

**Dissertation zur
Erlangung der Doktorwürde
der Elektrotechnischen Fakultät
der
Universität Gesamthochschule Kassel**



mit dem Thema

„Untersuchung des Einflusses von Steuerungsalgorithmen
auf die Qualität von Multimediadaten im B-ISDN“

vorgelegt von

Dipl.-Ing. Stefan Pitz

Erstgutachter : Prof. Dr.-Ing. E. Sinemus
Zweitgutachter : Prof. Dr.-Ing. H. Früchtling

Erklärung

Hiermit versichere ich, daß ich die vorliegende Dissertation selbständig und ohne unerlaubte Hilfe angefertigt und andere als in der Dissertation angegebene Hilfsmittel nicht benutzt habe. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder unveröffentlichten Schriften entnommen sind, habe ich als solche kenntlich gemacht. Kein Teil dieser Arbeit ist in einem anderen Promotions- oder Habilitationsverfahren verwendet worden.

Kassel

Dipl.-Ing. Stefan Pitz

Danksagung

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Fachgebiet Übertragungstechnik der Universität Gesamthochschule Kassel. Mein Dank gilt allen, die dazu beigetragen haben, daß diese Arbeit durchgeführt und erfolgreich zum Abschluß gebracht werden konnte. Insbesondere möchte ich Herrn Prof. Dr. Erich Sinemus meinen Dank für die Aufnahme in seine Arbeitsgruppe, seine Betreuung und seine wertvollen Anregungen aussprechen.

Mein herzlicher Dank gilt den mittlerweile zu Diplom-Ingenieuren avancierten Studenten Tanja Hartmann, Andreas Rauchhaus, Dieter Weese, Klaus Maßmann, Alexander Bolz, Markus Herzog, Klaus Hommel, Ngoc-Thai Nguyen, Peter Wursthorn, Denis Blum, Martin Wagner und Thorsten Kampe, die durch ihre Studien- und Diplomarbeiten wesentlich zum Gelingen dieser Arbeit beigetragen haben. Ohne ihre Hilfe wäre ein solch umfangreiches Softwareprojekt nicht zu verwirklichen gewesen.

Den Mitarbeitern des Fachgebiets danke ich für ihre Unterstützung bei der Bewältigung der vielen täglichen Probleme im technischen Bereich als auch im Umgang mit Hierarchien und Entscheidungswegen der GhK.

Schließlich möchte ich den Herren Diplom-Ingenieuren Andreas Siegl und Peter Rinder für ihre aufopfernde Hilfe bei der Korrektur dieser Arbeit danken.

Kassel, im Februar 2000

Stefan Pitz

Inhaltsverzeichnis

KAPITEL 1 Einleitung	1
1.1 Motivation	1
1.1.1 Digitale Fernsprechnetze	1
1.1.2 Digitale Datennetzwerke	6
1.1.3 Netzwerkangepaßte Dienste	9
1.2 Problemstellung	10
1.2.1 Diensteintegration in bestehenden Netzwerken	10
1.2.2 Zukünftige Dienste	10
1.2.3 Zusammenfassung	12
1.3 Vorgehensweise	13
KAPITEL 2 Statistik und Verkehrstheorie	15
2.1 Eigenschaften von Zufallsgrößen	15
2.1.1 Zufallsvariable	15
2.1.2 Verteilungsfunktionen	15
2.1.3 Momente	16
2.1.4 Wichtige Verteilungsfunktionen und Momente	17
2.1.5 Momentengenerierende Funktionen	18
2.2 Stochastische Prozesse	19
2.2.1 Kenngrößen stochastischer Prozesse	19
2.2.2 Stationarität und Ergodizität	19
2.3 Bedientheorie	20
2.3.1 Klassifizierung von Bediensystemen	20
2.3.2 Analyse von Bediensystemen	21
2.4 Erzeugung von Zufallszahlen	24

2.4.1	Erzeugung von Zufallszahlen mit beliebigen Verteilungsfunktionen	24
2.4.2	Generatoren für gleichverteilte Zufallszahlen	24
2.5	Stichproben und Hypothesentests	25
2.5.1	Stichprobentheorie	25
2.5.2	χ^2 -Anpassungstest	25
2.6	Ankunftsprozesse realer Datenquellen	26
2.6.1	Quellen mit VBR-Charakteristik	27
2.6.2	Multiplexgewinn bei VBR-Quellen	31
2.7	Zusammenfassung	32
KAPITEL 3 Grundlagen der Daten-übertragung im B-ISDN		33
3.1	Übermittlungsprinzip im B-ISDN	33
3.1.1	Der Asynchrone Transportmodus ATM	33
3.1.2	Multiplex im B-ISDN	34
3.1.3	End-to-End Delay	35
3.2	Datenformat im B-ISDN	37
3.3	Vermittlungstechnik im B-ISDN	38
3.3.1	Verbindung zwischen Endgeräten	39
3.3.2	ATM-Vermittlungsstellen (Switches)	39
3.4	Telekommunikationsstandards	40
3.5	Referenzmodelle für Telekommunikationssysteme	41
3.5.1	Das OSI-7-Schichtenmodell	41
3.5.2	Terminologie des OSI-Modells	42
3.5.3	Das ATM-Referenzmodell	46
3.6	Physical Layer	48
3.6.1	Referenzkonfiguration	48
3.6.2	Physical Medium Sublayer	48
3.7	ATM-Layer	56
3.7.1	Funktionen der Protokollinstanzen im ATM-Layer	57
3.7.2	Service Primitives	61
3.8	AAL-Layer	62
3.8.1	AAL-Typ 0	64
3.8.2	AAL-Typ 1	64
3.8.3	AAL-Typ 2	71
3.8.4	AAL-Typ 3/4	75
3.8.5	AAL-Typ 5	81
3.8.6	SAAL	84

3.9	Signalisierung und Adressierung	90
3.9.1	Aufgaben der Signalisierung	90
3.9.2	Ablauf der Signalisierung	91
3.10	Zusammenfassung	94
KAPITEL 4 Traffic Control und Congestion Control gemäß ITU		95
4.1	Zielsetzung	95
4.2	Connection Admission Control gem. ITU Rec. I.371	96
4.2.1	Traffic Contract	96
4.2.2	ATM Transfer Capability ATC	98
4.3	Referenzkonfiguration	102
4.4	Zusammenfassung	103
KAPITEL 5 Adaptive Connection Admission Control		105
5.1	Motivation	105
5.2	Referenzkonfiguration	106
5.3	Prinzipielles Vorgehen	107
5.4	Berechnungsgrundlagen der Adaptiven CAC	108
5.4.1	Bezeichnungen	108
5.4.2	Berechnungsgrundlagen	108
5.4.3	Ablauf der Adaptiven CAC	110
5.5	Zusammenfassung	112
KAPITEL 6 Simulationssoftware		113
6.1	Simulationsziel	113
6.2	Simulationstechnik	114
6.2.1	Zeitdiskrete ereignisorientierte Simulationen	114
6.2.2	Programmiersprache	116
6.3	Ablaufprinzip der Simulation	117
6.3.1	Initialisierung des Simulationsmodells	117
6.3.2	Durchführung der Simulation	119
6.3.3	Analyse des Simulationsergebnisses	121
6.4	Ermittlung der Bediendauern	121
6.4.1	Schichtenmodell der Simulationssoftware	121
6.4.2	Bedienung von Forderungen in den Hardwareinstanzen	122

6.5	Struktur der Software	125
6.6	Kurzbeschreibung der Module	127
6.6.1	Notation	127
6.6.2	Basismodul	127
6.6.3	Statistik- und Quellenmodul	128
6.6.4	Netzwerkmodul	129
6.6.5	OSI-Modul	131
6.6.6	Hardwaremodul	132
6.6.7	Konfigurationsdateien	133
6.6.8	Ergebnisdateien	133
6.6.9	Visualisierung	133
6.6.10	Ablaufsteuerung	134
6.7	Bedienung und Umfang	136
6.8	Zusammenfassung	136
KAPITEL 7	Ergebnisse	139
7.1	Ablaufdiagramme	140
7.1.1	Netzwerkinitialisierung	140
7.1.2	Verbindungsaufbau	140
7.1.3	Datentransfer	142
7.1.4	Verbindungsabbau	143
7.1.5	Auswertung	144
7.2	Simulationslauf 1	144
7.2.1	Quellenpool	145
7.2.2	Ergebnisse	146
7.3	Simulationslauf 2	151
7.4	Zusammenfassung	153
KAPITEL 8	Zusammenfassung und Ausblick	155
KAPITEL 9	Literaturverzeichnis	157
KAPITEL 10	Abkürzungsverzeichnis	165
KAPITEL 11	Formelzeichen	175
KAPITEL 12	Anhang	177
	Anhang A Datenelemente	178

A.1	Informationsfeld der F1- bzw. F3-OAM-Zelle	178
A.2	Aufbau einer Metasignalisierungszelle	179
A.3	Datenformat des AAL-Typs 2	180
A.4	Datenformat des AAL-Typs 3/4	181
A.5	Datenformat des AAL-Typs 5	182
A.6	Signalisierungsnachrichten	182
Anhang B Service Primitives		187
B.1	Zwischen Physical Layer und ATM-Layer	187
B.2	Zwischen ATM-Layer und ATM-Layermanagement	188
B.3	Zwischen ATM-Layer und AAL-Layer	188
B.4	Zwischen AAL-Benutzer und AAL-1-Layer	189
B.5	Service Primitive des AAL-Typs 1	190
B.6	Service Primitive des AAL-Typs 2	191
B.7	Service Primitive des AAL-Typs 3/4	192
B.8	Service Primitive des AAL-Typs 5	193
B.9	Service Primitive des AAL-Typs S	194
Anhang C Klassenübersichten		195
C.1	Basismodul	195
C.2	Statistikmodul	201
C.3	Netzwerkmodul	205
C.4	OSI-Modul	214
C.5	Hardwaremodul	227
C.6	Konfigurationsdateien	231
C.7	Ergebnisdateien	233
C.8	Visualisierung	235
Anhang D Konfigurationsdateien		237
D.1	Simulationslauf 1	237
D.2	Simulationslauf 2	259
Anhang E Makefiles		260
E.1	Oberstes Makefile	260
Anhang F Auswertung		272
F.1	Aufbau der Ergebnisdateien	272
F.2	Auswertungsskript	280

1.1 Motivation

Die Kommunikation zwischen Menschen beruht zum überwiegenden Teil auf dem Austausch analoger audiovisueller Daten wie beispielsweise der durch Gestik und Mimik unterstützten Sprache. Auch das Fernsprechnet als wichtigstes zur Übertragung dieser Informationen eingesetzte technische System, war ursprünglich analoger Natur. Als Folge des Beweises des Abtasttheorems durch C. Shannon [Sha45] und der Entwicklung der Pulsmodulation durch A.H. Reeves in den Labors der Firma AT&T in Paris wurden in zunehmendem Maße die analogen Übertragungssysteme durch ihre digitalen Äquivalente ersetzt.

1.1.1 Digitale Fernsprechnetze

Die Digitalisierung der Sprachsignale erfolgt unter der Voraussetzung, daß zum Erreichen einer ausreichenden Verständlichkeit die Übertragung der Frequenzanteile zwischen 300 und 3.400 Hz notwendig ist. Davon ausgehend und unter Beachtung des Abtasttheorems wurde international verbindlich eine Abtastrate von 8.000 Hz für Sprachsignale festgelegt. Bei einer Quantisierung der Abtastwerte mit jeweils acht Bit ergibt sich ein Datenaufkommen von 64 kBit/s für einen Fernsprechkanal.

1.1.1.1 Internationale plesiochrone Multiplexhierarchien

Der Zwang zum wirtschaftlichen Betrieb der Fernsprechnetze machte die Mehrfachnutzung der zur Verfügung stehenden Übertragungskanäle notwendig. Als Pendant zu der im analo-

gen Netz verwendeten Frequenzmultiplextechnik FDM¹ wird im digitalen Netz die Zeitmultiplextechnik TDM² eingesetzt. Die im regelmäßigen Abstand von $t_0 = 125\mu\text{s}$ digitalisierten Sprachabstastwerte der einzelnen Teilnehmer werden bei diesem Verfahren in einer festgelegten Reihenfolge gesendet. Durch Zusammenfassung von anzahlmäßig immer mehr Kanälen entstanden so die ersten plesiochronen³ digitalen Fernsprechnetze. Sie bestehen aus hierarchisch angeordneten Vermittlungsstellen mit Multiplexfunktionen, in denen der Übergang zwischen den benachbarten Ebenen erfolgt. Abbildung 1 zeigt einen Ausschnitt aus den unteren Multiplexebenen des europäischen PDH⁴-Netzes:

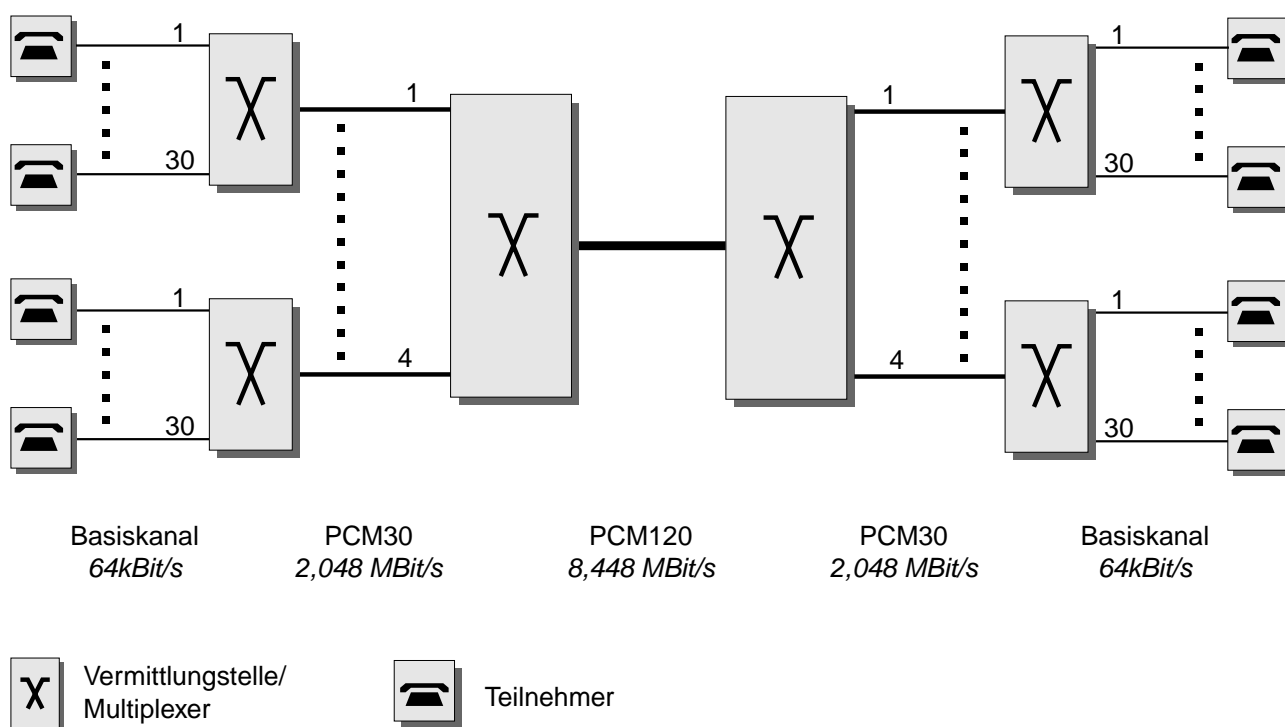


Abbildung 1: Plesiochrones digitales Fernsprechnet (Europa)

Durch das Fehlen eines weltweit verbindlichen Standards entwickelten sich in Europa, Nordamerika und Japan unterschiedliche Multiplexhierarchien, zwischen denen nur wenige Übergänge möglich sind (siehe Abbildung 2 auf Seite 3).

1. Frequency Division Multiplex
 2. Time Division Multiplex
 3. nahezu synchron (alle Netzelemente arbeiten unabhängig voneinander mit eigener Taktversorgung)
 4. Plesiochronous Digital Hierarchy

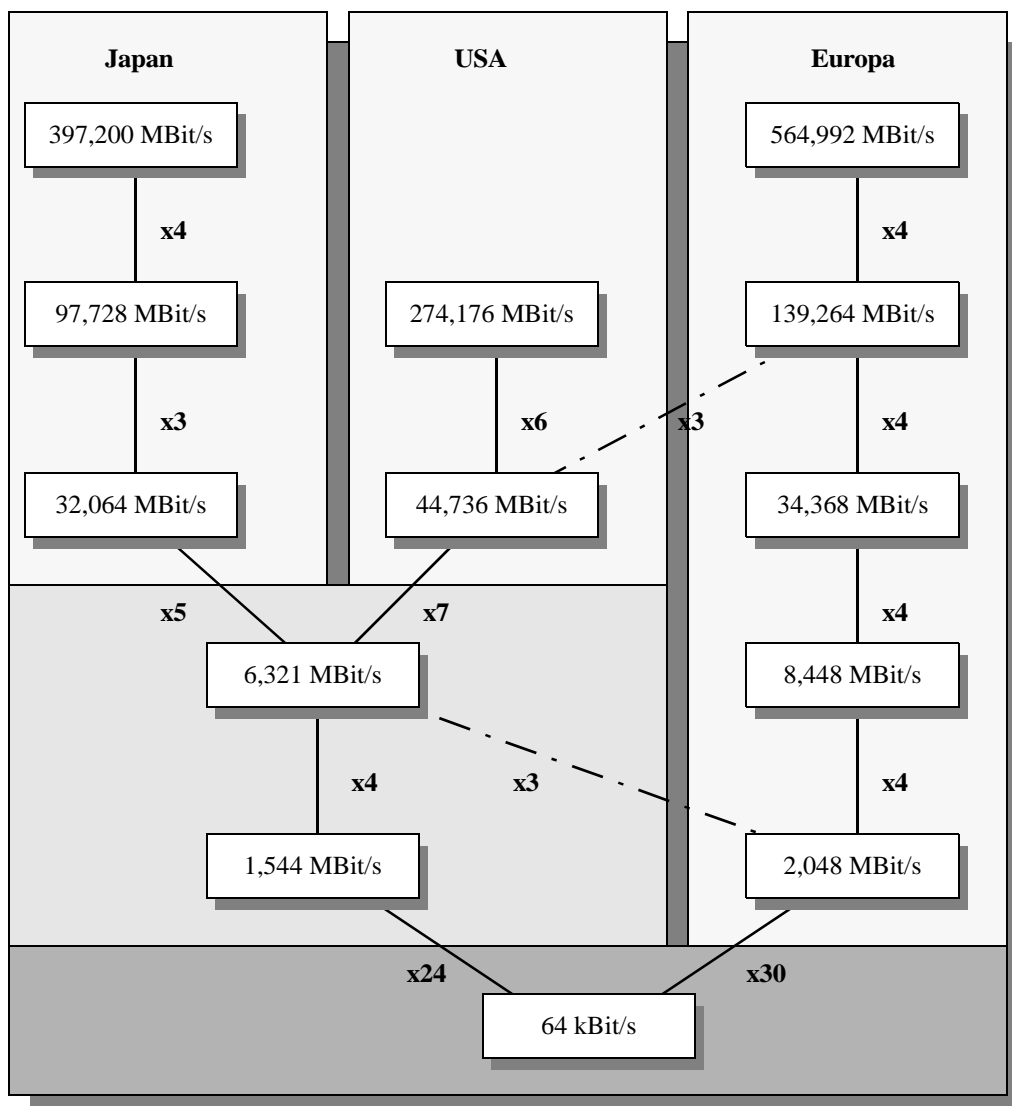


Abbildung 2 : Internationale pliosichrone Multiplexhierarchien

Die wichtigsten Eigenschaften dieser Netzwerke sind in der folgenden Tabelle zusammengestellt:

Verbindungsaufbau	Vor jedem Datenaustausch
Route durch das Netzwerk	Für die gesamte Dauer der Verbindung festgelegt
Ressourcenzuteilung in den Vermittlungsstellen	Exklusiv zu einer Verbindung für die gesamte Dauer des Datentransfers
Verarbeitungsdauer in den Vermittlungsstellen	Konstant
Verzögerung der Nutzdaten	Konstant

Tabelle 1 : Eigenschaften digitaler Fernsprechnetze

Der Tabelle kann die Erhaltung der Synchronität der Abtastwerte als das vorrangige Ziel bei der Vermittlung der Daten innerhalb der Fernsprechnetze entnommen werden. Der hohe Stellenwert dieser *zeitlichen Transparenz* begründet sich durch die harten Anforderungen von Sprachdaten an die Maximalwerte von Verzögerung (*Delay*) und Verzögerungsschwankungen (*Delay Variation*) nach der Übertragung zwischen den Endteilnehmern (*End-to-End*).

1.1.1.2 End-To-End Delay

Die konstante Übertragungsdauer zwischen den Teilnehmern wird zum Ersten durch die Festlegung des Weges für alle Daten vor Beginn des eigentlichen Transfers erreicht. Laufzeitschwankungen, die durch unterschiedliche Wege der Daten durch das Netzwerk verursacht würden, sind daher ausgeschlossen. Zum Zweiten beschränkt sich die Verarbeitung in den Netzknoten auf die bereits während des Verbindungsaufbaus festgelegte Umsetzung der Zeit- und Raumlage, die zu einem konstanten zeitlichen Versatz der Sprachabtastwerte führt, wie es in der folgenden Abbildung für eine Vermittlungsstelle mit vier PCM30¹-Kanälen gezeigt ist.

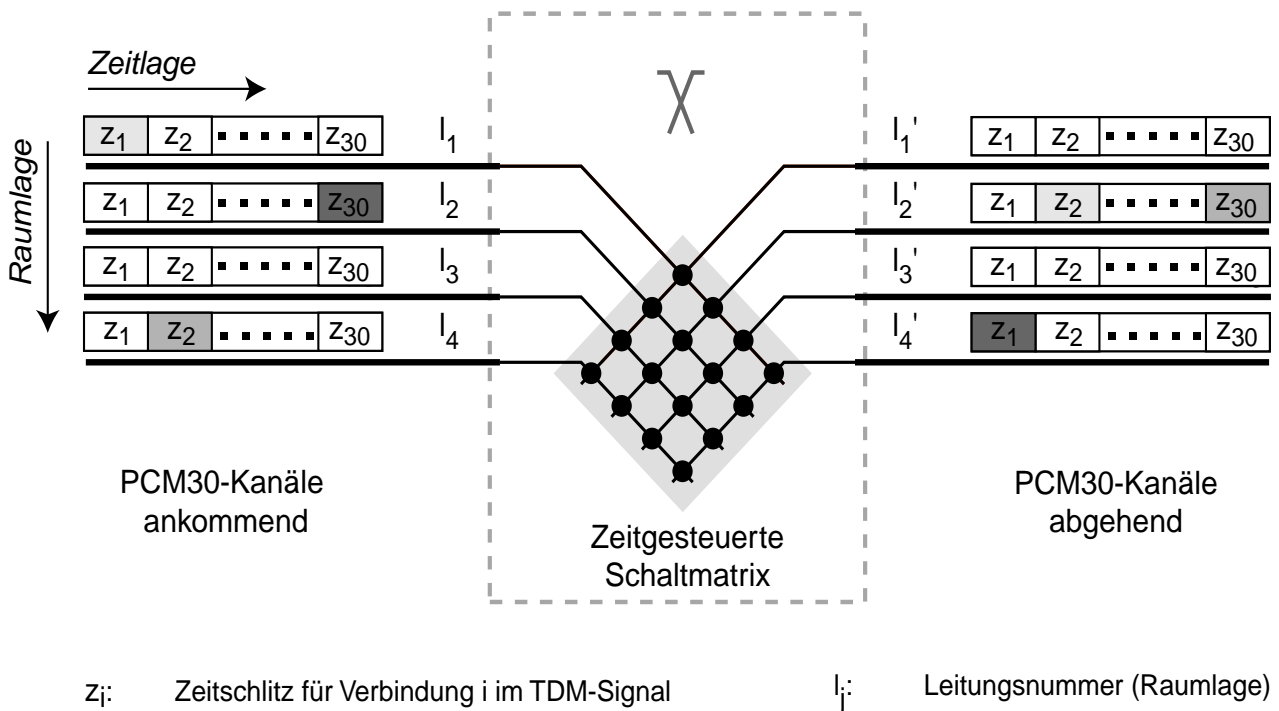


Abbildung 3 : Raum- und Zeitlagenwechsel in einer Fernsprechvermittlungsstelle

1. 1. Europäische Multiplexebene (vgl. Abbildung 2 auf Seite 3)

Eingesetzt wird die *Leitungsvermittlung* z.B. in den Vermittlungsstellen *System 12* oder *EWSD*¹ im Fernsprechnet der Deutschen Telekom, wobei blockierungsfreie Koppelvielfache (Clos'sche Anordnungen) die Schaltmatrix bilden. Da die Vermittlungsstelle für jeden der ankommenden Kanäle über ausreichende Verarbeitungskapazitäten verfügt, ist eine gegenseitige Blockierung der Daten ausgeschlossen, und es kann von einer konstanten Verzögerung für alle Abtastwerte einer Verbindung ausgegangen werden. Die Gesamtdauer der Übertragung zwischen den Teilnehmeranschlüssen ergibt sich demnach als die Summe der konstanten Ausbreitungszeiten auf den Teilstrecken und aller Verarbeitungsdauern in den Vermittlungsstellen. Die Daten erreichen den Empfänger daher mit im Vergleich zum Sendevorgang identischen zeitlichen Abständen.

Ein Nachteil der Leitungsvermittlung ist in der unzureichenden Ausnutzung der Ressourcen in den Vermittlungsstellen zu sehen. Rechenleistung und Pufferspeicher werden den Verbindungen zur alleinigen Nutzung zur Verfügung gestellt und dürfen auch während der Gesprächspausen nicht von anderen Verbindungen genutzt werden.

1.1.1.3 Integration netzfremder Dienste

Für „Nicht-Fernsprech“-Anwendungen wie Telefax oder den Datentransfer über Modem spielt die Realzeitfähigkeit des Fernsprechnetes nur eine untergeordnete Rolle, zudem können sie aufgrund ihrer Verkehrscharakteristik die ihnen während der Dauer der Verbindung exklusiv zur Verfügung stehende Übertragungsrate nicht voll ausschöpfen. Die Abwicklung dieser Dienste über das Fernsprechnet kann daher nur eine Interimslösung darstellen, bis die auf solche Anwendungen abgestimmten, leistungsfähigen Netze in jedem Haushalt verfügbar sind.

Während im Netzzinnern die Digitalisierung bereits abgeschlossen ist, wurden im Teilnehmerbereich noch längere Zeit ausschließlich analoge Endgeräte verwendet. Unter der Bezeichnung *ISDN*² werden zur Zeit digitale Endgeräte eingeführt. Die Neuerungen umfassen jedoch vorrangig die Bereitstellung neuer Dienste für den Teilnehmer, die Struktur des Fernsprechnetes bleibt davon unberührt; es handelt sich nach wie vor um ein leitungsvermittelndes, auf Sprachübertragung optimiertes Spezialnetz.

1. Elektronisches Wählsystem Digital
2. Integrated Services Digital Network

1.1.2 Digitale Datennetzwerke

Im Gegensatz zu den Fernsprechnetzen erfolgt der Datenaustausch zwischen Rechnern von Beginn an ausschließlich digital. Während im Fernsprechnet die *zeitliche Transparenz* das wichtigste Ziel ist und die fehlerfreie Übertragung der Abtastwerte im Hintergrund steht¹, ist die wichtigste Forderung an die Netze bei der Übertragung digitaler textueller Daten die *syntaktische Transparenz*, d.h. die fehlerfreie Übermittlung der Informationen. Klassische Datennetze weisen daher Eigenschaften auf, die sich von denen der Fernsprechnetze erheblich unterscheiden (vgl. Tabelle 1 auf Seite 3) :

Verbindungsaufbau	Nicht zwingend erforderlich
Route durch das Netzwerk	Kann während des Datentransfers variieren
Ressourcenzuteilung in den Netzknoten	Aufteilung unter allen Datenströmen
Verarbeitungsdauer in den Netzknoten	lastabhängig
Verzögerung der Nutzdaten	lastabhängig

Tabelle 2 : Eigenschaften digitaler Datennetze

Innerhalb der Datennetze werden sowohl das Übertragungsmedium als auch die Kapazitäten der Router² zwischen allen Teilnehmern aufgeteilt, wie es in Abbildung 4 dargestellt ist.

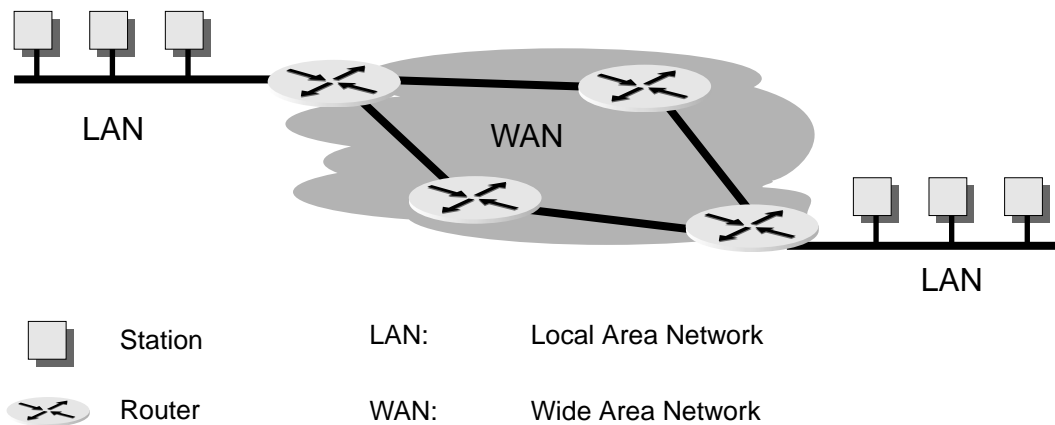


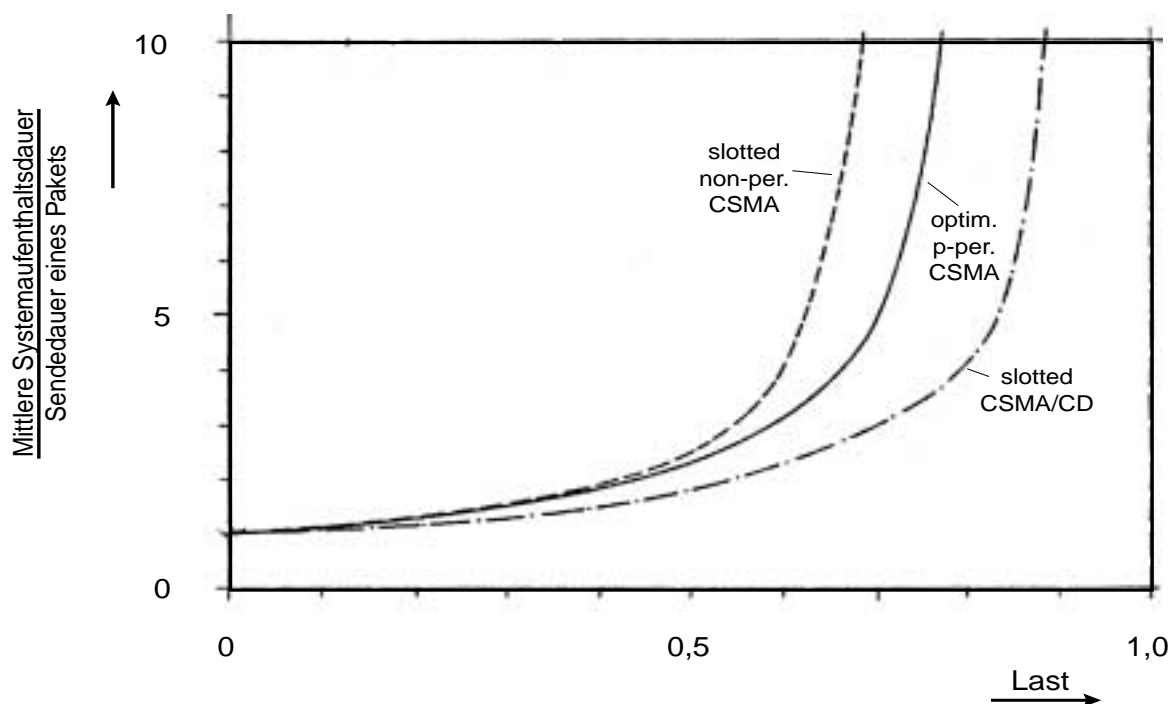
Abbildung 4 : Gemeinsame Nutzung der Ressourcen in Datennetzwerken

1. selbstverständlich nur bis zu einem gewissem Grad
 2. Vermittlungsstellen

Im LAN¹ wird der Zugriff auf das Übertragungsmedium entweder über zufallsgesteuerte Verfahren wie CSMA/CD² oder Verfahren, bei denen eine Sendeberechtigung zwischen den Stationen weitergereicht wird, geregelt. Bei Verwendung von CSMA/CD prüft die sendende Station, ob das Medium nicht bereits durch einen anderen Sendevorgang belegt ist. Ist dies der Fall, unternimmt die Station nach einer zufällig ermittelten Wartezeit einen erneuten Sendeversuch. Wird eine Sendeberechtigung (Token) verwendet (z.B. Token Ring-LAN³), darf nur diejenige Station senden, die momentan über diese Sendeberechtigung verfügt; je nach Verfahren muß sie die Sendeberechtigung nach einer bestimmten Zeit oder erst nach Abwicklung all ihrer Sendewünsche weitergeben.

1.1.2.1 End-to-End Delay

Die Übertragungsdauern in paketvermittelten Netzen sind daher im Gegensatz zum Fernsprechnet lastabhängig und nicht konstant. Bei gleichzeitigen Sendeversuchen einer großen Anzahl der angeschlossenen Stationen kommt es zu langen Aufenthaltsdauern⁴ der Datenpakete im System, wie die folgende Abbildung am Beispiel unterschiedlicher Varianten des CSMA-Verfahrens zeigt:



1. Local Area Network

2. Carrier Sense Multiple Access / Collision Detection gem. IEEE 802.3 [IEEE802_3]

3. Token Ring gem. IEEE802.4 [IEEE802_4]

4. Wartedauer bis zur erfolgreichen Übertragung und Übertragungsdauer des Paketes

Abbildung 5 : Mittlere Systemaufenthaltsdauer bei CSMA/CD [Kle75]

Die Weitervermittlung der Datenpakete nach dem Verlassen des LAN über die Router im WAN (siehe Abbildung 6 auf Seite 8) verändert zusätzlich deren zeitlichen Abstände. In den Protokollen der klassischen Datennetze wird teilweise auf den Aufbau von Verbindungen verzichtet (z.B. UDP¹). Die Pakete müssen sich ihren Weg eigenständig durch das Netz suchen, sie benutzen daher unter Umständen verschiedene Wege und erreichen den Empfänger mit unterschiedlichen Verzögerungen. Erschwerend wirkt sich aus, daß in den Routern, im Gegensatz zu den Fernsprechvermittlungsstellen, keine Kapazitäten exklusiv reserviert werden. Ein ankommendes Datenpaket muß daher abwarten, bis der Vermittlungsrechner genügend Ressourcen zur Verfügung hat, um es zu verarbeiten. Diese Wartedauer hängt von der momentanen Last ab, die der Router zu bewältigen hat, und variiert zudem von Paket zu Paket.

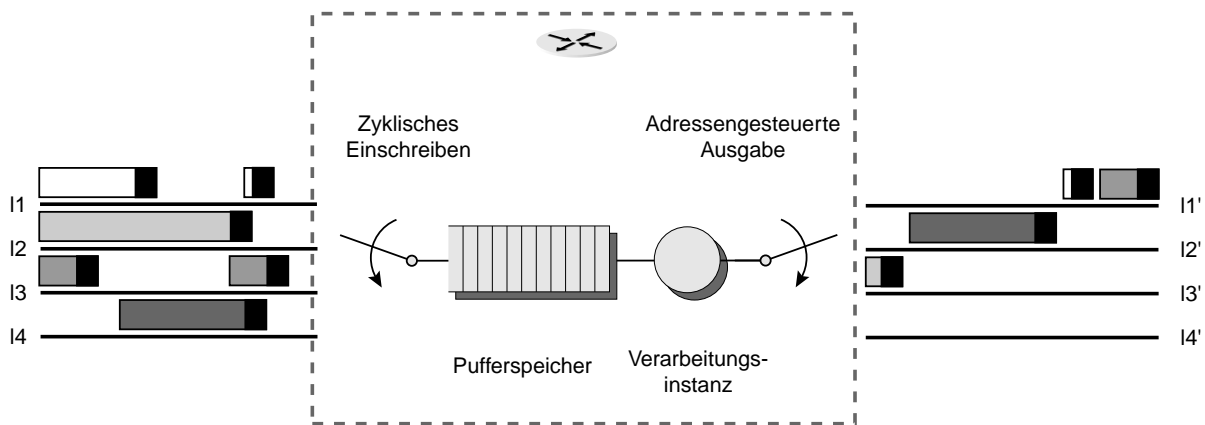


Abbildung 6 : Speichervermittlung in einem Router eines paketvermittelnden Netzes

Um die Datenverluste bzw. -verfälschungen zu beheben, die durch Bitfehler auf dem Übertragungsmedium oder durch Pufferüberläufe in den Netzknoten verursacht werden, müssen die Pakete in den Routern zeitaufwendige Bearbeitungsroutinen durchlaufen. Die Rekonstruktion der verfälschten Daten erfolgt dann entweder durch Fehlerkorrektur oder durch Wiederholungsanforderungen an die sendende Station (vgl. [IEEE802_2]).

Die gesamte Übertragungsdauer in den paketvermittelten Datennetzen setzt sich demnach aus mehreren nichtkonstanten Anteilen zusammen:

- Signallaufzeiten auf den Teilstrecken (abhängig vom gewählten Weg)
- Zugriffszeit auf das Übertragungsmedium (abhängig von der Last auf dieser Teilstrecke)

1. User Datagram Protocol gem. [RFC768]

- Wartedauern in den Zwischenpuffern und Verarbeitungsdauer innerhalb der Router (abhängig von der Last, den Verarbeitungskapazitäten und -strategien des Routers)
- Anzahl von Wiederholungen bis zum korrekten Empfang der Pakete (abhängig von der Bitfehlerwahrscheinlichkeit des Übertragungsmediums und dem Korrekturverfahren)

1.1.2.2 Integration netzfremder Dienste

Solange diese Netzwerke, wie ursprünglich vorgesehen, ausschließlich zur Übertragung textueller Daten (z.B. Email- oder Filetransfer) eingesetzt werden, beeinträchtigt die schwankende Übertragungsdauer die Qualität der Übertragung nicht. Synchrone Daten mit Realzeitforderungen wie Sprache oder Bewegtbilder, wie sie heute in zunehmendem Maße in den weltweiten Datennetzen übertragen werden, können über diese Netze jedoch nur bei schwacher Gesamtlast ohne hohen Qualitätsverlust transportiert werden (vgl. Abbildung 5 auf Seite 8).

1.1.3 Netzwerkangepaßte Dienste

Bis vor wenigen Jahren wurde das Fernsprechnet ausschließlich für Sprachübertragung genutzt, die vorhandenen Datennetze dienten der Übertragung textueller Informationen zwischen Rechnern, wie die beiden folgenden Tabellen zeigen:

Dienst	Zweck
Fernsprechen	Sprachübertragung

Tabelle 3 : Dienste in Fernsprechnetzen

Dienst	Zweck
Simple Mail Transport Protocol (SMTP)	Elektronische Post
File Transport Protocol (FTP)	Dateitransfer
Network File System (NFS)	Gemeinsame Nutzung von Speichermedien über Netzwerkverbindungen
Remote Procedure Call (RPC)	Ausführen von Programmen auf entfernten Rechnern

Tabelle 4 : Wichtige Dienste in Datennetzwerken

1.2 Problemstellung

1.2.1 Dienstintegration in bestehenden Netzwerken

Die Zweckentfremdung der spezialisierten Netzwerke zur Nutzung anderer Dienste nimmt, wie Tabelle 5 auf Seite 10 am Beispiel des Fernsprechnetzes zeigt, immer mehr zu:

Dienst	Bedeutung
Fax	Faksimile-Übertragung
Eurofiletransfer	Dateitransfer über Modem bzw. ISDN-Karte
Internet-Zugang	Nutzung von Datendiensten aus dem Internet
Fernüberwachung	Übertragung von Telemetriedaten
Short Message Service	Versand kurzer Nachrichten (Mails)

Tabelle 5 : „Fremde“ Dienste in Fernsprechnetzen

Da das Fernsprechnet das einzige Kommunikationsnetz ist, das üblich in jedem Haushalt zur Verfügung steht, wird die schlechte Anpassung der netzfremden Dienste und die daraus resultierende schwache Ausnutzung der Netzressourcen in Kauf genommen. Im Falle der Datennetze ist die sprunghaft gestiegene Leistungsfähigkeit der Arbeitsplatzrechner, insbesondere in den Bereichen Bild-, Audio- und Videodatenverarbeitung für das Aufkommen von Anwendungen, welche an die Datennetze Realzeitanforderungen stellen, verantwortlich (vergl. Tabelle 6 auf Seite 10).

Dienst	Bedeutung
Video on Demand	Echtzeitübertragung von Bewegtbildern
Audiodienste	Echtzeitübertragung von Sprache und Musik

Tabelle 6 : „Fremde“ Dienste in Datennetzwerken

1.2.2 Zukünftige Dienste

Zukünftige Dienste haben hohe und vielfältige Ansprüche an die Übertragungsnetzwerke. Einige dieser Dienste beschränken sich auf die Forderung nach höherer Übertragungsgeschwindigkeit, andere jedoch kombinieren unterschiedliche Dienste zu neuen Anwendungen mit oft sich widersprechenden Anforderungen an die zeitliche und syntaktische Transparenz der Übertragungsnetze. Tabelle 7 auf Seite 11 zeigt diese unterschiedlichen Bedürfnisse anhand einiger wichtiger neuerer Telekommunikationsdienste in

einem Überblick. Der Forderung nach höheren Übertragungsgeschwindigkeiten kann aufgrund des Fortschritts auf dem Gebiet der Übertragungs- und Halbleitertechnik durch eine einfache Skalierung bekannter Netzwerkprotokolle nachgekommen werden (z.B. Gigabit-Ethernet nach [IEEE802_3ab]). Eine bloße Steigerung der Übertragungsraten bedeutet keine Lösung für das Problem der prinzipiell nicht erreichbaren zeitlichen Transparenz in paketvermittelnden Rechnernetzen. Da eine multimediale Kommunikation mit gleichzeitigen Video-, Audio- und Datentransfers weder über klassische Datennetze (lastabhängige Übertragungszeiten, schlechte zeitliche Transparenz für synchrone Daten) noch über leitungsvermittelnde Netze (hohe Latenzzeiten beim Verbindungsaufbau, mangelhafte Ressourcenausnutzung beim Transport asynchroner Daten) möglich ist, mußte ein neuartiges, breitbandiges und diensteintegrierendes digitales Netzwerk, das B-ISDN¹, als Kompromiß zwischen den bestehenden Netztypen entwickelt werden.

Anwendung		Super-Computer-Verbindung	LAN-Verbindungen	Bildübertragungen	Videokonferenz	Multimedia-Verbindungen	Multimedia-Abriefdienste	Programm-Austausch	Programmverteilung TV	Programmverteilung HDTV
		Übertragungsrate	< 10 MBit/s		•	•	•	•	•	
	< 30 MBit/s		•	•		•	•	•		•
	> 100 MBit/s	•						•		•
Verkehrseigenschaften	Konstante Bitrate				•	•	•	•	•	•
	Variable Bitrate				•	•	•		•	•
	Burstartig	•	•	•		•	•			
Verkehrsbeziehung	Punkt-zu-Punkt	•	•	•	•	•	•	•		
	Mehrpunkt		•	•	•	•		•	•	•
	Punkt-zu-Mehrpunkt								•	•
Symmetrie	Unidirektional			•		•		•	•	•
	Bidirektional (asym.)	•	•				•			
	Bidirektional (sym.)	•	•		•	•				

Tabelle 7 : Vergleich der Eigenschaften einiger Telekommunikationsdienste

1. Broadband Integrated Services Digital Network

Anwendung		Super-Computer-Verbindung	LAN-Verbindungen	Bildübertragungen	Videokonferenz	Multimedia-Verbindungen	Multimedia-Abrufdienste	Programm-Austausch	Programmverteilung TV	Programmverteilung HDTV
Verbindungs- beziehung	Verbindungsorientiert	•	•	•	•	•	•	•	•	•
	Verbindungslos		•	•		•	•			

Tabelle 7 : Vergleich der Eigenschaften einiger Telekommunikationsdienste

1.2.3 Zusammenfassung

Ursprünglich bestand die Kommunikationsinfrastruktur aus mehreren nebeneinander existierenden spezialisierten Netzwerken, die im Hinblick auf einen bestimmten Dienst entworfen und optimiert wurden. Aufgrund dieser Spezialisierung sind sie jedoch nur eingeschränkt zum Transport von Daten anderer Dienste einsetzbar. Wegen des stark steigenden Interesses an multimedialen Kommunikationsformen, und mit dem Fortschritt auf den Gebieten der Halbleiter- und Softwaretechnologie stehen solche Dienste nach immer kürzerer Entwicklungszeit zur Verfügung. Da keines der vorhandenen Netze allen Anforderungen zufriedenstellend gerecht werden kann, wurde der Aufbau eines neuen diensteintegrierenden, hochbitratigen digitalen Netzwerks notwendig. Um den unterschiedlichen Anforderungsprofilen in etwa gerecht werden zu können, mußte die Vermittlungstechnik dieses Netzes ein Kompromiß zwischen der bisher verwendeten Leitungs- und Paketvermittlungstechnik sein. Standardisiert wurde ein solches Netzwerk von der ITU¹ (vormals CCITT²) unter der Bezeichnung B-ISDN³. Als Kompromiß zwischen den beiden Vermittlungstechniken vereint der im B-ISDN eingesetzte *Asynchrone Transfermodus ATM* die Vorteile beider Verfahren, weist aber auch deren Nachteile auf. Verluste, Verzögerungszeiten und -schwankungen nehmen auch hier mit steigender Last im Netz zu und beeinträchtigen insbesondere die synchronen Dienste mit ihren hohen Anforderungen an die zeitliche Transparenz des Übertragungssystems. Um diese Qualitätseinbußen möglichst klein zu halten, wurden daher spezielle Festlegungen bezüglich der Abwicklung des Datentransfers und der Verkehrslenkung getroffen.

1. International Telecommunication Union
 2. Comité Consultatif International Télégraphique et Téléphonique
 3. Broadband Integrated Services Digital Network

1.3 Vorgehensweise

Die *Connection Admission Control*¹ CAC, als ein wichtiger Bestandteil der Verkehrslenkung im B-ISDN, entscheidet über die Erfüllung der Verbindungswünsche zwischen den angeschlossenen Stationen. Ziel der vorliegenden Arbeit ist die Entwicklung eines effizienten Algorithmus zur Durchführung der CAC sowie die Erfassung der Auswirkung dieses Algorithmus auf die Qualität der beim Empfänger eintreffenden Multimediadatenströme. Im zweiten Kapitel sind einige der wichtigsten Gesetzmäßigkeiten der Stochastik und Bedientheorie zusammengestellt, welche die mathematische Grundlage zur Beschreibung von Vorgängen in Nachrichtennetzwerken bilden. Im anschließenden dritten Kapitel werden die wichtigsten Merkmale der Datenübertragung im B-ISDN anhand des von der ITU standardisierten Referenzmodells erläutert. Im vierten Kapitel werden die von der ITU standardisierten CAC-Verfahren beschrieben und diesen im darauffolgenden fünften Kapitel das im Verlauf der Arbeit entwickelte adaptive Verfahren gegenübergestellt. Zum Nachweis der Leistungsfähigkeit dieses Verfahrens im realen Betrieb wurde eine Simulationssoftware für ATM-Netzwerke entwickelt, deren Struktur in Kapitel sechs kurz erläutert wird. Die mit seiner Hilfe erzielten Ergebnisse stellt das darauffolgende Kapitel vor. Im abschließenden achten Kapitel werden die Ergebnisse zusammengefaßt; es enthält weiterhin Vorschläge zu möglichen Verbesserungen des Verfahrens, die im Rahmen dieser Arbeit leider nicht mehr untersucht werden konnten. Details zur Softwareimplementierung und den beschriebenen Protokollen können den jeweils erwähnten Abschnitten des Anhangs entnommen werden.

1. Verbindungsannahme

Statistik und Verkehrstheorie

Telekommunikationssysteme bestehen in der Regel aus einer Zusammenschaltung von Vermittlungsstellen und Endgeräten, deren Aufgabe es ist, Daten nach Vorgabe bestimmter Protokolle zu bearbeiten und anschließend weiterzuleiten. Die Verarbeitung der Daten hängt dabei üblich von mehreren Parametern ab (Länge der Daten, aktueller Zustand des Netzknotens, usw.). Die Zeit, die zur Abarbeitung der Protokollschritte notwendig ist, kann daher nicht vorausgesagt werden, ebensowenig wie die Ankunftszeit der Daten auf den Zubringerleitungen. Diese Vorgänge können folglich nur durch Zufallsgrößen beschrieben werden; die Anwendung der Gesetze der Statistik liefert dann eine Aussage über die Eigenschaften der Datenströme beim Verlassen des Systems.

2.1 Eigenschaften von Zufallsgrößen

2.1.1 Zufallsvariable

Def.: Eine Zufallsvariable ist eine Zahl $x(\xi)$, die jedem Ausgang ξ_i eines stochastischen Experimentes E zugeordnet wird.

2.1.2 Verteilungsfunktionen

Entscheidend zur Beschreibung stochastischer Experimente ist die *Verteilungsfunktion* $F_x(x)$, welche die Wahrscheinlichkeit dafür angibt, daß die Zufallsvariable x kleiner als eine beliebige Schranke x ist:

$$F_x(x) = P\{\mathbf{x} \leq x\}. \quad (Gl. 1)$$

Die Verteilungsdichtefunktion

$$f_x(x) = \frac{d}{dx}F_x(x) = P\{x \leq \mathbf{x} \leq x + dx\} \quad (Gl. 2)$$

gibt die Wahrscheinlichkeit dafür an, daß die Zufallsvariable \mathbf{x} zwischen den Schranken x und $x+dx$ liegt. Nimmt die Zufallsvariable \mathbf{x} ausschließlich diskrete Werte x_i an, so gilt für ihre Verteilungsfunktion

$$F_x(x) = P\{\mathbf{x} \leq x_i\} = \sum_{k=0}^i P\{\mathbf{x} = x_k\}, \quad (Gl. 3)$$

die entsprechende Verteilungsdichtefunktion lautet:

$$f_x(x) = \sum_{k=0}^{\infty} P\{\mathbf{x} = x_k\} \cdot \delta(x - x_k). \quad (Gl. 4)$$

2.1.3 Momente

Zur Charakterisierung von Zufallsvariablen werden häufig deren *statistische Momente* M_i (*Ensemblemittelwerte*) eingesetzt, für die gilt:

$$M_i = \int_{-\infty}^{\infty} f_x(x) \cdot x^i dx. \quad (Gl. 5)$$

Die wichtigsten Momente zeigt die folgende Tabelle:

Ordnung	Bezeichnung	Berechnung
1	Erwartungswert	$M_1 = E\{x\} = \int_{-\infty}^{\infty} f_x(x) \cdot x dx$
2	Quadratischer Erwartungswert	$M_2 = E\{x^2\} = \int_{-\infty}^{\infty} f_x(x) \cdot x^2 dx$

Tabelle 8 : Momente erster und zweiter Ordnung

Zentralmomente n -ter Ordnung sind definiert durch:

$$Z_i = \int_{-\infty}^{\infty} f_x(x) \cdot (x - E\{x\})^i dx, \quad (Gl. 6)$$

z.B. die Varianz :

$$Z_2 = \sigma^2 = \int_{-\infty}^{\infty} f_x(x) \cdot (x - E\{x\})^2 dx. \quad (Gl. 7)$$

Für diskrete Zufallsvariable vereinfachen sich diese Gleichungen zu:

$$M_i = \int_{-\infty}^{\infty} f_x(x) \cdot x^i dx = \int_{-\infty}^{\infty} \left(\sum_{k=0}^N P\{x=x_k\} \delta(x-x_k) \right) x^i dx = \sum_{k=0}^N P\{x=x_k\} x_k^i \quad (Gl. 8)$$

$$Z_i = \int_{-\infty}^{\infty} f_x(x) \cdot (x_k - E\{x\})^i dx = \sum_{k=0}^N P\{x=x_k\} (x_k - E\{x\})^i \quad (Gl. 9)$$

2.1.4 Wichtige Verteilungsfunktionen und Momente

Einige der wichtigsten Verteilungsfunktionen und deren erste Momente zeigen die beiden folgenden Tabellen.

Typ	$f_x(x)$	$E\{x\}$	Varianz
Negativ-Exponentielle Verteilung	$\lambda \cdot e^{-\lambda x}$	$\frac{1}{\lambda}$	$\frac{1}{\lambda^2}$
Gleichverteilung	$\begin{cases} \frac{1}{b-a} & \text{für } a \leq x \leq b \\ 0 & \text{sonst} \end{cases}$	$\frac{a+b}{2}$	$\frac{(b-a)^2}{12}$
Erlang _k -Verteilung	$\lambda \cdot \frac{(\lambda x)^{k-1}}{(k-1)!} \cdot e^{-\lambda x}$	$\frac{k}{\lambda}$	$\frac{k}{\lambda^2}$
Normalverteilung	$\frac{1}{\sigma\sqrt{2\pi}} \cdot e^{-(x-\eta)^2/2\sigma^2}$	η	σ^2
χ^2 -Verteilung mit f Freiheitsgraden	$\frac{1}{2^{f/2}\Gamma(f/2)} \cdot x^{f/2-1} \cdot e^{-x/2}$	f	$2f$

Tabelle 9 : Verteilungsdichtefunktionen kontinuierlicher Zufallsvariablen

Typ	$P\{x=k\}$	$E\{x\}$	Varianz
Geometrische Verteilung	$p \cdot (1-p)^{k-1}$	$\frac{1}{p}$	$\frac{1-p}{p^2}$
Binomialverteilung	$\binom{n}{k} p^k \cdot (1-p)^{n-k}$	np	$np(1-p)$
Poissonverteilung	$\frac{\lambda^k}{k!} \cdot e^{-\lambda}$	λ	λ
Hypergeometrische Verteilung	$\frac{\binom{R}{k} \cdot \binom{N-R}{n-k}}{\binom{N}{n}}$	$\frac{nR}{N}$	$\frac{nR}{N} \cdot \left(1 - \frac{R}{N}\right) \cdot \left(1 - \frac{n-1}{N-1}\right)$

Tabelle 10 : Verteilungsdichtefunktionen diskreter Zufallsvariablen

2.1.5 Momentengenerierende Funktionen

Kontinuierliche Zufallsvariable

Die Momentengenerierende Funktion einer Zufallsvariablen x ist definiert als die Laplace-Transformierte ihrer Verteilungsdichtefunktion :

$$\Phi(s) = \int_{-\infty}^{\infty} f_x(x) \cdot e^{sx} dx. \quad (Gl. 10)$$

Mit ihrer Hilfe können alle Momente der Zufallsvariablen berechnet werden, da (siehe [PAP89], S.116):

$$E\{x^n\} = \left. \frac{\partial^n \Phi(s)}{\partial s^n} \right|_{s=0} \quad (Gl. 11)$$

Diskrete Zufallsvariable

Die Momentengenerierende Funktion diskreter Zufallsvariablen ist definiert durch die Z-Transformierte ihrer Verteilungsdichtefunktion:

$$\Gamma(z) = \sum_{i=-\infty}^{\infty} P\{x=x_i\} \cdot z^i. \quad (Gl. 12)$$

Die Momente ergeben sich sukzessive aus der Gleichung

$$E\{\mathbf{x} \cdot (\mathbf{x} - 1) \cdot \dots \cdot (\mathbf{x} - k + 1)\} = \left. \frac{\partial^k \Gamma(z)}{\partial z^k} \right|_{z=1}. \quad (Gl. 13)$$

2.2 Stochastische Prozesse

Def.: Ein statistischer Prozeß $\mathbf{x}(t)$ ordnet jedem Ausgang ξ_i eines stochastischen Experimentes E eine Funktion $\mathbf{x}(t, \xi)$ zu.

Durch diese Zuordnung entsteht durch ein Experiment E ein Ensemble von zeitabhängigen Funktionen, die zusätzlich von einem Parameter ξ abhängen. Als Beispiel mag hier die Brown'sche Bewegung von Partikeln in Flüssigkeiten dienen; der Prozeß stellt die Gesamtheit der Bewegung dar, eine Realisierung die Bewegung eines bestimmten Partikels. Eine ausführliche Darstellung der Eigenschaften stochastischer Prozesse ist in [PAP89] zu finden.

2.2.1 Kenngrößen stochastischer Prozesse

Verteilungsfunktionen

$$F_x(x, t) = P\{\mathbf{x}(t) \leq x\}. \quad (Gl. 14)$$

$$f_x(x, t) = \frac{d}{dx} F_x(x, t) = P\{x \leq \mathbf{x}(t) \leq x + dx\}. \quad (Gl. 15)$$

Momente

$$M_i(t) = \int_{-\infty}^{\infty} f_x(x, t) \cdot x^i dx. \quad (Gl. 16)$$

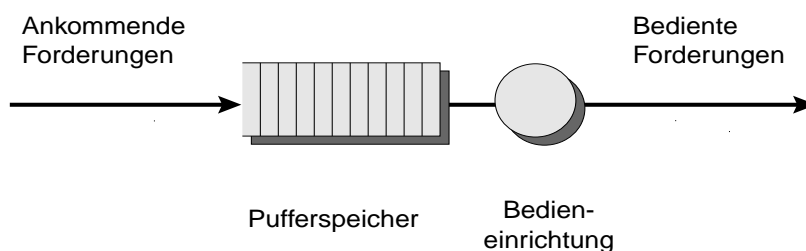
2.2.2 Stationarität und Ergodizität

Ein stochastischer Prozeß wird *stationär* genannt, wenn seine statistischen Eigenschaften nicht zeitabhängig sind. In diesem Fall gilt beispielsweise $M_1(t) = \eta(t) = \eta$ für alle t. Ein Prozeß ist *ergodisch*, wenn die *Ensemblemittelwerte* den durch zeitliche Mittelung an einer Realisierung gebildeten Mittelwerten entsprechen.

2.3 Bedientheorie

2.3.1 Klassifizierung von Bediensystemen

Die Bedien- oder Warteschlangentheorie beschäftigt sich u.a. mit der Beschreibung der Verarbeitung von Daten in Netzknoten. Ankommende Daten stellen Forderungen dar, die zu Zeitpunkten T_i eintreffen und vom System entweder sofort bearbeitet werden oder in einer Warteschlange zur späteren Bearbeitung abgelegt werden, wie die folgende Abbildung zeigt.



Beschreibung durch Kendall-Notation: $n/A/B/m/L$

- n: Anzahl der Ankunftsprozesse
- A: Ankunftsprozess, Verteilung der Zwischenankunftsabstände
- B: Bedienprozess, Verteilung der Bediendauern
- m: Anzahl der Bedieneinrichtungen
- L: Länge des Pufferspeichers

Abbildung 7: Bediensystem

Die wichtigsten Zufallsvariablen in diesem System sind in der folgenden Tabelle zusammengestellt:

Zufallsvariable	Bedeutung
T_n	Ankunftszeitpunkt der n-ten Forderung im System
$Z_n = T_n - T_{n-1}$	Zwischenankunftszeit der Forderungen
T_{Bn}	Service- oder Bediendauer
W_n	Wartedauer, Aufenthaltsdauer in der Warteschlange bis zur Bedienung
$V_n = T_{Bn} + W_n$	Verweildauer im System
D_n	Zeitpunkt, zu dem die bediente Forderung das System verläßt
$n(t)$	Anzahl der Forderungen im System

Tabelle 11: Zufallsvariable im Bediensystem aus Abbildung 7

Unter der Voraussetzung, daß Z_n und S_n unabhängige Zufallsgrößen sind, können bei gegebenen Verteilungen für Z_n und T_{Bn} die Momente der interessierenden Größen W_n oder n berechnet werden. Die wichtigsten Prozesse bei der analytischen Berechnung von Bediensystemen sind in Tabelle 12 auf Seite 21 zusammengefaßt.

Bezeichnung	Bedeutung
M	Markov-Prozeß Ein Prozeß mit negativ-exponentiell verteilten Zwischenankunftszeiten.
D	Deterministischer Prozeß Ein Prozeß mit festen (deterministischen) Zwischenankunftszeiten.
G	Genereller Prozeß Ein Prozeß ohne die Angabe einer konkreten Verteilungsfunktion
NG	Ein Summenprozeß aus N gleichen Prozessen.

Tabelle 12 : Wichtige Prozesse

2.3.2 Analyse von Bediensystemen

Die Bedienung von Forderungen in einem Bediensystem stellt einen stochastischen Prozeß dar. Einen Überblick zu den unterschiedlichen Klassen statistischer Prozesse, die zur analytischen Berechnung von Vorgängen in Bediensystemen herangezogen werden, zeigt Abbildung 8.

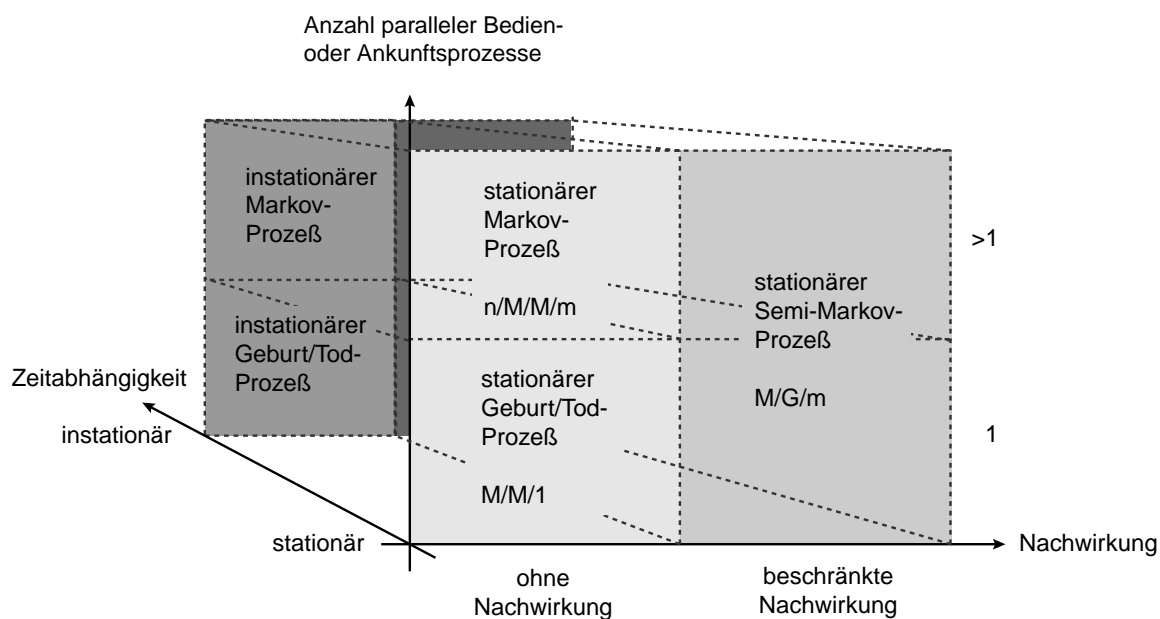


Abbildung 8 : Klassen von Bedienprozessen

Unter der Voraussetzung, daß zumindest entweder der Ankunftsprozeß oder der Bedienprozeß ein Markov-Prozeß¹ ist, lassen sich die Eigenschaften des Abgangsprozeß in einer solchen Bedieneinrichtung relativ einfach berechnen.

2.3.2.1 Bediensysteme M/G/1

Für die in Abbildung 7 gezeigte Bedieneinrichtung mit einer unendlich großen Pufferkapazität und einem Markov'schen Ankunftsprozeß gilt für die Anzahl der Forderungen im System (*Pollaczek-Khintchine-Formel*):

$$\Gamma_n(z) = \frac{p_0 \cdot (1 - z)}{1 - \frac{z}{\Phi_c(\lambda z - \lambda)}} \quad (\text{Gl. 17})$$

mit $\Phi_c(s)$ als Momentengenerierender Funktion der Bediendauern und p_0 der Wahrscheinlichkeit dafür, daß eine eintreffende Forderung das System leer antrifft. Für die Aufenthaltsdauer im System ergibt sich daraus:

$$\Phi_v(s) = \Gamma_n\left(1 + \frac{s}{\lambda}\right) = \frac{s \cdot \Phi_c(s) \cdot p_0}{(\lambda + s) - \lambda \Phi_c(s)} \quad (\text{Gl. 18})$$

Aus diesen Gleichungen können die Momente der Anzahl von Forderungen im System, der Verweil- sowie der Wartedauern durch Differentiation gewonnen werden (vgl. Gl. 11 auf Seite 18 und Gl. 13 auf Seite 19).

2.3.2.2 Verlustwahrscheinlichkeiten in Systemen mit endlicher Pufferlänge

Puffer in realen Systemen verfügen nur über eine endliche Anzahl von Warteplätzen; sind diese belegt, kommt es zu Datenverlust infolge Pufferüberlaufs. Die Wahrscheinlichkeit dieses Datenverlustes für Bediensysteme, die häufig als Modelle für Bedienvorgänge in ATM-Netzen herangezogen werden, zeigt Tabelle 13 auf Seite 23.

1. Gedächtnisloser, nachwirkungsfreier Prozeß, bei dem Ereignisse nicht von den Ereignissen in vorherigen Zeitintervallen abhängen (vgl. [PAP89], Kap. 12-4)

Bediensystem	Verlustwahrscheinlichkeit
M/M/1/L	ρ^L
M/D/1/L	$\approx e^{-2L(1-\rho)}$
ND/D/1/L	$\sum_{i=L+1}^N \left\{ \frac{N!}{i! \cdot (N-i)!} \cdot \left(\frac{i-L}{D}\right)^i \cdot \left[1 - \left(\frac{i-L}{D}\right)\right]^{N-i} \cdot \frac{D-N+L}{D-i+L} \right\}$ $\approx e^{-\left[2 \cdot L \cdot \left(\frac{L}{N} + \frac{1-\rho}{\rho}\right)\right]}$

Tabelle 13 : Verlustwahrscheinlichkeit in einfachen Bediensystemen mit $\rho = E\{T_B\}/E\{T\}$

2.3.2.3 End-to-End delay einer Kette von Bediensystemen

Zusätzliche Schwierigkeiten bei der analytischen Berechnung bereitet die Tatsache, daß Daten bei der Übertragung über ein Netzwerk in der Regel nacheinander mehrere Vermittlungsknoten passieren, bis sie ihren Bestimmungsort erreichen. Diese Situation zeigt Abbildung 9 auf Seite 23.

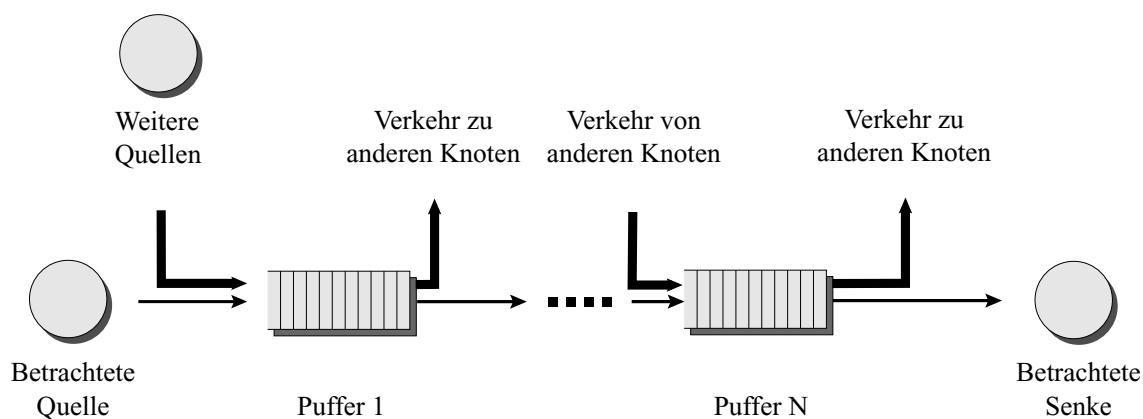


Abbildung 9 : End-to-End delay bei mehreren aufeinanderfolgenden Bediensystemen

In dem Fall einheitlich großer Datenpakete (z.B. Zellen) ergibt sich die Verteilungsfunktion der Gesamtverzögerung der Daten zwischen Quelle und Senke aus der sukzessiven Faltung der Wartezeitverteilungsfunktionen der Vermittlungsknoten. Diese Aussage gilt jedoch nur für den Fall, daß die Ankunftsprozesse der Puffer voneinander unabhängig sind. Diese Voraussetzung ist sicherlich nicht erfüllt, wenn der gesamte Verkehrsstrom am Ausgang von System n den Eingangsstrom für das System $n+1$ bildet. Eine Berechnung der Gesamtverzögerung mithilfe dieser Ableitung ist daher nur

dann möglich, wenn der betrachtete Verkehrsstrom nur einen sehr geringen Anteil am Verkehr auf jeder Teilstrecke hat und die Verkehrseigenschaften von den unabhängigen Verkehrsströmen der anderen Verbindungen dominiert werden.

2.3.2.4 Analytische Berechnung komplexerer Bediensysteme

Erfüllt weder der Ankunfts- noch der Bedienprozeß die Voraussetzung der Gedächtnislosigkeit, steigt der Aufwand bei der analytischen Berechnung sprunghaft an (vgl. [Kle75_3] Abschnitt 1.10 oder [Gne83] Kapitel 9). Bei Bediensystemen, die zur Beschreibung realer Systeme eingesetzt werden, ist jedoch häufig weder die Gedächtnislosigkeit noch die Unabhängigkeit des Bedienprozesses vom Ankunftsprozeß gegeben. Systeme mit komplexen Bedienvorgängen, die von mehreren internen Zuständen und von vorausgegangenen Ereignissen abhängen, können daher mit den analytischen Methoden nur näherungsweise berechnet werden (siehe z.B.: [Gne83], [Kle75_2], [Kle75_3], [Pap89], [Pit96], [Kou95] oder [Her91]). In diese Fällen ist eine Simulation zum Erhalt numerischer Lösungen einer analytischen Berechnung vorzuziehen.

2.4 Erzeugung von Zufallszahlen

Im Rahmen der Simulation eines Bediensystems müssen Zufallsgrößen wie Zwischenankunftsabstände, Länge von Datenpaketen usw. mithilfe von Zufallszahlengeneratoren erzeugt werden.

2.4.1 Erzeugung von Zufallszahlen mit beliebigen Verteilungsfunktionen

Die Erzeugung von Zufallszahlen mit beliebigen vorgebbaren Verteilungsfunktionen ist mithilfe der *Inversionsmethode* möglich. Eine im Intervall $[0,1]$ gleichverteilte Zufallsvariable x wird durch Anwendung der Umkehrfunktion $F^{-1}(x)$ transformiert und weist danach die gewünschte Verteilungsfunktion $F(x)$ auf (vgl. [Lie92], Kap. 3). Benötigt wird zur Erzeugung von Zufallszahlen mit beliebigen Verteilungsfunktionen demnach nur ein Generator, der gleichverteilte Zufallszahlen im Intervall $[0,1]$ erzeugt.

2.4.2 Generatoren für gleichverteilte Zufallszahlen

Mithilfe von Digitalrechnern können zwar lediglich Pseudozufallszahlen generiert werden, deren Verteilungsfunktionen bei geeigneten Algorithmen jedoch nicht von denen „echter“ Zufallszahlen zu unterscheiden sind. Die meisten heute verwendeten Generatoren verwenden die *lineare Kongruenzmethode*

zur Erzeugung der Pseudozufallszahlen. Dabei ergeben sich die Zufallszahlen aus folgender Rekursionsvorschrift:

$$x_i = (a \cdot x_{i-1} + c) \bmod m \quad (\text{Gl. 19})$$

Die statistischen Eigenschaften der so erzeugten Pseudozufallszahlenfolge hängen stark von den Parametern a , c und m ab. Wird die Konstante $c = 0$ gewählt, ergibt sich der Spezialfall des ebenfalls weit verbreiteten *multiplikativen Kongruenzgenerators*. Statistische Untersuchungen an den so erzeugten Zufallszahlenfolgen lieferten eine Reihe geeigneter Werte für die Konstanten a, c und m ; eine Aufstellung ist in [Bra87] zu finden.

2.5 Stichproben und Hypothesentests

2.5.1 Stichprobentheorie

Die Berechnung statistischer Kenngrößen, ausgehend von den Ergebnissen der Simulation eines Bediensystems, muß rechenzeitbedingt an einer begrenzten Anzahl von erzeugten Zufallsvariablen vorgenommen werden. Aussagen über die Zulässigkeit der Schätzung statistischer Größen aufgrund der Auswertung einer endlichen Anzahl von Proben liefert die *Stichprobentheorie* (vgl. [Spie90]). Sie ermöglicht die Durchführung von

- Hypothesentests,
- Schätzungen von Parametern und
- Anpassungstests an Verteilungen.

2.5.2 χ^2 -Anpassungstest

Mittels des χ^2 -Tests (vgl. [Spie90], Kap. 12) kann die Übereinstimmung einer Eigenschaft einer Stichprobe (Erwartungswert, Varianz, Verteilungsfunktion, ...) mit einem vermuteten Wert getestet werden. Hierzu wird der Wertebereich der Zufallsgröße in N Intervalle aufgeteilt und die Differenzen zwischen den beobachteten Werten \mathbf{B}_i und den theoretisch erwarteten Werten b_i gebildet:

$$\mathbf{D} = \sum_{i=0}^{N-1} \frac{(\mathbf{B}_i - b_i)^2}{b_i} \quad (\text{Gl. 20})$$

Die Differenz \mathbf{D} ist χ^2 -verteilt mit $f = N-1$ Freiheitsgraden (vgl. Tabelle 9 auf Seite 17 und Abbildung 10 auf Seite 26) und stellt eine Maß für die Abweichung zwischen der hypothetischen und der wahren Verteilung dar. Die Hypothese, daß die Meßwerte einer vermuteten Verteilung genügen, muß verworfen werden, wenn \mathbf{D} den kritischen Wert χ^2_α zum *Signifikanzniveau* α überschreitet. Für eine große Anzahl von Freiheitsgraden geht die χ^2 -Verteilung in die Normalverteilung über; χ^2_α kann dann über $\chi^2_\alpha = \frac{1}{2} \cdot (z_\alpha + \sqrt{2f-1})^2$ aus den kritischen Werten z_α der Normalverteilung berechnet werden.

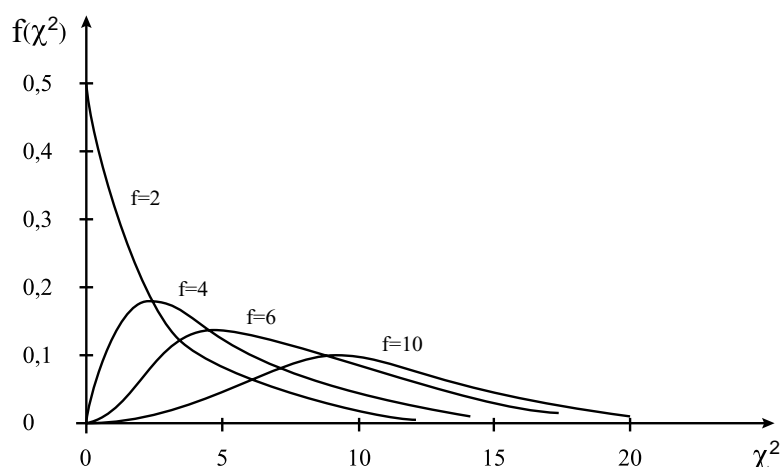


Abbildung 10 : χ^2 -Verteilung für verschiedene Freiheitsgrade f

2.6 Ankunftsprozesse realer Datenquellen

Die Ankunftsprozesse der Bediensysteme werden in der Realität durch Anwendungen erzeugt, welche Datenströme mit konstanter (*Constant Bitrate CBR*) oder mit variabler Bitrate (*Variable Bitrate VBR*) hervorrufen. Die Nachbildung innerhalb einer Simulation erfolgt durch entsprechend dimensionierte Zufallszahlengeneratoren.

Quellen mit CBR-Charakteristik

Eine Quelle mit CBR-Charakteristik weist eine konstante Bitrate λ_0 während der gesamten Dauer ihres Bestehens auf. Als Beispiele für eine solche Datenquelle können eine digitale Fernsprechverbindung mit einer konstanten Bitrate von 64 kBit/s oder der digitale Ausgang eines DAT¹-Recorders mit 768 kBit/s gelten.

1. Digital Audio Tape

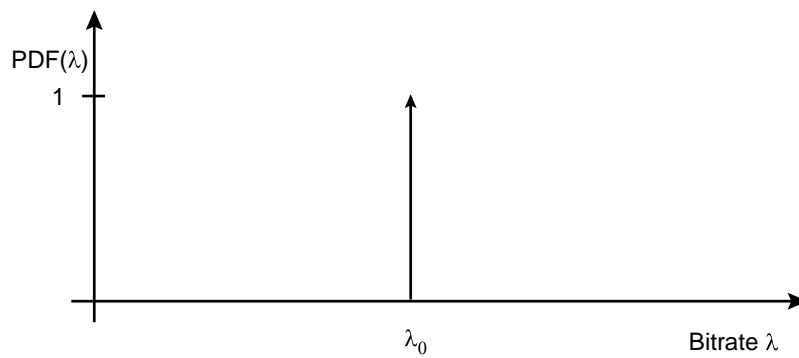


Abbildung 11 : PDF^1 der Bitrate einer CBR-Quelle

2.6.1 Quellen mit VBR-Charakteristik

Quellen mit VBR-Charakteristik weisen eine beliebige, sich mit der Zeit verändernde Bitrate $\lambda(t)$ auf (vgl. Abbildung 12 auf Seite 27).

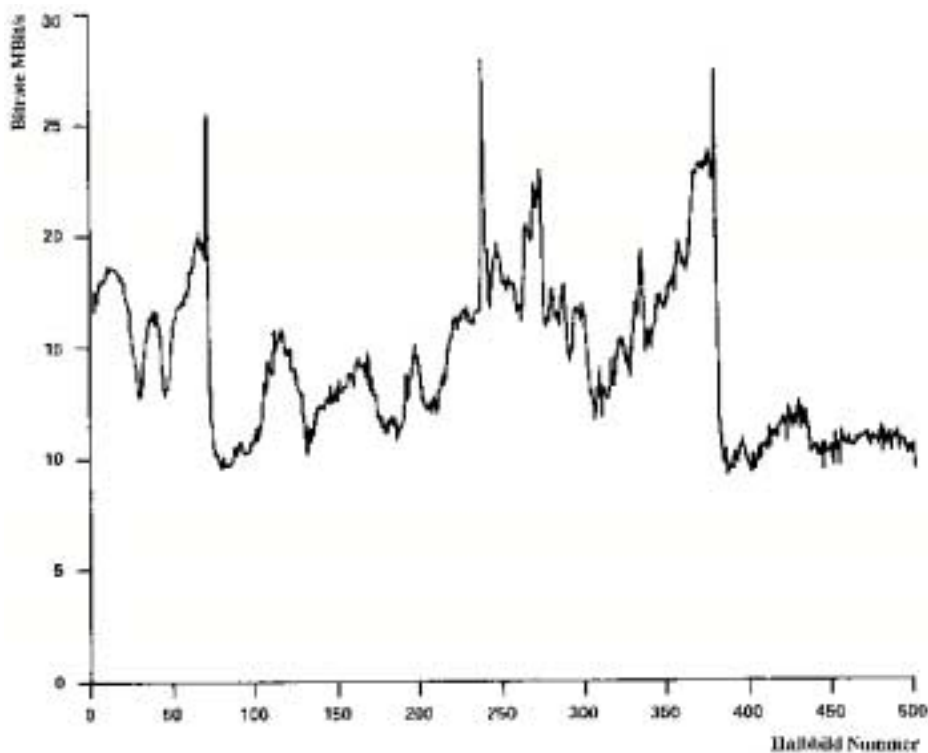


Abbildung 12 : Bitratenvariation bei TV-Übertragung [Ver89]

Für die in der Praxis üblichen relevanten Quellen lassen sich jedoch Nachbildungen durch die Überlagerung einfacher im Folgenden beschriebenen „Elementarquellen“ angeben.

1. Probability Density Function

2.6.1.1 Markov-Quelle

Die für analytische Berechnungen wichtigste Quelle mit variabler Bitrate ist die Markov-Quelle mit einer negativ-exponentiellen Verteilung der Zwischenankunftszeiten (vgl. Tabelle 9 auf Seite 17).

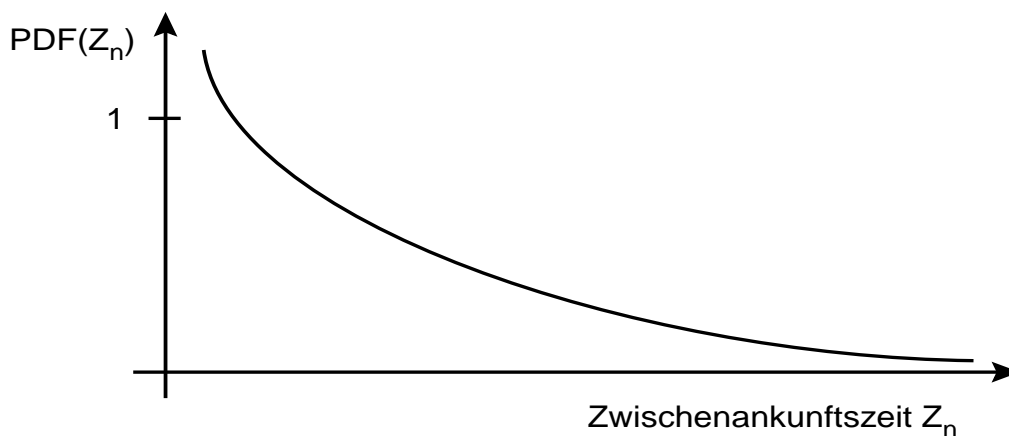


Abbildung 13 : PDF der Zwischenankunftszeiten einer VBR-Quelle mit NE-Charakteristik

2.6.1.2 Weitere elementare Quellen

Weitere Datenquellen mit den in Tabelle 9 und Tabelle 10 auf Seite 18 gegebenen Verteilungsfunktionen können mit Hilfe der Inversionsmethode (siehe [Neu93], S. 705 ff.) erzeugt werden.

2.6.1.3 Zweistufige Prozesse

Eine Vielzahl von Quellen weist eine Unterteilung in Aktivitäts- und Pausenintervalle auf. Solche Quellen werden durch ein System mit zwei unterschiedlichen Zuständen modelliert, denen jeweils eine feste Datenrate zugeordnet ist (siehe Abbildung 14).

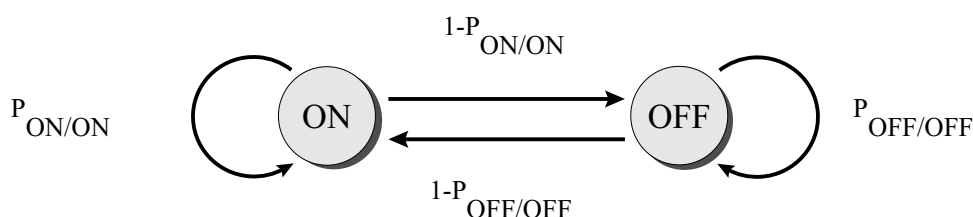


Abbildung 14 : Zustandsdiagramm einer ON/OFF-Quelle

Das System befinde sich im Zustand „ON“, in dem es mit der Rate λ_{ON} Datenpakete erzeugt. Es verläßt diesen Zustand mit der Wahrscheinlichkeit $1-P_{\text{ON/ON}}$ und wechselt in den Zustand „OFF“, in dem es keine Datenpakete verschickt. In allgemeiner Form werden solche Modelle als *zweistufige* oder *modulierte Prozesse* behandelt. Die erste Stufe bezieht sich hierbei auf den Wechsel zwischen den einzelnen Zuständen, der Prozeß der zweiten Stufe beschreibt die Generierung von Daten in einem der Zustände.

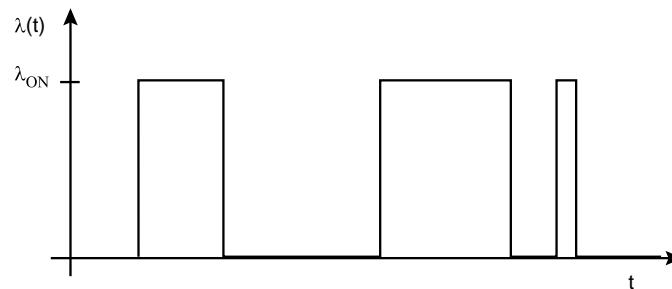
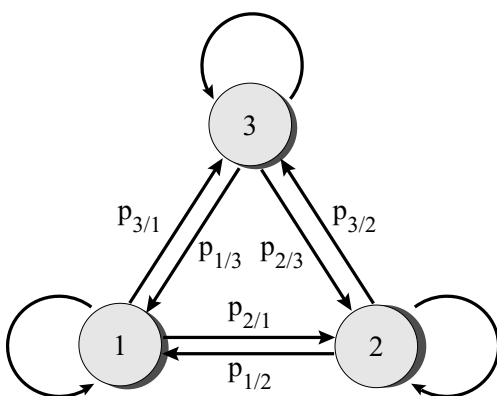


Abbildung 15 : Bitrate $\lambda(t)$ einer ON/OFF-Quelle

Zur Modellierung realer Quellen muß das Modell aus zwei Zuständen auf mehrere Zustände erweitert werden. Unterliegen die Aufenthaltsdauern in den Zuständen einer negativ-exponentiellen Verteilung, so führt dieser Ansatz zu den *Markov Modulierten Prozessen*. Bei festen Senderaten λ_i in jedem der Systemzustände z_i (siehe Abbildung 15) handelt es sich um einen Markov Modulated Deterministic Process MMDP, bei negativ-exponentiell verteilten Zwischenabgangszeiten um einen Markov Modulated Poisson Process MMPP.



Übergangswahrscheinlichkeiten

$$P = \begin{bmatrix} p_{1/1} & p_{1/2} & p_{1/3} \\ p_{2/1} & p_{2/2} & p_{2/3} \\ p_{3/1} & p_{3/2} & p_{3/3} \end{bmatrix}$$

Abbildung 16 : Modulierter Prozeß mit drei Systemzuständen

Am Ende des Aufenthaltes in einem Zustand i wechselt das System mit einer bestimmten Wahrscheinlichkeit $p_{j/i}$ in den Folgezustand j . Die Übergänge werden von einer $N \times N$ -Matrix P der Über-

gangswahrscheinlichkeiten gesteuert, wie sie in Abbildung 16 gezeigt ist. Die vielfältigen Konfigurierungsmöglichkeiten eines solchen Verkehrsmodells erlauben die Nachbildung komplexerer Datenquellen, wie sie von realen Applikationen erzeugt werden. Ausgewählte Beispiele für solche VBR-Quellen zeigen die beiden folgenden Abbildungen:

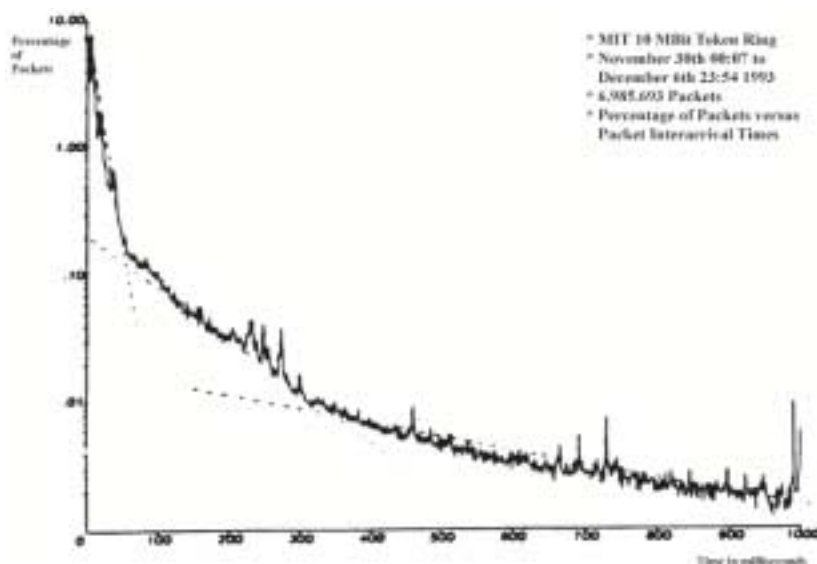


Abbildung 17 : PDF der Paketzwiseankunftsabstände in einem LAN (Token Ring) gem. [Gag95]

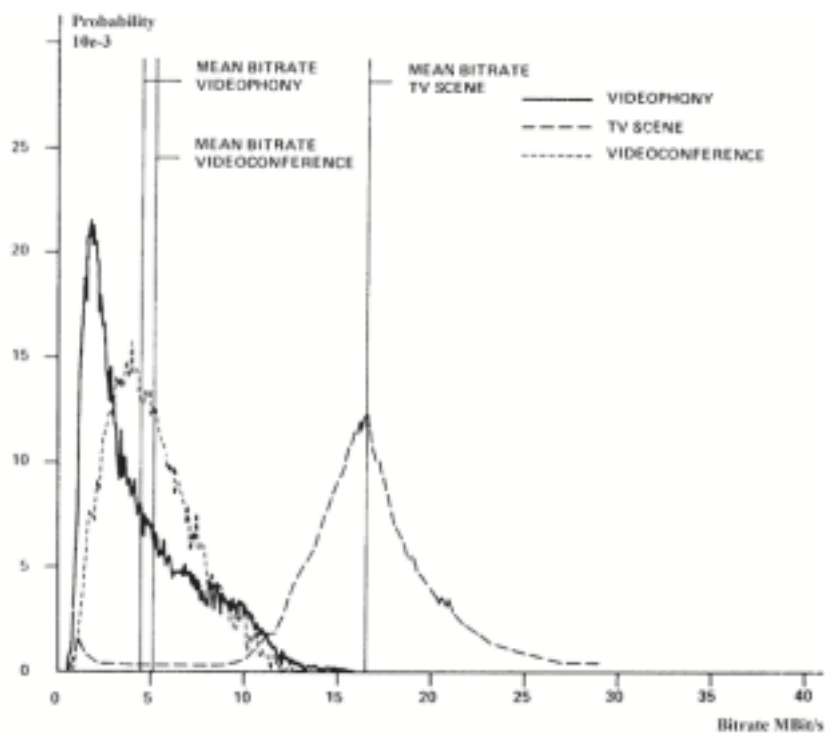


Abbildung 18 : PDF der Bitrate einiger Videoapplikationen nach [Ver89], Fig. 1

Zur Gewinnung von Quellenmodellen für Simulationszwecke werden die Parameter eines modulierten Prozesses an die aus Messungen an den gezeigten realen Datenquellen ermittelten Werte angepaßt (siehe z.B. [Gag95] für den LAN-Verkehr aus Abbildung 17 auf Seite 30, [Li90] oder [Leh95]).

2.6.2 Multiplexgewinn bei VBR-Quellen

Betrachtet man stark bursthaltige Verkehrstypen wie in Abbildung 12 auf Seite 27, zeigt sich eine Möglichkeit zur Einsparung an der Systemressource „Puffergröße“, wenn mehrere solcher Quellen gemultiplext und übertragen werden.

