

Universität Ulm  
Fakultät für Informatik



Punkt-zu-Punkt- und  
Mehrpunkt-basierende  
LAN-Integrationsstrategien  
für den digitalen Mobilfunkstandard  
DECT

Gerhard Partsch  
*Universität Ulm*

Nr. 97-06  
Ulmer Informatik-Berichte  
März 1997

# Inhaltsverzeichnis

## Zusammenfassung

Der europäische Mobilfunkstandard DECT<sup>1</sup> wird zur Zeit noch fast ausschließlich zur Sprachkommunikation eingesetzt. In dem vorliegenden Bericht werden Konzepte und prototypische Implementierungen vorgestellt, die DECT-Systeme auch für die Datenkommunikation und den LAN<sup>2</sup>-Betrieb nutzbar machen. Die wichtigsten Ergebnisse dieses Berichtes lassen sich in folgenden Punkten zusammenfassen:

- Basis für die in diesem Bericht vorgestellten DECT-basierenden Datenkommunikationslösungen sind Punkt-zu-Punkt- und Mehrpunkt-Verbindungen.
- Die entwickelten DECT/LAN-Netzwerktreiber unterscheiden sich bezüglich ihrer Software-Schnittstellen (zum Betriebssystem und zu den Transportprotokollen) nicht von klassischen Netzwerktreibern wie Ethernet oder Token-Ring. Daraus folgt, daß sich über die in diesem Bericht entwickelten DECT/LAN-Netzwerktreiber alle am Markt verfügbaren Netzwerkarchitekturen (z.B. Microsofts LAN-Manager, Novells Netware oder der TCP/IP-Protocolstack) und LAN-Applikationen ohne Modifikationen betreiben lassen, d.h. es ist volle Software-Kompatibilität gegeben
- Die Kopplung mehrerer DECT-basierender LANs untereinander und mit klassischen LAN-Lösungen wie z.B. Ethernet oder Token-Ring ist möglich, d.h. die in diesem Bericht vorgestellten DECT/LAN-Lösungen stellen keine Insellösung dar.

---

<sup>1</sup> Digital European Cordless Telephone

<sup>2</sup> Local Area Network

# Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>1</b>
1.1	Motivation - Aufgabenstellung	1
1.2	Abgrenzungen	1
1.3	Aufbau dieses Berichtes	2
<b>2</b>	<b>Vorüberlegungen zur DECT/LAN-Integration</b>	<b>3</b>
2.1	LAN-Merkmale	3
2.2	DECT- Merkmale	5
<b>3</b>	<b>DECT/LAN-Integrationsstrategien</b>	<b>7</b>
3.1	LAN-Emulationen über Punkt-zu-Punkt-Verbindungen	7
3.1.1	DECT/LAN-Koppelemente	7
3.1.2	Leistungsfähige Verbindungssteuerung	8
3.2	LAN-Integrationen über Mehrpunktschaltungen	10
3.2.1	Logische Mehrpunktschaltungen	10
3.2.2	Physikalische Mehrpunktschaltungen	11
3.2.2.1	Kollisionsbehaftete Mehrpunktschaltungen	11
3.2.2.2	Kollisionsfreie Mehrpunktschaltungen (Paketfilter)	12
3.2.2.3	Paketfilter- und Paketvermittlungsprinzipien	13
3.2.2.4	Durchsatzabschätzung für Paketfilter	16
<b>4</b>	<b>Konzepte zur Realisierung der DECT/LAN-Netzwerksoftware</b>	<b>19</b>
4.1	Vorüberlegungen	19
4.1.1	Die Einbindung der DECT/LAN-Software in das Funktionsschichtenmodell eines modernen Computers	19
4.1.2	Netzwerktreiber und Treiberschnittstellen	22
4.1.3	Multiprotokollschnittstellen	25
4.1.4	Die Mehrkanalkommunikation	26
4.2	Das Connection-Manager-Konzept	28
4.2.1	Die Architektur des Connection-Managers	28
4.2.2	Connection-Manager-Zusatzfunktionen	29
4.2.3	Exemplarische Implementierung des Connection-Managers	32
4.3	DECT/LAN-Netzwerktreiber	36
4.3.1	Hardware-Komponenten	36
4.3.2	Modulstruktur	37
4.4	DECT/LAN-Prototypenkonfigurationen	39
4.4.1	DECT/LAN-Treiberversion	39
4.4.2	Connection-Manager Version	40
<b>5</b>	<b>Zusammenfassung und Ausblick</b>	<b>43</b>
	<b>Literaturverzeichnis</b>	<b>45</b>

# Kapitel 1

## Einführung

### 1.1 Motivation - Aufgabenstellung

Prädestiniert für die digitale und schnelle Datenübertragung im lokalen Bereich sind die LANs, die ein hohes Datenverkehrsaufkommen bewältigen können (10-100 MBit/s) und aus der heutigen DV-Umgebung nicht mehr wegzudenken sind. LANs erfüllen zahlreiche Zielsetzungen und Nutzeffekte, wovon an dieser Stelle nur die wesentlichsten Punkte angeführt werden sollen:

