferbefehlen auf.

Dabei sind folgende Randbedingungen ebenfalls von Bedeutung:

- das Profil der Befehlsstatistik ist gleichartig für Verbindungseinrichtung, Betrieb der Verbindung, Verbindungsauslösung.
- beide LAPD-Protokollvarianten weisen eine sehr ähnliche Befehlsstatistik auf, obwohl sie mit verschiedenen Sprachen, Werkzeugen und von unterschiedlichen Entwicklern realisiert wurden.

Eine weitere, nicht bildlich dargestellte Untersuchung zeigt, daß bei beiden LAPD-Protokollen nur ca. 25-45 % des gesamten Befehlssatzes genutzt werden; betrachtet man nur die Befehle mit einer Auftrittshäufigkeit größer 1 %, so liegt der Anteil zwischen 15 und 20 %. Die Ergebnisse sind nicht repräsentativ genug, um daraus eine allgemeingültige Aussage ableiten zu können, sie bieten einen Ansatzpunkt für umfangreiche Untersuchungen, die über den Rahmen der vorliegenden Arbeit hinausgehen. Sollte sich eine ausgeprägte Befehlsstatistik mit hohem Anteil an Datentransferbefehlen und eine begrenzte Nutzung eines CISC-Prozessorbefehlssatzes durch verfügbare Entwicklungswerkzeuge als typisch für Kommunikationsprotokolle nachwei-

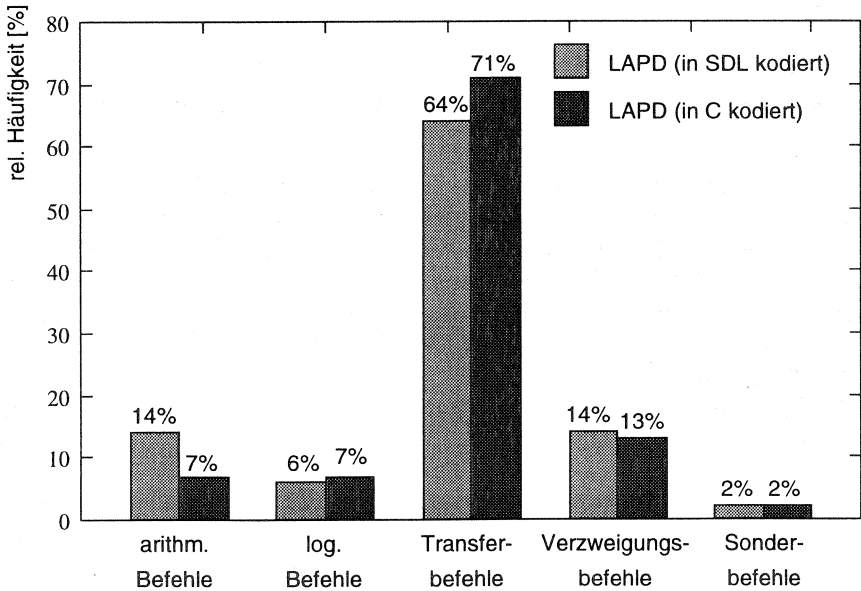


Abbildung 8.11: Gemittelte Befehlsstatistik für das LAPD-Protokoll

sen lassen, so sind optimierte Prozessorsysteme für Kommunikationsanwendungen z.B.

- mit besonders leistungsfähigen Speichersystemen bezüglich des Datentransfers auszustatten, evtl. durch Realisierung einer hierarchischen Speicherarchitektur (Primary Level Cache innerhalb des Prozessors, Second Level Cache zwischen Prozessor und Hauptspeicher).
- anstelle von (heute typischerweise eingesetzten) CISC-Prozessoren spezialisierte Mikroprozessoren mit eingeschränktem Befehlssatz zu verwenden, die auf die Ausführung von Datentransferbefehlen optimiert sind.
- μ -programmierbaren Prozessoren auszustatten, die entsprechend der jeweiligen Anforderung mit einem optimierten Befehlssatz versehen werden können.