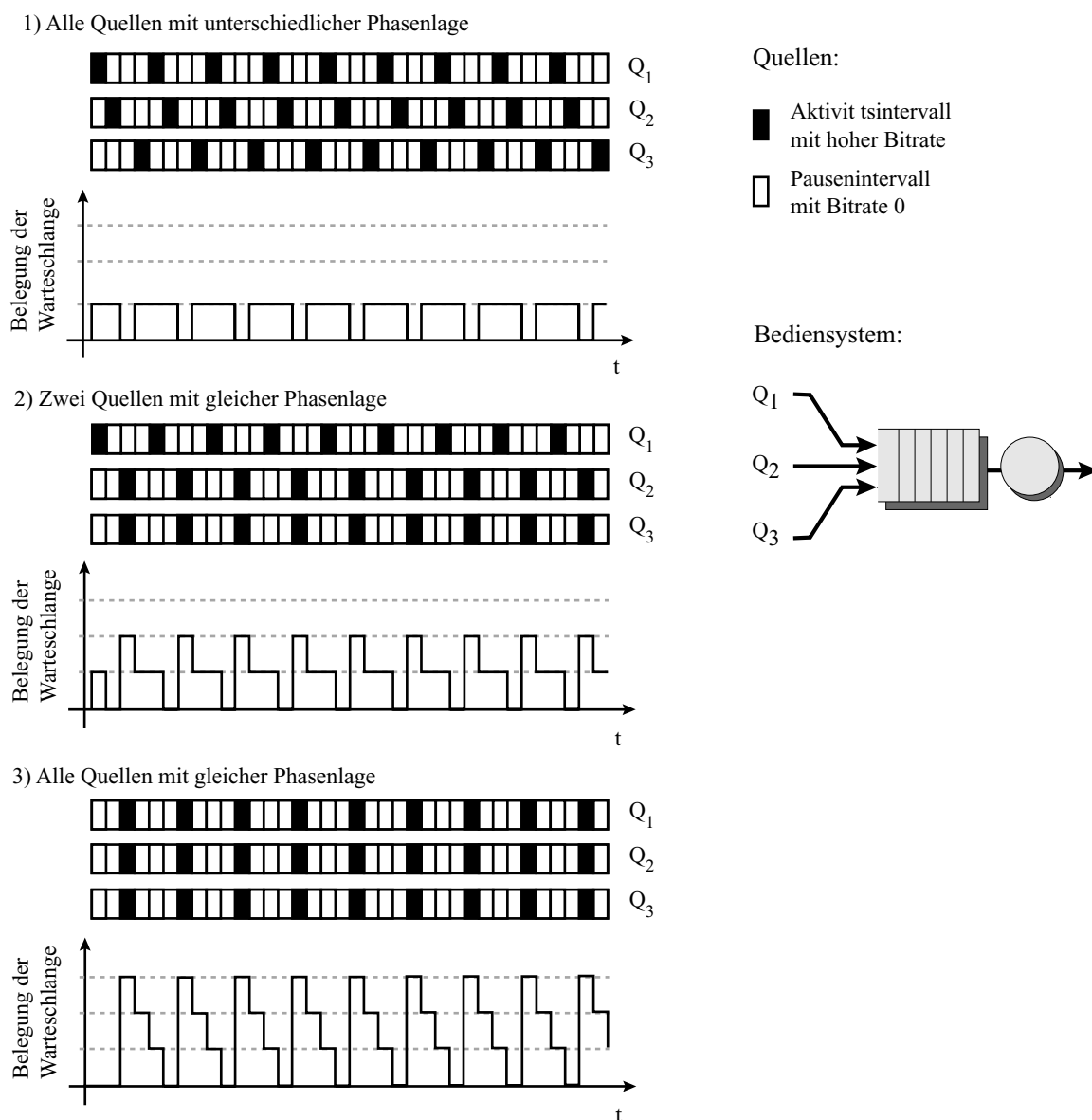


Abbildung 19 : Pufferbelegung beim Multiplex dreier VBR-Quellen

Die Ausnutzung dieses *Multiplexgewinns* und die daraus resultierende Einsparung an Systemressourcen war eines der wichtigsten Argumente zur Einführung des B-ISDN, die Ausnutzung dieses Effekts ist eine der Aufgaben der Verkehrslenkung im B-ISDN. Wie das Beispiel aus Abbildung 19 auf Seite 31 zeigt, hat die Phasenlage zwischen den bursthaltigen Quellen einen entscheidenden Einfluß auf die Pufferbelegung, und daraus resultierend auf die Paketverlustwahrscheinlichkeit und die Wartedauerverteilung. Die zeitlichen Verschiebungen zwischen den Datenpaketen der Quellen Q_i werden bestimmt durch die *Korrelation* zwischen den Datenströmen. Dieses Maß für die statistische Bindung variiert zwischen niedrigen Werten für unabhängige Quellen (Daten unterschiedlichen Ursprungs) und hohen Werten für zusammenhängende Quellen (z.B. Teilströmen von Multimediatelefonkonferenzen).

2.7 Zusammenfassung

Die in Kapitel 2 erläuterten Zusammenhänge zeigen die Schwierigkeiten bei der analytischen Berechnung statistischer Prozesse in realen Bediensystemen auf. Deren Einsatzgebiet beschränkt sich aufgrund der unerläßlichen idealisierenden Annahmen über Quelleigenschaften und Bedienvorgänge auf relativ einfache Systeme oder Grenzwertbetrachtungen (vgl. [Mit89], [Li93], [Gué91], [Ngo91], [Mit95]). Korrelationen zwischen komplexen Quellen, Abhängigkeiten des Bedienvorgangs von vorhergehenden Ereignissen sowie die gegenseitige Beeinflussung mehrerer in Reihe angeordneter Bediensysteme müssen weitgehend unberücksichtigt bleiben, daher sind die Ergebnisse nur begrenzt auf reale Systeme übertragbar. Aufschluß über die tatsächliche Leistungsfähigkeit eines vernetzten Gesamtsystems beim Betrieb unter realen Protokollen kann infolgedessen nur eine Simulation bringen.

Grundlagen der Daten- übertragung im B-ISDN

3.1 Übermittlungsprinzip im B-ISDN

3.1.1 Der Asynchrone Transportmodus ATM

Als Kompromiß zwischen Leitungs- und Paketvermittlung (vgl. Kapitel 1.1 auf Seite 1 ff.) wurde für das B-ISDN der Asynchrone Transfermodus ATM eingeführt, der die Vorteile beider Vermittlungsprinzipien vereinen sollte. Von der Leitungsvermittlung wird der dem Datentransport vorausgehende Verbindungsaufbau und die festgelegten Wege übernommen, die Paketvermittlung steuert das Datenformat und das Speichervermittlungsprinzip in den Netzknoten bei.

Virtuelle Verbindungen

Im Gegensatz zur Leitungsvermittlung werden beim ATM nur sog. *Virtuelle Verbindungen* aufgebaut. Diese legen zwar den Weg für alle Pakete einer Verbindung im Netz fest und fordern Ressourcen in den Netzknoten an, jedoch findet keine exklusive Reservierung dieser Ressourcen statt.

Einheitliche Paketlänge

Es werden kurze Pakete einheitlicher Länge, sog. „Zellen“ verwendet. Alle diese Zellen weisen eine Gesamtlänge von 53 Oktetts auf, davon 48 Oktetts Nutzinformation sowie 5 Oktett Steuerinformationen im Zellkopf (Header). Dieser Aufbau vereinfacht die Verarbeitung in den ATM-Vermittlungsstellen und die Verarbeitungsdauer ist daher nahezu konstant.

Zellstrom auf dem Medium

Die Zellen werden asynchron in einen synchronen Zellstrom auf dem Übertragungsmedium eingespeist, ungenutzte Zeitschlitze werden mit Leerzellen (*Idle Cells*) aufgefüllt (siehe Abbildung 20).



Abbildung 20 : Asynchrones Einfügen von Nutzdatenzellen in den synchronen Zellstrom auf dem Übertragungsmedium

Datenpakete, deren Länge 48 Byte überschreitet, werden auf mehrere Zellen aufgeteilt, kürzere Datenpakete (z.B. Sprachabstastwerte) entweder bis zur Länge von 48 Byte gesammelt oder mit Füllinformation ergänzt. Alternativ hierzu ist auch der Einbau der Zellen in einen höheren Übertragungsrahmen, beispielsweise aus der *Synchronen Digitalen Hierarchie SDH*, möglich.

Fehlersicherung

Aufgrund der niedrigen Bitfehlerwahrscheinlichkeit der modernen optischen Übertragungsmedien im Vergleich zu den älteren galvanischen Leitern konnte die früher im Netzzinnern abschnittsweise durchgeführte Fehlersicherung in die Verbindungsendpunkte am Netzrand verlagert werden.

Flußsteuerung

Auf den Verbindungsabschnitten im Netzzinnern findet wegen der hohen Übertragungsraten und der geforderten Verarbeitungsgeschwindigkeiten keine Datenflußkontrolle statt.

3.1.2 Multiplex im B-ISDN

An jedem ATM-Anschluß können gleichzeitig mehrere parallele Virtuelle Verbindungen bestehen, deren Daten in den Zellstrom auf dem Übertragungsmedium eingefügt werden müssen. Treffen pro Zellslot auf dem Medium mehrere Zellen ein, so werden diese in einem Puffer zwischengespeichert (siehe

Abbildung 21 auf Seite 35). Liefern die Quellen mehr Zellen als der Puffer aufnehmen kann, gehen diese verloren.

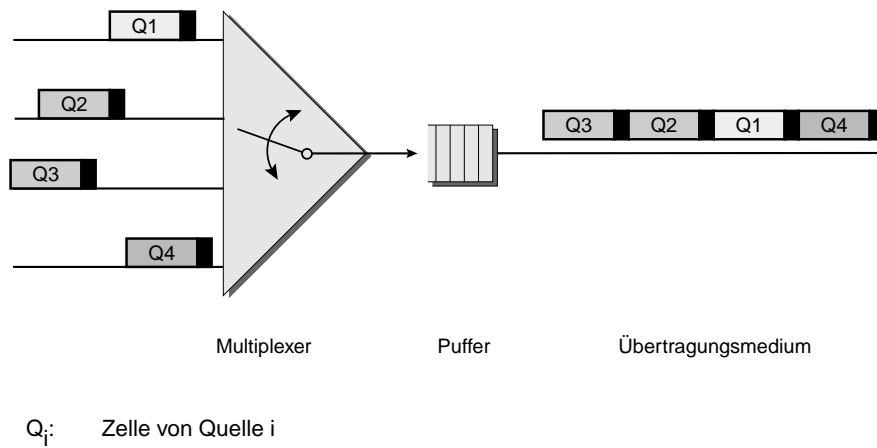


Abbildung 21 : Multiplex Virtueller Verbindungen

3.1.3 End-to-End Delay

Die Nutzdaten erfahren bis zum Eintreffen auf dem Übertragungsmedium Verzögerungen, die sich aus zwei wesentlichen Anteilen zusammensetzen:

- Paketierungsverzögerung
- Zellverzögerung durch den Multiplexvorgang

Paketierungsverzögerung

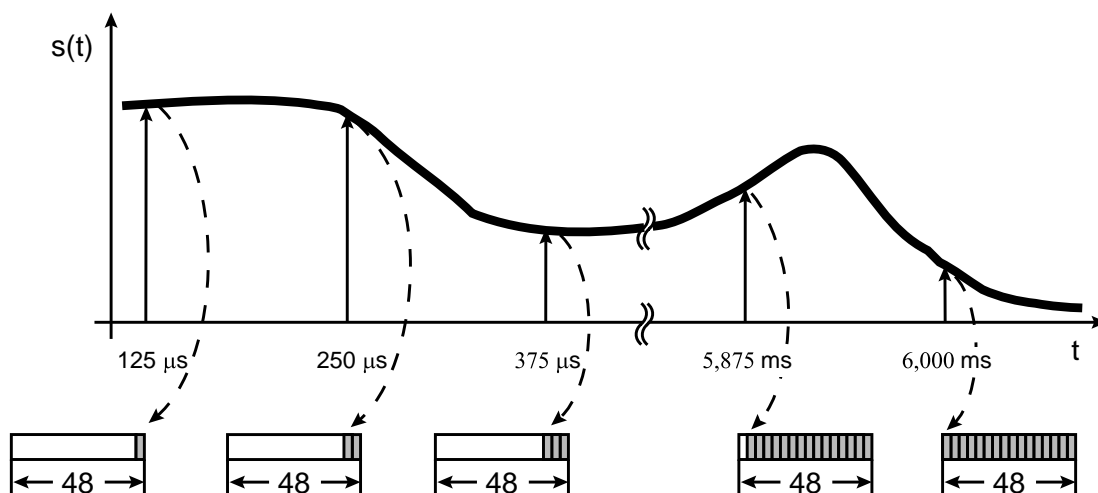


Abbildung 22 : Füllen des Nutzlastbereichs einer Zelle mit Sprachabtastwerten

Die *Paketierungsverzögerung* entsteht während der Bildung der Zellnutzinformation aus den Quelldaten bei der Übertragung kurzer Nutzdatenpakete, z.B. Abtastwerten von Fernsprechquellen; so vergehen 6 ms bis das Informationsfeld einer Zelle vollständig mit Sprachabtastwerten gefüllt ist. Da dieser Wert für Sprachverbindungen nicht akzeptabel ist, müssen Zellen mit unvollständig gefülltem Informationsfeld versendet werden, wodurch die Effizienz der Übertragung sinkt.

Verzögerungen durch den Multiplexvorgang

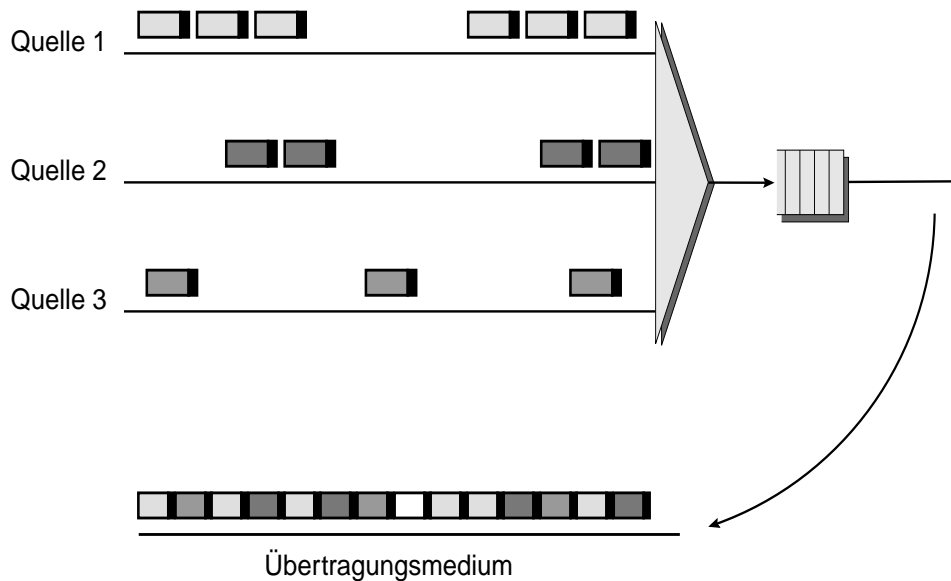


Abbildung 23 : Multiplex der Zellen dreier Quellen

Wie in Abbildung 23 gezeigt, erfahren die Zellen beim Multiplexen auf eine Anschlußleitung durch die gemeinsame Nutzung des Puffers unterschiedliche Verzögerungen. Vergleicht man die Ankunftszeitpunkte der Zellen auf dem Übertragungsmedium mit denen auf den Zubringerleitungen, erkennt man die gegenseitige Beeinflussung der Datenströme (siehe Abbildung 24 auf Seite 37). Die Verzögerungen, die durch den Multiplexvorgang verursacht werden, verändern die zeitlichen Abstände der Zellen; bei Erhöhung der Ankunftsraten gehen zudem durch Pufferüberlauf Zellen verloren. In einem solchen System kann demnach weder die syntaktische noch die zeitliche Transparenz für alle Lastfälle garantiert werden.

Gesamtverzögerung

Die Gesamtverzögerung der Daten ergibt sich hier als die nichtkonstante Summe aus der Paketierungsverzögerung, den konstanten Ausbreitungszeiten auf den Teilstrecken und der lastabhängigen Verzögerung durch den Multiplex- oder Vermittlungsvorgang in jedem Netzknoten.

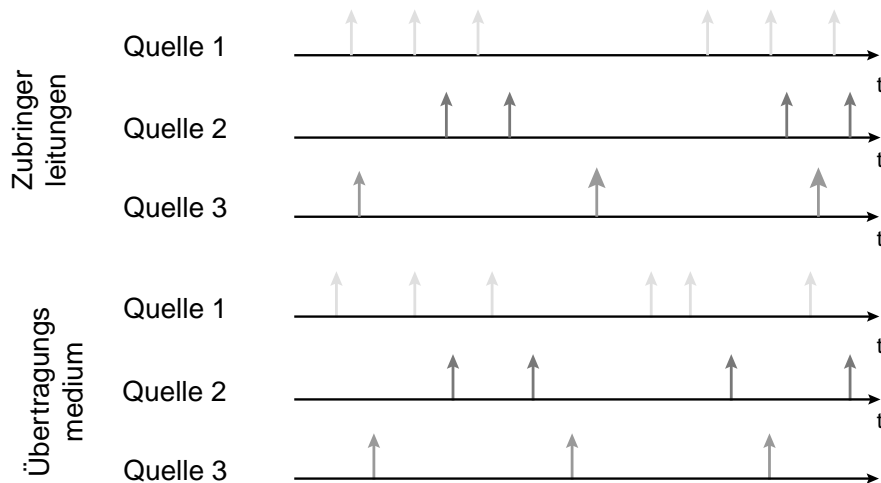


Abbildung 24 : Ankunftszeiten der Zellen auf Zubringerleitungen und dem Übertragungsmedium aus Abbildung 23

3.2 Datenformat im B-ISDN

Bits	8	7	6	5	4	3	2	1	Octet
	GFC				VPI				1
	VPI				VCI				2
	VCI								3
	VCI				PT		CLP		4
	HEC								5
	Information Field								6
									52
									53

Tabelle 14 : Aufbau einer ATM-Zelle am UNI [I361]

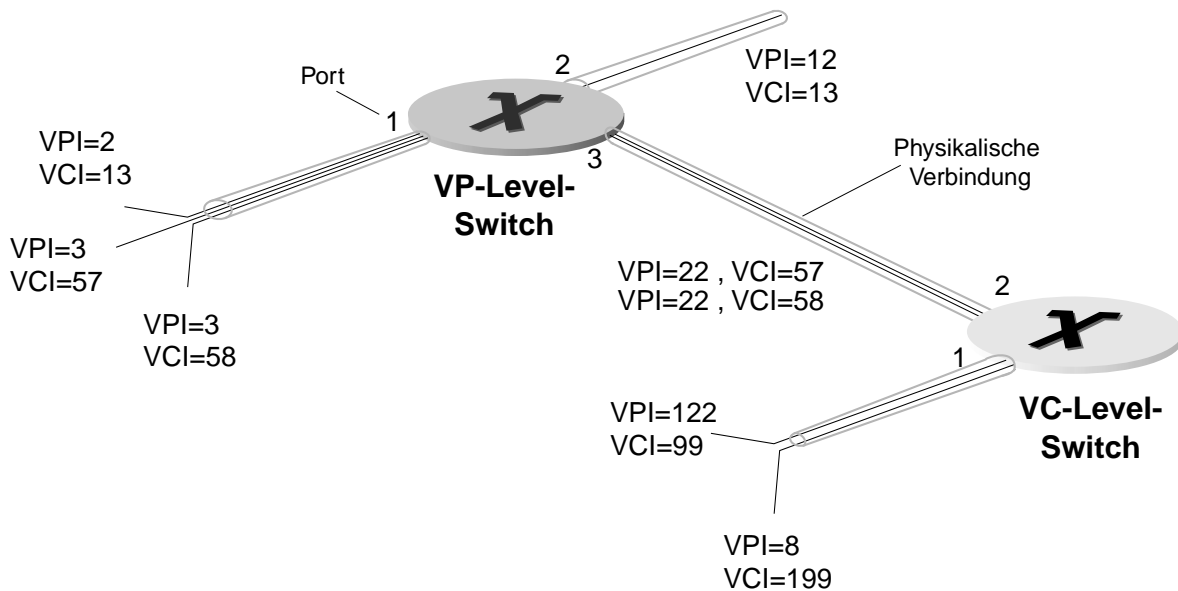
GFC	Generic Flow Control
VPI	Virtual Path Identifier
VCI	Virtual Channel Identifier
PT	Payload Type
CLP	Cell Loss Priority
HEC	Header Error Control

Tabelle 15 : Bedeutung der Felder im Zellkopf

Tabelle 14 zeigt den von der ITU in der Empfehlung I.361 [I361] festgelegten Aufbau einer ATM-Zelle an der Teilnehmer-/Netzchnittstelle (*User Network Interface UNI*).

3.3 Vermittlungstechnik im B-ISDN

Der *Virtual Path Identifier VPI* und der *Virtual Channel Identifier VCI* stellen die Verbindungsnummern auf den Teilstrecken innerhalb des ATM-Netzes dar. Zur Erläuterung ist in der folgenden Abbildung ein Ausschnitt aus einem ATM-Netz gezeigt. Beim Aufbau der virtuellen Verbindungen wird jedem Link¹ ein nur für diesen Streckenabschnitt gültiger *Connection Identifier*, bestehend aus dem VPI und dem VCI, zugeordnet und von den Vermittlungsstellen in einer *Routing-Table* gespeichert (vgl. Abbildung 25 auf Seite 38). Die Aufgabe der *Switches*² besteht darin, den VPI- und VCI-Wert der ankommenden Zellen im Kopf zu ersetzen, bevor sie auf den weiterführenden Abschnitt gesendet werden. Dabei schaltet eine VP-Level-Switch ausschließlich Virtuelle Pfade, die mehrere Virtuelle Kanäle umfassen, während eine VC-Level-Switch sowohl Kanäle als auch Pfade vermittelt.



Routing-Table VP-Level Switch

IN			OUT		
Port	VPI	VCI	Port	VPI	VCI
1	2	*	2	12	*
1	3	*	3	22	*

Routing-Table VC-Level Switch

IN			OUT		
Port	VPI	VCI	Port	VPI	VCI
2	22	57	1	8	199
2	22	58	1	122	99

Abbildung 25 : Vermittlung im ATM-Netz

1. Teilstrecke
2. Vermittlungsstellen

3.3.1 Verbindung zwischen Endgeräten

Der Aufbau einer Virtuellen Verbindung zwischen den Endteilnehmern im B-ISDN erfolgt durch Signalisierung innerhalb separater Kanäle. Diese Kanäle müssen, wie die Nutzdatenkanäle, vor dem Datenaustausch ausgehandelt und aufgebaut werden, wozu permanent vorhandene *Metasignalkanäle MSVC*¹ mit VPI = X (X ≠ 0) und VCI=1 dienen. Die weiteren reservierten Werte für VCI und VPI sowie deren Verwendung sind in [I361], Table 2/I.361 zu finden. Die Switches bilden die Endpunkte aller Teilstrecken der Gesamtverbindung, der *Virtual Channel Connection VCC* (vgl. Abbildung 26).

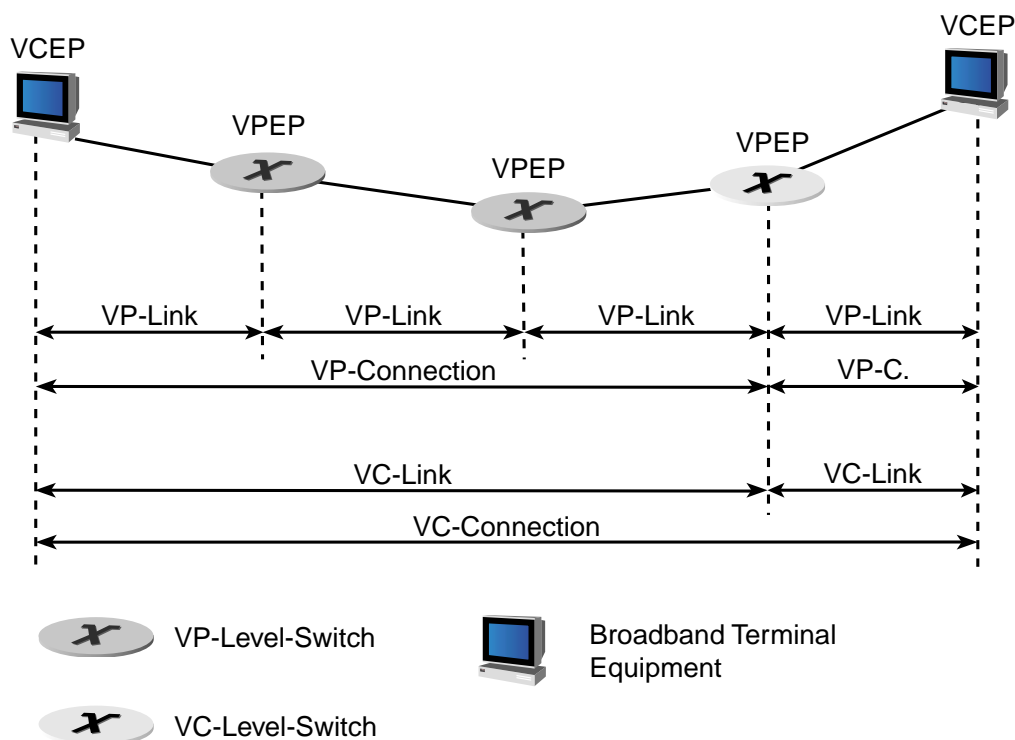


Abbildung 26 : Virtual Channel Connection

3.3.2 ATM-Vermittlungsstellen (Switches)

Treffen an einer Switch während einer Zelldauer mehrere Zellen für einen bestimmten Ausgang ein, kann nur eine Zelle direkt weitergesendet werden, die restlichen müssen in Zwischenpuffern für einen späteren Sendevorgang gespeichert werden. Abhängig davon, an welcher Position in der Vermittlungsstelle diese Puffer angeordnet sind, unterscheidet man drei Basistypen: Vermittlungsstellen mit

1. Metasignalling Virtual Channel

Puffern am Eingang (*Input Queuing*), Puffern am Ausgang (*Output Queuing*) und zentralen Puffern (*Central Queuing*). Den prinzipiellen Aufbau dieser Vermittlungsstellen zeigt Abbildung 27.

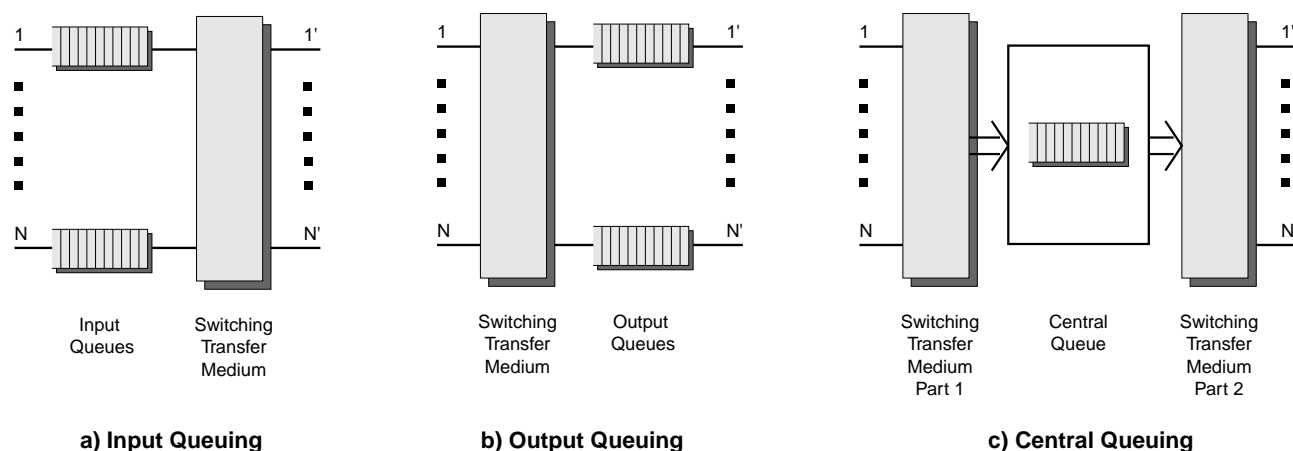


Abbildung 27 : Prinzipieller Aufbau von ATM-Vermittlungsstellen nach : [Pry93] Sec. 4.3

Trotz der verbindungsorientierten Vermittlung kann es in ATM-Netzwerken, im Gegensatz zu den leitungsvermittelnden Fernsprechnetzen, zu Blockierungen, Verzögerungen und Datenverlusten kommen. Eine vereinfachte analytische Abschätzung der Verlustwahrscheinlichkeiten und Wartedauern für die drei genannten Varianten gibt [Pry93] in Abschnitt 4.3.2.1. Eine Beschreibung bekannter auf diesen Prinzipien beruhender Switches ist in Abschnitt 4.4. desselben Werkes zu finden.

3.4 Telekommunikationsstandards

Um die Interoperabilität zwischen Telekommunikationssystemen unterschiedlicher Hersteller zu gewährleisten, erarbeiten Gremien innerhalb der ISO¹ und der ITU international gültige Standards für Kommunikationsnetze und -protokolle. Durch Zusammenarbeit von Industrie, Kunden und akademischen Organisationen entstehen die Richtlinien für den Einsatz neuer Technologien und die Entwicklung neuer Produkte auf dem Telekommunikationssektor. Die beiden oben genannten Organisationen sind unter anderem verantwortlich für die Definition des OSI²-Modells, das heute die Grundlage für die gesamte Telekommunikation bildet.

1. International Standardization Organisation
 2. Open Systems Interconnection

3.5 Referenzmodelle für Telekommunikationssysteme

Komplexe Telekommunikationssysteme werden durch Modelle aus übereinander angeordneten Schichten (*Layern*) beschrieben. Jede dieser Schichten übernimmt in dem Gesamtvorgang des Nachrichtenaustausches spezielle Teilaufgaben (*Dienste*), die von der ITU in der Recommendation X.200 [X200] festgelegt sind.

3.5.1 Das OSI-7-Schichtenmodell

Das ursprüngliche von der ITU in Zusammenarbeit mit der ISO entwickelte Modell besteht aus insgesamt sieben Schichten, wie es Abbildung 28 zeigt :

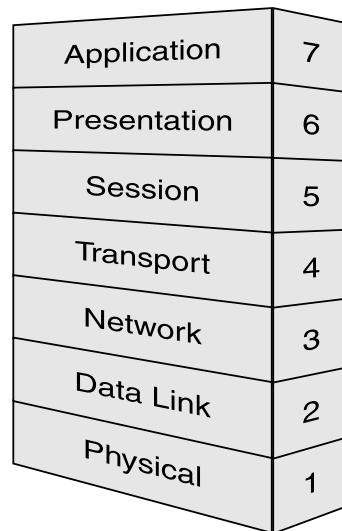


Abbildung 28 : Das OSI-7-Schichtenmodell

Die Aufgabe jedes Layers besteht darin, von dem unter ihm liegenden Layer Dienste anzufordern, diesen seine eigenen Dienste hinzuzufügen und dem über ihm liegenden Layer zur Verfügung zu stellen. Im einzelnen haben die sieben Layer folgende Aufgaben:

Layer	Nummer	Dienst
Physical	(1)	Transparente Übertragung von Bits über das Übertragungsmedium
Data Link	(2)	Absicherung des Datentransports über eine Teilstrecke, Steuerung des Zugriffs auf das Übertragungsmedium
Network	(3)	Wegsuche, Übertragung der Daten zwischen Endsystemen

Tabelle 17 : Aufgaben der Layer im OSI-Modell

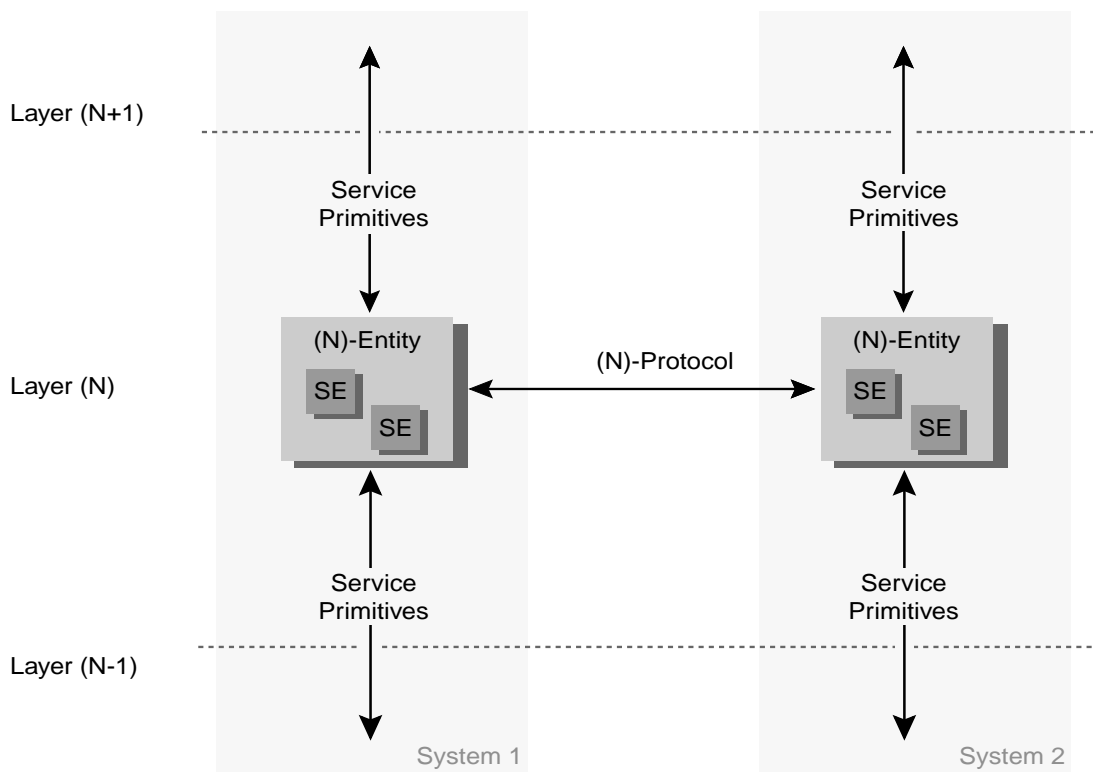
Layer	Nummer	Dienst
Transport	(4)	Flußsteuerung, Sicherung der Datenqualität
Session	(5)	Eröffnung, Steuerung, Beendigung und Wiederaufnahme der Sitzungen
Presentation	(6)	Festlegung der Datensyntax, Datenstrukturen und Konversionsregeln
Application	(7)	Schnittstellenfunktionen für netzwerkgestützte Anwendungen

Tabelle 17 : Aufgaben der Layer im OSI-Modell

3.5.2 Terminologie des OSI-Modells

3.5.2.1 Protokollinstanzen

Die Aufgaben der Schichten werden von den darin befindlichen Protokollinstanzen (*Protocol Entities*) wahrgenommen. Abbildung 29 zeigt die wichtigsten Begriffe im Zusammenhang mit Protokollinstanzen innerhalb des OSI-Modells.



SE : Service Elements

Abbildung 29 : Protokollinstanzen

Eine Protokollinstanz stellt eine Sammlung von Diensten dar, die in der Lage ist, Informationen zu empfangen und zu versenden. Diese Dienste wiederum bestehen aus einer Reihe einzelner Dienstelemente (*Service Elements*).

Eine Protokollinstanz in Schicht (N) kommuniziert mit den (N-1)- und (N+1)-Instanzen mittels Austausch von Dienstanforderungselementen (*Service Primitives*). Ein solches *Service Primitive* benachrichtigt die (N)-Instanz über die Erbringung eines Dienstes von einer (N-1)-Instanz, oder veranlaßt sie dazu, der über ihr liegenden (N+1)-Instanz einen Dienst zu erbringen. *Service Primitive* sind nur in ihrer Semantik festgelegt, alle zu übergebenden Daten und die Bedeutung der Befehle sind standardisiert, nicht jedoch deren Syntax oder die Realisierung.

Um den geforderten Dienst zu erbringen, muß eine Instanz mit ihrer Partnerinstanz (*Peer Entity*) auf einer entfernten Station Nachrichten austauschen. Dazu bedient sie sich einer formalen Sprache, dem in Semantik und Syntax exakt definierten Schicht (N)-Protokoll (*(N)-Protocol*).

3.5.2.2 Dienstzugangspunkte

Ein Dienstzugangspunkt (*Service Access Point SAP*) definiert den Übergabepunkt der *Service Primitives* zwischen den unter Umständen mehrfach vorhandenen *Protocol Entities* in benachbarten Layern innerhalb einer Station (siehe Abbildung 30).

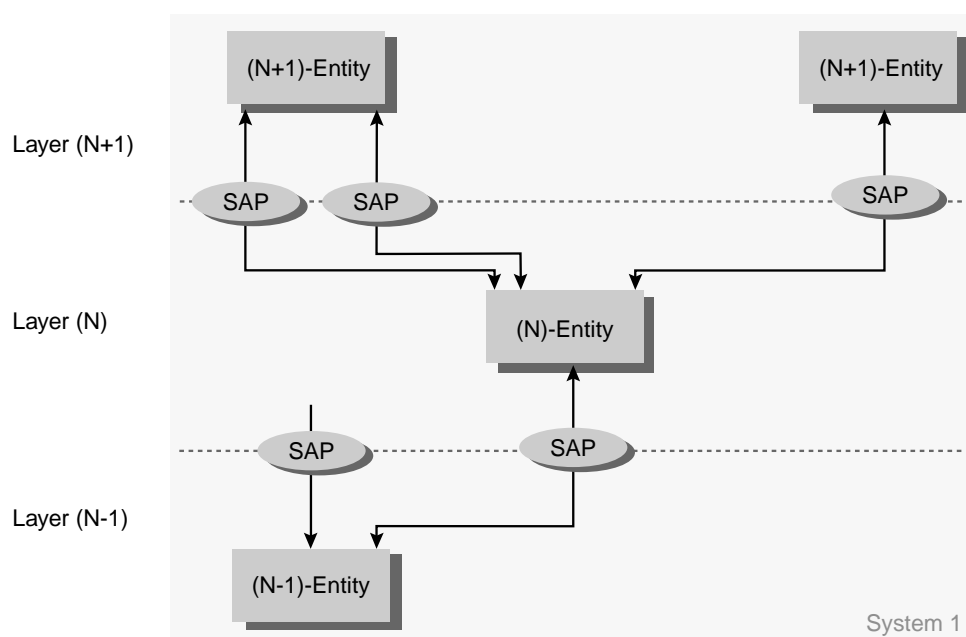


Abbildung 30 : Service Access Points

Jeder SAP verbindet zu einem Zeitpunkt genau zwei Entities miteinander, jede Entity kann jedoch mehrere SAPs gleichzeitig unterhalten.

3.5.2.3 Verbindungsendpunkte

Über jeden SAP können mehrere logische Verbindungen zwischen *Peer Entities* bestehen, denen jeweils ein Verbindungsendpunkt (*Connection Endpoint CEP*) zugeordnet ist. Multiplex und Demultiplex der logischen Verbindungen ist in jeder Entity möglich (vgl. Abbildung 31).

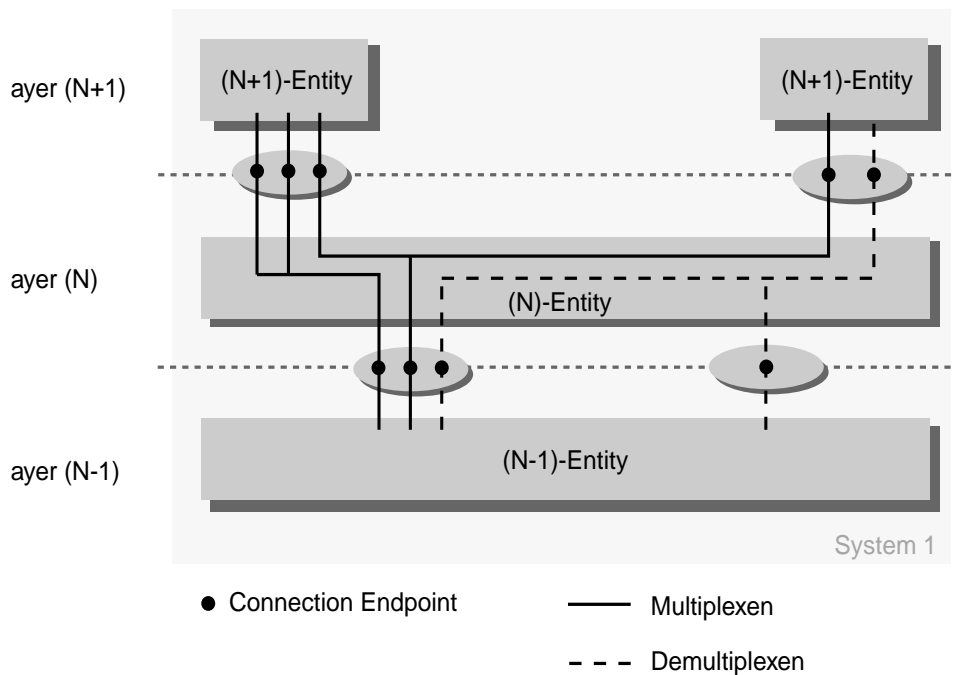


Abbildung 31 : Multiplex und Demultiplex logischer Verbindungen

3.5.2.4 Dienstanforderungselemente

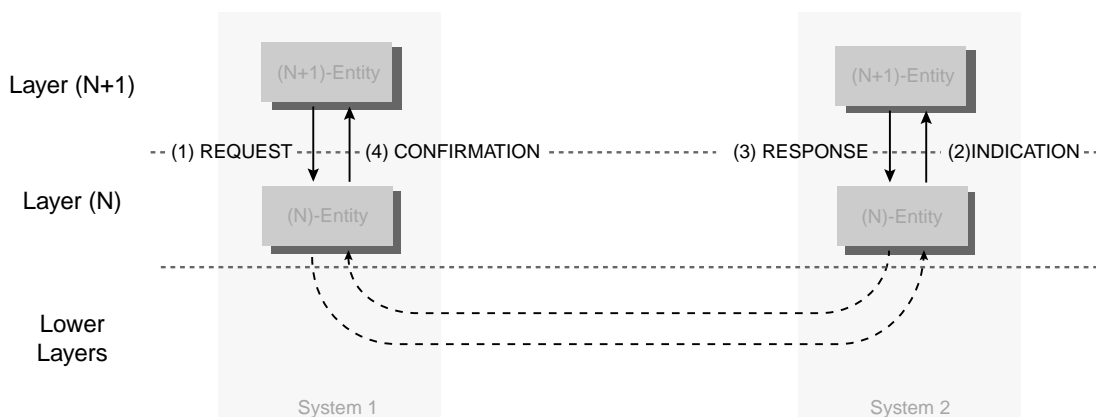


Abbildung 32 : Abfolge der Service Primitives bei einem „Confirmed Service“

Zur Kommunikation zwischen benachbarten Entities existieren insgesamt vier unterschiedliche Typen von Dienstanforderungselementen: *REQUEST* (Anfrage), *INDICATION* (Anzeige), *RESPONSE* (Antwort) und *CONFIRMATION* (Bestätigung). Wird ein Dienst mit Rückmeldung (*Confirmed Service*) verlangt, sind alle vier Primitives beteiligt, in andere Fällen müssen nicht alle auftreten. Die Abfolge der Primitives zeigt Abbildung 32 auf Seite 44.

3.5.2.5 Datenfluß zwischen Partner-Protokollinstanzen

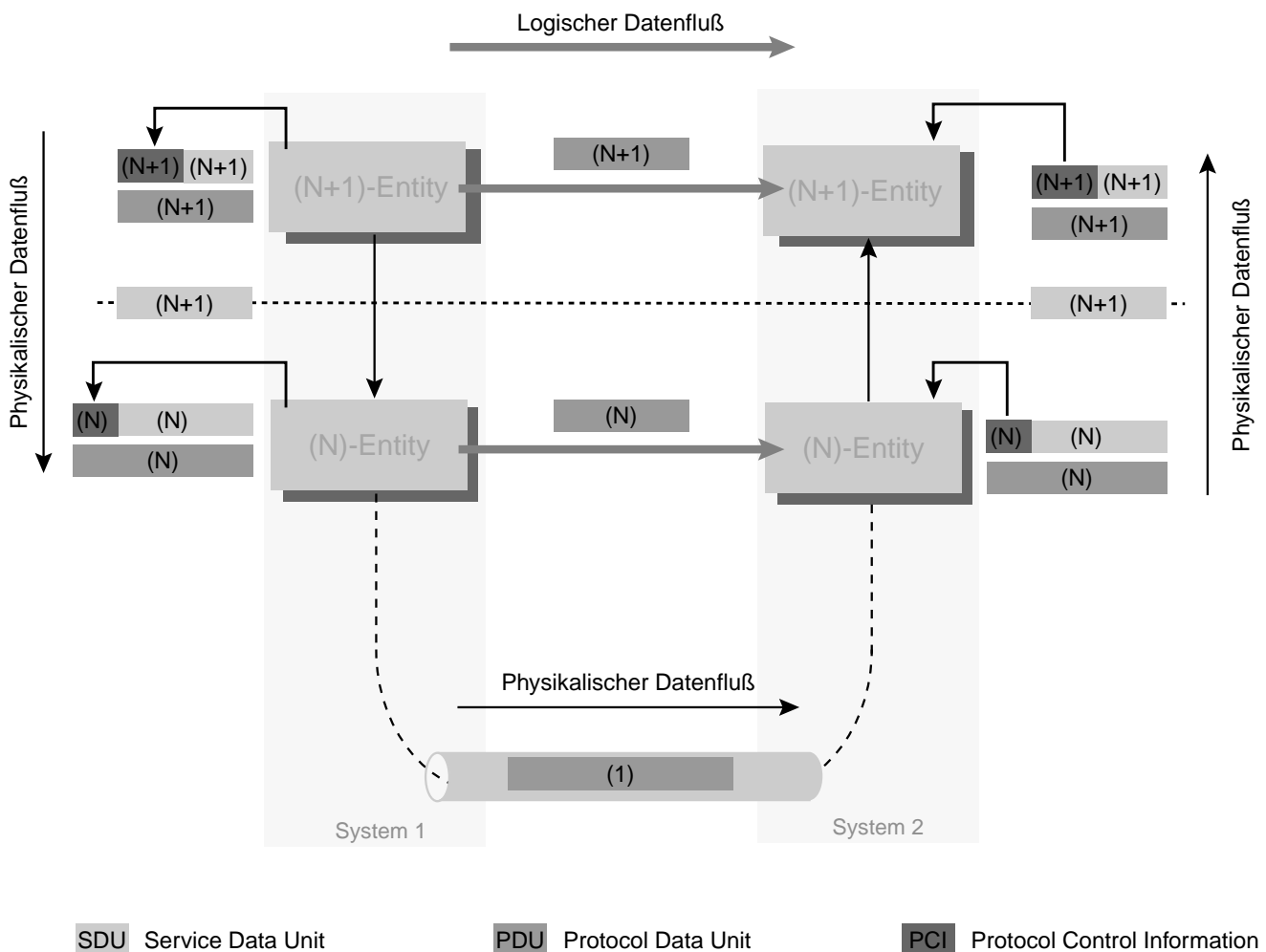


Abbildung 33 : Datenfluß im OSI-Modell

Da zwischen den (*N*)-*Protocol Entities*, abgesehen von denen des Physical Layers, keine direkte Verbindung besteht, müssen die im Rahmen des (*N*)-*Protocols* mit der *Peer Entity* auszutauschenden Daten (*Protocol Data Units PDU*) von den Instanzen der unterlagerten Schichten transportiert werden. Die *PDU*s werden beim Sendevorgang in jeder Schicht um Steuerinformationen (*Protocol Control Information PCI*) ergänzt und als Nutzdatenfeld (*Service Data Unit SDU*) an die unterlagerte

Schicht abgegeben. Beim Empfang wird die Steuerinformation abgetrennt, um das Schichtenprotokoll abzuwickeln.

3.5.3 Das ATM-Referenzmodell

Das in Abbildung 34 gezeigte Schichtenmodell für das B-ISDN gem. I.321 [I321] basiert auf den Prinzipien des OSI-Referenzmodells (vgl. Abbildung 28 auf Seite 41). Es erweitert das ursprüngliche Modell lediglich um einige ATM-spezifische Zusätze, die sich nicht bestimmten Layern im OSI-Referenzmodell zuordnen lassen.

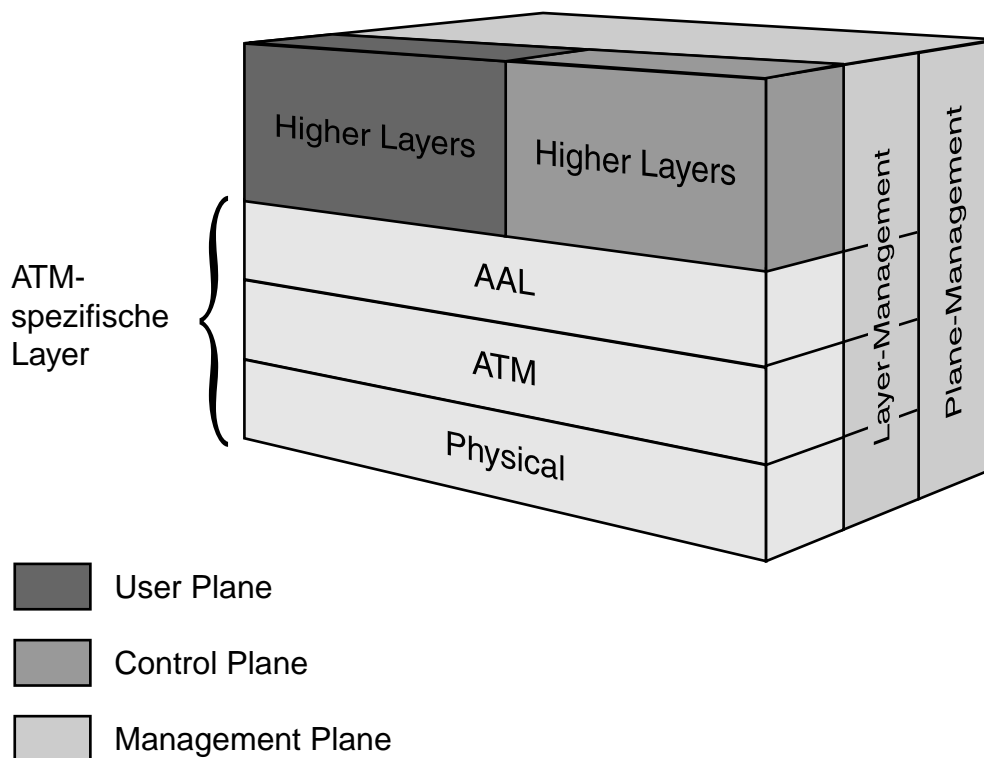


Abbildung 34 : ATM-Referenzmodell

Die neu eingeführten Layer „ATM“ und „AAL¹“ umfassen die ATM-spezifischen Funktionen im B-ISDN. Alle Dienste, die zur Übertragung der Nutzdaten in Form einheitlicher Zellen notwendig sind, stellt der ATM-Layer zur Verfügung. Innerhalb des AAL-Layers erfolgt die Umsetzung der Nutzerdaten in das einheitliche Zellformat. Zusätzlich zu den ursprünglichen Funktionen des Physical Layers im OSI-Referenzmodell hat der Physical Layer hier weitere Aufgaben. Alle drei Layer werden in den folgenden Abschnitten ausführlicher behandelt.

1. ATM Adaption Layer

Im Gegensatz zum ursprünglichen OSI-Referenzmodell weist das ATM-Referenzmodell drei Kommunikationsebenen (*Planes*) auf. Die Nutzerebene (*User Plane*) stellt den Anwendungen alle Funktionen zum Transport ihrer Nutzinformationen zur Verfügung. Die Steuerebene (*Control Plane*) dient dem Aufbau, der Unterhaltung und dem Abbau der Nutzdatenverbindungen. Die Managementebene (*Management Plane*) gliedert sich in zwei Bereiche: Ebenenmanagement (*Plane Management*) und Schichtenmanagement (*Layer Management*). Das Plane Management ist verantwortlich für die Koordination aller Ebenen und übernimmt koordinierende Aufgaben im Netz. Das Layermanagement steuert die Protokollinstanzen in den einzelnen Schichten und übernimmt Aufgaben aus den Bereichen Verwaltung und Überwachung OAM¹.

Tabelle 18 zeigt die Aufgabenverteilung unter den Instanzen der Layer des ATM-Referenzmodells:

Management	Higher layer functions	Higher layers	
	Convergence	CS	AAL
	Segmentation and Reassembly	SAR	
	Generic Flow Control Cell Header Generation/extraction Cell VPI/VCI translation Cell multiplex and demultiplex	ATM	
	Cell rate decoupling HEC header sequence generation/verification Cell delineation and scrambling Transmission frame adaption Transmission frame generation/recovery	TC	PL
	Bit timing Physical medium	PM	

Tabelle 18 : Funktionen der Schichten im B-ISDN-Referenzmodell gem. [I321], Figure 2/I.321

Die zum Verständnis der nächsten Kapitel wichtigsten Funktionen werden in den folgenden Abschnitten erläutert. Eine vollständige Beschreibung kann den in den Abschnitten jeweils aufgeführten Standards der ITU entnommen werden.

1. Operation, Administration and Maintenance

3.6 Physical Layer

3.6.1 Referenzkonfiguration

Die Definitionen der physikalischen Schnittstelle beziehen sich auf die Übergabepunkte S_B und T_B der B-ISDN-Referenzkonfiguration (siehe Abbildung 35 auf Seite 48) des UNI, in der die Anschlußmöglichkeiten von Endgeräten an das B-ISDN festgelegt sind. Endgeräte mit einer B-ISDN-Schnittstelle werden direkt am Referenzpunkt S_B angeschlossen, Endgeräte ohne diese Schnittstelle müssen über Terminaladapter angepaßt werden.

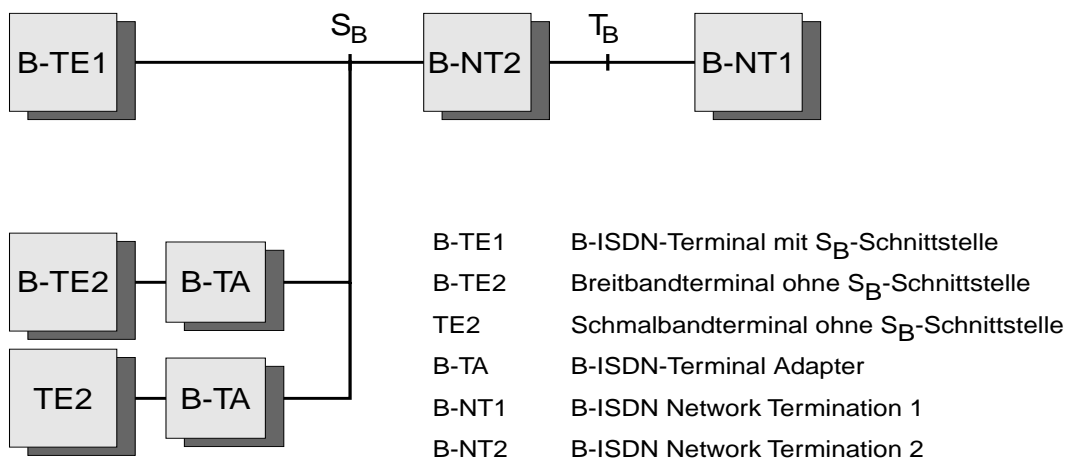


Abbildung 35 : Referenzkonfiguration am UNI

Der Network Terminator B-NT1 bildet den Abschluß des Übertragungsnetzwerks, B-NT2 enthält die Nebenstellenfunktionalitäten innerhalb größerer Anlagen. Der Standardisierung unterworfen sind ausschließlich die Übergabepunkte S_B und T_B ; hier sind zwei Schnittstellen definiert: 155 MBit/s und 622 MBit/s.

3.6.2 Physical Medium Sublayer

Aufgabe des *PM-Sublayers* ist die Anpassung der Bits an die physikalischen Gegebenheiten der Übertragungsstrecke.

Parameter	Schnittstelle 155 MBit/s	Schnittstelle 622 MBit/s
Bitrate	155.520 kBit/s	622.080 kBit/s
Verkabelungsart	sternförmig	sternförmig

Tabelle 19 : Physikalische Eigenschaften der Schnittstelle T_B

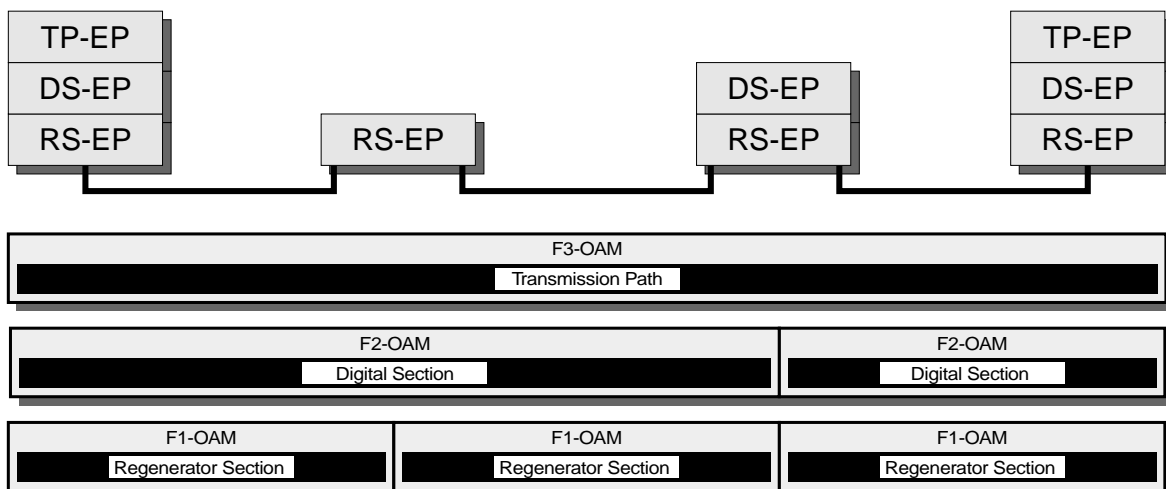
Parameter	Schnittstelle 155 MBit/s	Schnittstelle 622 MBit/s
Symmetrie	Symmetrisch	Symmetrisch oder Asymmetrisch (622/155 MBit/s)
Übertragungsmedium elektrisch	Koaxialkabel, $75 \Omega \pm 5\%$ von 50- 200 MHz	ffs ^a .
Signalform	Tabelle 11/G.703 Fig. 24,25 /G.703	ffs.
Leitungscode	CMI 12.1/G.703	ffs.
Übertragungsmedium optisch	Monomodenfaser gem. [G652]	Monomodenfaser gem. [G652]
Leitungscode	NRZ ^b	NRZ
Wellenlänge	1310 nm	1310 nm

Tabelle 19 : Physikalische Eigenschaften der Schnittstelle T_B

- a. for further studies
b. Non Return to Zero

3.6.2.1 Transmission Convergence Sublayer

Wie im ATM-Layer (siehe Abbildung 42 auf Seite 58) besteht der Übertragungsweg auch im Physical Layer aus aneinander anschließenden Teilstrecken:



TP-EP: Transmission Path Endpoint RS-EP: Regenerator Section Endpoint
DS-EP: Digital Section Endpoint

Abbildung 36 : Teilstrecken im Physical Layer mit zugeordneten OAM-Flüssen

Die *Regenerator Section* erstreckt sich zwischen zwei Signalregeneratoren, die *Digital Section* zwischen den Endpunkten eines untergeordneten digitalen Übertragungssystems (z.B. SDH), der *Transmission Path* erstreckt sich zwischen zwei Netzelementen, an denen Instanzen des ATM-Layers vorhanden sind, und die Zellen den Physical Layer verlassen können.

3.6.2.2 Cell Rate Decoupling

Die Nutzdatenzellrate wird von der Zellrate auf dem Übertragungsmedium entkoppelt (Abbildung 20 auf Seite 34).

3.6.2.3 HEC generation/verification

Die ersten vier Bytes des Zellheaders werden durch einen zyklischen Code mit dem Generatorpolynom $P(x) = x^8 + x^2 + x + 1$ geschützt. Der Empfänger ist hiermit in der Lage, Einzelbitfehler zu korrigieren und Mehrfachbitfehler zu erkennen (vgl. die in [I432], Abschnitt 4.3 beschriebenen Fehlerbehandlungsroutine).

3.6.2.4 Cell Delineation and Scrambling

Der Prozeß der Zelldelineation dient dem Erkennen der Zellgrenzen im Bitstrom. Zum ersten Auffinden der Zellgrenzen wird der Datenstrom Bit für Bit untersucht, bis fünf Bytes gefunden sind, wobei das letzte mit der nach der Bildungsvorschrift für die HEC über die Bytes 1 bis 4 übereinstimmen muß. Anschließend wird die Übereinstimmung nur noch im Abstand von jeweils 53 Bytes geprüft (vgl. Fig.13/ I.432).

3.6.2.5 Transmission Frame Adaption / Transmission Frame Generation/Recovery

Dem *TC-Sublayer* obliegt die Anpassung des Bitstromes an einen Übertragungsrahmen (*Transmission Frame Adaption am DS-EP*). Zwei Möglichkeiten, die *zellbasierte* und die *SDH-basierte Schnittstelle*, sind hier vorgesehen:

Zellbasierte Schnittstelle

Das Datenformat an dieser Schnittstelle besteht aus einem kontinuierlichen Zellstrom, der vom TC-Sublayer durch das Einfügen von *Idle-Cells* (Leierzellen) aufrechterhalten wird. Steuerung und Überwachung der Übertragungsabschnitte geschieht über *OAM-Cells*, die in den Nutzdatenstrom eingefügt

werden und gegenüber dieser Priorität genießen. Durchgeführt werden die OAM-Funktionen in den jeweiligen Endpunkten der Teilstrecken des Übertragungsweges (siehe Abbildung 36 auf Seite 49).

An der zellbasierten Schnittstelle existiert kein untergeordnetes Übertragungssystem, daher entfällt hier die Funktionalität auf der Ebene der *Digital Section*. Die verbleibenden OAM-Flüsse F1 und F3 überwachen die Nutzzellen auf Übertragungsfehler und benachrichtigen das Layer-Management sowie ihre Pendanten beim Auftreten von Übertragungsfehlern, dem Verlust der Zellsynchronisation oder von OAM-Zellen (vgl. [I432.x] und [I610]). Der Payload der OAM-Zellen (Anhang A.1 auf Seite 178) trägt detaillierte Informationen über die festgestellte Bitfehler auf der Übertragungsstrecke sowie Angaben über Fehlerzustände, die von den benachbarten OAM-Managern festgestellt wurden. Die Aufgabe der Instanzen in den Managementebenen ist es, entsprechende Gegenmaßnahmen einzuleiten (z.B. Schaltung eines Ersatzpfades).

Schnittstelle zur Synchronen Digitalen Hierarchie SDH

Die Definition der SDH-Schnittstelle im B-ISDN definiert das *Mapping* der ATM-Zellen in die Übertragungsrahmen der Synchronen Digitalen Hierarchie, wie sie in den ITU-Empfehlungen G.707, G708 und G.709 (siehe [G707], [G708] und [G709]) definiert sind.

Ein gravierender Nachteil der PDH (vgl. Abschnitt 1.1.1 auf Seite 1) ist die fehlende Möglichkeit, in dem Signal einer hohen Multiplexstufe direkt auf die Basiskanäle zugreifen zu können. Der Zugriff auf ein solches Signal erfordert es, die komplette Multiplexhierarchie in umgekehrter Reihenfolge zu durchlaufen, ein Vorgehen, das zu Netzen mit starrem und unflexiblem Aufbau führt. Eine Verbesserung wurde erst durch die Entwicklung des *SONET*¹ durch die Firma Bellcore, in Zusammenarbeit mit dem Standardisierungsgremium ANSI, in den USA erreicht. Die ITU vereinigte die Festlegungen im *ANSI-SONET* und dessen europäischen Variante *ETSI-SONET* unter der Bezeichnung *Synchrone Digitale Hierarchie (SDH)*.

Die SDH stellt eine Hierarchie von Übertragungsrahmen $STM-n^2$ zur Verfügung, die im Abstand von jeweils 125 μ s aufeinander folgen. Der Rahmenaufbau in den höheren Multiplexebenen erlaubt über Einsprunganadressen (Pointer) den direkten Zugriff auf alle enthaltenen Subrahmen, ohne den Übertra-

1. Synchronous Optical Network
2. Synchrones Transportmodul der Stufe n

gungsrahmen aufzulösen. Die Basisübertragungsrate STM-1 in der SDH wurde mit 155.520 kBit/s festgelegt, die höheren Raten sind ganzzahlige Vielfache dieser Grundbitrate (siehe Tabelle 20).

SDH-Bezeichnung	Bitrate [MBit/s]
STM-1	155,520
STM-3	466,560
STM-4	622,080
STM-6	933,120
STM-8	1.244,160
STM-12	1.866,240
STM-16	2.488,320
STM-64	9.953,280

Tabelle 20 : Übertragungsraten innerhalb der SDH

Die kleinste Einheit beim Aufbau eines STM-n ist der *Container (C)*. Zusammen mit den Steuerinformationen im *Path Overhead (POH)* bildet er einen *Virtuellen Container (VC)*. Mehrere VC können zu einer *Tributary Unit Group (TUG)* zusammengefaßt werden, um in einem VC höherer Ordnung transportiert zu werden. Abbildung 37 auf Seite 53 zeigt zwei der vielfältigen Möglichkeiten ein STM-1-Signal zu erzeugen; im Standard G.709 [G709] sind zahlreiche weitere Möglichkeiten angegeben, ein STM-n aus Signalen niedrigerer Bitraten aufzubauen. An dieser Stelle soll nur der Aufbau der für die Übertragung im B-ISDN relevanten Transportmodule STM-1 und STM-4 genauer betrachtet werden. Im rechten Teil der Abbildung 37 auf Seite 53 ist gezeigt, wie 2,048 MBit/s-Kanäle aus der PDH in einem C-12, über mehrere Zwischenstufen hinweg, zu einem STM-1 zusammengesetzt werden. Im linken Teil der Abbildung werden ATM-Zellen in den Payload eines VC-4-Containers eingesetzt. Sie bilden zusammen mit dem VC-4-POH die Nutzlast eines Synchronen Transportmoduls der Ebene 1, welches zum Transport von Zellen an der 155 MBit/s-Schnittstelle eingesetzt wird. Dieses Modul wird im SDH-Netz geschlossen bis zum nächsten Digital Section Endpoint übertragen, wo die Zellen wieder dem STM-1-Rahmen entnommen werden.

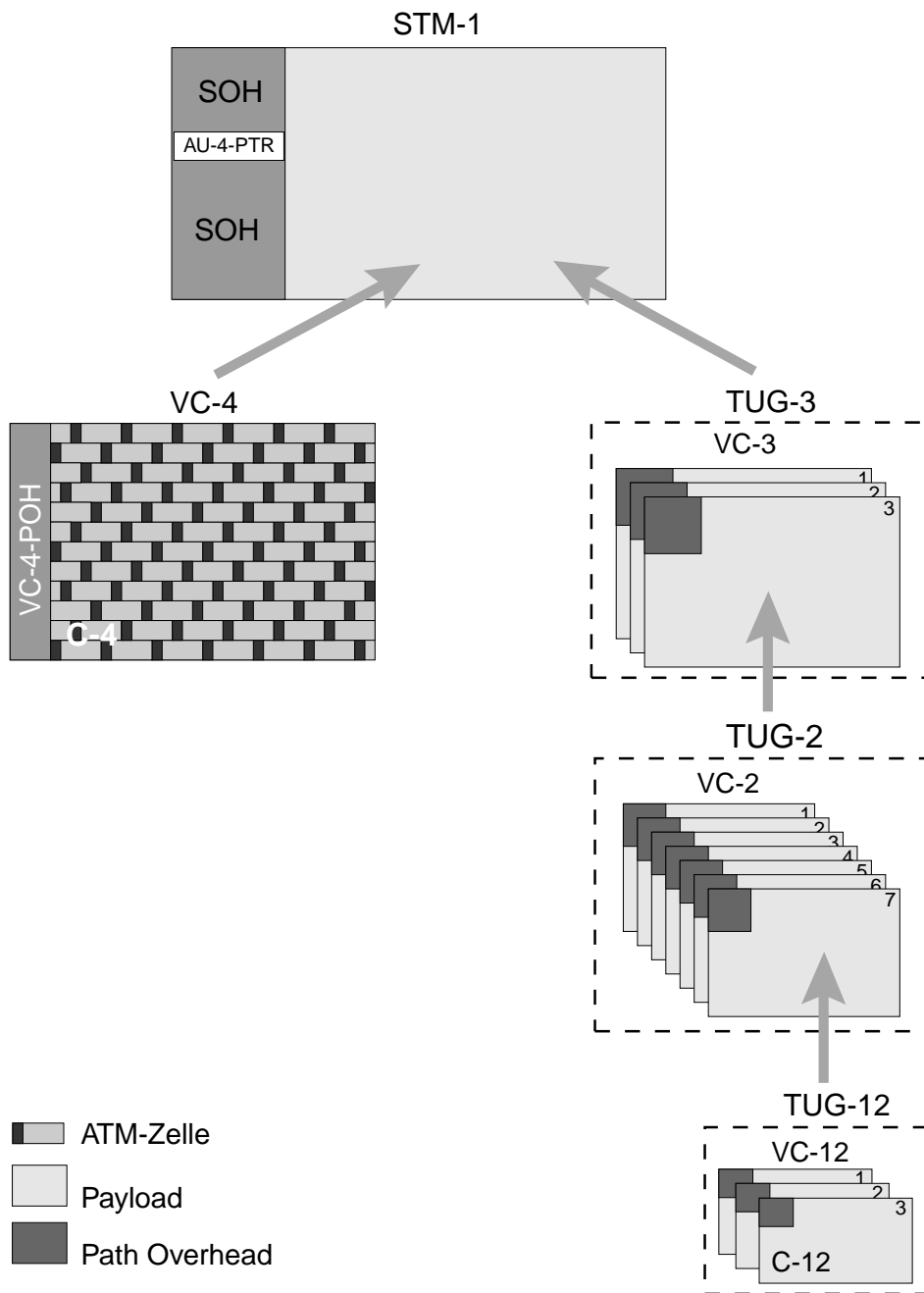


Abbildung 37 : Aufbau eines STM-1 (Bsp.)

Die Rahmenstruktur des STM-1 am UNI zeigt Abbildung 38 auf Seite 54. Die Positionen im *Section Overhead SOH* dienen der Verkehrslenkung und Qualitätsüberwachung im SDH-Netz. Der AU-4-Pointer beschreibt die Position des ersten Bytes der Nutzlast VC-4 im STM-1-Rahmen. Da die Kapazität des C-4 (2340 Oktetts) kein ganzzahliges Vielfaches der Zellgröße von 53 Bytes ist, variiert der Beginn der Zellen innerhalb des VC-4-Payload.

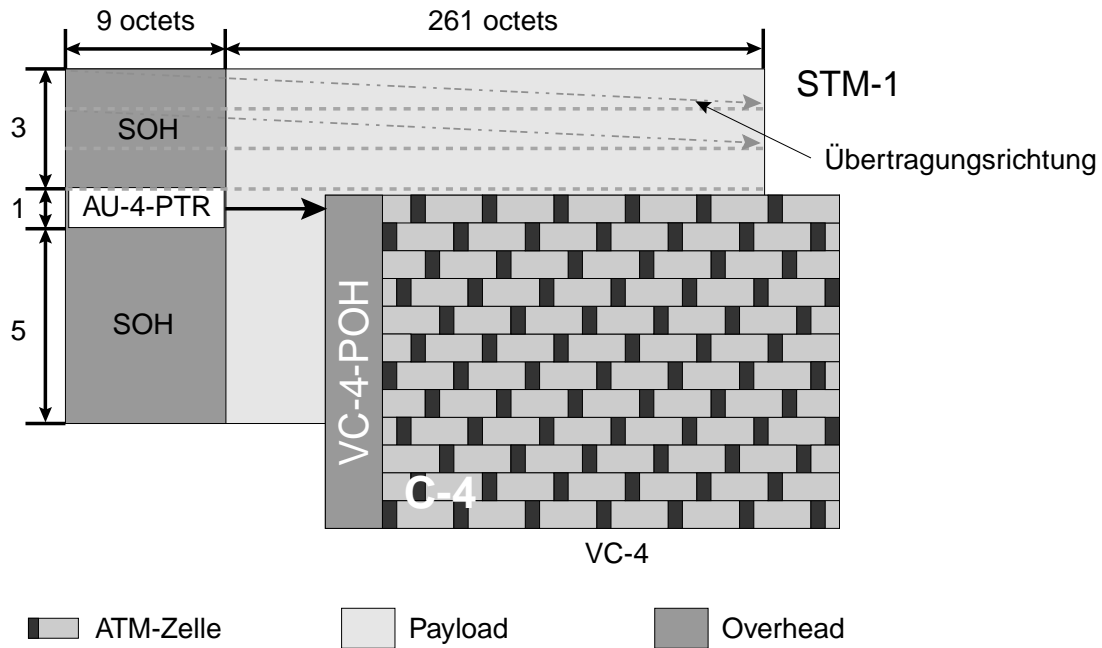


Abbildung 38 : Rahmenstruktur am SDH-basierten UNI 155 MBit/s

An der 622 MBit/s-Schnittstelle werden die Zellen direkt in das Nutzlastfeld eines VC-4-4c-Containers eingebracht und bilden mit dem VC4-4-4c Path Overhead einen STM-4-Rahmen:

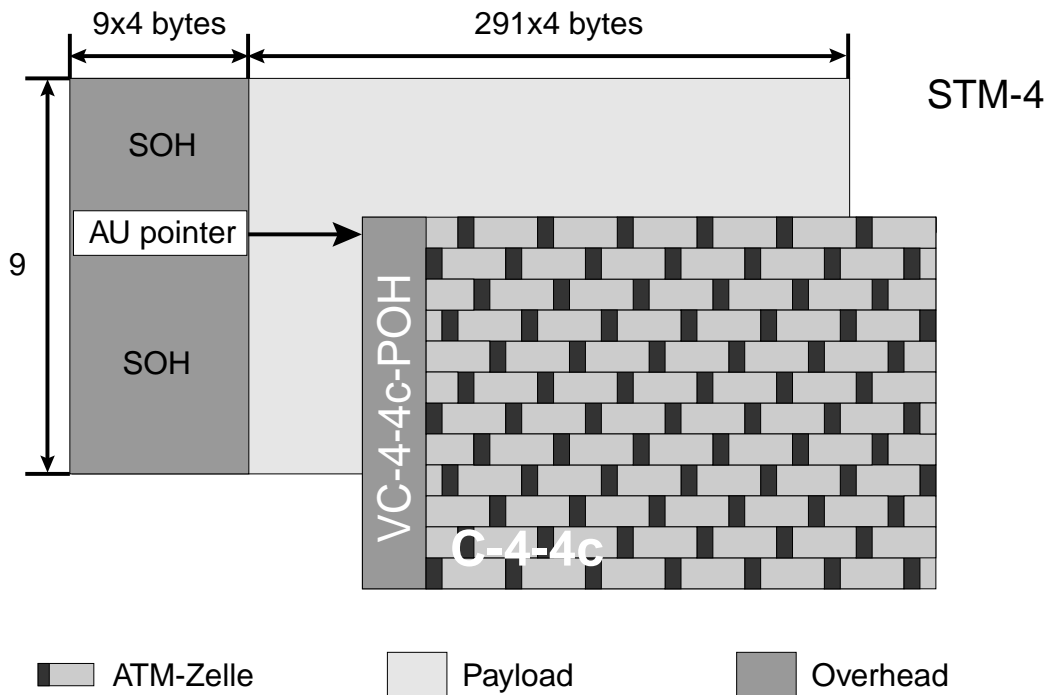


Abbildung 39 : Rahmenstruktur am SDH-basierten UNI 622 MBit/s

Die höheren Transportmodule werden durch Multiplex innerhalb des SDH-Übertragungsnetzwerks erzeugt (siehe Abbildung 40).

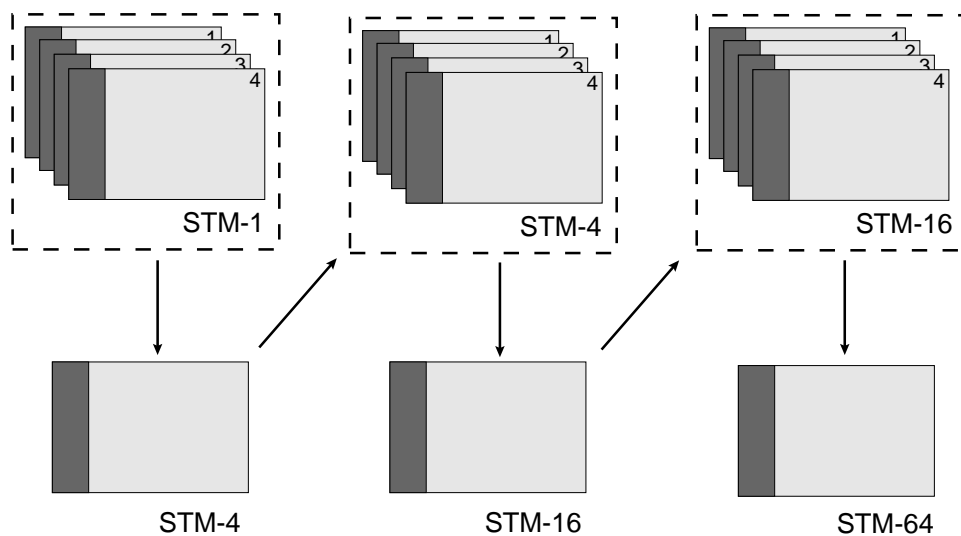


Abbildung 40 : Transportmodule STM-n

Die Oktetts im SOH und POH der Virtuellen Container dienen der Durchführung der OAM-Operationen; die in den SDH-Spezifikationen [G708] und [G709] definierte Funktionalität ist für das ATM-UNI zu übernehmen. Zusätzlich zu den OAM-Flüssen F1 und F3 der zellbasierten Schnittstelle existiert hier ein OAM-Fluß F2, der sich auf den Übertragungsabschnitt zwischen zwei SDH-Netzwerkelementen bezieht.

3.6.2.6 Dienstanforderungselemente¹

Die Service Primitives, welche über den SAP zwischen ATM-Layer und Physical Layer übergeben werden, zeigt die folgende Tabelle:

Service-Primitive	Bedeutung
PHY_DATA_REQUEST	Befehl zum Senden einer ATM-Zelle
PHY_DATA_INDICATION	Übergabe einer Zelle an die ATM-Protokollinstanz

Tabelle 21 : Serviceprimitive zwischen ATM-Layer und Physical Layer

1. Die Auflistung der Parameter aller Dienstanforderungselemente sowie eine diesbezügliche Erläuterung ist in Anhang B auf Seite 187 ff. zu finden.

3.7 ATM-Layer

Die folgende Abbildung 41 zeigt den schematischen Aufbau des ATM-Layers

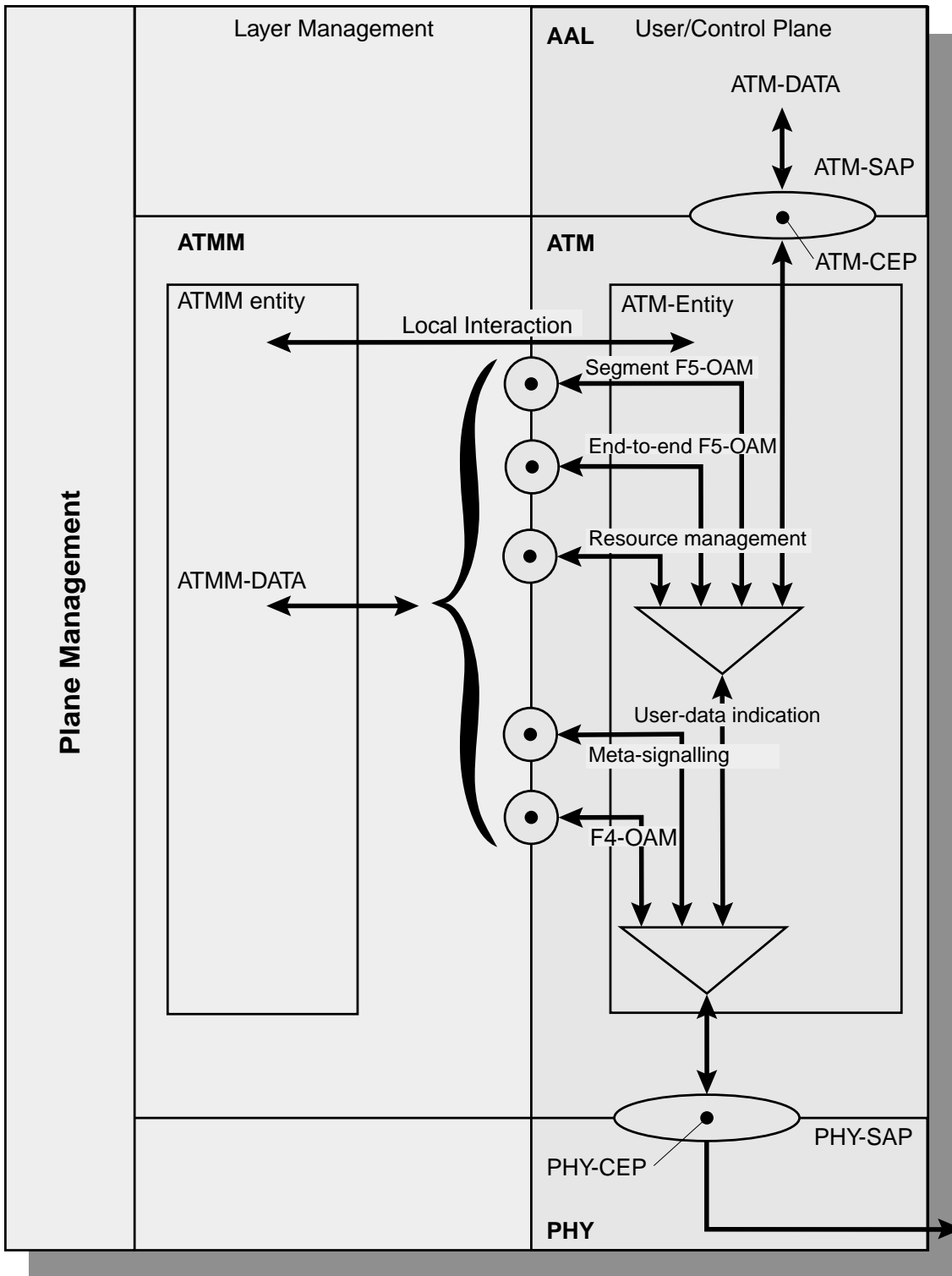


Abbildung 41 : Schematischer Aufbau des ATM-Layers gem. I.361 [I361]

3.7.1 Funktionen der Protokollinstanzen im ATM-Layer

Tabelle 22 zeigt als Übersicht die von der ATM- und ATM-Layermanagement-Entity zu erbringenden Dienste.

Dienst
Multiplex und Vermittlung Virtueller Verbindungen und Pfade
Erkennen der Zellverlustpriorität und selektives Verwerfen von Zellen
Netzlastformung
Analyse der Zelltypen anhand des Zellheaders
Verarbeitung von OAM- und Ressource-Management-Zellen
Abwicklung des Metasignalisierung
Überwachung der Verkehrsparameter der Verbindungen
Auf- und Abbau von Nutzdatenverbindungen

Tabelle 22 : Dienste des ATM-Layers

3.7.1.1 Multiplex und Vermittlung Virtueller Verbindungen und Pfade

Eine der wichtigsten Aufgaben einer ATM-Entity ist die Bereitstellung der Vermittlungsfunktion auf Kanal- und Pfadenebene (vgl. Abschnitt 3.3 auf Seite 38). Die an einem vermittelnden Netzknoten ankommenden Zellen werden entsprechend den beim Verbindungsaufbau gefüllten *Routing Tables* weitergesendet. Innerhalb der B-TE¹ führt die Protokollinstanz den statistischen Multiplex aller Virtuellen Verbindungen durch.

3.7.1.2 Erkennen der Zellverlustpriorität und selektives Verwerfen von Zellen

Treten bei der Vermittlung der Zellen Engpässe in einem Netzknoten auf (z.B. drohender Pufferüberlauf) ist es die Aufgabe der ATM-Entity, zunächst die Zellen niedriger Priorität, anschließend auch die mit hoher Priorität zu verwerfen, um einer Überflutung des dahinterliegenden Netzwerkes vorzubeugen.

3.7.1.3 Netzlastformung und Überwachung der Verbindungen

Eine präventive Maßnahme, bevor es zum Verwerfen von Zellen kommt, ist die Netzlastformung (*Traffic Shaping*). Hierbei werden Lastspitzen aus den ankommenden Datenströmen aufgefangen.

1. Broadband Terminal Equipment (Breitbandiges Endgerät mit ATM-Schnittstelle)

Solche Spitzen entstehen entweder durch momentane Überlastsituationen im Netz oder durch Anwender, welche die ihnen zugestandenen Zellraten überschreiten. Die Aufgabe der *Usage Parameter Control UPC* bzw. dem *Policing* ist es, solche Überschreitungen aufzuspüren und den Verursacher zu sanktionieren. Zu diesem Zweck werden unterschiedliche Verfahren wie *Leaky Bucket*, *Moving Window* oder *Jumping Window* eingesetzt (siehe [Rath97]).

3.7.1.4 Auf- und Abbau von Nutzdatenverbindungen

Den Daten jeder Virtuellen Verbindung ist eine erforderliche Dienstgüte (*Quality of Service QoS*) zugeordnet. Ausgehend von den noch verfügbaren Ressourcen, der bereits bestehenden Last und den Qualitätsansprüchen der neuen Verbindung entscheidet die *Connection Admission Control CAC* in dem Netzknoten über Annahme oder Verweigerung eines Verbindungswunsches.

3.7.1.5 Erkennen des Zelltyps

Die ATM-Entity analysiert den Header aller eintreffenden Zellen zur Feststellung ihres Typs, der durch VPI, VCI- und PT-Feld eindeutig definiert ist. Die Bedeutung der Werte dieser Felder ist durch Tabelle 2/I.361 in der ITU-Recommendation I.361 [I361] festgelegt.

Verarbeitung von OAM-, Signalling-, und Ressourcenzellen

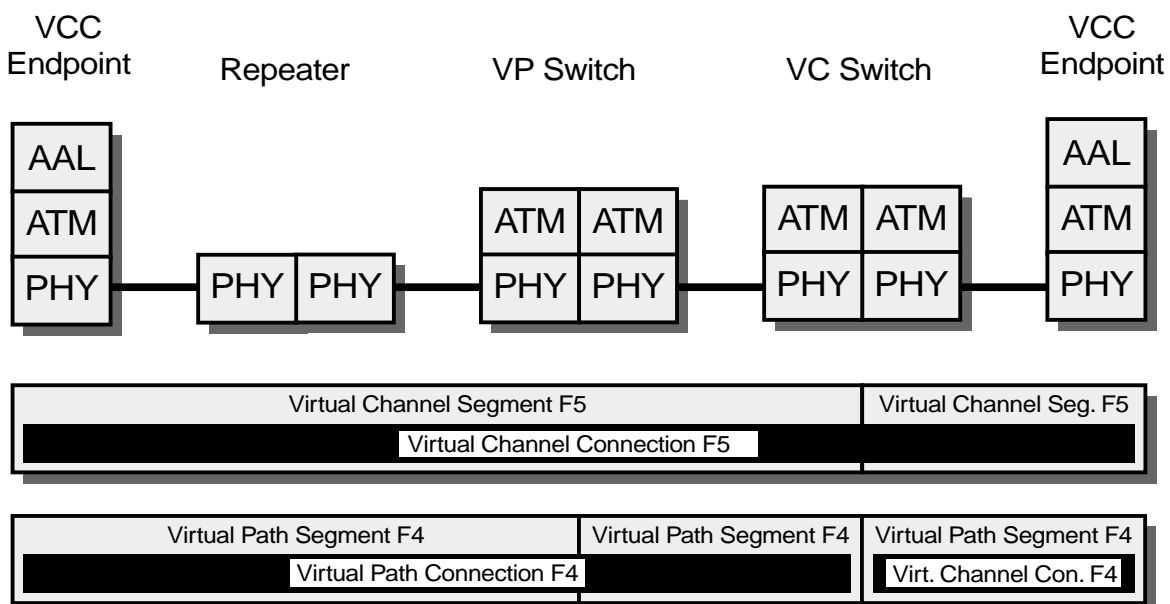


Abbildung 42 : Reichweite der OAM-Flüsse im ATM-Layer

Während die ATM-Entity selbst nur Nutzzellen verarbeitet und sie an den ATM Adaption Layer abgibt, werden OAM-, Signalling- und Ressourcezellen zur Layermanagement-Entity weitergereicht. Auf dem Niveau des ATM-Layers existieren wie im Physical Layer (vgl. Abbildung 36 auf Seite 49) mehrere OAM-Flüsse, die entweder einen Virtuellen Pfad (F4) oder einen Virtuellen Kanal (F5) überwachen (siehe Abbildung 42 auf Seite 58). *Segment-OAM-Zellen* werden am Ende der entsprechenden Teilstrecke entnommen und analysiert, *End-to-end OAM-Zellen* werden erst am Ende der Virtuellen Verbindung auf Pfad oder Kanalniveau entfernt und passieren die dazwischenliegenden Vermittlungsstellen unverändert. Die Einrichtung erfolgt zu einem beliebigen Zeitpunkt nach dem Verbindungsaufbau. Eine detaillierte Beschreibung aller Vorgänge ist in der ITU-Recommendation I.610 [I610] zu finden.

Aufgaben der OAM-Flüsse innerhalb des ATM-Layers

- Alarmierung

F4- bzw. *F5-AIS-Zellen* werden von den Fehlerzuständen auf dem Physical Layer ausgelöst. Die Auslösung der *FERF-Zellen* erfolgt gemäß einem ähnlichen Verfahren wie auf dem Physical Layer (vgl. hierzu I.610/Sec. 6.2 [I610]).

- Überwachung der Verfügbarkeit eines Pfades oder Kanals

Sendet ein Endpunkt während einer bestimmten Zeitspanne keine Nutzzellen und ist kein Alarm ausgelöst, wird eine *Continuity Check Cell* gesendet, um den Empfänger vom ungerechtfertigten Auslösen einer FERF-Bedingung abzuhalten. Um eine Fehlerquelle im Netz zu lokalisieren besteht die Möglichkeit, *Loopback Cells* in bestehende Zellströme einzubringen. In ihrem Nutzlastfeld ist der Netzknoten vermerkt, der die Zelle zum Absender zurücksenden soll. Auf diese Art kann die Übertragungsstrecke gezielt nach Fehlerstellen abgesucht werden. Ebenfalls einsetzbar sind solche Zellen zum Zweck eines *on-demand delay mesurement* zur Bestimmung der aktuellen Zellverzögerung in einem Netzbereich.

- Überwachung der Leistungsfähigkeit (Performance Monitoring)

Vom Nutzer oder dem Netzwerkmanagement zu einem beliebigen Zeitpunkt ausgelöst besteht die Möglichkeit, Datenströme auf Pfad- und Kanalniveau auf Fehler zu überwachen. Nach Abwicklung eines Handshake-Algorithmus zur Information beider Endpunkte wird eine Überwachung von

Zellblöcken durch das Einfügen von *Error Detection Codes* eingeleitet. Durch diese Maßnahme können Verbindungen mit hohen Fehlerraten frühzeitig erkannt und zu anderen Übertragungswegen umgeleitet werden.

3.7.1.6 Ressource-Management-Zellen

Ressource-Management-Zellen dienen der Zuteilung bzw. Neuzuteilung von Ressourcen an Virtuelle Verbindungen. Auf die Funktionen dieser Zellen im Rahmen der Sicherung der Übertragungsqualität wird in Abschnitt 4.2 auf Seite 96 ff. ausführlicher eingegangen.