- gemeinsame Nutzung qualitativ hochwertiger und teurer Betriebsmittel
- redundanzfreie und konsistente Datenhaltung (zentral oder verteilt)
- Erschließung neuer Kommunikationsformen (elektronische Post, Konferenzsysteme, Joint Editing and Viewing etc.)
- Verbesserung der Informationsversorgung (Erfüllung der Maxime: Die richtige Information, zur richtigen Zeit, am richtigen Ort)
- Integration von Einzelkomponenten zu einem Gesamtsystem (Corporate Networks)
- Erhöhung der Zuverlässigkeit, Wirtschaftlichkeit, Effizienz und Produktivität

Der Mobilfunkstandard DECT hingegen dient z.Zt. fast ausschließlich der Sprachvermittlung und bedient sich dabei der für diese Kommunikationsform typischen Punkt-zu-Punkt-Verbindungen. Doch diese Verbindungsart widerspricht in vielerlei Hinsicht den in klassischen LANs vorhandenen Mehrpunktverbindungen, weshalb Punkt-zu-Punkt basierende DECT/LAN-Integrationsansätze oftmals nicht die volle LAN-Funktionalität zur Verfügung stellen können. In dieser Arbeit werden deshalb u. a. auch mehrpunktbasierende DECT/LAN-Integrationsansätze vorgestellt.

### 1.2 Abgrenzungen

Es erscheint bereits an dieser Stelle wichtig darauf hinzuweisen, daß die nachfolgend vorgestellten DECT/LAN-Integrationslösungen keinen Ersatz für ein klassisches LAN darstellen. Sie sollen vielmehr überall dort eine wertvolle Alternative und Ergänzung zu herkömmlichen

PC-Vernetzungskonzepten sein, wo aus anwendungstechnischer, organisatorischer oder wirtschaftlicher Sicht der Einsatz sinnvoll erscheint<sup>1</sup>.

### 1.3 Aufbau dieses Berichtes

Um zwei unterschiedliche Techniken (in unserem Fall LANs und den Mobilfunkstandard DECT) möglichst homogen miteinander verschmelzen zu können, muß man die zu integrierenden Techniken sehr genau kennen, ihre wesentlichen Bestandteile erkennen und bei der Integration erhalten. Dies setzt ein grundlegendes Verständnis für die zu „verschmelzenden“ Techniken voraus. Zu diesem Zweck werden in **Kapitel 2** die für die DECT/LAN-Integration wesentlichen DECT- und LAN-Merkmale kurz zusammengefaßt.

In **Kapitel 3** werden Punkt-zu-Punkt- und mehrpunktbasierende DECT/LAN-Integrationsansätze vorgestellt.

Um die breit eingeführten LAN-Applikationen und Kommunikationsprotokolle möglichst modifikationsfrei nutzbar zu machen, benötigt man Endgeräte, die mit DECT/LAN-Netzwerktreibern ausgestattet sind, die ihrerseits wiederum die Standard-Netzwerkschnittstellen (NDIS, ODI etc.) einhalten. Aufbauend auf diesen Überlegungen wird im letzten Abschnitt von **Kapitel 4** ein exemplarischer DECT/LAN-Treiber implementiert.

**Kapitel 5** faßt die Ergebnisse dieses Berichtes zusammen.

---

<sup>1</sup> z.B. dann, wenn keine neuen Kabel verlegt werden können, weil man in historischen Gebäuden lebt.

# Kapitel 2

## Vorüberlegungen

Wie in der Einleitung zu diesem Bericht bereits festgestellt wurde, sollte man beim Versuch der homogenen „Verschmelzung“ zweier unterschiedlicher Techniken die zu integrierenden Techniken genau kennen, um die wesentlichen Bestandteile beider Techniken bei der Integrationslösung erhalten zu können. Aus diesem Grund werden nachfolgend die für die angestrebte DECT/LAN-Integration wesentlichen LAN- und DECT-Merkmale kurz vorgestellt.

### 2.1 Für die DECT/LAN-Integration wesentliche LAN-Merkmale

Die im Rahmen der geplanten DECT/LAN-Integration wichtigste LAN-Eigenschaft ist die Mehrpunkt- bzw. Broadcast-Fähigkeit des Übertragungsmediums. Auf dieser Eigenschaft basieren eine Vielzahl von LAN-Funktionen, so z.B. die Möglichkeit verfügbare Netzressourcen einfach abfragen oder gleichzeitig mit mehreren Partnern eine Kommunikationsbeziehung unterhalten zu können. Durch die Broadcast-Fähigkeit des Mediums läßt sich die Kommunikation in vielen Situationen wesentlich effizienter realisieren als bei nicht broadcast-fähigen Medien, da nur eine einzige physikalische Nachricht benötigt wird, um mehrere Empfänger zu erreichen. Als weitere Vorteile sind die Reduzierung der Netzwerklast und der Nachrichtengenerierungszeiten beim Sender zu nennen, da der Sendeprozess insgesamt nur eine Multicast-Operation durchlaufen muß und nur eine Nachricht über das Medium verschickt wird. In verteilten Systemen kann darüber hinaus die Parallelität erhöht werden, weil die Empfänger die Nachricht schneller und gleichzeitig erhalten<sup>1</sup>. Generell kann gesagt werden, daß der Broadcast ein in der LAN-Technik viel genutzter Kommunikationsmechanismus ist<sup>2</sup>, dessen wesentliche Vorteile in der einfachen und effizienten Veröffentlichung von Informationen und des einfachen und effizienten Suchens von Informationen liegen.