Zusammenfassung und Ausblick

Der zunehmende Bedarf an mobiler Kommunikation und die Liberalisierung des Telekommunikationsmonopols im Jahre 1998 werden zu einem verstärkten Einsatz breitbandiger Mobilfunknetze führen. In diesem Zusammenhang gewinnen dezentral organisierte Mobilfunknetze, z.B. für den Aufbau funkbasierter Ortsnetze, zunehmend an Bedeutung, weshalb derartige Systeme derzeit verstärkt entworfen, entwickelt und standardisiert werden. Wesentliche Komponenten von Funknetzen werden heute in Software realisiert, die verwendeten Entwicklungsmethoden und auf dem Markt verfügbaren Werkzeuge weisen noch erhebliche Schwächen im Hinblick auf eine ingenieurmäßige Konstruktion von Software auf. Im Rahmen der vorliegenden Arbeit wurden kanal- und auf Basis des ATM-Verfahrens paketvermittelnde Kommunikationsprotokolle für dezentral organisierte Multi-Hop Funknetze entwickelt und einer detaillierten simulativen Leistungsmessung unterzogen, darüber hinaus eine zugehörige Softwareentwicklungsmethodik zur Realisierung hergeleitet und die daraus resultierenden Werkzeuge für formal spezifizierte Protokolle anhand von Messungen an realen Systemen validiert.

Ausgehend von der konzeptionellen Beschreibung eines dezentral organisierten Funknetzes für innen und außen, das die Bezeichnung Radio Metropolitan Area Network (RMAN) erhielt, realisieren die entwickelten Verfahren und Kommunikationsprotokolle eine dienstgüteabhängige Zuteilung von Übertragungskapazität, den parallelen Betrieb kanalvermittelter und virtueller Verbindungen, sowie wesentliche Erweiterungen für den Einsatz des ATM-Übertragungsmodus in (Multi-Hop) Funknetzen. Einen Schwerpunkt der Protokollentwicklung bildeten außerdem die integrierten Verfahren zur Eliminierung des Einflusses sog. versteckter Stationen in teilvermaschten Funknetzen.

Mit dem RMAN-Simulator entstand im Verlauf der Arbeit ein komplexes und modulares Werkzeug zur Leistungsbewertung, in den detaillierte Modelle aller relevanten RMAN-Komponenten und die oben angeführten Verfahren bzw. Protokolle vollständig integriert sind.

Über die Leistungsfähigkeit dezentral organisierter Multi-Hop Funknetze ist derzeit noch wenig bekannt. Untersuchungen für ein vollvermaschtes RMAN konnten mit einer äquivalenten Untersuchung für das noch in der Standardisierung befindliche HIPERLAN-Netz verglichen werden. Die Simulationsergebnisse zeigen offensichtliche Vorteile der RMAN-Konzeption (Protokolle und Einteilung der verfügbaren Übertragungskapazität) gegenüber dem HIPERLAN-Netz. Dies gilt insbesondere im Bezug auf den erzielbaren Durchsatz und die Stabilität des Gesamtsystems. Weitere Untersuchungen für teilvermaschte RMANs haben die Wirksamkeit der entwickelten Verfahren zum Schutz gegen den Störeinfluß versteckter Stationen aufgezeigt, alle wesentlichen Leistungskenngrößen des Netzes blieben unter Berücksichtigung der für die Teilvermaschung benötigten Ressourcen bzw. Übertragungszeiten auf äquivalentem Niveau zum vollvermaschten RMAN.

Die Softwareentwicklungsmethodik zur Realisierung von RMAN-Systemen wurde ausgehend von einer werkzeuggestützten, formalen Spezifikation von Protokollen in SDL hergeleitet. Im Gegensatz zu bisher bekannten Verfahren werden die formal spezifizierten Protokolle als Hauptbestandteil in eine stochastische Umgebungssimulation zur Leistungsbewertung integriert. Das zugehörige Entwicklungswerkzeug mit der Bezeichnung SDT Performance Evaluation Toolkit (SPEET) ermöglicht bereits in einem frühen Stadium eines Systementwicklungszyklus Aussagen über die Leistungsfähigkeit, Zuverlässigkeit, Parameteroptimierung der später real implementierten Software. In einem zweiten Schritt wurde das SPEET-Werkzeug durch Komponenten zur Prädiktion der erforderlichen Systemverarbeitungsleistung erweitert, um die derzeit häufig divergent verlaufende Entwicklung von Hard- und Software in einer Methode stärker zu integrieren. Dadurch wird es z.B. möglich, die Auslastung von Mikroprozessorsystemen durch den Protokollbetrieb, die Performance formal spezifizierter Protokolle, die benötigten Systemressourcen usw. im Rahmen der Softwareerstellung simulativ zu ermitteln.