3.7.1.7 Metasignalisierung

Signalisierungskanäle, die dem Aufbau der Nutzdatenverbindungen dienen, bedürfen ebenso wie diese eines vorherigen Verbindungsaufbaus. Daher sind im ATM-Netzwerk spezielle Kanäle (in jedem Pfad der Kanal mit VCI=1) festgelegt, die in der Lage sind, auch ohne einen expliziten Verbindungsaufbau einige Zellen (max. 42 Zellen/s) zu transportieren. Über diesen permanent vorhandenen Kanal wird zwischen den *Peer Metasignalling Entities* das relativ einfache Metasignalisierungsprotokoll gem. ITU Recommendation Q.2120 [Q2120] abgewickelt. Die Nachrichten dieses Protokolls zeigt die folgenden Abbildung:

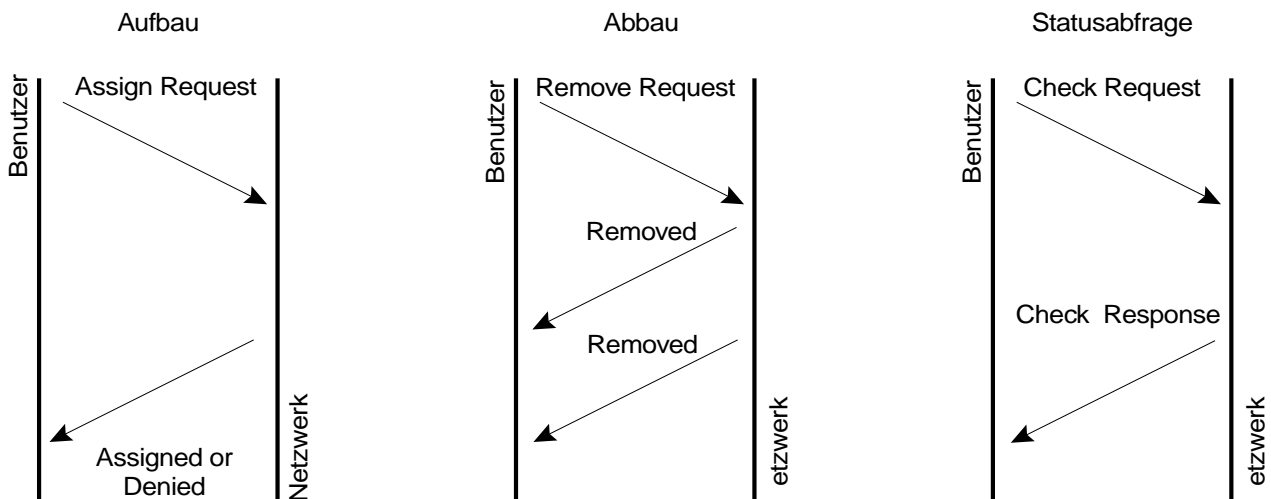


Abbildung 43 : Ablauf der Metasignalisierung

Alle diese Nachrichten werden innerhalb einer einzigen Zelle transportiert, deren Aufbau Tabelle 53 im Anhang auf Seite 179 zeigt. Nachdem mit Hilfe der Metasignalisierung ein Signalisierungskanal eröffnet ist, können die Instanzen der höheren Schichten über diesen die eigentliche Signalisierung zum Aufbau der Nutzdatenkanäle (siehe Kapitel 3.9 auf Seite 90 ff.) abwickeln.

3.7.2 Service Primitives

Alle Dienstanforderungselemente zwischen ATM-Layer und Physical Layer sind bereits in Abschnitt 3.6.2.6 auf Seite 55 aufgeführt. Die zwischen der ATM-Entity und den Instanzen innerhalb der restlichen benachbarten Schichten ausgetauschten Service Primitives zeigen die beiden folgenden Tabellen.

Name	Bedeutung
ATMM_DATA_REQUEST	Befehl zum Senden einer ATM-Zelle vom ATM-Layermanagement
ATMM_DATA_INDICATION	Übergabe einer empfangenen Zelle an die ATM-Layermanagement-Protokollinstanz

Tabelle 23 : Dienstanforderungselemente zwischen ATM-Layer und ATM-Layermanagement

Name	Bedeutung
ATM_DATA_REQUEST	Befehl zum Senden einer ATM-Zelle
ATM_DATA_INDICATION	Übergabe einer empfangenen Zelle an die AAL-Protokollinstanz

Tabelle 24 : Dienstanforderungselemente zwischen ATM-Layer und AAL Layer

3.8 AAL-Layer

Die von den Anwendungen in den höheren Schichten des OSI-Modells an den ATM-Layer abgegebenen Daten stellen unterschiedliche Anforderungen an das Übertragungsnetzwerk; die Aufgabe der Protokollinstanzen im AAL-Layer ist es, diese Datenströme unter Berücksichtigung ihrer besonderen Eigenschaften an die einheitliche Zellübertragung im ATM-Layer anzupassen. Innerhalb des AAL-Layers existieren mehrere verschiedene Ausprägungen der Protokollinstanzen, die jeweils hinsichtlich einer Klasse von Daten mit ähnlichen Eigenschaften optimiert sind:

	Klasse A	Klasse B	Klasse C	Klasse D
Zeitbindung	zeitkontinuierlich		nicht zeitkontinuierlich	
Bitrate	konstant	variabel		
Verbindungsart	verbindungsorientiert			verbindungslos
AAL-Typ	AAL 1	AAL 2	AAL 3/4	
			AAL 5	

Tabelle 25 : Dienst-Klassen und AAL-Typen

Die beiden in Tabelle 25 nicht genannten Typen AAL 0 und S¹-AAL nehmen eine Sonderstellung ein. Der AAL-Typ 0 ist für Anwendungen vorgesehen, die bereits selbst Zellverkehr erzeugen, also keine eigene Adaptionsschicht benötigen. Die S-AAL stellt einen für die Übertragung von Daten der Signalisierungsprotokolle erweiterte und optimierte AAL-Typ 5 dar. Wie die ITU-Recommendation I.363 [I363] festlegt, unterteilt sich der AAL-Layer zunächst grob in zwei Sublayer: *Convergence Sublayer* und *Segmentation and Reassembly Sublayer* (siehe Abbildung 44 auf Seite 63). Die Aufgabenverteilung zwischen den beiden Sublayern ist wie folgt definiert:

Sublayer	Aufgabe
Convergence Sublayer CS	Dienstspezifische Funktionen für die Instanzen in den höheren Schichten
Segmentation and Reassembly Sublayer SAR	Segmentierung der Datenpakete der höheren Schichten, so daß sie vom Nutzdatenfeld einer Zelle aufgenommen werden können

Tabelle 26 : Aufgabenverteilung innerhalb der AAL-Sublayer

1. Signalling

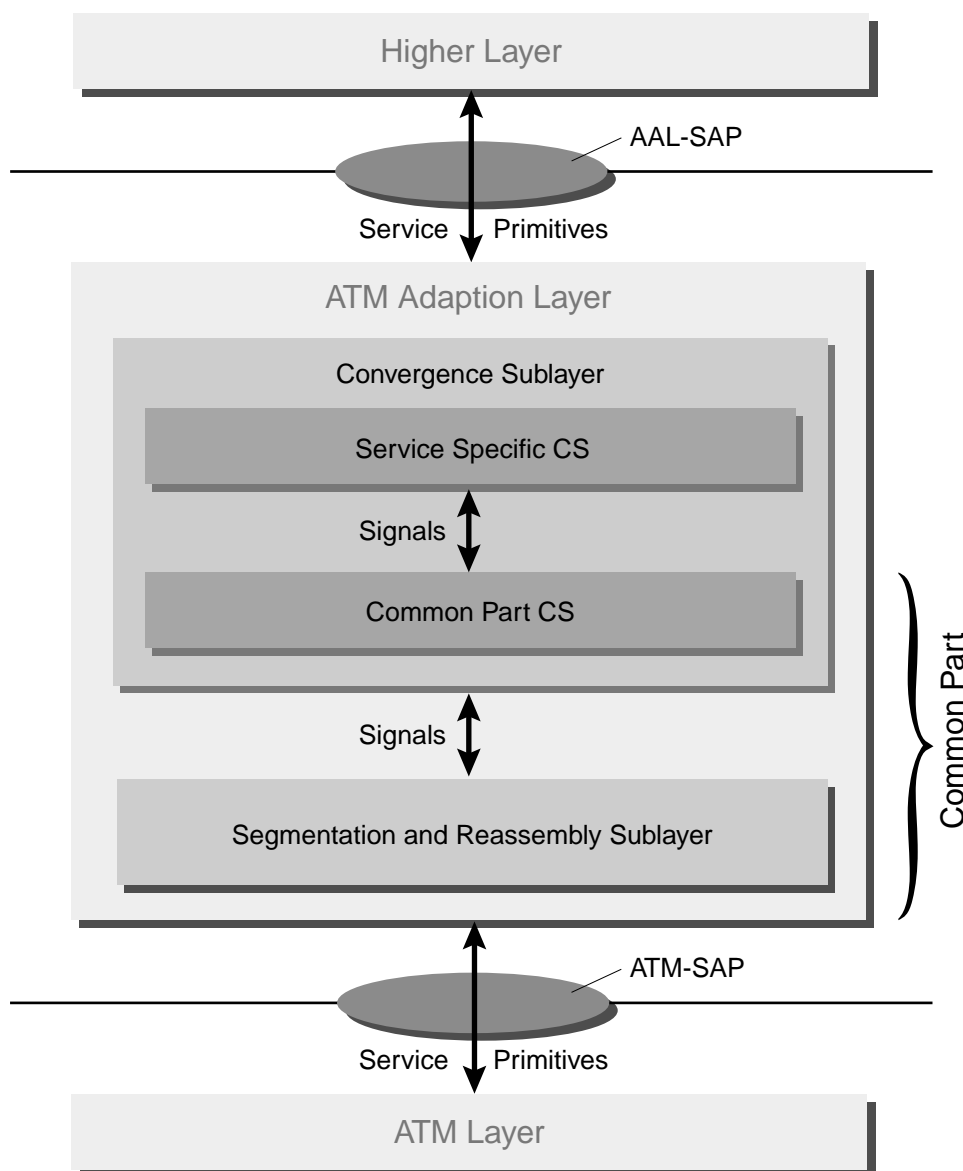


Abbildung 44 : Unterteilung des AAL-Layers

Aufgrund der Komplexität der Aufgaben im *Convergence Sublayer* wird dieser häufig weiter in den *Common Part Convergence Sublayer CPCS* und den *Service Specific Convergence Sublayer SSCF* unterteilt. Welche Funktionalitäten von den Sublayern wahrgenommen werden, hängt stark von den Diensten ab, die von der AAL zu erbringen sind, und variiert zwischen den verschiedenen AAL-Typen. Daher werden die jeweils definierten Sublayer in den folgenden Abschnitten separat für jeden AAL-Typ behandelt.

Bem.: Dienstanforderungselemente zwischen Sublayern werden zur Unterscheidung von den „echten“ über einen SAP übergebenen *Service Primitives* mit *Signals* bezeichnet.

3.8.1 AAL-Typ 0

Instanzen vom Typ AAL 0 sind ausschließlich für Daten vorgesehen, die bereits in Zellenform vorliegen; daher enthalten sie weder Sublayer noch eigene Funktionen.

3.8.2 AAL-Typ 1

Protokollinstanzen des AAL-Typs 1 gem. ITU-Recommendation I.363.1 [I363_1] dienen der Übertragung synchroner Daten mit konstanter Bitrate. Die Dienste dieses AAL-Typs umfassen:

- Übertragung von SDUs mit konstanter Quellenbitrate
- Übertragung von Taktinformationen zwischen Sender und Empfänger
- Übertragung von Angaben zur Datenstruktur zwischen Sender und Empfänger
- Anzeige verlorener bzw. fehlerhafter Nutzdaten, die nicht rekonstruiert werden konnten.

3.8.2.1 Dienstelemente des AAL-Typs 1

Die o.g. Dienste werden mit Hilfe folgender Dienstelemente erbracht:

Dienstelement	Erbringer
Überwachung der AAL-PCI auf Fehler	SAR-Sublayer
Bearbeitung von Fehlern in der AAL-PCI	
Segmentierung und Assemblierung der Nutzinformation	CS-Sublayer
Blocking ^a und Deblocking ^b	
Funktionen zur Reduzierung bzw. Elimination der <i>Cell delay variation CDV</i> (Varianz der Zellzwischenankunftszeiten) beim Empfänger	
Behandlung der Zeitdauer bis zum vollständigen Auffüllen des Zellnutzlastfeldes mit Nutzerinformationen	
Regenerierung des Diensttakts aus den Daten im Empfänger	
Bearbeitung der vom SAR-Sublayer festgestellten AAL-PCI-Fehler	
Übertragung strukturierter Daten	
Überwachung der Nutzerinformation auf Bitfehler und deren Korrektur	

Tabelle 27 : Dienstelemente des AAL-Typs 1

a. „Sammeln“ von AAL-User-SDUs bis das Nutzdatenfeld einer Zelle vollständig gefüllt ist

b. Umkehroperation zum Blocking

3.8.2.2 Dienstanforderungselemente und Signale

Name	Bedeutung
AAL-UNITDATA-REQUEST	Befehl zum Senden einer AAL-SDU zur AAL-Peer-Entity
AAL-UNITDATA-INDICATION	Übergabe einer empfangenen Zelle an die AAL-User-Protokollinstanz

Tabelle 28 : Dienstanforderungselemente zwischen AAL-Benutzer und AAL-1-Layer

Name	Bedeutung
SAR-UNITDATA-INVOKE	Befehl zum Senden der CS-SDU zur Peer-Entity
SAR-UNITDATA-SIGNAL	Übergabe einer empfangenen SAR-SDU an die CS-Protokollinstanz

Tabelle 29 : Signale zwischen SAR-Sublayer und CS-Sublayer

Bezüglich der Service Primitives zwischen AAL-1-Layer und ATM-Layer siehe Tabelle 24 auf Seite 61 mit den Parameterdefinitionen aus Anhang B.5 auf Seite 190.

3.8.2.3 Funktionen des SAR-Sublayers

Die SAR-Entity empfängt eine CS-IDU mit der Länge von 47 Oktett, fügt ein Oktett SAR-PDU-Header an und gibt diesen 48 Oktett langen Datenblock an die ATM-Entity ab.

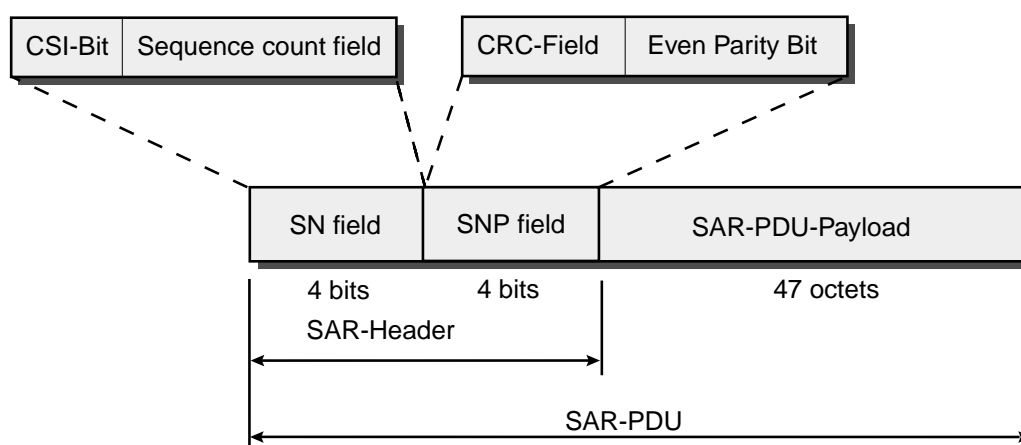


Abbildung 45 : Aufbau einer SAR-PDU

Der Header der SAR-PDU besteht neben dem vom CS-Sublayer übernommenen CSI-Bit aus einem *Sequence Count Field* (Sequenznummernfeld), welches im *CRC-Field* durch einen 3-Bit-CRC-Code gegen Übertragungsfehler geschützt ist. Die höherwertigen sieben Bit sind durch ein Paritätsbit noch-

mals geschützt. Durch Prüfung der empfangenen CRC und Parität kann der Empfänger Fehler in der Numerierung feststellen und fehlende Zellen detektieren. Der Empfänger verwendet hierbei ein Verfahren mit zwei unterschiedlichen Zuständen (vgl. Abbildung 46), welches die Korrektur eines einzeln auftretenden fehlerhaften SN-Wertes zuläßt. Das Ergebnis der Prüfung wird dem CS-Sublayer im Parameter *Check Status* des *SAR-UNITDATA-SIGNAL* (vgl. Tabelle 66 im Anhang auf Seite 190) mitgeteilt.

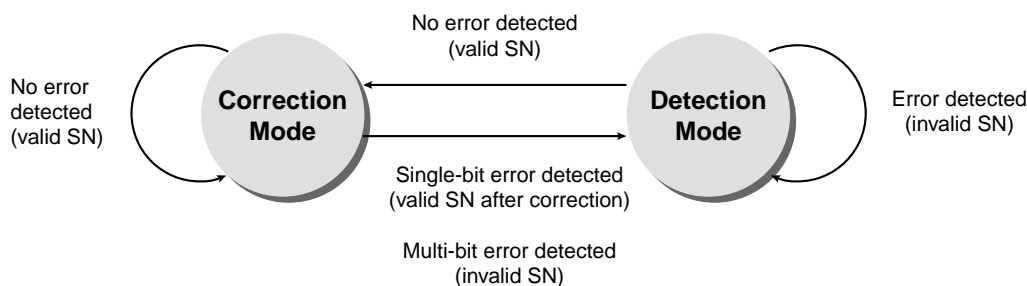


Abbildung 46 : SN-Prüfung im AAL-1-Empfänger gem. Figure 4/I.363.1

3.8.2.4 Funktionen des CS-Sublayers

Der CS-Sublayer des AAL-Typ 1 stellt eine Sammlung von Funktionen dar (siehe Tabelle 27 auf Seite 64), aus denen sich der Anwender die von ihm gewünschten Merkmale zusammenstellen kann.

Blocking und Deblocking

Der CS-Sublayer führt das Einfügen der Nutzinformationen in die SAR-PDU-Payload durch (siehe Abbildung 22 auf Seite 35 mit einem Nutzdatenfeld der Länge 47 Oktett). Der Zeitpunkt, zu dem die CS-PDU abgesendet werden kann, hängt direkt vom Füllgrad des Nutzlastfeldes ab; ist die Wartedauer bis zum vollständig gefüllten Nutzlastfeld zu lange, werden teilweise gefüllte PDUs übertragen.

Reduzierung der *Cell delay variation*

Der Empfänger speichert in einem Pufferspeicher eine bestimmte Menge von Daten, bevor er beginnt, diese mit dem regenerierten Diensttakt zum Anwender zu senden. Die Größe dieses Puffers ist in der ITU Recommendation I.356 [I356] festgelegt.

Bearbeitung der vom SAR-Sublayer festgestellten AAL-PCI-Fehler

Die vom SAR-Sublayer festgestellten Fehler in der PDU-Numerierung werden gemäß dem in der ITU-Recommendation I.363.1, Appendix III, Figure 1 und Figure 2 [I363_1] beschriebenen Algorithmus

behandelt. Abhängig von der Anzahl festgestellten Fehler werden fehlerhafte Zellen entfernt oder Füllzellen zur Aufrechterhaltung der synchronen Datenrate eingefügt.

Regenerierung des Quelltaktes aus den Daten im Empfänger (*Clock Recovery*)

Die Methode zur Taktregenerierung hängt vom Vorhandensein eines stabilen Netzwerktaktes ab. Die Genauigkeit der Taktsignale gem. ITU Recommendation G.823 [G823] und G.824 [G824] bei der Übertragung von synchronen Daten aus der SDH-Hierarchie gem. ITU Recommendation G.702 [G702] ist ausreichend; hier sind keine korrigierenden Eingriffe notwendig. Bei Diensten, die nicht an einem genauen Netzwerktakt angekoppelt sind, wird entweder die *Adaptive Clock Recovery* oder die *SRTS*¹-Methode angewendet. Im ersten Fall wird aus den Ankunftszeitpunkten der empfangenen Daten eine mittlere Taktrate bestimmt, die zum Ausspielen der Daten an den Benutzer verwendet wird. Voraussetzung bei der *SRTS-Methode* ist ein stabiler Referenznetztakt. Der Sender vergleicht einen aus der Rate der vom Benutzer eintreffenden Daten extrahierten Servicetakt mit diesem Referenztakt. Er sendet die Abweichung in den CSI-Positionen über mehrere CS-PDUs verteilt zum Empfänger, der mithilfe dieser Abweichung und demselben Referenztakt den Servicetakt des Senders regeneriert (vgl. I.363.1, Sec. 2.5.2.2.2).

Übertragung strukturierter Daten

Dieser Dienst des CS-Sublayers unterstützt die Übertragung von Datenstrukturen mit fester Länge. Die Datenblöcke werden über mehrere CS-PDUs verteilt übertragen. Die erste dieser PDUs wird durch den STRUCTURE-Parameter START gekennzeichnet, der Beginn der Struktur innerhalb der PDUs wird durch einen Pointer im Payload einer PDU gekennzeichnet (vgl. I.363.1 Sec. 2.5.2.3).

Überwachung der Nutzerinformation auf Bitfehler und Vorwärtsfehlerkorrektur

Die Methoden zur Korrektur von Bitfehlern im Empfänger beruhen auf der *Forward Error Correction FEC* mit Reed-Solomon-Codes der Familie (128,126). Eine Fehlerkorrektur über Wiederholungsanforderungen kann wegen der erforderlichen Synchronität nicht eingesetzt werden. Die Prüfoktets werden aus den Daten über Galois-Felder der Basis 256 mit dem Generatorpolynom

1. Synchronous Residual Time Stamp

$$\prod_{i=0}^3 (x - \alpha^{i+120}) \quad (\text{Gl. 21})$$

gebildet, wobei α die Wurzel des Polynoms $x^8 + x^7 + x^2 + x + 1$ ist¹. Die hier verwendeten Codes erzeugen als Schutz für jeweils 124 Oktetts ein Prüfwort aus 4 Oktetts und sind damit in der Lage, bis zu zwei fehlerhafte Oktetts zu korrigieren. Ein FEC-Block entspricht einer CS-PDU und besteht aus 47 aufeinanderfolgenden FEC-Rahmen à 128 Oktetts, demnach insgesamt 6016 Oktetts (siehe nachfolgende Abbildung).

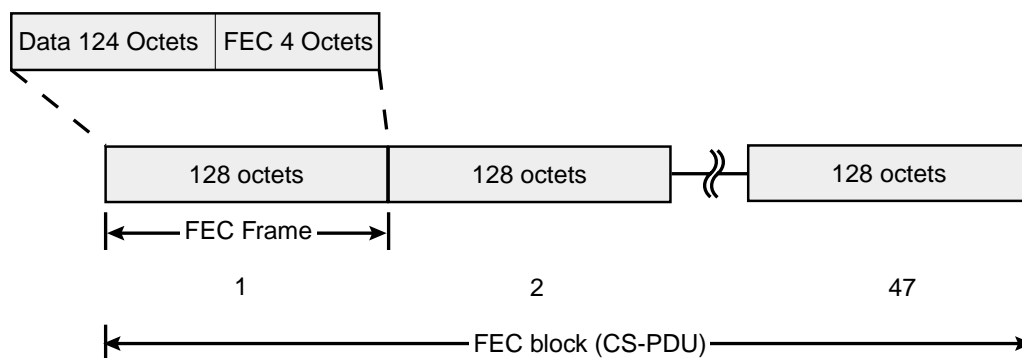


Abbildung 47 : Struktur eine FEC-Blocks

- Korrekturmethode für Bitfehler und Zellverluste ohne Verzögerungsbeschränkungen

Ergänzend zum Schutz durch die RS-Codes werden die Daten einer CS-PDU durch einen speziellen Schreib-/Lesevorgang, dem *Octet interleaving* (siehe Abbildung 48 auf Seite 69) auf mehrere SAR-PDUs verteilt. Diese Methode erlaubt die Korrektur von zwei fehlerhaften Oktetts oder vier Auslösungen². Der Verlust einer SAR-PDU führt durch diese Erweiterung nur zur Auslöschung eines Oktetts innerhalb jeder Zeile im Empfänger, die durch die Vorwärtsfehlerkorrektur behoben werden kann. Insgesamt sind durch Einsatz dieses Algorithmus folgende Fehler behebbar:

- * 4 Zellverluste,
- * 2 Zellverluste und ein fehlerhaftes Oktett in jeder Zeile oder
- * 2 fehlerhafte Oktetts in jeder Zeile, wenn kein Zellverlust aufgetreten ist.

1. Ausführliche Darstellungen der Theorie zu Galoisfeldern und Reed-Solomon-Codes sind in [Swe92] und [Pet67] zu finden.
 2. Fehlerhafte Oktetts, deren Position im Datenstrom bekannt ist.

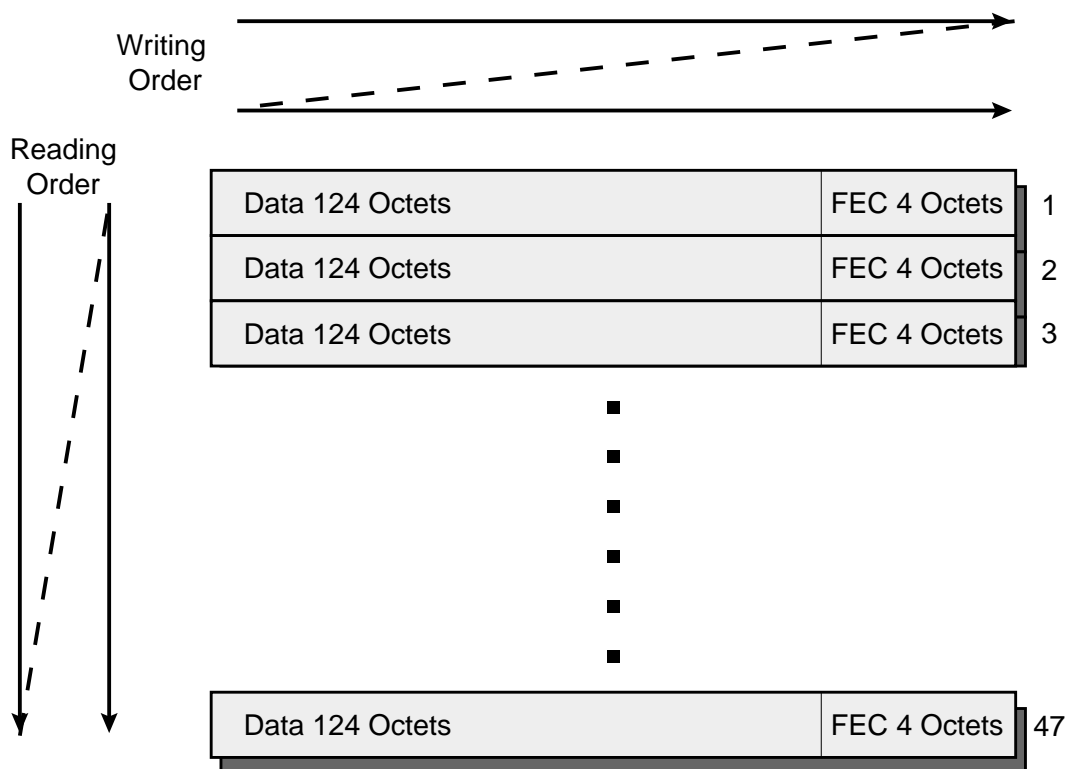


Abbildung 48 : Format der Interleaver-Matrix

Nachteilig ist der Zeitverlust, da in Sender und Empfänger jeweils die Matrix komplett gefüllt sein muß, bevor mit dem Auslesen begonnen werden kann.

- Korrekturmethode für Bitfehler und Zellverluste mit Verzögerungsbeschränkungen

Zur Vermeidung der Verzögerungen aus dem vorherigen Verfahren wird eine kleinere Interleaving-Matrix (siehe Abbildung 49 auf Seite 70) und ein anderer RS-Code (94,88) über dem Galoisfeld (256) benutzt. Das Generatorpolynom lautet dann

$$\prod_{i=0}^5 (x - \alpha^{i+120}) \tag{Gl. 22}$$

Vorteilhaft an dieser Art des Lesevorgangs ist, daß nur bis zur 15. SAR-PDU gewartet werden muß, bis das Auslesen beginnen kann, da dann ein Lesevorgang nur auf bereits gefüllte Positionen trifft. Im Empfänger ist das Vorgehen invers zum Sendevorgang, d.h. hier wird diagonal geschrieben und horizontal gelesen.

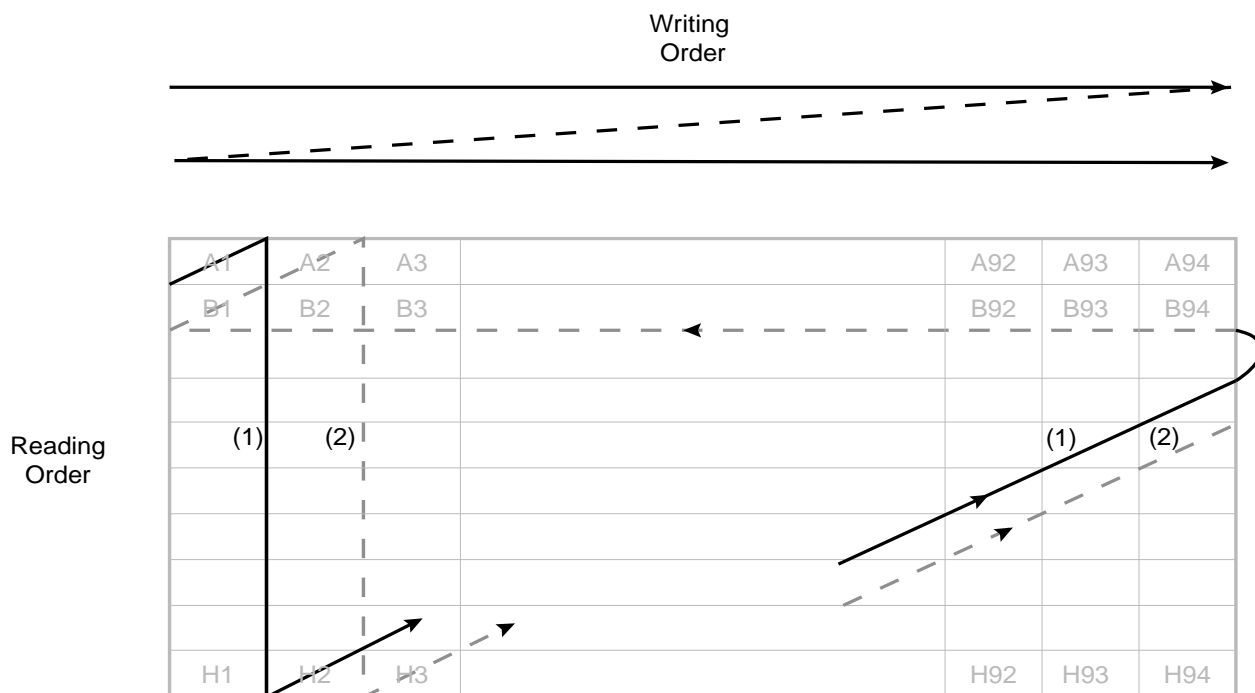


Abbildung 49 : Format der Short Interleaving Matrix

Das geschilderte Verfahren ist in der Lage,

- * einen Zellverlust in einer Gruppe von 16 Zellen oder
 - * 3 fehlerhafte Oktetts innerhalb einer Zeile
- zu korrigieren (vgl. I.363.1, Sec. 2.5.2.4.3).

3.8.2.5 Datenaustausch mit der Management-Entity

Informationen über folgende Ereignisse werden von der User-Plane zum Layermanagement gesendet:

- Fehler in der Nutzdatenübertragung,
- Verlorene oder falsch vermittelte Zellen,
- Zellen mit fehlerhaften AAL-Headern,
- Verlust von Takt oder Synchronität,
- Pufferüberlauf oder -leerlauf.

3.8.3 AAL-Typ 2

Der AAL-Typ 2, dessen Aufbau in Abbildung 50 gezeigt ist, dient der Übertragung zeitkontinuierlicher Daten mit variabler Bitrate (vgl. Tabelle 25 auf Seite 62).

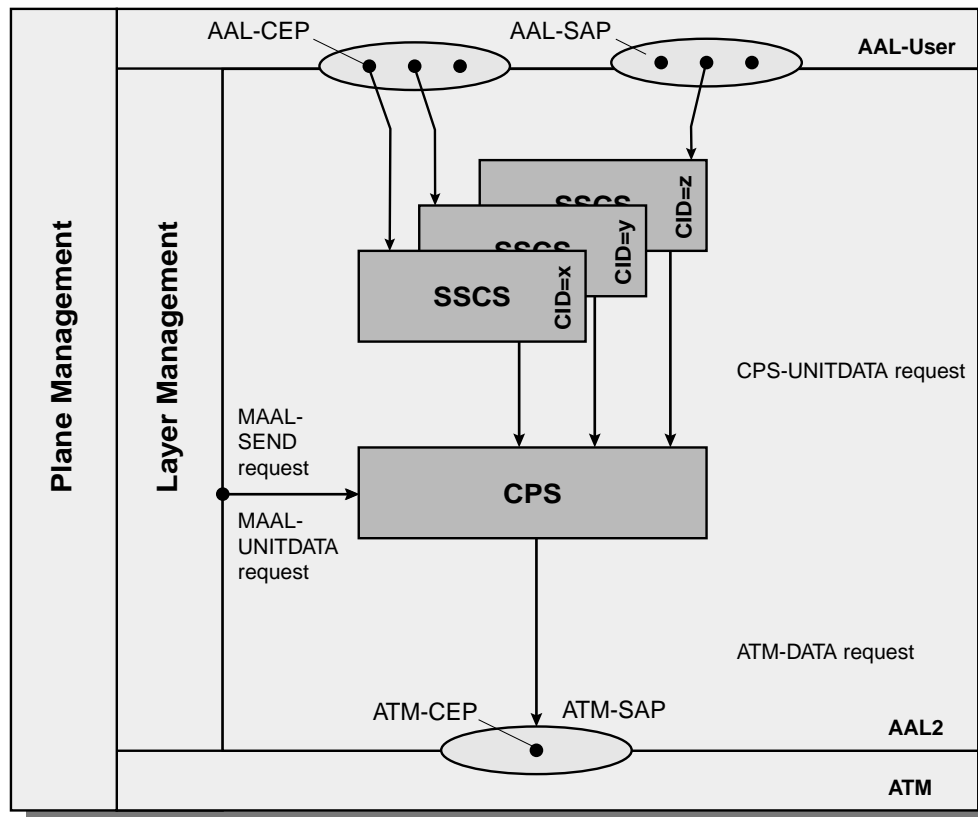


Abbildung 50 : Funktionsmodell des AAL-Typs 2

Der Sublayer SSCS ist bei diesem AAL-Typ leer, er dient ausschließlich der Umsetzung der AAL-Service Primitive in die entsprechenden CPS-Primitive und umgekehrt. Die Funktion des Layermanagement geht hier über die sonst üblichen Aufgaben hinaus. Es unterhält mit seiner Peer-Entity eine eigenständige Verbindungsverwaltung (ANP¹ gem. I.363.2 Annex C [I363_2]) und kontrolliert über das Dienstanforderungselement *MAAL-SEND-REQUEST* das Versenden der Nutzdaten in der User-Plane (vgl. Abbildung 50).

1. AAL type 2 Negotiation Procedures

3.8.3.1 Dienste des Common Part Sublayers

Der Common Part Sublayer erbringt der SSCS die in der folgenden Tabelle aufgeführten Dienste.

Dienste des Common Part Sublayers CPS
Ungesicherter Transport von Nutzdaten von bis zu 45 oder 64 Oktetts
Multiplex mehrerer CPS-Verbindungen
Sichern der Reihenfolge der CPS-PDUs innerhalb jeder CPS-Verbindung
Verlorene PDUs werden nicht wiederholt angefordert

Tabelle 30 : Dienste des Common Part Sublayers des AAL-Typs 2

3.8.3.2 Service Primitives

Die Service Primitives zwischen CPS und ATM-Layer sind bereits in Tabelle 24 auf Seite 61 aufgeführt; die weiteren Dienstanforderungselemente zeigen die folgenden beiden Tabellen.

Name	Bedeutung
CPS-UNITDATA-REQUEST	Befehl zum Senden der CPS-SDU zur Peer-Entity
CPS-UNITDATA-INDICATION	Übergabe einer empfangenen CPS-SDU an die SSCS-Protokollinstanz

Tabelle 31 : Service Primitives zwischen SSCS und CPS

Name	Bedeutung
MAAL-UNITDATA-REQUEST	Befehl zum Senden der CPS-SDU zur Peer-Layermanagement-Entity
MAAL-UNITDATA-INDICATION	Abgabe einer empfangenen CPS-SDU an die Layermanagement-Entity
MAAL-SEND-REQUEST	Befehl zum Senden einer User-CPS-PDU

Tabelle 32 : Service Primitives zwischen Layermanagement und CPS

3.8.3.3 Datenformat des AAL-Typs 2

Bit								Octet
8	7	6	5	4	3	2	1	
CID								1
LI				PPT				2

Tabelle 33 : Aufbau der CPS-PDU

Bit								Octet
8	7	6	5	4	3	2	1	
UUI			HEC					3
OSF					SN	P	4 (STF)	
INFO								5 . .
PAD								48/67

Tabelle 33 : Aufbau der CPS-PDU

Die Bedeutung der Felder der CPS-PDU ist in Anhang A.3 auf Seite 180 erläutert. Das Erstellen der CPS-PDUs aus den Paketen des AAL-2-Benutzers geschieht durch byteweise Übernahme der Nutzerdaten in das Nutzlastfeld der zum Versenden anstehenden CPS-PDU. Sollte sich am Beginn des Nutzlastfeldes noch ein „Rest“ des vorangegangenen Nutzerdatums befinden, wird der Beginn des aktuellen Paketes im „Offset Field“ (vgl. Anhang A.3 auf Seite 180) vermerkt. Den gesamten Vorgang bis zur Erstellung der Zellpayload für die ATM-Protokollinstanz zeigt die folgende Abbildung.

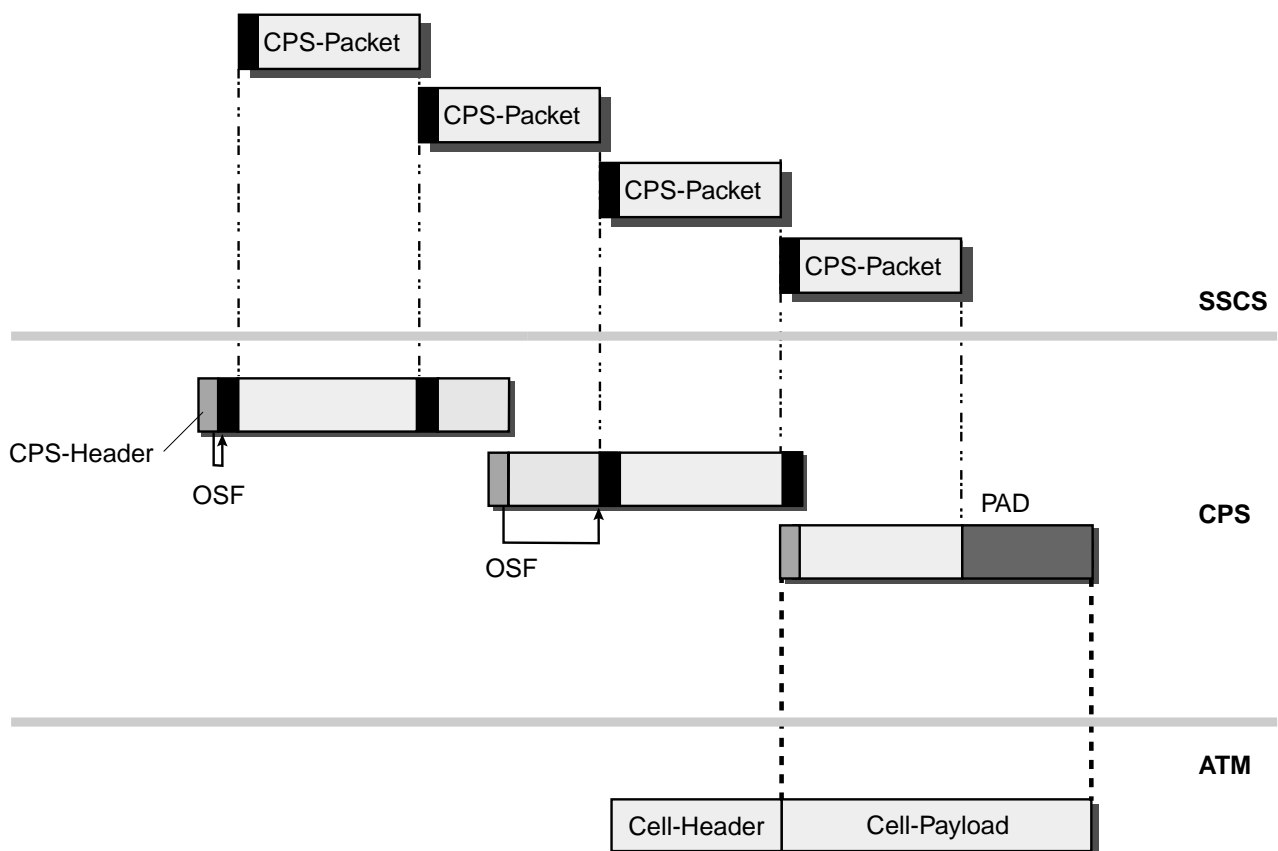


Abbildung 51 : Erzeugung von ATM-Zellen in einer AAL-Typ 2-Instanz

3.8.3.4 Ablauf- und Zustandsdiagramme

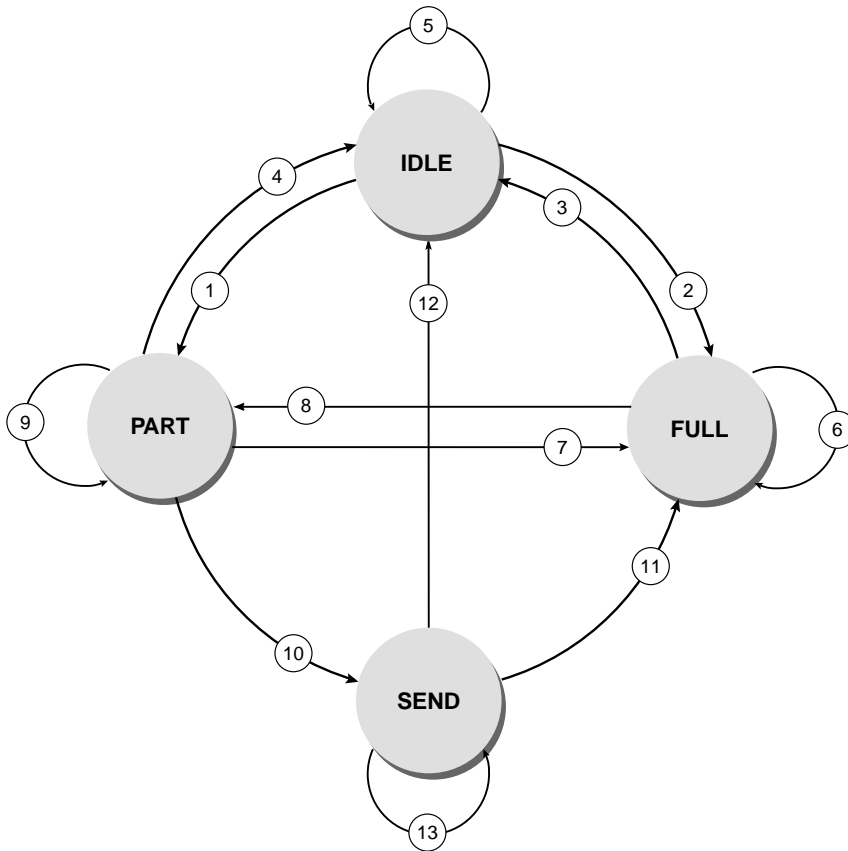


Abbildung 52 : Zustandsdiagramm AAL-Typ 2 (Sender)

Die Abläufe innerhalb einer Protokollinstanz des AAL-Typs 2 werden von dem in Abbildung 52 gezeigten Zustandsdiagramm gesteuert. Die Bedeutung der einzelnen Zustände kann der folgenden Tabelle entnommen werden.

Zustand	Bedeutung
IDLE	Die als nächste zu versendende CPS-PDU ist leer
PART	Die CPS-PDU ist teilweise gefüllt, es ist noch Platz für weitere Pakete vom CPS-Benutzer
FULL	Die CPS-PDU ist vollständig gefüllt, das aktuelle Paket vom CPS-Benutzer ragt in die nächste PDU hinein, der Sender wartet auf den Befehl des Layermanagements, um die CPS-PDU zu senden
SEND	Das Layermanagement hat innerhalb einer bestimmten Zeit (Timer_CU) keinen Sendeauftrag für eine vollständig gefüllte CPS-PDU erteilt

Tabelle 34 : Interne Zustände der AAL-2

Die Übergangsbedingungen 1 - 13 in Abbildung 52 auf Seite 74 hängen von einer Vielzahl interner Statusvariablen der AAL-2-Entity ab und sind daher an dieser Stelle nicht näher erläutert. Eine detaillierte Beschreibung ist in der ITU-Recommendation I.363.2 in Abschnitt 10 [I363_2] zu finden.

3.8.4 AAL-Typ 3/4

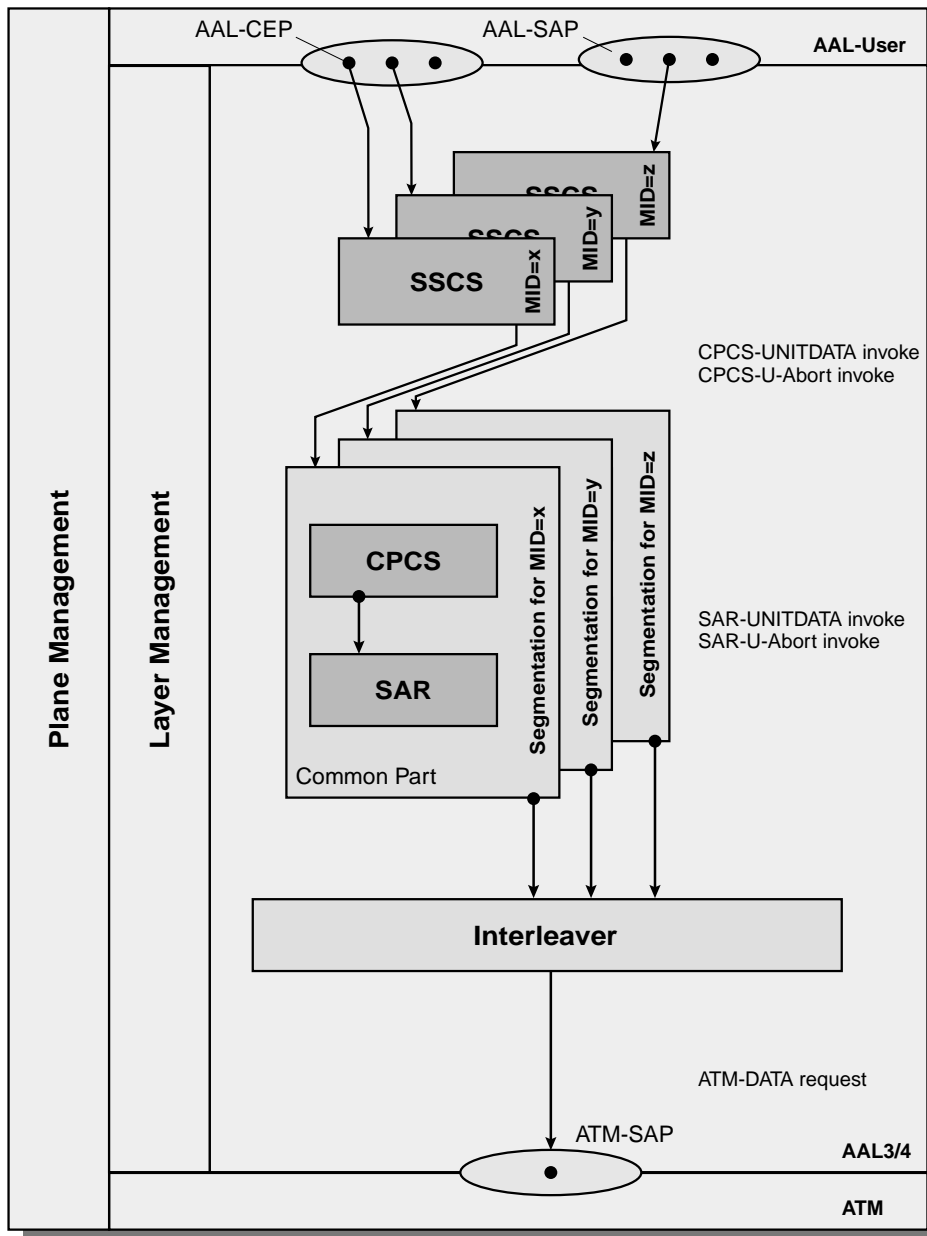


Abbildung 53 : Aufbau einer Protokollinstanz des AAL-Typs 3/4

Die ITU empfiehlt den Einsatz des AAL-Typs 3/4 zum Transfer von Daten, die empfindlich gegenüber Verlust, nicht jedoch gegenüber Verzögerungen (vgl. Tabelle 25 auf Seite 62) sind. Diese AAL kann sowohl für verbindungsorientierte als auch für verbindungslose Kommunikationsbeziehungen

eingesetzt werden und unterstützt das Multiplexen mehrerer Verbindungen über eine ATM-Verbindung. Der Anwender hat die Möglichkeit, durch Auswahl eines geeigneten AAL-SAPs unterschiedliche Funktionalitäten zu nutzen, um einen für die QoS-Parameter seiner Daten geeigneten Funktionsumfang auszuwählen.

3.8.4.1 Aufbau

Der interne Aufbau des AAL-Typs 3/4 entspricht der in Abbildung 53 auf Seite 75 gezeigten Struktur. Die Sublayer SAR und CPCS sind in Recommendation I.363.3 [I363_3] definiert, spezielle SSCS-Funktionalitäten werden in separaten Recommendations aufgeführt.

3.8.4.2 Dienste des Common Part Sublayers CPS

Dienste des Common Part Sublayers
Fehlererkennung und -anzeige
Multiplex mehrerer Verbindungen
Sichern der Reihenfolge der CPCS-PDUs innerhalb jeder CPCS-Verbindung
Ungesicherter Transport von Nutzdaten in beliebiger Länge zwischen 1 und 65535 Oktetts

Tabelle 35 : Dienste des CP-Sublayers des AAL-Typs 3/4

Fehlererkennung und -anzeige

Der Standard sieht zwei Möglichkeiten zur Behandlung der als fehlerhaft erkannten PDUs vor:

- Verwerfen innerhalb der CPCS oder
- Weitergabe an die SSCS mit einer Fehleranzeige.

Multiplex mehrerer Verbindungen

Eine oder mehrere CPCS-Verbindungen können zwischen zwei AAL3/4-Peer-Entities bestehen, die alle dieselbe ATM-Verbindung benutzen.

Ungesicherter Transport von Nutzdaten

Unterschieden werden hier zwei Betriebsmodi, der *Message Mode* und der *Streaming Mode*. Im *Message Mode* wird jedes Benutzer-Datenpaket (CPCS-IDU) in genau einer CPCS-PDU transportiert. Bei

Einsatz des *Streaming Mode* werden mehrere aufeinanderfolgende CPCS-IDUs in einer CPCS-PDU zusammengefaßt und übertragen. In diesem Modus ist ein vorzeitiges Senden der vom Benutzer eintreffenden IDUs (*Pipelining*) möglich. Die Entity beginnt bereits während des Eintreffens der IDUs mit dem Senden der CPCS-PDU. Schlägt die Übertragung einer CPCS-PDU im Pipeline-Betrieb fehl, kann der bereits begonnene Datentransfer durch die *Abort*-Funktion wieder abgebrochen werden.

Datenflußkontrolle

Diese Funktion ist optional und bisher nicht standardisiert.

3.8.4.3 Funktionen des SAR-Sublayers

Dienst	Bedeutung
Preservation of SAR-PDU	Markieren der Länge des Nutzlastfelds der SAR-PDU und Kennzeichnung der SAR-PDU als <i>Begin of Message</i> (BOM), <i>Continuation of Message</i> (COM) oder <i>Single Segment Message</i> (SSM)
Error detection and handling	Behandlung fehlerhafter, verlorener und falsch eingefügter SAR-PDUs. Fehlerhafte PDUs werden verworfen oder mit einer Fehleranzeige an die CPCS-Entity abgegeben
SAR-PDU sequence integrity	Sicherstellen der Reihenfolge innerhalb einer SAR-Verbindung
Multiplexing/demultiplexing	Multiplexen mehrerer SAR-Verbindungen über eine ATM-Verbindung
Abort	Verwerfen einer bereits teilweise gesendeten SAR-PDU
Handling of congestion information	Weiterleitung der <i>congestion information</i> zwischen CPCS- und ATM-Entity; bisher nicht standardisiert
Handling of loss priority information	Weiterleitung der Zellprioritäten zwischen CPCS- und ATM-Entity; bisher nicht standardisiert

Tabelle 36 : Dienste der SAR-Entity

3.8.4.4 Dienstanforderungselemente und Signale

Name	Bedeutung
CPCS-UNITDATA-INVOKE	Befehl zum Senden der CPCS-SDU zur CPCS-Peer-Entity
CPCS-UNITDATA-SIGNAL	Übergabe einer empfangenen CPCS-PDU an den Benutzer
CPCS-U-Abort -INVOKE	Aufforderung zum Verwerfen eines bereits teilweise gesendeten Datenpaketes

Tabelle 37 : Signale zwischen SSCS und CPCS

Name	Bedeutung
CPCS-U-Abort -SIGNAL	Benachrichtigung des Empfängers, daß ein bereits teilweise empfangenes Datenpaket auf Anforderung des Senders zu verwerfen ist
CPCS-P-Abort -SIGNAL	Benachrichtigung des Empfängers, daß ein bereits teilweise empfangenes Datenpaket aufgrund von internen Fehlern zu verwerfen ist

Tabelle 37 : Signale zwischen SSCS und CPCS

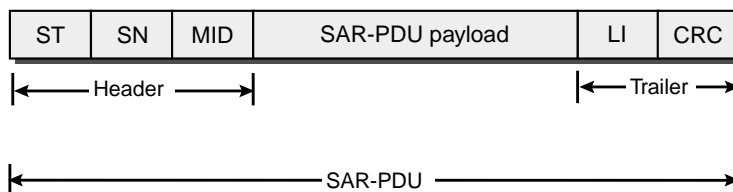
Name	Bedeutung
SAR-UNITDATA-INVOKE	Befehl zum Senden der IDU zur Peer-Entity
SAR-UNITDATA-SIGNAL	Übergabe einer SAR-PDU an die CPCS-Protokollinstanz
SAR-U-Abort -INVOKE	Aufforderung zum Verwerfen eines bereits teilweise gesendeten Datenpaketes
SAR-U-Abort -SIGNAL	Benachrichtigung des Empfängers, daß ein bereits teilweise empfangenes Datenpaket auf Anforderung des Senders zu verwerfen ist
SAR-P-Abort -SIGNAL	Benachrichtigung des Empfängers, daß ein bereits teilweise empfangenes Datenpaket aufgrund von internen Fehlern zu verwerfen ist

Tabelle 38 : Signale zwischen CPCS und SAR

3.8.4.5 Datenformat

Den Aufbau der PDUs des AAL-Typs 3/4 zeigen die beiden folgenden Abbildungen, eine Erläuterung der PDU-Felder ist in Anhang A.4 auf Seite 181 zu finden.

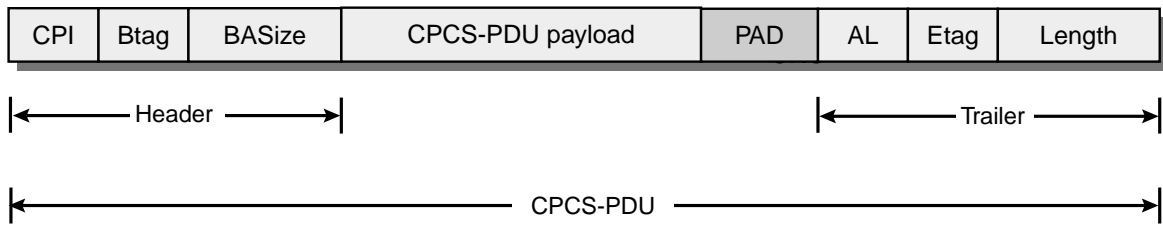
SAR-PDU



ST	Segment Type	(2 bits)	LI	Length Indication	(6 bits)
SN	Sequence Number	(4 Bits)	CRC	Cyclic Redundancy Check Code	(10 bits)
MID	Multiplexing Identification	(10 bits)			

Abbildung 54 : Format der SAR-PDU des AAL-Typs 3/4

CPCS-PDU



CPI	Common Part Indicator	(1 octet)
Btag	Beginning Tag	(1 octet)
BASize	Buffer Allocation Size	(2 octets)
PAD	Padding	(0..3 octets)
AL	Alignement	(1 octet)
Etag	End Tag	(1 octet)
Length	Length of CPCS-PDU payload	(2 octets)

Abbildung 55 : Format der CPCS-PDU des AAL-Typs 3/4

Segmentierung

Die Verarbeitung der AAL 3/4-PDUs im Streaming Modus zeigt Abbildung 56.

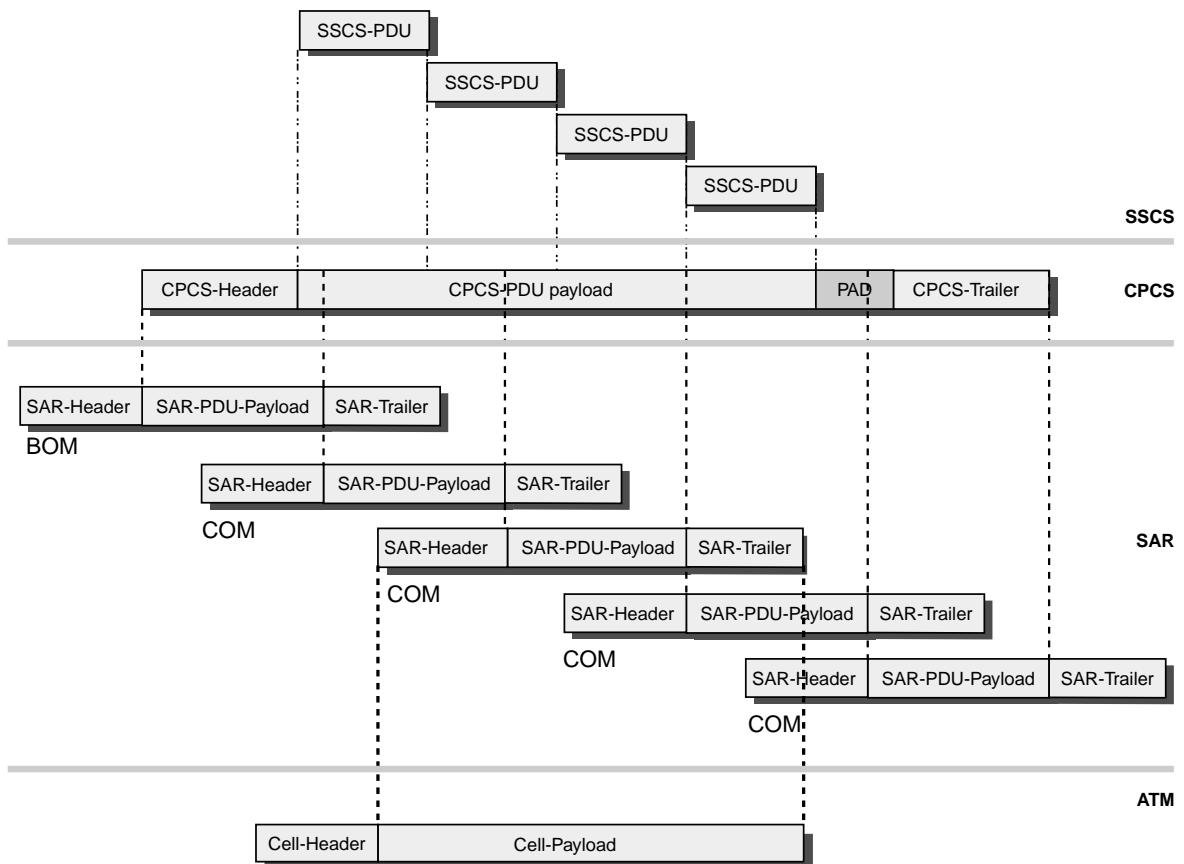


Abbildung 56 : Segmentierung der PDUs im AAL-Typ 3/4 im Streaming-Modus

3.8.4.6 Zustandsdiagramme

Bezogen auf den Segmentierungsvorgang aus Abbildung 56 auf Seite 79 werden im folgenden beispielhaft die Zustands-Diagramme der Protokollabläufe innerhalb der Sublayer SAR und CPCS gezeigt. Weitere Abläufe sind in der Recommendation I.363.3 [I363_3] oder [Ngu97] zu finden.

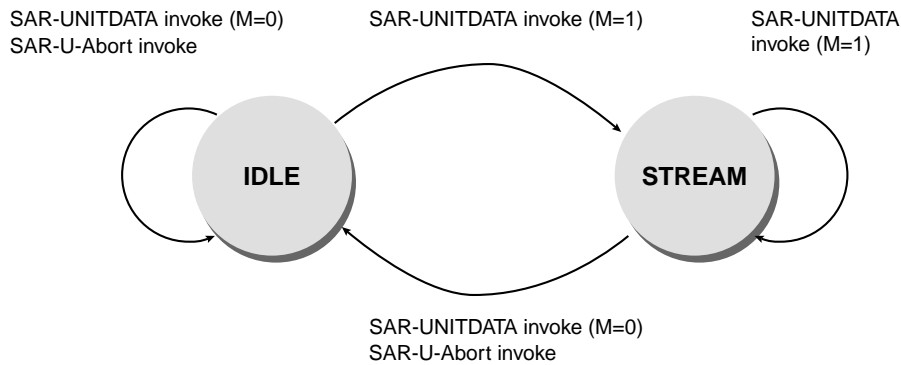


Abbildung 57 : Zustandsdiagramm SAR-Sender

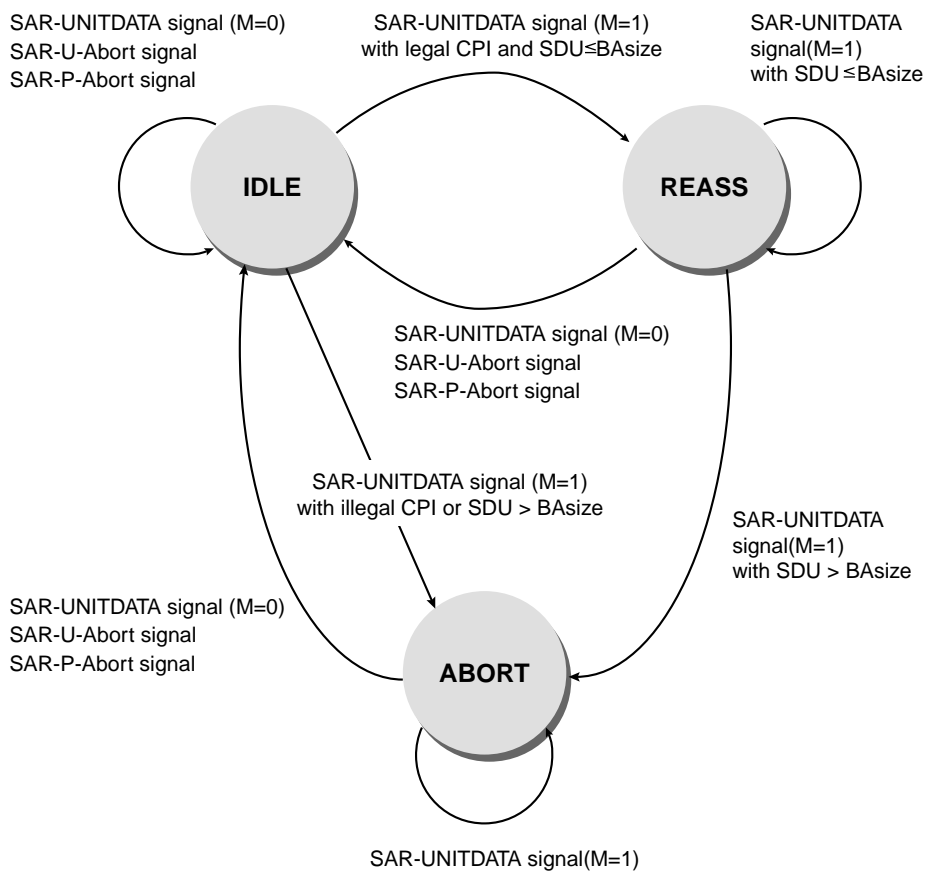


Abbildung 58 : Zustandsdiagramm CPCS-Empfänger

Abbildung 57 zeigt den Sendevorgang eines SAR-Sublayers im Streaming Modus. Beim ersten Sendeauftrag sendet der SAR-Sublayer die SAR-PDU als *BOM* und wechselt vom Zustand „IDLE“ in den „STREAM“-Zustand. Solange der SAR-User kein Abbruchsignal oder einen Sendeauftrag mit Endekennung (M=0) erteilt, verschickt die SAR-Instanz Teile der CPCS-PDU als SAR-PDU mit der Kennzeichnung *COM* (vgl. Abbildung 56 auf Seite 79). Abbildung 58 auf Seite 80 zeigt das Zustandsdiagramm des Empfängers im CPCS-Sublayer bei der Reassemblierung der ursprünglichen CPCS-PDU. Eintreffende SAR-PDUs werden auf Einhaltung der Längenbegrenzung (BASize) und Korrektheit überprüft. Wird eine fehlerhafte PDU empfangen oder trifft ein Abbruchsignal vom CPCS-User oder der SAR-Instanz ein, wird der Vorgang abgebrochen („ABORT“) und die gesamte CPCS-PDU gehen verloren.

3.8.5 AAL-Typ 5

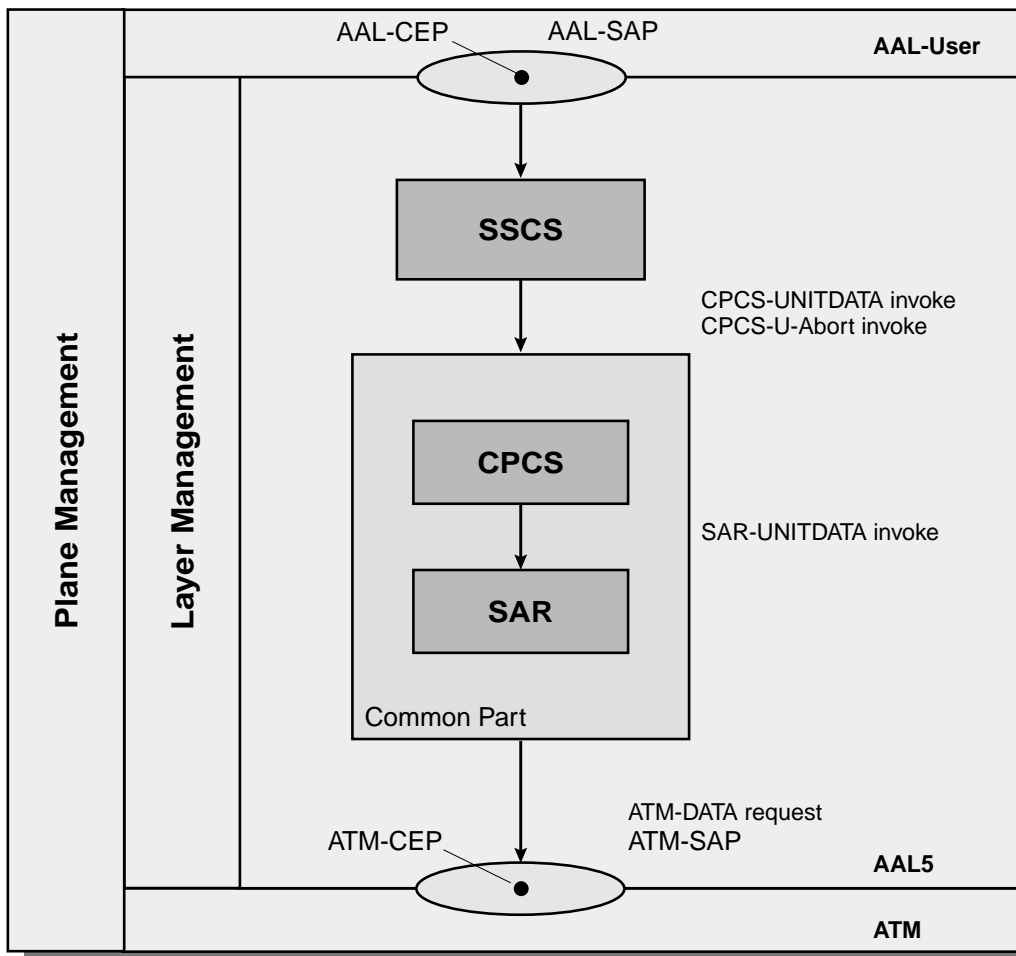


Abbildung 59 : Funktionsmodell des AAL-Typs 5

Der AAL-Typ 5 stellt eine vereinfachte Variante des Typs 3/4 dar; er wird zum Transfer von Daten ohne Zeitbindung und vorherigen Verbindungsaufbau eingesetzt (z.B. Frame Relay mittels FR-SSCS gem. ITU-Recommendation I.365.1 [I365_1]). Im Gegensatz zur AAL-Typ 3/4 entfällt hier die Möglichkeit zum Multiplexen mehrerer CPCS-Verbindungen über eine einzelne ATM-Verbindung. Weiterhin ist keine Möglichkeit einer Pufferallokation beim Empfänger (vgl. Feld „BAsize“ bei AAL3/4-PDUs) vorgesehen. Die von dem AAL-Typ 3/4 her bekannten Betriebsmodi *Message Modus* und *Streaming Modus* sowie die Möglichkeit des internen *Pipelining* existieren auch bei diesem AAL-Typ.

3.8.5.1 Dienstelemente des Common Parts

Dienstelemente des Common Parts
Ungesicherter Transport von Nutzdaten in beliebiger Länge zwischen 1 und 65535 Oktetts
Fehlererkennung und ggf. Fehleranzeige
Bewahrung der Reihenfolge der CPCS-PDUs

Tabelle 39 : Dienstelemente des Common Parts des AAL-Typs 5

3.8.5.2 Dienstanforderungselemente und Signale

Name	Bedeutung
CPCS-UNITDATA-INVOKE	Befehl zum Senden der CPCS-SDU zur CPCS-Peer-Entity
CPCS-UNITDATA-SIGNAL	Übergabe einer empfangenen CPCS-PDU an den Benutzer
CPCS-U-Abort -INVOKE	Aufforderung zum Verwerfen eines bereits teilweise gesendeten Datenpaketes
CPCS-U-Abort -SIGNAL	Benachrichtigung des Empfängers, daß ein bereits teilweise empfangenes Datenpaket auf Anforderung des Senders zu verwerfen ist
CPCS-P-Abort -SIGNAL	Benachrichtigung des Empfängers, daß ein bereits teilweise empfangenes Datenpaket aufgrund von internen Fehlern zu verwerfen ist

Tabelle 40 : Signale zwischen SSSCS- und CPCS-Sublayer

Name	Bedeutung
SAR-UNITDATA-INVOKE	Befehl zum Senden der IDU zur Peer-Entity
SAR-UNITDATA-SIGNAL	Übergabe einer SAR-PDU an die CPCS-Protokollinstanz

Tabelle 41 : Signale zwischen CPCS- und SAR-Sublayer

3.8.5.3 Datenformat

Im Vergleich zu den PDUs des Typs 3/4 haben die Pakete des AAL-Typs 5 einen wesentlich einfacheren Aufbau (siehe Abbildung 60 und Abbildung 61).

Aufbau der SAR-PDU

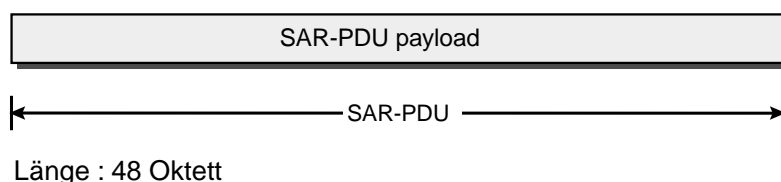


Abbildung 60 : Format der SAR-PDU für den AAL-Typ 5

Aufbau der CPCS-PDU

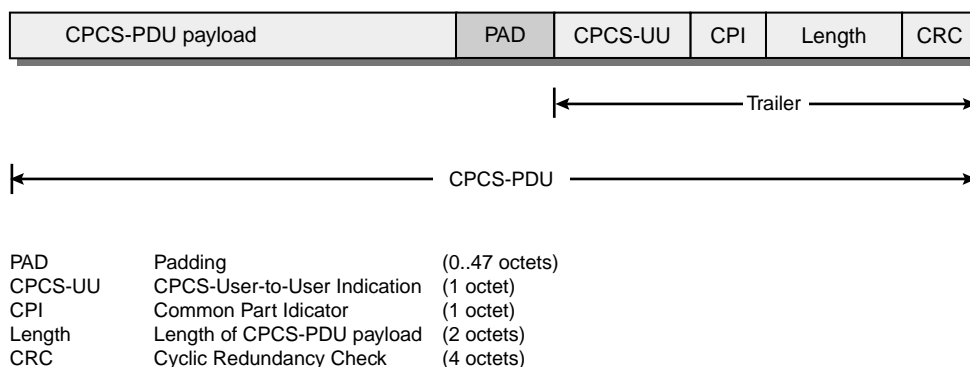


Abbildung 61 : Format der CPCS-PDU für den AAL-Typ 5

Die Bedeutung der Felder kann Abschnitt A.5 auf Seite 182 entnommen werden.

3.8.5.4 Ablaufbeispiel für Segmentierung

Die Aufteilung der CPCS-PDU auf mehrere SAR-PDUs und ATM-Zellen entspricht dem Segmentierungsvorgang in Abbildung 56 auf Seite 79 mit dem einzigen Unterschied, daß das AUU-Feld der ATM-Zelle anstelle des Feldes „Segment Type“ zur Kennzeichnung des Endes einer CPCS-PDU verwendet wird.

3.8.5.5 Zustandsdiagramme

In den Instanzen des AAL-Typs 5 werden keine Zustände unterschieden. Die Ablaufdiagramme der Vorgänge innerhalb der Sublayer sind in I.363.1, Annex D [I363_5] detailliert beschrieben.

3.8.6 SAAL

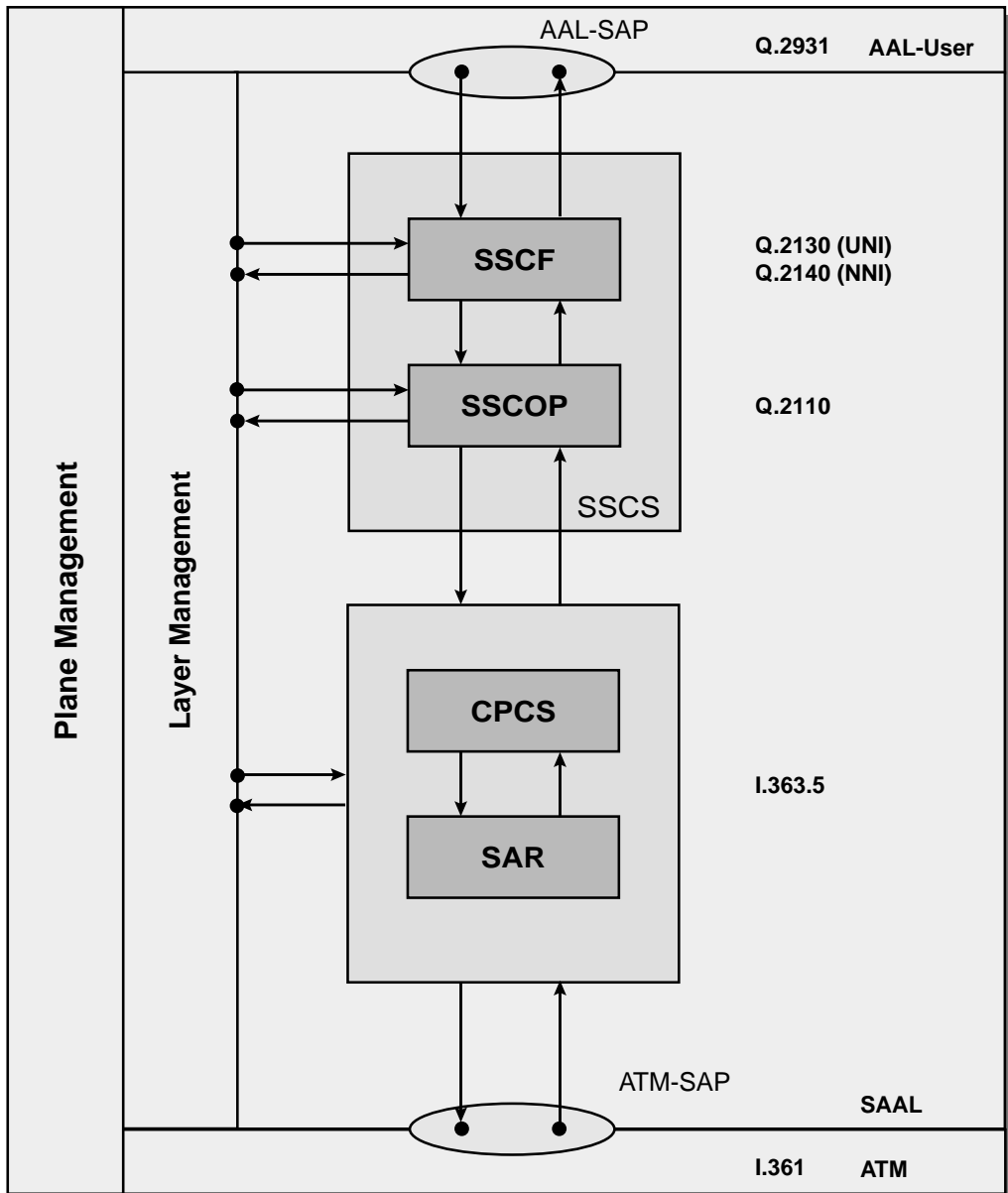


Abbildung 62 : Struktur der AAL für Signalisierungsanwendungen (gem. Fig. 1/Q.2100 [Q2100])

Keiner der bisher betrachteten AAL-Typen stellt Funktionen zur Verfügung, mit deren Hilfe eine gesicherte Datenübertragung zwischen den AAL-Benutzern gewährleistet werden könnte; lediglich die Anzeige von erkannten Fehlern ist vorgesehen. Um dem Anwender eine gesicherte Datenübertragung ähnlich der Logical Link Control gemäß IEEE 802.2 [IEEE802_2] anbieten zu können, wurden spezielle Protokolle für den SSCS-Sublayer des AAL-Typs 5 definiert. Gesicherte Übertragungsverfahren werden von den Signalisierungsprotokollen (ITU Recommendation Q.2931 [Q2931]) beim Aufbau von ATM-Verbindungen benötigt, als weitere Anwendungen sind jedoch auch andere Datendienste denkbar.

3.8.6.1 Dienste der SAAL

Dienste der SAAL-Entity
Ungesicherter Datentransfer
Gesicherter Datentransfer
Transparenz der übertragenen Daten
Auf- und Abbau gesicherter Verbindungen für den gesicherten Transfer von Daten

Tabelle 42 : Dienste der SAAL-Entity

Ungesicherter Datentransfer

Die SAAL ermöglicht die ungesicherte Übertragung von PDUs mit bis zu 4096 Oktetts auf Punkt-zu-Punkt oder Punkt-zu-Multipunkt-Verbindungen. Es erfolgt keine Benachrichtigung des Benutzers beim Auftreten von Fehlern oder Verlusten.

Gesicherter Datentransfer

Dieser Dienst stellt dem Benutzer die gesicherte Übertragung von PDUs mit einer Länge von bis zu 4096 Oktetts zur Verfügung. Die empfangende Instanz versucht fehlerhafte PDUs durch Wiederholungsanforderungen an den Sender zu korrigieren. Die SAAL-Entity benachrichtigt den Benutzer bei Verlust, Verfälschung und fehlerhafter Reihenfolge der Daten.

Transparenz der übertragenen Daten

Weder der Inhalt noch das Format der Nutzer-PDUs werden von der SAAL-Entity verändert, noch versucht sie diese PDUs zu analysieren.

Auf- und Abbau gesicherter Verbindungen für den gesicherten Transfer von Daten

Die SAAL baut Verbindungen für einen gesicherten Datentransfer für den Benutzer auf; beim Abbau der Verbindungen durch die SAAL kann es zum Datenverlust kommen.

3.8.6.2 Dienste des Service Specific Convergence Sublayers

Die Aufgaben des *Service Specific Convergence Sublayers* SCS bestehen darin, die Dienste des *Common Parts* des AAL-Typs 5 zu erweitern (vgl. Abschnitt 3.8.5.1 auf Seite 82), um Signalisierungsanwendungen in höheren Schichten eine gesicherte Übertragung zur Verfügung zu stellen. Diese

Aufgaben werden unter dem *Service Specific Connection Oriented Protocol SSCOP* gem. ITU Recommendation Q.2110 [Q2110] und der *Service Specific Coordination Function SSCF* nach den ITU Recommendations Q.2130 und Q.2140 ([Q2130], [Q2140]) aufgeteilt.

Dienste des SSCOP
Bewahrung der Reihenfolge der Nutzerdaten
Fehlerkorrektur durch selektive Wiederholung
Flußkontrolle
Verbindungsaufbau, -abbau und -wiederaufbau sowie eine ungesicherte Übertragung von Nutzer- zu -Nutzerinformationen
Senden von Fehlerberichten zum Layermanagement
Aufrechterhaltung der Verbindung auch bei längeren Pausen während der Übertragung der Nutzdaten
Möglichkeit einer lokalen Datenrücknahme durch den Benutzer
Fehlererkennung im Protokollablauf
Austausch von Statusinformationen zwischen Sender und Empfänger

Tabelle 43 : Dienste des SSCOP

Dienste der SSCF
Koordinierende Aufgaben zwischen SAAL-Benutzer und SSCOP

Tabelle 44 : Dienste der SSCF

3.8.6.3 Dienstanforderungselemente und Signale

Name	Aufgabe
AA-ESTABLISH	Aufbau einer Punkt-zu-Punktverbindung für gesicherten Datentransfer
AA-RELEASE	Abbau einer Punkt-zu-Punktverbindung für gesicherten Datentransfer
AA-DATA	Gesicherte Übertragung von Nutzdaten
AA-RESYNC	Wiederaufnahme einer SSCOP-Verbindung
AA-RECOVER	Beenden von Fehlerzuständen innerhalb des Protokollablaufs
AA-UNITDATA	Übertragung nicht gesicherter Daten
AA-RETRIEVE	Zurücknehmen bereits an den SSCOP-Sender abgegebener aber noch nicht versendeter Daten
AA-RETRIEVE - COMPLETE	Anzeige an den Benutzer, daß keine Daten zurückgegeben werden können.

Tabelle 45 : SSCOP Signale

Name	Aufgabe
MAA-ERROR	Bericht über Fehler und bestimmte Zustände an das Layermanagement
MAA-UNITDATA	Nicht gesicherte Datenübertragung zwischen Layermanagement-Entities

Tabelle 45 : SSCOP Signale

Die Service Primitive zwischen SSCOP und CPCS sind in Tabelle 40 auf Seite 82 definiert, alle Parameter sind auf „0“ zu setzen.

3.8.6.4 Datenformat

Wegen der Vielzahl unterschiedlicher PDUs im Protokoll der S-AAL werden hier nur die Pakettypen aufgezählt; das Format der einzelnen SSCOP-PDUs ist in der Empfehlung Q.2110 Figure 3 bis Figure 16 definiert.

Name	Beschreibung
BGN (Begin)	Aufbau einer Verbindung
BGAK (Begin Acknowledge)	Bestätigung der Annahme des Verbindungswunsches
BGREJ (Begin Reject)	Ablehnung eines Verbindungswunsches
END (end)	Abbau einer Verbindung
ENDAK (End Acknowledge)	Bestätigung des Verbindungsabbaus
RS (Resynchronization)	Neusynchronisation
RSAC (Resynchronization Acknowledge)	Bestätigung der Neusynchronisation
ER (Error Recovery)	Beenden von Protokollfehlern
ERAK (Error Recovery Acknowledge)	Bestätigung des Beendigung eines Protokollfehlers
SD (Sequenced Data)	Datenpaket im Rahmen der gesicherten Übertragung
POLL (Status Request)	Einholen von Statusinformationen von der Peer-Entity
STAT (Solicited Status Information)	Angeforderte Statusinformation
USTAT (Unsolicited Status Information)	Nicht angeforderte Statusinformation
UD (Unnumbered Data)	Paket im ungesicherten Übertragungsmodus
MD (Management Data)	Ungesicherte Übertragung von Daten zwischen Peer Layermanagement-Entities

Tabelle 46 : SSCOP-PDUs

3.8.6.5 Wichtige Abläufe und Zustandsdiagramme

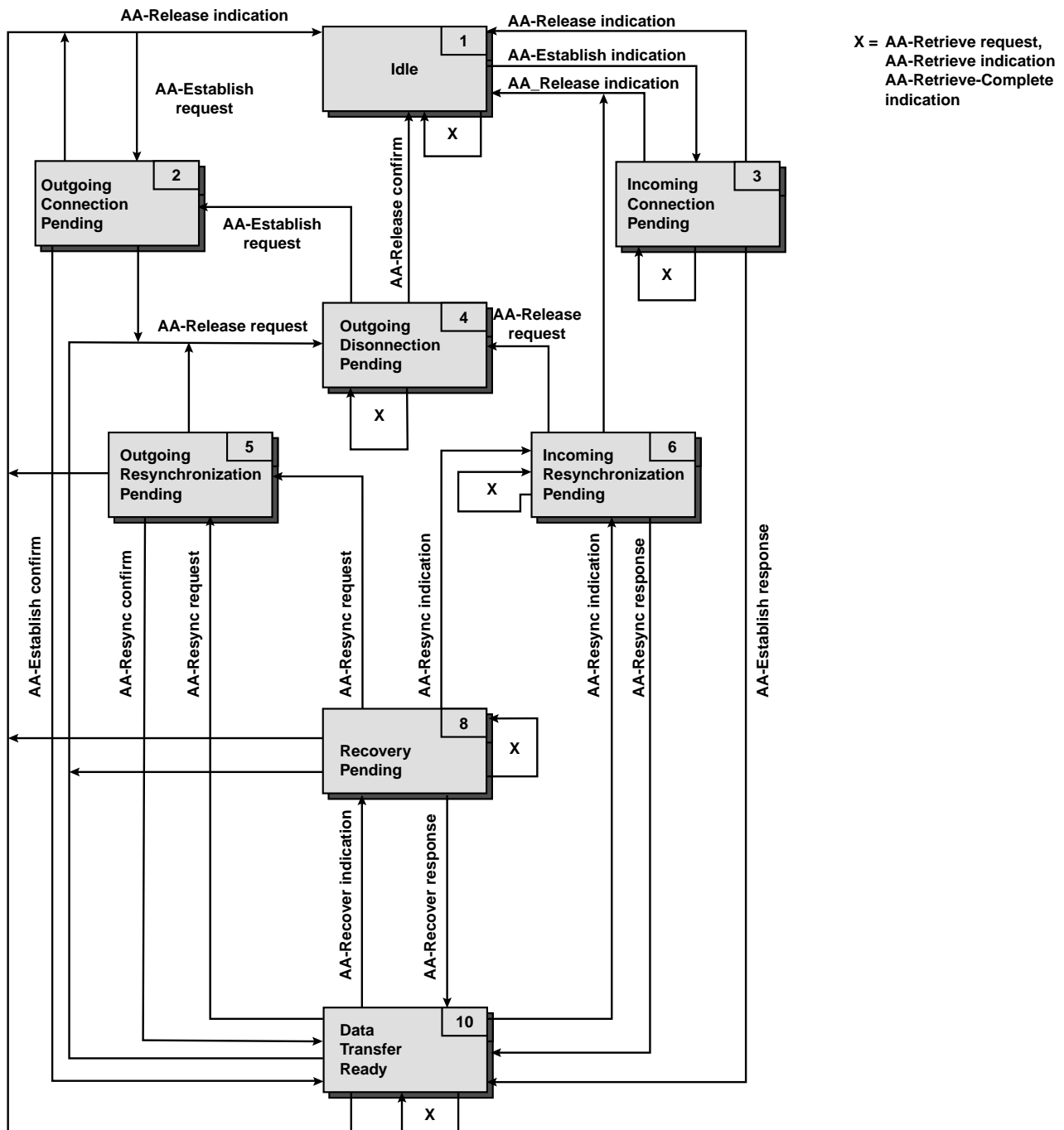


Abbildung 63 : Zustandsdiagramm bei Signalen zwischen SSOP und SSCF

Aufgrund der Komplexität des SSOP-Protokolls (vgl. Abbildung 63) wird an dieser Stelle lediglich eine Schilderung der wichtigsten Mechanismen innerhalb des Protokolls gegeben, eine detaillierte Beschreibung kann den Ablaufdiagrammen der Recommendation Q.2110 Figure 17/Q.2120 bis Figure

20/Q.2120 entnommen werden. Grundsätzlich werden alle Datenpakete (SD-PDUs) innerhalb der gesicherten Übertragung numeriert versendet. Bis zur Bestätigung des Empfanges der SD-PDU durch den Empfänger mittels einer STAT- oder USTAT-PDU hält der Sender Kopien der PDUs in seinen lokalen Puffern zurück.

Lokale Flußkontrolle

Stellt die SSCOP-Entity Engpässe bei der Übertragung der Datenpakete fest (etwa durch lange Aufenthaltszeiten von Datenpaketen im Sendepuffer) entscheidet sie, daß eine lokale *congestion* vorliegt und setzt den Transfer von SD-PDUs aus. Bis zum Abklingen des Engpasses werden ausschließlich Steuerpakete versendet, um die Verbindung aufrecht zu erhalten.

Flußkontrolle

Eine Datenflußkontrolle auf der SSCOP-Verbindung wird durch Einsatz eines *Fenstermechanismus*, wie er aus den älteren Datenübertragungsprotokollen bekannt ist (vgl. [IEEE802_2]), erreicht. Der Empfänger erlaubt dem Sender durch die Mitteilung der höchsten Sequenznummer, die er akzeptieren wird, alle SD-PDUs zwischen der bisherigen höchsten erlaubten Nummer und der neuen (*Übertragungsfenster*) zu senden. Für diese Anzahl von PDUs hält der Empfänger Pufferspeicher vor. Ein empfohlener Wert für diesen *Kreditrahmen* (*credit*) ist in Q.2110 Appendix IV zu finden. Durch diesen Mechanismus ist der Empfänger in der Lage, die Paketrage des Senders zu kontrollieren.

Selektive Wiederholung von PDUs

Anhand der Sequenznummern der empfangenen SD-PDUs kann der Empfänger das Fehlen einer oder mehrerer Datenpakete feststellen. Im Nutzlastfeld einer STAT- oder USTAT-PDU teilt der Empfänger dem Sender die Nummer der lückenlos empfangenen PDUs, die Nummern der fehlenden PDUs sowie den neuen *credit* mit. Timer auf der Sendeseite sorgen dafür, daß der Sender auch bei Verlust aller SD-PDUs, durch eine gezielte Statusabfrage mittels einer POLL-PDU Auskunft über verlorene SD-PDUs erhält. Beispiele für den Ablauf dieser *Error Recovery* in verschiedenen Situationen sind in Q.2110, Appendix II zu finden, wo auch die Details zur Numerierung bei den Wiederholungsanforderungen angegeben sind.

3.9 Signalisierung und Adressierung

Wie das leitungsvermittelte Fernsprechnet so setzt auch das B-ISDN einen expliziten Verbindungsaufbau vor dem Austausch der Nutzdaten voraus. Den Aufbau der Nutzdatenkanäle im B-ISDN übernehmen daher Protokolle, die eng mit den im Fernsprechnet eingesetzten Signalisierungsprotokollen Q.931 [Q931] und dem Signalisierungsverfahren Nr. 7 [Q.7xx] verwandt sind. Diese wurden um spezielle Funktionen erweitert, die zur Steuerung des B-ISDN notwendig sind. Im Bereich der Teilnehmer/Netz-Schnittstelle (UNI) wird die Signalisierung gem. ITU Rec. Q.2931 (eine Erweiterung der Rec. Q.931) vorgenommen. Im Netzzinnern (NNI) wird das gegenüber dem ISUP¹ modifizierte Protokoll B-ISUP, aufbauend auf dem Protokoll MTP-3, benutzt (siehe Abbildung 64).