Jedes LAN stellt die Broadcast-Fähigkeit in irgendeiner Art und Weise zur Verfügung. Nicht broadcast-fähige Topologien, wie z.B. der aus lauter Punkt-zu-Punkt-Verbindungen zu-

---

<sup>1</sup> Die besondere Bedeutung der Broadcast- und Multicast-Kommunikation in verteilten Systemen wird z.B. beschreiben in [Wybr].

<sup>2</sup> manchmal auch zu intensiv genutzt – Überflutung des Netzes mit Broadcast-Nachrichten (Broadcast Storms).

sammengesetzte Ring, realisieren die Mehrpunktfähigkeit über die zyklische Informationsweiterleitung – Bildung logischer Mehrpunktverbindungen bzw. logischer Broadcasts<sup>1</sup>. **Bild 2.1-1** verdeutlicht die beiden in der LAN-Technik am häufigsten anzutreffenden Netzwerktopologien und die dabei benutzten Arten der Informationsausbreitung zur Realisierung der logischen bzw. physikalischen Broadcast-Fähigkeit.

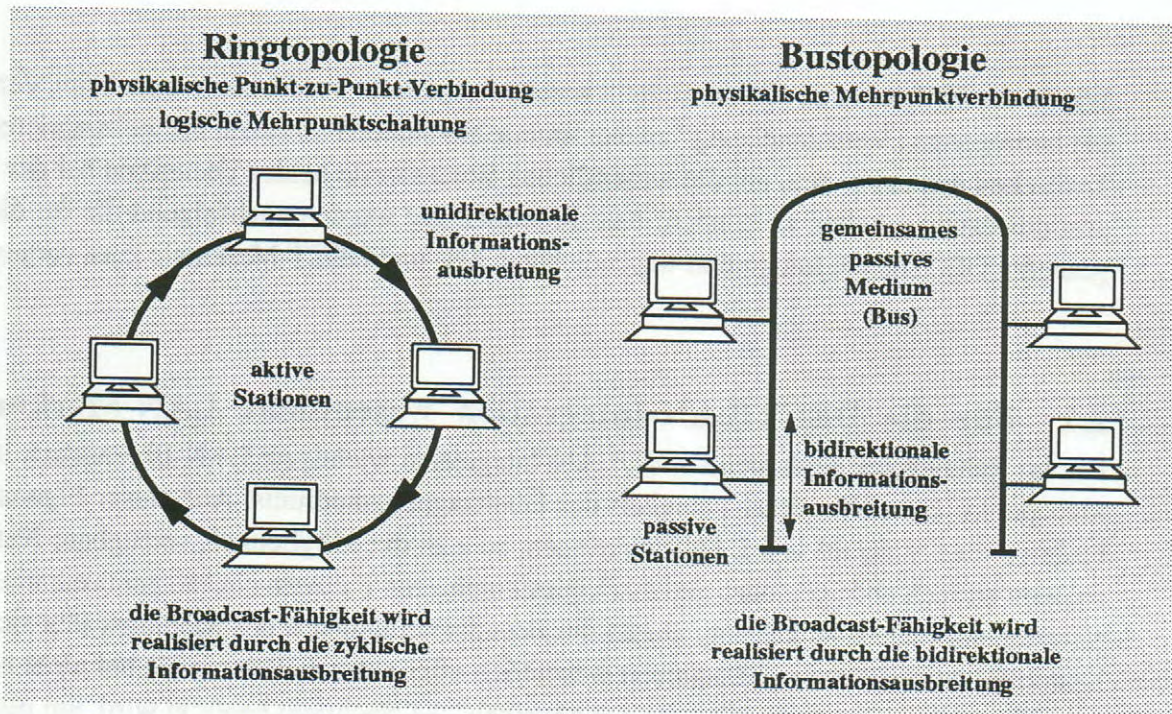


Bild 2.1-1: Realisierung der Broadcast-Fähigkeit bei Ring- und Busstrukturen

Um einen geregelten Zugriff auf das Übertragungsmedium sicherstellen zu können, halten sich alle LAN-Stationen an ein für alle Teilnehmer gleiches Mediumszugriffsverfahren (z.B. CSMA/CD oder Token-Verfahren). Aufbauend auf diesem gemeinsamen Zugriffsverfahren können die Stationen verschiedene Kommunikationsprotokolle (z.B. TCP/IP, DECnet etc.) und LAN-Applikationen über das gleiche LAN-Medium betreiben, ohne sich gegenseitig zu stören. Die zu übertragenden Daten werden paketisiert und mit einer Zieladresse versehen, so daß der Adressat das Datenpaket als für ihn bestimmt erkennen kann. Neben den Adreßinformationen werden jedem Datenpaket auch noch zusätzliche Protokollinformationen bei