Die entstandene Softwareentwicklungsmethodik und die zugehörige Simulationsumgebung sind für die Realisierung von RMAN-Systemen entworfen worden, sie besitzen jedoch eine universelle Konzeption, so daß sie zur Entwicklung beliebiger Kommunikationsinfrastrukturen angewendet werden können. Da die RMAN-Protokolle bisher nicht formal spezifiziert vorliegen und keine RMAN-Komponenten für Vergleichsmessungen verfügbar waren, wurde das SPEET-Entwicklungswerkzeug durch Untersuchungen an existierenden (formal spezifizierten) Kommunikationsprotokollen und parallele

Messungen an realen Systemen validiert. Die ausgewählten Anwendungsfälle aus den Bereichen GSM, Dateitransfer und ISDN zeigen die universelle Einsetzbarkeit des SPEET-Werkzeugs für (Tele-) Kommunikationssysteme. Die Vergleichsmessungen an realen Systemen haben außerdem eine hohe Genauigkeit der simulativ ermittelten Meßgrößen ergeben. Als besonders interessant erwiesen sich in diesem Zusammenhang Untersuchungen bezüglich der Befehlsstruktur bei der Ausführung von Kommunikationsprotokollen; die Ergebnisse zeigten einen überproportional hohen Bedarf an Schiebeoperationen im Vergleich zu anderen Befehlskategorien (arithmetische, logische Operationen usw.), was auf eine weniger gute Eignung heute typischerweise verwendeter CISC-Mikroprozessoren für die Abarbeitung von Kommunikationsprotokollen schließen läßt.

Weitere Arbeiten bieten sich insbesondere im Bezug auf Entwurf und Leistungsbewertung von intelligenten Managementprotokollen für das RMAN an, die hier nicht betrachtet wurden. Ausgehend von weiteren Entwicklungen und Untersuchungen mit der SPEET-Simulationsumgebung stellt die Entwicklung eines für die Ausführung von Kommunikationsprotokollen optimierten Mikroprozessorsystems ein interessantes Themengebiet für nachfolgende wissenschaftliche Arbeiten dar.

QUELLENVERZEICHNIS

- [1] F. Belina, D. Hogrefe, A. Sarma. *SDL with Applications from Protocol Specification*. Hanser Verlag, 1991. ISBN 3-446-16505-3.
- [2] W. Beyers. *Lastmodelle für multimediale Anwendungen und ihre Realisierung in C++*. Diplomarbeit, Lehrstuhl Kommunikationsnetze, RWTH Aachen, 1993.
- [3] F. Billenkamp. *Leistungsuntersuchung struktureller Verbesserungen eines Radio LANs durch Simulation*. Diplomarbeit, Lehrstuhl für Kommunikationsnetze, RWTH Aachen, 1995.
- [4] H. Bischl, W. Schäfer, E. Lutz. *Modell für die Berechnung der Paketfehlerrate unter Berücksichtigung der Antennencharakteristik*. Technischer Bericht, Deutsche Forschungsanstalt für Luft- und Raumfahrt (DLR) e.V., 1991.
- [5] S. Böhmer. *Direkte Benutzer-zu-Benutzer Kommunikation auf vernetzten Arbeitsplatzrechnern: Betriebssystemdienste und Benutzungsschnittstelle*. Diplomarbeit, Lehrstuhl Betriebssysteme, RWTH Aachen, 1990.
- [6] S. Böhmer. *Direkte Benutzer-zu-Benutzer Kommunikation auf vernetzten Arbeitsplatzrechnern*. Ausgewählte Beiträge der Fakultät Informatik, Universität Dresden, 1991.
- [7] S. Böhmer. *Betriebliche und überbetriebliche Vernetzung von Arbeitsplätzen: Ergebnisse einer Rahmenstudie über Techniken und Aufwand der Vernetzung*. pp. C537.03–19, Hamburg, Online '93, Symposium V-4, 1993.
- [8] S. Böhmer. *Performance Analysis of the GSM Signalling Protocol LAPD_m*. In *Proceedings of 44th IEEE Vehicular Technology Conference*, Stockholm, 1994.
- [9] S. Böhmer, R. Klafka. *A New Approach to Performance Evaluation of Formally Specified Protocols*. In *IFIP VII Conference on Formal*