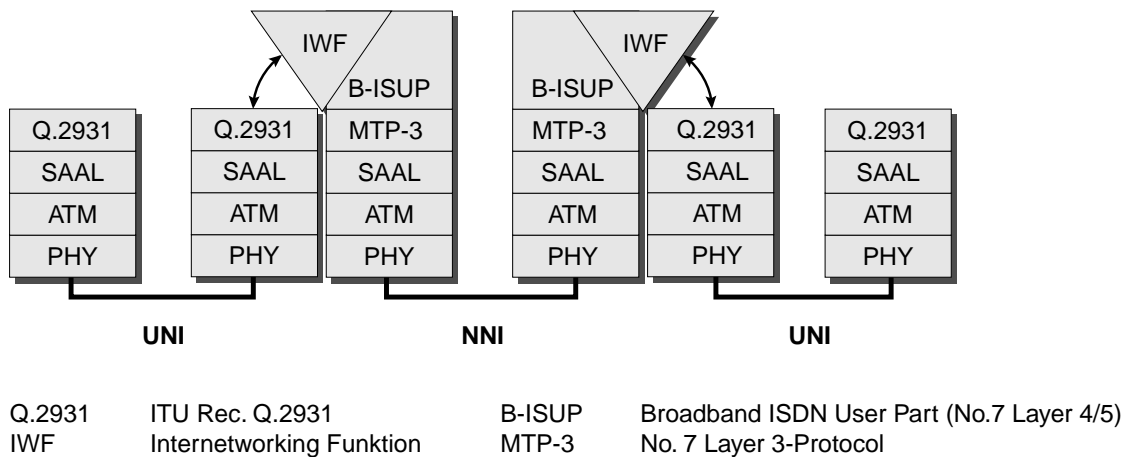


Abbildung 64 : Signalisierungsprotokolle im B-ISDN

3.9.1 Aufgaben der Signalisierung

Wegen der Vielzahl der im B-ISDN zu übertragenden Verkehrsklassen sind die Aufgaben der Signalisierung vielfältiger als diejenigen ihrer Pendanten aus dem digitalen Fernsprechnetzen:

Aufgabe
Aufbau Virtueller Kanal- und Pfadverbindungen
Einrichtung von Punkt-zu-Punkt und Punkt-zu-Mehrpunkt-Verbindungen
Aufbau von Verbindungen mit symmetrischen oder asymmetrischen Bandbreiteanforderungen
Unterstützung aller Verkehrsklassen im B-ISDN

Tabelle 47 : Aufgaben der Signalisierung im B-ISDN

1. ISDN User Part

Aufgabe
Definition eines systemübergreifend verständlichen Satzes von Dienstparametern
Abwicklung der Signalisierungsfunktionen durch Protokollnachrichten, Informationselemente und Protokollabläufe
Unterstützung eines fest definierten Außerbandsignalisierungskanals für alle Signalisierungsnachrichten
Unterstützung verschiedener Adreßformate zur Identifizierung der Verbindungsendpunkte
Verwaltung mehrerer paralleler Signalisierungsvorgänge am UNI
Definition eines systemübergreifend verständlichen Satzes von Dienstparametern
Zuweisung von Verbindungsnummern
Fehlerbehebung

Tabelle 47 : Aufgaben der Signalisierung im B-ISDN

3.9.2 Ablauf der Signalisierung

Eine vereinfachte Darstellung des Nachrichtenaustauschs bei Verbindungsauf- und -abbau im B-ISDN zeigt die folgende Abbildung:

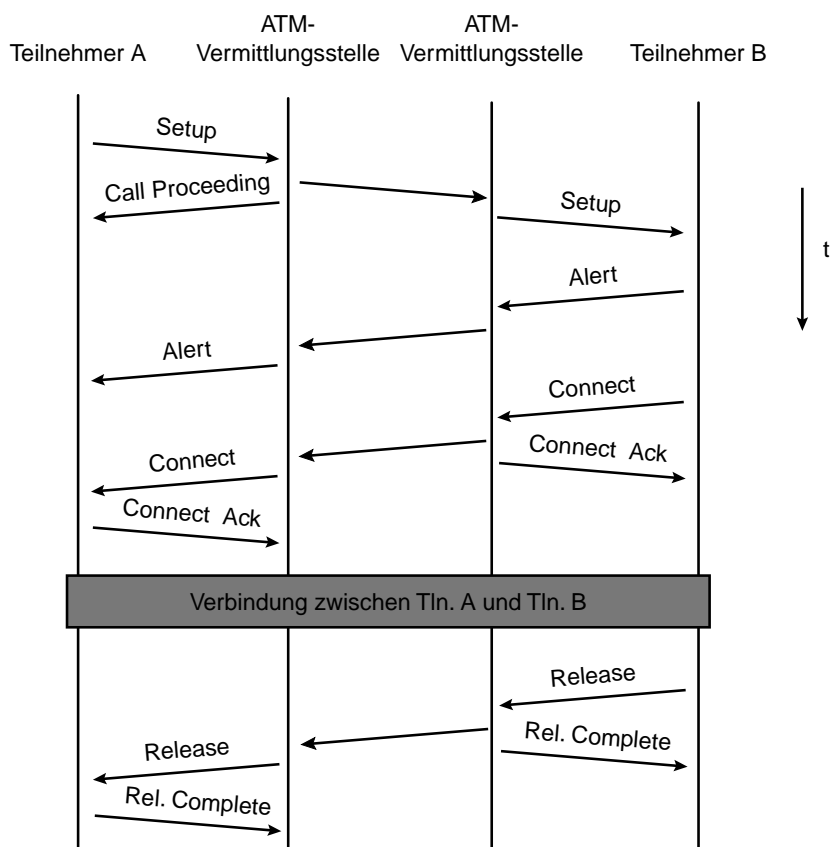


Abbildung 65 : Signalisierung zum Auf- und Abbau einer B-ISDN-Verbindung

Eine Verbindung wird von Teilnehmer A durch das Versenden einer SETUP-Nachricht initiiert. Die erste Vermittlungsstelle quittiert dies mittels der CALL PROCEEDING-Nachricht. Nachdem die Setup-Nachricht Teilnehmer B erreicht hat, wird zunächst eine ALERT-Nachricht zurückgesendet, um Teilnehmer A die Bearbeitung des Verbindungsaufbauwunsches anzuzeigen. Kann Teilnehmer B den Wunsch befriedigen, sendet er die Bestätigungsnachricht CONNECT. Nach Eintreffen der Bestätigungen CONNECT ACKNOWLEDGE beginnt der Datenaustausch. Der Verbindungsabbau kann durch beide Teilnehmer erfolgen. Er wird durch eine RELEASE-Nachricht eingeleitet und bei Vollendung durch RELEASE COMPLETE bestätigt.

Mit der SETUP-Nachricht werden die vom Teilnehmer A geforderten Eigenschaften der Nutzdatenverbindung gesendet. Kann Teilnehmer B und das Übertragungsnetzwerk diesen Ansprüchen genügen, wird die Verbindung aufgebaut; andernfalls wird sie abgelehnt oder Teilnehmer A werden die noch möglichen Verkehrsparameter angeboten.

Die Parameter der Nachrichten sowie eine detaillierte Definition der Abläufe innerhalb der Signalisierungsprotokolle sind in den ITU Empfehlungen [Q27xx, Q29xx] bzw. der UNI-Spezifikation des ATM-Forums¹ angegeben. Aus Übersichtlichkeitsgründen wird an dieser Stelle nur ein Ausschnitt aus den Elementen der SETUP-Message angegeben und erläutert, die vollständige Aufstellung ist im Anhang A.6 auf Seite 182 zu finden.

Nachrichtenelement	Richtung	Typ	Länge ^a
Protocol discriminator Kennzeichnung des verwendeten Protokolls	beide	M	1
Call reference Nummer zur Identifikation des Signalisierungsvorganges	beide	M	4
Message type Kennzeichnung des Nachrichtentyps (SETUP, CONNECT, RELEASE ...)	beide	M	2
AAL parameters Forderung der Verbindung an den AAL-Typ	beide	O	4-21
ATM traffic descriptor Verkehrsparameter der Verbindung, z.B. Zellrate	beide	M	12-30

Tabelle 48 : Elemente der SETUP-Nachricht

1. Zusammenschluß von Firmen außerhalb der Standardisierungsgremien (<http://www.atmforum.com>)

Nachrichtenelement	Richtung	Typ	Länge ^a
Broadband bearer capability Eigenschaften der Nutzverbindung (z.B. Bitrate, Zeitbindung), die von allen Netzknoten innerhalb der Verbindung akzeptiert werden müssen	beide	M	6-7
Called party number Identifikation des Zielsystems Adreßformat : E.164 [E164] oder ISO NSAP gem. X.213 [X213]	beide	M	4-25
...			
M O beide	Mandatory (erforderlich) Optional (optional) Vom Tln. zum Netzwerk und umgekehrt		

Tabelle 48 : Elemente der SETUP-Nachricht

a. Oktetts

Die Bedeutung und Kodierung der variablen Informationselemente unterliegen der Standardisierung durch die ITU. Die vollständige Definition aller Elemente kann der ITU Recommendation I.2931 bzw. den Vorschlägen des ATM-Forums zur UNI-Spezifikation [UNI31] entnommen werden. Als Beispiel sind im Anhang die Informationselemente *AAL parameters* (siehe Tabelle 59 im Anhang auf Seite 184) zur Definition der Parameter einer AAL-Typ1-Entity und der *ATM-Traffic Descriptor* (Tabelle 60 im Anhang auf Seite 185) angegeben. Durch das Informationselement *AAL parameters* ist eine vollständige Beschreibung des Funktionsumfangs gegeben, den die geforderte AAL-Typ1-Entity erbringen muß. Mithilfe des Informationselements *ATM-Traffic Descriptor* gibt die Nachrichtenquelle eine möglichst zutreffende Beschreibung des von ihr zu erwartenden *Zellstromes*. Aufgrund dieser Angaben entscheidet die *Connection Admission Control* des B-ISDN über die Annahme der Verbindung.

Die Algorithmen, die zur Akzeptanz bzw. Ablehnung der so spezifizierten Verbindung führen, liegen im Verantwortungsbereich des Netzbetreibers. Hier stehen sich das Bestreben des Netzbetreibers, seine Ressourcen möglichst gut auszunutzen und die dann bei hohen Lasten zurückgehende Qualität der Verbindungen diametral gegenüber.

3.10 Zusammenfassung

Das aktuelle Kapitel vermittelt einen ersten Eindruck von der Komplexität der Vorgänge bei der Datenübertragung im B-ISDN, die mithilfe des von der ITU verbindlich vorgegebenen B-ISDN-Referenzmodells beschrieben wurden. Aus der Ungleichbehandlung der Nutzerdaten durch die einzelnen AAL-Typen wird bereits der entscheidende Einfluß dieser Protokolle auf den beim Empfänger eintreffenden Datenstrom deutlich. So hat beispielsweise der Verlust einer Zelle auf dem ATM-Layer je nach eingesetztem AAL-Typ unterschiedlich starke Auswirkungen. Während eine AAL-Instanz vom Typ 1 die verlorene Zelle rekonstruieren kann, verliert eine Instanz des AAL-Typs 3/4 u.U. ein Paket von bis zu 65535 Oktett, d.h. etwa 1.490 weitere Zellen. Aufgrund dieser Effekte sollte die Beurteilung von Steuerungsmechanismen nicht ausschließlich aufgrund ihrer Leistungsfähigkeit auf dem Niveau des ATM-Layers beurteilt werden.

Die innerhalb des Referenzmodells standardisierten Abläufe sind herstellerunabhängig in den unterschiedlichen Ausführungen der Protokollinstanzen einzuhalten. Berechnungen dieser Vorgänge müssen sich daher möglichst eng an diesen Vorgaben orientieren, um ein realitätsnahes Ergebnis zu liefern. In den folgenden Kapiteln wird gezeigt werden, mit welchen Mitteln die Vorgänge innerhalb solcher Systeme berechnet werden können, und unter welchen Einschränkungen diese Ergebnisse Gültigkeit haben.

Traffic Control und Congestion Control gemäß ITU

4.1 Zielsetzung

Das B-ISDN ist daraufhin ausgelegt, eine Reihe von Verkehrsklassen mit unterschiedlichsten Anforderungen an das Transportnetz zu übertragen, und den Nutzern die einmal zugesicherte Übertragungsqualität für die gesamte Dauer der Verbindung zu garantieren. Unvorhersehbare Fehlerzustände oder statistische Schwankungen in den Datenströmen können jedoch dazu führen, daß das Übertragungsnetzwerk zeitweise nicht in der Lage ist, die zugesicherte Qualität aufrecht zu erhalten (*Congestion*).

Die *Congestion Control CC* umfaßt alle Maßnahmen zur zeitlichen und räumlichen Begrenzung der Auswirkungen einer *Congestion*; Aufgabe der *Traffic Control TC* ist es, durch präventive Maßnahmen zu vermeiden, daß solche Situationen auftreten. Die in diesem Zusammenhang von der ITU in Recommendation I.371 ([I371], Abschnitt 4.2) definierten Funktionen sind in Tabelle 49 zusammengestellt.

Funktion	Aufgabe
Network Resource Management (NRM)	Reservierung von Netzwerkressourcen, um Datenströme so zu führen, daß ihre Dienstanforderungen erfüllt sind
Connection Admission Control (CAC)	Alle Maßnahmen, die während der Verbindungsaufbauphase zur Entscheidung über die Akzeptanz einer VCC oder VPC führen

Tabelle 49 : Funktionen von TC und CC im B-ISDN

Funktion	Aufgabe
ATM Layer Resource Management (RM)	Änderung der Zuordnung von Ressourcen zu Verbindungen durch den ATM Layer
Usage/Network Parameter Control (UPC/NPC)	Überwachung bestehender Datenströme am UNI und NNI, dadurch Schutz des Netzes und bestehender Verbindungen vor Überlastsituationen, die absichtlich oder unabsichtlich durch andere Benutzer erzeugt werden

Tabelle 49 : Funktionen von TC und CC im B-ISDN

Die *Connection Admission Control* entscheidet anhand statistischer Analysen des Zellverkehrs auf dem Niveau des ATM-Layers, ob es möglich ist, einen Verbindungswunsch anzunehmen und die darin geforderte Qualität zu erbringen. Durch diese Maßnahme sollen das Netz und die bereits bestehenden Verbindungen vorausschauend vor Überlast geschützt werden.

4.2 Connection Admission Control gem. ITU Rec. I.371

4.2.1 Traffic Contract

Die folgende Abbildung zeigt die Situation im B-ISDN-Netzwerk beim Eintreffen eines Verbindungsaufbauwunsches, ausgelöst von einem potentiellen Nutzer (*User*).

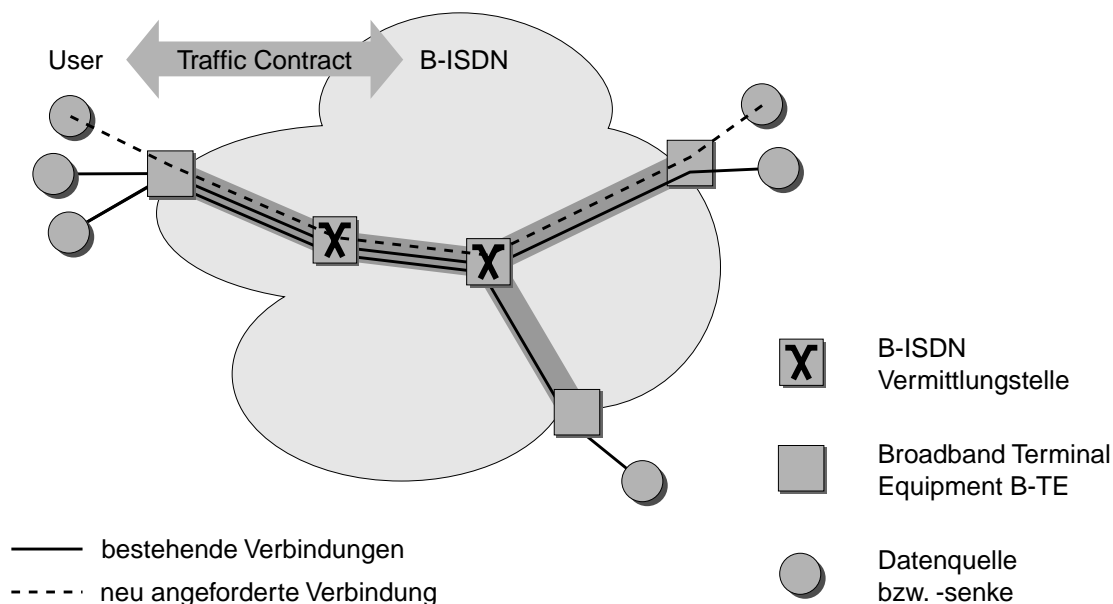


Abbildung 66 : Verbindungsaufbau im B-ISDN

Nutzer und Netzbetreiber schließen beim Verbindungsaufbau einen Vertrag, den sog. *Traffic Contract*. Dieser verpflichtet den Netzbetreiber, die vereinbarte Übertragungsqualität für die gesamte Dauer der Verbindung zu garantieren. Dem Nutzer schreibt er die Anerkennung der von ihm beim Verbindungsaufbau im *Traffic Descriptor* (vgl. Tabelle 48 auf Seite 92) angegebenen Verkehrsparameter vor.

Jeder von einem Nutzer erzeugte Datenstrom weist spezielle *QoS-Anforderungen* an das Netzwerk auf. Diese Anforderungen sind als objektiv erfaßbare Kenngrößen eines Übertragungsnetzwerks in der ITU Recommendation I.356 [I356] definiert. Es handelt sich hierbei um die Zellverlustwahrscheinlichkeit (*Cell Loss Probability CLP*), die Zellübertragungsdauer (*Cell Transfer Delay CTD*) und die Schwankungen in der Zellübertragungsdauer (*Cell Delay Variation CDV*). Da durch diese Parameter unendlich viele unterschiedliche Ansprüche formuliert werden können, definiert I.356 eine Auswahl an Parameterkombinationen als *QoS classes*, die zur Charakterisierung der Ansprüche gängiger Datenquellen ausreichend sind. Die so beim Verbindungsaufbau vereinbarte *QoS class* stellt den ersten Teil des *Traffic Contract* dar.

Im *Traffic Descriptor* beschreibt der Nutzer die Eigenschaften des von ihm verursachten Verkehrs als eine Liste von *Traffic Parameters*. Hierzu gehören die maximale Zellrate (*Peak Cell Rate PCR*), die mittlere Zellrate (*Sustainable Cell Rate SCR*), die Toleranzparameter bzgl. CDV, die *Burstiness* und maximale Anzahl von Zellen mit Spitzenzellrate (*Maximum Burst Size MBS*). Für die Traffic Parameter gelten diskrete Werte (vgl. I.371, Abschnitt 5.4 [I371]), die während des Signalisierungsvorgangs vom Benutzer zum Netzwerk übertragen (vgl. Tabelle 60 im Anhang auf Seite 185) werden. Auch sie bilden einen Bestandteil des *Traffic Contracts*.

Den dritten Vertragsbestandteil bildet die *ATM Transfer Capability ATC*. Sie definiert die Funktionsumfang, die der Netzbetreiber dem Benutzer zur Verfügung stellt, um dessen Datenstrom mit möglichst geringem Qualitätsverlust zu transportieren. Sie stellt einen ersten Ansatz dar, die unterschiedlichen Qualitätsansprüche der Datenströme im Rahmen der Verbindungsannahmestrategie zu berücksichtigen. Die Realisierung der *ATC* liegt im Verantwortungsbereich des Netzbetreibers, die ITU macht lediglich Vorschläge zum prinzipiellen Vorgehen bei der Implementierung. Fünf *ATC-Klassen*, die im folgenden kurz erläutert werden, sind derzeit von der ITU definiert: *DBR*, *SBR*, *ABR*, *ABT/DT* und *ABT/IT*. Eine ausführliche Beschreibung ist in I.371, Abschnitt 5.5 [I371] zu finden.

4.2.2 ATM Transfer Capability ATC

4.2.2.1 Deterministic Bit Rate Transfer Capability (DBR)

DBR wird für Verbindungen eingesetzt, die eine zeitlich unveränderliche Übertragungsrate erzeugen. Der Verkehr dieser CBR-Quellen wird ausschließlich durch Angabe der Peak Cell Rate für User- und OAM-Zellen beschrieben. Aus Sicht der Connection Admission Control läßt sich das Problem durch die in Abbildung 67 gezeigte Anordnung beschreiben.

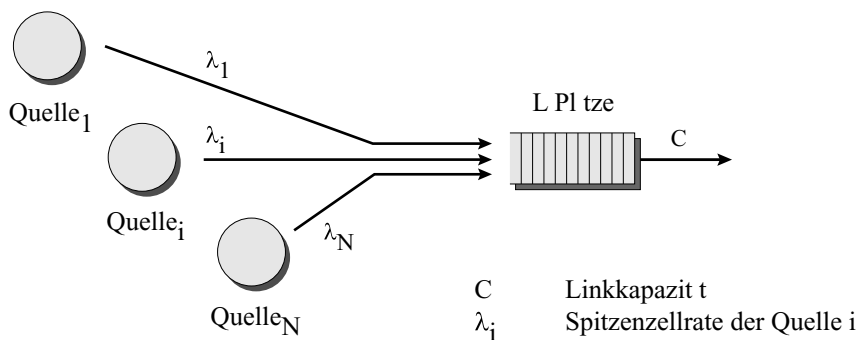


Abbildung 67 : N Quellen mit konstanten Zellraten an einem ATM-Netznoten

Ein solches System kann näherungsweise durch ein Bediensystem $M/D/1/L$ oder $ND/D/1/L$ beschrieben werden. Bei Kenntnis der Puffergröße L und Vorgabe einer maximalen Zellverlustwahrscheinlichkeit CLP ergeben sich, abhängig vom verwendeten Modell, folgende Abschätzungen für die maximal zulässige Last ρ_{\max} (siehe [Pit96], Kap. 6) :

$M/D/1/L$:

$$\rho_{\max} = \frac{2 \cdot L}{2 \cdot L - \ln(CLP)} \quad (Gl. 23)$$

$ND/D/1/L$:

$$\rho_{\max} = \frac{2 \cdot L \cdot N}{2 \cdot L \cdot N - (2 \cdot L^2 + N \cdot \ln(CLP))} \quad (Gl. 24)$$

Dabei gilt:

$$\rho_{\max} = \sum_{i=1}^N \lambda_i / C. \quad (Gl. 25)$$

Für eine neu hinzukommende Verbindung mit einer Spitzenzellrate von $\lambda_{\max, N+1}$ ergeben sich aus Gl. 23 bzw. Gl. 24 Ungleichungen, die zur Durchführung der Verbindungsannahme herangezogen werden können:

M/D/1/L:

$$\rho_{\text{ges}} = \frac{\lambda_{N+1}}{C} + \sum_{i=1}^N \frac{\lambda_i}{C} \leq \frac{2 \cdot L}{2 \cdot L - \ln(\min\{\text{CLP}_i\})} \quad (\text{Gl. 26})$$

ND/D/1/L:

$$\rho_{\text{ges}} = \frac{\lambda_{N+1}}{C} + \sum_{i=1}^N \frac{\lambda_i}{C} \leq \frac{2 \cdot L \cdot (N+1)}{2 \cdot L \cdot (N+1) - [2 \cdot L^2 + (N+1) \cdot \ln(\min\{\text{CLP}_i\})]} \quad (\text{Gl. 27})$$

Erfüllt die neu hinzukommende Verbindung N+1 diese Ungleichung, so kann sie akzeptiert werden, andernfalls muß sie abgelehnt werden. Wie Gl. 26 und Gl. 27 zeigen, ist hier lediglich die Kenntnis der Spitzenzellrate λ_i und der Zellverlustwahrscheinlichkeit CLP_i aller bestehenden Verbindungen notwendig. Ausschlaggebend für die Entscheidung ist die Verbindung mit der geringsten zulässigen Verlustwahrscheinlichkeit, was zu einer relativ geringen Auslastung der weiterführenden Strecke führt. Eine Verbesserung, im Sinne einer höheren Systemauslastung, kann beispielsweise durch Vergabe unterschiedlicher Prioritäten an die Verkehrsströme erreicht werden (siehe [Pit96] Kap.11).

4.2.2.2 Statistical Bit Rate Transfer Capability (SBR)

SBR ist für Datenquellen vorgesehen, die keine konstanten Datenraten erzeugen. Über deren Eigenschaften muß dementsprechend mehr als die Spitzenzellrate bekannt sein. Bei diesem Verfahren wird der statistische Gewinn beim Multiplex mehrerer bursthaltiger Quellen über einen gemeinsam genutzten Pufferspeicher ausgenutzt (vgl. Abschnitt 2.6.2 auf Seite 31). Die fehlenden Einschränkungen bezüglich der von den einzelnen Quellen erzeugten Datenströme verhindern jedoch eine allgemeingültige analytische Berechnung mithilfe eines Bediensystems. Um die Beschreibung zu vereinheitlichen definiert die ITU eine *Äquivalente Quelle* (*Equivalent Source*, siehe Abbildung 68 auf Seite 100), welche die Eigenschaften der Datenquellen aller bestehenden Verbindungen in sich vereinen soll. Durch diese Ersatzquelle soll die Durchführung der Connection Admission Control mithilfe ähnlicher Beziehungen wie in Gl. 26 oder Gl. 27 ermöglicht werden.

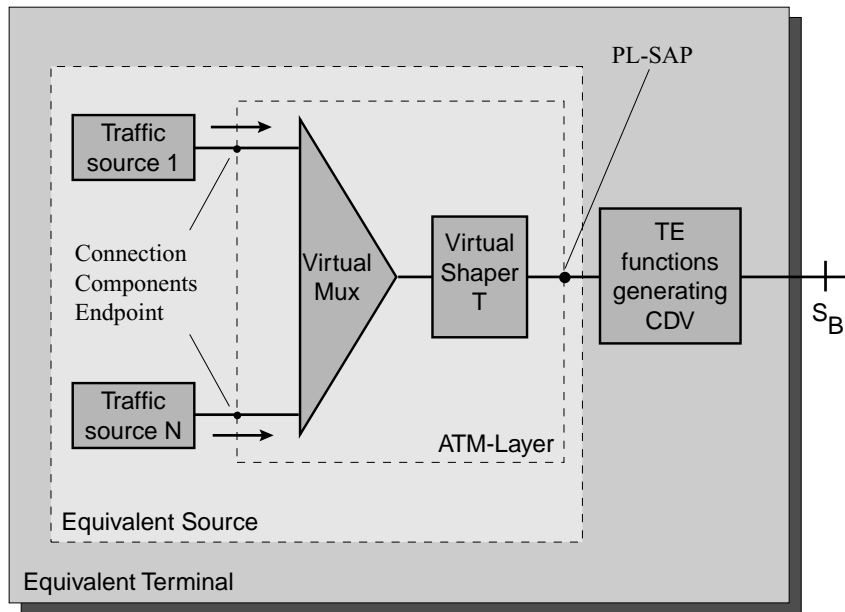


Abbildung 68 : Äquivalente Quelle gem. Figure5/1.371 [1371]

Die Ermittlung der statistischen Eigenschaften der Äquivalenten Quelle stellt den Netzbetreiber jedoch vor Schwierigkeiten. Zur Zeit wird zur Beschreibung der ursprünglichen Quellen ein eingeschränkter Parametersatz verwendet, der lediglich aus der Spitzenzellrate, der mittleren Zellrate und der Maximalanzahl von aufeinanderfolgenden Zellen in einem Burst mit Spitzenzellrate besteht. Die Angabe dieser Parameter impliziert eine Modellierung aller Quellen durch *ON/OFF-Quellen* (vgl. Abbildung 14 auf Seite 28).

In diesem Fall sieht sich die Connection Admission Control mit folgender Situation konfrontiert:

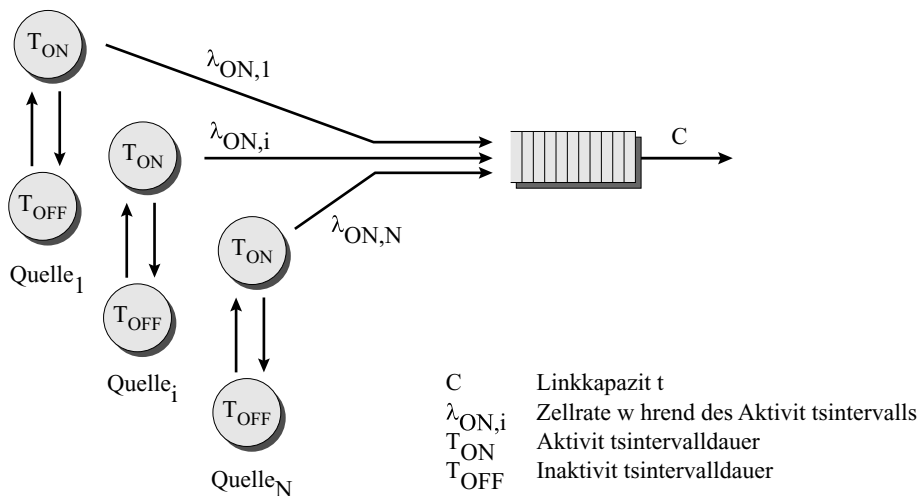


Abbildung 69 : ON/OFF-Quellen

Die Quellen verfügen über zwei Zustände, denen jeweils eine bestimmte Zellrate zugeordnet ist. Übersteigt die Summe aller Quellraten während der Aktivitätsphasen die Linkkapazität C , kann beim zufälligen Zusammentreffen mehrerer Aktivitätsintervalle ein Pufferüberlauf und ein daraus resultierender Datenverlust auftreten (siehe Abbildung 70 auf Seite 101).

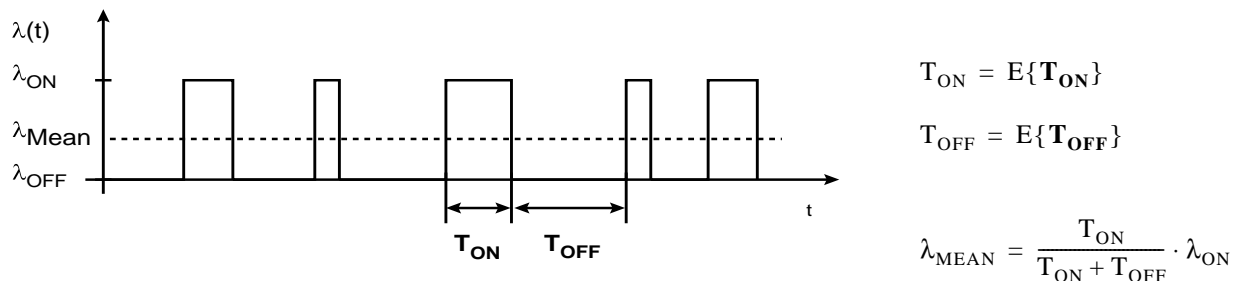


Abbildung 70 : Datenraten $\lambda(t)$ einer ON/OFF-Quelle

Bei dem Versuch, eine Verbindungsannahmestrategie mithilfe ähnlicher Beziehungen wie in Gl. 26 oder Gl. 27 auf Seite 99 zu formulieren, stellt sich hier die Frage nach der zugrundezulegenden Zellrate λ_i . Setzt man jeweils die Spitzenrate $\lambda_{\text{ON},i}$ an, liefert der Algorithmus eine sehr hohe Abschätzung der bereits bestehenden Last. Demzufolge werden Verbindungswünsche bereits früh abgelehnt. Die Folge ist ein geringer Multiplexgewinn und eine relativ schwache Auslastung des Übertragungsnetzwerks. Legt man eine niedrigere Rate (z.B. die mittlere Rate $\lambda_{\text{MEAN},i}$) zugrunde, werden häufiger Verbindungen akzeptiert, woraus sich eine höhere Auslastung des Netzwerks ergibt. Jedoch treffen auch häufiger die Aktivitätsphasen vieler Quellen zusammen, was zu einer erhöhten Zellverlustwahrscheinlichkeit und damit einer reduzierten Datenqualität führt.

4.2.2.3 ATM Block Transfer Capability with Delayed Transmission (ABT/DT)

Der ABT-Mechanismus unterteilt den unregelmäßigen Zellstrom in Blöcke mit jeweils konstanten Zellraten. Für jeden dieser Blöcke wird eine erneute Verhandlung entsprechend der Methode DBR mit dem Netzwerk initiiert. Die Variante ABT/DT setzt das Einverständnis des Netzwerks mit der verlangten Zellrate im Block (*Block Cell Rate BCR*) voraus.

4.2.2.4 ATM Block Transfer Capability with Immediate Transmission (ABT/IT)

Diese Variante des ABT beginnt sofort mit dem Senden des Zellblocks, ohne eine positive Antwort des Netzwerks auf die Ressourcenanforderung abzuwarten. Dadurch verursachte Überlastzellen dürfen vom Netzwerk verworfen werden.

4.2.2.5 Available Bit Rate Transfer Capability (ABR)

Einige Anwendungen können ihre Datenrate bei Netzengpässen ohne Qualitätsverlust reduzieren. Mittels Rückmeldungen über den internen Netzzustand regelt das Netzwerk das Datenaufkommen der Quelle. Bei Netzengpässen wird die Datenrate reduziert, bei frei werdenden Ressourcen erlaubt das Netzwerk das Senden mit höherer Rate.

4.3 Referenzkonfiguration

Die CAC-Mechanismen, die in der ITU Rec. I.371 [I371] empfohlen werden, beziehen sich ausschließlich auf den ATM Layer, wie das entsprechende Referenzmodell in Abbildung 71 auf Seite 102 zeigt.

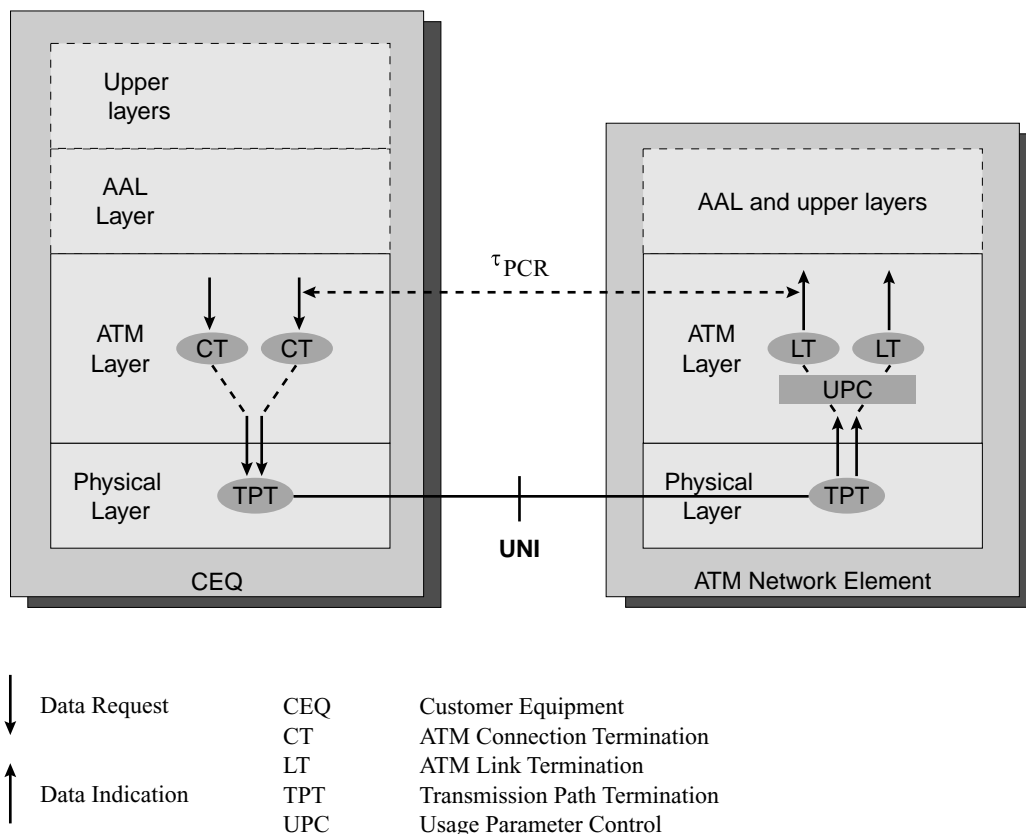


Abbildung 71 : Referenzkonfiguration der ITU bzgl. Traffic Parameter (aus : Fig4/I371 [I371])

Ausgehend von Approximationen des Zellverkehrs auf dem Niveau des ATM-Layers werden Modelle für die Zellströme entwickelt, um durch Berechnungen der zugrundegelegten Bediensysteme Maximalwerte für die zulässige Anzahl gleichzeitiger Verbindungen festzulegen.

4.4 Zusammenfassung

Die Connection Admission Control im B-ISDN entscheidet während der Verbindungsaufbauphase oder späterer Neuverhandlungen auf dem Niveau des ATM Layers über die Annahme oder Verweigerung eines Verbindungswunsches. Sie bedient sich dabei der Informationen über die Quellcharakteristik, die verlangte Dienstqualität und den Funktionsumfang des Übertragungsnetzes (siehe Abbildung 72 auf Seite 103). Die Realisierung des CAC-Algorithmus obliegt der Verantwortung des Netzbetreibers, der aufgefordert ist, leistungsfähige Algorithmen mit hohem Multiplexgewinn zu entwickeln.

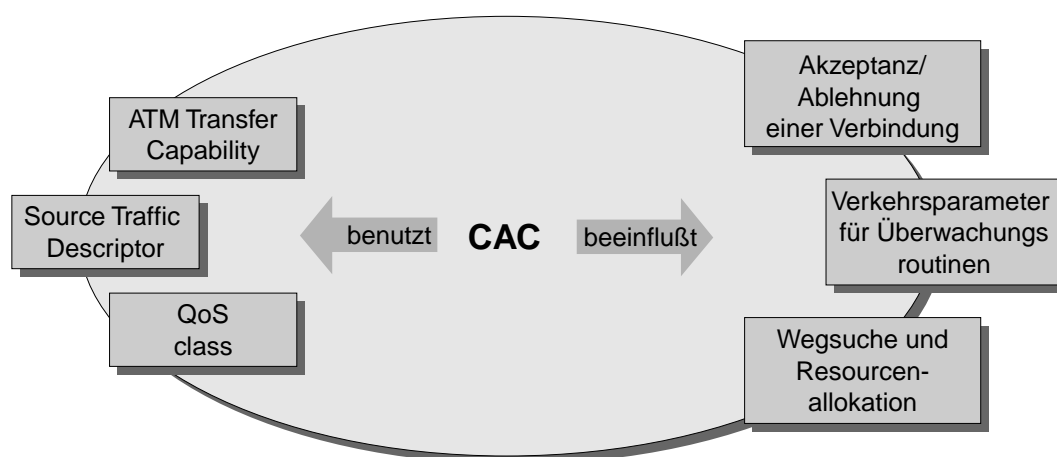


Abbildung 72 : Überblick über die CAC

Die von der ITU definierte Connection Admission Control bezieht sich zur Zeit ausschließlich auf die Zellströme im ATM-Layer (vergl. Abbildung 71 auf Seite 102). Zur Charakterisierung des Zellverkehrs der bereits im Netz bestehenden Verbindungen wird eine *Äquivalente Quelle* definiert, deren Bestimmung jedoch insbesondere bei bursthaltigem Verkehr mit großen Schwierigkeiten verbunden ist. Bisherige Vorschläge zur Realisierung der Connection Admission Control beruhen auf unterschiedlichen Annahmen bezüglich des Zellverkehrs in bestimmten Lastsituationen. Mithilfe der Bedientheorie werden darauf aufbauende, analytisch formulierte Verbindungsannahmestrategien entwickelt.

Als Alternative zu diesem Vorgehen wurde in der vorliegenden Arbeit ein adaptives Verfahren entwickelt, das sich nicht auf die Beurteilung der Datenqualität auf dem ATM-Layer beschränkt. Durch dessen Anwendung werden die Probleme bei der Definition einer geeigneten *Äquivalenten Quelle*

vermieden, die sich bei dem Versuch, einen analytisch formulierten Zusammenhang zwischen zulässiger Last und erreichbarer Qualität zu formulieren, zwangsläufig ergeben.

Adaptive Connection Admission Control

5.1 Motivation

Die CAC-Mechanismen, die in der ITU Rec. I.371 [I371] empfohlen werden, beziehen sich ausschließlich auf den ATM Layer, wie das entsprechende Referenzmodell in Abbildung 71 auf Seite 102 zeigt. Ausgehend von Approximationen des Zellverkehrs werden Modelle für die Zellströme angesetzt, um durch Berechnungen der zugrundegelegten Bediensysteme Maximalwerte für die zulässige Anzahl gleichzeitiger Verbindungen festzulegen.

Für den Netzbetreiber verbleibt das Problem einer geeigneten Definition der *Äquivalenten Quelle* für die VBR-Quellen in allen Netzknoten, um eine optimale Auslastung seiner Betriebsmittel bei gleichzeitig hoher Qualität der Nutzdatenströme zu erreichen. Die Ansiedelung der Call Admission Control auf dem ATM-Layer hat den Vorteil, daß eine zumindest näherungsweise analytische Berechnung der Vorgänge möglich wird (siehe z.B. [Tur92], [Sai91_2], [Tak91], [Gan95]). Die auf diese Weise festgelegten CAC-Funktionen beziehen sich jedoch nur mittelbar auf die vom Anwender generierten Datenströme. Zwischen der Steuerungsebene auf dem Niveau des ATM-Layers und der zu optimierenden Größe (Qualität der von den Anwenderinstanzen innerhalb der *Higher Layers* generierten Datenströme) befinden sich u.a. die Instanzen des ATM Adaption Layers. Eine Beurteilung der erreichten *QoS* auf dem ATM-Layer des Empfängers erlaubt nur Aussagen über die Qualität des Zellstromes. Eine Aussage über die Datenqualität in der Anwenderschicht kann daraus strenggenommen nur bei Verwendung einer Anpassungsschicht vom AAL-Typ 0 (vgl. Abschnitt 3.8.1 auf Seite 64) abgeleitet werden. Dies zeigt bereits eine Betrachtung zum Verhalten der unterschiedlichen AAL-Typen beim Verlust von Zellen oder bei Schwankungen in den Empfangs-

zeitpunkten. Während Instanzen vom AAL-Typ 1 eine verlorene Zelle mithilfe ihrer Vorwärtsfehlerkorrekturmechanismen (siehe Kapitel 3.8.2.4 auf Seite 66 ff.) rekonstruieren können, verwerfen Instanzen des AAL-Typs 3/4 daraufhin eine vollständige CPCS-PDU mit bis zu 64 kByte Nutzdaten. Dieser Effekt verstärkt sich zusätzlich bei Verlust „sensibler“ Pakete, etwa BOM- oder EOM-PDUs (vgl. Abbildung 56 auf Seite 79). Eine Instanz der S-AAL wiederum kann die verlorenen Daten, mithilfe von Wiederholungsanforderungen zu Lasten einer hohen Verzögerung, neu anfordern. Während sich Schwankungen in den Übertragungszeiten bei Instanzen vom AAL-Typ 1 aufgrund der Empfangspuffer und der Algorithmen zur Clock Recovery erst nach einiger Zeit auf den Datenempfang beim Nutzer auswirken, verfügen die übrigen AAL-Typen nicht über derartige Mechanismen.

Der in dieser Arbeit gewählte Ansatz geht daher über die ausschließliche Beurteilung der Übertragungsqualität auf Zelleniveau hinaus. Der Grundgedanke des von mir entwickelten und im folgenden beschriebenen Verfahrens ist es, die Beurteilung der Datenqualität beim Benutzer der AAL durchzuführen, und durch ständige Beobachtung der Datenqualität einen an die aktuelle Situation im Netz angepassten CAC-Algorithmus abzuleiten.

5.2 Referenzkonfiguration

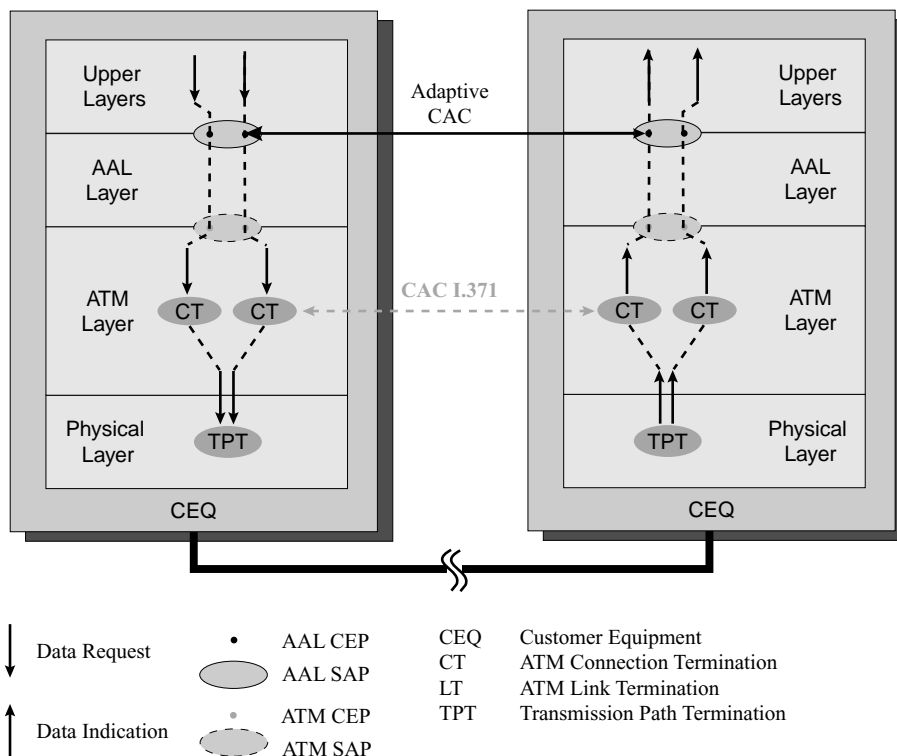
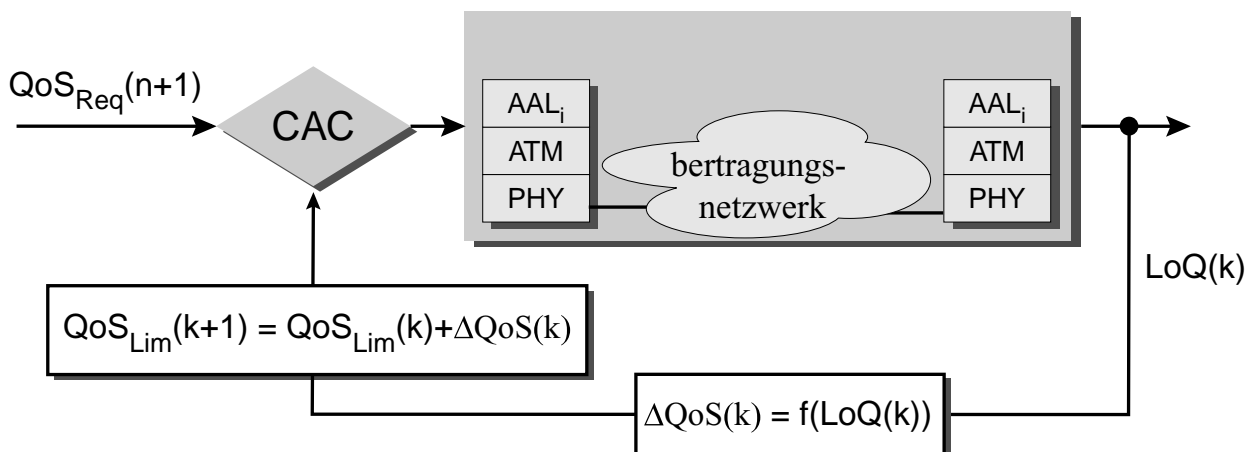


Abbildung 73 : Referenzkonfiguration Adaptive CAC

Die Modifikationen des von der ITU vorgeschlagenen Verfahrens spiegeln sich u.a. in der Referenzkonfiguration für die Connection Admission Control wieder. Abbildung 73 auf Seite 106 zeigt diese Veränderungen gegenüber der Referenzkonfiguration der ITU aus Abbildung 71 auf Seite 102.

5.3 Prinzipielles Vorgehen

Den prinzipiellen Ablauf dieses adaptiven Algorithmus zeigt Abbildung 74.



$QoS_{Req}(n)$:	Vektor der Qualitätsansprüche der Verbindung n an das Übertragungssystem
$QoS_{Lim}(k+1)$:	Vektor der vom Übertragungssystem nach Abbau der Verbindung k erfüllbaren Qualitätsansprüche
$\Delta QoS(k)$:	Änderung des Vektors der erfüllbaren Qualitätsansprüche nach Abbau der Verbindung k
$LoQ(k)$:	Qualitätsverlust der Daten der Verbindung k

Abbildung 74 : Prinzip der adaptiven CAC

Er beruht im Gegensatz zu den bekannten Algorithmen nicht auf einem bestimmten analytischen Berechnungsverfahren zur Bestimmung der zulässigen Last im Netzwerk, sondern auf einer iterativen Annäherung an diese Belastungsgrenze. Zur Modifikation des zulässigen Verkehrsprofils beim Sender werden die vom Empfänger bezüglich der einzelnen *Traffic Parameter* festgestellten Qualitätseinbußen am AAL-SAP herangezogen. Die ständige Anpassung dieser Höchstgrenzen sorgt so für die Zulassung einer optimalen Verkehrsmischung zur Teilnahme am Datenverkehr im Netz.

5.4 Berechnungsgrundlagen der Adaptiven CAC

5.4.1 Bezeichnungen

Jeder Datenquelle wird in diesem Modell ein Vektor $\overrightarrow{QoS_{Req}}$ mit N_Q Parametern zugeordnet, der ihre Anforderungen an das Übertragungsnetzwerk beschreibt:

1. Spitzenbitrate PBR
2. Mittlere Bitrate MBR
3. Bitfehlerwahrscheinlichkeit BEP
4. Paketverlustwahrscheinlichkeit PLP
5. Mittlere Paketverzögerung MPD
6. Varianz der Paket-Zwischenankunftszeiten CMD2
7. 3. Zentralmoment der Paket-Zwischenankunftszeiten CMD3
8. 4. Zentralmoment der Paket-Zwischenankunftszeiten CMD4

Höhere Werte für einen Parameter sollen sich in größeren Zahlenwerten des jeweiligen Parameters ausdrücken, daher werden in den Berechnungen die Kehrwerte der Parameter 3 bis 8 verwendet. Die während der Übertragung erreichte Qualität wird mit $\overrightarrow{QoS_{Att}}$ bezeichnet. Der Vektor $\vec{\epsilon}_\tau$ beschreibt die relative Empfindlichkeit einer Datenquelle vom Typ τ gegenüber Qualitätseinbußen in jeder Komponente.

5.4.2 Berechnungsgrundlagen

Qualitätsverlust

Zur Kennzeichnung des Qualitätsverlustes, den die Daten einer Quelle bei der Übertragung über das Netzwerk erfahren, dient der Vektor \overrightarrow{LoQ} ¹. Seine Komponenten berechnen sich durch die relative quadratische Abweichung des erreichten vom angestrebten Wert:

1. Loss of Quality

$$\text{LoQ}[i] = \begin{cases} \frac{(\text{QoS}_{\text{Req}}[i] - \text{QoS}_{\text{Att}}[i])^2}{\text{QoS}_{\text{Req}}[i]^2} & \text{für } \text{QoS}_{\text{Att}}[i] \leq \text{QoS}_{\text{Req}}[i] \\ 0 & \text{sonst} \end{cases} \quad (\text{Gl. 28})$$

Der mittlere Qualitätsverlust wird berechnet durch Mittelwertbildung über alle Komponenten unter Berücksichtigung der jeweiligen Empfindlichkeit :

$$\overline{\text{LoQ}} = E\{\text{LoQ}\} = \frac{1}{N_Q} \cdot \sum_{i=1}^{N_Q} \varepsilon_\tau[i] \cdot \text{LoQ}[i]. \quad (\text{Gl. 29})$$

Aufgrund dieser Definition bewegt sich der Wert von $\overline{\text{LoQ}}$ immer zwischen den Grenzen 0 und 1.

Wert einer Verbindung

Als Bezugsgröße zum späteren Vergleich der Datenströme wird ein Referenzverkehr mit den Qualitätsansprüchen $\overrightarrow{\text{QoS}_{\text{Ref}}}$ festgesetzt. Der Wert w_k einer Verbindung k im Sinne eines Verbindungsentgeltes das der Netzbetreiber seinem Kunden in Rechnung stellen kann, wird durch

$$w_k = \frac{1}{N_Q} \cdot \sum_{i=1}^{N_Q} \varepsilon_\tau(i) \cdot (1 - \text{LoQ}[i]) \cdot \left(\frac{\text{QoS}_{\text{Req}}[i]}{\text{QoS}_{\text{Ref}}[i]} \right) \quad (\text{Gl. 30})$$

festgelegt.

Wertdurchsatz

Als Vergleichskriterium zwischen den verschiedenen CAC-Verfahren wird der Wert aller während eines Zeitintervalls T akzeptierten Verbindungen N_{Acc} definiert:

$$\bar{w} = \frac{1}{T} \cdot \sum_{k=1}^{N_{\text{Acc}}} w_k \quad (\text{Gl. 31})$$

5.4.3 Ablauf der Adaptiven CAC

Verbindungsaufbau

Beim Aufbau einer neuen Verbindung prüft die adaptive CAC bei Sender und Empfänger, ob die aktuellen Grenzwerte QoS_{Lim} durch die Annahme der Verbindung überschritten würden. Eine Verbindung $n+1$ wird angenommen, wenn bei Sender und Empfänger folgende Ungleichung für alle $1 \leq k \leq N_Q$ erfüllt ist:

$$\sum_{i=1}^n QoS_{Req,i}[k] + QoS_{Req,n+1}[k] \leq QoS_{Lim}[k] \quad (Gl. 32)$$

Die gleiche Überprüfung findet mit den entsprechenden Verkehrsparametern für die entgegengesetzte Übertragungsrichtung statt. Werden beide Verbindungen angenommen, beginnt die Datentransferphase.

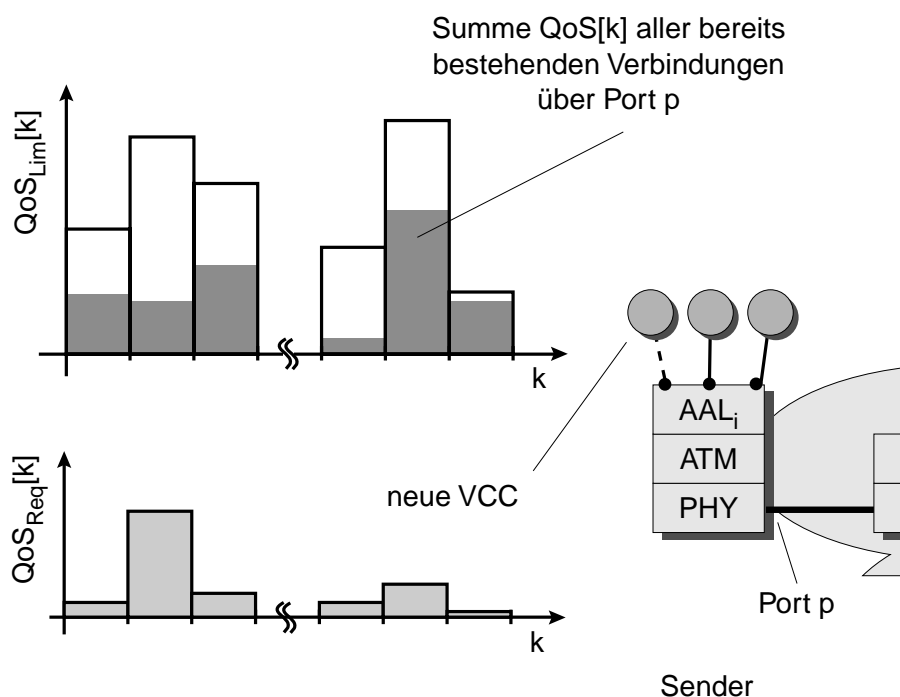


Abbildung 75 : Adaptive CAC beim Verbindungsaufbau

Datentransferphase

Die Quellen innerhalb senden Datenpakete entsprechend den von ihnen beim Verbindungsaufbau angegebenen Verkehrsparametern. Die Empfängerinstanzen nehmen die Daten in Empfang und erfassen die aufgetretenen Bitfehler, Paketverluste und Zwischenankunftszeiten am AAL-SAP.

Verbindungsabbau

Vor dem Abbau der Verbindung führt die Endstation der Virtual Channel Connection die Berechnung der erreichten Datenqualität durch. Liegt der mittlere Qualitätsverlust gem. Gl. 29 auf Seite 109 unterhalb eines vom Netzbetreiber zu definierenden Wertes $\overline{LoQ_{Max}}$, werden die Obergrenzen für die QoS-Parameter um die soeben erfolgreich übertragenen Werte erhöht (siehe Abbildung 76 auf Seite 111).

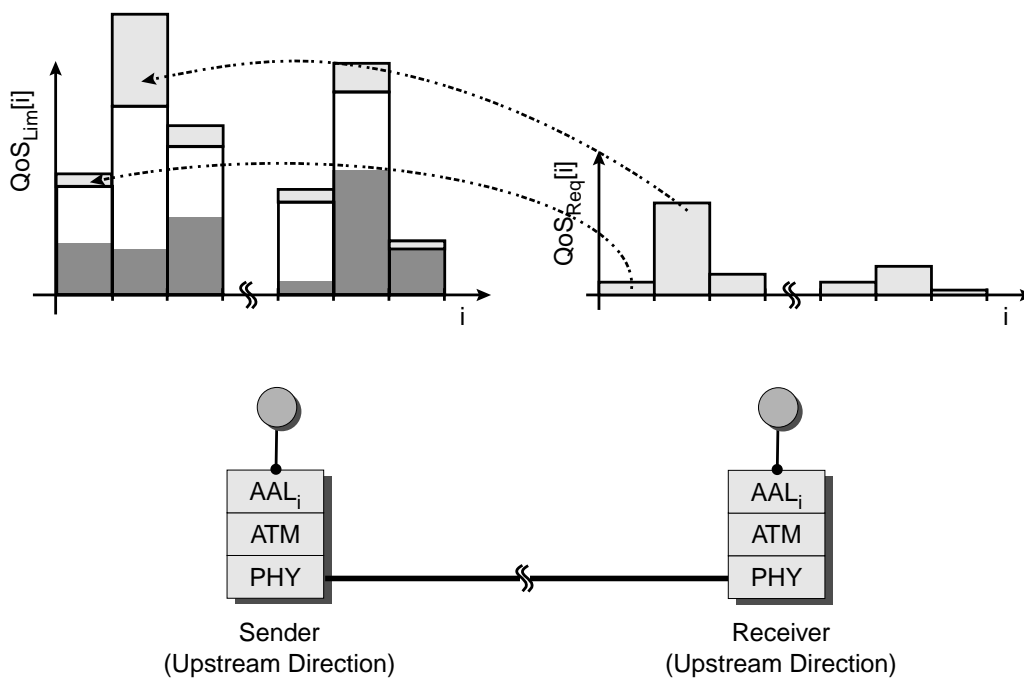


Abbildung 76 : Erhöhen der Grenzwerte für die QoS-Parameter

Ist aufgrund hoher Lasten im Netzinnern die Beeinträchtigung der Qualität höher als $\overline{LoQ_{Max}}$, erfolgt eine Absenkung der Grenzwerte jeder Komponente proportional zur jeweils erlittenen Beeinträchtigung um $LoQ[k] \cdot QoS_{Req}[k]$ für alle $1 \leq k \leq N_Q$. Dies führt bei zukünftigen Verbindungsaufbauten zur Ablehnung der Verbindungen, die hohe Ansprüche bezüglich bereits stark beeinträchtigter Komponenten aufweisen.

Ziel dieses Regelalgorithmus ist das Auffinden einer für den momentan im Übertragungsnetzwerk vorherrschenden Lastfall optimalen Grenzwertkurve $\overrightarrow{QoS_{Lim}}$. Diese Kurve strebt im stationären Fall für jede Komponente einen konstanten Wert an. Im instationären Fall häufig wechselnder Quellcharakteristiken paßt sich die Kurve dem jeweiligen Lastfall an.

5.5 Zusammenfassung

Als Alternative zu der von der ITU in der Recommendation I.371 [I371] definierten Verfahren zur Connection Admission Control auf dem Niveau des ATM-Layers wurde in der vorliegenden Arbeit ein adaptives Verfahren entwickelt, das sich nicht auf die Beurteilung der Datenqualität auf dem ATM-Layer beschränkt. Es berücksichtigt zusätzlich den starken Einfluß der AAL-Protokolle auf die Qualität der zwischen den Endteilnehmern übertragenen Daten. Die Probleme bei der Definition einer entsprechenden *Äquivalenten Quelle* werden durch Einsatz eines im Verlauf der Arbeit entwickelten Algorithmus vermieden, der sich ständig an die aktuelle Leistungsfähigkeit des Netzwerks anpaßt, ohne einen analytisch formulierten Zusammenhang zwischen zulässiger Last und erreichbarer Qualität vorauszusetzen. Das Verfahren ist daher sehr flexibel, und auch für heute noch nicht berücksichtigte Verkehrsprofile einsetzbar. Ein weiterer Vorteil des Verfahrens ist die ausschließliche Betrachtung von Sender und Empfänger am Ende der Virtual Channel Connection; das Netz selbst wird als „Black Box“ betrachtet. Engpässe im Netz zeigen sich in Beeinträchtigungen der Komponenten des Qualitätsvektors, die Ursachen sind für die Verbindungsannahmestrategie nur von untergeordnetem Interesse.

Die Leistungsfähigkeit dieses adaptiven Algorithmus wird durch Simulation unterschiedlicher ATM-Netzwerke mithilfe der im nächsten Kapitel geschilderten Software nachgewiesen.

6.1 Simulationsziel

Die in Kapitel 2 beschriebenen Methoden der Verkehrstheorie zur Analyse realer Nachrichtennetzwerke sind nur unter vereinfachenden Annahmen bezüglich der Ankunfts- und Bedienprozesse auf konkrete Szenarien anwendbar. Der Einfluß der Protokolle, die zusätzliche Abhängigkeiten in die statistischen Eigenschaften der Datenströme einbringen, bleibt dabei ebenso unberücksichtigt wie die Netztopologie und der interne Aufbau der Netzknoten. Simulationsstudien, die ohne solche einschränkenden Annahmen durchgeführt werden, können daher aussagekräftigere numerische Ergebnisse liefern.

Ziel der hier vorgestellten Simulation ist die Optimierung der Connection Admission Control im B-ISDN (siehe Kapitel 4) bezüglich der pro Zeiteinheit über das Netzwerk übertragenen Menge qualitativ hochwertiger Daten. Da reale Betriebsbedingungen berücksichtigt werden sollten, ist das Simulationsmodell eng an die tatsächlichen Gegebenheiten in ATM-Netzwerken angelehnt. Alle Protokollinstanzen innerhalb der Simulationssoftware wickeln die von der ITU vorgeschriebenen Schichtenprotokolle ab. Die zur Durchführung der Protokollschritte benötigte Zeit wird durch den Einsatz detaillierter Modelle für die Hardware der Netzknoten bestimmt. Die verschiedenen Hardwaremodelle sind unabhängig von den Protokollinstanzen implementiert. Sie können jederzeit gegeneinander ausgetauscht oder modifiziert werden, ohne in die Implementierung der Protokollabläufe ändern zu müssen.

6.2 Simulationstechnik

6.2.1 Zeitdiskrete ereignisorientierte Simulationen

Obwohl Vorgänge in Nachrichtennetzwerken, wie etwa die Signalausbreitung auf Übertragungsleitungen *kontinuierlicher* Natur sind, wird im Rahmen der Simulation solcher Systeme häufig nur mit *diskreten* Zeitpunkten gearbeitet. Diese Einschränkung stellt das Ergebnis des Abstraktionsprozesses bei der Bildung des *Modells* für das zu simulierende System dar. Bei einer Simulation von Warteschlangensystemen sind die zeitkontinuierlichen physikalischen Vorgänge bei der Datenübertragung nicht von Interesse. Es genügt die Kenntnis des Zeitpunktes, zu dem Daten vollständig an einer Bedieneinrichtung oder Warteschlange eingetroffen sind und weiterverarbeitet werden können. Im Hinblick auf die Zielsetzung einer solchen Simulation genügt es demnach, die zu *diskreten Zeitpunkten* auftretenden wichtigen *Ereignisse* zu erfassen und in Abhängigkeit vom *Systemzustand* die aus ihnen zu einem späteren Zeitpunkt resultierenden *Folgeereignisse* zu bestimmen. Dies ist das Prinzip der *zeitdiskreten, ereignisorientierten Simulation*, die sich durch sprunghafte Änderungen sowohl der simulierten Zeitpunkte als auch der Systemzustände von der zeitkontinuierlichen Simulation eines durch Differentialgleichungen beschriebenen physikalischen Vorgangs unterscheidet. Den typischen Ablauf eines Simulationsschrittes im Rahmen einer ereignisorientierten Simulation zeigt die folgende Abbildung.

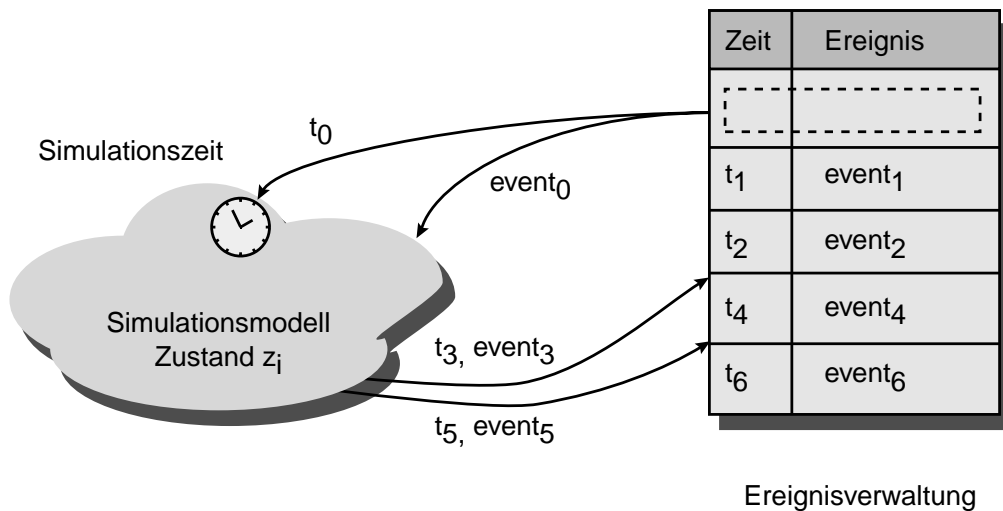


Abbildung 77 : Simulationsschritt

Das Ereignis „event₀“ soll zum Zeitpunkt t_0 im simulierten System eintreten. Zunächst wird die Simulationszeit auf t_0 eingestellt, danach bearbeitet das Simulationsmodell das Ereignis entsprechend dem Zustand z_i , in dem es sich zu diesem Zeitpunkt befindet. Die Folgeereignisse (hier „event₃“ und

„event₅“, die zu den Zeitpunkten t_3 und t_5 auftreten sollen) ordnet eine Ereignisverwaltung anhand des Zeitpunktes ihres Auftretens in ihre Liste der noch ausstehenden Ereignisse ein. Danach ist dieser Schritt beendet und die Ereignisverwaltung gibt für den nächsten Simulationsschritt das Ereignis mit dem kleinsten „Zeitstempel“ t_i aus ihrer Liste an das Simulationsmodell ab. Als Beispiel zeigt die folgende Abbildung einen Ausschnitt aus der Simulation eines Wartesystems M/D/1 (vgl. Abschnitt 2.3.1 auf Seite 20).

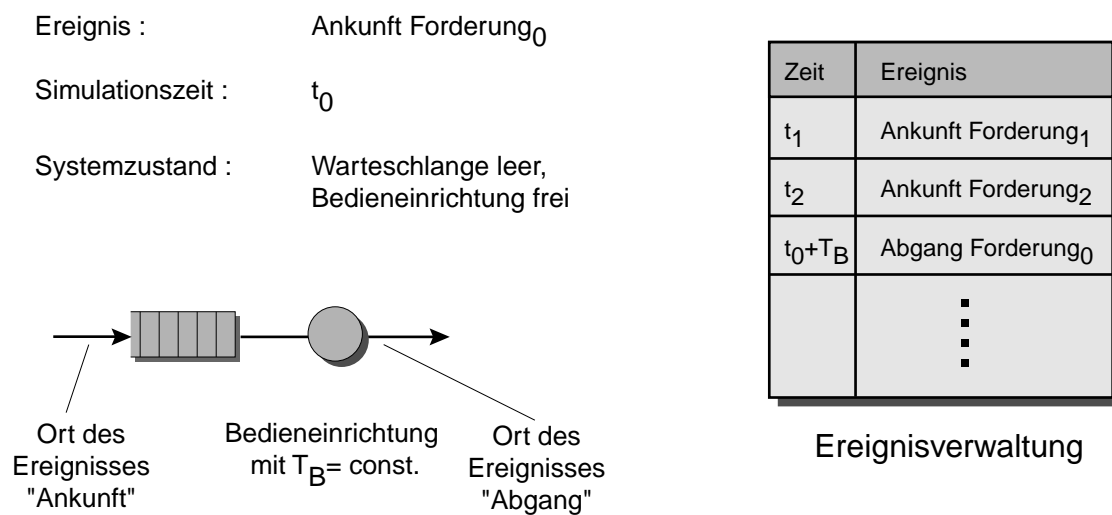


Abbildung 78 : Simulation eines Wartesystems M/D/1

In dem gezeigten Systemzustand kann eine zum Zeitpunkt t_0 eintreffende Forderung sofort bedient werden. Da die Bedieneinrichtung nun bis zum Zeitpunkt t_0+T_B mit der Bearbeitung der Forderung 0 beschäftigt ist, müssen die Forderungen 1 und 2, die vor Beendigung der Bedienung zu den Zeitpunkten t_1 und t_2 eintreffen, in der Warteschlange zwischengepuffert werden. Sie können von der Bedieneinrichtung erst nach dem Zeitpunkt t_0+T_B bedient werden.

Eine statistisch verlässliche numerische Bestimmung der Parameter des Abgangsprozesses eines Bediensystems erfordert die Erwartungswertbildung über die Ergebnisse einer Vielzahl von Durchläufen. Eines der wichtigsten Ziele beim Entwurf einer Simulationssoftware ist daher die Minimierung der zur Berechnung der Folgeereignisse benötigte Rechenzeit. Diese Vorgabe darf jedoch bei größeren Anordnungen nicht zu Lasten einer klaren Softwarestruktur gehen.

6.2.2 Programmiersprache

6.2.2.1 Anforderungen

Aufgrund der Anforderungen an die Programmlaufzeit und -struktur bietet sich die Hochsprache C++ als Programmiersprache an. Sie erweitert die „hardwarenahe“ Programmiersprache C um objektorientierte Elemente (vgl. [Str94]). Sprachelemente aus C machen die Implementierung laufzeitoptimierter Algorithmen, die objektorientierten Erweiterungen das Design einer klar strukturierten Software möglich. Umfang und Laufzeit der Simulation erforderten eine leistungsfähige und stabile Rechnerplattform. Eingesetzt wurde daher die Entwicklungssoftware *C++-Workshop* mit Arbeitsgruppenfunktionalität auf UNIX-Workstations vom Typ SUN UltraSparc. Das Tool *TeamWorks* unterstützte dabei die auf mehrere Programmierer verteilte Softwareentwicklung, die mit dem C++-Compiler SUN ProCompiler CC in der Version 4.2 durchgeführt wurde.

6.2.2.2 Wichtige Eigenschaften der Programmiersprache C++

Direkte Speicherzugriffe

Die Programmiersprache C++ erlaubt (im Gegensatz zu anderen „Nachfolgern“ von C wie JAVA) nach wie vor den direkten Speicherzugriff über *Pointer* aus der Programmiersprache C. Dadurch können zeit- aufwendige Kopieroperationen zwischen häufig benötigten Daten vermieden werden, was sich positiv auf die Programmlaufzeit auswirkt.

Klassen

In diesen benutzerdefinierten Strukturen stellen Datenelemente und die zu ihrer Bearbeitung notwendigen Funktionen (Methoden) eine abgeschlossene Einheit dar. Eigenschaften und Funktionalitäten einer Klasse können *vererbt*, d.h. an andere, von ihnen *abgeleitete* Klassen weitergegeben werden. Die abgeleitete Klasse übernimmt den gesamten Leistungsumfang ihrer *Basisklasse*, und kann durch zusätzliche eigene Daten oder Bearbeitungsroutinen deren Funktionalität erweitern.

Polymorphie

Der Aufruf von Funktionen erfolgt in C++ durch die Angabe des Funktionsnamens und des aufrufenden Klassenobjekts. Gleichartige Funktionsnamen innerhalb unterschiedlicher Klassen sind demnach möglich; die Aufrufe können während des Compilationsvorgangs oder zur Programmlaufzeit eindeutig

zugeordnet werden. Durch sinnvollen Einsatz dieses Sprachelementes kann der Programmcode für andere Programmierer eines Entwicklerteams verständlicher gestaltet werden.

Containerklassen

C++ bietet die Möglichkeit der Definition sog. *Containerklassen*. In diesen sind Methoden definiert, ohne den Typ der zu verarbeitenden Daten zu spezifizieren. Diese fehlenden Angaben werden beim Aufbau einer Instanz dieser Klasse „nachgereicht“, der notwendige Programmcode wird vom Compiler generiert. Der Code wird dadurch kompakter und vielseitig einsetzbar.

Durch diese Eigenschaften wird die Programmiersprache C++ zu einer geeigneten Softwareplattform für die Entwicklung und Durchführung zeitdiskreter, ereignisorientierter Simulationen komplexer Übertragungssysteme.

6.3 Ablaufprinzip der Simulation

Den prinzipiellen Ablauf der Simulation zeigt Abbildung 79 auf Seite 118. Die Zielsetzung der Simulation verlangt die Untersuchung einer vorgebbaren Mischung des Verkehrsaufkommens multimediale Datenquellen zwischen zwei Endstationen eines ATM-Netzwerks unter variablen Lastbedingungen. Die einzelnen Verkehrsströme werden durch Datenquellen mit vorgebbarer Charakteristik (vgl. Abschnitt 2.6 auf Seite 26) erzeugt. Nach Beendigung des Datentransfers einer Quelle findet im Empfänger eine Analyse der empfangenen Daten statt, um die erreichte *Quality of Service* zu beurteilen. Das Ergebnis dieser Analyse wird zur Modifizierung der *Call Admission Control* genutzt.

6.3.1 Initialisierung des Simulationsmodells

Die Zuordnung aller Parameter zu den Elementen des Simulationsmodells (Stationen, Protokollinstanzen, usw.) erfolgt dateigestützt bei Simulationsstart, so daß Parameterstudien ohne Eingriffe in den Quellcode erfolgen können. Der Start der Simulation wird bei inkonsistenten oder unvollständigen Angaben unterbunden.

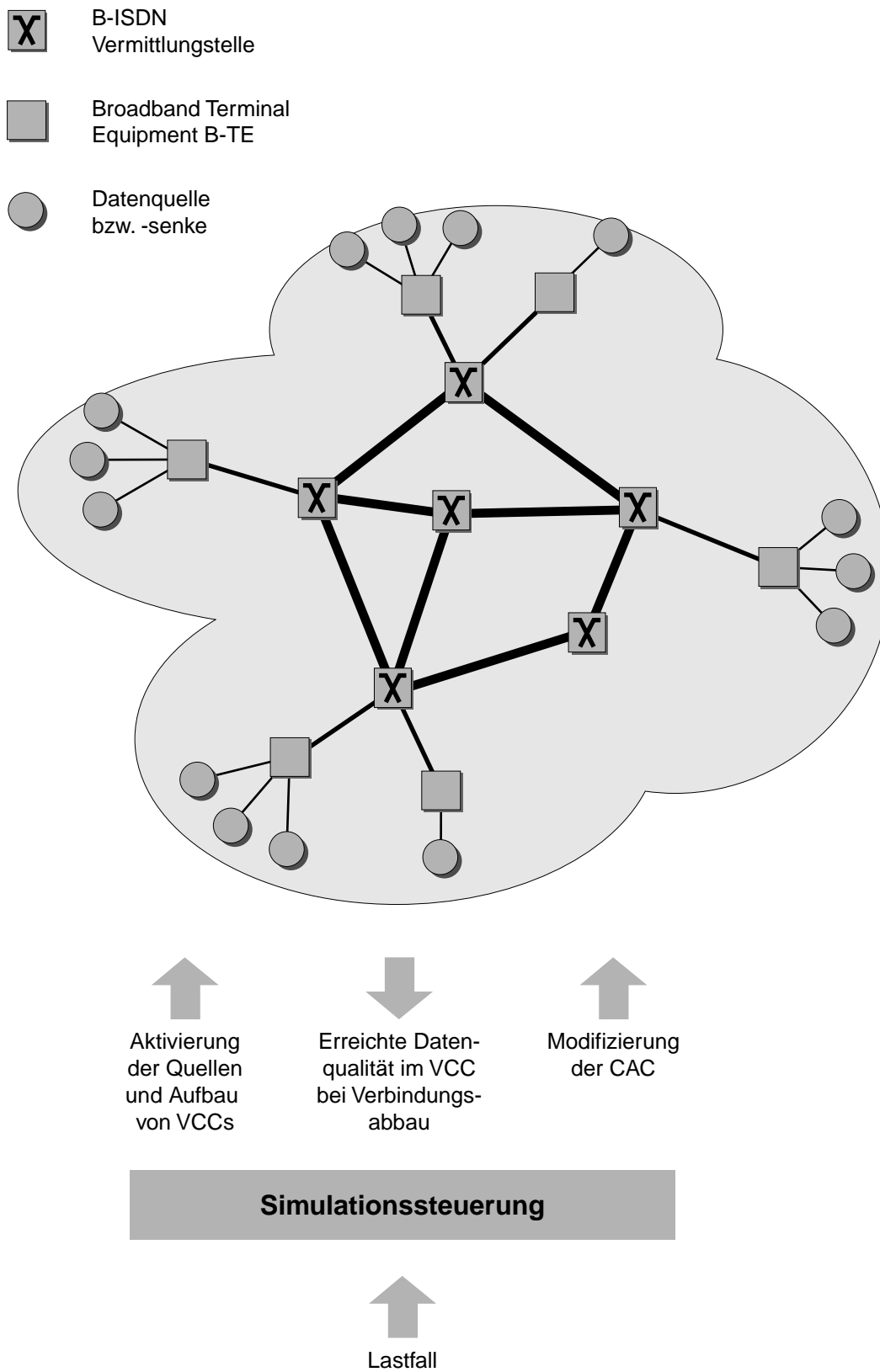


Abbildung 79 : Prinzipieller Ablauf der Simulation

6.3.2 Durchführung der Simulation

6.3.2.1 Verbindungsaufbau

Alle Datenquellen werden zu Beginn des Simulationslaufs erzeugt. Sie veranlassen mittels Zufallszahlengeneratoren in unterschiedlichen zeitlichen Abständen Datenübertragungen. Die Rate, mit der diese Verbindungsaufbauten initiiert werden, ist abhängig von einem bei Programmaufruf zu übergebenden Lastparameters. Bevor der Datentransfer stattfinden kann, muß jedoch eine Verbindung zwischen den beteiligten Stationen aufgebaut werden. Das *Network Management Center* prüft die Verfügbarkeit eines Weges zwischen Sende- und Empfangsstation und der für diesen Datentyp geeigneten AAL-Instanzen (vgl. Tabelle 25 auf Seite 62). Der Verbindungsaufbau erfolgt bidirektional; die Verkehrscharakteristik für beide Übertragungsrichtungen ist in den Definitionen der auslösenden Datenquellen festgelegt.

6.3.2.2 Datentransfer

Die der Verbindung zugeordnete Quelle beginnt gemäß ihrer Charakteristik Datenpakete zum Empfänger zu versenden. Der Zeitpunkt des Datentransfers, Länge und Füllmuster der Daten werden zufällig erzeugt. Eine Kopie jedes abgesandten Datenpakets wird verbindungsbezogen zurückbehalten, um in der empfangenden Protokollinstanz die während der Übertragung aufgetretenen Datenverluste und -verfälschungen analysieren zu können.

6.3.2.3 Analyse des Datenstroms im Empfänger

Abbildung 80 auf Seite 120 zeigt beispielhaft die Situation bei einer fehlerbehafteten Übertragung von Nutzdaten. Zur Bestimmung der Anzahl von Bitfehlern und Paketverlusten werden die beim Empfänger eintreffenden Pakete mit den im Kopienpuffer enthaltenen Datenpaketen verglichen. Stimmen Bitmuster und Länge des zuletzt eingetroffenen Pakets mit dem ersten Eintrag im Kopienpuffer überein, gilt das Datenpaket als fehlerfrei übertragen. Weist das Paket $n \leq n_{Fmax}$ ¹ Bitfehler auf, entscheidet die Analyse, daß n Bitfehler aufgetreten sind. Andernfalls wird Datenverlust angenommen und das eingetroffene Paket nacheinander mit den restlichen Einträgen im Kopienpuffer verglichen. Findet sich im Puffer ein Paket, das nach den obigen Kriterien dem ursprünglich versendeten entspricht, so werden alle vergeblich getesteten Pakete als verloren gewertet. Bei einer Übertragung

1. $n_{Fmax} = 1 + \text{Paketlänge in Bit} * 10^{-3}$

über den AAL-Typ 1 haben die Datenpakete eine Länge von nur einem Bit, daher werden im Empfänger nach Entfernen der Füllbits Hilfspakete der Länge einer SAR-PDU-Payload (vgl. Abbildung 45 auf Seite 65) gebildet und mit den Einträgen im Kopienpuffer verglichen. Bei Datenverlust werden durch bitweisen Vergleich der Hilfspakete mit dem Senderbitstrom die Anzahl verlorener Bits bestimmt.

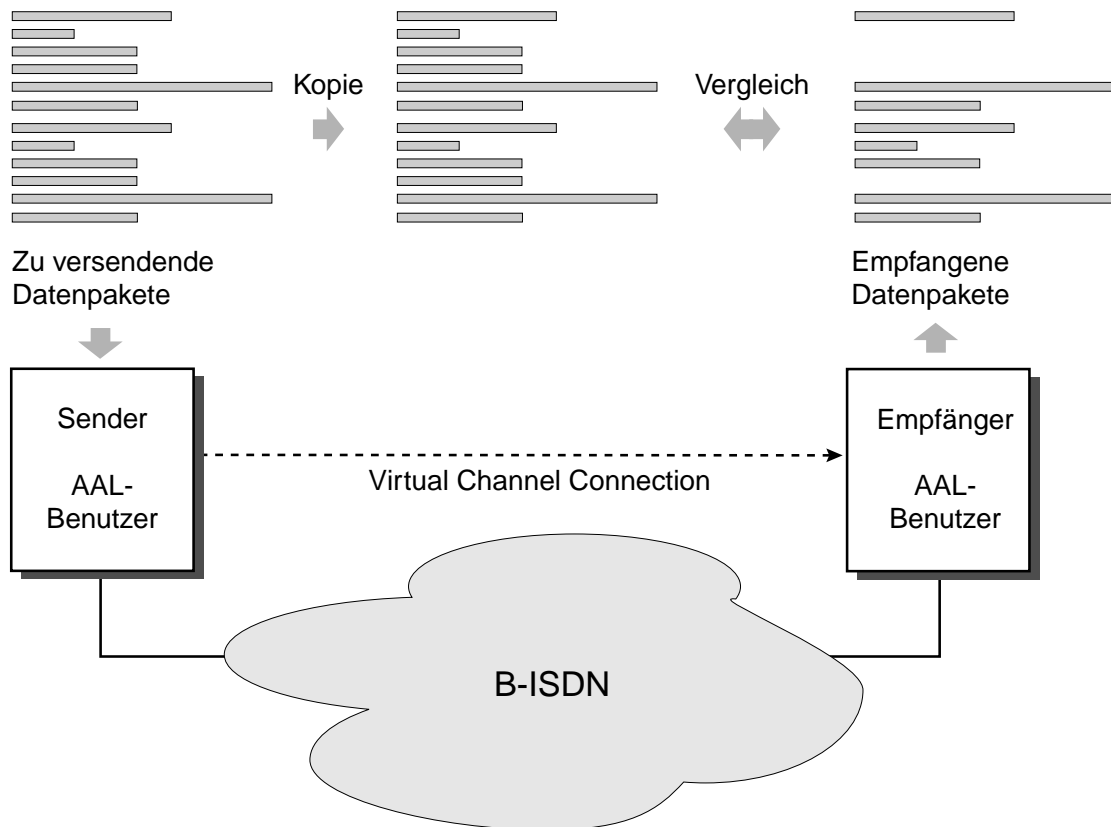


Abbildung 80 : Bitfehler- und Paketverlustanalyse

Mithilfe der so gewonnenen Werte für Bitfehler und Paketverluste und der Momente der Zwischenankunftszeiten wird durch Anwendung des Verfahrens aus Abschnitt 5.3 auf Seite 107 die Qualitätsbeeinträchtigung der Daten dieser Verbindung berechnet.

6.3.2.4 Beeinflussung der CAC

Bei Verwendung des adaptiven CAC-Algorithmus (vgl. Abschnitt 5.4.3 auf Seite 110 ff.) werden, abhängig vom Ergebnis der Qualitätsbeurteilung die Grenzwerte für die zulässigen Qualitätsparameter an den betroffenen Ports geändert.

6.3.2.5 Verbindungsabbau

Beim Abbau der Virtual Channel Connection wird das Ergebnis der Qualitätsanalyse in einer Datei („VCCLog“, vgl. Kapitel 6.6.8 auf Seite 133) festgehalten. Anschließend wird der Zeitpunkt des nächsten Verbindungsaufbaus für diese Quelle zufällig ermittelt.

6.3.3 Analyse des Simulationsergebnisses

Im Anschluß an die Simulationsdurchführung findet die Analyse des Simulationsergebnisses statt. Durch Mittelwertbildung über alle erfolgreich aufgebauten Verbindungen wird der erreichte Wertdurchsatz (vgl. Gl. 31 auf Seite 109) für diesen Lastfall bestimmt .

6.4 Ermittlung der Bediendauern

6.4.1 Schichtenmodell der Simulationssoftware

Zur Berechnung der Bediendauern T_B in den Protokollinstanzen (vgl. Abschnitt 2.3.1 auf Seite 20) enthält die Simulationssoftware ein an das OSI-Modell angelehntes Schichtenmodell aus zwei „Layern“:

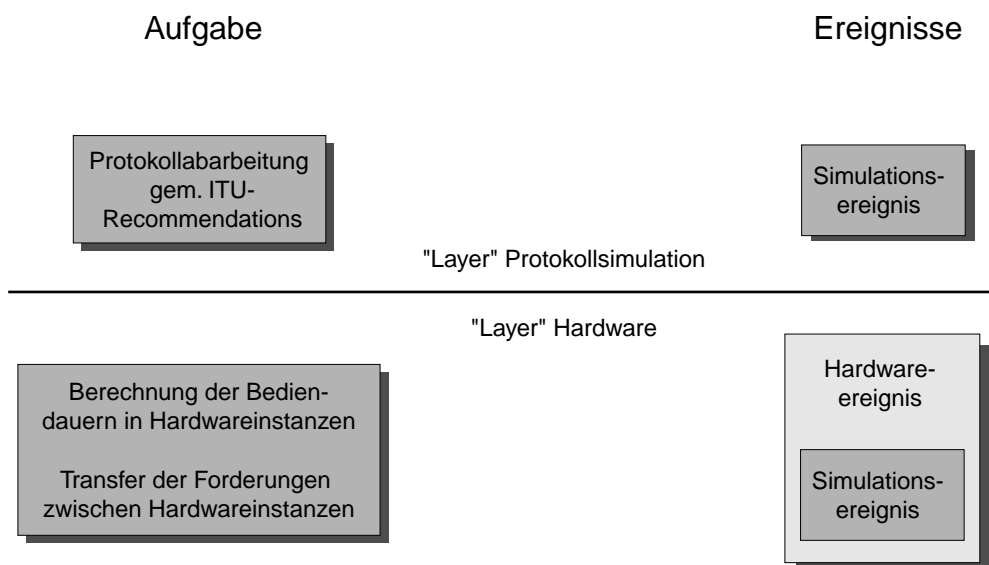


Abbildung 81 : Schichtenmodell der Simulation

Die Protokollsimulation ermittelt abhängig vom Systemzustand die zur Abwicklung der Protokolle entsprechend den Empfehlungen der ITU notwendigen Bearbeitungsschritte. Sie zerlegt diese in ele-

mentare Operationen (z.B. Speicherzugriffe oder arithmetische Verknüpfungen), die von der ihr unterlagerten Hardware ausgeführt werden müssen.

Die erste Aufgabe der Hardwareinstanzen ist die Ermittlung des Zeitbedarfs für die von „ihrer“ Protokollinstanz vorgegebenen Bearbeitungsschritte. Diese Bediendauer ist abhängig von Strukturparametern der Hardware wie Systemaufbau, Busbreite, Taktrate, usw.. Ihre zweite Aufgabe ist die Abarbeitung dieser Forderungen gemäß ihrer Verarbeitungsstrategie (*Scheduling*). Ebenfalls in ihren Aufgabenbereich fällt der Transfer der Service Primitives und Signale zwischen den Bedieneinrichtungen. Zu diesem Zweck kapselt sie die zu transportierenden Simulationsereignisse innerhalb von Hardwareereignissen und transportiert sie transparent zwischen den Puffern der Hardwareinstanzen. Die Transferdauer ermittelt sie anhand der Struktur des Verbindungsnetzwerkes, der Länge der zu transportierenden Daten und den Schreib-/Lesegeschwindigkeiten der beteiligten Speicherbausteine. Dabei überprüft sie ständig die Speicherkapazität der Puffer; bei einer Überschreitung verwirft sie das betroffene Ereignis.

6.4.2 Bedienung von Forderungen in den Hardwareinstanzen

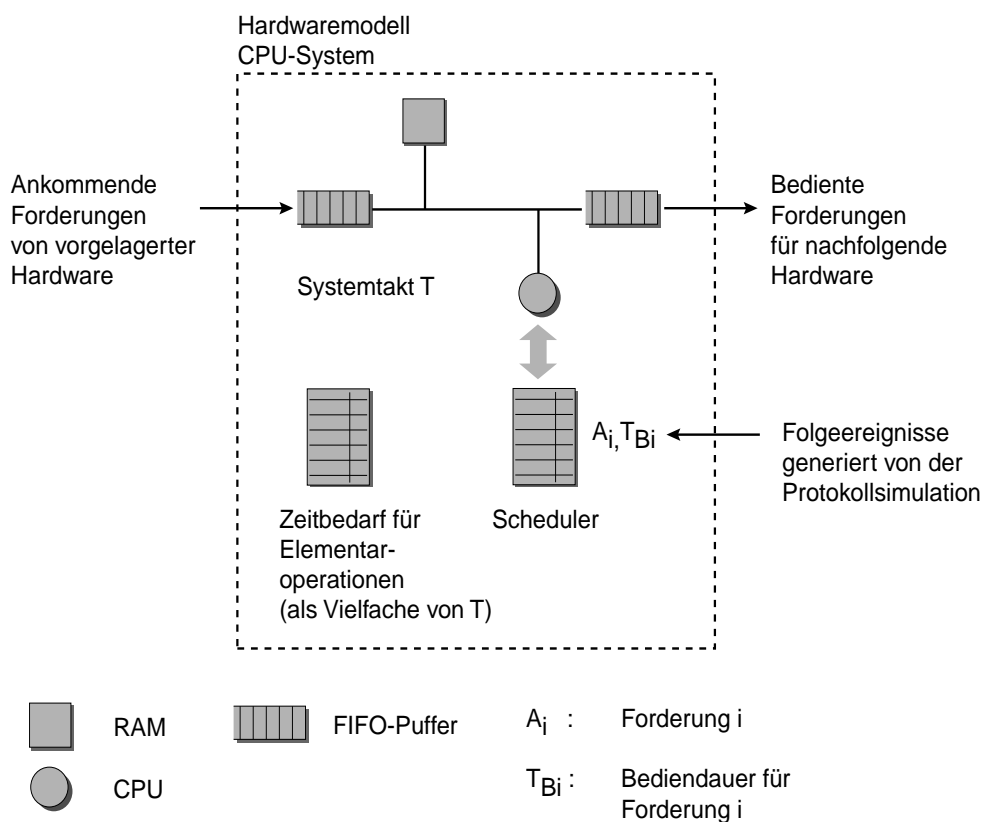


Abbildung 82 : Aufbau einer Hardwareinstanz (Bsp.)

Ein in der Simulation verwendetes Hardwaremodell (siehe *ClassicMicroProzessorSystem* im Anhang C.5.4 auf Seite 230) zeigt Abbildung 82 auf Seite 122. Diese Anordnung umfaßt eine CPU (*Central Processing Unit*) mit einem Scheduler, in dem alle im System befindlichen Forderungen mit der zu ihrer Bedienung notwendigen Zeit abgelegt sind, sowie Ein- und Ausgabepuffer zur Ankopplung weiterer Hardwareinstanzen und einem RAM-Speicher zur Datenbearbeitung. Gemäß der Reihenfolge der Forderungen im Scheduler werden diese von der CPU bedient. Auskunft über die Bediendauern für die von den Protokollabläufen benötigten Operationen erhält die Hardware durch Abgabe der Service Primitives an die ihr zugeordnete Instanz im Layer „Protokollsimulation“. Der Zeitbedarf für den Datentransfer zwischen den Speichern ergibt sich als Teil des Bedienprozesses aus der Taktrate des Systems und den Schreib-/Lesezyklen der beteiligten Speicherbausteine. Nachdem eine Forderung vollständig bedient ist, verläßt sie die Hardwareinstanz über den Ausgangspuffer und wird von der folgenden Verarbeitungsstufe übernommen. Die Interaktionen zwischen den Instanzen der Protokollsimulation und der Hardware zeigt Abbildung 83.