<sup>1</sup> So werden z.B. in [Dalal], [Wall] und [AwEv] Verfahren vorgeschlagen, die in beliebig zusammenhängenden Graphen als Verbindungstopologie logische Broadcasts mit einer minimalen Anzahl physikalischer Nachrichten realisieren.

gefügt, wodurch zwischen den Kommunikationspartnern eine gesicherte Datenübertragung realisiert werden kann.

Die typischen Bitübertragungsraten der LANs liegen im Bereich zwischen 1-100 MBit/s und die Ausdehnungen beschränkt sich auf wenige Kilometer. Mittels geeigneter Strukturierungs- und Koppellelemente (Bridge, Router etc.) und unter Benutzung der öffentlichen Netze (Telefonnetz, Datex-P, ISDN etc.) können unternehmensweite Netzverbunde (Corporate Networks) aufgebaut werden. Die Bitübertragungsraten für den LAN-überschreitenden Verkehr entsprechen i.d.R. denen der öffentlichen Netze (ca. 2,4 – 64 kBit/s).

## 2.2 Für die DECT/LAN-Integration wesentliche DECT-Merkmale

DECT (Digital European Cordless Telephone) ist ein digitaler europäischer Standard für schnurlose Telefone. Die DECT-Spezifikation umfaßt mehrere Teile, die als „Common Interface Part 1-11“ beschrieben sind - [DECT]

Der DECT-Frequenzbereich (1880-1900 MHz) ist in 10 Trägerfrequenzen zu jeweils 1728 kHz aufgeteilt (FDMA = Frequency Division Multiple Access - wahlfreier Frequenzzugriff). Jeder Träger wiederum wird im TDMA-Verfahren betrieben (TDMA - Time Division Multiple Access - wahlfreier Zugriff auf Zeitschlitze). Die 24 Zeitschlitze eines jeden Trägers dienen zur Hälfte der Verbindung Basisstation zum Mobilteil und zur anderen Hälfte für die Verbindung Mobilteil zum Basisstation. Damit erhält man für ein DECT-System 120 Voll-duplexkanäle mit einer Einzelkanalbitrate von jeweils 32 kBit/s. Die Reichweite einer Basisstation ist, bei einer maximalen Abstrahlleistung von 0,25 W, auf einen Radius von ca. 200 m beschränkt.

Unter dem Aspekt der DECT/LAN-Integration wesentliche DECT-Merkmale sind:

- DECT-Systeme arbeiten prinzipiell leitungsvermittelt und Punkt-zu-Punkt-orientiert.
- Die Bitübertragungsrate pro Kanal beträgt 32 kBit/s.
- Die maximal mögliche Datenrate pro Träger beträgt 1152 kBit/s.
- Die Verbindungsaufbauzeiten liegen etwa bei ca. 0,2 - 0,5 Sekunden.
- Belegtsituationen sind möglich.
- Die Bitfahleraten schwanken sehr stark. Sie liegen in der Größenordnung um ca.  $10^{-2}$  bis  $10^{-3}$  und sind damit wesentlich höher als bei kabelgebundenen LANs (ca.  $10^{-6}$  bis  $10^{-12}$ ).



getriggert, wodurch zwischen den Kommunikationssystemen eine gesicherte Quasibindung  
realisiert werden kann.

Die typischen Bitübertragungsraten der LANs liegen im Bereich zwischen 1-100 Mbit/s und  
die Ausdehnungen beschränken sich auf wenige Kilometer. Mittels geeigneter Struktur-  
träger- und Kopplerelemente (Brücken, Router etc.) und unter Benützung der öffentlichen  
Netze (Telefonnetz, Datanet, ISDN etc.) können unimolekulare Netzwerke (Corporate  
Intranet) aufgebaut werden. Die Bitübertragungsraten für den LAN-Bereich sind  
Vielzahl entsprechen i.d.R. denen der öffentlichen Netze (ca. 2,4 - 64 Kbit/s).