- Description Techniques*. Chapman and Hall, London, 1994. ISBN 0 412644509.
- [10] S. Böhmer, B. Walke. *RMAN Interworking Service and Protocol (RISP)*. Internal Paper, Communication Networks, Aachen University of Technology, 1994.
- [11] S. Böhmer, B. Walke. *Verfahren, Protokolle und Einrichtungen für den Betrieb kanal- und paketvermittelnder Mobilfunknetze mit teilweiser oder vollständiger Vermaschung der Stationen, insbesondere ATM-basierter Zugangsnetze*. Patentanmeldung Aktenzeichen 19536379.5-35, 1995.
- [12] V. Brass. *Ein Quittierungsverfahren für Multi-Hop Paketzuknetze*. Dissertation, FernUniversität Hagen, Hochschulschriften Band 22, 1991.
- [13] V. Brass, B. Walke. *Information distribution for network management in a multi-hop packet-radio network*. In *39th IEEE Vehicular Technology Conference*, pp. 793–798, San Francisco, 1989.
- [14] P. Bratley, B. L. Fox, L. E. Schrage. *A Guide to Simulation*. Springer Verlag, New York, 1983.
- [15] G. Calhoun. *Digital Cellular Radio*. Artech House Inc., Norwood, 1988.
- [16] C. J. Chang, C.-H. Wu. *Slot Allocation for an Integrated Voice/Data TDMA Mobile Radio System with a Finite Population of Buffered Users*. IEEE Transactions on Vehicular Technology, Vol. 43, No. 1, pp. 21–26, 94.
- [17] K.-C. Chen. *Medium Access Control of Wireless LANs for Mobile Computing*. IEEE Network, No. 9, pp. 50–63, 1994.
- [18] M. De Prycker. *ATM Switching on Demand*. IEEE Network Magazine, Vol. 6, No. 2, pp. 25–28, 1992.
- [19] L. Dittmann, S. B. Jacobsen, K. Moth. *Flow Enforcement Algorithms for ATM-Networks*. IEEE Journal on Selected Areas in Communication, Vol. 9, No. 3, pp. 343–350, 1991.
- [20] A. E. Eckberg. *B-ISDN/ATM Traffic and Congestion Control*. IEEE Network Magazine, Vol. 6, No. 5, pp. 28–37, 1992.

- [21] M. Erlinghagen. *Simulation eines lokalen Funknetzes bezüglich des Kanalbündels unter besonderer Berücksichtigung der Kanalökonomie*. In N. Gerner, O. Spaniol, editors, *Kommunikation in verteilten Systemen*, Heidelberg, GI/NTG-Fachtagung Aachen, Springer Verlag, 1987.
- [22] ETSI. *Digital European Cordless Telecommunications (DECT), Reference Document*. European Telecommunication Standards Institute, Sophia Antipolis, France, 1991.
- [23] ETSI. *Radio Equipment and Systems, Digital European Cordless Telecommunications (DECT), Draft prETS 300 175 (Part 1-6)*. European Telecommunication Standards Institute, Sophia Antipolis, France, 1991.
- [24] ETSI. *GSM Recommendation 04.06, MS-BSS Interface, Data Link Layer Specification*. European Telecommunication Standards Institute, Sophia Antipolis, France, 1993.
- [25] ETSI. *GSM Recommendation 04.08, Mobile Radio Interface, Layer 3 Specification*. European Telecommunication Standards Institute, Sophia Antipolis, France, 1993.
- [26] ETSI. *GSM Recommendation 05.02, Multiplexing and Multiple Access on the Radio Path*. European Telecommunication Standards Institute, Sophia Antipolis, France, 1993.
- [27] ETSI. *GSM Recommendation 05.03, Channel Coding*. European Telecommunication Standards Institute, Sophia Antipolis, France, 1993.
- [28] ETSI. *HIPERLAN Services and Facilities, ETR-069*. European Telecommunication Standards Institute, Sophia Antipolis, France, 1993.
- [29] ETSI. *HIPERLAN System Definition, ETR-133*. European Telecommunication Standards Institute, Sophia Antipolis, France, 1994.
- [30] ETSI. *High Performance Radio Local Area Network (HIPERLAN), Functional Specification*. European Telecommunication Standards Institute, Sophia Antipolis, France, 1995.
- [31] ETSI. *Wireless Base Station, Draft prETS 300 175-10*. European Telecommunication Standards Institute, Sophia Antipolis, France, 1995.