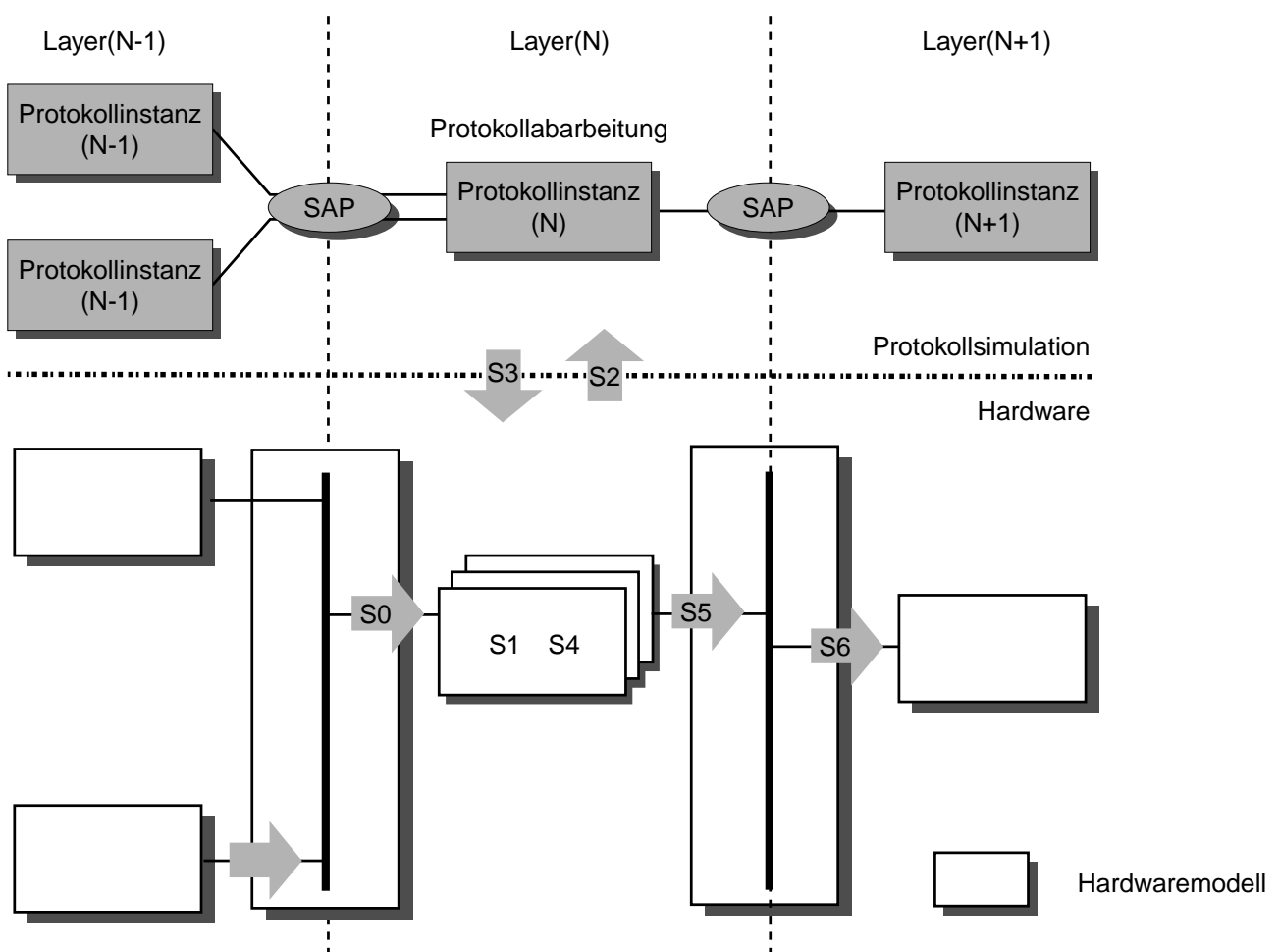


Abbildung 83 : *Bearbeitung und Transfer von Forderungen in der Simulation*

- Schritt S0** Ein Dienstprimitiv einer Protokollinstanz aus Layer (N-1) trifft über den SAP im Eingangspuffer der Hardwareinstanz im Layer (N) ein. Die Hardwareinstanz nimmt den Auftrag zum Transfer dieser Forderung aus dem Eingangspuffer in den Arbeitsspeicher (RAM) in ihren Scheduler auf.
- Schritt S1** Die Forderung wird vom Eingangspuffer zum Arbeitsspeicher der CPU transportiert.
- Schritt S2** Entsprechend der Abarbeitungsstrategie innerhalb der Hardware entnimmt die CPU dem Scheduler eine Forderung. Handelt es sich um ein für die Protokollinstanz bestimmtes Hardwareereignis, übergibt sie das enthaltene Simulationsereignis der Protokollsimulation, um die Bediendauern für die Bearbeitung berechnen zu können.
- Schritt S3** Während der Abarbeitung teilt die Protokollsimulation die durchzuführenden elementaren Bearbeitungsschritte und Folgeereignisse mit. Zu jedem Bearbeitungsschritt berechnet die Hardwareinstanz die Anzahl benötigter Systemtakte. Sie übernimmt die Folgeereignisse mit dem Vermerk der Bearbeitungsdauer in ihren Scheduler.
- Schritt S4** Ist die Bearbeitung einer Forderung beendet, entnimmt die CPU ihrem Scheduler die nächste Forderung und führt die Bedienung für den in der Forderung vermerkten Zeitraum aus. Forderungen für benachbarte Instanzen werden in den Ausgangspuffer geschrieben. Ist die Forderung für die „eigene“ Protokollinstanz bestimmt, wird sie nach der im Scheduler vermerkten Bearbeitungsdauer an diese abgegeben.
- Schritt S5** Die Forderung verläßt den Ausgangspuffer und wird über den SAP (hier eine Busstruktur) zur Hardware der nächsten Protokollinstanz transportiert. Die Transferdauer ist durch die Geschwindigkeit des Busses und die Breite der angeschlossenen Puffer bestimmt.
- Schritt S6** Die Forderung erreicht den Eingangspuffer der Protokollinstanz im Layer (N-1).

Die Trennung der Simulation in die beiden Ebenen ermöglicht eine Simulation der Protokollabläufe unabhängig von der Art der Hardware, die zu ihrer Abwicklung eingesetzt ist. Neben dem in Abbildung 82 auf Seite 122 gezeigten Modell für eine Hardwareinstanz sind weitere Modelle implementiert (vgl. Anhang C.5 auf Seite 227). Spezialisierte Hardware für Instanzen des Physical Layers sowie Variationen des o.g. CPU-Modells mit Dualport-RAMs anstelle der FIFO-Puffer sind verfügbar. Die Zuordnung der Hardwareinstanzen zu den Protokollinstanzen ist frei definierbar, darüberhinaus besteht die Möglichkeit, einer Protokollinstanz mehrere parallele Hardwareinstanzen zuzuordnen.

6.5 Struktur der Software

Den grundlegenden Aufbau der Simulationssoftware zeigt die folgende Abbildung, die Aufgaben der einzelnen Module erläutert Tabelle 50.

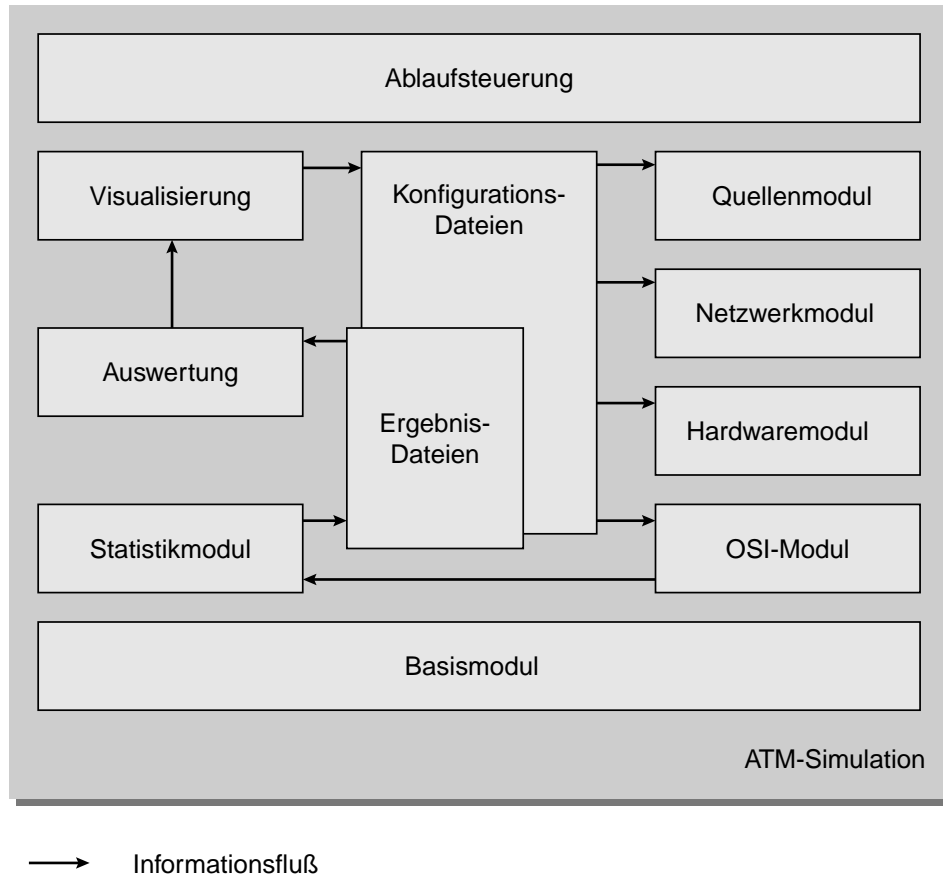


Abbildung 84 : Struktur der Simulationssoftware

Modul	Aufgabe	Beschreibung
Basismodul	Bereitstellung grundlegender, laufzeitoptimierter Datentypen für alle weiteren Module	6.6.2 auf Seite 127
Statistikmodul	Generierung und Analyse von Zufallsvariablen	6.6.3 auf Seite 128
Quellenmodul	Aufbau multimedialer Datenquellen aus Zufallszahlengeneratoren	
Netzwerkmodul	Verwaltung aller Netzwerkknoten und Verbindungen	6.6.4 auf Seite 129
OSI-Modul	Implementierung der Protokolle im B-ISDN gem. Kapitel 3	6.6.5 auf Seite 131

Tabelle 50 : Aufgaben der Module der Simulationssoftware

Modul	Aufgabe	Beschreibung
Hardwaremodul	Nachbildung des Verhaltens der Hardware in den Netzknoten	6.6.6 auf Seite 132
Konfigurationsdateien	Bereitstellung aller Daten, die zur Initialisierung aller Objekte der Simulation notwendig sind	6.6.7 auf Seite 133
Ergebnisdateien	Aufnahme der vorausgewerteten Teilergebnisse in einem Simulationslauf	6.6.8 auf Seite 133
Visualisierung	Visualisierung von Vorgängen und Ergebnissen sowie graphische Eingabe der Netzwerkkonfiguration	6.6.9 auf Seite 133
Auswertung	Auswertung der Simulationsergebnisse	6.6.10 auf Seite 134

Tabelle 50 : Aufgaben der Module der Simulationssoftware

6.6 Kurzbeschreibung der Module

Die folgenden Abschnitte beschreiben kurz den internen Aufbau der Simulationssoftware, insbesondere die Zusammenhänge zwischen den implementierten Klassen. Eine detailliertere Auflistung aller Klassen mit ihren Aufgaben und Besonderheiten ist jeweils im entsprechenden Abschnitt im Anhang C auf Seite 195 ff. zu finden. Auf die Beschreibung der Implementierung wird an dieser Stelle aus Übersichtlichkeitsgründen verzichtet, eine kommentierte Zusammenstellung aller Klassenmethoden ist in [Doc] zu finden.

6.6.1 Notation

Die in den folgenden Diagrammen häufig verwendeten Symbole und ihre Bedeutung zeigt Abbildung 85.

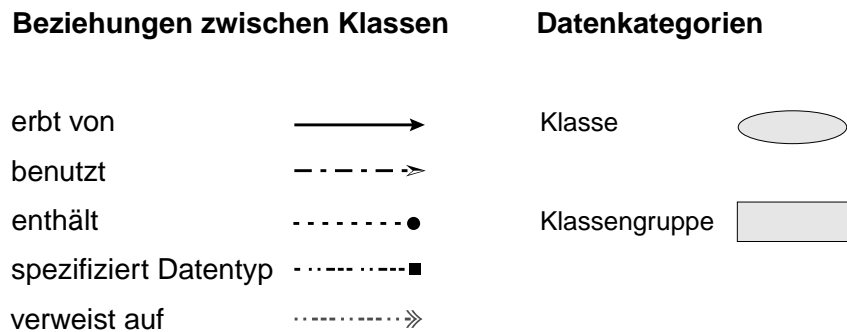


Abbildung 85 : Häufig verwendete Symbole

6.6.2 Basismodul

6.6.2.1 Aufbau

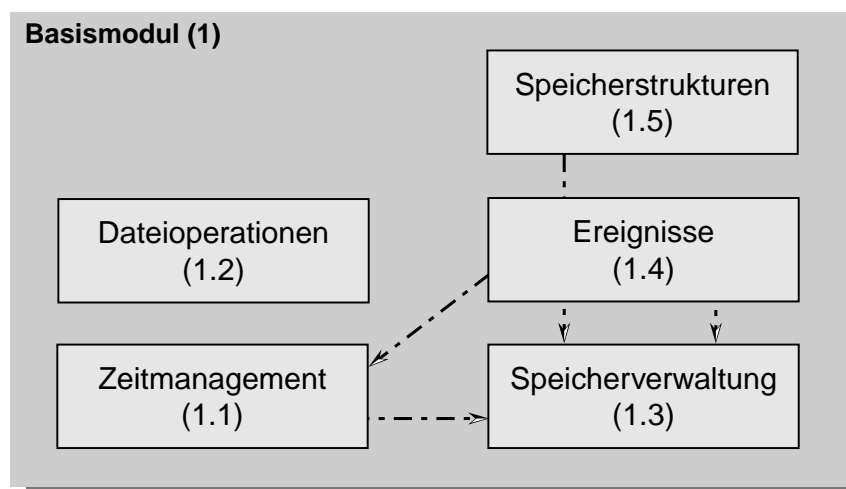


Abbildung 86 : Funktionsgruppen des Basismoduls

- Zeitmanagement**
(1.1) Diese Klassengruppe wickelt alle Operationen ab, die sich auf die Simulationszeit beziehen. Zeitaufwendige Operationen auf den Basistypen *double* bzw. *long double*, sowie deren Ungenauigkeiten bei arithmetischen Operationen bei stark unterschiedlichen Operanden werden durch Einführung einer eigenen Festkommaklasse vermieden. Zweite Aufgabe ist die Bereitstellung einer laufzeitoptimierten Realisierung der Ereignisverwaltung aus Abbildung 77 auf Seite 114.
- Dateioperationen**
(1.2) Sichere Operationen für das Lesen und Schreiben von Textdateien. Vor jedem Zugriff wird der Dateizustand geprüft. Im Falle einer fehlerhaften Datei wird der Zugriff zurückgewiesen, ohne daß ein Programmabbruch erfolgt.
- Speicher-
verwaltung**
(1.3) Aufgabe dieser Gruppe von Klassen ist die Minimierung der zur Verwaltung von dynamisch allokiertem Speicherplatz benötigten Rechenzeit. Da die von den Standardoperatoren *new* und *delete* genutzten Betriebssystemroutinen sehr zeitintensiv sind, wird eine eigene Verwaltung der Speicherblöcke implementiert.
- Ereignisse**
(1.4) Die Basisklassen aller in der Simulation verwendeten unterschiedlichen Ereignisse.
- Speicher-
strukturen**
(1.5) Eine Sammlung von Klassen zur Speicherung temporärer Daten in der Simulation: Vektoren, Matrizen, Listen mit unterschiedlichen Ein- und Ausgabestrategien und verschiedene Typen binärer Bäume. Alle Klassen sind als Containerklassen implementiert und können beliebige Datentypen aufnehmen.

6.6.3 Statistik- und Quellenmodul

Zur Nachbildung multimedialer Datenquellen in der Simulation wird der Zeitpunkt des Sendens eines Datenpaketes durch Zufallszahlengeneratoren erzeugt, welche dieselben statistischen Eigenschaften wie die von ihnen repräsentierten Datenquellen aufweisen. Die Übereinstimmung der Verteilungsfunktion der so erzeugten Zufallsvariablen mit dem theoretisch erwarteten Verlauf (siehe Tabelle 9 auf Seite 17) wurde mithilfe von χ^2 -Tests zum Signifikanzniveau 0,995 (0,5% Irrtumswahrscheinlichkeit) mit den Methoden der Klassen aus dem Modul 2.2 nachgewiesen. Simulationen von Bediensystemen M/G/1 und

Vergleich der Ergebnisse mit den für diese Systeme bekannten analytischen Lösungen aus Abschnitt 2.3 auf Seite 20 bestätigten die korrekte Funktionsweise der Generatoren- und Analyseklassen.

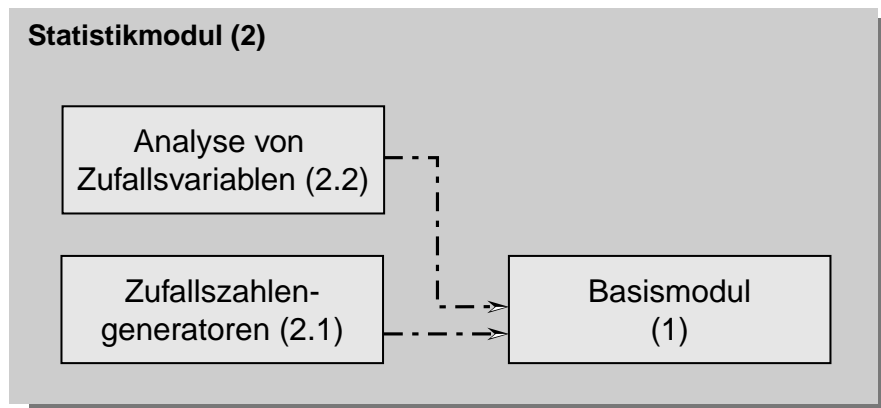


Abbildung 87 : Funktionsgruppen des Statistikmoduls

6.6.4 Netzwerkmodul

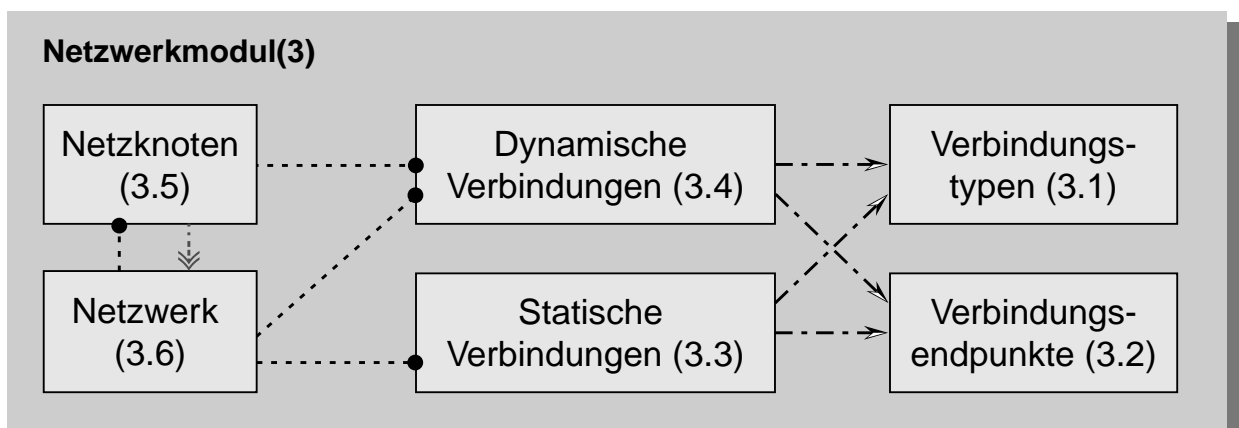


Abbildung 88 : Funktionsgruppen des Netzwerkmoduls

Aufgabe des Netzwerkmoduls ist die Abbildung des zu simulierenden B-ISDN-Netzwerks mit allen Verbindungsfunktionen, wie sie in Abschnitt 3.3.1 auf Seite 39 geschildert sind, auf eine Klassenstruktur. Die Netz-knoten beinhalten sowohl die Protokollinstanzen in den verschiedenen Schichten des B-ISDN-Referenzmodells gem. Abschnitt 3.5.3 auf Seite 46, als auch Datenstrukturen zur Durchführung der Vermittlung auf Pfad- und Kanalebene. Die Adressierung der Stationen innerhalb des

Netzwerks erfolgt gemäß den Vorgaben der ITU durch Adressen im Format E.164 [E164] oder ISO NSAP gem. X.213 [X213].

Verbindungstypen Die Klassen dieses Moduls repräsentieren alle Verbindungstypen auf der
(3.1) physikalischen Ebene, sowie auf Pfad- bzw. Kanalebene (vgl. Abschnitt 3.3.1 auf Seite 39).

Verbindungs- Die Verbindungsendpunkte auf den unterschiedlichen Verbindungsebenen
endpunkte werden durch eine Datenstruktur repräsentiert, in der sich ihr hierarchischer
(3.2) Zusammenhang widerspiegelt. Jeder Endpunkt verwaltet in einer Baumstruktur alle Verbindungen der ihm untergeordneten Ebene. Auf diese Art können beim Abbau einer Verbindung auf einem höheren Niveau, z.B. einer *Virtual Path Connection*, alle von diesem Verbindungsabbau betroffenen unterlagerten *Virtual Channel Connections* ebenfalls abgebaut werden. Innerhalb einer Vermittlungsstelle auf VC- bzw. VP-Level verfügt jeder Eingangs-Endpunkt über eine Liste der mit ihm verbundenen Verbindungsendpunkte auf der Ausgangsseite. So wird das Einrichten sowohl von Punkt-zu-Punkt als auch von Punkt-zu-Mehrpunkt-Verbindungen unterstützt. Jedem Endpunkt ist eine Klasse zugeordnet, deren Aufgabe die Verkehrsanalyse auf dieser Ebene ist (siehe Klassen 3.2.6 - 3.2.8 in Kapitel C.3 auf Seite 205 ff.).

Statische Diese Gruppe von Klassen verwaltet alle im Netzwerk (3.6) vorhandenen
Verbindungen nicht veränderlichen Verbindungen auf der physikalischen Ebene und der
(3.3) Ebene virtueller Pfade. Die Informationen über die bestehenden Verbindungen erhält sie aus den Konfigurationsdateien mithilfe der Klasse *ConfigDatei* (siehe Klassengruppe 1.2.3). Ihre Aufgabe ist die Suche eines optimalen Wegs im Netzwerk beim Aufbau einer neuen Nutzdatenverbindung unter Verwendung der im Netzwerk vorhanden Pfadverbindungen.

Dynamische Im Gegensatz zu den statischen physikalischen Verbindungen und VPCs
Verbindungen werden die Virtual Channel Connections erst auf Anforderung des Benutzers auf- und abgebaut (vgl. Signalisierung in Abschnitt 3.9 auf Seite 90).
(3.4)

Netzknoten Die Klassenobjekte dieses Moduls repräsentieren die Stationen in einem
(3.5) ATM-Netzwerk. Der geographische Ort der Station, ihre Adresse(n) sowie alle in ihr vorhandenen Protokollinstanzen sind an dieser Stelle zusammengefaßt.

Netzwerk Die Netzwerk-Klasse stellt das zu simulierende ATM-Netz dar; sie verwaltet
(3.6) alle Stationen sowohl anhand ihrer Adressen (3.5.2) als auch unter Verwendung interner Nummern zur Minimierung der Suchdauern.

6.6.5 OSI-Modul

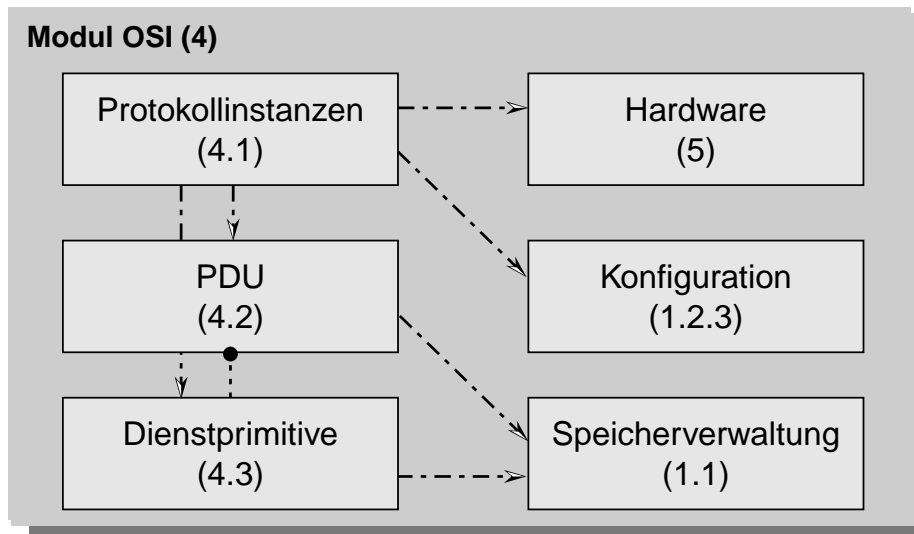


Abbildung 89 : Klassengruppen des OSI-Moduls

Die Elemente des OSI-7-Schichtenmodells (vgl. Abschnitt 3.5.1 auf Seite 41 ff) für Nachrichtennetze werden von den Klassen innerhalb des OSI-Moduls repräsentiert.

Protokollinstanzen (4.1)

Die Abarbeitung der Protokolle in den Schichten des B-ISDN-Referenzmodells wird von den Klassen der Gruppe (4.1) übernommen. Den implementierten Abläufen liegen direkt die Recommendations der ITU zugrunde. Flußdiagramme zum Ablauf der Protokolle innerhalb der Instanzen sowie die Details zur Implementierung sind jeweils in den angegebenen Studien- und Diplomarbeiten ([Har98], [Her96], [Hom97], [Ngu97], [Rau95], [Wur98] und [Wee97]) zu finden. Über die Dauer der einzelnen Arbeitsschritte entscheiden die unterlagerten Hardwareklassen des Moduls (5). Alle zur Konfiguration der Protokollinstanzen notwendigen Informationen werden im Rahmen der Stationsinitialisierung bei Programmstart aus Konfigurationsdateien (Klassengruppe 1.2.3) entnommen.

PDU (4.2)

Die Klassen der Gruppe PDU implementieren die zu transportierenden *Protocol Data Units* aller Instanzen.

Dienstprimitive (4.3)

Die Klassen dieser Gruppe umfassen alle zwischen den Instanzen ausgetauschten Dienstprimitive.

6.6.6 Hardwaremodul

Alle Arbeitsschritte, die im Verlauf der Abwicklung des Protokolls in einer Instanz durchgeführt werden müssen, benötigen einen von der Leistungsfähigkeit und dem internen Aufbau der Hardware abhängigen Zeitraum. Die Klassen innerhalb dieses Moduls stellen unterschiedliche Hardwareanordnungen für Protokollinstanzen und die Verbindungsnetzwerke zwischen diesen zur Verfügung. Alle Klassen verfügen über detaillierte Angaben über ihren internen Aufbau und die Geschwindigkeit, mit der sie elementare Rechen- und Speicheroperationen abwickeln können. Die implementierten Klassen sind frei konfigurierbar und an beliebige Hardwarekonfigurationen anpassbar; in einem praxisnahen Beispiel orientieren sich die in der Simulation eingesetzten Hardwaremodelle am Aufbau der ATM-NIC¹ gem. [Bro94], [Bro95], [Nec95_2] und [Nec95_3].

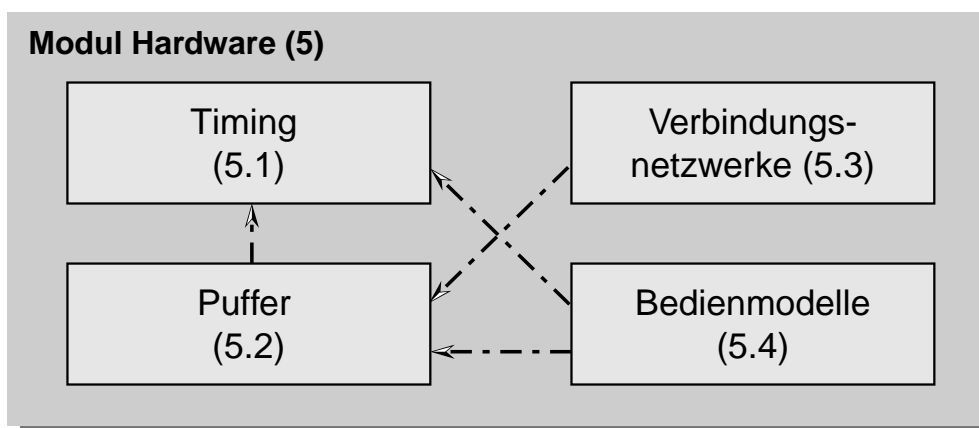


Abbildung 90 : Funktionsgruppen des Hardwaremoduls

Timing (5.1)	Klassen zur Berechnung des Zeitbedarfs zur Abwicklung aller Operationen in Hardwareanordnungen.
Puffer (5.2)	Diese Klassen dienen der Modellierung unterschiedlicher Speicherbausteine. Mithilfe der Angaben über die Busbreite, Taktrate und die Taktzyklen beim Ein- und Auslesevorgang des Speicherbausteins wird das reale Verhalten der Puffer nachgebildet. Bei Überschreitung des Speichervermögens wird das verursachende Datenpaket gelöscht. Die Berechnung des benötigten Pufferspeichers berücksichtigt gleichzeitige Zugriffe auf den Pufferspeicher, falls die Art des Speichers solche Zugriffe erlaubt (z.B. ein FIFO-Puffer).

1. ATM Network Interface Card

Verbindungs- netzwerke (5.3)	Simulationsmodelle für die Verbindungsnetzwerke zwischen den Hardwarekomponenten. Die Verknüpfung kann sowohl über Busstrukturen als auch über Switches (Crossconnects) erfolgen.
Bedienmodelle (5.4)	Die Klassen innerhalb dieser Gruppe stellen die Simulationsmodelle für unterschiedliche Hardwareanordnungen zur Abwicklung der Protokolle zur Verfügung. Implementiert sind sowohl Modelle für auf spezielle Anforderungen hin optimierte Anordnungen als auch „klassische“ Anordnungen aus Prozessoren, Programmspeicher und Ein-/Ausgabeeinheiten. Jeder Protokollinstanz können beliebig viele parallele Anordnungen zugeordnet werden. Die Vorgabe der Eigenschaften der Bedieneinrichtungen erfolgt über die Konfigurationsdateien (Klassengruppe 1.2.3) beim Simulationsstart.

6.6.7 Konfigurationsdateien

Diese Dateien stellen den Konstruktoren aller Netzknoten bei Simulationsstart die Konfigurationsinformationen zur Verfügung. Analog zum hierarchischen Aufbau des zu simulierenden Netzwerks sind die Konfigurationsdateien unterhalb des Wurzelverzeichnis des Simulationslaufes angeordnet.

6.6.8 Ergebnisdateien

Zu Beginn eines Simulationslaufes werden mehrere Dateien erzeugt, worin die Ergebnisse sowie diverse Fehler- und Protokollausgaben abgelegt werden.

6.6.9 Visualisierung

Die Komplexität der Textdateien zur Konfigurierung des zu simulierenden Netzwerks verlangt nach einer graphisch unterstützten Eingabehilfe, um einen vollständigen und fehlerfreien Datensatz (vgl. Kapitel C.6 auf Seite 231 ff.) erstellen und abspeichern zu können. Diese Aufgabe übernimmt das unter X11¹ mit dem Widget²-Set von OSF/Motif erstellte Visualisierungstool, das als spezialisiertes CAD³-Programm implementiert ist.

1. Graphische Benutzeroberfläche auf Unix-Rechnern
2. Graphische Elemente, aus denen sich die X-Oberfläche zusammensetzt
3. Computer aided design

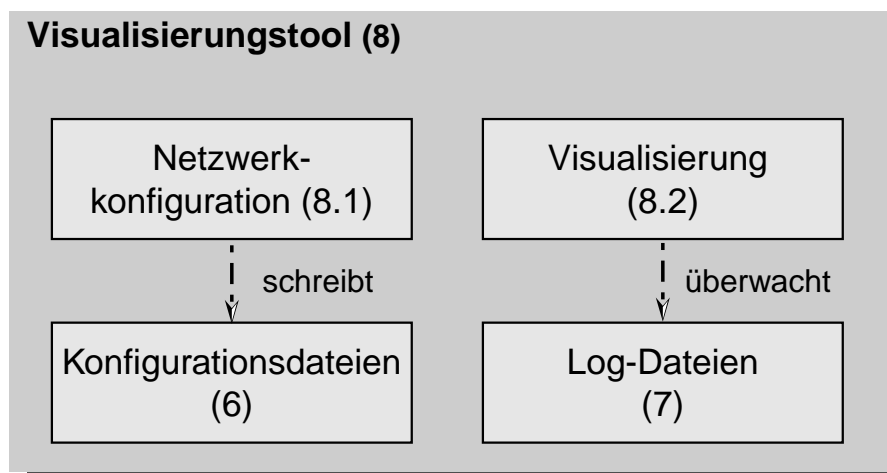


Abbildung 91 : Visualisierungstool

**Netzwerk
konfiguration
(8.1)**

Klassen, deren Funktionen es erlauben, das gesamte Netzwerk mit allen Stationen und Verbindungen zeichnerisch zu definieren. Jedem Netzknoten sind hierarchisch angeordnete Eingabemasken zugeordnet, deren Gestaltung den Benutzer zur vollständigen Angabe aller Konfigurationsparameter zwingt. Dies erleichtert den Aufbau eines Netzwerkes und die gezielte Veränderung einzelner Parameter innerhalb eines Netzbetriebsmittels bis hinab zur untersten Ebene der Datenstrukturen (vgl. [Bo197]). Die Gestaltung der Eingabemasken aller Ebenen wird durch externe Textdateien vorgenommen, so daß Ergänzungen innerhalb der bestehenden Masken jederzeit ohne Eingriff in den Quelltext möglich sind.

**Visualisierung
(8.2)**

Graphische Aufbereitung der Vorgänge während der Simulation zu Demonstrationszwecken oder zur Fehlersuche (Debugging).

6.6.10 Ablaufsteuerung

Dieses Modul übernimmt die gesamte Steuerung eines Simulationslaufs. Bei Simulationsstart wird die Konfigurationsinformation (vgl. Abschnitt 6.6.7 auf Seite 133) übernommen und das zu simulierende Netzwerk aufgebaut. Anschließend werden die Datenquellen entsprechend den Informationen in der Datei *StartUp* konfiguriert. Zufallsgeneratoren übernehmen den Auf- und Abbau von *Virtual Channel Connections* abhängig von einem bei Simulationsstart anzugebenden Lastfaktor. Nachdem alle Ergebnisdateien erzeugt sind, wird die Simulation durch die ersten Verbindungsaufbauten gestartet. Während

des Programmlaufs übernehmen die Instanzen diese Moduls die zeitrichtige Verteilung aller Ereignisse auf die Protokollinstanzen (vgl. Abschnitt 6.2 auf Seite 114) sowie die Ausgabe aller Fehler- und Protokollmeldungen in die dafür vorgesehenen Dateien. Nach dem Abbau einer VCC wird die Auswertung der statistischen Parameter des Datenstroms beim Empfänger angestoßen und das Ergebnis zur späteren Analyse der Datei *VCCLog* (vgl. Abschnitt C.7 auf Seite 233) hinzugefügt.

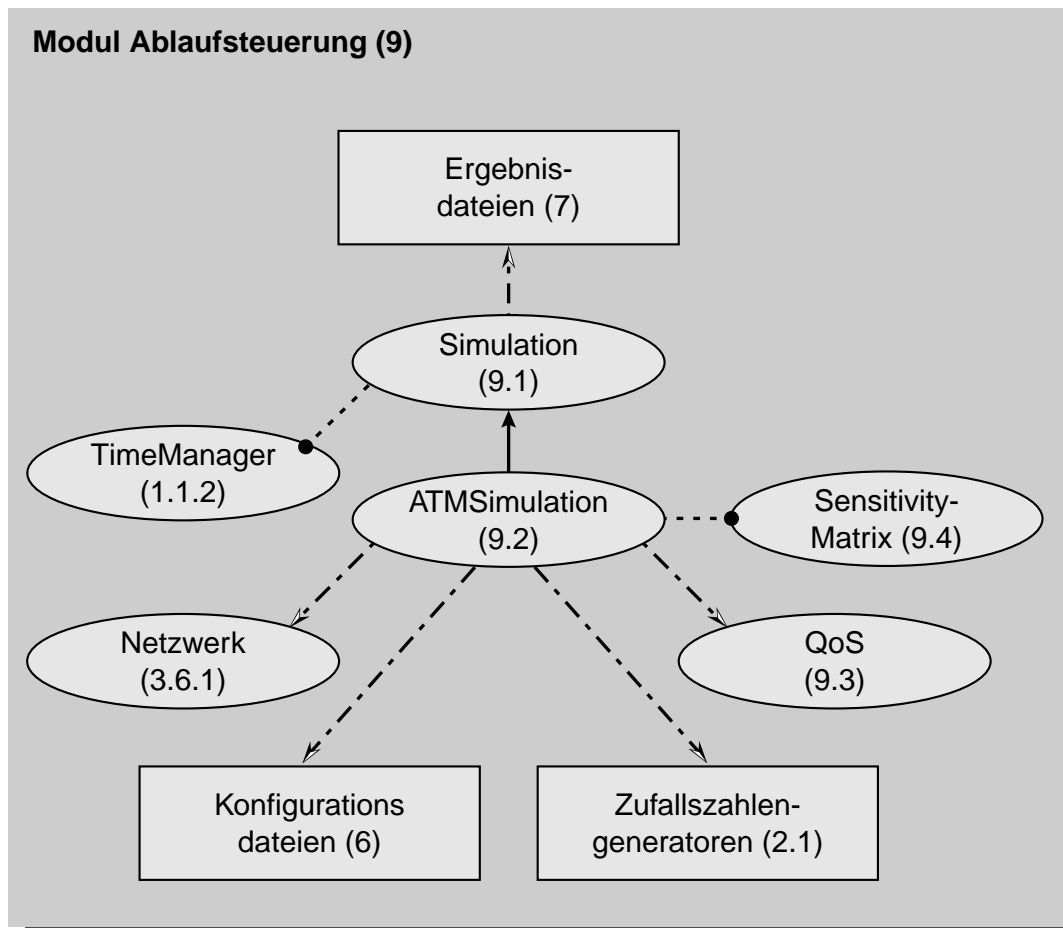


Abbildung 92 : Ablaufsteuerung

**Simulation
(9.1)**

Die Basisklasse übernimmt die Steuerung aller Zugriffe auf die *Ergebnisdateien (7)* sowie die Verwaltung aller Ereignisse innerhalb der Simulation im *TimeManager (1.1.2)*.

**ATMSimulation
(9.2)**

Die Instanz dieser Klasse übernimmt die ATM-spezifischen Aufgaben während der Simulation. Sie steuert den Auf- und Abbau von VCCs, die Berechnung der erreichten Übertragungsqualität und die Modifikationen am *CAC-Algorithmus*. (vgl. Abschnitt 5.4.3 auf Seite 110).

QoS (9.3)	Die QoS-Klasse beinhaltet die statistischen Kenngrößen eines Datenstroms, die zur Qualitätsbeurteilung (vgl. Abschnitt 5.4.3 auf Seite 110 ff.) herangezogen werden.
SensitivityMatrix (9.4)	In der Matrix sind die Empfindlichkeiten der unterschiedlichen Datenströme gegenüber Beeinträchtigungen aller Komponenten der QoS-Klasse abgespeichert.

6.7 Bedienung und Umfang

Die im Rahmen der vorliegenden Arbeit entstandene Software hat einen Umfang von etwa 100.000 Zeilen und ist in die im vorherigen Abschnitt geschilderten Module unterteilt. Die Erstellung des ablauffähigen Programms erfolgt, wie unter UNIX üblich, durch das Dienstprogramm *make*. Der Ablauf wird durch mehrere hierarchisch angeordnete *makefiles* gesteuert, die jeweils für die Erstellung einer Modulbibliothek verantwortlich sind. Das in der Hierarchie am höchsten stehende *makefile* sorgt für das abschließende Linken der Bibliotheken und des dynamisch generierten Codes der Schablonenklassen. Einige der kommentierten *makefiles* sind Anhang E auf Seite 260 zu finden.

6.8 Zusammenfassung

Im Rahmen der vorliegenden Arbeit wurde unter Verwendung der Programmiersprache C++ eine strukturierte und laufzeitoptimierte Simulationssoftware für ATM-Netzwerke entwickelt. Sie ist in der Lage, die Übertragung von Daten multimedialer Quellen mithilfe der von der ITU standardisierten Protokolle nachzubilden.

Die Software trennt die zu simulierenden Protokollabläufe von der Hardware in den Netzknoten, so daß eine Zuordnung unterschiedlicher Hardwarestrukturen zu den Protokollinstanzen möglich ist und deren Einfluß auf den Gesamtübertragungsvorgang untersucht werden kann. Die Schnittstelle zwischen Hardware und Protokollsimulation bietet die Möglichkeit, die Gesamtbediendauer für die einzelnen Protokollschritte aus elementaren Teiloperationen zu berechnen. Über die Abwicklung der Schritte entscheidet der interne Aufbau und die Abarbeitungsstrategie der Hardware. Dieser Ansatz macht das Hardwaremodell flexibel und an beliebig gestaltete Protokollinstanzen in den Netzknoten anpaßbar.

Eingesetzt wird die Simulationssoftware zur Untersuchung und Optimierung von Algorithmen zur Connection Admission Control im B-ISDN, wie im folgenden Kapitel anhand ausgewählter Szenarien beschrieben wird.

Zum Nachweis der besonderen Eignung des *Adaptiven CAC-Algorithmus* zur Verkehrslenkung in ATM-Netzwerken wird nun die im vorherigen Kapitel beschriebene Simulationssoftware eingesetzt. Zu diesem Zweck werden ausgewählte Netzwerkstrukturen unter jeweils gleichen Lastbedingungen, aber unterschiedlichen Verbindungsannahmestrategien simuliert. Die dadurch gewonnenen statistischen Kenngrößen werden mithilfe der in Abschnitt 5.4.2 auf Seite 108 ff. definierten Gleichungen zum Vergleich des erreichten mittleren Wertdurchsatz herangezogen. Als Vergleichsverfahren werden zwei ausschließlich auf der Betrachtung der Zellraten (vgl. CAC gem. ITU Rec. I.371 , Abschnitt 4.2 auf Seite 96 ff.) basierende Verfahren herangezogen. Das erste Vergleichsverfahren, im folgenden „*PeakAllocation*“ genannt, legte die Spitzenbitrate aller bereits vorhandenen Verbindungen bei der Durchführung der CAC zugrunde, das Verfahren „*MeanAllocation*“ deren mittlere Bitrate. Die beiden Verfahren stellen Extremfälle bei der Definition einer *Äquivalenten Quelle* (vgl. Abbildung 68 auf Seite 100) für die bereits bestehenden Verbindungen dar. Während das auf der Spitzenrate beruhende Verfahren eine pessimistische Abschätzung liefert, tendiert das zweite Verfahren zu einer Überschätzung der Leistungsfähigkeit des Netzes (siehe Abschnitt 4.2.2.2 auf Seite 99). Die Ergebnisse zeigen, daß das als Kompromiß neu entwickelte adaptive Verfahren die Leistungsfähigkeit des Übertragungsnetzwerks besser einschätzt. Dadurch ist es möglich, eine größeren Menge an Daten bei akzeptablen Qualitätseinbußen zu transportieren.

7.1 Ablaufdiagramme

Der prinzipielle Ablauf der Simulation wurde bereits in Abschnitt 6.3.2 auf Seite 119 ff. beschrieben. An dieser Stelle soll nun näher auf die am Ablauf beteiligten Programmodule, die Ausführung und die wichtigsten Ablaufparameter eingegangen werden.

7.1.1 Netzwerkinitialisierung

- Mit Unterstützung des Netzwerkeditors (vgl. Abschnitt 6.6.9 auf Seite 133) wird das zu simulierende Netzwerk graphisch definiert. Diese Konfiguration des Netzwerkes und aller Stationen wird in der vorgegebenen Dateistruktur (vgl. Anhang C.6 auf Seite 231) abgelegt. Die Konfigurationsdatei „*StartUp*“ (vgl. Anhang D.1.1 auf Seite 237 ff.) legt den Verkehrsmix, eine Grundwiederholrate $\lambda_{\text{WDH}} = 1/T_{\text{WDH}}$ für die Verbindungsaufbauten und die Art des zu verwendenden CAC-Algorithmus fest.
- Bei Simulationsstart liest die Software die Konfigurationsdateien aus und erzeugt daraus die Netzwerkstruktur (Klassengruppe (3)) sowie alle Datenquellen. Die Zufallszahlengeneratoren (Klassengruppe (2.1)) werden als „Kopiervorlage“ in dem Quellenobjekt abgelegt. Eine spezielle Station im Netzwerk, das *Network Management Center NMC*, übernimmt wie im realen B-ISDN die Informationen über die physikalischen Verbindungen und statischen Virtuellen Pfade in ihre Netzwerkverbindungsdatenbank (Klassengruppe (3.3))
- Abhängig von dem beim Programmstart angegebenen Lastfaktor α_{rel} wird ein Zufallsgenerator initialisiert, der für die Generierung aller Verbindungsaufbauwünsche im Netzwerk verantwortlich ist. Der Generator genügt einer negativ-exponentiellen Verteilungsfunktion (Klasse (2.1.4)) mit dem Verteilungsparameter $\lambda_{\text{WDH}}' = \lambda_{\text{WDH}} \cdot \alpha_{\text{rel}}$.
- Der Zeitpunkt $t_{\text{Setup},1,i}$ des ersten Verbindungsaufbaus für jede Quelle i wird ausgewürfelt und der Zeitverwaltung (Klasse (1.1.2)) zugeführt.

7.1.2 Verbindungsaufbau

Zum Zeitpunkt $t_{\text{Setup},n,i}$ erfolgt der n -te Verbindungsaufbauversuch der Quelle i . Mithilfe der in der Netzwerkverbindungsdatenbank im *Network Management Center* abgelegten Informationen wird für die gewünschte Virtual Channel Connection (Klassengruppe (3.4)) der kürzeste Weg durch das Netz-

werk (Klassengruppe (3.6)) gesucht. Ist die Wegsuche erfolgreich verlaufen und die CAC stimmt dem Verbindungsaufbau zu, wird eine Kopie des Zufallszahlengenerators der Quelle für die Generierung von Datenpaketen initialisiert und der Datentransfer beginnt. Einen Überblick über den gesamten Ablauf beim Aufbau einer Virtual Channel Connection zeigt Abbildung 93 auf Seite 141.

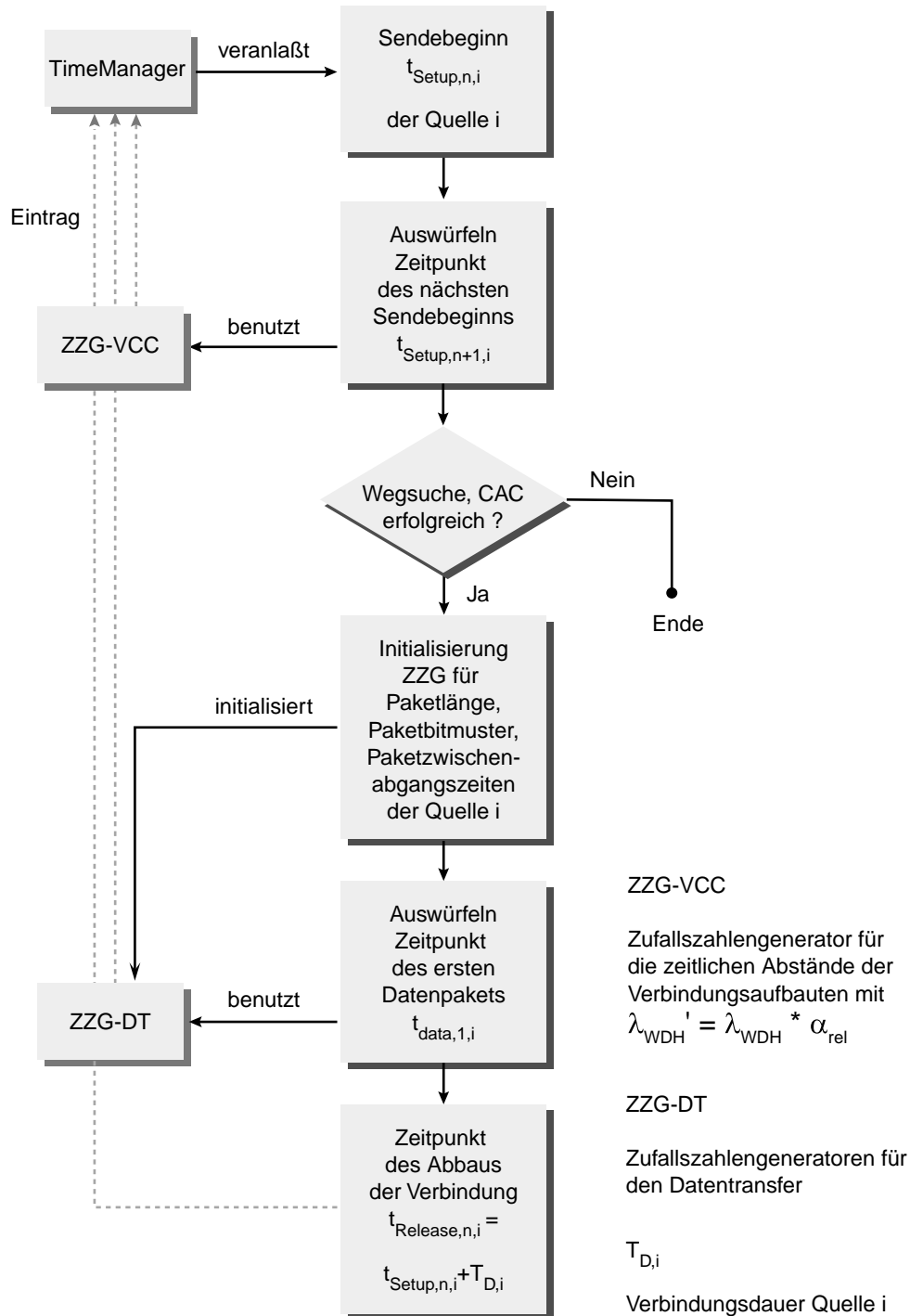


Abbildung 93 : Ablaufdiagramm Verbindungsaufbau

7.1.3 Datentransfer

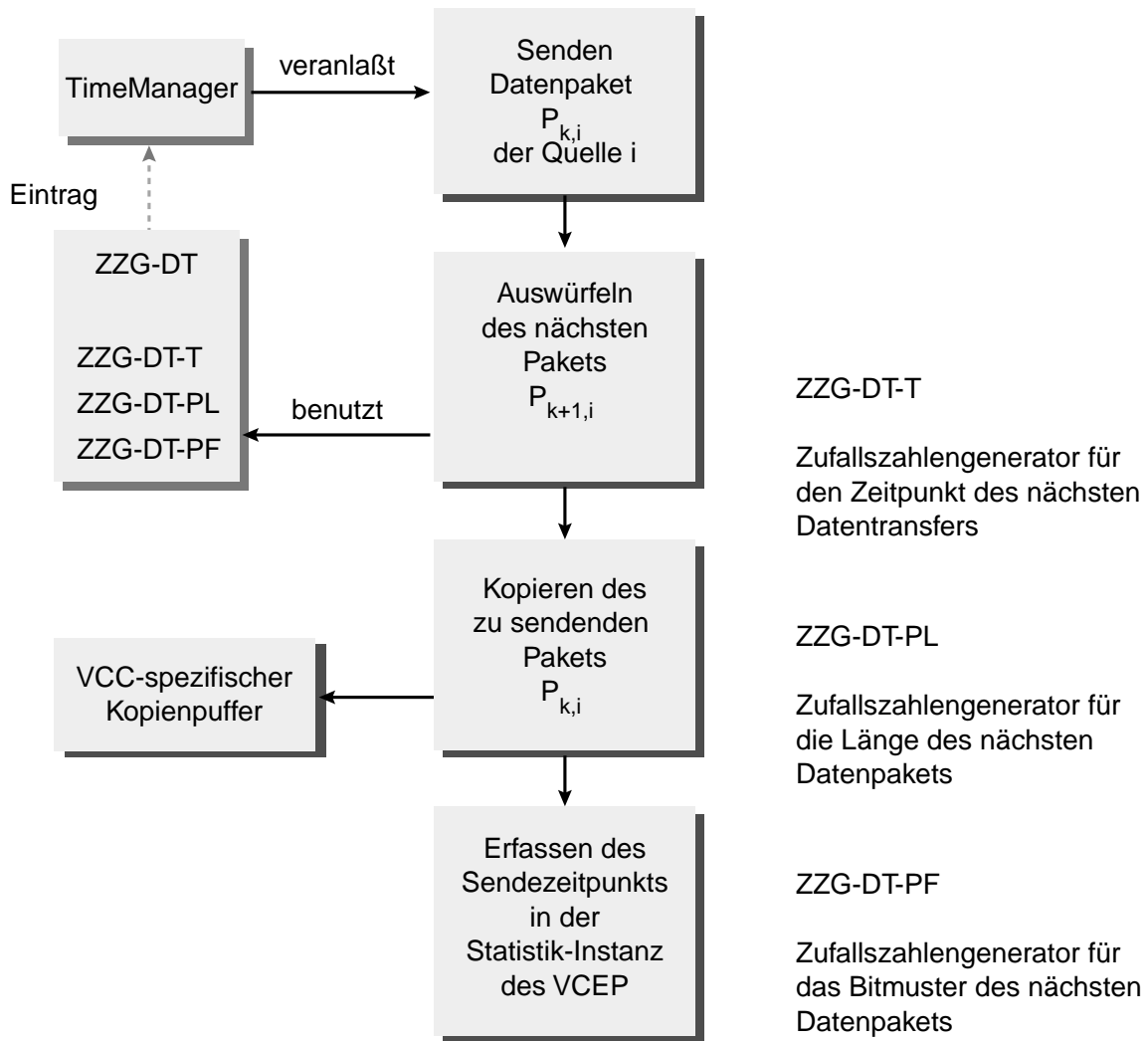


Abbildung 94 : Sendevorgang

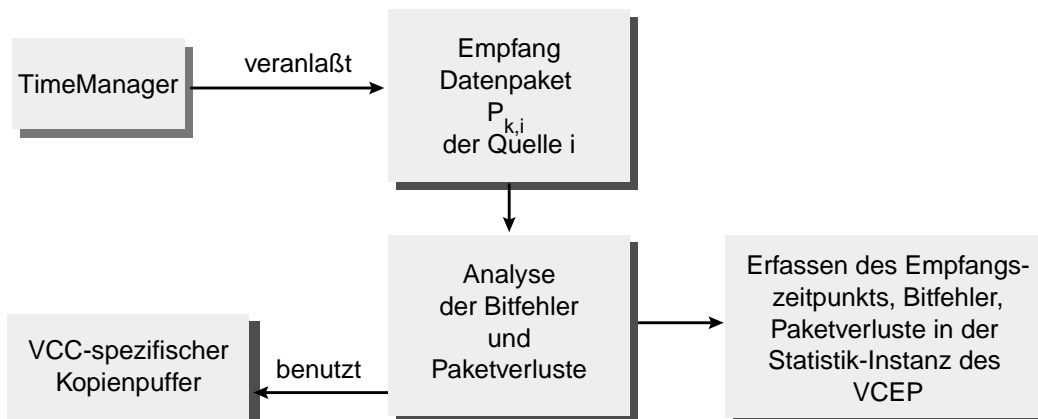


Abbildung 95 : Empfang eines Paketes

Während dieser Phase erzeugen die Zufallsgeneratoren innerhalb der sendenden Station Datenpakete mit zufälliger Länge und Füllmustern. Die Zwischenabgangszeiten ergeben sich aus der in der Quel-
 lendefinition abgelegten Verteilungsfunktion. Vor der Übertragung (Klassengruppen (3),(4) und (5))
 wird eine Kopie jedes Paketes angefertigt, um im Empfänger (Klasse (3.5.1)) einen Vergleich des
 empfangenen mit dem ursprünglich gesendeten Paket durchführen zu können. Solange die Verbin-
 dung besteht, werden im Empfänger alle Bitfehler und Paketverluste aufgenommen. Aus den Emp-
 fangszeitpunkten wird die Dichtefunktion der Zwischenankunftszeiten ermittelt (Klassengruppe
 (2.2)) (siehe Abbildung 95 auf Seite 142).

7.1.4 Verbindungsabbau

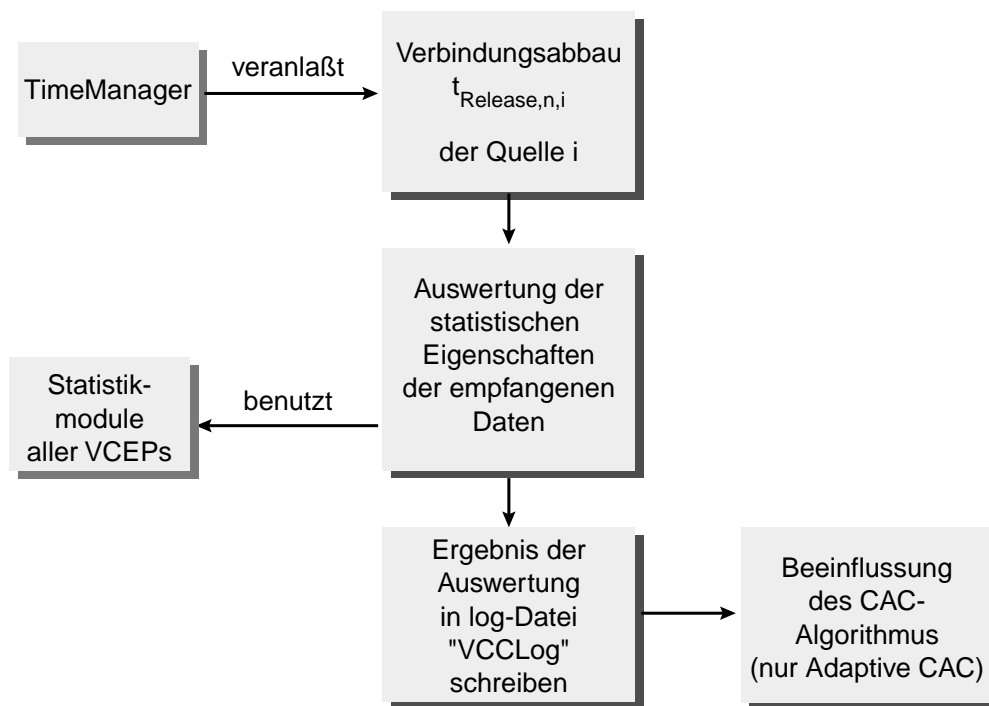


Abbildung 96 : Verbindungsabbau

Nach Abbau der Verbindung werden die gesammelten Qualitätsmerkmale zur späteren Auswertung in einer Datei („VCCLog“) abgespeichert; bei Anwendung der adaptiven CAC erfolgt an dieser Stelle die Beeinflussung der zulässigen Senderparameter (siehe Abschnitt 5.4.3 auf Seite 110).

7.1.5 Auswertung

Nach Beendigung der Simulation erfolgt die Analyse der in der Ergebnisdatei „VCCLog“ aufgezeichneten Qualitätsparameter aller Datentransfers. Die Berechnung des erreichten mittleren Wertdurchsatz \bar{w} erfolgt anhand der Gleichungen aus Abschnitt 5.4.2 auf Seite 108 ff. mithilfe eines *awk*¹-Skriptes. Der Aufbau der Ergebnisdateien und des Auswerteskriptes ist in Anhang F auf Seite 272 ff. beschrieben. Eine graphische Aufbereitung der Ergebnisse wird in den folgenden Abschnitten vorgenommen.

7.2 Simulationslauf 1

Der erste hier vorgestellte Simulationslauf bezieht sich auf das in der folgenden Abbildung dargestellte Netzwerk aus vier Stationen, die über eine VC-Level-Switch verbunden sind. Alle Verbindungen sind als optische Übertragungsstrecken auf Lichtwellenleiterbasis ausgeführt und werden mit einer Übertragungsrate von 155,52 MBit/s und einer Bitfehlerwahrscheinlichkeit von $BER=10^{-12}$ betrieben.

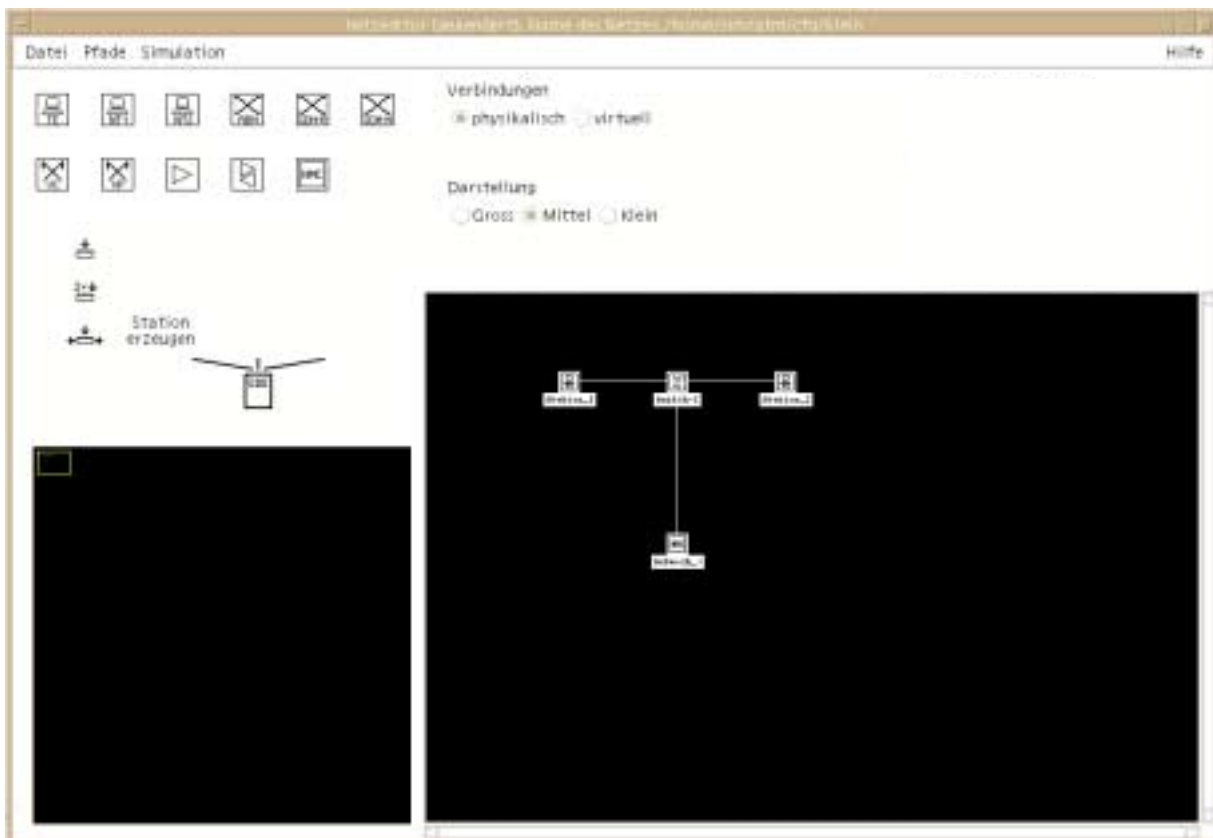
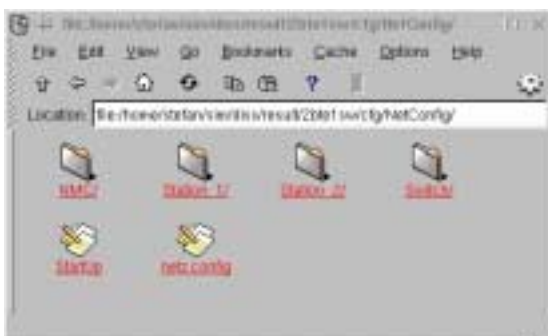


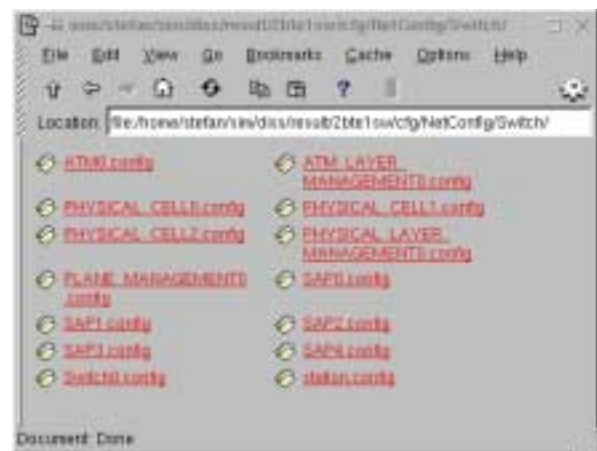
Abbildung 97 : Netzwerk zur Simulation 1 (Darstellung im Neteditor)

1. Unix-Reportgenerator zur automatisierten Textverarbeitung (siehe [Gu89] und [Her92])

Abbildung 98 zeigt die Dateistruktur der Konfigurationsdateien; einige der Dateien sind beispielhaft im Anhang (Abschnitt D auf Seite 237 ff) zu finden.



Netzwerk- und
Quelldefinition



Konfiguration der Station
"Switch"

Abbildung 98 : Konfigurationsdateien Simulationslauf 1

7.2.1 Quellenpool

Die Eigenschaften der Quellen, aus denen die sendende Station („Station_1“ in Abbildung 97 auf Seite 144) auswählt, sind in Tabelle 51 auf Seite 145 angegeben, die genauen Festlegungen sind in Abschnitt D.1.1 auf Seite 237 ff. zu finden.

Nr.	Typ	MBR [kBit/s]	PBR [kBit/s]	α^a	ZZG-DT-T
0,1	CBR	8.448	8.448	1	D
2,3	VBR	2.048	10.000	0,1	MMPP
4,5	VBR	384	1.024	0,2	MMPP
6,11	VBR	10.000	15.000	0,1	M
7,9,10	VBR	5.000	7.500	0,5	M
8	VBR	32	64	0,5	MMDP

Tabelle 51 : Quellen Simulation 1

a. Mittlerer Anteil der Intervalle mit Spitzenbitrate

Diese Quellen definieren den Verkehrsmix, der von „Station_1“ ausgehend auf 155 MBit/s-Links zur „Station_2“ übertragen wird.

7.2.2 Ergebnisse

Das in Abbildung 97 auf Seite 144 gezeigte ATM-Netzwerk wurde unter Verwendung der drei unterschiedlichen CAC-Algorithmen *MeanAllocation*, *PeakAllocation* und *Adaptive* bei ansonsten unveränderten Parametern simuliert. Die mittlere Wiederholrate der Verbindungsaufbauwünsche jeder der 12 Quellen beträgt 10 ms, was bei den festgelegten Verbindungsdauern¹ und der Variation des Lastfaktors α_{rel} (siehe Abschnitt 7.1.1 auf Seite 140) zu einem sehr variablen Verkehrsangebot führt. Während die Dimensionierung des Netzwerkes es erlaubt, den angebotenen Verkehr für Werte von $\alpha_{\text{rel}} \Rightarrow 0$ zu bewältigen, ohne überlastet zu werden, führt $\alpha_{\text{rel}} \approx 1$ zu einem Überangebot, von dem das betrachtete Netzwerk nur einen Teil akzeptieren kann. Objektives Gütekriterium beim Vergleich der CAC-Algorithmen ist das bei gleichem Verkehrsangebot und -mix pro Zeiteinheit über das Netzwerk übertragene Produkt aus Übertragungsqualität und Datenmenge \bar{w} (vgl. Abschnitt 5.4.2 auf Seite 108 ff.). Bevor ein Vergleich zwischen den drei CAC-Verfahren vorgenommen wird, soll an dieser Stelle zunächst der Regelvorgang im Rahmen der *Adaptiven CAC* anhand zweier ausgewählter Beispiele erläutert werden.

Beispiele für Regelvorgänge bei Adaptiver CAC

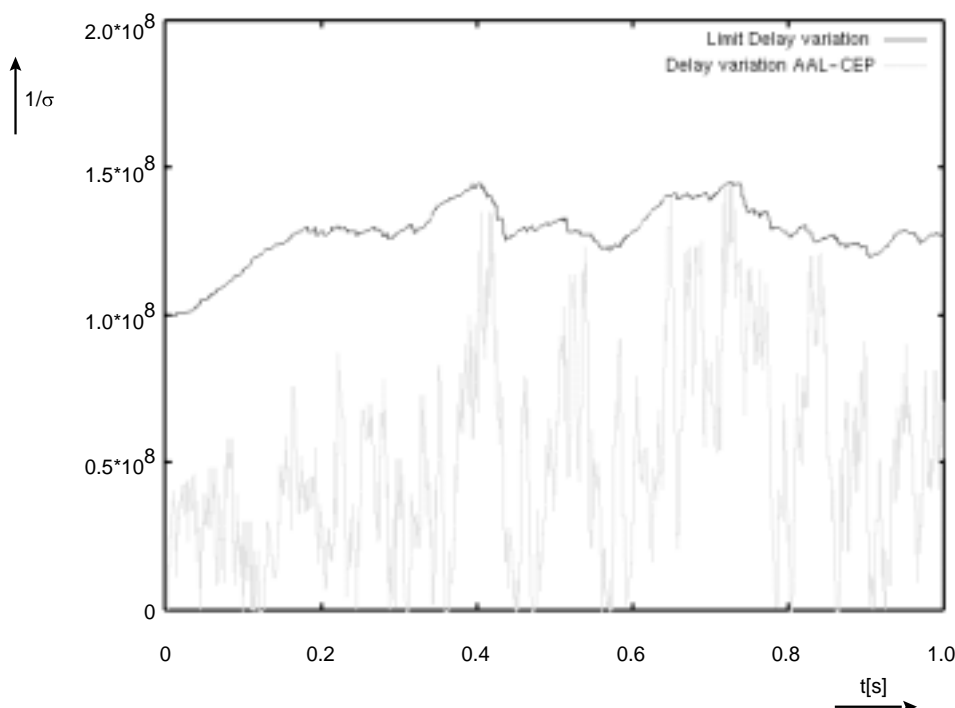


Abbildung 99 : Regelvorgang am Beispiel des Grenzwerts für den Parameter „Delay Variance“

1. Parameter „Duration“ in Abschnitt D.1.1 auf Seite 237 ff.

Während die CAC-Verfahren *MeanAllocation* und *PeakAllocation* ausschließlich die bereits beim Sender allokierten mittleren bzw. Spitzen-Bitraten zur Durchführung der CAC zugrundelegen, verwendet die *Adaptive CAC* Grenzwerte für jeden der in Abschnitt 5.4.2 auf Seite 108 definierten Verkehrsparameter. Ist der „Vorrat“ des Senders bezüglich einer Komponente erschöpft, werden weitere Verbindungen abgelehnt. Die Anpassung der Grenzen erfolgt jeweils beim Abbau einer Verbindung durch Analyse der erreichten Übertragungsqualität (vgl. Abbildung 76 auf Seite 111). Ein Beispiel für das Erreichen eines solchen Grenzwertes und dessen Modifikation zeigt die Abbildung 99 auf Seite 146 für den Qualitätsparameter „Varianz der Paketzwehenankunftsabstände σ “.

Zusätzlich zu der unmittelbaren Begrenzung der zulässigen Bitraten durch den entsprechenden Qualitätsparameter beim Sender beeinflusst jeder andere Qualitätsparameter mittelbar die vom Sender zugelassene Bitrate. Abbildung 100 zeigt den Einfluß eines solchen Eingriffs.

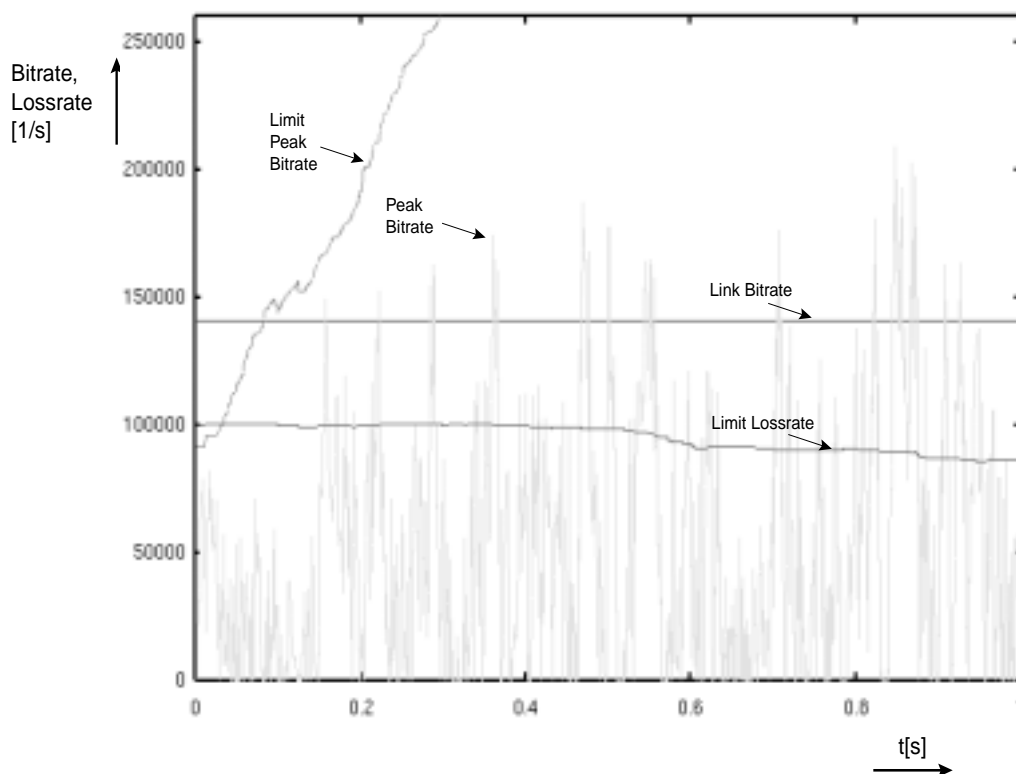


Abbildung 100 :Beispiel eines indirekten Regelvorgangs

An dem mit der Zeit t absinkenden Verlauf des Grenzwertes für die momentan vom Netzwerk zugelassene Paketverlustrate (*Limit Lossrate*) ist abzulesen, daß die Protokollinstanzen im Netzzinnern überlastet sind und demzufolge eine große Anzahl von Paketen verloren gehen. Obwohl der Grenzwert für die Spitzenbitrate beim Sender beständig anwächst (*Limit Peak Bitrate*) und damit wesentlich

höhere Werte für diese Komponente erlauben würde, verhindert die Ablehnung von weiteren Verbindungen aufgrund der schlechten Qualität bezüglich der Paketverlustrate die Zunahme der genutzten Spitzenbitrate (*Peak Bitrate*). Auf diese Weise verhindert der Qualitätsparameter *Packet Lossrate* die Steigerung der zugelassenen Last, bevor sich Einbußen in dem dafür „zuständigen“ Parameter *Peak Bitrate* zeigen.

7.2.2.1 Auslastung des Netzwerks

Abbildung 101 auf Seite 148 zeigt die Auslastung ρ der ersten Teilstrecke zwischen der sendenden BTE „Station 1“ und der VC-Level-Switch „Switch-1“ als Verhältnis der mittleren Anzahl der durch Nutzdatenzellen belegten Zellslots im Verhältnis zur maximal möglichen Anzahl. Zu erkennen sind hier drei unterschiedliche Bereiche, den Bereich I mit wenig angebotenen Verkehr bis $\alpha_{rel} \approx 0,4$, der Bereich III für $\alpha_{rel} > 0,9$ mit zahlreichen (~1.200) Verbindungsaufbauten und ein Übergangsbereich II zwischen den Bereichen I und III. Die Begründung für die Unterschiede in der akzeptierten Last in den drei Bereichen sind im folgenden Abschnitt dargelegt.

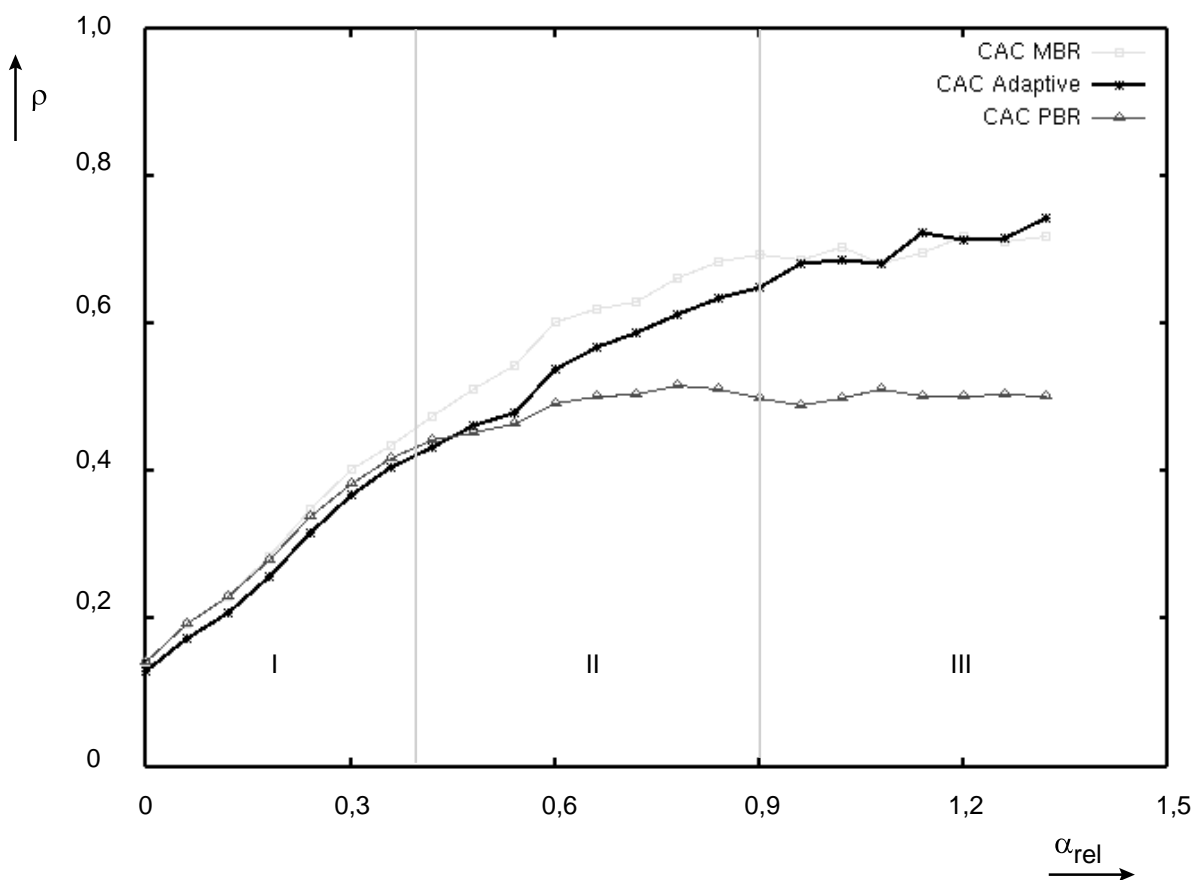


Abbildung 101 : Mittlere Auslastung des Links von Station_1 zur VC-Switch

7.2.2.2 Vergleich der CAC-Verfahren

Die erreichte Verbesserung des Produktes \bar{w} aus Datenmenge und erreichter Übertragungsqualität bei Verwendung der *Adaptiven CAC* gegenüber den einfacheren CAC-Varianten zeigt die Abbildung 102 auf Seite 149.

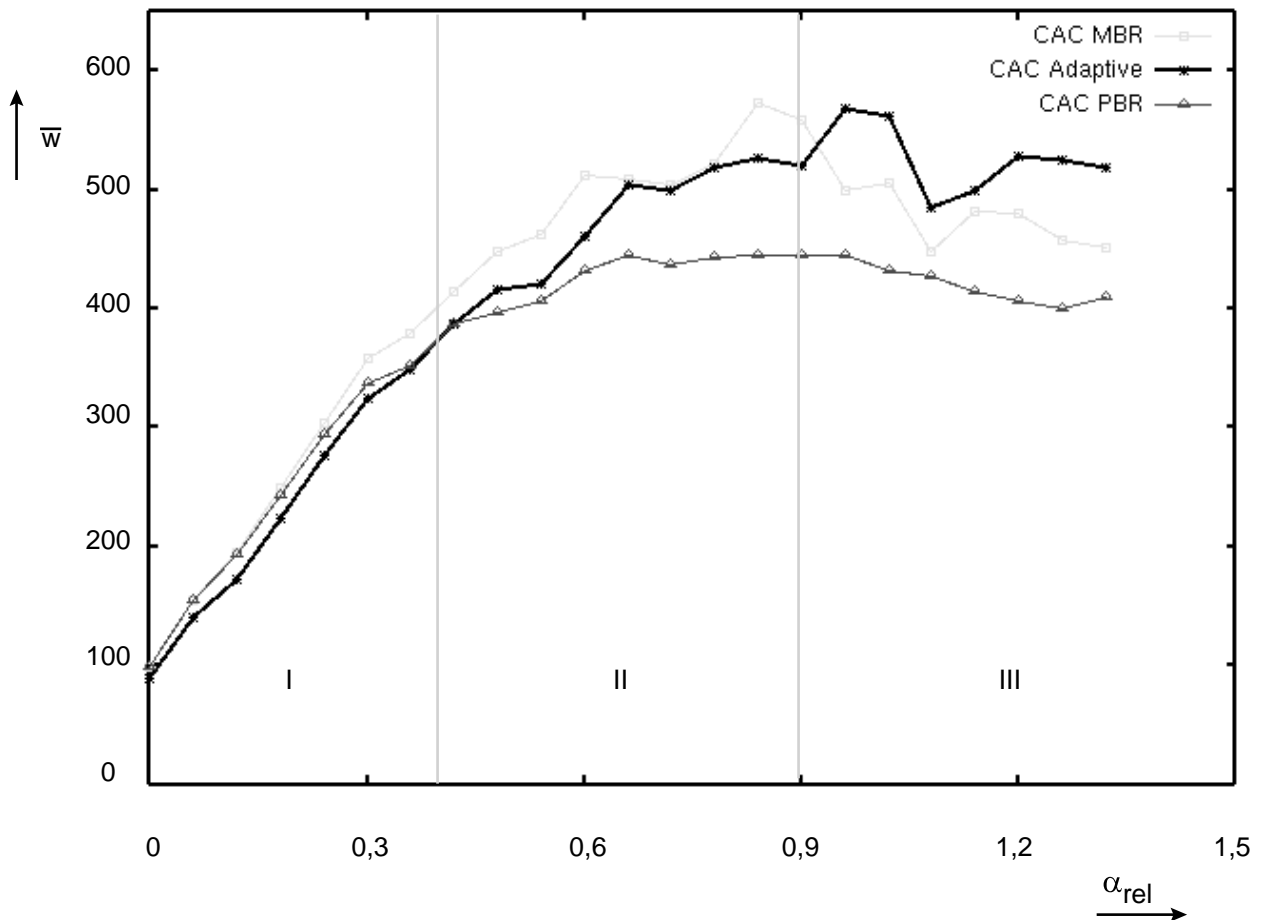


Abbildung 102 :Vergleich der CAC-Verfahren in Simulation 1

Die Berechnung der erfolgreich übertragenen Qualitätsmenge erfolgt durch Analyse der während der Simulation gemessenen Qualitätsparameter mithilfe von Gl. 32 auf Seite 110. Der von einer Quelle mit folgenden QoS-Parametern erzeugte Verkehr wird als Referenzqualitätsanspruch $\overrightarrow{QoS_{Ref}}$ mit dem Wert „1.0“ angenommen:

- Mittlere Bitrate : 1.0 MBit/s
- Spitzenbitrate : 1.0 MBit/s

- Bitfehlerwahrscheinlichkeit: 10^{-6}
- Paketverlustwahrscheinlichkeit: 10^{-8}
- Verzögerung: 10^{-4} s
- Varianz der Verzögerung: 10^{-7} s²
- 3. Moment der Verzögerung: 10^{-9} s³
- 4. Moment der Verzögerung: 10^{-11} s⁴

Im Bereich niedriger Last im Netzwerk ($\alpha_{\text{rel}} \Rightarrow 0$) können nahezu alle Verbindungswünsche akzeptiert werden. Hier sind die einfachen Verfahren *MeanAllocation* und *PeakAllocation* der adaptiven Regelung leicht überlegen; der Grund hierfür ist im Regelalgorithmus zu suchen. Da in diesen Lastsituationen nur wenige Verbindungen auf- und abgebaut werden, erfolgt die Nachführung der Grenzwerte seltener und mit der durch die Dauer der Verbindung vorgegebenen Verzögerung. Dadurch ergibt sich eine zu pessimistische Abschätzung der Leistungsfähigkeit des Netzwerks, insbesondere in der Anlaufphase. Diesem Effekt kann jedoch durch die in Abschnitt 7.4 auf Seite 153 genannten Verbesserungsvorschläge begegnet werden. Im Hochlastfall ($\alpha_{\text{rel}} \geq 1$) werden mehr Verbindungen an das Netzwerk herangetragen, als dieses aufgrund seines internen Aufbaus ohne hohen Qualitätsverlust bewältigen kann. Hier zeigt sich die Überlegenheit des adaptiven Verfahrens, das sich flexibel auf die jeweilige Leistungsfähigkeit des Übertragungsnetzwerks einstellt (siehe z.B. Abbildung 100 auf Seite 147). Im Vergleich zur CAC mittels *PeakAllokation* wird eine größere Anzahl von Verbindungsaufbauwünschen akzeptiert und erfolgreich mit leichtem Qualitätsverlust übertragen. Die auf Betrachtung der mittleren Bitrate basierende CAC akzeptiert zu optimistisch Verbindungen. Durch die Konkurrenzsituation bei der Bearbeitung durch die Netzknoten erleiden diese erhebliche Qualitätseinbußen aufgrund von Wartedauern und Pufferüberläufen. Vergleicht man in Abbildung 102 auf Seite 149 die verbleibende Restmenge an Daten mit akzeptabler Qualität, so zeigt sich im Bereich $0.9 \leq \alpha_{\text{rel}} \leq 1.3$ ein Gewinn von bis zu 15% gegenüber der *MeanAllocation* und von 30% gegenüber der *PeakAllocation* pro Zeiteinheit.

7.3 Simulationslauf 2

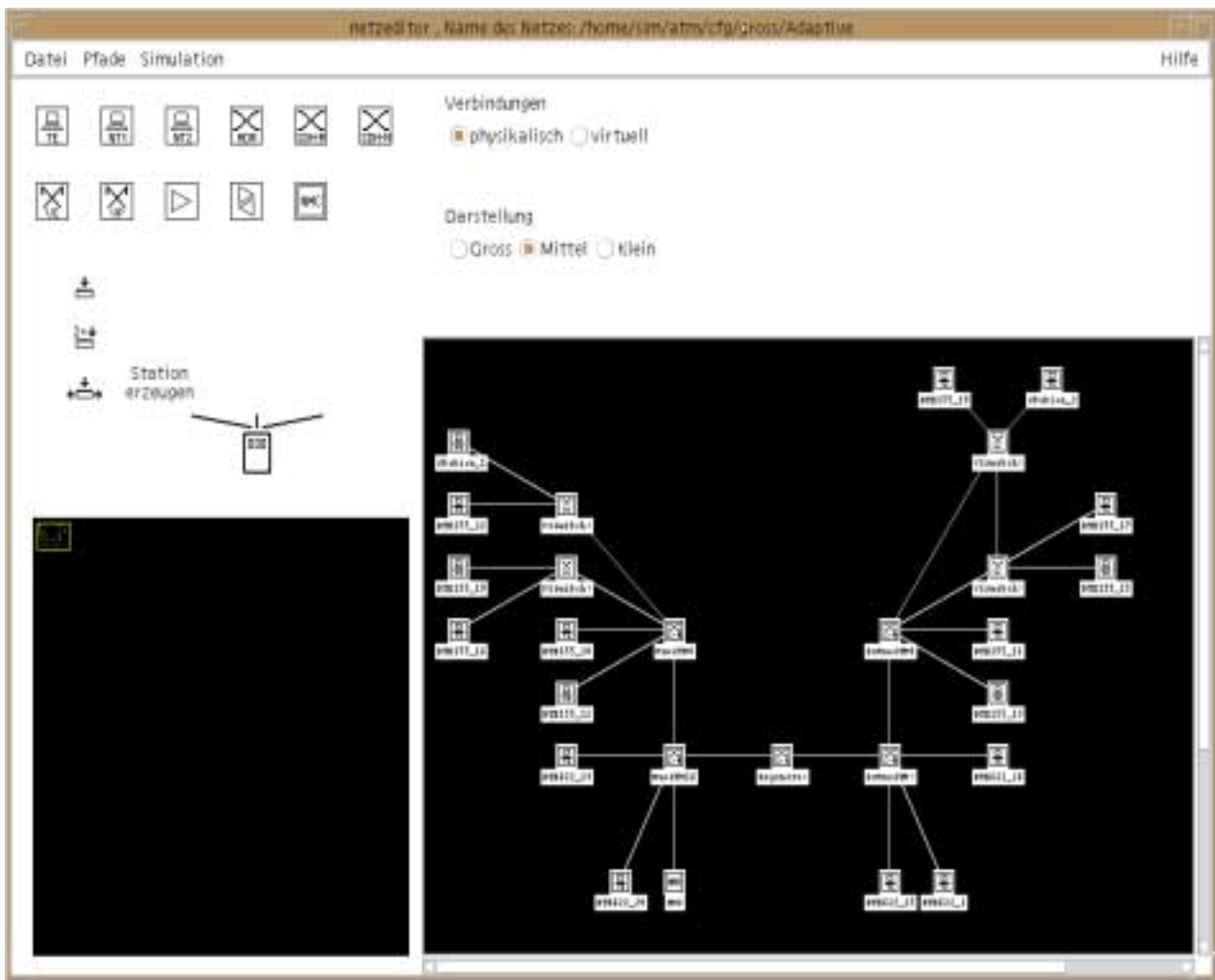


Abbildung 103 :Netzwerkconfiguration Simulationslauf 2

Die Eignung der Adaptiven CAC auch für größere realistischere Netzwerke wurde durch Simulation des in Abbildung 103 gezeigten ATM-Netzwerks nachgewiesen. Im Netzzinnern sind SDH-Übertragungssysteme der Ebenen STM-1 bis STM-16 (vgl. Tabelle 20 auf Seite 52) angeordnet, wie sie in den heutigen ATM-Netzen zum Einsatz kommen. Im Teilnehmerzugangsbereich sind die dafür vorgesehenen zellbasierten Schnittstellen mit 155 und 622 MBit/s (vgl. Abschnitt 3.6.2 auf Seite 48 ff.) eingesetzt. Betrachtet wird auch hier der Transport multimedialer Datenströme zwischen den Endsystemen „Station_1“ und „Station_2“ unter Verwendung der drei CAC-Algorithmen *MeanAllocation*, *PeakAllocation* und *Adaptive*.

Das Ergebnis der Simulation bei Verwendung des gleichen Verkehrsmix wie im Simulationslauf 1 (siehe Tabelle 51 auf Seite 145) zeigt die folgende Abbildung.