### 2.3 Für die DECT/AM-Integration wesentliche DECT-Merkmale

DECT (Digital European Cordless Telephony) ist ein digitaler europäischer Standard für  
Kabellose Telefone. Die DECT-Spezifikation enthält mehrere Teile, die als „Common Inter-  
face Part 1-11“ bezeichnet sind - [DECT]

Der DECT-Frequenzbereich (1880-1900 MHz) ist in 10 Trägerfrequenzen zu jeweils 17,28  
kHz unterteilt (FDMA - Frequency Division Multiple Access - wahlweise Frequenzwahl).  
Jeder Träger wiederum wird im TDMA-Verfahren betrieben (TDMA - Time Division Mul-  
tiple Access - wahlweise Zugriff auf Zeitintervalle). Die 24 Zeitintervalle eines jeden Trägers  
teilen zur Hälfte der Verbindung Basisstation zum Mobilteil und zur anderen Hälfte für die  
Verbindung Mobilteil zum Basisstation. Damit erhält man für ein DECT-System 120 Voll-  
duplexkanäle mit einer Kanalbreite von jeweils 32 kHz. Die Reichweite einer Basis-  
station ist bei einer maximalen Abstrahlleistung von 0,25 W, auf einen Radius von ca. 300  
m beschränkt.

Unter dem Aspekt der DECT/AM-Integration wesentliche DECT-Merkmale sind:

- DECT-Systeme arbeiten prinzipiell leitungsvermittelt und Punkt-zu-Punkt-orientiert.
- Die Bitübertragungsraten pro Kanal betragen 32 Kbit/s.
- Die maximal mögliche Datenrate pro Träger beträgt 1152 Kbit/s.
- Die Verbindungsintervallen liegen etwa bei ca. 0,2 - 0,3 Sekunden.
- Basisstationen sind mobil.
- Die Bitübertragungsraten schwanken sehr stark. Sie liegen in der Größenordnung um ca.  $10^4$   
bis  $10^5$  und sind damit wesentlich höher als bei kabelgebundenen LANs (ca.  $10^7$  bis  
 $10^{12}$ ).

## Kapitel 3

# DECT/LAN-Integrationsstrategien

### 3.1 LAN-Emulationen über Punkt-zu-Punkt-Verbindungen

Von LAN-Emulationen spricht man einerseits dann, wenn über die DECT/LAN-Lösung nicht alle LAN-Dienste realisiert werden können (z.B. nur File-Transfer oder Drucken über das Netz) oder andererseits die DECT/LAN-Lösung kein integraler Bestandteil des DECT-Systems ist.

#### 3.1.1 DECT/LAN-Koppelemente

Die einfachste Art LAN-Dienste über ein DECT-System anzubieten besteht darin, das DECT-System als reinen LAN-Zubringer zu nutzen. Das DECT-System fungiert also lediglich als „Verbindungskabel“ zwischen einem sogenannten DECT/LAN-Koppler und den Remote-Workstations. Wie in **Bild 3.1-1** dargestellt, ist das DECT/LAN-Koppelement LAN-seitig ausgestattet mit einer klassischen LAN-Karte (z.B. Ethernet oder Token-Ring) und DECT-seitig mit einer oder mehreren DECT-spezifischen Schnittstellenkarten.

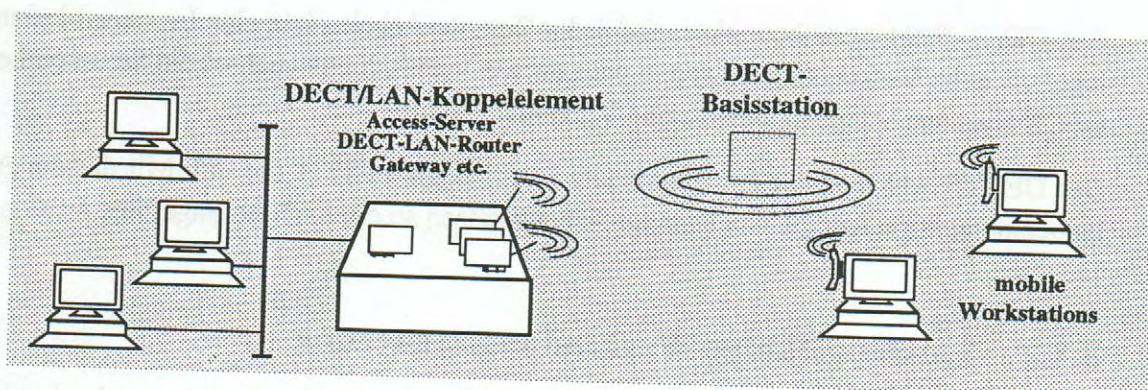


Bild 3.1-1: LAN-Emulationen über DECT/LAN-Koppelemente

Die Software des DECT/LAN-Kopplers bestimmt, auf welche Art und Weise Remote-Arbeitsplätze LAN-Dienste nutzen können. Hier unterscheidet man grundsätzlich zwischen Terminal-Teilnehmern und vollwertigen LAN-Teilnehmern.

Beim ferngesteuerten Terminal-Prozess arbeitet der Remote-Teilnehmer („intelligenter“ PC oder „dummes“ Terminal) ausschließlich im Terminal-Modus, bei dem, wie beim klassi-

schen Terminalbetrieb, lediglich Bildschirm- und Tastatureingaben übertragen werden. Die Abarbeitung der Applikationen (z.B. Textverarbeitung oder Tabellenkalkulation) und die Datenhaltung erfolgt im sogenannten Access-Server<sup>1</sup>. Großer Nachteil dieser Arbeitsweise ist, daß der Access-Server (entsprechend einem Mainframe) sehr leistungsfähig und entsprechend teuer ausgestattet sein muß.