- [32] O. Færgemand, A. Olsen. *Systems Engineering using SDL-92*. Korrektorexemplar des Buches, 1993.
- [33] O. Færgemand, A. Olsen. *Introduction to SDL-92*. Computer Networks and ISDN Systems, No. 26, pp. 1143–1167, 1994.
- [34] O. Færgemand, A. Sarma, editors. *Proceedings of the Sixth SDL Forum: SDL '93 Using Objects*, Darmstadt, North-Holland, 1993. ISBN 0 444 81486 8.
- [35] C. Falckenberg. *Weiterentwicklung eines Hardwareemulators zur Leistungsbewertung formal spezifizierter Telekommunikationsprotokolle*. Diplomarbeit, Lehrstuhl Kommunikationsnetze, RWTH Aachen, 1995.
- [36] D. J. Goodman, R. A. Valenzuela, B. Gayliard, K.T. Ramamurthi. *Packet Reservation Multiple Access for Local Wireless Communication*. IEEE Transactions on Communications, Vol. 37, No. 8, pp. 885–890, 1988.
- [37] D. J. Goodman, S. X. Wei. *Factors Affecting the Bandwidth Efficiency of Packet Reservation Multiple Access*. In *Proceedings of the 39th IEEE Vehicular Technology Conference*, San Francisco, 1989.
- [38] R. Gotzhein, J. Brederecke, editors. *Formale Beschreibungstechniken für verteilte Systeme*, Universität Kaiserslautern, 5. GI/ITG–Fachgespräch, Juni 1995.
- [39] A. Hall. *Seven Myths of Formal Methods*. IEEE Software, Vol. 4, No. 5, pp. 11–19, 1990.
- [40] R. Händel, M. N. Huber. *Integrated Broadband Networks: An Introduction to ATM-Based Networks*. Addison Wesley, 1991. ISBN 0-201-54444-X.
- [41] T. Hanebrink. *Simulative Leistungsbewertung von Zugriffsprotokollen für die Übertragung von ATM-Zellen in dezentral organisierten Funknetzen*. Diplomarbeit, Lehrstuhl für Kommunikationsnetze, RWTH Aachen, 1994.
- [42] T. Hellmich. *Formale Spezifikation und Leistungsbewertung von Viel-fachzugriffsprotokollen in Mobilfunknetzen*. Dissertation, FernUniversität Hagen, 1993.

- [43] Hewlett Packard. *DIGITIZING OSCILLOSKOP 54200A/D - Operating, Programming and Service Manual*. Colorado Springs, 1985.
- [44] S. Hinzen. *Simulative Leistungsbewertung von Routingverfahren für dezentral organisierte Funknetze*. Diplomarbeit, Lehrstuhl für Kommunikationsnetze, RWTH Aachen, 1994.
- [45] R. Höhn. *Simulation und Leistungsbewertung von Verfahren zur dynamischen Kanalvergabe in dezentral organisierten Funknetzen*. Diplomarbeit, Lehrstuhl für Kommunikationsnetze, RWTH Aachen, 1994.
- [46] IEEE. *Information processing systems - Local area networks - Carrier Sense Multiple Access with Collision Detection (CSMA/CD)*. The Institute of Electrical and Electronics Engineers, New York, 1989.
- [47] IEEE. *Information processing systems - Local area networks - Logical Link Control*. The Institute of Electrical and Electronics Engineers, New York, 1989.
- [48] Intel Corporation. *386 DX Microprocessor - Programmer's Reference Manual*. Santa Clara, CA9501, 1990.
- [49] Intel Corporation. *16-/32-Bit Embedded Processors*. Santa Clara, CA9501, 1991.
- [50] Intel Corporation. *Memory Products*. Santa Clara, CA9501, 1991.
- [51] ISO. *Open Systems Interconnection - Basic Reference Model*. ISO 7498. International Standards Organization, Geneva, 1984.
- [52] ITU-T. *Recommendation Z.100: CCITT Specification and Description Language (SDL)*. International Telecommunication Union, Geneva, 1988.
- [53] ITU-T. *Recommendation Z.110: Criteria for the use and applicability of Formal Description Techniques*. International Telecommunication Union, Geneva, 1988.
- [54] ITU-T. *Digital Subscriber Signalling System No. 1, Data Link Layer*. International Telecommunication Union, Geneva, 1989.
- [55] ITU-T. *Recommendation I.121: Broadband Aspects of ISDN*. International Telecommunication Union, Geneva, 1989.