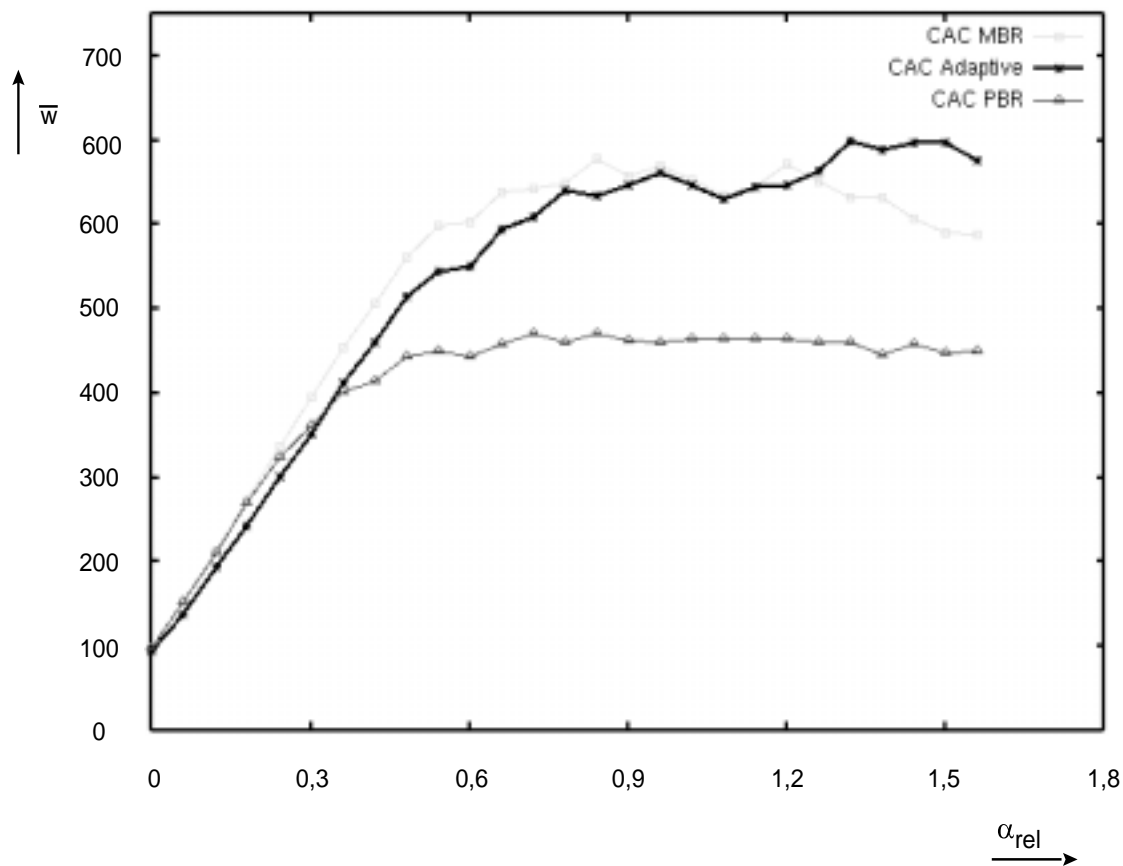


Abbildung 104 :Vergleich der CAC-Verfahren im Netzwerk aus Abbildung 103 auf Seite 151

Auch in diesem realistischeren Anwendungsfall zeigt sich die Überlegenheit des adaptiven Verfahrens im Hochlastfall. Die Interpretation des Ergebnisses verläuft analog zu der Argumentation in Abschnitt 7.2.2.2 auf Seite 149. Der im Vergleich zum ersten Simulationslauf höhere Wertdurchsatz, sowie der spätere Einsatz des Gewinns gegenüber der *MeanAllocation*, beruht auf der größeren Anzahl von Datenpuffern auf dem Übertragungsweg und der hohen Übertragungs- und Verarbeitungsgeschwindigkeit im Netzinnern. Durch die Vielzahl der beteiligten Netzknoten können sich die Pakete in Überlastsituationen über mehrere Puffer verteilen, wodurch die Anzahl von Paketverlusten sinkt. Daher erfolgt der Einbruch bei Verwendung der *MeanAllocation* erst bei höheren angebotenen Lasten.

7.4 Zusammenfassung

Die Simulationsergebnisse zeigen die Vorteile des entwickelten *Adaptiven CAC-Verfahrens* bei Hochlastsituationen im B-ISDN auf der Basis des ATM. Die Zunahme an Bandbreitenbedarf durch Verbreitung moderner multimedialer Kommunikationsendgeräte wird das, durch die Einführung optischer Übertragungswege und leistungsfähiger Hardware, derzeit herrschende Überangebot an Bandbreite innerhalb kurzer Zeit abbauen. Dies wird die miteinander konkurrierenden Netzwerkbetreiber zu einer möglichst hohen Auslastung ihrer Netzbetriebsmittel zwingen. Der entwickelte adaptive CAC-Algorithmus erlaubt dem Netzbetreiber eine höhere Anzahl von Verbindungen pro Zeiteinheit bei geringerer Qualitätseinbuße über sein Netzwerk zu transportieren als beim Einsatz einfacher CAC-Verfahren auf der Basis der mittleren oder Spitzenzellrate. Die Simulation des ATM-Netzwerks unter Verwendung der von der ITU standardisierten Protokolle zeigt die Überlegenheit des adaptiven Verfahrens insbesondere bei häufigen Verbindungsaufbauwünschen von Quellen mit unterschiedlichen Qualitätsansprüchen an das Netzwerk. Hier findet der Algorithmus immer sehr schnell eine „passende“ Quelle, um die Kapazitäten der sendenden Stationen bezüglich aller Komponenten optimal auszulasten.

Eine Verbesserung des Verfahrens könnte durch die Verkürzung der Totzeit des Regelalgorithmus erreicht werden, etwa durch ständige Analyse der Datenqualität während des Datentransfers. Als zusätzliche Erweiterung des Verfahrens könnte der beschriebene Algorithmus seine Erkenntnisse über die in den unterschiedlichen Lastsituationen durchsetzbare Datenqualität sammeln und die Entscheidung über die Akzeptanz einer neuen Verbindung aufgrund dieser „Erfahrungswerte“ treffen. Dies würde die langsame Annäherung an die für eine neue Lastsituation optimalen Grenzwerte des Adaptiven CAC-Algorithmus vermeiden.

Zusammenfassung und Ausblick

Zukünftige Kommunikationsformen mit multimedialen Komponenten stellen spezielle Anforderungen an das universelle Übertragungsnetzwerk der Zukunft, das B-ISDN auf der Basis des ATM. Dieses Netzwerk wird, im Gegensatz zu den früheren stark spezialisierten Netzwerken, zur Übertragung von Daten unterschiedlichster Herkunft eingesetzt, was eine aufwendige Verkehrslenkung erfordert. Ziel ist es, die erforderliche *Quality of Service* für alle Datenströme zwischen den Endsystemen sicherzustellen sowie eine möglichst effiziente Ausnutzung der Netzwerkressourcen zu erreichen. Versuche, mithilfe der Bedientheorie für alle Situationen geeignete, analytische Lösungen für diese Probleme zu finden, scheiterten bisher an der Komplexität der statistischen Bindungen zwischen Ankunfts- und Bedienprozessen.

Die im Rahmen dieser Arbeit erstellte laufzeitoptimierte Simulationssoftware vermeidet weitreichende Annahmen bei der Lösung von bedientheoretischen Problemen im Umfeld von ATM Netzwerken. Sie enthält Softwaremodule zur exakten Simulation der Vorgänge innerhalb der ATM-spezifischen Schichten des B-ISDN-Referenzmodells. Die implementierten Protokolle entsprechen dabei den Vorgaben des jeweiligen Standards der ITU. Unabhängig von den Abläufen auf der Protokollebene ist die zur Ermittlung der Bediendauern in den Protokollinstanzen erforderliche Nachbildung der Hardware entwickelt worden. Modifikationen an den Hardwareeigenschaften im Rahmen von Parameterstudien sind ohne Eingriffe in den Simulationscode möglich. Die Operationen auf der Hardwareebene werden durch flexible Modelle nachgebildet, die an beliebige reale Hardwarestrukturen angepaßt werden können.

Mithilfe dieser Simulationssoftware wurde ein neues Verfahren zur Verkehrslenkung von multimediale Datenströmen auf dem *Call-Level* entwickelt und dessen Leistungsfähigkeit in realitätsnah gestalteten ATM-Netzwerken nachgewiesen. Das Verfahren paßt sich selbständig an die sich ändernden Lastfälle im Netzinnern an und im Hochlastfall wird eine größere Datenmenge bei moderaten Qualitätseinbußen übertragen, als dies im Vergleich zu den von der ITU vorgesehenen Verfahren möglich ist. Das Verfahren kann von dem Netzbetreiber als in seinen Verantwortungsbereich fallenden Verbindungsannahmealgorithmus implementiert werden. Der Algorithmus sorgt sowohl für eine optimale Auslastung der Netzbetriebsmittel, als auch dafür, daß im Interesse der Netzwerknutzer eine hohe Qualität bei den Empfangsdaten gewährleistet ist. Da dem Verfahren keinerlei analytisch formulierte Zusammenhänge zwischen Netzwerkparametern und Regelparametern zugrundeliegen, ist es flexibel auch für neue Dienste mit unvorhersehbaren Verkehrsprofilen einsetzbar. Schließlich wurden Vorschläge beschrieben mit dem Ziel, die Effizienz des Algorithmus noch weiter zu steigern.

- [Bla95] Black, U.D.: *ATM-Foundation for broadband networks*, Prentice Hall Inc., Englewood Cliffs 1995, ISBN 0-13-297178-X
- [Blu96] Blum, D.: *Programmierung grundlegende Softwaretools zur Auswertung und Durchführung zeitdiskreter, ereignisorientierter Simulationen in C++*, Projektarbeit am FG Übertragungstechnik, Universität Gesamthochschule Kassel, 1996
- [Bol97] Bolz, A.: *Entwicklung eines graphischen Interfaces für die Simulation eines Nachrichtennetzwerkes unter OSF/Motiv*, Diplomarbeit am FG Übertragungstechnik, Universität Gesamthochschule Kassel, 1997
- [Bra87] Bradley, P.: *A guide to simulation*, Springer Verlag New York, Berlin, Heidelberg 1987, ISBN 0-387-96467-3
- [Bre95] Brede, H.-J.: *Programmieren mit OSF/Motiv*, Addison-Wesley GmbH 1995, ISBN 3-89319-727-3
- [Bro94] Brooktree Corp.: *Bt8222, ATM Receiver/Transmitter with UTOPIA Product Brief*, Brooktree Corporation, San Diego 1994
- [Bro95] Brooktree Corp.: *Bt8230, ATM Segmentation and Reassembly Controller (SRC) Product Brief*, Brooktree Corporation, San Diego 1995
- [Bru93] Bruneel, H.: *Discrete-Time models for communicating systems including ATM*, Kluwer Academic Publishers, Norwell (Massachusetts) 1993, ISBN 0-7923-9292-2
- [Cyp91] Cypser, R.J.: *Communications for cooperating systems : OSI, SNA, and TCP/IP*, Addison-Wesley Publishing Company, Inc. 1991, ISBN 0-201-50775-7
- [Dei95] Deimling, D.: *C++ Tuning*, Verlag Heinz Heise GmbH & Co KG, Hannover 1995, ISBN 3-88229-054-4
- [Doc] Dokumentierte Klassenbibliothek der ATM-Simulationssoftware, <http://www.uet.e-technik.uni-kassel.de/atm/befehle.html>

- [E164] International Telecommunication Organization: *Recommendation E.164 „Numbering plan for the ISDN era“*, Genf 1991
- [G652] International Telecommunication Organization: *Recommendation G.652 „Characteristics of a single mode optical fibre cable“*, Genf 1993
- [G703] International Telecommunication Organization: *Recommendation G.703 „Physical/electrical characteristics of hierarchical digital interfaces“*, Genf 1991
- [G707] International Telecommunication Organization: *Recommendation G.707 „Synchronous digital hierarchie bit rates“*, Genf 1993
- [G708] International Telecommunication Organization: *Recommendation G.708 „Network node interface for the synchronous digital hierarchie“*, Genf 1993
- [G709] International Telecommunication Organization: *Recommendation G.709 „Synchronous multiplexing structure“*, Genf 1993
- [G.823] International Telecommunication Organization: *Recommendation G.823 „The control of jitter and wander within digital networks which are based on the 2048 kbit/s hierarchy“*, Genf 1993
- [G.824] International Telecommunication Organization: *Recommendation G.824 „The control of jitter and wander within digital networks which are based on the 1544 kbit/s hierarchy“*, Genf 1993
- [G957] International Telecommunication Organization: *Recommendation G.957 „Optical interfaces for equipments and systems relating to the synchronous digital hierarchy“*, Genf 1993
- [Gag95] Gagnaire, M., Kofman, D., *An analytical Description of the Packet Train Model for LAN traffic characterization*, Ecole Nationale Supérieure des Télécommunications, Paris 1995
- [Gan95] Ganos, P.A. et al.: *A Novel Dynamic Priority Scheduling Method For Multiple Classes Of ATM Traffic in an ATM Statistical Multiplexer*, aus [Kou95]
- [Gne83] Gnedenko, B.W.: *Handbuch der Bedienungstheorie*, Akademie-Verlag Berlin 1983, ISSN 0076-5430
- [Gu95] Gu X., Sohraby K., Vaman D.R.: *Control and Performance in Packet, Circuit and ATM Networks*, Kluwer Academic Publishers, Boston, Dordrecht, London 1995
- [Gué91] Guérin, R. et al.: *Equivalent Capacity and its application to bandwidth allocation*, IEEE Journal on Selected Areas in Communications, Vol.9, No. 7, 1991
- [Gul88] Gulbins, J.: *UNIX, Version 7 bis System V.3*, 3. überarbeitete und erweiterte Auflage; Springer Verlag Berlin; Heidelberg; New York 1988, ISBN 3-540-19248-4
- [Hal92] Halsall, F.: *Data Communications, computer networks and OSI*, Addison Wesley Limited 1992, ISBN 0-201-56506-4
- [Har98] Hartmann, T.: *Simulation des Protokolls SAAL im B-ISDN gemäß ITU-T-Empfehlungen Q.2130 und Q.2110 in C++*, Diplomarbeit am FG Übertragungstechnik, Universität Gesamthochschule Kassel, 1998
- [Her91] Herrmann, D.: *Statistik in C*, Vieweg Verlagsgesellschaft, Braunschweig 1991, ISBN 3-528-05129-9

-
- [Her92] Herold, H.: *UNIX und seine Werkzeuge*, Addison Wesley, Bonn; Paris; Reading [u.a.] 1992, ISBN 3-89319-381-2
- [Her96] Herzog, M.: *Modellierung und Auswertung statistischer Vorgänge in Nachrichtennetzwerken*, Projektarbeit am FG Übertragungstechnik, Universität Gesamthochschule Kassel, 1996
- [Hom97] Hommel, K.: *Simulation des Protokolls AAL 5 im B-ISDN gemäß ITU-T-Recommendation I.363 in C++*, Diplomarbeit am FG Übertragungstechnik, Universität Gesamthochschule Kassel, 1997
- [Hym91] Hyman, J.M. et al.: *Realtime Scheduling with Quality of Service Constraints*, IEEE Journal on Selected Areas in Communications, Vol.9, No. 7, 1991
- [I321] International Telecommunication Organization: *Recommendation I.321 „B-ISDN Protocol Reference Model and its application“*, Genf 1991
- [I356] International Telecommunication Organization: *Recommendation I.356 „B-ISDN ATM layer cell transfer performance“*, Genf 1996
- [I361] International Telecommunication Organization: *Recommendation I.361 „B-ISDN ATM layer specification“* Genf, 1993
- [I362] International Telecommunication Organization: *Recommendation I.362 „B-ISDN ATM adaption layer (AAL) functional description“*, Genf 1993
- [I363] International Telecommunication Organization: *Recommendation I.363 „B-ISDN ATM adaption layer (AAL) specification“*, Genf 1993
- [I363Add] International Telecommunication Organization: *Recommendation I.363 „Addendum B-ISDN ATM adaption layer specification“*, Genf 1993
- [I363_1] International Telecommunication Organization: *Recommendation I.363.1 „B-ISDN ATM adaption layer specification: Type 1 AAL“*, Genf 1996
- [I363_2] International Telecommunication Organization: *Recommendation I.363.2 „B-ISDN ATM adaption layer specification: Type 2 AAL“*, Genf 1997
- [I363_3] International Telecommunication Organization: *Recommendation I.363.3 „B-ISDN ATM adaption layer specification: Type 3/4 AAL“*, Genf 1997
- [I363_5] International Telecommunication Organization: *Recommendation I.363.5 „B-ISDN ATM adaption layer specification: Type 5 AAL“*, Genf 1997
- [I365_1] International Telecommunication Organization: *Recommendation I.365.1 „B-ISDN ATM adaptation layer sublayers: Frame relaying service specific convergence sublayer (FR-SSCS)“*, Genf 1993
- [I371] International Telecommunication Organization: *Recommendation I.371 „Traffic control and congestion control in B-ISDN“*, Genf 1995
- [I432.1] International Telecommunication Organization: *Recommendation I.432.1 „User-network interface - Physical layer specification: General characteristics“*, Genf 1999
- [I432.3] International Telecommunication Organization: *Recommendation I.432.3 „User-network interface - Physical layer specification: 1544 kbit/s and 2048 kbit/s operation“*, Genf 1999
-

- [I432.4] International Telecommunication Organization: *Recommendation I.432.4 „User-network interface - Physical layer specification: 51 840 kbit/s operation“*, Genf 1999
- [I432.5] International Telecommunication Organization: *Recommendation I.432.5 „User-network interface - Physical layer specification: 25 600 kbit/s operation“*, Genf 1997
- [I610] International Telecommunication Organization: *Recommendation I.610 „B-ISDN Operation and maintenance principles and functions“*, Genf 1993
- [IEEE802_2] Institute of Electrical and Electronic Engineers: *8802-2 : 1998 (ISO/IEC) [ANSI/IEEE 802.2, 1998 Edition] Information technology--Telecommunications and information exchange between systems--Local and Metropolitan area networks--Specific requirements--Part 2: Logical link control*, New York 1998
- [IEEE802_3] Institute of Electrical and Electronic Engineers: *802.3, 1998 Edition Information technology--Telecommunications and information exchange between systems--Local and metropolitan area networks--Specific requirements--Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications*, New York 1998
- [IEEE802_3ab] Institute of Electrical and Electronic Engineers: *802.3ab-1999 Supplement to IEEE Std 802.3, 1998 Edition: Physical Layer Parameters and Specifications for 1000 Mb/s Operation over 4-Pair of Category 5 Balanced Copper Cabling, Type 1000BASE-T*, New York 1999
- [IEEE802_4] Institute of Electrical and Electronic Engineers: *802.4h-1997 IEEE Standard for Token Passing Bus Access Method and Physical Layer Specifications--Alternative use of BNC-connectors and Manchester Encoded Signaling Methods for Single Channel Bus Physical Layer Entities [Description]*, New York 1997
- [Jos95] Josuttis, N.: *Objektorientiertes Programmieren in C++; von der Klasse zur Klassenbibliothek*, Addison-Wesley, Bonn; Paris, Reading 1995, ISBN 3-89319-637-4
- [Kam96] Kampe, T.: *Programmierung eines dateigestützten Netzwerkkonfigurationstools unter C++*, Diplomarbeit am FG Übertragungstechnik, Universität Gesamthochschule Kassel, 1996
- [Ker75] Kerner, H.: *Rechnernetze nach OSI*, Addison-Wesley, Bonn, Paris, Reading 1993, 2. Auflage ISBN 3-89319-632-3
- [Kim95] Kimball, P.E.: *The X Toolkit Cookbook*, Prentice Hall , Englewood Cliffs 1995, ISBN 0-13-973132-6
- [Kle75] Kleinrock und Tobagi: *IEEE Transactions on Communications COM-23 (1975)*, S. 1400-1416
- [Kle75_2] Kleinrock, L.: *Queuing Systems, Volume I : Theory*, John Wiley & Sons, Inc. 1975, ISBN 0-471-49110-1
- [Kle75_3] Kleinrock, L.: *Queuing Systems, Volume II : Computer Applications*, John Wiley & Sons, Inc. 1976, ISBN 0-471-49110-X
- [Knu75] Knuth, D.E.: *The art of computer programming Vol 1 : Fundamental algorithms*, Addison-Wesley, Reading, London, Amsterdam 1975, ISBN 0-201-03822-6

-
- [Knu75_2] Knuth, D.E.: *The art of computer programming Vol 2 : Seminumerical algorithms*, Addison-Wesley, Reading, London, Amsterdam 1975, ISBN 0-201-03809-9
- [Knu75_3] Knuth, D.E.: *The art of computer programming Vol 3 : Sorting and Searching*, Addison-Wesley, Reading, London, Amsterdam 1975, ISBN 0-201-03803-X
- [Kou95] Kouvastos, D.D., *Performance Modelling and Evaluation of ATM Networks, Volume I*, Chapman & Hall, London, Glasgow, Weinheim, New York, Tokyo, Melbourne 1995, ISBN 0-412-71140-0
- [Kya95] Kyas, O.: *ATM-Netzwerke: Aufbau, Funktion, Performance*, DATACOM-Verlag, Bergheim 1995, ISBN 3-89238-108-9
- [Leh95] Lehnert, R.: *Kontroll- und Dimensionierungsaspekte bei der Realisierung von Breitbandnetzen mit ATM*, Philips Telecommunication Review Vol.52, No.1, 1995
- [Li90] Li, S., Mark, J.W.: *Traffic Characterization for Integrated Services Networks*, IEEE Transactions on Communications, Vol. 38, No. 8, 1990
- [Li93] Li, S., Hwang C.: *Queue Response to Input Correlation Functions: Discrete Spectral Analysis*, IEEE Transactions on Communications, Vol. 1, No. 1, 1993
- [Lie92] Liebl, F.: *Simulation: problemorientierte Einführung*, R. Oldenburg Verlag GmbH, München 1992, ISBN 3-486-22109-4
- [Lip95] Lipman, S.: *C++*, Addison-Wesley, Bonn; München 1995, ISBN 3-89319-375-8
- [Mit89] Mitsuru, N.: *Basic Characteristics of Variable Rate Video Coding in ATM Environment*, IEEE Journal on Selected Areas in Communications, Vol.7, No. 5, 1989
- [Mit95] Mitrou, N.: *Bursty Traffic Modelling and Multiplexing Performance Analysis in ATM Networks: A three Moment Approach*, aus [Kou95]
- [Nec95] NEC Corporation: *DATA Sheet μ PD98402A, Local ATM Sonet Framer*, Document No. S10835EJ1V0DS00 (1st edition), Published December 1995, NEC Corporation 1995
- [Nec95_2] NEC Corporation: *User's Manual μ PD98402, Local ATM Sonet Framer*, Document No. IEU-1385A, Published March 1995, NEC Corporation 1995
- [Nec95_3] NEC Corporation: *User's Manual μ PD98401, Local ATM SAR Chip*, Document No. IEU-1384A, Published March 1995, NEC Corporation 1995
- [Neu93] Neumann, K.: *Operations Research*, Verlag Hanser, München, Wien 1993, ISBN 3-446-15771-9
- [Neuf93] Neufeld, G.W.: *Parallel Host Interface for an ATM Network*, IEEE Network Juli 1993
- [Ngo91] Ngo, B.: *Queuing Analysis of Traffic Access Control Strategies with Preemptive and Nonpreemptive Disciplines in Wideband Integrated Networks*, IEEE Journal on Selected Areas in Communications, Vol.9, No. 7, 1991
- [Ngu97] Nguyen, T.: *Simulation des Protokolls AAL 3/4 im B-ISDN gemäß ITU-T-Recommendation I.363 in C++*, Diplomarbeit am FG Übertragungstechnik, Universität Gesamthochschule Kassel, 1997
- [Nye92] Nye : *Xlib Programming Manual*, O'Reilly 1992
-

- [Nye92_2] Nye : *Xlib Reference Manual for Version 11*, O'Reilly 1992
- [Nye92_3] Nye : *X Windows System User guide*, O'Reilly 1992
- [OSF95_1] Open Software Foundation Inc.: *OSF/Motif Programmer's Guide, 1995*, ISBN 0-13-143158-7
- [OSF95_2] Open Software Foundation Inc.: *OSF/Motif Programmer's Reference, 1995*, ISBN 0-13-143166-8
- [Pap89] Papoulis, A.: *Probability, random variables and stochastic processes*, 6th edition, McGraw-Hill, Singapore 1989
- [Pet67] Peterson, W.W.: *Prüfbare und korrigierbare Codes*, R.Oldenburger Verlag GmbH, München 1967
- [Pit96] Pits, J.M., Ph.D.: *Introduction to ATM design and performance*, John Wiley & Sons Ltd., Chichester 1996, ISBN 0-471-96340-2
- [Pry93] Prycker, Martin de: *Ansynchronous transfer mode : solution for broadband ISDN*, 2nd Edition, Ellis Horwood Limited, Sussex 1993, ISBN 0-13-178542-7
- [Q7xx] International Telecommunication Organization: *Recommendations Q.700- Q.788*, Genf 1988-1995
- [Q931] International Telecommunication Organization: *Recommendation Q.931 „Digital subscriber signalling system no. 1 (DSS1)- User-network interface layer 3 specification for basic call control“*, Genf 1993
- [Q2100] International Telecommunication Organization: *Recommendation Q.2100 „B-ISDN Signalling ATM Adaption Layer (SAAL) Overview Description“*, Genf 1994
- [Q2110] International Telecommunication Organization: *Recommendation Q.2110 „B-ISDN ATM Adaption Layer - Service Specific Connection oriented Protocol (SSCOP) , Genf 1994*
- [Q2120] International Telecommunication Organization: *Recommendation Q.2120 „B-ISDN Metasignalling protocol“*, Genf 1995
- [Q2130] International Telecommunication Organization: *Recommendation Q.2130 „B-ISDN Signalling ATM Adaption Layer - Service Specific Coordination Function for support of Signalling at the User Network Interface“*, Genf 1994
- [Q27xx] International Telecommunication Organization: *Recommendations Q.2721- Q.2767*, Genf 1996-1998
- [Q.2931] International Telecommunication Organization: *Recommendation Q.2931 „Broadband Integrated Services Digital Network (B-ISDN) - Digital subscriber signalling system no. 2 (DSS 2)- User-network interface (UNI) - Layer 3 specification for basic call/connection control“*, Genf 1995
- [Q.2931_1] International Telecommunication Organization: *Amendment 1 to Recommendation Q.2931 „Broadband Integrated Services Digital Network (B-ISDN) - Digital Subscriber Signalling System No. 2 - User-network interface (UNI) layer 3 specification for basic call/connection“*, Genf 1997

-
- [Q29xx] International Telecommunication Organization: *Recommendations Q.2931- Q.2767*, Genf 1995-1998
- [Ram91] Ramamurthy, G.: *Distributed Source Control: A Network Access Control for Integrated Broadband Packet Networks*, IEEE Journal on Selected Areas in Communications, Vol.9, No. 7, 1991
- [Rath97] Rathgeb, E.: *ATM-Infrastruktur für die Hochleistungskommunikation*, Springer Verlag Berlin, Heidelberg, New York 1997, ISBN 3-540-60370-0
- [Rau95] Rauchhaus, A.: *Ereignisorientierte Simulation der Protokolle im zellbasierten Physical- und ATM-Layer im B-ISDN gemäß den ITU-T-Recommendations I.361, I.432 und I.610 unter C++*, Diplomarbeit am FG Übertragungstechnik, Universität Gesamthochschule Kassel, 1995
- [RFC768] Defense Advanced Research Projects Agency: *User Datagram Protocol*, DARPA RFC 768, Arlington
- [RFC791] Defense Advanced Research Projects Agency: *Internet Protocol*, DARPA RFC 791, Arlington
- [RFC793] Defense Advanced Research Projects Agency: *Transmission Control Protocol*, DARPA RFC 793, Arlington
- [Sai91] Saito, H., Kawarasaki M., Yamada H.: *An Analysis of Statistical Multiplexing in an ATM Transport Network*, IEEE Journal on selected areas in Communications Vol.9, No.3, April 1991
- [Sai91_2] Saito, H.: *Dynamic Admission Control in ATM Networks*, IEEE Journal on Selected Areas in Communications, Vol.9, No. 7, 1991
- [Sed94] Sedgewick R.: *Algorithmen in C++*, Addison-Wesley, Bonn; München; Paris [u.a.] 1. Nachdruck 1994, ISBN 3-89319-462-2
- [Sha45] Shannon, C.E.: *Communication Theory of Secrecy Systems*, Bell System Technical Journal, Vol. 28, 1945
- [Sig95] Sigmund, G.: *ATM - Die Technik des Breitband-ISDN*, R. v. Decker's Verlag, Heidelberg 1994, 2. Auflage, ISBN 3-7685-3994-6
- [Ske93] Skelly et al.: *A Histogram-Based Model for Video Traffic Behavior in an ATM Multiplexer*, IEEE/ACM Transactions on Networking, Vol.1, No. 4, 1993
- [Smi93] Smith J.M., Traw B.S.: *Giving Applications Access to GBit/s Networking*, IEEE Network Juli 1993
- [Spie90] Spiegel, M.: *Statistik*, McGraw-Hill, Hamburg, New York 1990, 2. Auflage, ISBN 3-89028-908-8
- [Sta93] Stapelberg, S.: *UNIX System V.4 für Einsteiger und Fortgeschrittene: Bourne-Shell, Korn-Shell, C-Shell, TCP/IP, UUCP, E-Mail & News, X Window System*, Addison-Wesley, Bonn; Paris; Reading [u.a.] 1993, ISBN 3-89319-433-9
- [Str94] Stroustrup, B.: *Die C++-Programmiersprache*, Addison-Wesley, Bonn; München; Paris [u.a.] 5. Nachdruck 1995, ISBN 3-89319-386-3
- [Swe92] Sweeny, P.: *Codierung zur Fehlererkennung und Fehlerkorrektur*, Hanser-Verlag 1992, ISBN 3-446-16439-1
-

- [Tak91] Takagi, Y.: *Priority Assignment Control of ATM Line Buffers with multiple QoS-classes*, IEEE Journal on Selected Areas in Communications, Vol.9, No. 7, 1991
- [Tur92] Turner, J.S.: *Managing Bandwidth in ATM Networks with Bursty Traffic*, IEEE Network Sept. 1992
- [UNI31] ATM Forum Technical Committee : *User-network interface (UNI) specification Version 3.1*, Prentice Hall 1995, ISBN 0-13-393828-X
- [Ver89] Verbiest, W., Pinno L., *Variable Rate Video Codec for ATM-Networks*, IEEE Journal on selected Areas in Communications, Vol. 7, No.5, June 1989
- [Wur98] Wursthorn, P.: *Simulation des Protokolls AAL 1 im B-ISDN gemäß ITU-T-Recommendation I.363 in C++*, Diplomarbeit am FG Übertragungstechnik, Universität Gesamthochschule Kassel, 1998
- [Wee97] Weese, D.: *Simulation der SDH-basierten Physikalischen Schicht im B-ISDN*, Diplomarbeit am FG Übertragungstechnik, Universität Gesamthochschule Kassel, 1997
- [X200] International Telecommunication Organization: *Recommendation X.200 „Information Technology - Open Systems Interconnection - Basic Reference Model : The basic model“*, Genf 1994
- [X213] International Telecommunication Organization: *Recommendation X.213 „Information Technology - Network service definition for Open Systems Interconnection“*, Genf 1992
- [Zit95] Zitterbart, M.: *Hochleistungskommunikation, Band I Technologie und Netze*, R.Oldenburger Verlag GmbH, München 1995, ISBN 3-486-22707-6
- [Zit96] Zitterbart, M.: *Hochleistungskommunikation, Band II Transportdienste und -protokolle*, R.Oldenburger Verlag GmbH, München 1996, ISBN 3-486-23088-3

A

AAL	ATM Adaptation Layer
ABR	Available Bitrate
ABT	ATM Block Transfer
ACK	Acknowledge
AIS	Alarm Indication Signal
AL	Alignment
α_{rel}	Relatives Angebot
ANP	AAL Type 2 Negotiation Procedure
ANSI	American National Standards Institute
ATC	ATM Transfer Capability
ATM	Asynchronous Transfer Mode
ATMM	ATM Layer Management Entity
AU	Administrative Unit
AUU	ATM User-to-User Indication
awk	Pattern Scanning and Processing Language

B

BASize	Buffer Allocation Size
BCR	Block Cell Rate
BEP	Bit Error Probability
BER	Bit Error Rate
BGN	Begin PDU
BGAK	Begin Acknowledge PDU

BGREJ	Begin Reject PDU
B-ISDN	Broadband ISDN
B-ISUP	B-ISDN User Part
B-NT1	B-ISDN Network Termination 1
B-NT2	B-ISDN Network Termination 2
BOM	Begin of Message
BR	Buffer Release
B-TA	B-ISDN Terminal Adapter
Btag	Begin Tag
B-TE1	B-ISDN Terminal Equipment with S _B -Interface
B-TE2	B-ISDN Terminal Equipment without S _B -Interface

C

C	Link Capacity
CAC	Connection Admission Control
CAD	Computer Aided Design
CBR	Continuous Bitrate
CC	Congestion Control
CCITT	Comité Consultatif International Télégraphique et Téléphonique
CDF	Cumulative Probability
PDF	Density Function
CDV	Cell Delay Variation
CEC	Cell Error Code
CEI	Connection Endpoint Identifier
CEP	Connection Endpoint
CEQ	Customer Equipment
CID	Channel Identifier
CLP	Cell Loss Priority
CLP	Cell Loss Probability
CMD_n	n-th Central Moment of delay
CMI	Coded Mark Inversion
C-n	SDH Container Level n
COM	Continuation of Message
CPI	Common Part Indicator
CPCS	Common Part Convergence Sublayer
CPS	Common Part Sublayer
CPU	Central Processing Unit

CRC	Cyclic Redundancy Check
CS	Convergence Sublayer
CSI	Convergence Sublayer Indication
CSMA/CD	Carrier Sense Multiple Access / Collision Detection
CT	Connection Termination
CTD	Cell Transfer Delay

D

D	Deterministic Process
DAT	Digital Audio Tape
DBR	Deterministic Bit Rate Transfer Capability
DS	Digital Section
DT	Delayed Transmission

E

EB	Error Block
EDC	Error Detection Code
END	End
ENDAK	End Acknowledge
EOM	End of Message
EP	Endpoint
ε_T	Empfindlichkeit
ER	Error Recovery
ERAK	Error Recovery Acknowledge
Etag	End Tag
EWSD	Elektronisches Wählsystem Digital

F

F1-OAM	OAM-Flow RS Level
F2-OAM	OAM-Flow DS Level
F3-OAM	OAM-Flow TP Level
F4-OAM	OAM-Flow VP Level
F5-OAM	OAM-Flow VC Level
FDM	Frequency Division Multiplex
FEC	Forward Error Correction
FERF	Far End Receive Failure
ff	und folgende Seiten

FIFO	First in - First out
ffs	For further studies
FR-SSCS	SSCS for Frame Relay
FTP	File Transfer Protocol

G

G	General Process
GMDP	Generally Modulated Deterministic Process
GFC	Generic Flow Control

H

HEC	Header Error Control
------------	----------------------

I

IBT	Intrinsic Burst Tolerance
ID	Interface Data
IEEE	Institute of Electrical and Electronic Engineers
ISDN	Integrated Services Digital Network
ISO	International Standards Organization
IT	Immediate Transmission
ITU	International Telecommunication Union
IP	Internet Protocol
IWF	Interworking Function

J

K

L

L	Puffergröße
λ	Rate
λ_{WDH}	Wiederholrate
LAN	Local Area Network
Length	Length of CPCS-PDU Payload
LI	Length Indication
LIFO	Last in - First out

LoQ	Loss of Quality
LSB	Least Significant Bit
LT	Line Termination

M

M	Mandatory
M	Markov Process
M	More
MBR	Mean Bitrate
MBS	Maximum Burst Size
MBS	Monitored Block Size
MD	Management Data
MID	Multiplexing Identification
ML	Maximum Length
MM	Message Mode
MID	Multiplexing Identification
MMDP	Markov Modulated Deterministic Process
MMPP	Markov Modulated Poisson Process
MPD	Mean Packet Delay
MSB	Most Significant Bit
MSVC	Metasignalling Virtual Channel
MTP	Message Transfer Part

N

N_{Acc}	Number of Accepted Connections
ND	Sum of N Deterministic Processes
NE	Negative Exponential
NFS	Network File System
NIC	Network Interface Card
NIC	Number of Included Cells
NMB	Number of Monitored Blocks
NMC	Network Management Center
NNI	Network Node Interface
NPC	Network Parameter Control
N_Q	Number of Quality Parameters
NRM	Network Resource Management
NRZ	Non Return to Zero

NSAP Network Service Access Point

O

O Optional
OAM Operation, Administration and Maintenance
OSF Offset Field
OSI Open Systems Interconnection

P

P Parity
PAD Padding
PBR Peak Bitrate
PCI Protocol Control Information
PCM Pulscodemodulation
PCM30 1. Europäische Multiplexstufe in der PDH
PCM120 2. Europäische Multiplexstufe in der PDH
PCR Peak Cell Rate
PDF Probability Density Function
PDH Plesiochronous Digital Hierarchy
PDU Protocol Data Unit
PHY Physical Layer
PL Physical Layer
PLP Packet Loss Probability
PM Physical Medium Sublayer
POH Path Overhead
PPT Packet Payload Type
PSN Physical Layer Cell Sequence Number
PT Payload Type
PLR Packet Loss Rate
PTR Pointer

Q

Q_n Quelle n
QoS Quality of Service
QoS_{Att} QoS Attained
QoS_{Lim} QoS Limit
QoS_{Ref} QoS Reference

QoS_{Req} QoS Required

R

R Reserved
RAM Random Access Memory
RFC Request for Comment
 ρ Last
RM Ressource Management
RN Retrieval Number
RPC Remote Procedure Call
RS Reception Status
RS Regenerator Section
RS Resynchronization
RS-Code Read-Solomon-Code
RSAK Resynchronization Acknowledge

S

S-AAL Signalling-AAL
SAP Service Access Point
SAR Segmentation and Reassembly
S_B B-ISDN Reference Point
SBR Statistical Bit Rate Transfer Capability
SCR Sustainable Cell Rate
SD Sequenced Data
SDH Synchronous Digital Hierarchy
SDU Service Data Unit
SE Service Element
SMTP Simple Mail Transport Protocol
SM Streaming Mode
SN Sequence Number
SOH Section Overhead
SONET Synchronous Optical Network
SRTS Synchronous Residual Time Stamp
SSCF Service Specific Coordination Function
SSCOP Service Specific Connection Oriented Protocol
SSCS Service Specific Convergence Sublayer
SSM Single Segment Message

ST	Segment Type
STAT	Solicited Status Information
STF	Start Field
STM-n	Synchronous Transport Module Level n

T

t	Zeitpunkt
T	(Takt-) Dauer
T_B	B-ISDN Reference Point
T_B	Bediendauer
TC	Traffic Control
TC	Transmission Convergence Sublayer
TCP	Transmission Convergence Protocol
T_D	Verbindungsdauer
TDM	Time Division Multiplex
TE-2	Terminal Equipment without S _B Interface
Tln	Teilnehmer
TP	Transmission Path
TPT	Transmission Path Termination
TUG	Tributary Unit Group
TU-n	Tributary Unit Level n
T_{OFF}	Dauer Inaktivitätsintervall
T_{ON}	Dauer Aktivitätsintervall
TV	Television

U

UDP	User Datagram Protocol
UNI	User Network Interface
UPC	Usage Parameter Control
UU	User-to-User
UUI	User-to-User Indication

V

VBR	Variable Bitrate
VC	Virtual Channel
VC-n	Virtual Container Level n
VCC	Virtual Channel Connection

VCEP	Virtual Channel Endpoint
VCI	Virtual Channel Identifier
VP	Virtual Path
VPC	Virtual Path Connection
VPEP	Virtual Path Endpoint
VPI	Virtual Path Identifier
VPN	Virtual Private Network

W

\bar{w}	Mittlerer Wert
w_k	Wert der Verbindung k
WAN	Wide Area Network

X

Y

Z

z	Zustand
ZZG-DT-PL	ZZG für Paketlängen
ZZG-DT-PF	ZZG für Bitmuster
ZZG-DT-T	ZZG für Sendezeitpunkte
ZZG-VCC	ZZG für Aufbau von VCCs

α_{rel}	Relatives Angebot
χ_{α}	Kritischer Wert zum Signifikanzniveau α der χ^2 -Verteilung
$\delta(t)$	δ -Distribution
ε_{τ}	Empfindlichkeit
$E\{\mathbf{x}\}$	Erwartungswert
f	Freiheitsgrad
$f_x(x)$	Verteilungsdichtefunktion
$F_x(x)$	Verteilungsfunktion
$\Phi(s)$	Momentengenerierende Funktion kontinuierlicher Zufallsvariablen
η	Erwartungswert
L	Puffergröße
λ	Rate
λ_{WDH}	Wiederholrate
L_{OQ}	Qualitätsverlust
M_i	Statistisches Moment i -ter Ordnung
N_{Acc}	Anzahl akzeptierter Verbindungen
N_{Q}	Anzahl der Qualitätsparameter
p	Wahrscheinlichkeit
p_0	Wahrscheinlichkeit, ein leeres System vorzufinden
P_{ij}	Bedingte Wahrscheinlichkeit
QoS	Dienstgüte
QoS_{Att}	Erreichte Qualitätsanspruch
QoS_{Lim}	Maximal zulässige Qualitätsansprüche
QoS_{Ref}	Qualitätsanspruch des Referenzverkehrs

QoS_{Req}	Qualitätsanspruch
ρ	Last
σ^2	Varianz
t	Zeitpunkt
T	(Takt-) Dauer
$\Gamma(z)$	Momentengenerierende Funktion diskreter Zufallsvariablen
T_B	Bediendauer
T_D	Verbindungsdauer
T_{OFF}	Dauer Inaktivitätsintervall
T_{ON}	Dauer Aktivitätsintervall
V	Verweildauer
\bar{w}	Mittlerer Wert
w_k	Wert der Verbindung k
W	Wartedauer
\vec{x}	Vektorielle Größe
x	Zufallsvariable
Z_i	Zentralmoment i-ter Ordnung
z	Zustand
z_α	Kritischer Wert zum Signifikanzniveau α der Normalverteilung

Anhang A Datenelemente

A.1 Informationsfeld der F1- bzw. F3-OAM-Zelle

Oktett	Belegung	Bedeutung
1	R	Reserved
2	AIS	<i>Alarm indication signal</i> Alarmsignal in Richtung der Übertragung, daß eine Fehler entdeckt und alarmiert wurde
3	PSN	Sequenznummer der Physical Layer OAM-Zelle
4	NIC	<i>Number of included cells</i>
5		Anzahl überwachter Zellen zwischen der letzten OAM-Zelle und der aktuellen
6	MBS	<i>Monitoring Block Size</i> Anzahl der Bytes in einem Überwachungsblock
7	NMB-EDC	<i>Number of monitored blocks</i> Anzahl der Überwachten Blöcke
8	EDC-B1	<i>Error detection Code Block 1</i> BIP ^a -8 über den 1. Block aus MBS Zellen
9	EDC-B2	<i>Error detection Code Block 2</i>
10	EDC-B3	<i>Error detection Code Block 3</i>
11	EDC-B4	<i>Error detection Code Block 4</i>
12	EDC-B5	<i>Error detection Code Block 5</i>
13	EDC-B6	<i>Error detection Code Block 6</i>
14	EDC-B7	<i>Error detection Code Block 7</i>
15	EDC-B8	<i>Error detection Code Block 8</i>
16	R	Reserved
...
29	R	Reserved
30	FERF	<i>Far end receive failure</i> Alarmierung der Station in der entgegengesetzten Übertragungsrichtung über aufgetretene Übertragungsfehler
31	NMB-EB	<i>Number of monitored blocks at far end</i> Anzahl der festgestellten Fehler in den folgenden Oktetts
32	EB 2	EB 1 <i>Block Error Block 1 & 2</i> Anzahl Paritätsfehler in den Blöcken 1 und 2

Tabelle 52 : : Informationsfeld OAM-F1/F3-Zelle

Oktett	Belegung		Bedeutung
33	EB 4	EB 3	<i>Block Error Block 3 & 4</i> Anzahl Paritätsfehler in den Blöcken 3 und 4
34	EB 6	EB 5	<i>Block Error Block 5 & 6</i> Anzahl Paritätsfehler in den Blöcken 5 und 6
35	EB 8	EB 7	<i>Block Error Block 7 & 8</i> Anzahl Paritätsfehler in den Blöcken 7 und 8
36	R		Reserved
...
46	R		Reserved
47	CEC		<i>Cell error control</i>
48			CRC ^b -10 zum Schutz der Informationen im Payload

Tabelle 52 : : Informationsfeld OAM-F1/F3-Zelle

- a. Bit interleaved parity
b. Cyclic redundancy check

A.2 Aufbau einer Metasignalisierungszelle

Bit								Octet
8	7	6	5	4	3	2	1	
Protocol Discriminator ^a								1
Protocol Version								2
Message Type ^b								3
Reference Identifier ^c								4
								5
Signalling Configuration ^d								6
Signalling Virtual Channel Identifier A ^e								7
								8
Signalling Virtual Channel Identifier B								9
								10
Point to point Signalling Virtual Channel Cell Rate ^f								11
Cause ^g								12

Tabelle 53 : Aufbau der Metasignalisierungszellen

Bit								Octet
8	7	6	5	4	3	2	1	
Service Profile Identifier ^h								13-23
0	0	0	0	0	0	0	0	24-40
0	0	0	0	0	0	0	0	41
0	0	0	0	0	0	0	0	42
0	0	0	0	0	0	0	0	43
0	0	1	0	1	0	0	0	44
CRC ($x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$)								45-48

Tabelle 53 : Aufbau der Metasignalisierungszellen

- a. Protokollbezeichner (immer 00000001)
- b. vergl. Abbildung 65
- c. Zufallszahl zur Kennzeichnung eines ASSIGN-Requests
- d. Point- to point oder Point-to Multipoint Signalisierung
- e. Zugeteilte Nummer für einen Signalisierungskanal
- f. Bitrate des Signalisierungskanals
- g. Grund für eine Ablehnung
- h. Informationen über das Netzwerk

A.3 Datenformat des AAL-Typs 2

Parameter	Bedeutung
Channel ID CID	Nummer des bidirektionalen AAL-2-Kanals
Length Indicator LI	Länge des Nutzdatenfeldes
Packet Payload Type PPT	OAM-PDUs : PPT=3, alle anderen Werte für Nutzerdaten oder AAL-2-Signalisierung
User-to-User-Indication UUI	Wird transparent zum CPS-Nutzer übertragen
Header Error Control HEC	Fehlerschutz durch Zyklischen Code mit dem Generatorpolynom x^5+x^2+1
Information INFO	Informationsfeld

Tabelle 54 : Parameter der CPS-PDU

Parameter	Bedeutung
Offset Field OSF	Position des ersten CPS-Paket relativ zum Ende des STF
Sequence Number SN	Nummer der CPS-PDUs modulo 2
Parity P	Ungerade Parität über das STF

Tabelle 54 : Parameter der CPS-PDU

A.4 Datenformat des AAL-Typs 3/4

PDU-Feld	Bedeutung
Segment Type	BOM, COM, EOM, oder SSM
Sequence Number	Numerierung modulo 15
Multiplexing Identifikation	Kennzeichnung der Verbindung beim Multiplex über eine ATM-Verbindung
Length Indication	Länge des Nutzdatenfeldes
Cyclic Redundancy Check	Prüfsumme über die SAR-PDU, Generatorpolynom: $x^{10}+x^9+x^5+x^4+x+1$

Tabelle 55 : Bedeutung der Felder in der SAR-PDU

PDU-Feld	Bedeutung
Common Part Identifier	Interpretation der Werte im Feld „BASize“ und „Length“, bisher ist nur 00000000 standardisiert.
Beginning Tag	Entspricht dem Wert des Feldes „End Tag“, ermöglicht die Überprüfung der Zusammengehörigkeit von Header und Trailer einer CPCS-PDU
BASize	Informiert den Empfänger über die zu reservierende Puffergröße, die zum Empfang der CPCS-PDU notwendig ist. Die Bedeutung der Zahl hängt vom Wert des Feldes „CPI“ ab. (CPI=0: Größe ist in Oktetts angegeben)
Padding	Ergänzt das Nutzdatenfeld zu einem Vielfachen von 4 Oktett
Alignment	Ergänzt den CPCS-Trailer zu einem Vielfachen von 4 Oktetts

Tabelle 56 : Bedeutung der Felder in der CPCS-PDU

PDU-Feld	Bedeutung
End Tag	Trägt denselben Wert wie das Feld „Btag“
Length	Die Länge des CPCS-Nutzdatenfeldes. Die Interpretation des Wertes hängt vom Wert des „CPI“-Feldes ab (CPI=0: Länge ist in Oktetts angegeben)

Tabelle 56 : Bedeutung der Felder in der CPCS-PDU

A.5 Datenformat des AAL-Typs 5

PDU-Feld	Bedeutung
Common Part Identifier	Auffüllen des CPCS-Trailers auf 64 Bit
CPCS-Payload	Nutzdatenfeld mit einer Länge zwischen 1 und 65535 Oktetts
Padding	Ergänzt das Nutzdatenfeld zu einer Länge von 48 Oktetts
CPCS-UU	Transparente Übertragung von CPCS-User-to-User-Informationen
Length	Die Länge des CPCS-Nutzdatenfeldes.
Cyclic Error Check	Schutz der CPCS-PDU vor Übertragungsfehlern. Generatorpolynom: $x^{32}+x^{26}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$

Tabelle 57 : Bedeutung der Felder in der CPCS-PDU

A.6 Signalisierungsnachrichten

Nachrichtenelement	Richtung	Typ	Länge ^a
Protocol discriminator Kennzeichnung des verwendeten Protokolls	beide	M	1
Call reference Nummer zur Identifikation des Signalisierungsvorganges	beide	M	4
Message type Kennzeichnung des Nachrichtentyps (SETUP, CONNECT, RELEASE ...)	beide	M	2
Message length Länge des Informationsfeldes	beide	M	2
AAL parameters (vergl. Tabelle 59 auf Seite 184) Forderung der Verbindung an den AAL-Typ	beide	O	4-21

Tabelle 58 : Elemente der SETUP-Nachricht

Nachrichtenelement	Richtung	Typ	Länge ^a
ATM traffic descriptor (vergl. Tabelle 60 auf Seite 185) Verkehrsparameter der Verbindung, z.B. Zellrate	beide	M	12-30
Broadband bearer capability Eigenschaften der Nutzverbindung, z.B. Bitrate, Zeitbindung, Verbindungsorientiert/verbindungslos, die von allen Netzknoten innerhalb der Verbindung akzeptiert werden müssen	beide	M	6-7
Broadband higher layer information Forderungen an die höheren Schichten (End-to-End, Layer 4-7)	beide	O	4-13
Broadband repeat indicator Verhalten bei Existenz mehrerer alternativer Informationselemente	beide	O	4-5
Broadband low layer information Forderungen an die niederen Schichten (End-to-End, Layer 1-3)	beide	O	4-17
Called party number Identifikation des Zielsystems Adreßformat : E.164 [E164] oder ISO NSAP gem. X.213 [X213]	beide	M	4-25
Called party subaddress Subadresse eines ATM-Endgerätes	beide	O	4-25
Calling party number vergl. Called party number	beide	O	4-26
Calling party subaddress vergl. Called party subaddress	beide	O	4-25
Connection Identifier VPI und VCI der zugewiesenen Verbindung	N->U	M	9
QoS parameter Definition der Qualitätsanforderung der Verbindung	beide	M	6
Broadband sending complete Anzeige, ob all Wählziffern in der SETUP-Nachricht enthalten sind	beide	O	4-5

Tabelle 58 : Elemente der SETUP-Nachricht

Nachrichtenelement	Richtung	Typ	Länge ^a
Transit network selection Auswahl des Übermittlungsnetzes	U->N	O	4-8
M	Mandatory (erforderlich)		
O	Optional (optional)		
N->U	Vom Netzwerk zum Tln.		
U->N	Vom Tln. zum Netzwerk		
beide	Vom Tln. zum Netzwerk und umgekehrt		

Tabelle 58 : Elemente der SETUP-Nachricht

a. in Oktetts

Bits								Oktett
8	7	6	5	4	3	2	1	Nr.
Subtype Identifier								6
1	0	0	0	0	1	0	1	
Subtype								6.1
CBR Rate Identifier								7
1	0	0	0	0	1	1	0	
CBR Rate								7.1
Multiplier Identifier								8
1	0	0	0	0	1	1	1	
Multiplier								8.1
Multiplier								8.2
Source Clock Frequency Recovery Method Identifier								9
1	0	0	0	1	0	0	0	
Source Clock Frequency Recovery Method								9.1
Error Correction Method Identifier								10
1	0	0	0	1	0	0	1	
Error Correction Method								10.1
Structured Data Transfer Blocksize Identifier								11
1	0	0	0	1	0	1	0	
Structured Data Transfer Blocksize								11.1

Tabelle 59 : Nachrichtenelement ATM Adaption Layer-Typ1 Parameter

Bits								Oktett
8	7	6	5	4	3	2	1	Nr.
Partially Filled Cells Identifier								12
1	0	0	0	1	0	1	1	
Partially Filled Cells Method								12.1

Tabelle 59 : Nachrichtenelement ATM Adaption Layer-Typ1 Parameter

Bits								Oktett
8	7	6	5	4	3	2	1	Nr.
ATM user cell rate (Information Element Identifier)								1
0	1	0	1	1	0	0	1	
1	Coding Standard		IE Instruction Field					2
Length of ATM user cell rate contents								3
Length of ATM user cell rate contents								4
Forward Peak Cell Rate (CLP=0) Identifier								5
1	0	0	0	0	0	1	0	
Forward Peak Cell Rate								5.1
Forward Peak Cell Rate								5.2
Forward Peak Cell Rate								5.3
Backward Peak Cell Rate (CLP=0) Identifier								6
1	0	0	0	0	0	1	1	
Backward Peak Cell Rate								6.1
Backward Peak Cell Rate								6.2
Backward Peak Cell Rate								6.3
Forward Peak Cell Rate (CLP=0+1) Identifier								7
1	0	0	0	0	1	0	0	
Forward Peak Cell Rate								7.1
Forward Peak Cell Rate								7.2
Forward Peak Cell Rate								7.3
Backward Peak Cell Rate (CLP=0+1) Identifier								8
1	0	0	0	0	0	1	1	
Backward Peak Cell Rate								8.1

Tabelle 60 : Nachrichtenelement ATM Traffic Descriptor

Bits								Oktett
8	7	6	5	4	3	2	1	Nr.
Backward Peak Cell Rate								8.2
Backward Peak Cell Rate								8.3
Forward Sustainable Cell Rate (CLP=0) Identifier								9
1	0	0	0	1	0	0	0	
Forward Sustainable Cell Rate								9.1
Forward Sustainable Cell Rate								9.2
Forward Sustainable Cell Rate								9.3
Backward Sustainable Cell Rate (CLP=0) Identifier								10
1	0	0	0	1	0	0	1	
Backward Sustainable Cell Rate								10.1
Backward Sustainable Cell Rate								10.2
Backward Sustainable Cell Rate								10.3
Forward Sustainable Cell Rate (CLP=0+1) Identifier								11
1	0	0	1	0	0	0	0	
Forward Sustainable Cell Rate								11.1
Forward Sustainable Cell Rate								11.2
Forward Sustainable Cell Rate								11.3
Backward Sustainable Cell Rate (CLP=0+1) Identifier								12
1	0	0	1	0	0	0	1	
Backward Sustainable Cell Rate								12.1
Backward Sustainable Cell Rate								12.2
Backward Sustainable Cell Rate								12.3
Forward Maximum Burst Size (CLP=0) Identifier								13
1	0	1	0	0	0	0	0	
Forward Maximum Burst Size								13.1
Forward Maximum Burst Size								13.2
Forward Maximum Burst Size								13.3
Backward Maximum Burst Size (CLP=0) Identifier								14
1	0	1	0	0	0	0	1	
Backward Maximum Burst Size								14.1
Backward Maximum Burst Size								14.2

Tabelle 60 : Nachrichtenelement ATM Traffic Descriptor

Bits								Oktett
8	7	6	5	4	3	2	1	Nr.
Backward Maximum Burst Size								14.3
Forward Maximum Burst Size (CLP=0+1) Identifier								15
1	0	1	1	0	0	0	0	
Forward Maximum Burst Size								15.1
Forward Maximum Burst Size								15.2
Forward Maximum Burst Size								15.3
Backward Maximum Burst Size (CLP=0+1) Identifier								16
1	0	1	1	0	0	0	1	
Backward Maximum Burst Size								16.1
Backward Maximum Burst Size								16.2
Backward Maximum Burst Size								16.3
Best Effort Indicator								17
1	0	1	1	1	1	1	0	
Traffic Management Options Identifier								18
1	0	1	1	1	1	1	1	
Reserved						Tag- ging Back- ward	Tag- ging Forward	18.1

Tabelle 60 : Nachrichtenelement ATM Traffic Descriptor

Anhang B Service Primitives

B.1 Zwischen Physical Layer und ATM-Layer

Name	Parameter	Bedeutung
PHY_DATA_REQUEST	Zelle	Befehl zum Senden einer ATM-Zelle
PHY_DATA_INDICATION	Zelle	Übergabe einer empfangenen Zelle an die ATM-Protokollinstanz

Tabelle 61 : Serviceprimitive zwischen Physical Layer und ATM-Layer

B.2 Zwischen ATM-Layer und ATM-Layermanagement

Name	Parameter	Bedeutung
ATMM_DATA_REQUEST	ATM-SDU, Congestion Indication, Submitted loss priority, PHY-CEI	Befehl zum Senden einer ATM-Zelle vom ATM-Layermanagement
ATMM_DATA_INDICATION	ATM-SDU, Congestion Indication, Received loss priority, PHY-CEI	Übergabe einer empfangenen Zelle an die ATM-Layermanagement-Protokollinstanz

Tabelle 62 : Serviceprimitive zwischen ATM-Layer und AAL Layer

B.3 Zwischen ATM-Layer und AAL-Layer

Name	Parameter	Bedeutung
ATM_DATA_REQUEST	ATM-SDU, Submitted loss priority ^a , congestion indication ^b , ATM-user-to-ATM-user indication ^c	Befehl zum Senden einer ATM-Zelle
ATM_DATA_INDICATION	ATM-SDU, congestion indication, ATM-user-to-ATM-user indication	Übergabe einer empfangenen Zelle an die AAL-Protokollinstanz

Tabelle 63 : Serviceprimitive zwischen ATM-Layer und AAL Layer

a. Priorität der Zelle

b. Anzeige, daß die Zelle über ein Netzelement in *congestion* übertragen wurde (optional)

c. Vom ATM-Layer transparent zu übertragen

B.4 Zwischen AAL-Benutzer und AAL-1-Layer

Name	Parameter	Bedeutung
AAL-UNITDATA-REQUEST	DATA (M ^a), STRUCTURE (O ^b)	Befehl zum Senden der AAL-SDU zur AAL-Peer-Entity
AAL-UNITDATA-INDICATION	DATA (M), STRUCTURE (O), STATUS (O)	Übergabe einer empfangenen Zelle an die AAL-Protokollinstanz

Tabelle 64 : Serviceprimitive zwischen AAL-Benutzer und AAL-1-Layer

a. Mandatory (erforderlich)

b. Optional (optional)

Parameter	Bemerkung
DATA	Die zu sendende oder empfangene AAL-SDU. Die Größe hängt von dem von der AAL-Entity zur Verfügung gestellten Diensten ab, sie kann entweder 1 Bit bzw. 1 Oktett betragen; der zeitliche Abstand zwischen den Daten ist konstant.
STRUCTURE	Dieser Parameter wird gesetzt, wenn die Anwenderinformation in Bitgruppen organisiert ist. Die Länge der Blöcke ist für jede AAL-Entity festgelegt, und beträgt ein Vielfaches von 8 Bit. Die beiden möglichen Werte des STRUCTURE-Parameters sind <i>START</i> und <i>CONTINUATION</i> , je nachdem, ob die Daten den ersten Teil eines Blockes oder die Fortsetzung der Datenfolge innerhalb des Blocks darstellen. Die Verwendung dieses Parameters muß beim Verbindungsaufbau zwischen AAL-Benutzer und AAL vereinbart werden.
STATUS	Dieser Parameter mit den möglichen Werten <i>VALID</i> und <i>INVALID</i> gibt an, ob die übergebenen Daten fehlerhaft oder fehlerfrei sind. Die Verwendung dieses Parameters muß beim Verbindungsaufbau zwischen AAL-Benutzer und AAL vereinbart werden.

Tabelle 65 : Parameter der AAL1-Primitive

B.5 Service Primitive des AAL-Typs 1

Die Dienstprimitive zwischen AAL-1-Layer und dem ATM-Layer sind in Abschnitt B.3 auf Seite 188 aufgeführt. Die dort angegebenen Parameter sind wie folgt zu besetzen:

- Der Parameter *Submitted loss priority* ist im REQUEST-Primitiv mit „0“ (hohe Priorität) zu besetzen, sein Wert beim Empfang ist zu ignorieren.
- *ATM-user-to-ATM-user indication* ist auf „0“ zu setzen.
- Der Wert der *Congestion Indication* ist zu ignorieren.

Name	Parameter	Bedeutung
SAR-UNITDATA-INVOKE	Interface Data(M), Convergence Sublayer Indication CSI (M), Sequence Count (M)	Befehl zum Senden der CS-SDU zur Peer-Entity
SAR-UNITDATA-SIGNAL	Interface Data (M), Convergence Sublayer Indication CSI (M), Sequence Count (M), Check Status	Übergabe einer empfangenen SAR-SDU an die CS-Protokollinstanz

Tabelle 66 : Serviceprimitive zwischen SAR-Sublayer und CS-Sublayer

Parameter	Bemerkung
Interface Data	Die zwischen CS- und SAR-Entity transportierte Dateneinheit mit einer Länge von 47 Oktetts.
CSI	Vergl. Abbildung 45 auf Seite 65.
Sequence Count	Sequenznummer der CS-PDU, beginnend bei 0 werden diese modulo 8 numeriert.

Tabelle 67 : Parameter der SAR-Signale

B.6 Service Primitive des AAL-Typs 2

Name	Parameter	Bedeutung
CPS-UNITDATA-REQUEST	CPS-Interface Data (CPS-INFO) (M), CPS User-to-User Indication (CPS-UUI) (M) CPS Packet Payload Type CPS-PTT (M)	Befehl zum Senden der CPS-SDU zur Peer-Entity
CPS-UNITDATA-INDICATION	CPS-Interface Data (CPS-INFO) (M), CPS User-to-User Indication (CPS-UUI) (M) CPS packet Payload Type (CPS-PTT) (M)	Übergabe einer empfangenen CPS-SDU an die SSCS-Protokollinstanz
M	Mandatory (Zwingend vorhanden)	

Tabelle 68 : Serviceprimitive zwischen SSCS und CPS

Name	Parameter	Bedeutung
MAAL-UNITDATA-REQUEST	CPS-Interface Data (CPS-INFO) (M), CPS User-to-User Indication (CPS-UUI) (M) CPS packet Payload Type CPS-PTT (M)	Befehl zum Senden der CPS-SDU zur Peer-Layermanagement-Entity
MAAL-UNITDATA-INDICATION	CPS-Interface Data (CPS-INFO) (M), CPS User-to-User Indication (CPS-UUI) (M) CPS Packet Payload Type CPS-PTT (M)	Abgabe einer empfangenen CPS-SDU an die Layermanagement-Entity
MAAL-SEND-REQUEST	-	Befehl zum Senden einer User-CPS-PDU
M	Mandatory (Zwingend vorhanden)	

Tabelle 69 : Serviceprimitive zwischen Layermanagement und CPS

B.7 Service Primitive des AAL-Typs 3/4

Parameter	Typ	MM	SM	Bemerkung
Interface-Data (ID)	request	M	M	Vollständige oder teilweise SDU
	indication	M	M	
More (M)	request	-	M	0: Ende der SDU
	indication	-	M	1: Nicht das Ende einer SDU
Maximum Length (ML)	request	-	M*	Maximale Länge der SDU
	indication	-	O*	
Reception Status (RS)	request	-	-	Anzeige fehlerhafter Daten
	indication	O	O	
MM	Message Mode Service			
SM	Streaming Mode Service			
M	Zwingend vorhanden			
O	Optional			
-	Nicht vorhanden			
M*	Im ersten Primitiv zwingend, danach nicht vorhanden			
O*	Im ersten Primitiv optional , danach nicht vorhanden			

Tabelle 70 : Parameter der CPCS-Primitive

Parameter	Bemerkung
Interface-Data (ID)	Zwischen CPCS und SAR-Entity ausgetauschten IDU mit einer Länge von Vielfachen eines Oktetts
More (M)	Markiert den letzten Teil einer SAR-SDU
Reception Status (RS)	Anzeige fehlerhafter Daten

Tabelle 71 : Parameter der SAR-Primitive

B.8 Service Primitive des AAL-Typs 5

Parameter	Typ	MM	SM	Bemerkung
Interface Data (ID)	invoke signal	M M	M M	Vollständige oder teilweise CPCS-SDU
More (M)	invoke signal	- -	M M	0: Ende der SDU 1: Nicht das Ende einer SDU
CPCS Loss Priority	invoke signal	M M	M ₁ M ₂	Entspricht dem CLP-Feld der ATM-Zelle
CPCS Congestion Information	invoke signal	M M	M ₂ M ₂	Entspricht dem Congestion Information Parameter der ATM-Service Primitives
CPCS-User-to-User-Indication	invoke signal	M M	M ₂ M ₂	Wird transparent von der CPCS übertragen
Reception Status (RS)	invoke signal	- M	- M ₂	Anzeige fehlerhafter Daten
MM	Message Mode Service			
SM	Streaming Mode Service			
M	Zwingend vorhanden			
M ₁	Zwingend vorhanden im ersten Primitiv, das sich auf eine bestimmte CPCS-PDU bezieht, ansonsten nicht vorhanden			
M ₂	Zwingend vorhanden im letzten Primitiv, das sich auf eine bestimmte CPCS-PDU bezieht, ansonsten nicht vorhanden			
-	Nicht vorhanden			

Tabelle 72 : Parameter der CPCS-UNITDATA-Primitive

Parameter	Bemerkung
Interface Data (ID)	Zwischen CPCS und SAR-Entity ausgetauschten IDU mit einer Länge von Vielfachen von 48 Oktetts.
More (M)	Markiert den letzten Teil einer SAR-SDU
SAR Loss Priority	siehe CPCS Loss Priority
SAR Congestion Information	siehe CPCS Congestion Information

Tabelle 73 : Parameter der SAR-UNITDATA-Primitive

B.9 Service Primitive des AAL-Typs S

Name	Typ			
	Request	Indication	Response	Confirm.
AA-ESTABLISH	SSCOP-UU BR	SSCOP-UU	SSCOP-UU BR	SSCOP-UU
AA-RELEASE	SSCOP-UU	SSCOP-UU Source	nicht definiert	-
AA-DATA	MU	MU SN	nicht definiert	nicht definiert
AA-RESYNC	SSCOP-UU	SSCOP-UU	-	-
AA-RECOVER	nicht definiert	-	-	nicht definiert
AA-UNITDATA	MU	MU	nicht definiert	nicht definiert
AA-RETRIEVE	RN	MU	nicht definiert	nicht definiert
AA-RETRIEVE COMPLETE	nicht definiert	-	nicht definiert	nicht definiert
MAA-ERROR	nicht definiert	Code Count	nicht definiert	nicht definiert
MAA-UNITDATA	MU	MU	nicht definiert	nicht definiert
MU	Message Unit, Datenblock variabler Länge			
SSCOP-UU	SSCOP User-to-User Information, Datenblock variabler Länge zum Austausch zwischen SSCOP-Benutzern während der Verbindungssteuerung.			
SN	Sequence Number, Folgenummer eines Datenpaketes während der gesicherten Übertragung			
RN	Retrieval Number, Nummer der Daten, die aus dem Sendepuffer an den Benutzer zurückzugeben sind			
BR	Buffer Release, Anordnung zum leeren bestimmter Puffer in der SSCOP-Entity			
Code	Fehlercode (Q.2110, Annex A)			
Source	Verantwortlicher für den Verbindungsabbau („SSCOP“ oder „User“)			
Count	Nummer der wiederholten PDU			

Tabelle 74 : SSCOP Signale und Parameter

Anhang C Klassenübersichten

C.1 Basismodul

C.1.1 Zeitmanagement

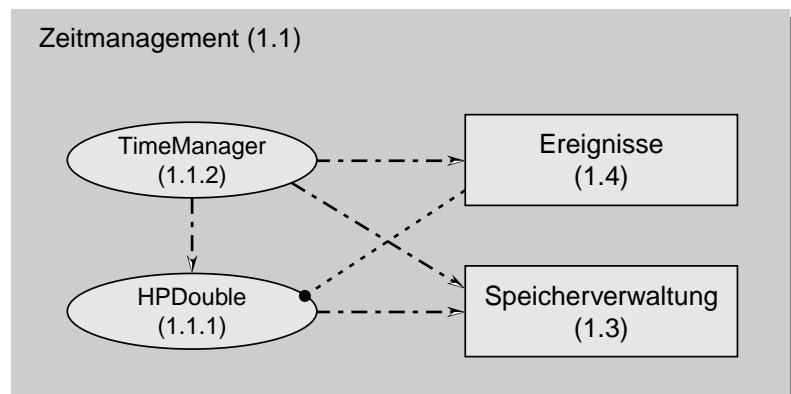


Abbildung 105 :Wichtige Klassen innerhalb des Zeitmanagements

HPDouble (1.1.1)

Anstelle des Basistyps *long double* wird jede Zahl durch eine vorzeichenlose Festkommazahl mit jeweils einem 64-Bit-Integer-Wert für die Vor- und Nachkommastellen dargestellt; alle Rechenoperationen arbeiten mit *Integer-Arithmetik* anstelle der rechenzeitintensiven *Fließkomma-Arithmetik*. Diese Zahlen weisen bei einem Wertebereich von 0 bis ($2^{64}-2^{-64}$) eine konstante Genauigkeit von 2^{-64} ($\approx 5,42 \cdot 10^{-20}$) auf. Aus Laufzeitgründen wird bei den Rechenoperationen vollständig auf die Prüfung von Bereichsüberschreitungen verzichtet.

TimeManager (1.1.2)

Die Ereignisverwaltung aus Abbildung 77 auf Seite 114, implementiert als binärer Baum, dessen Ordnungskriterium der Zeitstempel des Ereignisses ist. Die Suchoperationen im Baum zum Ordnen der eingetragenen Ereignisse erfolgen aus Laufzeitgründen nicht rekursiv. Zur Beschleunigung des Eintragsvorgangs, (üblicherweise eine Reihe von in geringen Abständen aufeinanderfolgenden Ereignissen) wird ein Hilfszeiger verwendet, der sich an der Position des zuletzt eingetragenen Wertes befindet, von der aus dann die Suche beim Eintrag des nächsten Ereignisses beginnt. Die Position des Wertes mit dem kleinsten Zeitstempel wird benutzt, um beim nächsten Austrag die Suche dort beginnen zu lassen. So wird das komplette Durchsuchen des Baumes bei jedem Ein- und Austragsvorgang vermieden. Die Einträge selbst sind nur Verweise (Pointer) auf das zu speichernde Ereignis, d.h. Pointer vom Typ *Event* (1.4.1). Die Ausgabereihenfolge bei Ereignissen mit gleichem Zeitstempel ist FIFO (First-In-First-Out).

C.1.2 Dateioperationen

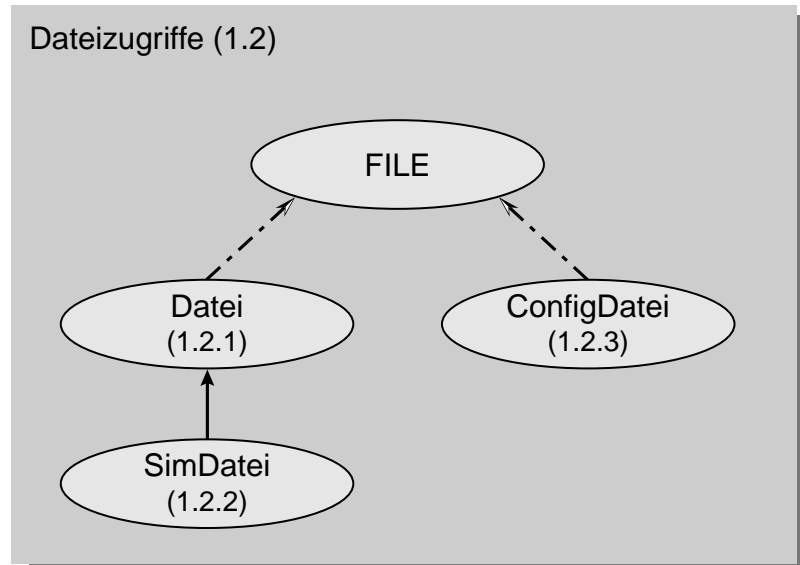


Abbildung 106 : Klassen zur Durchführung von Dateioperationen

Datei (1.2.1) Ein- und Ausgabeoperatoren für die in der Simulation verwendeten Datentypen werden bereitgestellt. Die Klasse wird verwendet zum Schreiben von Ablaufkontrolldateien (*log-Files*) und zur Speicherung von Zwischenergebnissen während der Simulationsläufe.

SimDatei (1.2.2) Eine Erweiterung der *Datei*-Klasse (1.2.1). Die Textdateien werden zusätzlich mit einem Header versehen, der Auskunft über die gespeicherten Daten gibt.

ConfigDatei (1.2.3) Objekte dieser Klasse lesen und schreiben speziell strukturierte Textdateien, die hauptsächlich zur Initialisierung von Variablen eingesetzt werden. Der Wert einer Variablen wird innerhalb der Datei durch zwei Kennwörter, einem Bereichskennwort und dem Name der Variablen markiert:

```

...
[Bereichskennwort]
...
Variablenname = Variablenwert
...

```

Ein Lesezugriff mit den Parametern *Bereichskennwort* und *Variablenname* liefert den *Variablenwert* an die aufrufende Funktion zurück. Um die Lesbarkeit der Dateien zu erhöhen sind Leerzeilen, Kommentarzeilen und Leerzeichen erlaubt. Spezielle Schlüsselwörter erhöhen den Funktionsumfang der *ConfigDatei*-Objekte; so ist beispielsweise das Nachladen von hierarchisch angeordneten Textdateien während der Anfragen möglich (vergl. [Kam96]).

C.1.3 Speicherverwaltung

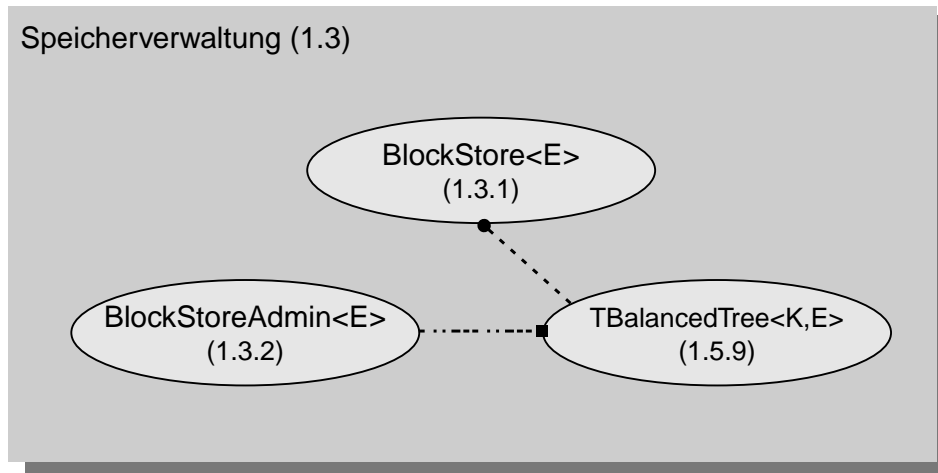


Abbildung 107 : Klassen zur Implementierung des Speichermanagements

Blockstore<E> (1.3.1) Klasse zur Verwaltung von Speicherbereichen, die von dynamisch erzeugten Objekten der Klasse E mit fester Größe genutzt wird. Einmal reservierte Speicherblöcke werden nicht an das Betriebssystem zurückgegeben, sondern in einer Liste zur späteren Wiederverwendung abgelegt.

BlockStoreAdmin<E> (1.3.2) Klasse zur Verwaltung von Speicherbereichen, die von dynamisch erzeugten Objekten der Klasse E mit Elementen variabler Größe genutzt wird. Objekte dieser Klasse bestehen aus dynamisch erzeugten *Blockstore-Objekten*, die gleichgroße Blöcke verwalten. Die Auswahl der geeigneten Speicherverwaltung geschieht über einen *Balancierten Baum* (1.5.9), der alle *Blockstore-Objekte* verwaltet und als Ordnungskriterium die Größe der verwalteten Speicherbereiche verwendet. Um die Zersplitterung des Speichers in viele nahezu gleich große Blöcke zu vermeiden, werden Speicherbereiche nur in ganzzahligen Vielfachen von 64 Byte zur Verfügung gestellt.

C.1.4 Ereignisse

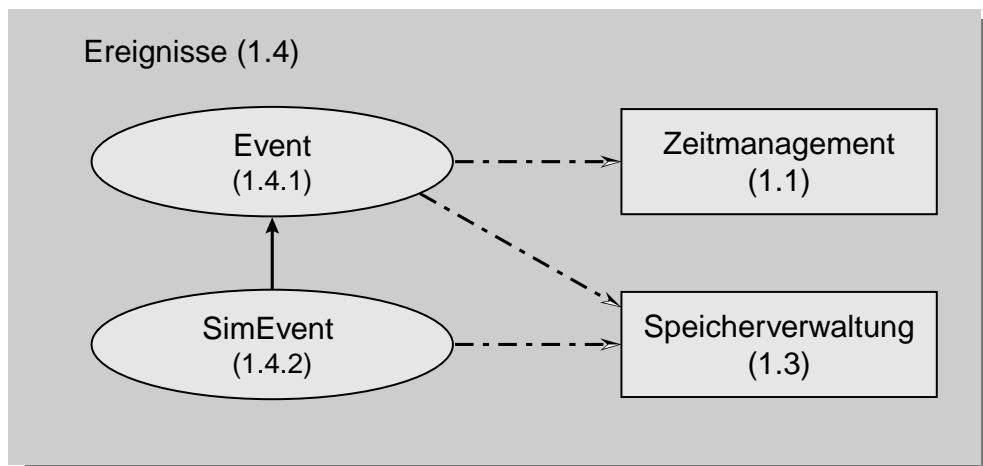


Abbildung 108 :Basisklassen zur Ereignisbehandlung

- Event (1.4.1)** Basisklasse aller Ereignisse in der Simulation. Als Format des Zeitstempels wird der Datentyp *HPDouble (1.1.1)* vorgegeben.
- SimEvent (1.4.2)** Die erste Erweiterung der *Event*-Klasse. Objekte der Klasse *SimEvent* repräsentieren Ereignisse innerhalb einer Simulation eines Nachrichtennetzwerkes auf der Grundlage des OSI-Modells. Spezifiziert wird die Protokollinstanz (vergl. Abschnitt C.4.1 auf Seite 214), welche das Ereignis ausgelöst hat, sowie die Protokollinstanz zu der das Ereignis gelangen soll. Zu diesem Zweck werden Netzwerkstation, eine eindeutige Zahl zur Kennzeichnung des anzusprechenden Protokolls sowie zur Realisierung des Protokolls angegeben. Das Ereignis selbst wird durch einen *Aufzählungs*-Typ beschrieben; in den überwiegenden Fällen handelt es sich hierbei um die Bezeichnungen der Service-Primitives der entsprechenden Protokollinstanzen. Eine weitere Funktion der Klasse ist die Bereitstellung der *virtuellen Funktionen* für die von ihr abgeleiteten Ereignisklassen.

C.1.5 Speicherstrukturen

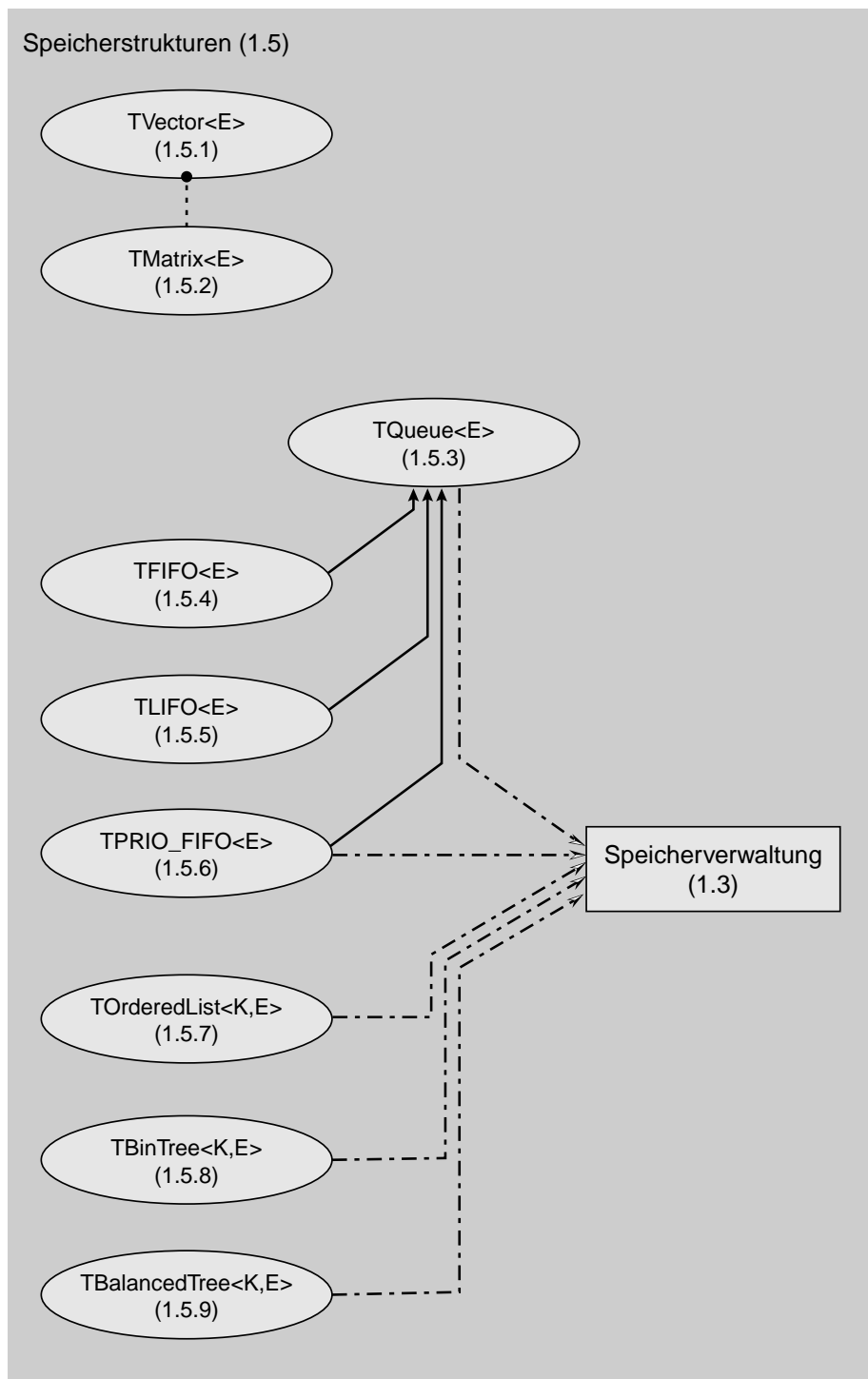


Abbildung 109 : Verschiedene Strukturen zur Speicherung temporärer Daten

Alle Klassen dieser Klassengruppe sind als Containerklassen implementiert, d.h. die von ihnen verwalteten Daten können Objekte beliebiger Klassen sein.

- TVector<E>**
(1.5.1) Ein Vektor aus Elementen gleichen Typs. Alle Operationen wie Addition, Subtraktion, Betragsbildung, Skalar- und Kreuzprodukt für beliebige Elemente sind definiert. Voraussetzung ist die Definition der Operatoren „+“, „-“, „*“ und „/“ auf der Elementklasse E.
- TMatrix<E>**
(1.5.2) Eine Matrix aus Elementen gleichen Typs. Alle Operationen wie die Addition, Subtraktion, und Multiplikation von Matrizen, sowie Operationen zwischen Matrizen und Vektoren sind definiert. Voraussetzung ist wie bei der Vektorklasse *TVector<E>* die Definition der elementaren Operationen auf der Elementklasse E.
- TQueue<E>**
(1.5.3) Die Basisklasse aller sequentiellen Listen. Sie stellt den abgeleiteten Klassen die Grundlagen für das Speichern beliebiger Elemente in Warteschlangen begrenzter Länge zur Verfügung.
- TFIFO<E>**
(1.5.4) Warteschlangen mit First-In-First-Out-Speicherstrategie. Beliebige Elemente können eingetragen und in der Reihenfolge ihres Eintreffens wieder ausgelesen werden.
- TLIFO<E>**
(1.5.5) Warteschlangen mit Last-In-First-Out-Speicherstrategie. Beliebige Elemente können eingetragen und in umgekehrter Eintragsreihenfolge wieder ausgelesen werden.
- TPRIO_FIFO<E>**
(1.5.6) Warteschlangen, deren Einträge über bis zu 256 unterschiedliche Prioritäten verfügen. Trifft ein Eintrag mit hoher Priorität auf einen vollbesetzten Puffer, wird zunächst versucht, einen Eintrag mit der niedrigsten vorhandenen Priorität aus der Warteschlange zu entfernen; erst in dem Fall, daß keine Einträge niedriger Priorität vorhanden sind, geht die Forderung zu Verlust.
- TBinTree<K,E>**
(1.5.7) Ein binärer Baum, in dem beliebige Einträge geordnet nach frei definierbaren Schlüsseln abgespeichert werden können. Voraussetzung ist die Definition der Vergleichsoperatoren „==“ und „<=“ auf der Elementklasse. Elemente mit gleichen Schlüsseln werden in der Reihenfolge ihres Eintrags ausgegeben.
- TBalancedTree<K,E>**
(1.5.8) Ein binärer Baum, der dem Effekt der Entartung des Baumes zu einer sequentiellen Liste durch Umordnung des Baumes (*Rotation*, vergl. [Sed94], Kap.15) entgegenwirkt, und so die damit verbundene höhere Anzahl von Vergleichen beim Suchen eines Eintrages vermeidet. Ein solcher Baum weist immer die maximale optimale Suchtiefe von $\log n$ (n : Anzahl der Baumeinträge) auf.
- TOrderedList<K,E>** Eine nach aufsteigenden Werten des Schlüsseldatentypes K geordnete Liste von Werten. Eine solche Liste ermöglicht das schnelle Auffinden aufeinanderfolgender Elemente.

C.2 Statistikmodul

C.2.1 Zufallszahlengeneratoren

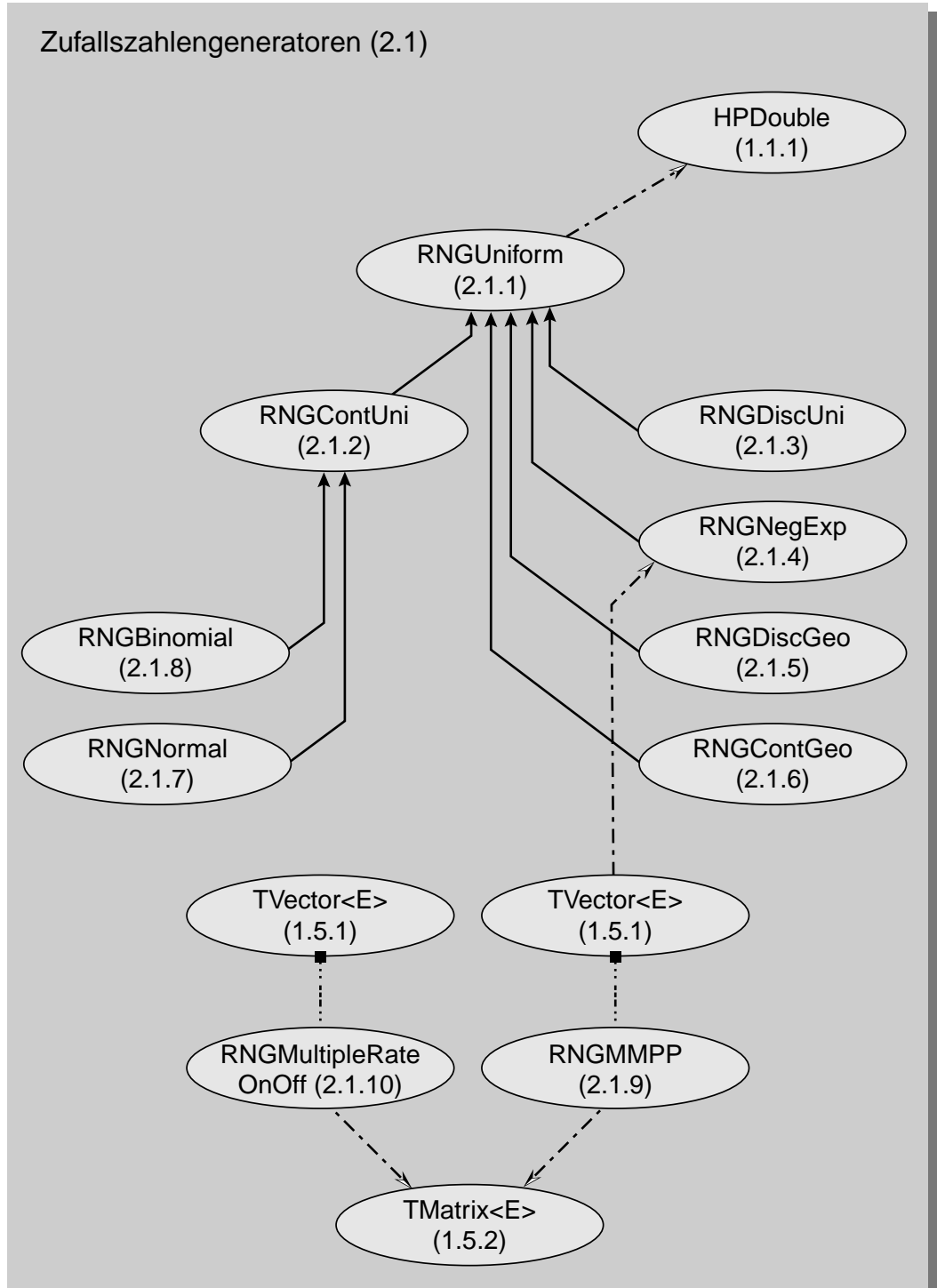


Abbildung 110 : Klassen zur Erzeugung von Zufallsvariablen mit unterschiedlichen Verteilungsfunktionen

RNGUniform (2.1.1)	Die Basisklasse aller Zufallszahlengeneratoren zur Erzeugung gleichverteilter Zufallszahlen im Intervall $[0,1]$. Reduktion der Rechenzeit erfolgt durch den Ersatz der Multiplikations- und Divisionsoperationen aus Gl.19 auf Seite 25 durch äquivalente Pointer- und Bitschiebeoperationen (Ableitung siehe [Her96], S.43 ff).
RNGContUni (2.1.2)	Erzeugung (quasi-)kontinuierlicher, gleichverteilter Zufallszahlen innerhalb eines beliebigen Intervalls $[a,b]$.
RNGDiscUni (2.1.3)	Erzeugung diskreter, gleichverteilter Zufallszahlen innerhalb eines beliebigen Intervalls $[a,b]$.
RNGNegExp (2.1.4)	Erzeugung (quasi-)kontinuierlicher, negativ-exponentiell verteilter Zufallszahlen.
RNGDiscGeo (2.1.5)	Erzeugung diskreter, geometrisch verteilter Zufallszahlen.
RNGContGeo (2.1.6)	Erzeugung (quasi-)kontinuierlicher, geometrisch verteilter Zufallszahlen in einem System mit Zeitraster.
RNGNormal (2.1.7)	Erzeugung (quasi-)kontinuierlicher, normalverteilter Zufallszahlen.
RNGBinomial (2.1.8)	Erzeugung diskreter, gleichverteilter Zufallszahlen, die einer Binomialverteilung unterliegen.
RNGMMPP (2.1.9)	Erzeugung von Zufallsvariablen durch einem MMPP ^a .
RNGMultiple- RateOnOff (2.1.10)	Dieser Generator unterscheidet sich vom RNGMPP (2.1.9) durch die Verwendung determiniert sendender Quellen unterschiedlicher Rate in den verschiedenen Zuständen.

a. Markov Modulated Poisson Process Abschnitt 2.6.1 auf Seite 27

C.2.2 Analyse von Zufallsvariablen

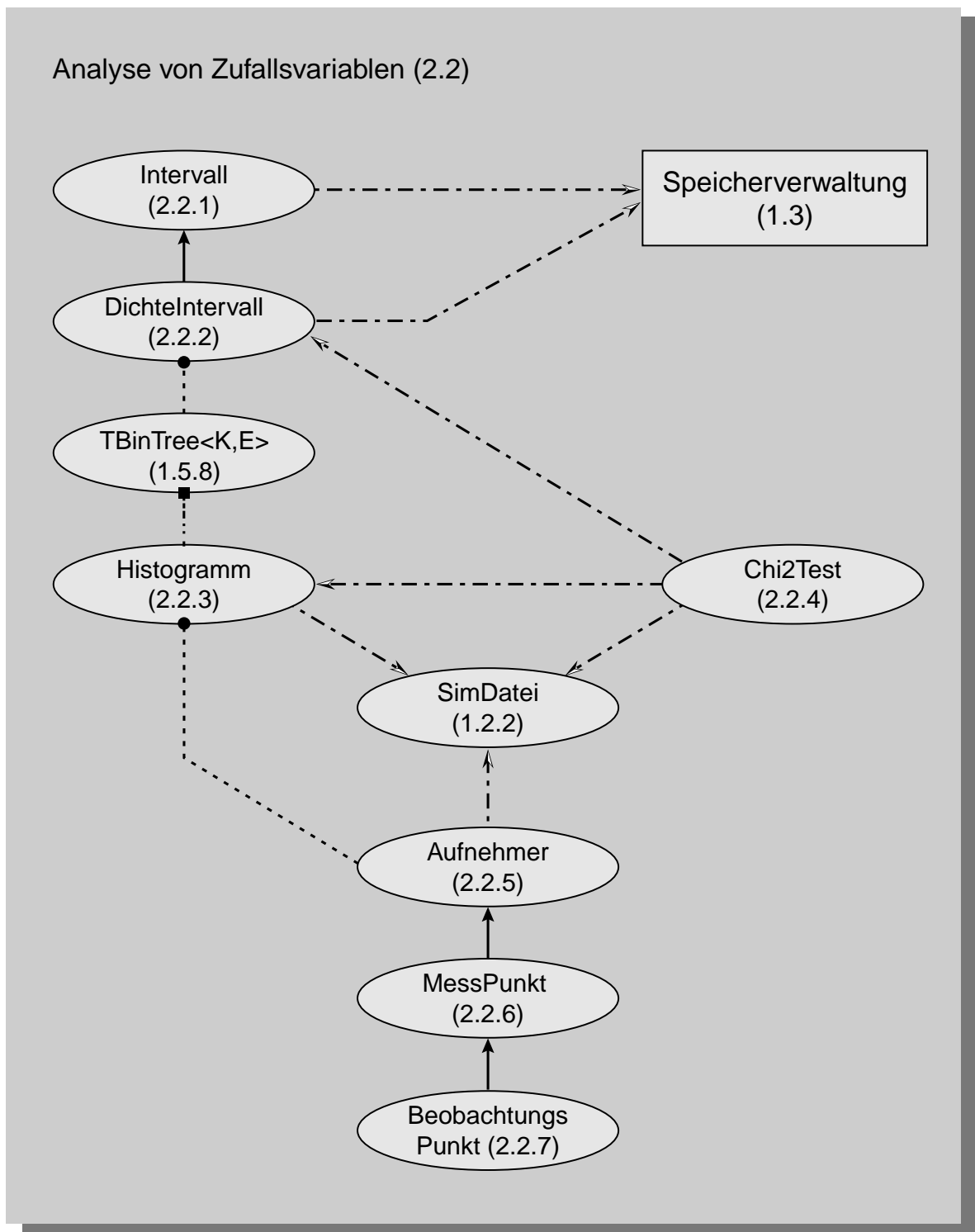


Abbildung 111 : Klassen zur Analyse von Zufallsvariablen

Intervall (2.2.1)	Die Klasse stellt ein mathematisches Intervall [a,b] dar. Operatoren zur Beurteilung, ob ein Wert innerhalb oder außerhalb des Intervalls liegt, werden bereitgestellt.
DichteIntervall (2.2.2)	Ein <i>Intervall</i> mit einer Speichermöglichkeit für eine zusätzlichen Zahl.
Histogramm (2.2.3)	Verteilungsdichtefunktion einer Zufallsgröße. Als Datenstruktur wird ein binärer Baum mit dem Ordnungskriterium <i>DichteIntervall</i> verwendet. Die Baumstruktur ermöglicht die beliebig feine Unterteilung des Beobachtungsbereiches und reduziert die Suchtiefe beim Einordnen eines neuen Wertes. Alle Momente und Zentralmomente aus Kapitel 2.1.3 auf Seite 16 können berechnet werden.
Chi2Test (2.2.4)	Diese Klasse führt χ^2 -Tests (vgl. <\$elemtag<\$elemtext<\$elemtag<\$elemtext<\$elemtag<\$elemtextAbschnitt 2.5 auf Seite 25) an Zufallsvariablen durch.
Aufnehmer (2.2.5)	Basisklasse aller „Meßeinrichtungen“ innerhalb der Simulation. Die Anlage der Ergebnisdateien; Öffnen und Schließen der Datenfiles wird von den Objekten dieser Klasse vorgenommen.
MessPunkt (2.2.6)	Eine Klasse, die Zufallsvariablen entgegennimmt und die Schnittstelle zur Abfrage von statistischen Eigenschaften der Zufallsgrößen bildet.
Beobachtungspunkt (2.2.7)	Objekte dieser Klasse verhindern die Analyse von Zufallsvariablen vor dem Eintreten des eingeschwungenen statistischen Zustandes. Indem sie die Weitergabe der Werte an ihre Basisklasse <i>MessPunkt</i> verhindern.