Die zweite Realisierungsmöglichkeit besteht darin, den DECT/LAN-Koppler so zu programmieren, daß dieser sich als Bridge, Router oder Gateway zwischen LAN und Remote-Arbeitsplatz („intelligenter“ PC) verhält. Bei dieser Arbeitsweise fungiert der Remote-PC als vollwertiger LAN-Teilnehmer. Die Netzapplikationen und Kommunikationsprotokolle werden im Remote-Arbeitsplatz abgearbeitet, transferiert werden mit Adressen und Protokollinformationen versehene Datenpakete. Da die Remote-Arbeitsplätze über vollduplexfähige Punkt-zu-Punkt-Verbindungen an den DECT/LAN-Koppler angeschlossen sind, kann auf ein Mediumszugriffsverfahren verzichtet werden, was die Kommunikationsprotokolle vereinfacht und das Durchsatzverhalten verbessert.

Beiden Lösungen (ferngesteuerter Terminal-Prozeß und DECT/LAN-Koppler in Form einer Bridge etc.) haften die Nachteile an, daß je DECT/LAN-Koppler nur eine relativ kleine Anzahl von Remote-Arbeitsplätzen an das LAN herangeführt werden können und daß Belegtsituationen über Modifikationen in den Kommunikationsprotokollen abgefangen werden müssen. Gerade Belegtsituationen sind in klassischen LANs nicht existent und führen zu zahlreichen Komplikationen, da z.B. die Verzögerungswerte oftmals von Paket zu Paket stark unterschiedlich sein können.

Aus DECT-Sicht ist zu bemerken, daß jeder Remote-Arbeitsplatz eine Innenverbindung belegt (statische Verkehrsleistung) und permanente Verbindungswechsel die Steuereinheit des DECT-Systems belasten (dynamische Verkehrsleistung).

Der DECT/LAN-Koppler-Lösungsansatz ist prinzipiell DECT neutral. Mit diesem Ansatz kann eine geringe Anzahl von Remote-Arbeitsplätzen an ein LAN herangeführt werden.

### 3.1.2 Leistungsfähige Verbindungssteuerung

Die DECT/LAN-Emulation über eine leistungsfähige Verbindungssteuerung basiert im wesentlichen auf der Umsetzung der physikalischen Netzadressen (z.B. Ethernet-Adressen) auf Telefonnummern. Zu diesem Zweck wird von einer, sich in den Endgeräten befindlichen, leistungsfähigen oder „intelligenten“ Verbindungssteuerung vor jeder Transaktion eine physikalische Verbindung aufgebaut und am Ende der Transaktion oder bei größeren Sendepausen wieder abgebaut („short-hold-mode“). Im Gegensatz zu echten LANs erlaubt das Prinzip

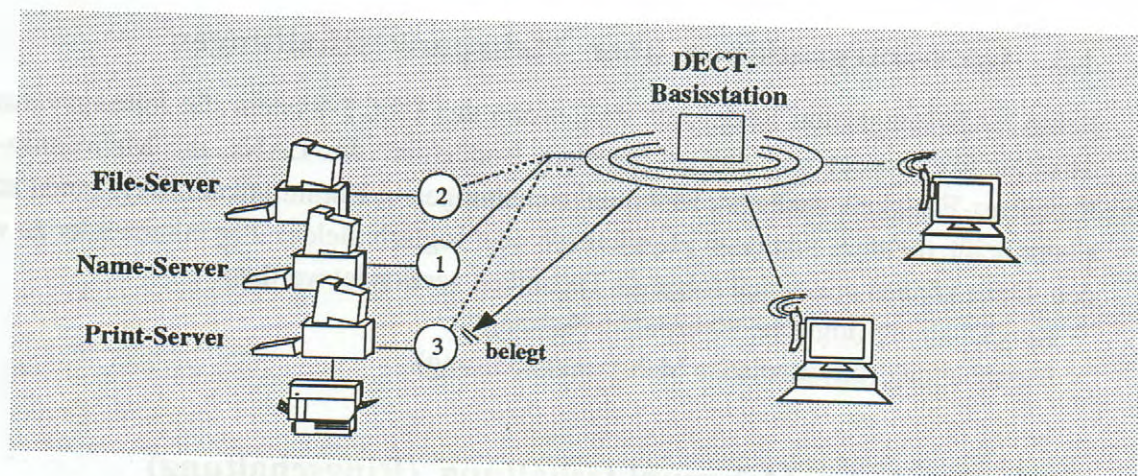
---

<sup>1</sup> nach Novell-Terminologie

der Leitungsvermittlung keine unnötige Aufrechterhaltung von Verbindungen, weshalb derartige LAN-Emulationen eine Strategie zur optimalen Schnittstellenausnutzung implementieren müssen, ohne den Arbeitsplatzrechner durch zu häufigen Verbindungsauf- und -abbau zu stark zu bremsen<sup>1</sup>.