- [56] ITU-T. *Recommendation G.804: ATM Cell Mapping into Plesiochronous Digital Hierarchy (PDH)*. International Telecommunication Union, Geneva, 1993.
- [57] ITU-T. *Recommendation I.150: B-ISDN asynchronous transfer mode functional characteristics*. International Telecommunication Union, Geneva, 1993.
- [58] ITU-T. *Recommendation Z.100: CCITT Specification and Description Language (SDL)*. International Telecommunication Union, Geneva, 1993.
- [59] ITU-T. *Recommendation Z.105: SDL Combined with ASN.1 (SDL/ASN.1)*. International Telecommunication Union, Geneva, 1993.
- [60] ITU-T. *Recommendation Z.120: Messages Sequence Chart (MSC)*. International Telecommunication Union, Geneva, 1993.
- [61] B. N. Jain, A. K. Agrawala. *Open Systems Interconnection: Its Architecture and Protocols*. Elsevier Verlag, Amsterdam, 1990. ISBN 0-444-88490-4.
- [62] M. A. Jalloul, S. Nanda, D. J. Goodman. *Packet Reservation Multiple Access over Slow and Fast Fading Channels*. In *Proceedings of the 40th IEEE Vehicular Technology Conference*, pp. 354–359, San Francisco, 1990.
- [63] F. James. *A review of pseudorandom generators*. In *Computer Physics Communications*, pp. 329–344. Elsevier Science Publication, Amsterdam, 1990.
- [64] S. Jangi, L. F. Merakos. *Performance Analysis of Reservation Random Access Protocols for Wireless Access Networks*. IEEE Transactions on Communications, Vol. 42, No. 2/3/4, pp. 1223–1234, 1994.
- [65] M. Junius, S. Böhmer, P. Decker. *Drei Ansätze zur Verbesserung der Dienstgüte im GSM-System*. In *8. Aachener Kolloquium Signaltheorie – Mobile Kommunikationssysteme*, pp. 133–136. VDE-Verlag, Aachen, 1994.
- [66] M. Junius, C. Görg. *Ein Programmpaket zur ereignisorientierten Simulation in C++*. In B. Walke, O. Spaniol, editors, *Messung*,

- Modellierung und Bewertung von Rechen- und Kommunikationssystemen, Kurzberichte und Werkzeugvorstellungen.* Verlag der Augustinus Buchhandlung, Aachen, 1993.
- [67] S. A. Kieschke. *Formale Spezifikation und Bewertung von Signalisierungsprotokollen der GSM-Vermittlungsschicht.* Diplomarbeit, Lehrstuhl für Kommunikationsnetze, RWTH Aachen, 1993.
 - [68] R. Klafka. *Entwurf und Realisierung eines Emulators für Intel Prozessorsysteme zur Leistungsbewertung formal spezifizierter Telekommunikationsprotokolle.* Diplomarbeit, Lehrstuhl Kommunikationsnetze, RWTH Aachen, 1993.
 - [69] Kontron Messtechnik. *Kontron Logik Analysator - Technische Beschreibung.* München, 1987.
 - [70] W. Körner, B. Walke. *Funknetz mit Mobilstationen.* Patentschrift DE 3337 643 C 2, 1991.
 - [71] W. Körner, B. Walke. *Verfahren zum Betrieb eines Funknetzes und Funknetz zur Durchführung des Verfahrens.* Patentschrift DE 3337 648 C 2, 1992.
 - [72] G. Krutt. *Formale Spezifikation, Simulation und Bewertung von GSM-Signalisierungsprotokollen an der Luftschnittstelle U_m .* Diplomarbeit, Lehrstuhl für Kommunikationsnetze, RWTH Aachen, 1993.
 - [73] H.T. Kung, R. Morris. *Credit-Based Flow Control for ATM Networks.* IEEE Network Magazine, Vol. 9, No. 2, pp. 40–48, 1995.
 - [74] R. L. LaMaire, A. Krishna, H. Ahmadi. *Analysis of a Wireless MAC Protocol with Client-Server Traffic and Capture.* IEEE Journal On Selected Areas In Communications, Vol. 12, No. 8, pp. 1299 ff, 1994.
 - [75] W. C. Lee. *Mobile Cellular Telecommunication Systems.* Mc Graw-Hill, New York, 1989. ISBN 0-07-037030.
 - [76] H. D. Lüke. *Signalübertragung.* Springer Verlag, Heidelberg, 1975. ISBN 0-387-07125-3.
 - [77] C. Malter. *Formale Spezifikation, Implementierung und Konformitätsprüfung des E-DSS 1 Signalisierungsprotokolls für den ISDN Primärratenanschluß.* Diplomarbeit, Lehrstuhl für Kommunikationsnetze, RWTH Aachen, 1992.