C.3 Netzwerkmodul

C.3.1 Netzwerkverbindungstypen

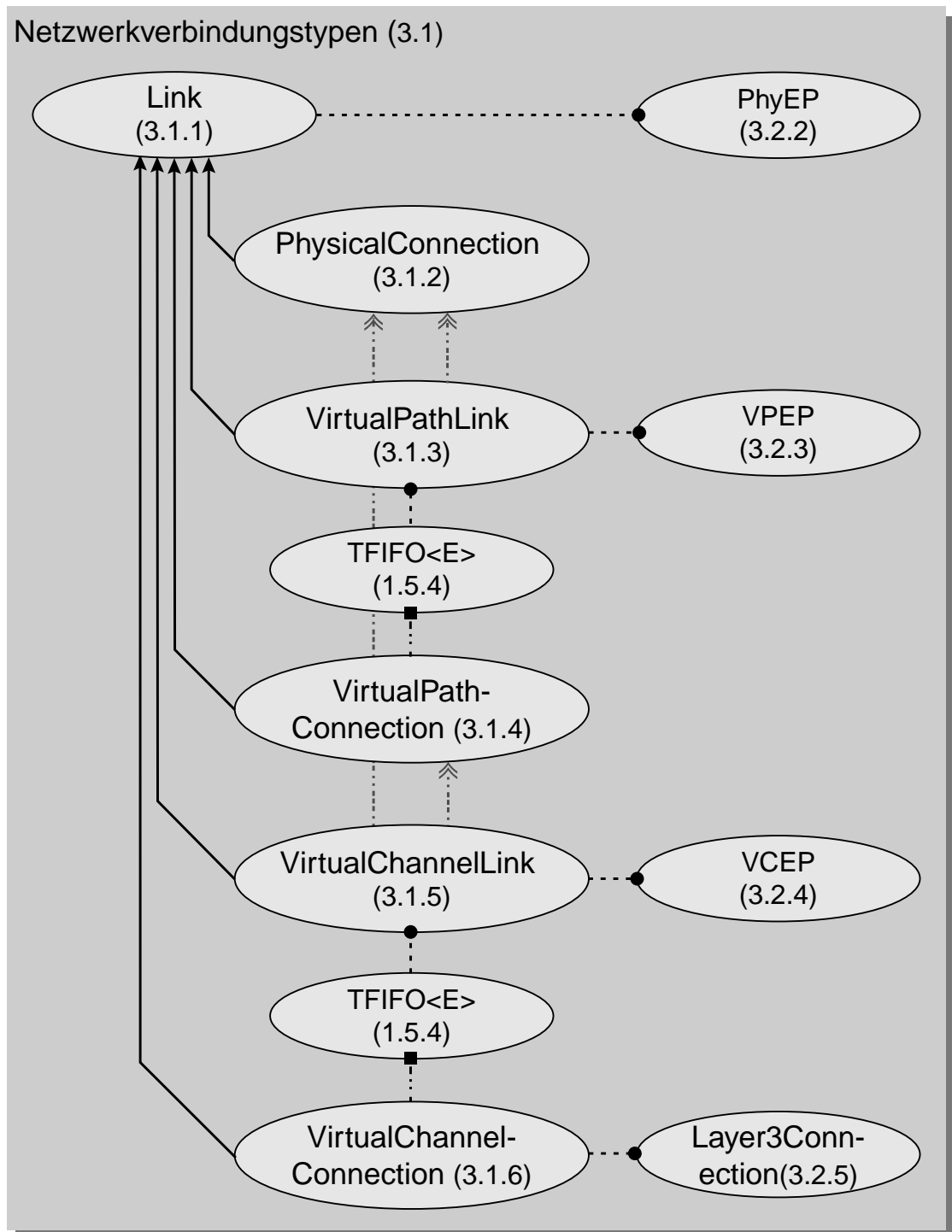


Abbildung 112 :Verbindungstypen im B-ISDN

- Link**
(3.1.1) Die Basisklasse aller Verbindungstypen auf den unterschiedlichen Ebenen. Jeder Link wird durch die Angabe der Station (3.5.1) und des physikalischen Ports am Anfang und Ende beschrieben.
- Physical Connection**
(3.1.2) Objekte dieser Klasse repräsentieren Verbindungen im Physical Layer im B-ISDN-Netzwerk. Die Endpunkte der Verbindungen sind *Physical Endpoints* PhyEP, die von einer „Vermittlungsstelle“ auf der physikalischen Ebene (Repeater) umgesetzt werden.
- VirtualPathLink**
(3.1.3) Ein Segment einer Virtuellen Verbindung auf Pfadebene. Jedes Segment ist durch einen *Virtual Path Endpoint* VPEP (3.2.3) begrenzt, an dem das Segment durch eine Switch auf Pfadniveau vermittelt wird.
- VirtualPath Connection**
(3.1.4) Eine virtuelle Verbindung auf Pfadebene, die mehrere virtuelle Kanäle umfaßt. Ihre Segmente (3.1.3) verwaltet sie in einer geordneten Liste (einem Objekt der Klasse TFIFO<VirtualPathLink>).
- VirtualChannelLink**
(3.1.5) Ein Segment einer Virtuellen Verbindung auf Kanalebene. Jedes Segment ist durch einen *Virtual Channel Endpoint* VCEP (3.2.4) begrenzt, an dem das Segment durch eine Switch auf Kanalniveau vermittelt wird.
- VirtualChannelConnection**
(3.1.6) Eine virtuelle Verbindung auf Kanalebene zwischen zwei Endteilnehmern im B-ISDN. Die Verwaltung der Segmente erfolgt analog zu (3.1.4).

C.3.2 Verbindungsendpunkte und Vermittlungsfunktionalität

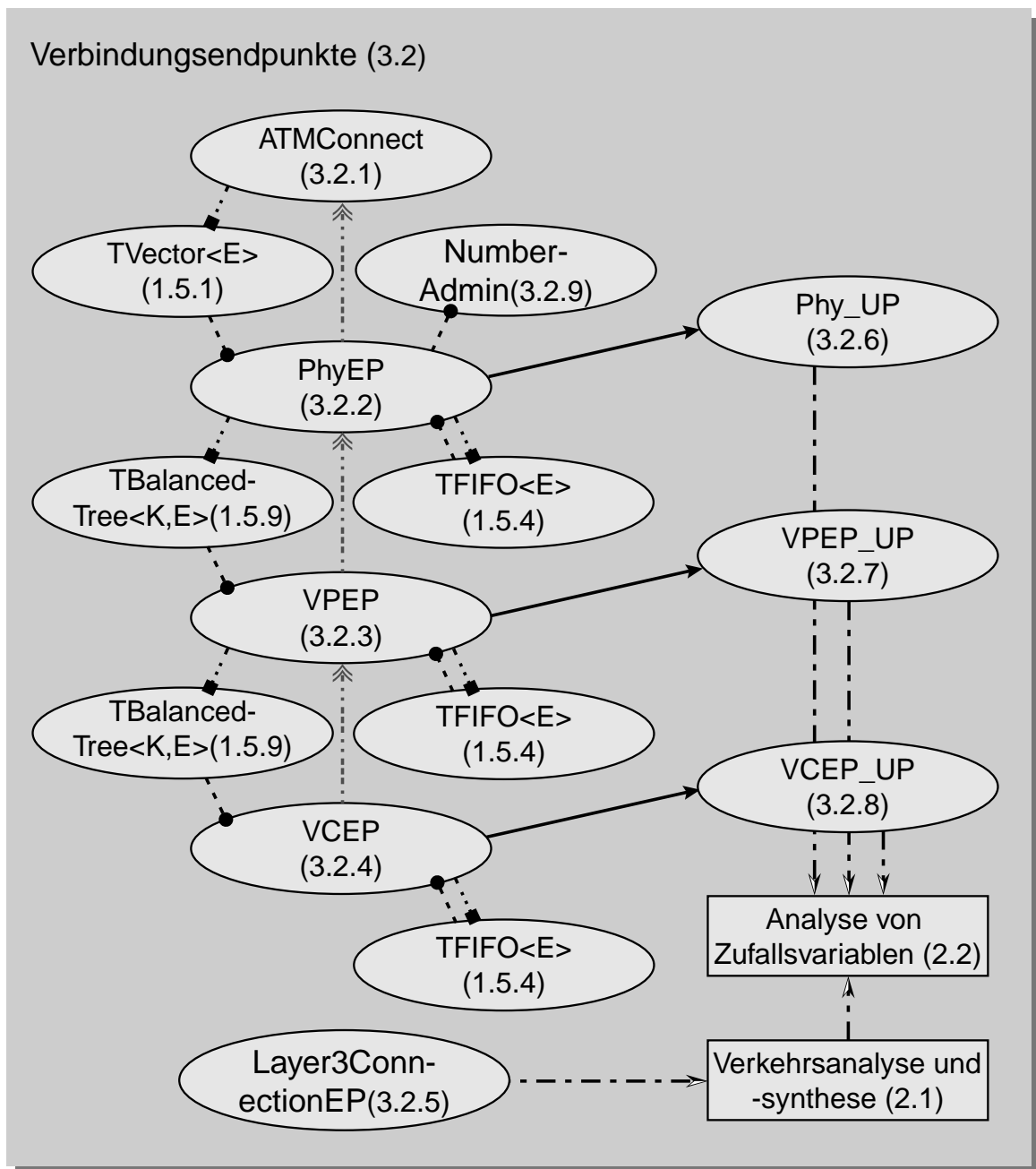


Abbildung 113 :Verbindungsendpunkte auf Verbindungs-, Pfad- und Kanalebene

ATMConnect (3.2.1) Schnittstellenklasse zu allen Vermittlungsfunktionen zwischen Verbindungsendpunkten an der betreffenden Station.

PhyEP (3.2.2)	Endpunkt einer physikalischen Verbindung. Er entspricht der Anschlußmöglichkeit für ein physikalisches Übertragungsmedium. Jeder Endpunkt ist unidirektional, d.h. er existiert für Sende- und Empfangsrichtung getrennt in Übereinstimmung mit den Vorgaben der ITU Rec. I.432 [I432]. Die „Vermittlungsfunktion“ beschränkt sich auf das Weiterleiten von Zellen oder SDH-Containern zu den in der Liste aufgeführten physikalischen Ausgangsports (<i>Repeater</i>).
VPEP (3.2.3)	Endpunkt einer Verbindung auf der Ebene Virtueller Pfade. Handelt es sich bei der Station um eine Vermittlungsstelle mit Pfadvermittlungsfunktion (<i>VP-Level-Switch</i> oder <i>VC/VP-Level-Switch</i>), enthält das Objekt der Klasse TFIFO<VPEP> die Liste der Virtuellen Pfade auf der Ausgangsseite, zu denen alle Zellen weiterzuleiten sind. Die im Virtuellen Pfad enthaltenen Kanäle werden in einer Baumstruktur vom Typ TBalancedTree<VCI,VCEP> mithilfe der VCI-Werte der Kanäle verwaltet.
VCEP (3.2.4)	Endpunkt einer Verbindung auf der Ebene virtueller Kanäle. Handelt es sich bei der Station um eine Vermittlungsstelle auf Kanalebene (<i>VC-Level-Switch</i>), enthält das Objekt der Klasse TFIFO<VCEP> die Liste der Virtuellen Kanäle auf der Ausgangsseite zu denen alle Zellen weiterzuleiten sind.
Layer3 ConnectionEP (3.2.5)	Diese Klasse stellt einen Verbindungsendpunkt auf der Ebenen des AAL-Benutzers dar. Er beinhaltet auf der Senderseite Funktionen zur Generation von Datenströmen mit unterschiedlichen statistischen Eigenschaften, auf der Empfängerseite Klassen zur statistischen Analyse der Zwischenankunftsabstände sowie von Bitfehlern und Paketverlusten.
Phy_UP (3.2.6)	Klasse zur Verkehrsanalyse auf Port-Ebene. Die zeitlichen Abstände zwischen allen auf einem Port versendeten oder empfangenen Zellen werden in einem Histogramm (2.2.3) aufgezeichnet.
VPEP_UP (3.2.7)	Klasse zur Verkehrsanalyse auf der Ebene Virtueller Pfade. Die Zellen aller unterlagerten Virtuellen Kanäle werden gemeinsam erfaßt und ausgewertet.
VCEP_UP (3.2.8)	Klasse zur Verkehrsanalyse auf der Ebene Virtueller Kanäle; die Zellen eines Virtuellen Kanals werden erfaßt und ausgewertet.
NumberAdmin (3.2.9)	Eine Klasse zur Verwaltung von Verbindungsnummern. Sie garantiert die Eindeutigkeit der vergebenen VPI- und VCI-Werte auf jeder Verbindungsebene.

C.3.3 Statische Verbindungen

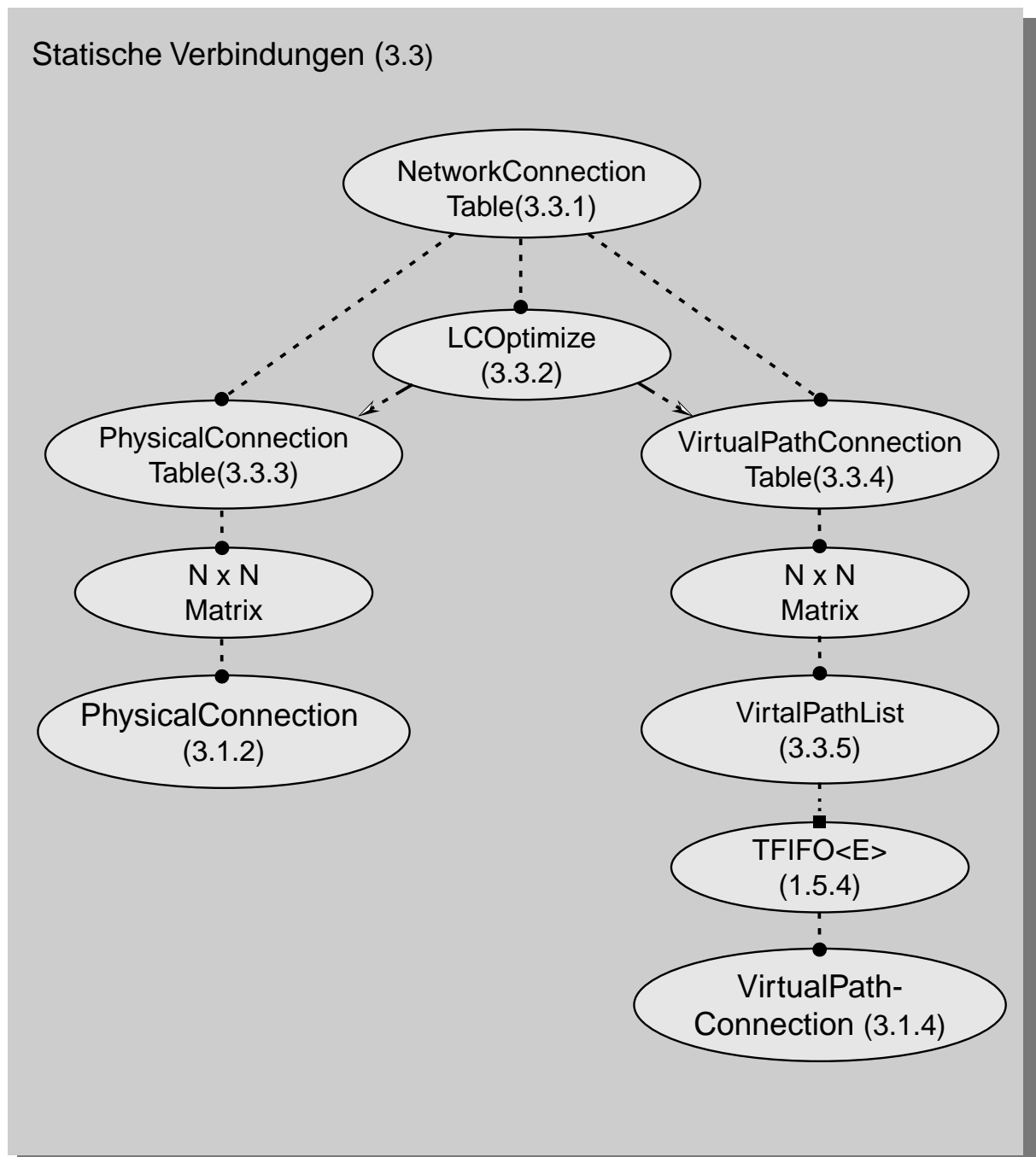


Abbildung 114 :Statische Verbindungsinformationen

- Network
ConnectionTable
(3.3.1)** Die Klasse stellt die Schnittstelle zur Simulation dar. Sie nutzt die Verbindungsdaten in (3.3.3) und (3.3.4) und den Wegsuchalgorithmus in (3.3.2) zum Aufbau einer *Virtual ChannelConnection* im Netzwerk. Auf dem durch (3.3.2) gelieferten Weg werden anhängig vom Verbindungstyp die VPI- und VCI-Werte auf den Segmenten (3.1) festgelegt sowie in allen Vermittlungsstellen die Einträge in die Verbindungslisten (3.2) vorgenommen. Beim Verbindungsabbau werden alle Stationen aufgefordert, die Verbindungsendpunkte und die zugeordneten Nummern wieder freizugeben.
- LCOptimize
(3.3.2)** Diese Klasse sucht den kürzesten Weg in einem Netzwerk unter Verwendung des iterativen *Label-Correcting-Verfahrens* aus der Graphentheorie ([Neu93], Abschnitt 2.4).
- Physical
ConnectionTable
(3.3.3)** Eine $N \times N$ Matrix, in der alle physikalischen Verbindungen (Leitungen) in einem Netzwerk aus N Stationen vermerkt sind.
- VirtualPath
ConnectionTable
(3.3.4)** Eine $N \times N$ Matrix, in der alle Verbindungen auf Pfadenebene in einem Netzwerk aus N Stationen vermerkt sind. In jedem Element der Matrix ist die Liste von Virtuellen Pfaden zwischen zwei Netzknoten abgelegt.
- VirtualPathList
(3.3.5)** Eine Liste Virtueller Pfade zwischen zwei Netzknoten, da im Gegensatz zu physikalischen Verbindungen zwischen zwei Stationen mehrere virtuelle Pfade über verschiedenen Wege im Netzwerk existieren können.

C.3.4 Dynamische Verbindungen

Im Gegensatz zur Verbindungen auf der Ebene physikalischer Verbindungen und virtueller Pfade werden Verbindungen auf Kanalebene erst auf Anforderung des Benutzers auf- und abgebaut (vergl. Signalisierung in Abschnitt 3.9 auf Seite 90).

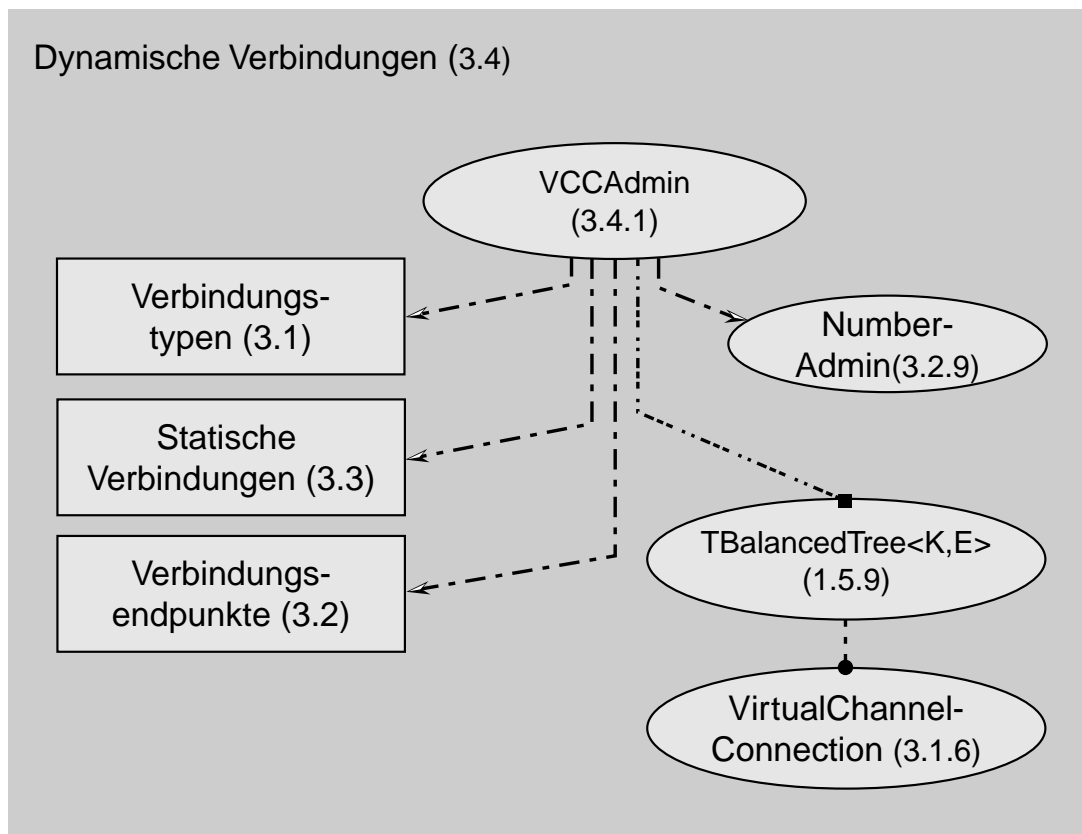


Abbildung 115 :Dynamische Verbindungen

VCCAdmin (3.4.1)

Diese Klasse verwaltet im *Network-Management-Center NMC* des B-ISDN alle bestehenden *Virtual Channel Connections (VCC)*. Trifft ein Verbindungsaufbauwunsch eines AAL-Benutzers über die Signalisierungs- bzw. Metasignalisierungskanäle am NMC ein, nutzt sie die Wegsuchalgorithmen aus (3.3) zum Aufbau der (unidirektionalen) VCCs für beide Kommunikationsrichtungen. Beim Verbindungsabbau werden die Kanäle in beide Richtungen abgebaut und die Verbindungsressourcen freigegeben.

C.3.5 Netzknoten

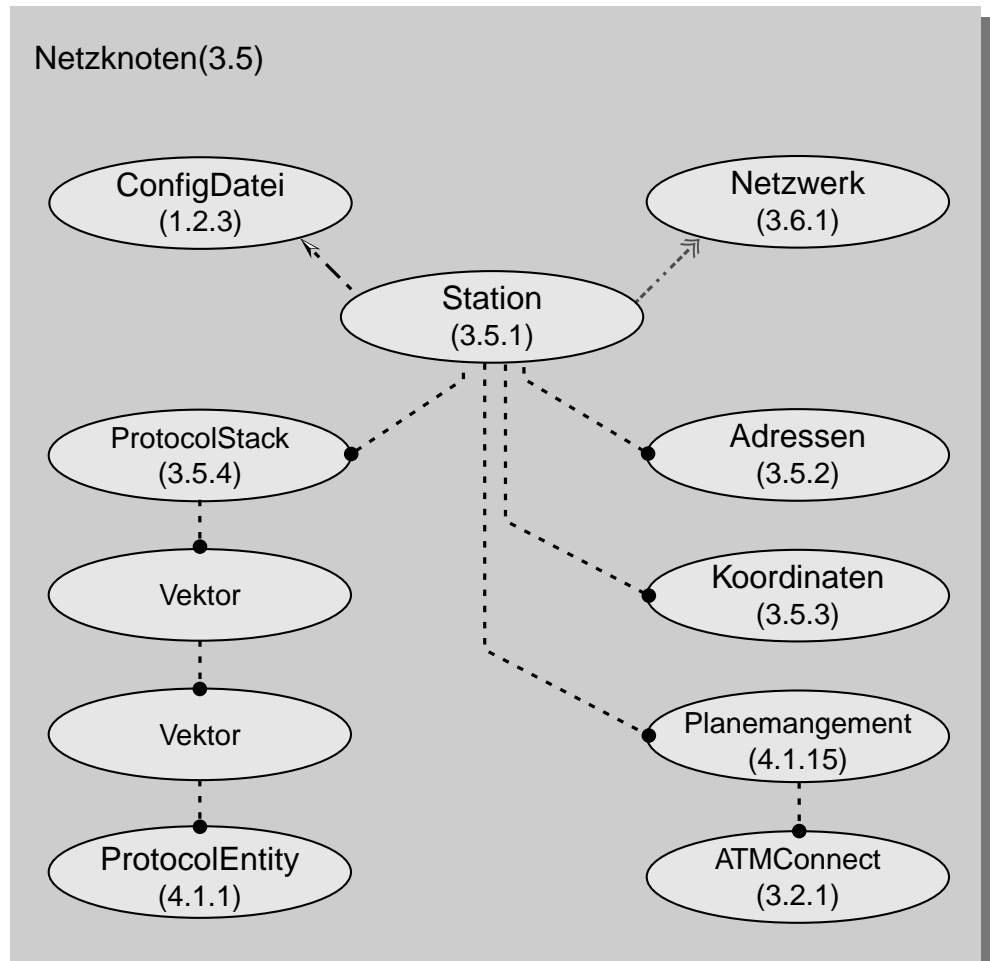


Abbildung 116 :Netzknoten im B-ISDN

**Station
(3.5.1)**

Diese Klasse beinhaltet alle relevanten Informationen über eine Netzknoten im B-ISDN. Abhängig vom Stationstyp werden die Protokollinstanzen und Vermittlungsfunktionen verwaltet. Die Informationen zur Konfiguration erhält das Stations-Objekt mithilfe der Klasse *ConfigDatei* (1.2.3) aus den Stationskonfigurationsdateien. Die Konfigurationsdatei enthält Angaben über alle an der Station vorhandenen Protokollinstanzen sowie die Pfade zu deren Konfigurationsdateien, so daß die Initialisierung aller Instanzen von ihr gesteuert werden kann.

**Adressen
(3.5.2)**

Alle Adressen der Station im Format E.164 [E164] oder ISO NSAP gem. X.213 [X213] zum Auffinden der Station innerhalb des Netzwerks (3.6)

- Koordinaten (3.5.3)** Der Ort, an dem sich die Station befindet, in Kugelkoordinaten, d.h. Längen- und Breitengradangaben des Standortes auf der Erdoberfläche.
- ProtocolStack (3.5.4)** Das Verzeichnis aller an der Station vertretenen Protokollinstanzen gem. Kapitel 3. Da alle Instanzen mehrfach auftreten können, ist eine Struktur aus verschachtelten Vektoren implementiert, wobei der erste Vektor die *Layer* beinhaltet, der zweite Vektor die Nummer der Instanz im entsprechenden *Layer*. Die Verwendung von Vektoren anstelle von Bäumen ermöglicht ein schnelles Auffinden der Protokollinstanzen während der Simulation.

C.3.6 Netzwerk

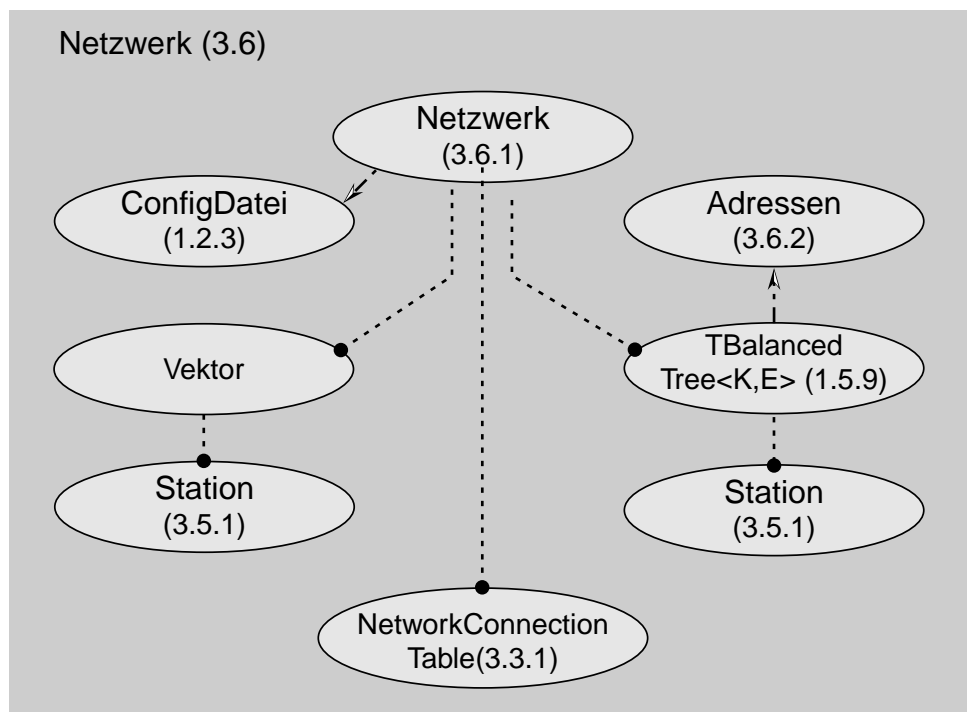


Abbildung 117 :Netzwerk-Klassen

- 3.6.1** Die Netzwerk-Klasse erfüllt stationsübergreifende Funktionen wie die Wegsuche oder die Bestimmung der Entfernung zwischen den Stationen zur Berechnung der Übertragungsdauern. Beim Start der Simulation sorgt sie für die korrekte Initialisierung aller Stationen und Verbindungstabellen (3.3), indem sie den Stationen ihre Konfigurationsdateien zur Verfügung stellt. Während des Simulationslaufes ist sie für das Auffinden der Stationen anhand von Verbindungsinformationen zuständig.

C.4 OSI-Modul

C.4.1 Protokollinstanzen

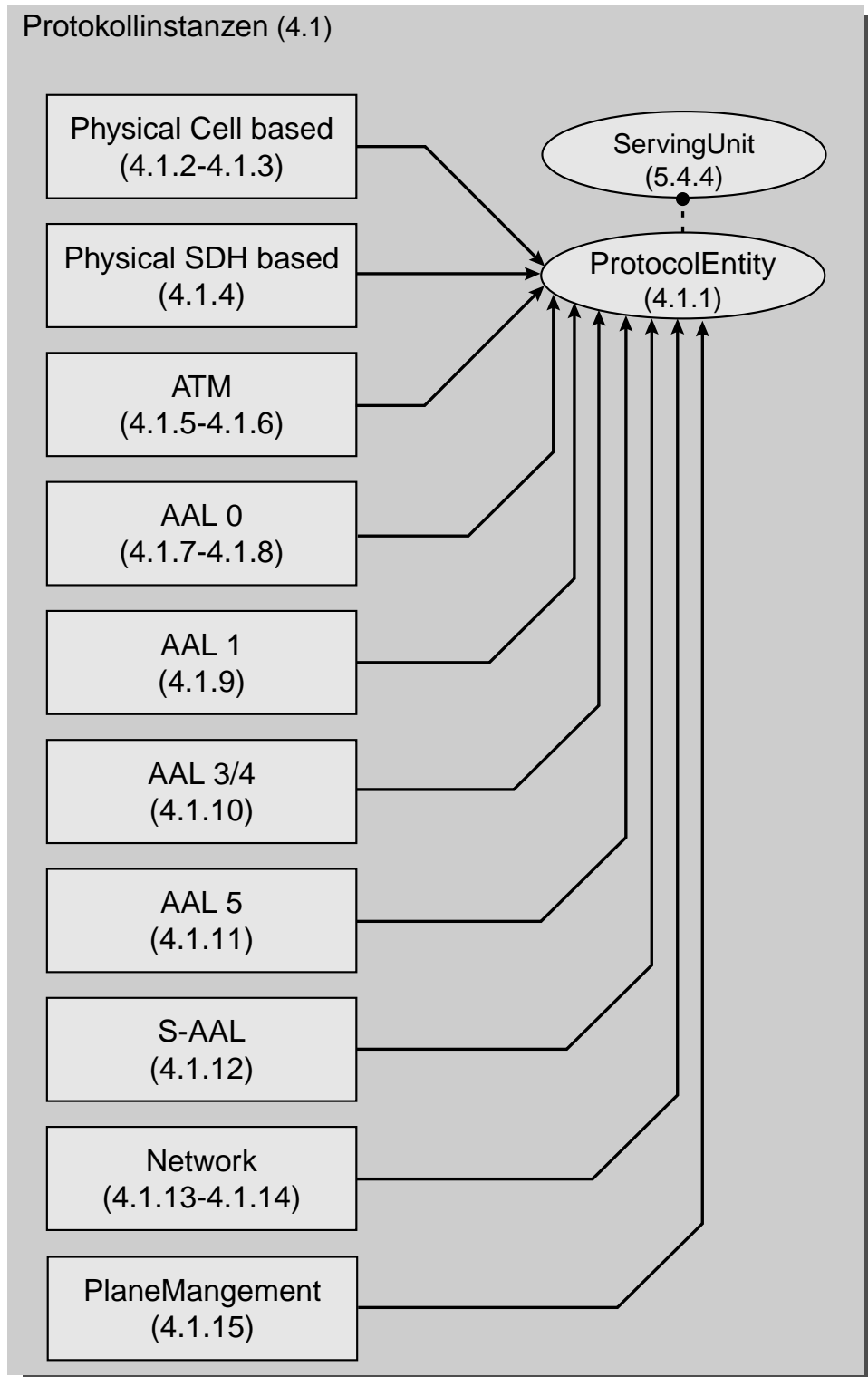


Abbildung 118 :Überblick über die Protokollinstanzen der Simulation

ProtocolEntity (4.1.1)	Die Basisklasse aller Protokollinstanzen. Sie entkoppelt die programmierten Protokollabläufe innerhalb der von ihr abgeleiteten Instanzen vom Hardwaremodell (5).
Physical Cell Based (4.1.2-4.1.3)	Objekte dieser Klassen wickeln das Protokoll im zellbasierten Physical Layer des B-ISDN ab (vergl. Seite 50 ff. sowie [Rau95]).
Physical SDH Based (4.1.4)	Das Protokoll im SDH-basierten Physical Layer des B-ISDN (vergl. Seite 51 ff. sowie [Wee97]).
ATM (4.1.5-4.1.6)	Das Protokoll im ATM Layer des B-ISDN (vergl. Abschnitt 3.7 auf Seite 56 ff. sowie [Rau95]).
AAL0 (4.1.7-4.1.8)	Das Protokoll des AAL-Typs 0 im B-ISDN (vergl. Abschnitt 3.8.1 auf Seite 64).
AAL1 (4.1.9)	Das Protokoll des AAL-Typs 1 im B-ISDN (vergl. Abschnitt 3.8.2 auf Seite 64 ff. sowie [Wur98]).
AAL3/4 (4.1.10)	Das Protokoll des AAL-Typs 3/4 im B-ISDN (vergl. Abschnitt 3.8.4 auf Seite 75 ff. sowie [Ngu97]).
AAL5 (4.1.11)	Das Protokoll des AAL-Typs 5 im B-ISDN (vergl. Abschnitt 3.8.5 auf Seite 81 ff. sowie [Hom96]).
S-AAL (4.1.12)	Das Protokoll der Signalisierungs-AAL im B-ISDN (vergl. Abschnitt 3.8.6 auf Seite 84 ff. sowie [Har98]).
Network (4.1.13-4.1.14)	Eine Nachbildung der AAL-Benutzer im Netzwerk. Diesen „Instanzen“ liegt kein spezielles Protokoll zugrunde, ihre Aufgabe besteht in der Anregung des Netzwerkes durch Quellen unterschiedlicher Charakteristiken beim Sender und der Entgegennahme der Daten nach der Übertragung über das B-ISDN beim Empfänger. Sie verwaltet alle Verbindungsendpunkte an einer Station und führt die simulierten Datenströme den entsprechenden Analysefunktionen zu.

Planmanagement
(4.1.15)

Das Planmanagement steuert den Verbindungsaufbau und -abbau an jeder Station. Es nimmt die Verbindungswünsche der Network-Instanz entgegen und sorgt für die Weitergabe an das NMC. Während des Verbindungsaufbaus vergibt sie die lokal gültigen Verbindungsnummern VPI und VCI für die Verbindungsendpunkte (3.2) und füllt die *Routing Table* einer Switch. Beim Abbau der Verbindung ist sie für die Freigabe der Verbindungsnummern, der Einträge in der *Routing Table* der Station sowie die Benachrichtigung aller Protokollinstanzen verantwortlich. Die Planmanagementinstanz des NMC führt die Wegsuche im Netz (3.3) sowie die Auswahl des zur Übertragung der Daten geeigneten AAL-Typs durch. Sie verwaltet alle im Netz bestehenden *Virtual Channel Connections VCC*.

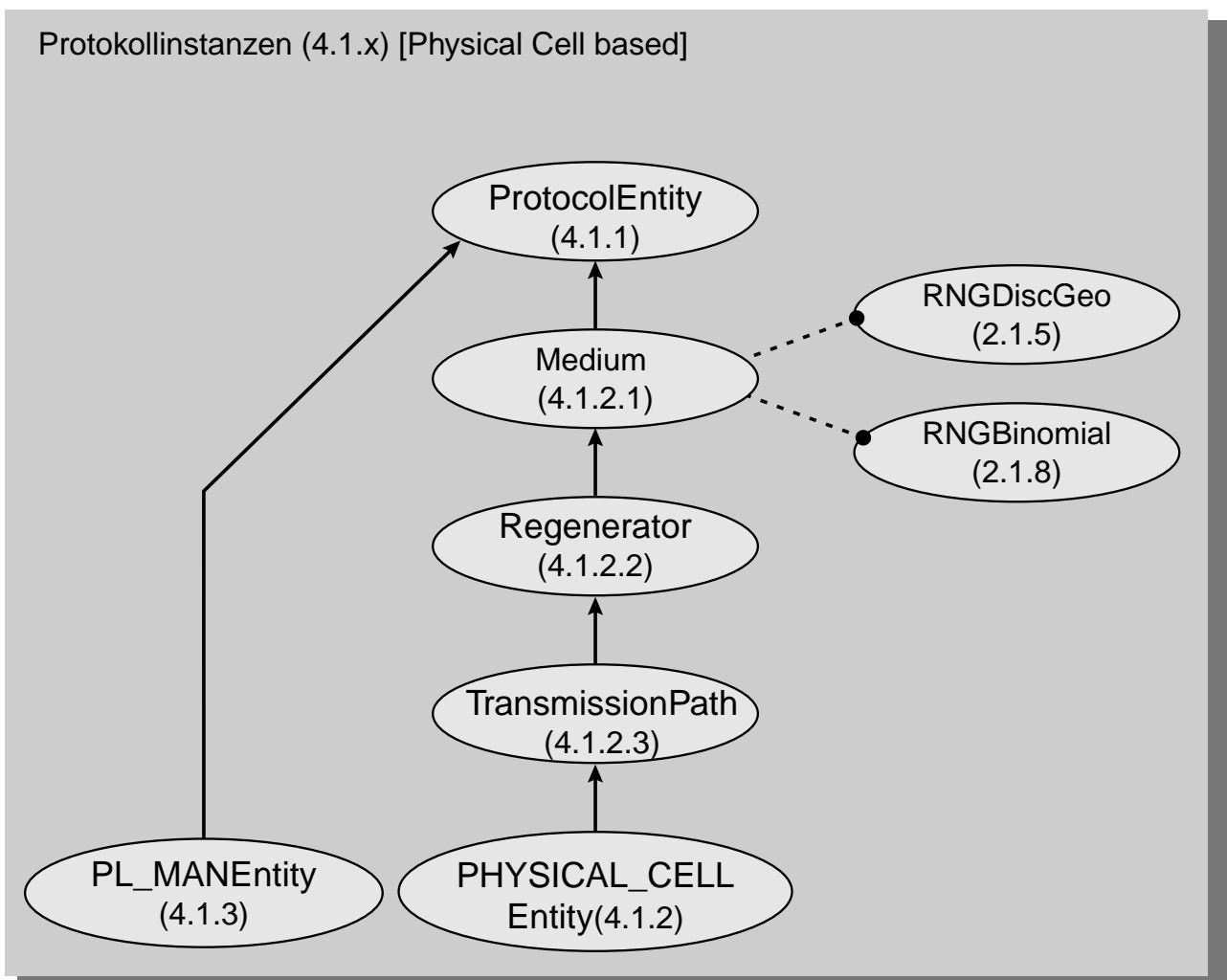
Protokollinstanzen des zellbasierten Physical Layers

Abbildung 119 : Interner Aufbau einer Instanz des zellbasierten Physical Layers

Physical_Cell Entity (4.1.2)	Die oberste Klasse in der Ableitungshierarchie. Sie nimmt alle Serviceprimitive für den Physical Layer entgegen und verteilt diese an die Sublayer (4.1.2.1- 4.1.2.3).
Medium (4.1.2.1)	Das Übertragungsmedium in der Simulation. Die Klasse versieht die Events mit der Angabe über die korrekte Empfangsstation und addiert die Übertragungsdauer zum Zeitstempel des Ereignisses. Mithilfe der in Abbildung 119 auf Seite 216 gezeigten Zufallszahlengeneratoren erzeugt sie Bitfehler in der zu übertragenden Zelle, was der Verfälschung der Daten während der Übertragung im realen Netz entspricht.
Regenerator (4.1.2.2)	Diese Klasse führt die F1-OAM-Funktionen auf dem Zellstrom aus. Idle-Zellen werden nicht übertragen, ihr Einfluß auf die Zählerstände und Prüfsummen (siehe Tabelle 52 auf Seite 178) wird im Sender und Empfänger jeweils beim Eintreffen einer OAM- oder Nutzdatenzelle nachträglich berechnet.
Transmission Path (4.1.2.3)	Diese Klasse ist verantwortlich für das synchrone Versenden der Zellen auf dem Übertragungsmedium entsprechend der definierten Übertragungsgeschwindigkeit (155,520 oder 622,080 MBit/s) und führt die F3-OAM-Funktionen auf dem Zellstrom aus.
PL_MANEntity (4.1.3)	Diese Klasse nimmt die Fehlerreports der Physical_CellEntity (4.1.2) entgegen.

Protokollinstanzen des SDH-basierten Physical Layers

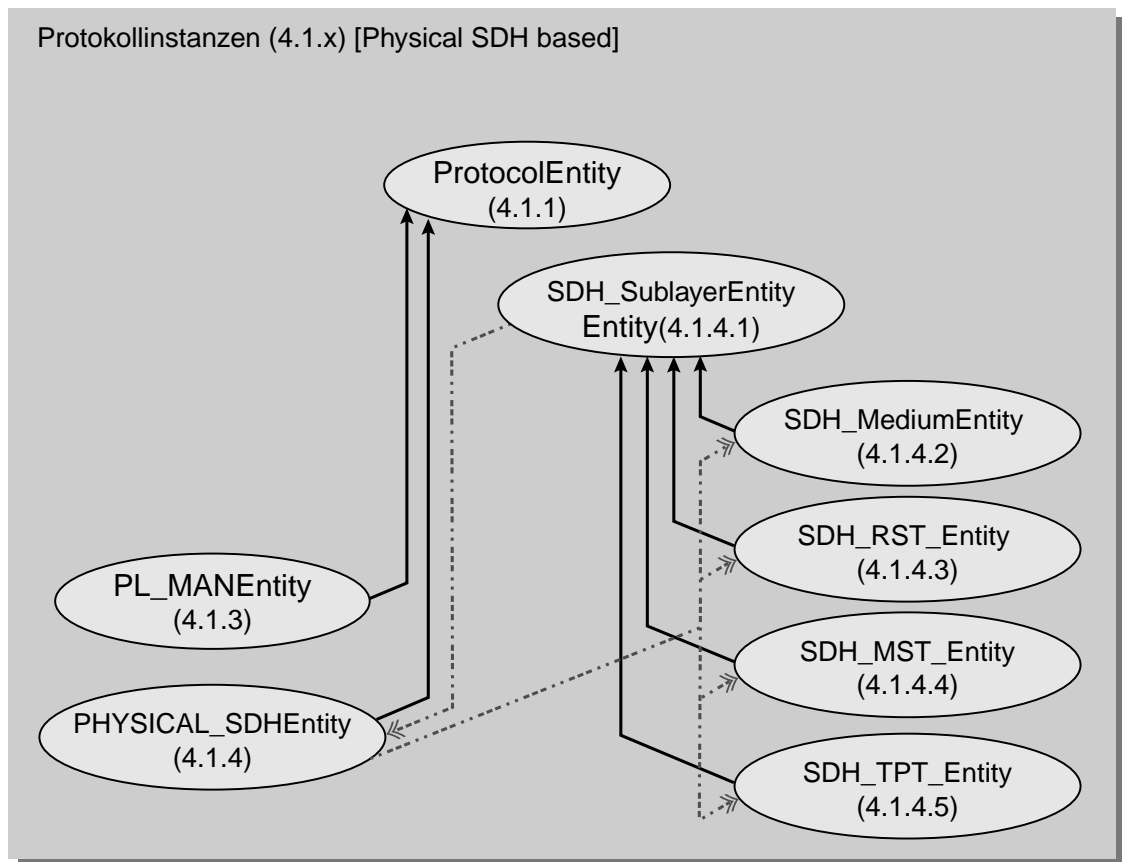


Abbildung 120 : Interner Aufbau einer Instanz des SDH-basierten Physical Layers

PHYSICAL_SDH Entity (4.1.4) Die Klasse stellt die gesamten Funktionen des SDH-basierten Physical Layers (vergl. Seite 51 ff) für die Simulation zur Verfügung. Sie ermöglicht die Simulation der Zellübertragung in einem SDH-Netzwerk bis zur Ebene STM-16 (vergl. Tabelle 20 auf Seite 52).

SDH_Sublayer Entity (4.1.4.1) Basisklasse der SDH-Sublayer (4.1.4.2 - 4.1.4.4).

SDH_Medium Entity (4.1.4.1) vergl. 4.1.2.1

SDH_RST_Entity (4.1.4.2) Instanz des *Regenerator Section Level* im SDH-Netz, Berechnung und Prüfung der Paritätsbits auf der Regenerator Section (F1-OAM).

SDH_MST_Entity (4.1.4.3) Instanz des *Digital Section Level* im SDH-Netz, Abwicklung der OAM-F2-Funktionen.

SDH_TPT_Entity (4.1.4.4) Instanz des *Transmission Path Level* im SDH-Netz. Die Zellen der ATM-Instanz werden in den entsprechenden Container (vergl. Abbildung 38 auf Seite 54 und Abbildung 39 auf Seite 54) „gemappt“ und die OAM-F3-Funktionen ausgeführt.

Protokollinstanzen des ATM Layers

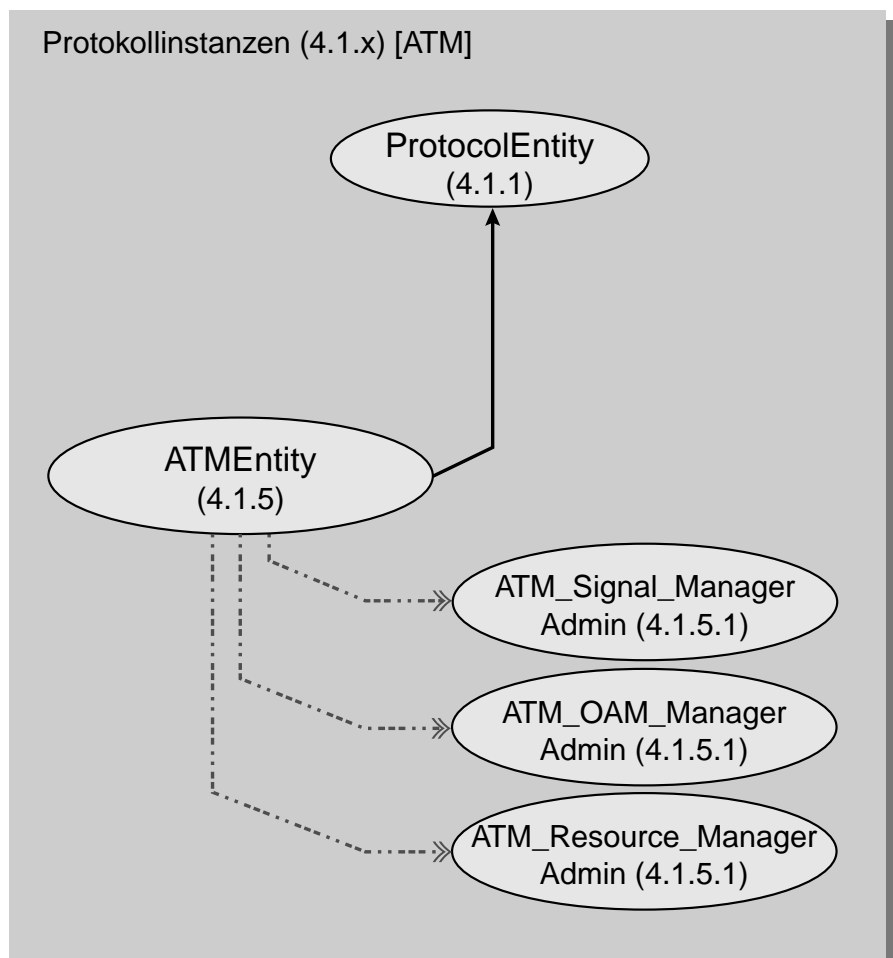


Abbildung 121 :Aufbau der Protokollinstanzen im ATM-Layer

ATMEntity (4.1.5) Instanz des ATM-Layers nach Abschnitt 3.7 auf Seite 56.

ATM_Signal_ManagerAdmin (4.1.5.1) Für jeden physikalischen Port der Station wird eine Instanz zur Abwicklung der Signalisierung im B-ISDN (3.9 auf Seite 90 ff) unterhalten. Da die Simulation nicht den Signalisierungsvorgang umfaßt, ist sie leer.

ATM_OAM_ Abwicklung der OAM-Funktionen der Level F4 und F5.
ManagerAdmin
(4.1.5.2)

ATM_Ressource_ Die Verwaltung der verfügbaren Ressourcen der Station. Gemäß betreiberspezi-
ManagerAdmin fischer Algorithmen wird die Entscheidung über Annahme oder Ablehnung der
(4.1.5.3) Verbindung getroffen (*Call Admission Control*).

Aufbau der Protokollinstanzen des AAL Layers (Typ 0)

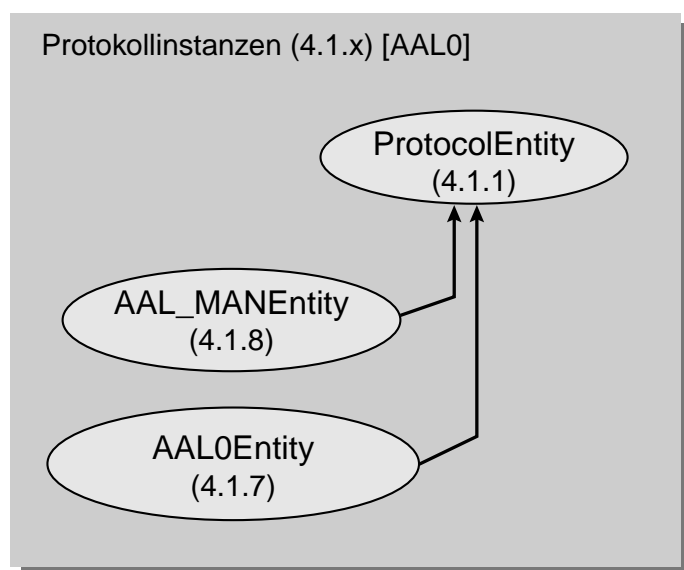


Abbildung 122 :Aufbau einer Protokollinstanz des AAL-Typs 0

AAL0Entity Die Instanzen des AAL-Typs 0 (vergl. 3.8.1 auf Seite 64) übertragen aus-
(4.1.7) schließlich Daten, die bereits in Zellform vorliegen. Es sind keinerlei Protokoll-
abläufe notwendig, nur eine Umsetzung der Service Primitive findet hier statt.

AAL_MANEntity Die Managementinstanz des AAL-Layers. Hier werden alle Fehler- und Status-
(4.1.8) meldungen der AAL-Instanzen aufgenommen und ausgewertet.

Protokollinstanzen des AAL Layers (Typ 1)

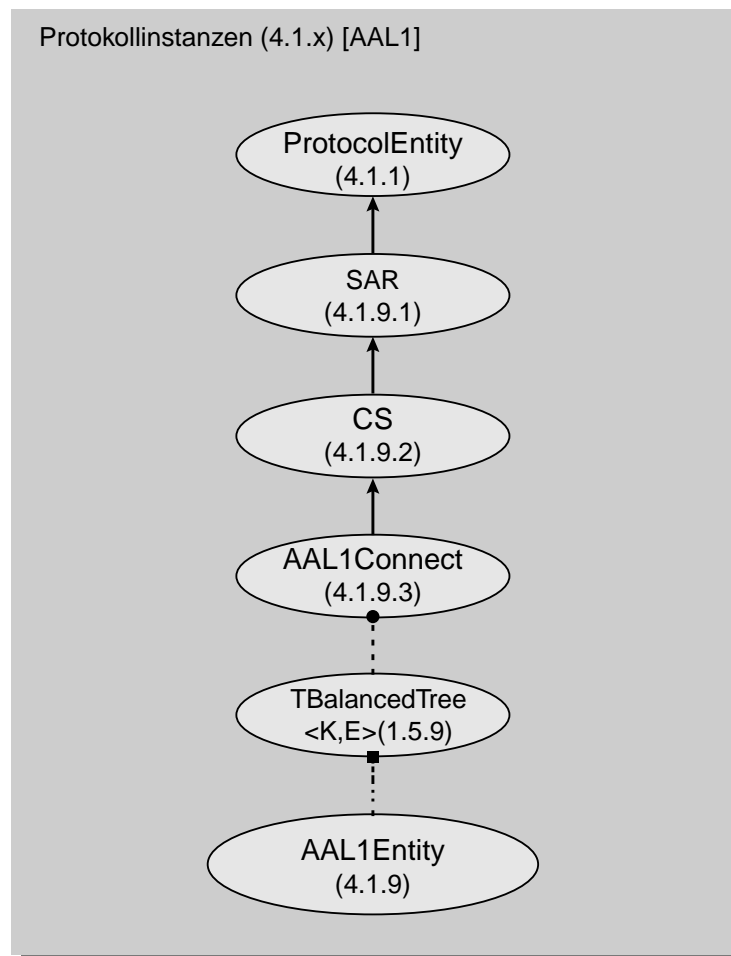


Abbildung 123 :Aufbau der Instanzen des AAL-Typs 1

- AAL1Entity (4.1.9)** Eine Klasse zur Repräsentation einer Instanz des AAL-Typs 1 gem. Abschnitt 3.8.2 auf Seite 64 ff. zur Übertragung synchroner Daten.
- AAL1Connect (4.1.9.3)** Objekte dieser Klasse stellen jeweils eine Verbindung über die AAL-Typ1-Instanz zwischen zwei Endgeräten dar. Die Verbindungen werden anhand einer zwischen AAL-Benutzer und AAL1 während des Verbindungsaufbaus ausgehandelten Verbindungsnummer identifiziert.
- CS (4.1.9.2)** Instanz des CS-Sublayers der AAL-Typ 1 - Instanz gemäß Abschnitt 3.8.2.4 auf Seite 66 ff.
- SAR (4.1.9.1)** SAR-Sublayer der AAL-Typ 1 - Instanz gemäß Abschnitt 3.8.2.3 auf Seite 65 f.

Protokollinstanzen des AAL Layers (Typ 3/4)

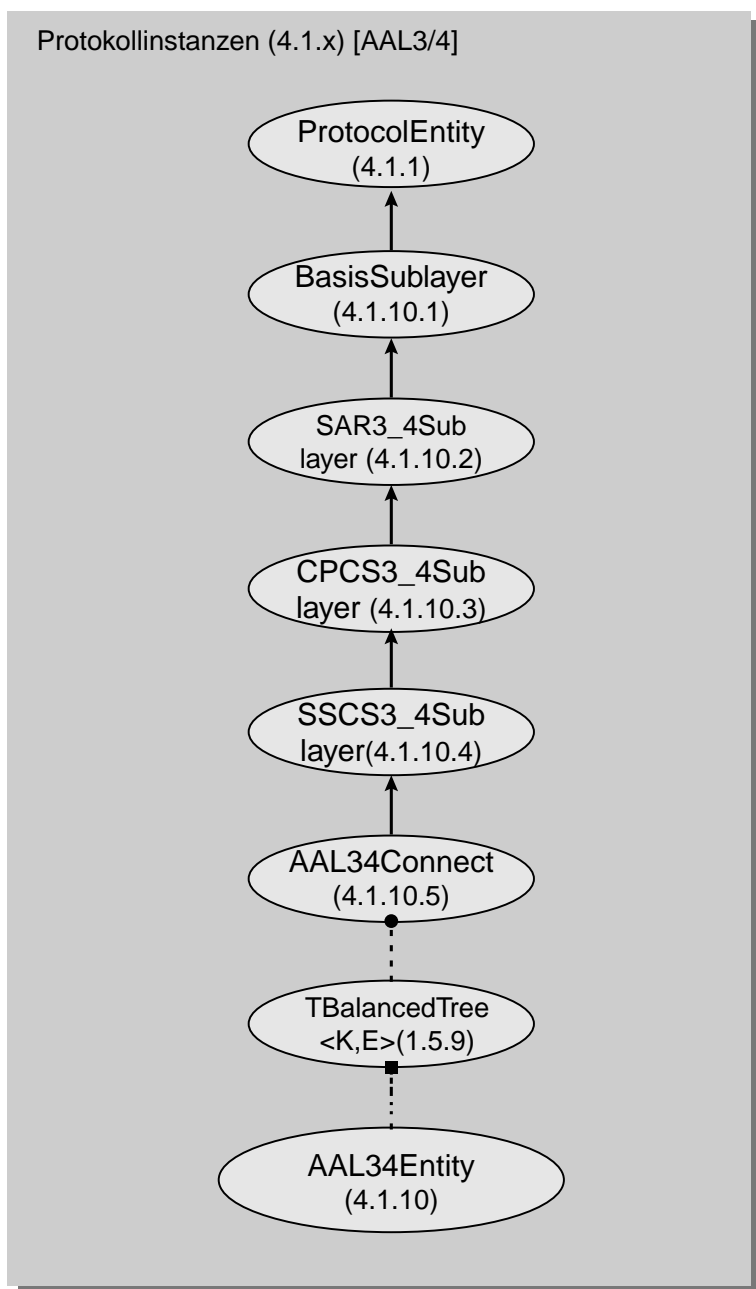


Abbildung 124 :Aufbau der Instanzen des AAL-Typs 3/4

AAL34Entity (4.1.10)

Eine Klasse zur Repräsentation einer Instanz des AAL-Typs 3/4 gem. Abschnitt 3.8.4 auf Seite 75 ff.

AAL34Connect (4.1.10.5)

Objekte dieser Klasse stellen jeweils eine Verbindung über die AAL-Typ 3/4 - Instanz zwischen zwei Endgeräten dar. Die Verbindungen werden anhand einer zwischen AAL-Benutzer und AAL3/4 während des Verbindungsaufbaus ausgehandelten MID identifiziert.

BasisSublayer (4.1.10.1)	Basisklasse aller Sublayer der AAL3/4-Instanz.
SAR3_4Sublayer (4.1.10.2)	Common Part der AAL3/4-Instanz gemäß Abschnitt 3.8.4.2 auf Seite 76 ff.
CPCS3_4 Sublayer (4.1.10.3)	
SSCS3_4Sublayer (4.1.10.4)	SSCS-Sublayer des AAL-Typs 3/4.

Protokollinstanzen des AAL Layers (Typ 5)

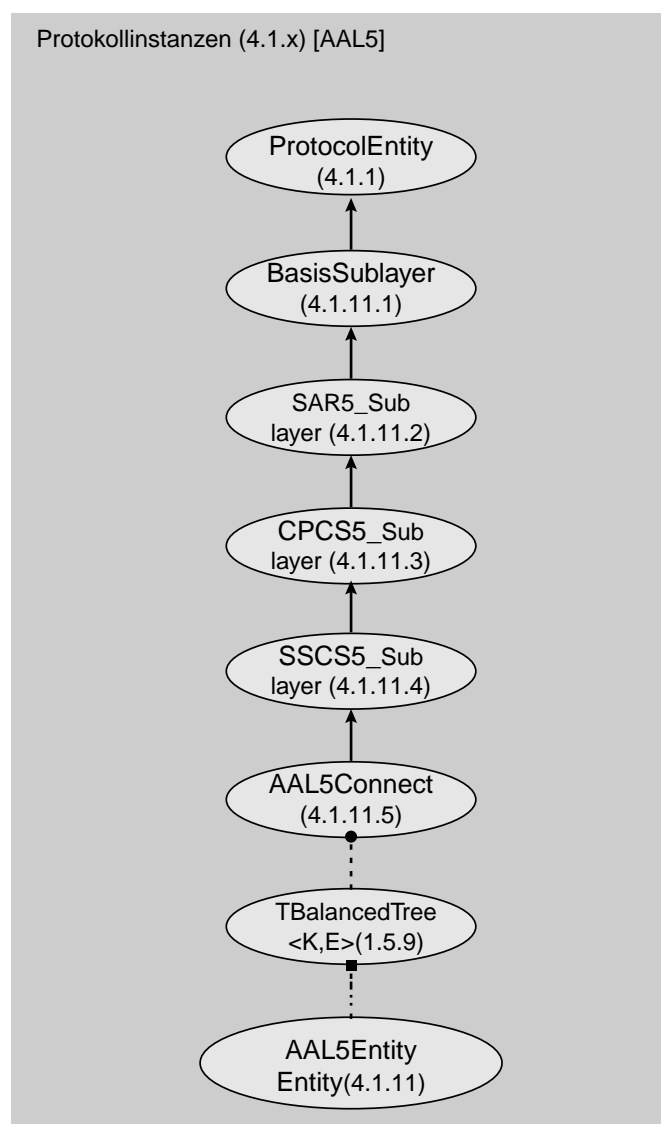


Abbildung 125 :Interner Aufbau der Protokollinstanzen vom AAL-Typ 5

AAL5Entity (4.1.11)	Eine Klasse zur Repräsentation einer Instanz des AAL-Typs 5 gem. Abschnitt 3.8.5 auf Seite 81 ff.
AAL5Connect (4.1.11.5)	Objekte dieser Klasse stellen jeweils eine Verbindung über die AAL-Typ 5 - Instanz zwischen zwei Endgeräten dar. Die Verbindungen werden anhand einer zwischen AAL-Benutzer und AAL5 während des Verbindungsaufbaus ausgehandelten Nummer identifiziert.
BasisSublayer (4.1.11.1)	Basisklasse aller Sublayer der AAL5-Instanz.
SAR5_Sublayer (4.1.10.2)	Common Part der AAL5-Instanz gemäß Abbildung 59 auf Seite 81.
CPCS5_Sublayer Sublayer (4.1.10.3)	
SSCS5_Sublayer (4.1.11.4)	Dieser Sublayer ist bisher nicht spezifiziert, hier wird lediglich die Umsetzung der Serviceprimitive wird vorgenommen.

Aufbau der Protokollinstanzen des AAL Layers (Typ S)

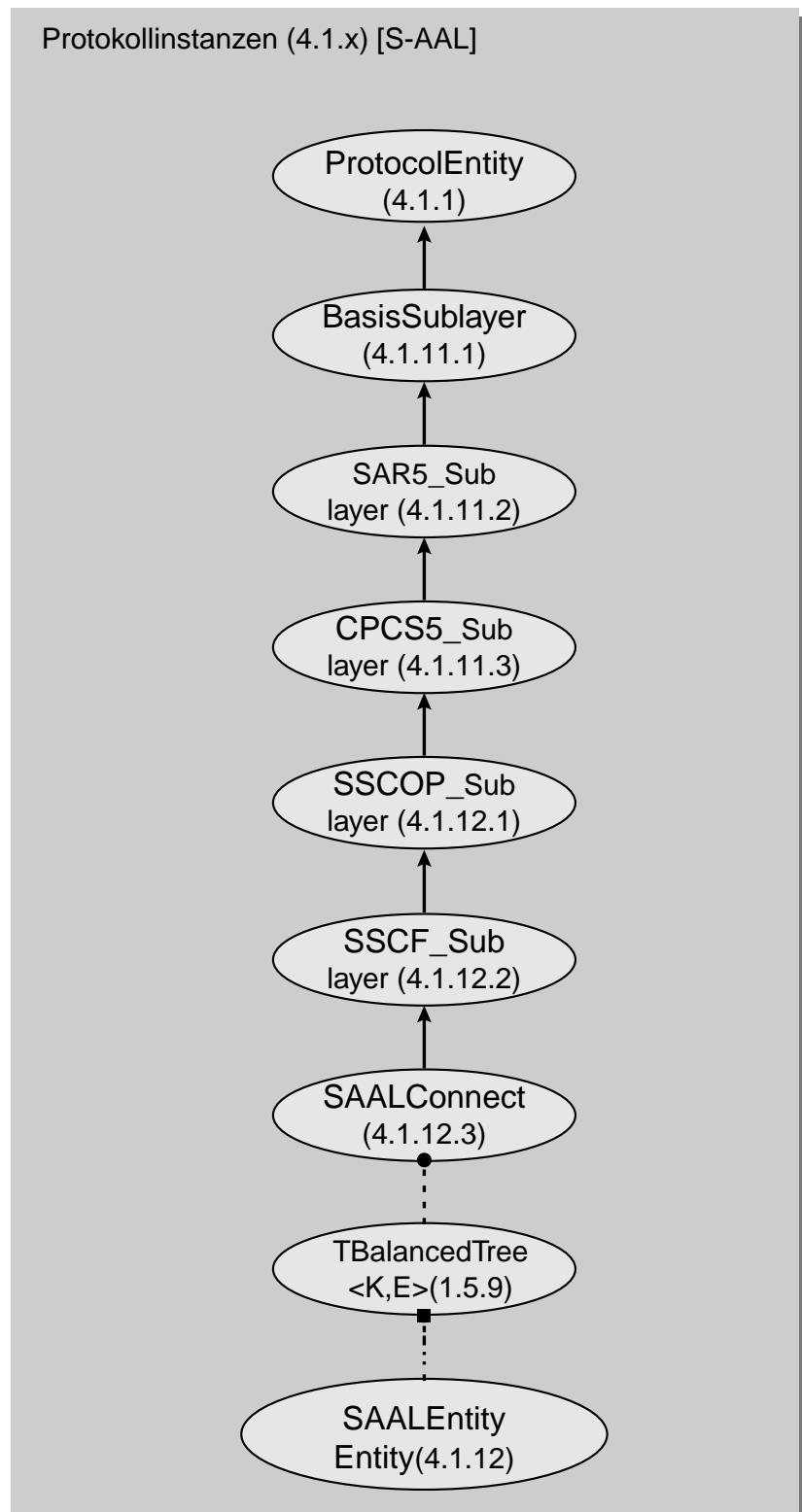


Abbildung 126 :Interner Aufbau der Protokollinstanzen vom AAL-Typ S

SAALEntity (4.1.12)	Eine Klasse zur Repräsentation einer Instanz des AAL-Typs S gem. Abbildung 62 auf Seite 84.
SAALConnect (4.1.12.3)	Objekte dieser Klasse stellen jeweils eine Verbindung über AAL-Typ S - Instanzen zwischen zwei Endgeräten dar. Die Verbindungen werden anhand einer zwischen AAL-Benutzer und SAAL während des Verbindungsaufbaus ausgehandelten Nummer identifiziert.
BasisSublayer (4.1.11.1)	Der Funktionsumfang des Common Parts der S-AAL entspricht demjenigen der AAL-Typ5-Instanz. Diese Klassen werden daher in die Ableitungshierarchie der S-AAL-Instanz übernommen.
SAR5_Sublayer (4.1.11.2)	
CPCS5_Sublayer Sublayer (4.1.11.3)	
SSCOP_Sublayer (4.1.12.1)	SSCOP-Sublayer gemäß Tabelle 43 auf Seite 86.
SSCF_Sublayer (4.1.12.2)	SSCF-Sublayer gemäß Tabelle 44 auf Seite 86.

C.5 Hardwaremodul

C.5.1 Hardware-Timing

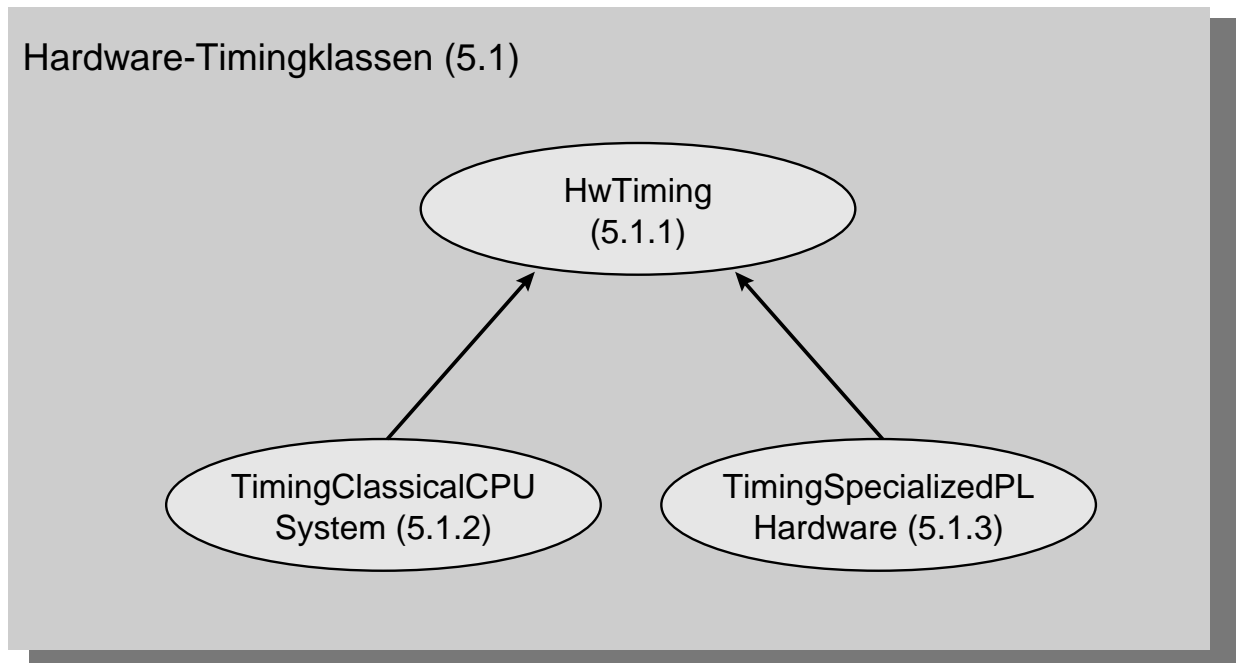


Abbildung 127 : Klassen zur Ermittlung der Bearbeitungszeiten der Hardware

HwTiming (5.1.1)	Basisklasse aller Timing-Klassen. Die Bearbeitungsdauern einer Hardwareinstanz für elementare Bearbeitungsschritte (Datenspeicherung, arithmetische und logische Verknüpfungen) werden als ganzzahlige Vielfache der Taktrate der Instanzen verwaltet. Bei der Bearbeitung einer Forderung überläßt die Hardwareinstanz dem Timing-Objekt die Summation der Zeiten für alle Teiloperationen.
TimingClassicalCPU-System (5.1.2)	Bearbeitungsdauern in Bediensystemen, die aus einer „klassischen“ Rechneranordnung (Neumann-Maschine), d.h. einen über einen Bus mit Programmspeicher (RAM) sowie Ein- und Ausgabeeinheiten verbundenen Prozessor (CPU ^a) bestehen.
TimingSpecialized-PLHardware (5.1.3)	Bearbeitungsdauern in einer auf die Abwicklung der Aufgaben des Physical Layers optimierten Hardwareanordnung.

^a. Central Processing Unit

C.5.2 Puffer

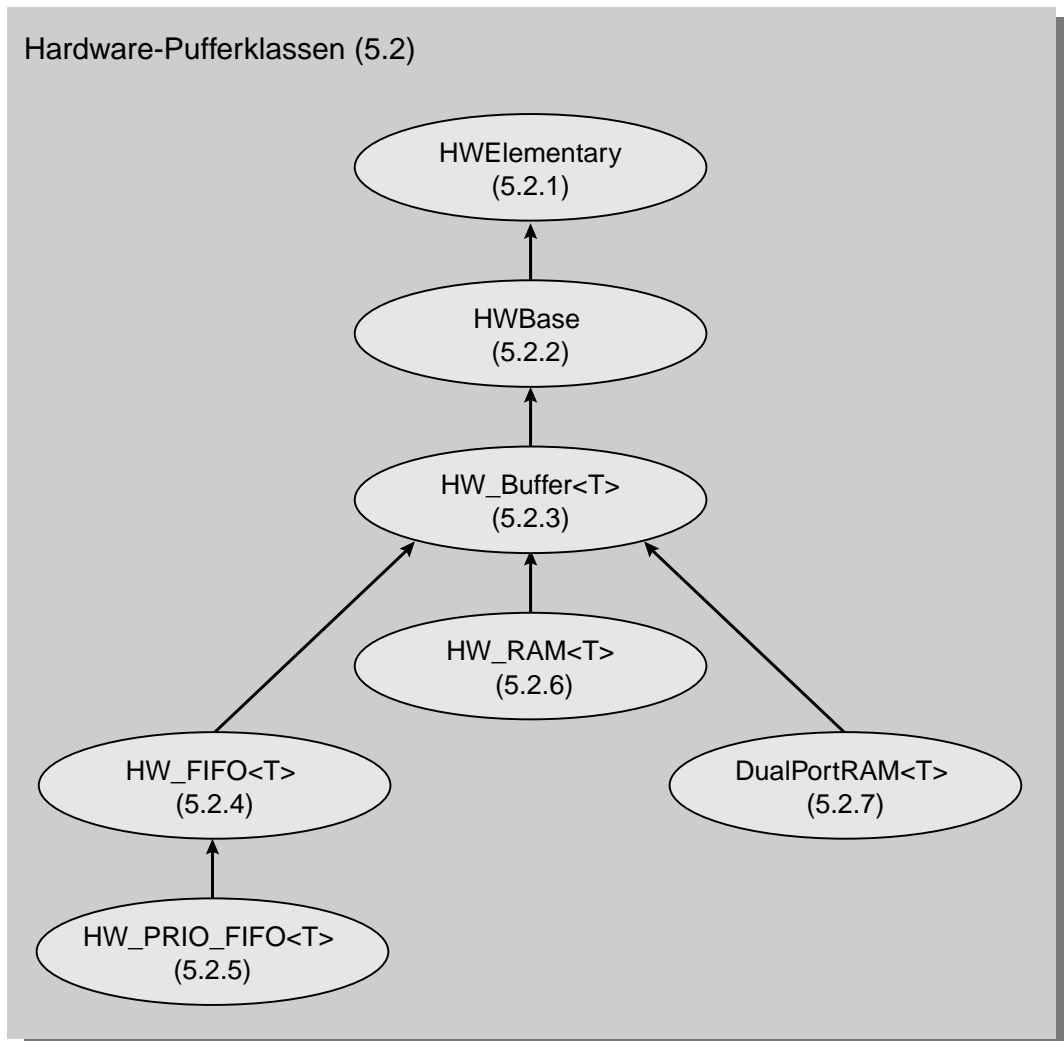


Abbildung 128 : Verschiedene Speicheranordnungen

HWElementary (5.2.1)	Basisklassen aller Hardwareinstanzen. Neben der Verwaltung der elementaren Kenngrößen wie Busbreite und Taktrate werden die virtuellen Funktionen für die abgeleiteten Klassen zur Verfügung gestellt.
HWBase (5.2.2)	
HW_Buffer<T> (5.2.3)	Basisklasse für alle Pufferklassen. Sie beherbergt die Angaben über Ein- und Auslesegeschwindigkeiten sowie das Puffervermögen der Speicher. Die virtuellen Funktionen zum Ein- und Auslesen von Daten werden gestellt.
HW_FIFO<T> (5.2.4)	Ein Speicherbaustein mit FIFO-Charakteristik.

HW_PRIO_FIFO<T> (5.2.5)	Pufferspeicher mit FIFO-Charakteristik und der Möglichkeit der Prioritätsvergabe für die Datenpakete. Bei drohendem Überlauf werden zuerst alle Pakete mit niedriger Priorität aus dem Pufferspeicher entfernt, bevor diejenigen mit hoher Priorität verloren gehen.
HW_RAM (5.2.6)	Das Modell für ein <i>Random Access Memory</i> .
DualPortRAM (5.2.7)	Ein <i>Random Access Memory</i> mit einem unabhängigen Schreib- und Leseport.

C.5.3 Verbindungsnetzwerke

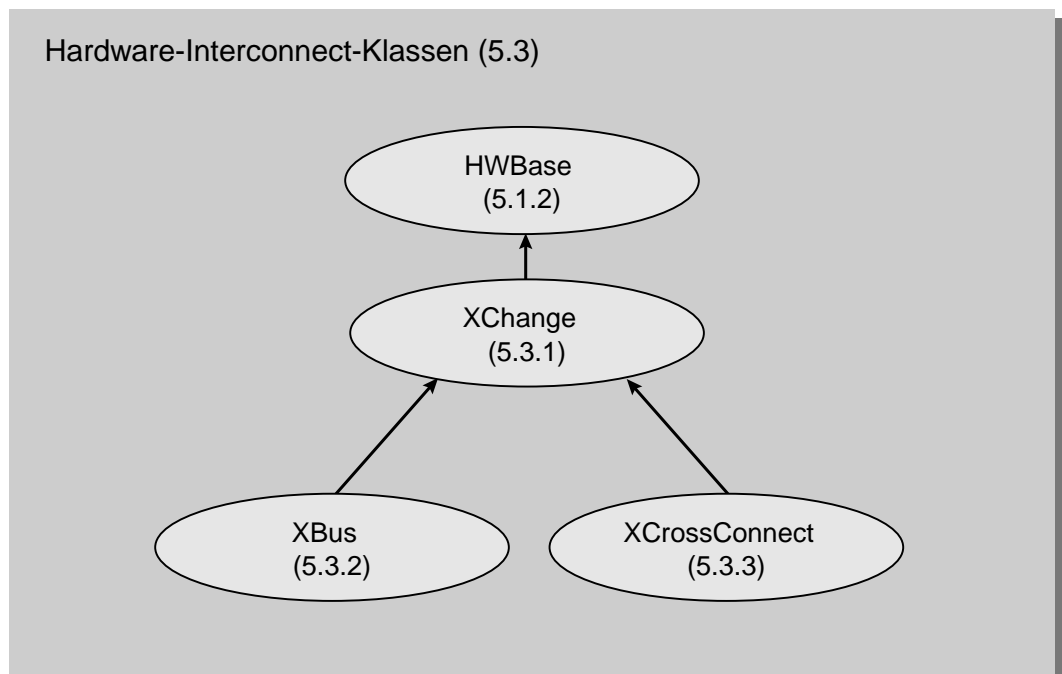


Abbildung 129 :Verbindungsnetzwerke

XChange (5.3.1)	Die Basisklasse aller Verbindungsnetzwerke.
XBus (5.3.2)	Eine busförmige Anordnung zur Verbindung von Hardwareinstanzen. Alle angeschlossenen Instanzen teilen sich das Verbindungsmedium, es ist nur ein Datentransfer zu einem Zeitpunkt möglich.
XCrossconnect (5.3.3)	Ein digitaler Crossconnect zur Verbindung zwischen Hardwareinstanzen. Gleichzeitige Datentransfervorgänge zwischen den angeschlossenen Einheiten sind möglich.

C.5.4 Bedieneinrichtungen

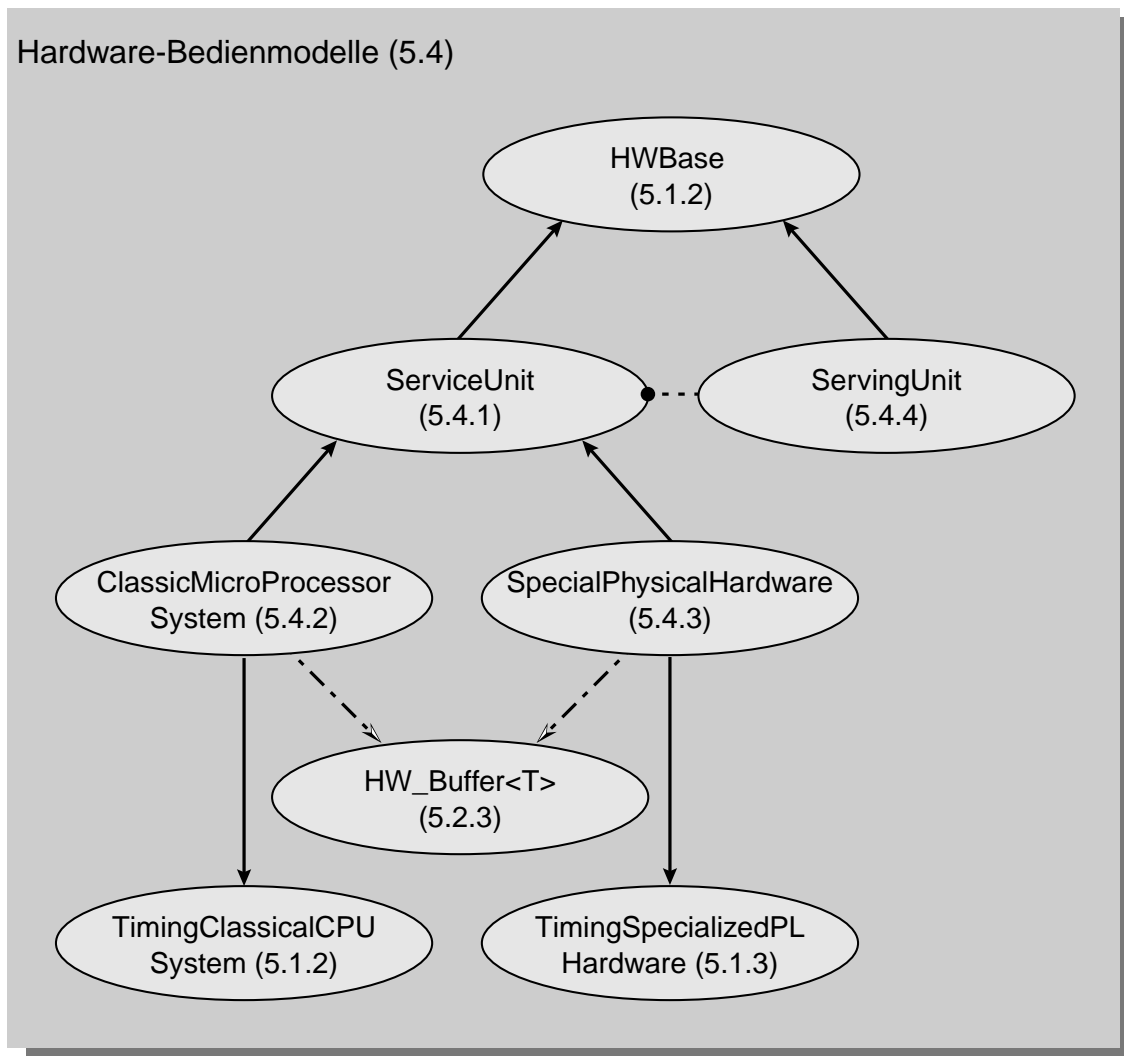


Abbildung 130 :Bedieneinrichtungen

ServiceUnit (5.4.1)	Die Basisklasse aller Bediensysteme.
ClassicMicro ProzessorSystem (5.4.2)	Das Modell einer Neumann-Maschine zur Abarbeitung von Forderungen.
SpecialPhysicalHard- ware (5.4.3)	Eine Hardwareanordnung, die auf die Abarbeitung der Aufgaben innerhalb der Instanzen des <i>Physical Layer</i> spezialisiert ist.

ServingUnit (5.4.4)

Die Objekte dieser Klasse sorgen für die Verteilung der Forderungen auf die einer Instanz zugeordneten (parallelen) Bedieneinrichtungen.

C.6 Konfigurationsdateien

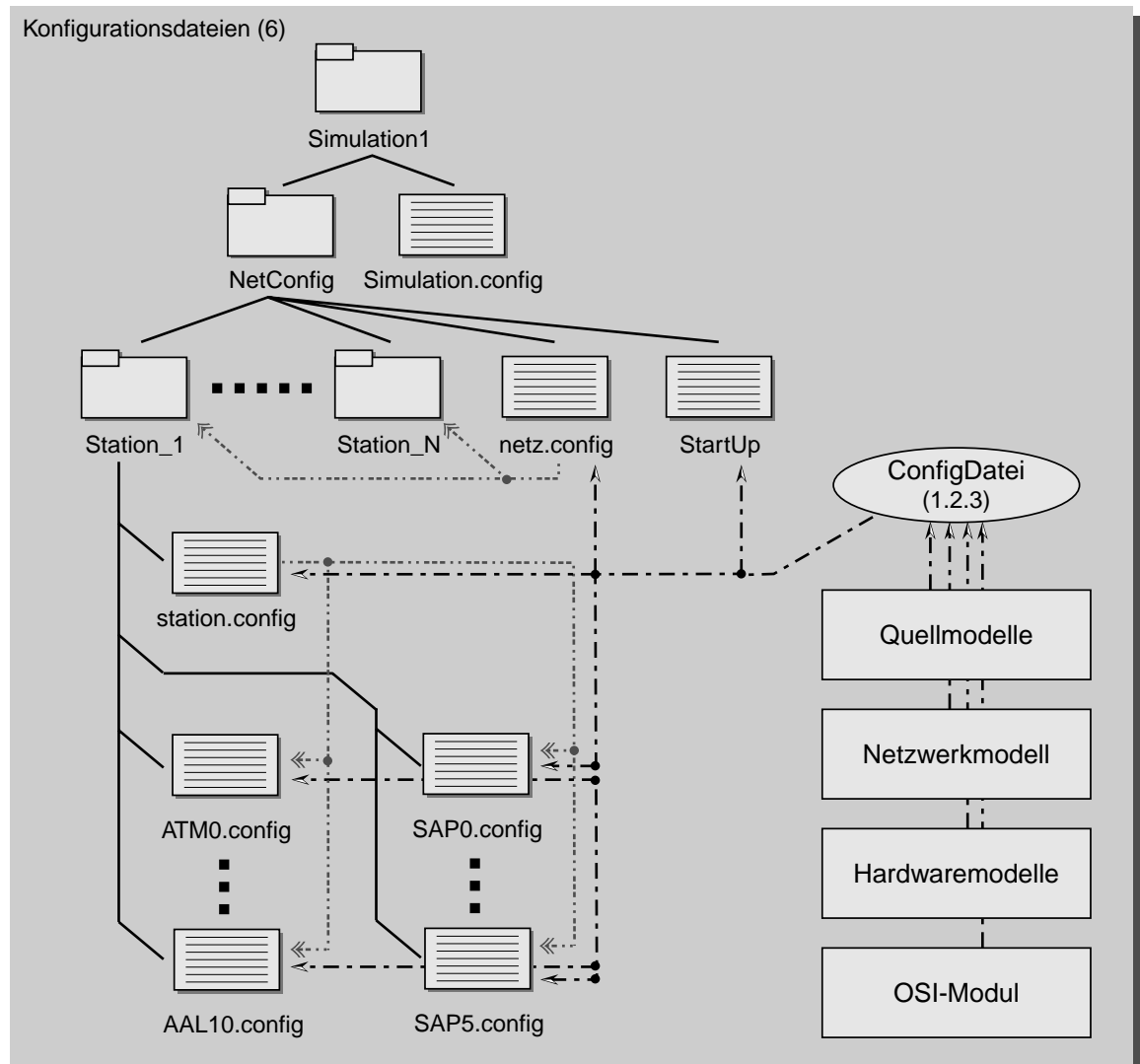


Abbildung 131 :Dateistruktur der Konfigurationsdateien

Die in Abbildung 131 auf Seite 231 gezeigten Module bedienen sich der Funktionen der Klasse *ConfigDatei* (1.2.3) um die Parameter aus dieser Dateistruktur auszulesen und in ihre Variablen zu übernehmen. Ein Beispiel für die Konfigurationsdateien eines kleinen Netzwerks ist in Anhang D auf Seite 237 ff. gegeben.