Die fehlende Broadcast-Fähigkeit des Mediums wird mittels aufwendiger Mechanismen nachgebildet. So kann z.B. ein zentraler Name-Server eingerichtet werden, der eine aktuelle Liste der Netzteilnehmer und deren Telefonnummer führt. Eine andere Möglichkeit besteht darin, jeden Netzteilnehmer selbst eine derartige Liste verwalten zu lassen. Auch Kombinationen beider Varianten sind möglich.

**Bild 3.1-2** verdeutlicht die für diesen Lösungsansatz typische Arbeitsweise. Will z.B. ein Netzteilnehmer eine große Datei vom File-Server holen und am Laserdrucker ausgeben, so muß dieser zuerst den Name-Server anwählen (Vorgang 1), um von dort die Rufnummer des File-Servers und des Print-Servers zu erfahren. Anschließend wählt der Netzteilnehmer den File-Server an (Vorgang 2), überträgt die zu druckende Datei in voller Größe in seinen lokalen Speicher<sup>2</sup>, baut die Verbindung ab und wählt anschließend die Nummer des Print-Servers an (Vorgang 3), um die komplette Datei erneut über die Leitung zu übertragen. Während dieses eventuell mehrere Minuten dauernden Vorgangs ist der Print-Server belegt, weshalb andere Druckkandidaten die Leitung periodisch auf den Freizustand überprüfen müssen.



**Bild 3.1-2:** DECT/LAN-Emulation über eine leistungsfähige Verbindungssteuerung

- 1 Strategien der automatischen Verbindungssteuerung finden sich z.B. in [Froitz] und [Lupp].
- 2 Hauptspeicher oder, bei sehr großen Datenmengen, auf die Festplatte.

Bei einer geschalteten Punkt-zu-Punkt-Verbindung steht ein transparenter voll duplexfähiger 32 kBit/s Kanal zur Verfügung. Die Übertragung zwischen den Endgeräten kann sich einfacher Punkt-zu-Punkt-Protokolle<sup>1</sup> bedienen. Auch kann auf eine Paketisierung (Zerstückelung) der zu transferierenden Daten verzichtet werden, wenn nicht andere Gründe (z.B. Bitfehlerrate, Flußkontrolle etc.) dagegen sprechen.

Zusammenfassend kann man feststellen, daß die DECT/LAN-Integration über eine leistungsfähige Verbindungssteuerung häufige Verbindungswechsel erfordert, die die Vermittlungstechnik sehr stark belasten (dynamische Verkehrsleistung) und die LAN-Performance reduzieren. Auch wird pro Verbindung jeweils eine Koppelfeldinnenverbindung belegt (statische Verkehrsleistung) und natürlich muß auch dieser Lösungsansatz Belegtsituationen und die fehlende Broadcast-Fähigkeit über modifizierte Kommunikationsprotokolle abfangen können.

Durchaus geeignet erscheint dieser Lösungsansatz für die Vernetzung weniger Teilnehmer und für stream-orientierte Anwendungen, wie z.B. den File-Transfer, Electronic-Mail oder das Drucken über das Netz. Nicht oder nur bedingt geeignet erscheint dieser Lösungsansatz für den Anschluß vieler Netzteilnehmer, burst-artige Dienste (z.B. den Terminal-Betrieb) und für alle Dienste und Protokolle die die Broadcast-Fähigkeit des Mediums voraussetzen.

## 3.2 LAN-Integrationen über Mehrpunktschaltungen

Echte LAN-Integration findet nur statt, wenn das DECT-System die leitungsvermittelte Struktur überwindet und das „elementare Bedürfnis“ der LANs, die Mehrpunkt- bzw. Broadcast-Fähigkeit, realisiert. Mehrpunktschaltungen in Kommunikationssystemen zu realisieren heißt, mehrere Stationen physikalisch oder logisch gleichzeitig miteinander zu verbinden. Dazu können prinzipiell folgende Techniken verwendet werden:

- Ringschaltung (logische Mehrpunktschaltung)
- Busschaltung (physikalische Mehrpunktschaltung)

### 3.2.1 Logische Mehrpunktschaltung (Ringschaltung)

Die Ringschaltung besteht aus lauter physikalischen Punkt-zu-Punkt-Verbindungen, also genau aus den Elementen, die ein DECT-System sehr einfach schalten kann. Der Ring innerhalb des DECT-Systems wird gebildet, indem der Sendekanal von Station n zum Empfangskanal von Station n+1 wird, und Station n+1 seine Sendedaten wiederum an Station

---

<sup>1</sup> ohne Mediumszugriffsverfahren

n+2 weiterleitet, bis der Ring geschlossen ist – siehe **Bild 3.2-1**. Als Zugriffsverfahren kann ein angepaßtes Token-Ring-Protokoll verwendet werden.