- [78] G. Marsaglia, A. Zaman. *Toward a Universal Random Number Generator*. In *Statistics & Probability Letter No. 8*, chapter Amsterdam, pp. 35–39. Elsevier Science Publication, 1990.
- [79] M. Nagl. *Softwaretechnik: Methodisches Programmieren im Großen*. Springer Verlag, Heidelberg, 1990.
- [80] S. Nanda. *Stability Evaluation and Design of the PRMA Joint Voice Data System*. IEEE Transactions on Communications, Vol. 42, No. 5, pp. 584–598, 1994.
- [81] S. Nanda, D. J. Goodman, U. Timor. *Performance of PRMA: A Packet Voice Protocol for Cellular Systems*. IEEE Transactions on Vehicular Technology, Vol. 40, No. 3, 1991.
- [82] F. D. Nicklisch. *Analyse der Leistungsfähigkeit von HIPERLAN*. Diplomarbeit, Lehrstuhl für Kommunikationsnetze, RWTH Aachen, 1996.
- [83] K. K. Ramakrishnan, P. Newman. *Integration of Rate and Credit Schemes for ATM Flow Control*. IEEE Network Magazine, Vol. 9, No. 2, pp. 49–56, 1995.
- [84] E. P. Rathgeb. *Modelling and Performance Comparison of Policing Mechanisms for ATM-Networks*. IEEE Journal on Selected Areas in Communication, Vol. 9, No. 3, pp. 325–334, 1991.
- [85] D. Raychanhuri, Wilson N. D. *ATM-Based Transport Architecture for Multiservices Wireless Personal Communication Network*. IEEE Journal on Selected Areas in Communication, Vol. 12, No. 8, pp. 1401–1414, 1992.
- [86] M. Richter. *Ein Rauschgenerator zur Gewinnung von quasi-idealen Zufallszahlen für die stochastische Simulation*. Dissertation, Lehrstuhl Kommunikationsnetze, RWTH Aachen, 1992.
- [87] H. Rudin, W. Bux, editors. *Performance of Computer-Communication Systems*. IFIP WG 7.3/TC6, North-Holland, Zurich, 1984.
- [88] A. Sarma, editor. *Tutorials and SDL Methodology in ETSI*. Sixth SDL Forum, Darmstadt, 1993.

- [89] B. R. Satzberg, [et al]. *ISDN Systems*. Prentice Hall, 1990. ISBN 0-13-505736-1.
- [90] A. Schnabel. *Leistungsuntersuchung eines Zugriffsprotokolls zur Übertragung von ATM-Zellen in dezentral organisierten Funknetzen*. Diplomarbeit, Lehrstuhl für Kommunikationsnetze, RWTH Aachen, 1995.
- [91] F. Schreiber. *The objective empirical distribution function and its error formula*. Archiv für Elektronik und Übertragungstechnik (AEÜ), Vol. 39, 1981.
- [92] F. Schreiber. *Effective Control of Simulation Runs by a New Evaluation Algorithm for Correlated Random Sequences*. Archiv für Elektronik und Übertragungstechnik (AEÜ), Vol. 42, No. 6, 1988.
- [93] B. Sklar. *Digital Communications*. Prentice Hall, 1988. ISBN 0-13-211-939-0 025.
- [94] E. Sourour. *Time Slot Assignment Techniques for TDMA Digital Cellular Systems*. IEEE Transactions on Vehicular Technology, Vol. 43, No. 1, 1994.
- [95] S. M. Sussman. *A Survivable Network of Ground Relays for Tactical Data Communication*. IEEE Transaction on Communication, Vol. COM-28, No. 9, 1980.
- [96] A. S. Tannenbaum. *Computer Networks*. Second Edition. International Editions, Prentice Hall, Englewood Cliffs, 1988.
- [97] TEKELEC AIRTRONIC. *ISDN Simulator / Analyser TE 921 - User Manual*. München, 1989.
- [98] Telelogic. *SDT 3.0 Reference Manual*. Telelogic AB, S-203 12 Malmö, Sweden, 1995.
- [99] The ATM Forum. *User-Network Interface (UNI Specification), Version 3.1*. New York, 1994.
- [100] H. J. Tomaszewski. *Analyse von Paket-Funknetzen mit Leistungssteuerung*. Dissertation, Lehrstuhl Kommunikationsnetze, RWTH Aachen, LIT Verlag, Hagen, 1993. ISBN 3-89473-804-9.