Ordner „Simulation1“	Wurzelverzeichnis der Simulation „Simulation1“. Unterhalb dieses Verzeichnis werden alle Konfigurationsinformationen und Ergebnisse dieses Simulationslaufes abgespeichert.
Ordner „NetConfig“	Ordner mit alle Konfigurationsdateien für diesen Simulationslauf.
Datei „Simulation.config“	Parameter, welche sich auf die gesamte Simulation beziehen.
Datei „netz.config“	Diese Datei enthält alle Eigenschaften des Netzwerkes (Adreßformat, Verbindungen, Stationsnamen ...).
Datei „StartUp“	Angaben über die in der Simulation zu verwendenden Datenquellen und deren Qualitätsansprüche (QoS-Parameter).
Ordner „Station1“ ... „StationN“	Jeder Ordner enthält die Definition einer Station des Netzwerkes. Der Name der Station entspricht dem Ordnernamen.
Datei „station.config“	Diese Datei enthält alle Eigenschaften des Station (Adressen, Protokollinstanzen, Koordinaten, ...).
Dateien „ATM0“ ... „AAL10“	Die Parameter der Protokollinstanzen, die an dieser Station verfügbar sind. Der Dateiname setzt sich zusammen aus der Protokollbezeichnung und einer fortlaufenden Nummer, die sich auf die Realisierung der Instanz bezieht.
Dateien „SAP0“ ... „SAP5“	Die Definition der SAPs an dieser Station.

C.7 Ergebnisdateien

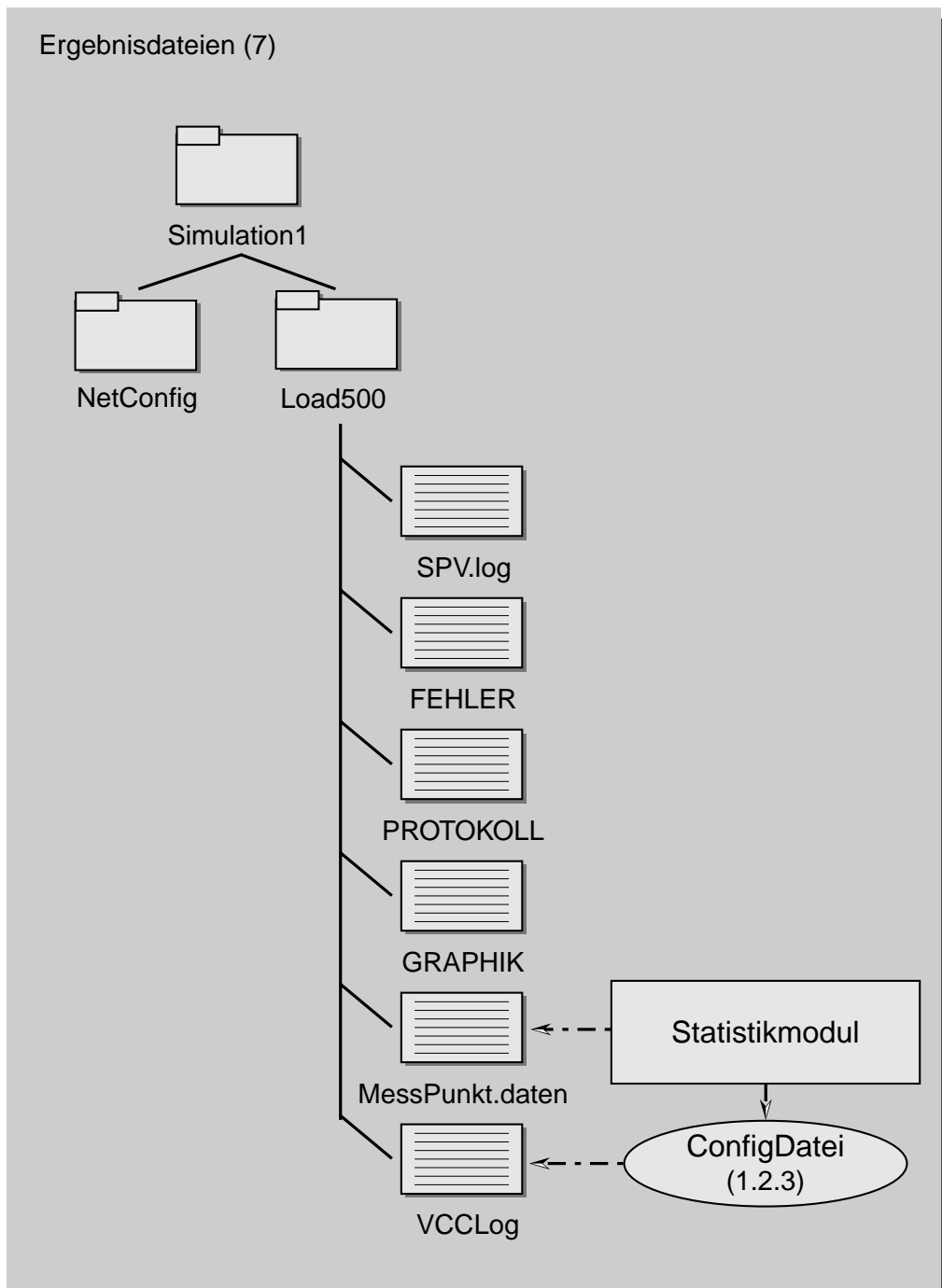


Abbildung 132 :Ergebnisdateien

Ordner
„Load500“

Alle Ergebnisse eines Simulationslaufes mit der Bezeichnung „Load500“ werden in diesem Ordner abgelegt.

Datei „SPV.log“	In dieser Datei zur Speicherverwaltungskontrolle wird bei der Allokation eines Speicherbereichs vom Betriebssystem (vergl. Abschnitt C.1.3 auf Seite 197) ein Vermerk über die verursachende Klasse gemacht. Die Datei dient dem Aufspüren von nicht freigegebenem dynamischem Speicher, der im Laufe der Simulation bis hin zur Erschöpfung der Speicherkapazität des Simulationsrechners und zum Simulationsabbruch führt.
Datei „FEHLER“	Log-Datei über die während des Simulationslaufs aufgetretenen internen Fehler. Die Datei dient der Eingrenzung von Fehlern in den Abläufen innerhalb der Simulation.
Datei „PROTOKOLL“	Diese Log-Datei wird von den Programmmodulen zur Ausgabe von Kontrollinformationen benutzt. Über diverse Präprozessor-Variablen wird die Ausgabe in mehreren Stufen von „sehr ausführlich“ bis „minimal“ eingestellt. Warnung : Eine ausführliche Dokumentation der internen Abläufe führt sehr schnell zu Dateien enormer Länge.
Datei „GRAPHIK“	Diese Datei dient der Ablage von Informationen, die dem Simulationstool (8) von der laufenden Simulation zur Verfügung gestellt werden. Das Datenformat entspricht dem bei der Konfiguration (vergl <i>ConfigDatei</i> (1.2.3)) verwendeten.
Datei „MessPunkt .daten“	Ein Datenfile, das von einem <i>MessPunkt</i> -Objekt (2.2.6) während der Simulation aufgenommen wurde. Die Kenngrößen der Zufallsvariablen können anschließend mithilfe der Auswertefunktionen der Klasse <i>MessPunkt</i> ermittelt werden.
Datei „VCCLog“	Diese date enthält Angaben über den Verlauf des Datentransfers über eine <i>Virtual Channel Connection</i> . Nach Abbruch des Datentransfers analysiert das Simulationstool die Qualität des empfangenen Datenstromes und legt das Ergebnis in dieser Datei ab.

C.8 Visualisierung

C.8.1 Konfigurationsdateien

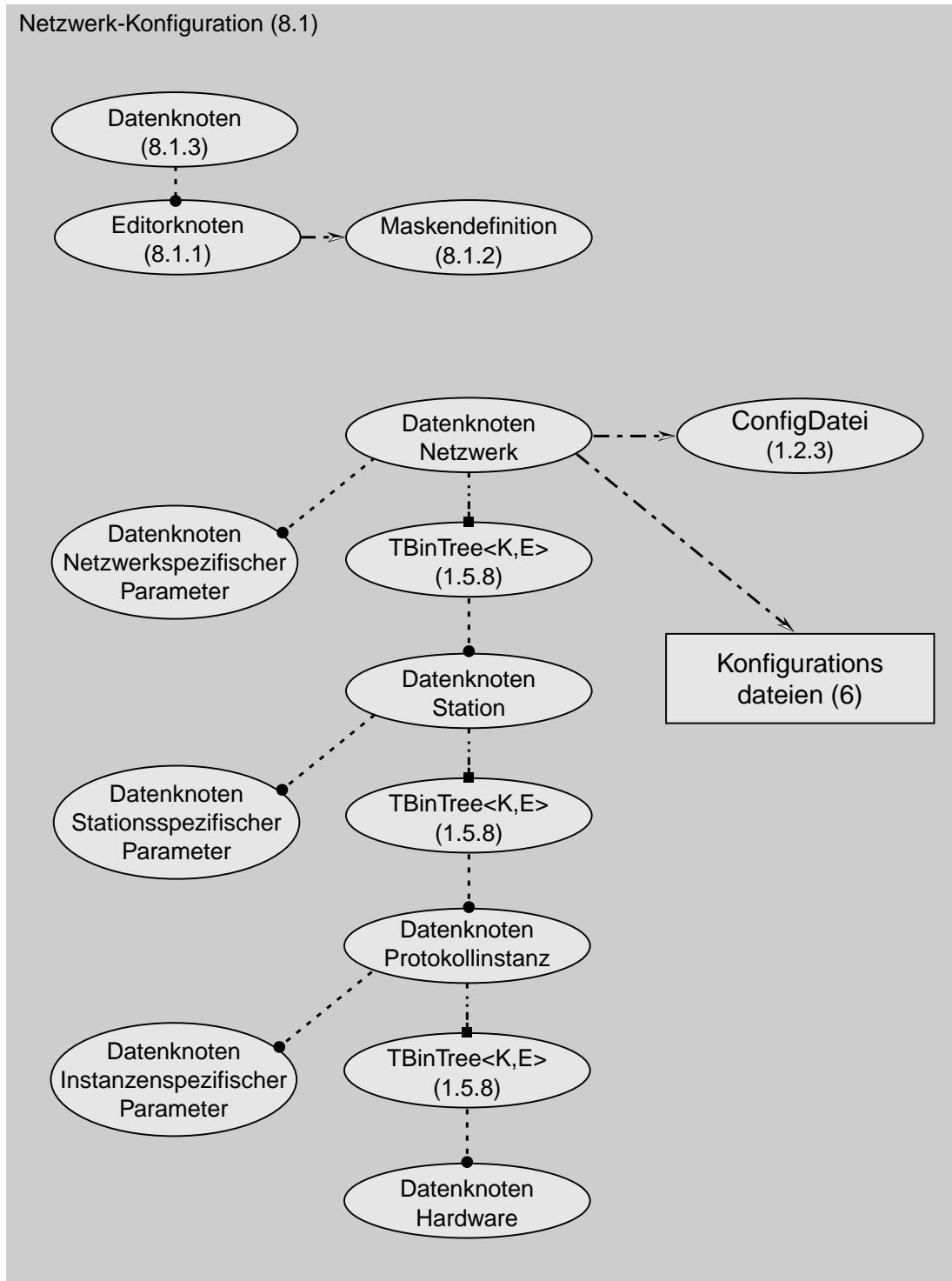


Abbildung 133 :Datenverwaltung im Netzwerkeditor

Editorknoten (8.1.1)	Eine Eingabemaske für ein beliebiges Element innerhalb der Netzwerk-hierarchie.
Maskendefinition (8.1.2)	Die Definition der Eingabemaske mit allen zu erfragenden Variablen und dem bei der Abfrage zu verwendenden Widget (Textfeld, Auswahlliste, ...) mit Minimal-, Maximal- und Default-Wert.
Datenknoten (8.1.3)	<p>Eine Instanz dieser Klasse enthält alle Parameter, die einem Objekt aus der Netzwerkhierarchie (Netzwerk, Station, Protokollinstanz o.ä.) zugeordnet sind .</p> <p>Alle Datenknoten sind in einer der Netzwerkstruktur in der Simulation entsprechenden mehrstufigen baumförmigen Datenstruktur angeordnet, wie in Abbildung 133 auf Seite 235 gezeigt ist.</p>

C.8.2 Visualisierung

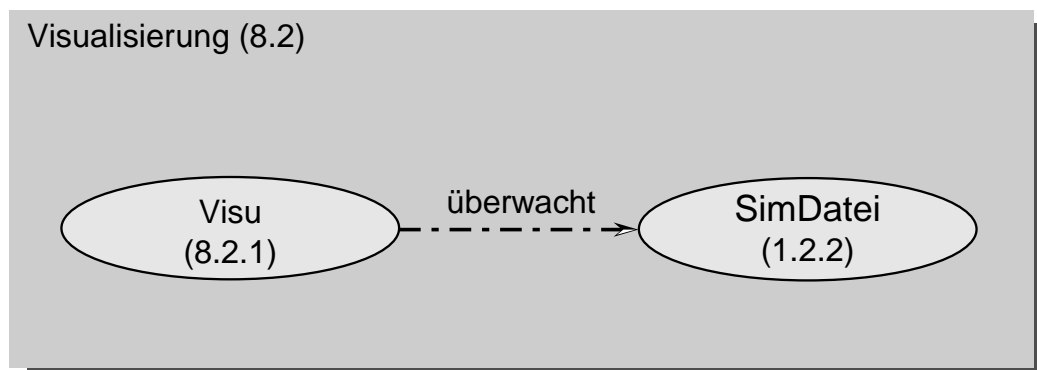


Abbildung 134 :Visualisierung von Abläufen und Ergebnissen

Visu (8.2.1)	Visualisierungsklasse zur graphischen Darstellung statistischer Größen während einer Simulation im Netzeditor. Die von Objekten der Klasse <i>MessPunkt</i> (2.2.6) erzeugten Dateien werden auf Änderungen überwacht. Statistische Kenngrößen oder Abläufe werden angezeigt.
---------------------	---

Anhang D Konfigurationsdateien

D.1 Simulationslauf 1

D.1.1 Quellenkonfiguration

```
#####
#                                     #
# Konfiguration der Datenquellen #
#                                     #
#####
#
# =====
# The CAC-Algorithm used
# =====
#
[GlobalParameters]
#
CallAdmissionControl = ALLOCATION_ADAPTIVE
#
# Restart of the VCCs
#
VCCRestartMeanTime           = 1.0e-2
TimeEndOfSimulation          = 1.0
#
#
# =====
# =
# = The sensitivity of different Datasources in percent
# = Defines the „severity“ of degradation of quality for
# = this type of data.
# =
# =====
#
[REALTIME_VIDEO]
#
MeanRate                      = 1.0 # the average packet rate
PeakRate                      = 1.0 # the peak bitrate
BitError                      = 0.4 # the probability of bit errors
PacketLoss                    = 0.4 # the probability of packet loss
Delay                         = 0.05 # the delay between sender and
receiver
DelayCentralMoment2           = 1.0 # the variance of delay
DelayCentralMoment3           = 0.8 # the 3rd central moment of
delay
DelayCentralMoment4           = 0.5 # the 4th central moment
#
#
```

```
[COMPRESSED_VIDEO]
#
MeanRate           = 1.0
PeakRate           = 0.8
BitError           = 0.7
PacketLoss         = 0.7
Delay              = 0.05
DelayCentralMoment2 = 1.0
DelayCentralMoment3 = 0.8
DelayCentralMoment4 = 0.5
#
[HIGH_QUALITY_AUDIO]
#
MeanRate           = 1.0
PeakRate           = 1.0
BitError           = 0.9
PacketLoss         = 0.9
Delay              = 0.01
DelayCentralMoment2 = 0.99
DelayCentralMoment3 = 0.8
DelayCentralMoment4 = 0.5
#
[LOW_QUALITY_AUDIO]
#
MeanRate           = 1.0
PeakRate           = 0.8
BitError           = 0.7
PacketLoss         = 0.7
Delay              = 0.5
DelayCentralMoment2 = 0.8
DelayCentralMoment3 = 0.5
DelayCentralMoment4 = 0.4
#
[SPEECH]
#
MeanRate           = 1.0
PeakRate           = 0.9
BitError           = 0.2
PacketLoss         = 0.2
Delay              = 0.9
DelayCentralMoment2 = 0.7
DelayCentralMoment3 = 0.6
DelayCentralMoment4 = 0.5
#
[TEXT_DATA]
#
MeanRate           = 0.6
PeakRate           = 0.1
BitError           = 1.0
```

```

PacketLoss                = 1.0
Delay                    = 0.1
DelayCentralMoment2      = 0.01
DelayCentralMoment3      = 0.01
DelayCentralMoment4      = 0.01
#
# =====
# The weight of each component when calculating the value
# of the VCC-data
# =====
#
[VALUE_CALCULATION]
#
MeanRate                  = 1.0
BitError                  = 1.0
PacketLoss                = 1.0
Delay                    = 1.0
DelayCentralMoment2      = 1.0
DelayCentralMoment3      = 1.0
DelayCentralMoment4      = 1.0
#
#-----
# The DataSources
#-----
#
[DataSources]
Anzahl = 12
#
# =====
[DataSource0]
# =====
#
DataType                  = REALTIME_VIDEO
SourceStation              = Station_1
SinkStation                = Station_2
Layer3Protokoll           = NETWORK
Layer3ProtokollNumber     = 0
Type                      = DATA_GENERATOR_CONST
TypePacketLengthGenerator = DATA_GENERATOR_CONST
MeanPacketSize[Byte]      = 47
SetupTime                 = 1.0e-6
Duration                  = 5.0e-2
#
# The QoS-Parameters
# for a AAL1
#
QoSUpstream(MeanBitRate[kBit/s]) = 8448
QoSUpstream(PeakBitRate[kBit/s]) = 8448
QoSUpstream(RelativeBurstDuration) = 1.0

```

```
QoSUpstream(BitRateType)           = CBR
QoSUpstream(ConnectionType)        = CONNECTION_ORIENTED
QoSUpstream(TimeRestriction)       = REALTIME_STRICTLY_REQUIRED
QoSUpstream(ErrorProbability)      = 1e-5
QoSUpstream(PacketLossProbability) = 1e-1
QoSUpstream(MaximumDelay)          = 1.0e-5
QoSUpstream(InterarrivalTimeVariance) = 1.0e-7
QoSUpstream(InterarrivalTimeCentralMoment3) = 1.0e-9
QoSUpstream(InterarrivalTimeCentralMoment4) = 1.0e-11
#
QoSDownstream(MeanBitRate[kBit/s]) = 8448
QoSDownstream(PeakBitRate[kBit/s]) = 8448
QoSDownstream(RelativeBurstDuration) = 1.0
QoSDownstream(BitRateType)         = CBR
QoSDownstream(ConnectionType)     = CONNECTION_ORIENTED
QoSDownstream(TimeRestriction)    = REALTIME_STRICTLY_REQUIRED
QoSDownstream(ErrorProbability)   = 1e-5
QoSDownstream(PacketLossProbability) = 1e-1
QoSDownstream(MaximumDelay)       = 1.0e-5
QoSDownstream(InterarrivalTimeVariance) = 1.0e-7
QoSDownstream(InterarrivalTimeCentralMoment3) = 1.0e-9
QoSDownstream(InterarrivalTimeCentralMoment4) = 1.0e-11
#
# =====
[DataSource1]
# =====
#
DataType           = REALTIME_VIDEO
SourceStation      = Station_1
SinkStation        = Station_2
Layer3Protokoll    = NETWORK
Layer3ProtokollNumber = 0
Type               = DATA_GENERATOR_CONST
TypePacketLengthGenerator = DATA_GENERATOR_CONST
MeanPacketSize[Byte] = 47
SetupTime         = 2.0e-6
Duration          = 5.0e-2
#
# The QoS-Parameters
# for a AAL1
#
QoSUpstream(MeanBitRate[kBit/s])           = 8448
QoSUpstream(PeakBitRate[kBit/s])          = 8448
QoSUpstream(RelativeBurstDuration)        = 1.0
QoSUpstream(BitRateType)                  = CBR
QoSUpstream(ConnectionType)               = CONNECTION_ORIENTED
QoSUpstream(TimeRestriction)              = REALTIME_STRICTLY_REQUIRED
QoSUpstream(ErrorProbability)             = 1e-5
QoSUpstream(PacketLossProbability)        = 1e-2
```

```

QoSUpstream(MaximumDelay)                = 1.0e-5
QoSUpstream(InterarrivalTimeVariance)    = 1.0e-7
QoSUpstream(InterarrivalTimeCentralMoment3) = 1.0e-9
QoSUpstream(InterarrivalTimeCentralMoment4) = 1.0e-11
#
QoSDownstream(MeanBitRate[kBit/s])        = 8448
QoSDownstream(PeakBitRate[kBit/s])        = 8448
QoSDownstream(RelativeBurstDuration)      = 1.0
QoSDownstream(BitRateType)                = CBR
QoSDownstream(ConnectionType)             = CONNECTION_ORIENTED
QoSDownstream(TimeRestriction)            = REALTIME_STRICTLY_REQUIRED
QoSDownstream(ErrorProbability)           = 1e-5
QoSDownstream(PacketLossProbability)      = 1e-2
QoSDownstream(MaximumDelay)               = 1.0e-5
QoSDownstream(InterarrivalTimeVariance)   = 1.0e-7
QoSDownstream(InterarrivalTimeCentralMoment3) = 1.0e-9
QoSDownstream(InterarrivalTimeCentralMoment4) = 1.0e-11
#
# =====
[DataSource2]
# =====
#
DataType                                  = HIGH_QUALITY_AUDIO
SourceStation                             = Station_1
SinkStation                               = Station_2
Layer3Protokoll                           = NETWORK
Layer3ProtokollNumber                     = 0
Type                                       = DATA_GENERATOR_MMPP
TypePacketLengthGenerator                  = DATA_GENERATOR_CONST
MeanPacketSize[Byte]                      = 47
SetupTime                                 = 3.0e-6
Duration                                  = 3.0e-2
#
#
QoSUpstream(MeanBitRate[kBit/s])          = 2048
QoSUpstream(PeakBitRate[kBit/s])          = 10000
QoSUpstream(RelativeBurstDuration)        = 0.1
QoSUpstream(BitRateType)                  = VBR
QoSUpstream(ConnectionType)               = CONNECTION_LESS
QoSUpstream(TimeRestriction)              = REALTIME_NOT_REQUIRED
QoSUpstream(ErrorProbability)             = 1.0e-5
QoSUpstream(PacketLossProbability)        = 1.0e-3
QoSUpstream(MaximumDelay)                 = 1.0e-5
QoSUpstream(InterarrivalTimeVariance)     = 1.0e-7
QoSUpstream(InterarrivalTimeCentralMoment3) = 1.0e-9
QoSUpstream(InterarrivalTimeCentralMoment4) = 1.0e-10
#
QoSDownstream(MeanBitRate[kBit/s])        = 2048
QoSDownstream(PeakBitRate[kBit/s])        = 10000

```

```
QoSDownstream(RelativeBurstDuration) = 0.1
QoSDownstream(BitRateType) = VBR
QoSDownstream(ConnectionType) = CONNECTION_LESS
QoSDownstream(TimeRestriction) = REALTIME_NOT_REQUIRED
QoSDownstream(ErrorProbability) = 1.0e-5
QoSDownstream(PacketLossProbability) = 1.0e-3
QoSDownstream(MaximumDelay) = 1.0e-5
QoSDownstream(InterarrivalTimeVariance) = 1.0e-7
QoSDownstream(InterarrivalTimeCentralMoment3) = 1.0e-9
QoSDownstream(InterarrivalTimeCentralMoment4) = 1.0e-10
#
# =====
[DataSource3]
# =====
#
DataType = HIGH_QUALITY_AUDIO
SourceStation = Station_1
SinkStation = Station_2
Layer3Protokoll = NETWORK
Layer3ProtokollNumber = 0
Type = DATA_GENERATOR_MMPP
TypePacketLengthGenerator = DATA_GENERATOR_UNIFORM
MeanPacketSize[Byte] = 47
SetupTime = 4.0e-6
Duration = 3.0e-2
#
#
QoSUpstream(MeanBitRate[kBit/s]) = 2048
QoSUpstream(PeakBitRate[kBit/s]) = 10000
QoSUpstream(RelativeBurstDuration) = 0.1
QoSUpstream(BitRateType) = VBR
QoSUpstream(ConnectionType) = CONNECTION_LESS
QoSUpstream(TimeRestriction) = REALTIME_NOT_REQUIRED
QoSUpstream(ErrorProbability) = 1.0e-5
QoSUpstream(PacketLossProbability) = 1.0e-3
QoSUpstream(MaximumDelay) = 1.0e-5
QoSUpstream(InterarrivalTimeVariance) = 1.0e-7
QoSUpstream(InterarrivalTimeCentralMoment3) = 1.0e-9
QoSUpstream(InterarrivalTimeCentralMoment4) = 1.0e-10
#
QoSDownstream(MeanBitRate[kBit/s]) = 2048
QoSDownstream(PeakBitRate[kBit/s]) = 10000
QoSDownstream(RelativeBurstDuration) = 0.1
QoSDownstream(BitRateType) = VBR
QoSDownstream(ConnectionType) = CONNECTION_LESS
QoSDownstream(TimeRestriction) = REALTIME_NOT_REQUIRED
QoSDownstream(ErrorProbability) = 1.0e-5
QoSDownstream(PacketLossProbability) = 1.0e-3
QoSDownstream(MaximumDelay) = 1.0e-5
```



```

QoSDownstream(InterarrivalTimeVariance)          = 1.0e-7
QoSDownstream(InterarrivalTimeCentralMoment3)    = 1.0e-9
QoSDownstream(InterarrivalTimeCentralMoment4)    = 1.0e-10
#
# =====
[DataSource4]
# =====
#
DataType                                           = LOW_QUALITY_AUDIO
SourceStation                                       = Station_1
SinkStation                                         = Station_2
Layer3Protokoll                                     = NETWORK
Layer3ProtokollNumber                             = 0
Type                                                = DATA_GENERATOR_MMPP
TypePacketLengthGenerator                         = DATA_GENERATOR_CONST
MeanPacketSize[Byte]                              = 124
SetupTime                                          = 3.0e-4
Duration                                            = 1.30e-1
#
# The QoS-Requirements
#
QoSUpstream(MeanBitRate[kBit/s])                  = 384
QoSUpstream(PeakBitRate[kBit/s])                  = 1024
QoSUpstream(RelativeBurstDuration)                 = 0.2
QoSUpstream(BitRateType)                          = VBR
QoSUpstream(ConnectionType)                       = CONNECTION_LESS
QoSUpstream(TimeRestriction)                      = REALTIME_NOT_REQUIRED
QoSUpstream(ErrorProbability)                     = 1.0e-5
QoSUpstream(PacketLossProbability)                 = 1.0e-3
QoSUpstream(MaximumDelay)                         = 1.0e-3
QoSUpstream(InterarrivalTimeVariance)             = 1.0e-6
QoSUpstream(InterarrivalTimeCentralMoment3)       = 1.0e-8
QoSUpstream(InterarrivalTimeCentralMoment4)       = 1.0e-10
#
QoSDownstream(MeanBitRate[kBit/s])                 = 384
QoSDownstream(PeakBitRate[kBit/s])                 = 1024
QoSDownstream(RelativeBurstDuration)               = 0.2
QoSDownstream(BitRateType)                        = VBR
QoSDownstream(ConnectionType)                     = CONNECTION_LESS
QoSDownstream(TimeRestriction)                    = REALTIME_NOT_REQUIRED
QoSDownstream(ErrorProbability)                   = 1.0e-5
QoSDownstream(PacketLossProbability)               = 1.0e-3
QoSDownstream(MaximumDelay)                       = 1.0e-3
QoSDownstream(InterarrivalTimeVariance)           = 1.0e-6
QoSDownstream(InterarrivalTimeCentralMoment3)     = 1.0e-8
QoSDownstream(InterarrivalTimeCentralMoment4)     = 1.0e-10
#
#
# =====

```

```
[DataSource5]
# =====
#
DataType                = LOW_QUALITY_AUDIO
SourceStation           = Station_1
SinkStation             = Station_2
Layer3Protokoll        = NETWORK
Layer3ProtokollNumber  = 0
Type                   = DATA_GENERATOR_MMPP
TypePacketLengthGenerator = DATA_GENERATOR_CONST
MeanPacketSize[Byte]   = 124
SetupTime              = 2.0e-4
Duration               = 1.30e-1
#
QoSUpstream(MeanBitRate[kBit/s]) = 384
QoSUpstream(PeakBitRate[kBit/s]) = 1024
QoSUpstream(RelativeBurstDuration) = 0.2
QoSUpstream(BitRateType)          = VBR
QoSUpstream(ConnectionType)       = CONNECTION_LESS
QoSUpstream(TimeRestriction)      = REALTIME_NOT_REQUIRED
QoSUpstream(ErrorProbability)     = 1.0e-5
QoSUpstream(PacketLossProbability) = 1.0e-2
QoSUpstream(MaximumDelay)         = 1.0e-4
QoSUpstream(InterarrivalTimeVariance) = 1.0e-6
QoSUpstream(InterarrivalTimeCentralMoment3) = 1.0e-8
QoSUpstream(InterarrivalTimeCentralMoment4) = 1.0e-10
#
#
QoSDownstream(MeanBitRate[kBit/s]) = 384
QoSDownstream(PeakBitRate[kBit/s]) = 1024
QoSDownstream(RelativeBurstDuration) = 0.2
QoSDownstream(BitRateType)          = VBR
QoSDownstream(ConnectionType)       = CONNECTION_LESS
QoSDownstream(TimeRestriction)      = REALTIME_NOT_REQUIRED
QoSDownstream(ErrorProbability)     = 1.0e-5
QoSDownstream(PacketLossProbability) = 1.0e-2
QoSDownstream(MaximumDelay)         = 1.0e-4
QoSDownstream(InterarrivalTimeVariance) = 1.0e-6
QoSDownstream(InterarrivalTimeCentralMoment3) = 1.0e-8
QoSDownstream(InterarrivalTimeCentralMoment4) = 1.0e-10
#
# =====
[DataSource6]
# =====
# (leads to AAL5)
#
DataType                = TEXT_DATA
SourceStation           = Station_1
SinkStation             = Station_2
```

```

Layer3Protokoll           = NETWORK
Layer3ProtokollNumber    = 0
Type                      = DATA_GENERATOR_NEG_EXP
TypePacketLengthGenerator = DATA_GENERATOR_UNIFORM
MeanPacketSize[Byte]     = 1024
SetupTime                 = 2.2e-4
Duration                  = 1.0e-2
#
QoSUpstream(MeanBitRate[kBit/s]) = 10000
QoSUpstream(PeakBitRate[kBit/s]) = 15000
QoSUpstream(RelativeBurstDuration) = 0.1
QoSUpstream(BitRateType)         = VBR
QoSUpstream(ConnectionType)     = CONNECTION_ORIENTED
QoSUpstream(TimeRestriction)    = REALTIME_NOT_REQUIRED
QoSUpstream(ErrorProbability)   = 1.0e-5
QoSUpstream(PacketLossProbability) = 1.0e-2
QoSUpstream(MaximumDelay)       = 1.0e-5
QoSUpstream(InterarrivalTimeVariance) = 1.0e-7
QoSUpstream(InterarrivalTimeCentralMoment3) = 1.0e-9
QoSUpstream(InterarrivalTimeCentralMoment4) = 1.0e-10
#
#
QoSDownstream(MeanBitRate[kBit/s]) = 10000
QoSDownstream(PeakBitRate[kBit/s]) = 15000
QoSDownstream(RelativeBurstDuration) = 0.1
QoSDownstream(BitRateType)         = VBR
QoSDownstream(ConnectionType)     = CONNECTION_ORIENTED
QoSDownstream(TimeRestriction)    = REALTIME_NOT_REQUIRED
QoSDownstream(ErrorProbability)   = 1.0e-5
QoSDownstream(PacketLossProbability) = 1.0e-2
QoSDownstream(MaximumDelay)       = 1.0e-5
QoSDownstream(InterarrivalTimeVariance) = 1.0e-7
QoSDownstream(InterarrivalTimeCentralMoment3) = 1.0e-9
QoSDownstream(InterarrivalTimeCentralMoment4) = 1.0e-10
#
# =====
[DataSource7]
# =====
# (leads to AAL5)
#
DataType                 = TEXT_DATA
SourceStation            = Station_1
SinkStation              = Station_2
Layer3Protokoll         = NETWORK
Layer3ProtokollNumber   = 0
Type                    = DATA_GENERATOR_NEG_EXP
TypePacketLengthGenerator = DATA_GENERATOR_NEG_EXP
MeanPacketSize[Byte]   = 2048
SetupTime               = 2.2e-4

```

```
Duration = 1.0e-2
#
QoSUpstream(MeanBitRate[kBit/s]) = 5000
QoSUpstream(PeakBitRate[kBit/s]) = 7500
QoSUpstream(RelativeBurstDuration) = 0.5
QoSUpstream(BitRateType) = VBR
QoSUpstream(ConnectionType) = CONNECTION_ORIENTED
QoSUpstream(TimeRestriction) = REALTIME_NOT_REQUIRED
QoSUpstream(ErrorProbability) = 1.0e-5
QoSUpstream(PacketLossProbability) = 1.0e-3
QoSUpstream(MaximumDelay) = 1.0e-5
QoSUpstream(InterarrivalTimeVariance) = 1.0e-7
QoSUpstream(InterarrivalTimeCentralMoment3) = 1.0e-9
QoSUpstream(InterarrivalTimeCentralMoment4) = 1.0e-10
#
#
QoSDownstream(MeanBitRate[kBit/s]) = 5000
QoSDownstream(PeakBitRate[kBit/s]) = 7500
QoSDownstream(RelativeBurstDuration) = 0.1
QoSDownstream(BitRateType) = VBR
QoSDownstream(ConnectionType) = CONNECTION_ORIENTED
QoSDownstream(TimeRestriction) = REALTIME_NOT_REQUIRED
QoSDownstream(ErrorProbability) = 1.0e-5
QoSDownstream(PacketLossProbability) = 1.0e-3
QoSDownstream(MaximumDelay) = 1.0e-5
QoSDownstream(InterarrivalTimeVariance) = 1.0e-7
QoSDownstream(InterarrivalTimeCentralMoment3) = 1.0e-9
QoSDownstream(InterarrivalTimeCentralMoment4) = 1.0e-10
#
# =====
[DataSource8]
# =====
#
DataType = SPEECH
SourceStation = Station_1
SinkStation = Station_2
Layer3Protokoll = NETWORK
Layer3ProtokollNumber = 0
Type = DATA_GENERATOR_ON_OFF
TypePacketLengthGenerator = DATA_GENERATOR_CONST
MeanPacketSize[Byte] = 8
SetupTime = 2.5e-4
Duration = 1.0e-2
#
QoSUpstream(MeanBitRate[kBit/s]) = 32
QoSUpstream(PeakBitRate[kBit/s]) = 64
QoSUpstream(RelativeBurstDuration) = 0.5
QoSUpstream(BitRateType) = VBR
```

```

QoSUpstream(ConnectionType)           = CONNECTION_LESS
QoSUpstream(TimeRestriction)          = REALTIME_NOT_REQUIRED
QoSUpstream(ErrorProbability)         = 1.0e-5
QoSUpstream(PacketLossProbability)    = 1.0e-3
QoSUpstream(MaximumDelay)             = 1.0e-5
QoSUpstream(InterarrivalTimeVariance) = 1.0e-7
QoSUpstream(InterarrivalTimeCentralMoment3) = 1.0e-9
QoSUpstream(InterarrivalTimeCentralMoment4) = 1.0e-10
#
QoSDownstream(MeanBitRate[kBit/s])    = 32
QoSDownstream(PeakBitRate[kBit/s])    = 64
QoSDownstream(RelativeBurstDuration)  = 0.5
QoSDownstream(BitRateType)           = VBR
QoSDownstream(ConnectionType)        = CONNECTION_LESS
QoSDownstream(TimeRestriction)       = REALTIME_NOT_REQUIRED
QoSDownstream(ErrorProbability)       = 1.0e-5
QoSDownstream(PacketLossProbability)  = 1.0e-3
QoSDownstream(MaximumDelay)          = 1.0e-5
QoSDownstream(InterarrivalTimeVariance) = 1.0e-7
QoSDownstream(InterarrivalTimeCentralMoment3) = 1.0e-9
QoSDownstream(InterarrivalTimeCentralMoment4) = 1.0e-10
#
# =====
[DataSource9]
# =====
# (leads to AAL5)
#
DataType                             = TEXT_DATA
SourceStation                         = Station_1
SinkStation                           = Station_2
Layer3Protokoll                      = NETWORK
Layer3ProtokollNumber                 = 0
Type                                  = DATA_GENERATOR_NEG_EXP
TypePacketLengthGenerator             = DATA_GENERATOR_UNIFORM
MeanPacketSize[Byte]                  = 128
SetupTime                             = 2.2e-4
Duration                              = 1.0e-2
#
#
QoSUpstream(MeanBitRate[kBit/s])      = 5000
QoSUpstream(PeakBitRate[kBit/s])     = 7500
QoSUpstream(RelativeBurstDuration)    = 0.1
QoSUpstream(BitRateType)              = VBR
QoSUpstream(ConnectionType)          = CONNECTION_ORIENTED
QoSUpstream(TimeRestriction)         = REALTIME_NOT_REQUIRED
QoSUpstream(ErrorProbability)         = 1.0e-5
QoSUpstream(PacketLossProbability)    = 1.0e-2
QoSUpstream(MaximumDelay)            = 1.0e-5
QoSUpstream(InterarrivalTimeVariance) = 1.0e-7

```

```
QoSUpstream(InterarrivalTimeCentralMoment3)      = 1.0e-9
QoSUpstream(InterarrivalTimeCentralMoment4)      = 1.0e-10
#
QoSDownstream(MeanBitRate[kBit/s])                = 5000
QoSDownstream(PeakBitRate[kBit/s])                = 7500
QoSDownstream(RelativeBurstDuration)              = 0.1
QoSDownstream(BitRateType)                        = VBR
QoSDownstream(ConnectionType)                     = CONNECTION_ORIENTED
QoSDownstream(TimeRestriction)                    = REALTIME_NOT_REQUIRED
QoSDownstream(ErrorProbability)                   = 1.0e-5
QoSDownstream(PacketLossProbability)              = 1.0e-2
QoSDownstream(MaximumDelay)                       = 1.0e-5
QoSDownstream(InterarrivalTimeVariance)           = 1.0e-7
QoSDownstream(InterarrivalTimeCentralMoment3)     = 1.0e-9
QoSDownstream(InterarrivalTimeCentralMoment4)     = 1.0e-10
#
# =====
[DataSource10]
# =====
# (leads to AAL5)
#
DataType                                           = TEXT_DATA
SourceStation                                     = Station_1
SinkStation                                       = Station_2
Layer3Protokoll                                  = NETWORK
Layer3ProtokollNumber                            = 0
Type                                              = DATA_GENERATOR_NEG_EXP
TypePacketLengthGenerator                        = DATA_GENERATOR_CONST
MeanPacketSize[Byte]                             = 47
SetupTime                                         = 2.2e-4
Duration                                          = 1.0e-2
#
QoSUpstream(MeanBitRate[kBit/s])                 = 5000
QoSUpstream(PeakBitRate[kBit/s])                 = 7500
QoSUpstream(RelativeBurstDuration)                = 0.1
QoSUpstream(BitRateType)                         = VBR
QoSUpstream(ConnectionType)                      = CONNECTION_ORIENTED
QoSUpstream(TimeRestriction)                     = REALTIME_NOT_REQUIRED
QoSUpstream(ErrorProbability)                    = 1.0e-5
QoSUpstream(PacketLossProbability)                = 1.0e-2
QoSUpstream(MaximumDelay)                        = 1.0e-2
QoSUpstream(InterarrivalTimeVariance)             = 1.0e-7
QoSUpstream(InterarrivalTimeCentralMoment3)      = 1.0e-9
QoSUpstream(InterarrivalTimeCentralMoment4)      = 1.0e-10
#
QoSDownstream(MeanBitRate[kBit/s])                = 5000
QoSDownstream(PeakBitRate[kBit/s])                = 7500
QoSDownstream(RelativeBurstDuration)              = 0.1
QoSDownstream(BitRateType)                       = VBR
```

```

QoSDownstream(ConnectionType)           = CONNECTION_ORIENTED
QoSDownstream(TimeRestriction)          = REALTIME_NOT_REQUIRED
QoSDownstream(ErrorProbability)         = 1.0e-5
QoSDownstream(PacketLossProbability)    = 1.0e-2
QoSDownstream(MaximumDelay)             = 1.0e-2
QoSDownstream(InterarrivalTimeVariance) = 1.0e-7
QoSDownstream(InterarrivalTimeCentralMoment3) = 1.0e-9
QoSDownstream(InterarrivalTimeCentralMoment4) = 1.0e-10

# =====
[DataSource11]
# =====
# (leads to AAL5)
#
DataType                               = TEXT_DATA
SourceStation                           = Station_1
SinkStation                              = Station_2
Layer3Protokoll                          = NETWORK
Layer3ProtokollNumber                   = 0
Type                                     = DATA_GENERATOR_NEG_EXP
TypePacketLengthGenerator                = DATA_GENERATOR_NEG_EXP
MeanPacketSize[Byte]                    = 47
SetupTime                               = 2.2e-4
Duration                                 = 1.0e-2
#
# The QoS-Parameters
# for a AAL5
#
QoSUpstream(MeanBitRate[kBit/s])        = 10000
QoSUpstream(PeakBitRate[kBit/s])       = 15000
QoSUpstream(RelativeBurstDuration)      = 0.1
QoSUpstream(BitRateType)                = VBR
QoSUpstream(ConnectionType)             = CONNECTION_ORIENTED
QoSUpstream(TimeRestriction)            = REALTIME_NOT_REQUIRED
QoSUpstream(ErrorProbability)           = 1.0e-5
QoSUpstream(PacketLossProbability)      = 1.0e-2
QoSUpstream(MaximumDelay)               = 1.0e-2
QoSUpstream(InterarrivalTimeVariance)   = 1.0e-7
QoSUpstream(InterarrivalTimeCentralMoment3) = 1.0e-9
QoSUpstream(InterarrivalTimeCentralMoment4) = 1.0e-10
#
QoSDownstream(MeanBitRate[kBit/s])      = 10000
QoSDownstream(PeakBitRate[kBit/s])     = 15000
QoSDownstream(RelativeBurstDuration)    = 0.1
QoSDownstream(BitRateType)              = VBR
QoSDownstream(ConnectionType)           = CONNECTION_ORIENTED
QoSDownstream(TimeRestriction)          = REALTIME_NOT_REQUIRED
QoSDownstream(ErrorProbability)         = 1.0e-5
QoSDownstream(PacketLossProbability)    = 1.0e-2

```

```
QoSDownstream(MaximumDelay)           = 1.0e-2
QoSDownstream(InterarrivalTimeVariance) = 1.0e-7
QoSDownstream(InterarrivalTimeCentralMoment3) = 1.0e-9
QoSDownstream(InterarrivalTimeCentralMoment4) = 1.0e-10
```

D.1.2 Netzwerkkonfiguration

Auszug aus der Konfiguration der VC-Level-Switch „Switch-I“

Stationskonfiguration

```
#####
#                                     #
# Konfiguration der Station          #
# Switch                             #
#                                     #
# automatisch generiertes File      #
#       von netedit                  #
#                                     #
#####

[Position]
Horizontal           = 250
Vertikal             = 250

[PortAnzahl]
Anzahl               = 3

[Stations_Name]
Name                 = Switch-I

[Stations_Typ]
Typ                  = VC_SWITCH

[Koordinaten]
Laengengrad         = 0.0
Breitengrad         = 50.0

[Richtung]
Laengengrad         = OESTLICH
Breitengrad         = NOERDLICH

[Protokolle]
Protokolle           = PHYSICAL_CELL, ATM,
ATM_LAYER_MANAGEMENT, PHYSICAL_LAYER_MANAGEMENT, PLANE_MANAGEMENT

[ATM]
```

```
Instanz0           = ATM0.config
Anzahl            = 1

[ATM_LAYER_MANAGEMENT]
Instanz0          = ATM_LAYER_MANAGEMENT0.config
Anzahl           = 1

[PHYSICAL_CELL]
Instanz0          = PHYSICAL_CELL0.config
Instanz1          = PHYSICAL_CELL1.config
Instanz2          = PHYSICAL_CELL2.config
Anzahl           = 3

[PHYSICAL_LAYER_MANAGEMENT]
Instanz0          = PHYSICAL_LAYER_MANAGEMENT0.config
Anzahl           = 1

[PLANE_MANAGEMENT]
Instanz0          = PLANE_MANAGEMENT0.config
Anzahl           = 1
#
#
# =====
#
# The SAP-Definitions
# and Switching Functions
#
# =====
#
[SAPs]
#
Anzahl            = 5
SAP0              = SAP0.config
SAP1              = SAP1.config
SAP2              = SAP2.config
SAP3              = SAP3.config
SAP4 = SAP4.config
#
# -----
# The Switching Function on ATM-Layer-Level
# -----
#
[Switches]
Anzahl            = 1
Switch0          = Switch0.config
#
[Anzahl_der_Adressen]
Anzahl            = 1
#
```

```
[Stations_Adresse0]
Adressformat           = ISO-NSAP
NSAPSubFormat         = E.164
DCC                    = 0049
Vorwahl                = 561
Durchwahl              = 8046304
RD                     = 0000
AREA                   = 0000
ESI                    = 000000000000
SEL                    = 00
AA                     = 000000
DFI                    = 00
```

ATM-Layer-Konfiguration

```
#####
#                               #
# Konfiguration der Instanz     #
#                               #
# Version 1.0 02/02/1997 STP   #
#                               #
# automatisch generiertes File #
#       von netedit           #
#                               #
#####
```

```
[ProtokollName]
Name           = ATM
```

```
[InstanzNummer]
Nummer        = 0
```

```
[TimingSettings]
BitRate[kBit/s] = 149760
```

```
[InternalSettings]
VCCTyp        = Endpoint
```

```
#
#=====
# Some Points that can be enabled to show the events
# in the visualization tool of netedit
#=====
```

```
[MessPunkte]
Anzahl        = 0
```

```
[MessPunkt0]
MessPunktName = Switch_ATM0_ATM_DATA_REQUEST
```

```

TriggerEvent = ATM_DATA_REQUEST

[MessPunkt1]
MessPunktName = Switch_ATM0_PHY_DATA_INDICATION
TriggerEvent = PHY_DATA_INDICATION
#
#=====
#
#
#=====
[HardwareDescription]
#=====
#
# The common part of the parallel systems
#
BufferSize[kByte] = 10
BusWidth[Byte] = 4
Clock[MHz] = 200
CyclesBufferAccess = 1
LowestPriority = 1 # high priority is „0“
#
NumberOfServiceUnits = 1
TypeOfServiceUnits = CLASSIC_CPU_SYSTEM
#
# -----
[ServiceUnit0]
# -----
Number = 0
CyclesInputBufferAccess = 1
OutputBufferSize[kByte] = 10
HardwareType = MODELL_3
#
# Bus Timing
#
Clock[MHz] = 200
CyclesOutputBufferAccess = 1
CyclesRAMAccess = 2
#
# DMA-Settings
#
CyclesToSetupDMA = 4
DMATransferCycles = 1
#
# CPU-Ops
#
CyclesModifyIntWithDashOps = 4
CyclesModifyIntWithPointOps = 10
CyclesAddInt = 4
CyclesSubtractInt = 4

```

```
CyclesIncrementInt           = 2
CyclesDecrementInt          = 2
CyclesCompareInt            = 2
CyclesModIntByLogicOperations = 4
#
CyclesMultiplyInt           = 10
CyclesDivideInt             = 20
CyclesModulusInt            = 20
#
CyclesShift1Position        = 1
CyclesAllocateMemory        = 10
CyclesFreeMemory            = 10
#
```

Physical Layer-Konfiguration

```
#####
#                               #
# Konfiguration der Instanz     #
#                               #
# automatisch generiertes File #
#           von netedit        #
#                               #
#####
```

```
[ProtokollName]
Name = PHYSICAL_CELL
```

```
[InstanzNummer]
Nummer = 0
```

```
[PortNummer]
Nummer = 0
```

```
[MediumSettings]
BitRate[kBit/s] = 155520
BitErrorProbability = 1.0e-12
```

```
[OAMSettings]
OAMSpace = 32
```

```
[InternalSettings]
InterfaceTyp = ATM_UNI
```

```
#=====
[MessPunkte]
```

```

Anzahl = 0
#
[MessPunkt0]
MessPunktName = S1_PHYSICAL0_PHY_DATA_REQUEST
TriggerEvent = PHY_DATA_REQUEST
#
[MessPunkt1]
MessPunktName = S1_PHYSICAL0_RS_DATA_REQUEST
TriggerEvent = RS_DATA_REQUEST
#
[MessPunkt2]
MessPunktName = S1_PHYSICAL0_NET_DATA_REQUEST
TriggerEvent = NET_DATA_REQUEST
#
[MessPunkt3]
#
MessPunktName = S1_PHYSICAL0_TIC
TriggerEvent = PHY_TIME_TIC
#
#
#=====
[HardwareDescription]
#=====
#
#
# The common part of the parallel processors
#
BufferSize[kByte] = 10
BusWidth[Byte] = 1
Clock[MHz] = 100
CyclesBufferAccess = 1
#
#
NumberOfServiceUnits = 1
TypeOfServiceUnits = HARDWARE_PHYSICAL_CELL
#
# -----
[ServiceUnit0]
# -----
Number = 0
OutputBufferSize[kByte] = 10
HardwareType = MODELL_3
#
# Bus Timing
#
Clock[MHz] = 100
CyclesInputBufferAccess = 1
CyclesOutputBufferAccess = 1
CyclesRAMAccess = 2

```

```
#
# DMA-Settings
#
CyclesToSetupDMA           = 4
DMATransferCycles         = 1
#
# CPU-Ops
#
CyclesModifyIntWithDashOps = 2
CyclesModifyIntWithPointOps = 10
CyclesAddInt               = 2
CyclesSubtractInt         = 2
CyclesIncrementInt        = 1
CyclesDecrementInt        = 1
CyclesCompareInt          = 1
CyclesModIntByLogicOperations = 1
#
CyclesMultiplyInt         = 10
CyclesDivideInt           = 20
CyclesModulusInt          = 20
#
CyclesShift1Position      = 1
CyclesAllocateMemory      = 10
CyclesFreeMemory          = 10
#
#
```

SAP-Konfigurationsdatei

```
#####
#                               #
# Konfiguration eines Service #
# Access Points               #
#                               #
# Version 1.0 02/02/1997 STP #
#                               #
# automatisch generiertes File #
#       von netedit           #
#                               #
#####

# =====
# The SAP-Definitions
# The SAP between the AAL-Layer-entities and their adjacent entities
# =====
```

[Definition]

```

LowerLayerEntities      = SAAL/0,AAL1/0,AAL3_4/0,AAL5/0
HigherLayerEntities    = NETWORK/0,AAL_LAYER_MANAGEMENT/0
XBusWidth[Byte]        = 4
XClock[MHz]            = 100
XType                  = Bus

```

Auszug aus der Konfiguration einer B-TE „Station_1“

```

#####
#
# Konfiguration der Station #
#
# Version 1.0 02/02/1997 STP #
#
# automatisch generiertes File #
#       von netedit       #
#
#####

```

[Position]

```

Horizontal      = 250
Vertikal        = 250

```

[PortAnzahl]

```

Anzahl          = 1

```

[Stations_Name]

```

Name            = Station_2

```

[Stations_Typ]

```

Typ             = B_TE

```

[Koordinaten]

```

Laengengrad     = 1.0
Breitengrad     = 50.0

```

[Richtung]

```

Laengengrad     = OESTLICH
Breitengrad     = NOERDLICH

```

[Protokolle]

```

Protokolle      = PHYSICAL_CELL,
PHYSICAL_LAYER_MANAGEMENT, ATM,ATM_LAYER_MANAGEMENT, AAL1, AAL5, AAL3_4, SAAL,
AAL_LAYER_MANAGEMENT, PLANE_MANAGEMENT,NETWORK

```

[ATM]

```
Instanz0           = ATM0.config
Anzahl            = 1

[ATM_LAYER_MANAGEMENT]
Instanz0           = ATM_LAYER_MANAGEMENT0.config
Anzahl            = 1

[SAAL]
Instanz0           = SAAL0.config
Anzahl            = 1

[AAL5]
Instanz0           = AAL50.config
Anzahl            = 1

[AAL3_4]
Instanz0           = AAL3_40.config
Anzahl            = 1

[AAL1]
Instanz0           = AAL10.config
Anzahl            = 1

[NETWORK]
Instanz0           = NETWORK0.config
Anzahl            = 1

[AAL_LAYER_MANAGEMENT]
Instanz0           = AAL_LAYER_MANAGEMENT0.config
Anzahl            = 1

[PHYSICAL_LAYER_MANAGEMENT]
Instanz0           = PHYSICAL_LAYER_MANAGEMENT0.config
Anzahl            = 1

[PLANE_MANAGEMENT]
Instanz0           = PLANE_MANAGEMENT0.config
Anzahl            = 1

[PHYSICAL_CELL]
Instanz0           = PHYSICAL_CELLO.config
Anzahl            = 1

# =====
#
# The SAP-Definitions
#
# =====
```

```
[SAPs]
Anzahl                = 6
SAP0                  = SAP0.config
SAP1                  = SAP1.config
SAP2                  = SAP2.config
SAP3                  = SAP3.config
SAP4                  = SAP4.config
SAP5                  = SAP5.config

[Anzahl_der_Adressen]
Anzahl                = 1

[Stations_Adresse0]
Adressformat          = ISO-NSAP
NSAPSubFormat        = E.164
DCC                   = 0049
Vorwahl               = 561
Durchwahl             = 8046304
RD                    = 0000
AREA                  = 0000
ESI                   = 000000000000
SEL                   = 00
AA                    = 000000
DFI                   = 00
```

D.2 Simulationslauf 2

Die Anzahl der Konfigurationsdateien zum zweiten Netzwerk ist aufgrund der zahlreichen Netzknoten wesentlich höher, daher wird auf eine Aufstellung an dieser Stelle verzichtet.

Anhang E Makefiles

E.1 Oberstes Makefile

```
# ~/atm/src/makefile
#####
#
#           Makefile fuer ATM-Simulation
#
#
#           Stefan Pitz 18.12.1995
#
#####
#
#-----
#
# Macros
#
#-----
#-----
# Maschinenspezifische Macros
# Werden ueber Shell-Environement-Variablen importiert
#-----
# Allg. gueltige Macros
#-----
#
#-----
# the make to use
#-----
#
MAKE                = make
#
#-----
# the paths to use
#-----
#
BASEDIR              = /home/sim/atm
INCDIR               = $(BASEDIR)/include
LIBDIR               = $(BASEDIR)/lib/SunOS
SRCDIR               = $(BASEDIR)/src
ODIR                 = $(BASEDIR)/ofiles/SunOS
BINDIR               = $(BASEDIR)/bin/SunOS
#
#-----
# the libraries to link together
```

```

#=====
#
BASELIBS          = -lbasis -lsimtools++ -lm
HARDWARELIBS     = -lhw
PHYLIBS          = -lphycell -lphysdh
ATMLIBS          = -latm
AALLIBS          = -laal5 -lsaal -laal3_4 -laal1 -laal0
LAYER3LIBS       = -lnetwork
LIBS              = -llaeufe -lnet $(LAYER3LIBS) $(AALLIBS) $(ATMLIBS) $(PHYLIBS)
$(HARDWARELIBS) $(BASELIBS)
#
#=====
# the name of the executable
#=====
BASENAME          = z++
VERSION           = 16
NAME              = $(BASENAME).$(VERSION)
#
#=====
# the analyzing software
#=====
#
ANALYZE_PROG      = analyze++
ANALYZE_NAME      = $(ANALYZE_PROG).$(VERSION)
#
#=====
# the flags to use at compilation time
#=====
#
COMPILE_ONLY      = -c
OPTIMIZE_FOR_SPARC = -xcg92
FULL_WARN         = +w
OPTIMIZE_FULL     = -O4
DEBUG             = -g
SOURCE_BROWSER    = -sb
FLAGS             = $(OPTIMIZE_FOR_SPARC) $(FULL_WARN) $(SOURCE_BROWSER)
#
SPV_LOG_DEPTH_MAX = -DSECURE_SPV
SPV_LOG_DEPTH_MIN = -DMINLOG_SPV
#
PROTOCOL_DEPTH_MAX = -DPROTOKOLL_1 -DPROTOKOLL_2 -DPROTOKOLL_3 -DDEBUG
$(SPV_LOG_DEPTH_MAX)
PROTOCOL_DEPTH_3   = -DPROTOKOLL_1 -DPROTOKOLL_2 -DPROTOKOLL_3 $(SPV_LOG_DEPTH_MIN)
PROTOCOL_DEPTH_2   = -DPROTOKOLL_1 -DPROTOKOLL_2 $(SPV_LOG_DEPTH_MIN)
PROTOCOL_DEPTH_1   = -DPROTOKOLL_1 $(SPV_LOG_DEPTH_MIN)
PROTOCOL_DEPTH_MIN = -DMIN_PROTOKOLL
#
#
GRAPHIK_PROTOCOL_DEPTH_3 = -DGRAPHIK_PROTOKOLL_3 -DGRAPHIK_PROTOKOLL_2 -

```

```
DGRAPHIK_PROTOKOLL_1
GRAPHIK_PROTOCOL_DEPTH_2 = -DGRAPHIK_PROTOKOLL_2 -DGRAPHIK_PROTOKOLL_1
GRAPHIK_PROTOCOL_DEPTH_1 = -DGRAPHIK_PROTOKOLL_1
GRAPHIK_PROTOCOL_DEPTH_MIN =
#
#
#=====
# the flags to use at archiving time
#=====
#
AR = CC
ARCHIVE_WITH_TEMPLATES = -xar
ARFLAGS = $(ARCHIVE_WITH_TEMPLATES) -o
#
#-----
# The Header-Files to include
#-----
#
INCLUDES = $(INCDIR)/ATMIncludeAll.h
#
#-----
# The modules on this level
#-----
#
#
MODULE = $(ODIR)/init.o
ANALYZE_MODULE = $(ODIR)/analyze.o
#
#-----
# the libraries to link together
#-----
#
#
LIBRARIES = $(LIBDIR)/libsimtools++.a \
            $(LIBDIR)/libbasis.a \
            $(LIBDIR)/libhw.a \
            $(LIBDIR)/libnet.a \
            $(LIBDIR)/libphycell.a \
            $(LIBDIR)/libphysdh.a \
            $(LIBDIR)/libatm.a \
            $(LIBDIR)/libaal0.a \
            $(LIBDIR)/libaal1.a \
            $(LIBDIR)/libaal3_4.a \
            $(LIBDIR)/libaal5.a \
            $(LIBDIR)/libsaal.a \
            $(LIBDIR)/libnetwork.a \
            $(LIBDIR)/liblaeufer.a
#
LIBDESCRIPT = LIBSIMTOOLS \
```

```

LIBBASE \
LIBHARDWARE \
LIBNET \
LIBPHYCELL \
LIBPHYS DH \
LIBATM \
LIBAAL0 \
LIBAAL1 \
LIBAAL34 \
LIBAAL5 \
LIBSAAL \
LIBNETWORK \
LIBTRIAL

#
#-----
-
# subdirectories with further makefiles
#-----
-
#
SUBDIRS          =          simtools++ basis hw phycell physdh \
                        atm aal0 aal5 saal aal1 aal3_4 \
                        network net laeufe

#
#-----
--
# Show the success of the „make“-run
#-----
--
#
MAKESUCCESS      =          .makesuccess
#
#
#=====
# global rules
#=====
#
.FAILED:
                @echo „make:fail“ > $(MAKESUCCESS);

#
.DONE:
                @echo „make:success“ > $(MAKESUCCESS);

#
#-----
#
# The creation of the simulation software
#
#=====
#

```

```
#
#-----
# Create the program by linking all Libraries together with the modules of the highest level
#-----
#
#
$(BINDIR)/$(NAME): $(LIBDESCRIPT) $(MODULE)
                $(LINKER) $(LINKFLAGS) \
                -L$(LIBDIR) \
                -o $(BINDIR)/$(NAME) $(MODULE) $(LIBS)
#
#-----
# The dependencies for the modules on this level
#-----
#
# %-rules are only valid for SunOS (can't be used on „herkules“) !
#
$(ODIR)/%.o : $(SRCDIR)/%.cc
                $(COMPILER) $(COMPILE_ONLY) $(DEBUG) -D_PHYSDH_RPT_ATMEVENTS $(FLAGS)
$(PROTOCOL_DEPTH_MIN) -I$(INCDIR) -o $@ $<
#
#
#=====
#
#-----
# The dependency of the makefiles in the subdirectories
# Every library can be built with specific options and preprocessor-variables
#-----
#
#-----
# simtools++
# [ Module und Library libsimtools++.a ]
#-----
#
LIBSIMTOOLS:
    @$(MAKE) -f simtools++/makefile „SRCDIR=$(SRCDIR)/simtools++“\
                „INCDIR=$(INCDIR)“\
                „ODIR=$(ODIR)“\
                „LIBDIR=$(LIBDIR)“\
                „ZFLAGS=$(FLAGS) $(OPTIMIZE_FULL)\
                $(SPV_LOG_DEPTH_MIN)“;
#
#
#-----
# basis (the base modules for the simulation)
# [library libbasis.a ]
#-----
#
```

```

LIBBASE :
    @$(MAKE) -f basis/makefile „SRCDIR=$(SRCDIR)/basis“\
        „INCDIR=$(INCDIR)“\
        „ODIR=$(ODIR)“\
        „LIBDIR=$(LIBDIR)“\
        „ZFLAGS=$(FLAGS) $(OPTIMIZE_FULL)“;

#
#
#-----
-----
# Hardware (the hardware-models)
# [ Library libhardware.a ]
#-----
-----
#
LIBHARDWARE :
    @$(MAKE) -f hw/makefile „SRCDIR=$(SRCDIR)/hw“\
        „INCDIR=$(INCDIR)“\
        „ODIR=$(ODIR)“\
        „LIBDIR=$(LIBDIR)“\
        „ZFLAGS= $(FLAGS) $(DEBUG) $(PROTOCOL_DEPTH_MIN) \
        $(GRAPHIK_PROTOCOL_DEPTH_MIN)“;
#
# ZFLAGS= $(FLAGS) $(DEBUG) -DDEBUG_BUFFER
$(PROTOCOL_DEPTH_MIN)
#
#-----
--
# netzkonfiguration ASCII
# [ Library libnet.a ]
#-----
--
#
LIBNET:
    @$(MAKE) -f net/makefile „SRCDIR=$(SRCDIR)/net“\
        „INCDIR=$(INCDIR)“\
        „ODIR=$(ODIR)“\
        „LIBDIR=$(LIBDIR)“\
        „ZFLAGS= $(FLAGS) $(DEBUG) $(PROTOCOL_DEPTH_MIN) \
        $(GRAPHIK_PROTOCOL_DEPTH_MIN)“;

#
#-----
-
# Physical Layer Cell-Based
# [ Library libphycell.a ]
#-----
-
LIBPHYCELL :
    @$(MAKE) -f phycell/makefile „SRCDIR=$(SRCDIR)/phycell“\
        „INCDIR=$(INCDIR)“\

```

```
        „ODIR=$(ODIR) \"\  
        „LIBDIR=$(LIBDIR) \"\  
        „ZFLAGS=$(FLAGS) $(DEBUG) $(PROTOCOL_DEPTH_MIN) \" ;  
#  
#-----  
# Physical Layer SDH-Based  
# [ Library libphysdh.a ]  
#-----  
LIBPHYSDH :  
    @$(MAKE) -f physdh/makefile      „SRCDIR=$(SRCDIR)/physdh\"\  
        „INCDIR=$(INCDIR) \"\  
        „ODIR=$(ODIR) \"\  
        „LIBDIR=$(LIBDIR) \"\  
        „ZFLAGS= $(FLAGS) $(DEBUG) $(PROTOCOL_DEPTH_MIN) \  
        $(GRAPHIK_PROTOCOL_DEPTH_MIN) \" ;  
#  
#-----  
# ATM-Layer  
# [ Library libatm.a ]  
#-----  
LIBATM :  
    @$(MAKE) -f atm/makefile      „SRCDIR=$(SRCDIR)/atm\"\  
        „INCDIR=$(INCDIR) \"\  
        „ODIR=$(ODIR) \"\  
        „LIBDIR=$(LIBDIR) \"\  
        „ZFLAGS=$(FLAGS) $(DEBUG) $(SPV_LOG_DEPTH_MIN) \" ;  
#  
#-----  
# AAL0  
# [Library libaal0.a ]  
#-----  
LIBAAL0 :  
    @$(MAKE) -f aal0/makefile      „SRCDIR=$(SRCDIR)/aal0\"\  
        „INCDIR=$(INCDIR) \"\  
        „ODIR=$(ODIR) \"\  
        „LIBDIR=$(LIBDIR) \"\  
        „ZFLAGS= $(FLAGS) $(SPV_LOG_DEPTH_MIN) \" ;  
#  
#-----  
# AAL1  
#[ Library libaal1.a ]  
#-----  
LIBAAL1 :  
    @$(MAKE) -f aal1/makefile      „SRCDIR=$(SRCDIR)/aal1\"\  
        „INCDIR=$(INCDIR) \"\  
        „ODIR=$(ODIR) \"\  
        „LIBDIR=$(LIBDIR) \"\  
        „ZFLAGS= $(FLAGS) $(DEBUG) $(PROTOCOL_DEPTH_MIN) \" ;  
#-----
```

```

# AAL3/4
# [Library libaal3_4.a ]
#-----
#
LIBAAL34 :
    @$(MAKE) -f aal3_4/makefile      „SRCDIR=$(SRCDIR)/aal3_4\"
        „INCDIR=$(INCDIR)\"
        „ODIR=$(ODIR)\"
        „LIBDIR=$(LIBDIR)\"
        „ZFLAGS= $(FLAGS) $(PROTOCOL_DEPTH_MIN)\";
#
# ZFLAGS= $(FLAGS) -DAAL3_4_FEHLER -DAAL3_4_PROTOKOLL
$(SPV_LOG_DEPTH_MIN)\";
#
#-----
# AAL5
# [Library libaal5.a ]
#-----
LIBAAL5 :
    @$(MAKE) -f aal5/makefile      „SRCDIR=$(SRCDIR)/aal3_4\"
        „INCDIR=$(INCDIR)\"
        „ODIR=$(ODIR)\"
        „LIBDIR=$(LIBDIR)\"
        „ZFLAGS= $(FLAGS) $(PROTOCOL_DEPTH_MIN)\";
#
#-----
# SAAL
# [ libsaal.a ]
#-----
LIBSAAL :
    @$(MAKE) -f saal/makefile      „SRCDIR=$(SRCDIR)/saal\"
        „INCDIR=$(INCDIR)\"
        „ODIR=$(ODIR)\"
        „LIBDIR=$(LIBDIR)\"
        „ZFLAGS= $(FLAGS) $(DEBUG)

$(PROTOCOL_DEPTH_MIN)\";
#
#-----
-
# Networl-Layer Protocol
# [ libnetwork.a ]
#-----
-
LIBNETWORK :
    @$(MAKE) -f network/makefile    „SRCDIR=$(SRCDIR)/network\"
        „INCDIR=$(INCDIR)\"
        „ODIR=$(ODIR)\"
        „LIBDIR=$(LIBDIR)\"
        „ZFLAGS= $(FLAGS) $(DEBUG)

$(PROTOCOL_DEPTH_MIN) $(GRAPHIK_PROTOCOL_DEPTH_MIN)\";

```

```
#
#
# =====
#-----
# delete executable and log-Files
#-----
#
removebin:
    @echo „Loeschen des executables ...“;\
    rm -f $(BINDIR)/$(NAME);\
    echo „-----Beendet-----“
#
removeolog:
    @echo „Loeschen der log-Files ...“;\
    rm -f $(BINDIR)/FEHLER;\
    rm -f $(BINDIR)/PROTOKOLL;\
    rm -f $(BINDIR)/GRAFIK;\
    rm -f $(BINDIR)/*.log;\
    echo „-----Beendet-----“
#
#-----
# remove local o-files
#-----
#
clean:
    @echo „Loeschen der o-Files ...“;\
    for i in $(MODULE); \
    do \
        echo $$i; \
        rm -f $$i; \
    done; \
    echo „Loeschen der SourceBrowser-Datenbank ...“;\
    rm -rf .sb;\
    echo „-----Beendet-----“
#
#-----
# remove all o-Files (also the ones originating from the sources in the subdirectories)
#-----
#
allclean:
    @$(MAKE) clean;
    @$(MAKE) removebin;
    @$(MAKE) removeolog;
    @for i in $(SUBDIRS); \
    do \
        echo $$i; \
        $(MAKE) -f $$i/makefile „SRCDIR=$$i“ clean; \
    done;
#
```

```

#-----
# unconditionally remake all
#-----
all :
    $(MAKE) allclean;
    $(MAKE) $(BINDIR)/$(NAME);

#
#-----
# remake all libs
#-----
softinstall :
    @echo „Neuerstellen aller Bibliotheken ...“; \
    for i in $(SUBDIRS); \
    do \
        echo $$i; \
        make -f $$i/Makefile „SRCDIR=$$i“; \
    done; \

#-----
-
# Ask for the configuration of the system
#-----
-
system:
    @echo „\n\n >> Rechnerkonfiguration <<\n\n“; \
    echo „Host-System      : „$(HOSTSYSTEM); \
    echo „C++-Compiler     : „$(COMPILER); \
    echo „Compiler-Options : „$(COMPFLAGS); \
    echo „Linker-Options   : „$(LINKFLAGS); \
    echo „Quellen         : „$(SRCDIR); \
    echo „Module          : „$(ODIR); \
    echo „Binaerfiles     : „$(BINDIR); \
    echo „Libraries       : „$(LIBDIR);

#-----
-
# install a stable version on the global directory for all programmers
#-----
-
#
init:
    @if [ ! -d $(ODIR) ]; then mkdir -p $(ODIR);fi; \
    if [ ! -d $(BINDIR) ]; then mkdir -p $(BINDIR);fi; \
    if [ ! -d $(INCDIR) ]; then mkdir -p $(INCDIR);fi; \
    if [ ! -d $(LIBDIR) ]; then mkdir -p $(LIBDIR);fi; \
    if [ ! -d $(SRCDIR) ]; then mkdir -p $(SRCDIR);fi; \
    date > lastprint;
    echo „“ > .makesuccess;

#

```

```
#
install:
    @echo Copying Sourcefiles to /opt/SIMshare/SIMatm.$(VERSION) ...; \
    mkdir -p /opt/SIMshare/SIMatm.$(VERSION); \
    echo src;cp -r /home/sim/atm/src /opt/SIMshare/SIMatm.$(VERSION); \
    echo includes;cp -r /home/sim/atm/include /opt/SIMshare/SIMatm.$(VERSION); \
    echo Configurationfiles;cp -r /home/sim/atm/cfg /opt/SIMshare/SIMatm.$(VERSION); \
    mkdir /opt/SIMshare/SIMatm.$(VERSION)/ofiles; \
    mkdir /opt/SIMshare/SIMatm.$(VERSION)/ofiles/SunOS; \
    mkdir /opt/SIMshare/SIMatm.$(VERSION)/lib; \
    mkdir /opt/SIMshare/SIMatm.$(VERSION)/lib/SunOS; \
    mkdir /opt/SIMshare/SIMatm.$(VERSION)/bin; \
    mkdir /opt/SIMshare/SIMatm.$(VERSION)/bin/SunOS;

#
#
```

E.1.1 Beispiel für ein unterlagertes Makefile (Physical Layer)

```
#-----
#
# Makefile fuer SimTools-Modul
#
#           <phycell>
#
# Stefan Pitz 15.12.1995
#
#-----
#
#=====  
#
# Macros
#
#=====  
#
LIBNAME           =           $(LIBDIR)/libphycell.a
#
#-----
# INCLUDE-Files
#-----
#
INCLUDES=         $(INCDIR)/SimTools++.h \
                  $(INCDIR)/LayerProtocols.h \
                  $(INCDIR)/SimEvents.h \
                  $(INCDIR)/ProtocolEntity.h \
                  $(INCDIR)/ProtocolStack.h
#
#-----
# Modules in the directory
#-----
```

```
#
MODULE = $(SRCDIR)/phycell.cc\
         $(SRCDIR)/PHYSICAL_CELLEntity.cc\
         $(SRCDIR)/PL_MANEntity.cc

#
#-----
# Subdirectories with dependend Makefile's
#-----
#
SUBDIRS = NONE
#
#-----
# create the lib, if necessary use ranlib
#-----
#
$(LIBNAME) : $(ODIR)/phycell.o
             $(AR) $(ARFLAGS) $(LIBNAME) $(ODIR)/phycell.o
#
#-----
#
$(ODIR)/phycell.o : $(MODULE) $(INCLUDES)
                   $(COMPILER) -c $(ZFLAGS) $(LOCALFLAGS) -I$(INCDIR) -I$(SRCDIR) -o
$(ODIR)/phycell.o $(SRCDIR)/phycell.cc
#
#-----
# Remove local o-Files
#-----
#
clean:
        @echo „ *** Loeschen phycell... *** „;\
        for i in $(ODIR)/phycell.o; \
        do \
            echo $$i; \
            rm -f $$i; \
        done; \
        echo $(LIBNAME); \
        rm -f $(LIBNAME) \
        echo „Beendet.“

#
```

Anhang F Auswertung

F.1 Aufbau der Ergebnisdateien

Spalte 1 der Datei „VCCLog“ unterscheidet den VCC-Aufbau vom Abbau, die Bedeutung der Größen in den weiteren Spalten zeigen die beiden folgenden Tabellen:

F.1.1 VCC-Aufbau

Spalte	Bedeutung (Mean/PeakAllocation)	Bedeutung (Adaptive CAC)
1	SETUP (fest)	
2	UPSTREAM / DOWNSTREAM	
3	VCC-Idendifier	
4	Zeitpunkt VCC-Aufbau	
5	Maximale Bitrate	QoS _{LIM} [MBR]
6	Summe der mittleren Bitraten	QoS _{LIM} [PBR]
7	Summe der minimalen Bitraten	QoS _{LIM} [BER]
8	Summe der maximalen Bitraten	QoS _{LIM} [PLR]
9	unbenutzt	QoS _{LIM} [Delay]
10	unbenutzt	-
11	unbenutzt	QoS _{LIM} [CDV]
12	unbenutzt	QoS _{LIM} [CMD3]
13	unbenutzt	QoS _{LIM} [CMD4]
14	Aktuell benutzte QoS[MBR]	Aktuell benutzte QoS[MBR]
15	Aktuell benutzte QoS[PBR]	Aktuell benutzte QoS[PBR]

Tabelle 75 : Bedeutung der Spalten der SETUP-Zeilen der Ergebnisdatei

16	Aktuell benutzte QoS[BER]	Aktuell benutzte QoS[BER]
17	Aktuell benutzte QoS[PLR]	Aktuell benutzte QoS[PLR]
18	Aktuell benutzte QoS[Delay]	Aktuell benutzte QoS[Delay]
19	-	-
20	Aktuell benutzte QoS[CDV]	Aktuell benutzte QoS[CDV]
21	Aktuell benutzte QoS[CMD3]	Aktuell benutzte QoS[CMD3]
22	Aktuell benutzte QoS[CMD4]	Aktuell benutzte QoS[CMD4]
23	QoS _{REQ} [MBR]	QoS _{REQ} [MBR]
24	QoS _{REQ} [PBR]	QoS _{REQ} [PBR]
25	QoS _{REQ} [BER]	QoS _{REQ} [BER]
26	QoS _{REQ} [PLR]	QoS _{REQ} [PLR]
27	QoS _{REQ} [Delay]	QoS _{REQ} [Delay]
28	-	-
29	QoS _{REQ} [CDV]	QoS _{REQ} [CDV]
30	QoS _{REQ} [CMD3]	QoS _{REQ} [CMD3]
31	QoS _{REQ} [CMD4]	QoS _{REQ} [CMD4]
32	Call Accepted	Call Accepted
33	Accepted/Denied	Accepted/Denied

Tabelle 75 : Bedeutung der Spalten der SETUP-Zeilen der Ergebnisdatei

F.1.2 VCC-Abbau

Spalte	Bedeutung	Ort
1	RELEASE (fest)	
2	UPSTREAM/DOWNSTREAM	
3	VCC Identifier	
4	Zeitpunkt des Abbaus	
5	Anzahl empfangener Bits	R ^a
6	Anzahl verlorener Bits	R
7	Detektierte Bitfehler	R
8	Mittlere Bitrate	R
9	Spitzenbitrate	R
10	Bitfehlerrate	R
11	Paketverlustrate	R
12	Mittlere Paketverzögerung	R
13	-	R
14	Varianz der Zwischenankunftszeiten	R
15	3. Zentralmoment der “	R
16	4. Zentralmoment der “	R
17	Anzahl gesendeter Bits	S ^b
18	0	S
19	0	S
20	Mittlere Bitrate	S

Tabelle 76 : Bedeutung der Spalten der RELEASE-Zeilen der Ergebnisdatei

21	Spitzenbitrate	S
22	0	S
23	0	S
24	0	S
25	-	S
26	Varianz der Zwischenabgangszeiten	S
27	3. Zentralmoment “	S
28	4. Zentralmoment “	S
29	QoSREQ[MBR]	R
30	QoSREQ[PBR]	R
31	QoSREQ[BER]	R
32	QoSREQ[PLR]	R
33	QoSREQ[Delay]	R
34	-	R
35	QoSREQ[PDV]	R
36	QoSREQ[CMD3]	R
37	QoSREQ[CMD4]	R
38	QoSREQ[MBR]	S
39	QoSREQ[PBR]	S
40	QoSREQ[BER]	S
41	QoSREQ[PLR]	S
42	QoSREQ[Delay]	S

Tabelle 76 : Bedeutung der Spalten der RELEASE-Zeilen der Ergebnisdatei

43	-	S
44	QoSREQ[PDV]	S
45	QoSREQ[CMD3]	S
46	QoSREQ[CMD4]	S
47	QD ^c (QoS[MBR])	R
48	QD(QoS[PBR])	R
49	QD(QoS[BER])	R
50	QD(QoS[PLR])	R
51	QD(QoS[Delay])	R
52	-	R
53	QD(QoS[PDV])	S
54	QD(QoS[CMD3])	S
55	QD(QoS[CMD4])	R
56	QD(QoS[MBR])	R
57	QD(QoS[PBR])	S
58	QD(QoS[BER])	S
59	QD(QoS[PLR])	S
60	QD(QoS[Delay])	S
61	-	S
62	QD(QoS[PDV])	S
63	QD(QoS[CMD3])	S
64	QD(QoS[CMD4])	S

Tabelle 76 : Bedeutung der Spalten der RELEASE-Zeilen der Ergebnisdatei

65	QDRS ^d (QoS[MBR])	S
66	QDRS(QoS[PBR])	S
67	QDRS(QoS[BER])	S
68	QDRS(QoS[PLR])	S
69	QDRS(QoS[Delay])	R
70	-	R
71	QDRS(QoS[PDV])	R
72	QDRS(QoS[CMD3])	R
73	QDRS(QoS[CMD4])	R
74	Empfindlichkeit bzgl. MBR	R
75	Empfindlichkeit bzgl. PBR	R
76	Empfindlichkeit bzgl. BER	R
77	Empfindlichkeit bzgl. PLR	R
78	Empfindlichkeit bzgl. Delay	R
79	-	R
80	Empfindlichkeit bzgl. PDV	R
81	Empfindlichkeit bzgl. CMD3	R
82	Empfindlichkeit bzgl. CMD4	R
83	unbenutzt (DebugInformation)	
84	unbenutzt (DebugInformation)	
85	unbenutzt (DebugInformation)	
86	unbenutzt (DebugInformation)	

Tabelle 76 : Bedeutung der Spalten der RELEASE-Zeilen der Ergebnisdatei

87	unbenutzt (DebugInformation)	
88	unbenutzt (DebugInformation)	
89	unbenutzt (DebugInformation)	
90	unbenutzt (DebugInformation)	
91	unbenutzt (DebugInformation)	
92	unbenutzt (DebugInformation)	
93	unbenutzt (DebugInformation)	
94	unbenutzt (DebugInformation)	
95	unbenutzt (DebugInformation)	
96	unbenutzt (DebugInformation)	
97	unbenutzt (DebugInformation)	
98	unbenutzt (DebugInformation)	
99	unbenutzt (DebugInformation)	
100	unbenutzt (DebugInformation)	
101	Δ QoS[MBR]	S
102	Δ QoS[PBR]	S
103	Δ QoS[BER]	S
104	Δ QoS[PLR]	S
105	Δ QoS[Delay]	S
106	-	S
107	Δ QoS[PDV]	S
108	Δ QoS[CMD3]	S

Tabelle 76 : Bedeutung der Spalten der RELEASE-Zeilen der Ergebnisdatei

109	$\Delta\text{QoS}[\text{CMD4}]$	S
110	Mittlerer Qualitätsverlust	R
111	Mittlerer Qualitätsverlust inkl. Empfindlichkeiten	R
112	Gültige Stichprobe ?	R
113	-	
114	Last auf Physikalischem Port während des Bestehens des VCC	S
115	Mittlere Last auf Physikalischem Port bis zu diesem Zeitpunkt	S

Tabelle 76 : Bedeutung der Spalten der RELEASE-Zeilen der Ergebnisdatei

- a. Receiver
- b. Sender
- c. Quality Degradation (Qualitätsverlust)
- d. QD am Empfänger bezogen auf den Sender

F.2 Auswertungsskript

Das awk-Skript „auswertung.awk“ zur Auswertung der Ergebnisdatei „VCCLog“:

```
# -----  
# Liefert die Summe der Spalten 29-37  
# -----  
#  
BEGIN {  
  
    #  
    # Initialisierung des Referenzverkehrs  
    #  
  
    ArrayValRef[0] = 2600.0;  
    ArrayValRef[1] = 2600.0;      #1 MBit/s in cells (nur payload)  
    ArrayValRef[2] = 1.0e6;  
    ArrayValRef[3] = 1.0e8;  
    ArrayValRef[4] = 1.0e5;  
    ArrayValRef[5] = 1.0e4;  
    ArrayValRef[6] = 1.0e7;  
    ArrayValRef[7] = 1.0e9;  
    ArrayValRef[8] = 1.0e11;  
  
    #  
    # Vorbesetzung der Felder zur Berechnung  
    # des akkumulierten Wertes VCC eines  
    # Simulationslaufs getrennt nach Komponenten  
    # des QoS-Vektors  
    #  
  
    ArrayValAcc[0] = 0;  
    ArrayValAcc[1] = 0;  
    ArrayValAcc[2] = 0;  
    ArrayValAcc[3] = 0;  
    ArrayValAcc[4] = 0;  
    ArrayValAcc[5] = 0;  
    ArrayValAcc[6] = 0;  
    ArrayValAcc[7] = 0;  
    ArrayValAcc[8] = 0;  
  
    #  
    # Anzahl der insgesamt analysierten VCCs  
    #  
    NumOfVCC = 0;  
  
    #  
    # Anzahl erfolgreich aufgebauter  
    # (von CAC zugelassener) VCCs
```

```
#
num_of_succ_vcc=0;

#
# Anzahl nicht analysierter VCCs
# (Stichprobe zu klein)
#
num_of_stat_insecure=0

#
# Während des Laufes akkumulierter
# mittlerer Qualitätsverlust
#
AccLossSquare = 0;

#
# Während des Laufes akkumulierter
# mittlerer Wert
#
MeanQualityAccumulatedUpToNow = 0;

#
# Während des Laufes erreichte
# mittlere Auslastung der Links
#
OverAllLoad = 0;

}

# die ganze zeile wird ausgegeben
# wenn in Spalte 5 „STAT_INSECURE“, wurde keine
# Auswertung vorgenommen, da die Stichprobe
# nicht ausreichend war

if ( $5 != „STAT_INSECURE“ )
{

    num_of_succ_vcc += 1;
    NumOfVCC += 1;
    OverAllLoad = $115;

    # Füllen der Berechnungsfelder mit den
    # Werten aus der aktuellen Zeile des
    # Ergebnisfiles
    dummy = CreateArrays();

    # Berechnung des Wertes des aktuelle VCC
    dummy = CreateValueReached();
```

```
# Berechnung des mittleren Qualitätsverlustes
# der Daten des VCC
MeanQualityLossUsingSquare = GetMeanLoQ();

# der verbleibende Wert dieser Verbindung
#
RestValueOfThisVCCUsingSquare = GetMeanValue();

# akkumulierter Qualitätsverlust aller VCC
AccLossSquare += MeanQualityLossUsingSquare;

# VCCs, deren Qualität zu stark beeinträchtigt ist,
# werden von der Auswertung ausgeschlossen
if ( MeanQualityLossUsingSquare < 0.1 )
{
    MeanQualityAccumulatedUpToNow += GetMeanQualityQ();
}

# Ausgabe des Analyseergebnis

printf „\n%e\t%e\t%e\t%e\t%e\t%e\t%d\t%d\t%e“,\
        MeanQualityLossUsingSquare, \
        RestValueOfThisVCCUsingSquare,\
        MeanQualityAccumulatedUpToNow, \
        AccLossSquare, \
        num_of_succ_vcc, \
        num_of_stat_insecure, \
        Time,
        Load;
}
else
{
    # Zählen der nicht analysierten VCCs
    #print „Unsichere Statistik nicht ausgewertet „
    num_of_stat_insecure += 1
}
}

# Berechnung der Quadratischen Abweichung
function sqrdiff(x,x_ref)
{
    if ( x_ref != 0 )
    {
        if(x<x_ref)
        {
            return ( (1.0-(x/x_ref)) * (1.0-(x/x_ref)));
        }
    }
    else
    {
```



```
        return 0;
    }
}
return 0;
}
#
# Berechnung der mittleren Qualität
#
function GetMeanQuality(mean)
{
    mean = 0;
    for ( i=0;i<9;i++ )
    {
        mean += ArrayValQDQ[i];
    }
    return ( mean/9.0 );
}

#
# Berechnung des mittleren Qualitätsverlustes
#
function GetMeanLoQ(mean)
{
    mean = 0;
    for ( i=0;i<9;i++ )
    {
        mean += ArrayQD[i];
    }
    return ( mean/9.0 );
}

#
# Berechnung des mittleren Wertes eines VCC
#
function GetMeanValue(mean)
{
    mean = 0;

    for ( i=0;i<9;i++ )
    {
        mean += ArrayValQD[i];
    }
    return ( mean/9.0 );
}

function CreateValueReached()
{
    # Für jede Komponente komponente:
```

```
# (1.0 - sqrdiff(r,reqi)) * val/valref

for ( i=0;i<9;i++ )
{
  # Relativer Wert des VCC bezogen auf den Referenzverkehr
  ArrayVal[i] = ArrayRequiredQuality[i] / ArrayValRef[i];

  # Verlust an Qualität (Quality Degradation QD) jeder Komponente
  ArrayQD[i] = ( sqrdiff(ArrayReceiverReachedQuality[i],ArrayRequiredQuality[i]) \
    - sqrdiff(ArraySenderReachedQuality[i],ArrayRequiredQuality[i])) ;

  # Verbelebender Wert nach Einbeziehung der erlittenen QD
  ArrayValQDM[i] = (1.0 - ArrayQD[i]) * ArrayVal[i];

  # Verbelebender Wert nach Einbeziehung der Empfindlichkeit
  ArrayValQDMS[i] = ArrayValQDM[i] * ArraySens[i];

  # Der Beitrag des VCC zur akkumulierten Qualität
  ArrayValAcc[i] += ArrayValQDMS[i] ;

}

}

# Übernahme der Werte aus der aktuellen Zeile in die
# Berechnungsfelder

function CreateArrays()
{

  ArraySenderReachedQuality[0] = $20;
  ArraySenderReachedQuality[1] = $21;
  ArraySenderReachedQuality[2] = $22;
  ArraySenderReachedQuality[3] = $23;
  ArraySenderReachedQuality[4] = $24;
  ArraySenderReachedQuality[5] = $25;
  ArraySenderReachedQuality[6] = $26;
  ArraySenderReachedQuality[7] = $27;
  ArraySenderReachedQuality[8] = $28;

  ArrayReceiverReachedQuality[0] = $8;
  ArrayReceiverReachedQuality[1] = $9;
  ArrayReceiverReachedQuality[2] = $10;
  ArrayReceiverReachedQuality[3] = $11;
  ArrayReceiverReachedQuality[4] = $12;
  ArrayReceiverReachedQuality[5] = $13;
  ArrayReceiverReachedQuality[6] = $14;
```

```
ArrayReceiverReachedQuality[7] = $15;  
ArrayReceiverReachedQuality[8] = $16;
```

```
ArrayRequiredQuality[0] = $29;  
ArrayRequiredQuality[1] = $30;  
ArrayRequiredQuality[2] = $31;  
ArrayRequiredQuality[3] = $32;  
ArrayRequiredQuality[4] = $33;  
ArrayRequiredQuality[5] = $34;  
ArrayRequiredQuality[6] = $35;  
ArrayRequiredQuality[7] = $36;  
ArrayRequiredQuality[8] = $37;
```

```
ArraySens[0] = $74;  
ArraySens[1] = $75;  
ArraySens[2] = $76;  
ArraySens[3] = $77;  
ArraySens[4] = $78;  
ArraySens[5] = $79;  
ArraySens[6] = $80;  
ArraySens[7] = $81;  
ArraySens[8] = $82;
```

```
ArrayVal[0] = $29 / ArrayValRef[0];  
ArrayVal[1] = $30 / ArrayValRef[0];  
ArrayVal[2] = $31 / ArrayValRef[0];  
ArrayVal[3] = $32 / ArrayValRef[0];  
ArrayVal[4] = $33 / ArrayValRef[0];  
ArrayVal[5] = $34 / ArrayValRef[0];  
ArrayVal[6] = $35 / ArrayValRef[0];  
ArrayVal[7] = $36 / ArrayValRef[0];  
ArrayVal[8] = $37 / ArrayValRef[0];
```

```
ArrayQD[0] = $65;  
ArrayQD[1] = $66;  
ArrayQD[2] = $67;  
ArrayQD[3] = $68;  
ArrayQD[4] = $69;  
ArrayQD[5] = $70;  
ArrayQD[6] = $71;  
ArrayQD[7] = $72;  
ArrayQD[8] = $73;
```

```
ArrayValQD[0] = 0;  
ArrayValQD[1] = 0;  
ArrayValQD[2] = 0;
```

```
ArrayValQD[3] = 0;
ArrayValQD[4] = 0;
ArrayValQD[5] = 0;
ArrayValQD[6] = 0;
ArrayValQD[7] = 0;
ArrayValQD[8] = 0;

ArrayValQDS[0] = 0;
ArrayValQDS[1] = 0;
ArrayValQDS[2] = 0;
ArrayValQDS[3] = 0;
ArrayValQDS[4] = 0;
ArrayValQDS[5] = 0;
ArrayValQDS[6] = 0;
ArrayValQDS[7] = 0;
ArrayValQDS[8] = 0;
Tim          = $4
Load         = $114
return 0;
}

END {
    # print „Anzahl :“ NUM „ Summe : „ SUM
    # Ausgabe der berechneten Mittelwerte
    # über alle VCCs

    printf „\n%d\t%d\t%e\t%e\t%e\t%e\t%e“,\
        num_of_succ_vcc,\
        num_of_stat_insecure,\
        AccLossSquare,\
        MeanQualityAccumulatedUpToNow, \
        OverAllLoad >> „all“;

}
```

Aufrufsyntax : `awk -f auswertung.awk < VCCLog > result.`