- [101] J. S. Turner. *Managing Bandwidth in ATM Networks with Bursty Traffic*. IEEE Network Magazine, Vol. 6, No. 5, 1992.
- [102] B. Walke. *Über Organisation und Leistungskenngrößen eines dezentral organisierten Funksystems*. In H. Gerner, O. Spaniol, editors, *Kommunikation in verteilten Systemen*. GI/NTG-Fachtagung Aachen, Springer Verlag, 1987.
- [103] B. Walke. *Technik des Mobilfunks (Dienste und Protokolle digitaler Mobilfunknetze)*. In J. Kruse, editor, *Zellularer Mobilfunk*, pp. 17–63, Heidelberg, R. v. Decker's Verlag, 1992. ISBN 3-7685-2191-5.
- [104] B. Walke. *Mobilfunknetze und ihre Protokolle*. Vorlesungsskript, Lehrstuhl für Kommunikationsnetze, RWTH Aachen, 1994.
- [105] B. Walke. *Technische Realisierbarkeit öffentlicher DECT-Anwendungen im Frequenzband 1880–1900 MHz*. Lehrstuhl Kommunikationsnetze, RWTH Aachen, 1995.
- [106] B. Walke, S. Böhmer, M. Bastian, K. H. Hörning. *Betriebliche und überbetriebliche Vernetzung von Arbeitsplätzen*. ISDN-Forschungskommission des Landes NRW, Vieweg Verlag, Wiesbaden, 1994. ISBN 3-528-06582-6.
- [107] B. Walke, R. Briechle. *A Local Cellular Radio Network (LRCN) for digital voice & data transmission at 60 GHz*. In *Cellular Communication International*, pp. 215–225. Online Conferences Ltd., London, 1985.
- [108] B. Walke, O. Spaniol, editors. *Messung, Modellierung und Bewertung von Rechen- und Kommunikationssystemen*. 7. ITG/Gi Fachtagung Aachen, Springer Verlag, 1993.
- [109] T. A. Welch. *A technique for high-performance data compression*. IEEE Computer, Vol. 6, Vol. 39, June 1984.
- [110] K. Wemheuer. *Untersuchung von Parallelisierungsmöglichkeiten eines formal spezifizierten Transportprotokolls*. Diplomarbeit, Lehrstuhl für Kommunikationsnetze, RWTH Aachen, 1992.
- [111] K. Wemheuer, S. Böhmer. *Parallelisierungsmöglichkeiten formal spezifizierter Protokolle*. In R. Grebe, editor, *Parallele Datenverarbeitung mit dem Transputer*. Informatik Aktuell, Springer Verlag, 1993.

- [112] J. M. Wing. *A Specifiers Introduction to Formal Methods*. IEEE Computer, Vol. 23, No. 9, 1990.
- [113] W. Zhu. *Synchronisation mobiler Stationen in Nahbereichsfunknetzen*. Dissertation, Lehrstuhl Kommunikationsnetze, RWTH Aachen, VDI Verlag Reihe 10 Nr. 302, 1994.

NACHWORT

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Lehrstuhl Kommunikationsnetze der RWTH Aachen.

Mein Dank gilt insbesondere Herrn Prof. Dr.-Ing. B. Walke für die Anregungen zum Thema der Dissertation, die sehr gute Betreuung und Förderung der Arbeit, sowie die kritische Durchsicht des Textes.

Herrn Prof. Dr. rer. nat. D. Haupt danke ich für die freundliche Übernahme des Koreferats und die wertvollen Hinweise zum Manuskript.

Bedanken möchte ich mich auch bei allen Diplomanden, Studienarbeitern und studentischen Hilfskräften, die durch ihre Arbeit zum Gelingen beigetragen haben.

Meiner Verlobten Vera Görden möchte ich besonders für ihre liebevolle Unterstützung sowie die Mühe beim Korrekturlesen dieser Arbeit danken, ebenso danke ich meiner Familie, die mir stets mit Rat und Tat zur Seite gestanden hat.

Aachen, September 1996

Stefan Böhmer

AACHENER BEITRÄGE ZUR MOBIL- UND TELEKOMMUNIKATION

ABMT Band 1

Herrmann, C.

Stochastische Modelle für ATM-Konzepte,

1. Auflage 1995, 138 Seiten;

ISBN 3-86073-380-X

ABMT Band 2

Lawniczak, D. R.

Modellierung und Bewertung der Datenverwaltungs-konzepte in UMTS,

1. Auflage 1995, 230 Seiten;

ISBN 3-86073-381-8

ABMT Band 3

Junius, M.

Leistungsbewertung intelligenter Hand-over-Verfahren für zellulare Mobilfunksysteme,

1. Auflage 1995, 208 Seiten;

ISBN 3-86073-382-6

ABMT Band 4

Steffan, H.

Stochastische Modelle für den Funkkanal und deren Anwendung,

1. Auflage 1996, 164 Seiten;

ISBN 3-86073-383-4

ABMT Band 5

Böhmer, S.

Entwurf eines ATM-basierten Funknetzes und Software-Entwurfsmethodik zur Implementierung,

1. Auflage 1996, 172 Seiten;

ISBN 3-86073-384-2

ABMT Band 6

Guntermann, M.

Universelle Benutzermobilität auf der Basis des Intelligenten Netzes - Entwurf,

Bewertung und Implementierung -,

1. Auflage 1996, 164 Seiten;

ISBN 3-86073-385-0

ABMT Band 7

Kleier, S.

Neue Konzepte zur Unterstützung von Mobilität in Telekommunikationsnetzen,

1. Auflage 1996, 204 Seiten;

ISBN 3-86073-386-9