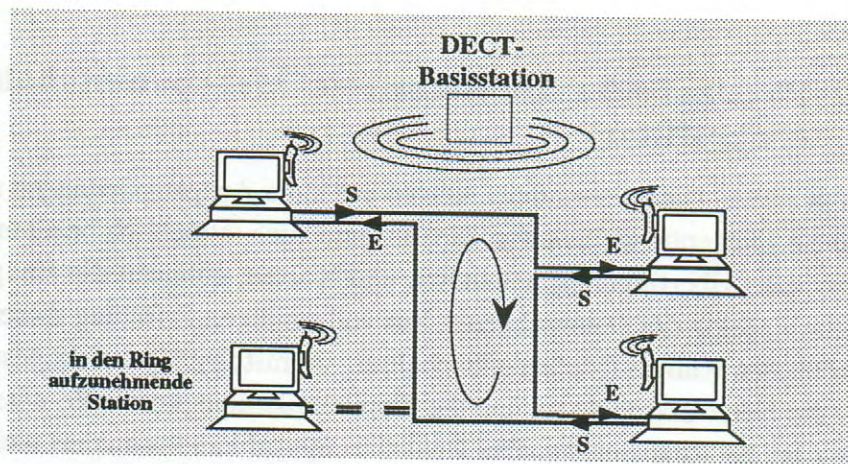


Bild 3.2-1: DECT/LAN-Integration über die Ringschaltung

Nachteil dieser Integrationslösung ist, daß pro LAN-Teilnehmer einen Vermittlungs-Slot (statische Verkehrsleistung) belegt wird, daß bei vielen Netzteilnehmern große Token-Umlaufzeiten (schlechtes Durchsatz- und Verzögerungsverhalten) entstehen und daß dieser Ring anfällig gegenüber Signalverfälschungen (Bitfehlerakkumulation) ist. Problematisch ist auch das Ringmanagement und die Fehlerbehandlung bei Token-Verlust, Ringinitialisierung usw. Um defekte Stationen aus dem Ring auskoppeln zu können, verbleibt dem DECT-System nur die Möglichkeit, die Verbindung zu allen Ringteilnehmern abzubauen, damit die korrekt arbeitenden Stationen sich daraufhin erneut in den Ring einwählen.

### 3.2.2 Physikalische Mehrpunktschaltung

Ein weiterer DECT/LAN-Integrationsansatz besteht darin, in einem DECT-System keine logische, sondern eine physikalische kollisionsbehaftete oder kollisionsfreie Mehrpunktschaltung zu realisieren.

#### 3.2.2.1 Physikalische kollisionsbehaftete Mehrpunktschaltung

Unter physikalischen kollisionsbehafteten Mehrpunktschaltungen versteht man Schaltungen, wie sie vom ALOHA oder Ethernet her allseits bekannt sind. Die erreichbaren Durchsatzraten liegen bei 18 % (pure ALOHA) bzw. 16 % (slotted ALOHA) – siehe Bild 3.2-5. Können die Stationen Kollisionen erkennen (Senden und Empfangen gleichzeitig) und ist die Signallaufzeit klein im Verhältnis zur Paketlänge, dann können in Kombination mit dem vom Ethernet her bekannten CSMA/CD-Zugriffsverfahren Durchsatzraten von ca. 90 % erreicht

werden. Auf die Realisierung physikalischer kollisionsbehafteter Mehrpunktschaltungen soll im Rahmen dieses Berichtes nicht eingegangen werden.

### 3.2.2.2 Physikalische kollisionsfreie Mehrpunktschaltung (Paketfilter)

Im Idealfall sollten Datenvermittlungsknoten auf ihren Ausgangsleitungen nur solche Datenpakete an die Endgeräte weiterleiten, die für diese auch tatsächlich bestimmt sind. Diese Arbeitsweise spart Übertragungskapazitäten und Kosten, entlastet die Endgeräte und erfüllt wesentliche Aspekte des Datenschutzes. Zur vollständigen Erfüllung dieser Forderung müssen die zentralen Knoten (Mehrpunktschaltungen) mit Intelligenz ausgestattet werden und viel über die Netzwerktopologie und die Kommunikationsprotokolle wissen, entsprechend aufwendig ist die Realisierung. Der Realisierungsaufwand reduziert sich allerdings ganz erheblich, wenn die Forderung dahingehend abgeschwächt wird, daß nur kollisionsfreie Pakete weitergeleitet werden. Zur Erfüllung dieser Anforderungen müssen die Mehrpunktschaltungen paketorientiert arbeiten, d.h. die in der Mehrpunktschaltung auszuführenden Operationen wie Empfangen, Weiterleiten oder Verwerfen beziehen sich stets auf ganze Paketeinheiten. Hervorzuheben ist, daß diese Arbeitsweise in der Mehrpunktschaltungen keinerlei Kenntnis über die verwendeten Protokolle oder die Netzwerktopologie erfordert und infolgedessen i.d.R. sehr einfach realisiert werden kann.

Bild 3.2-2 verdeutlicht den Kollisionsverhinderungsmechanismus einer paketorientierten Mehrpunktschaltung.

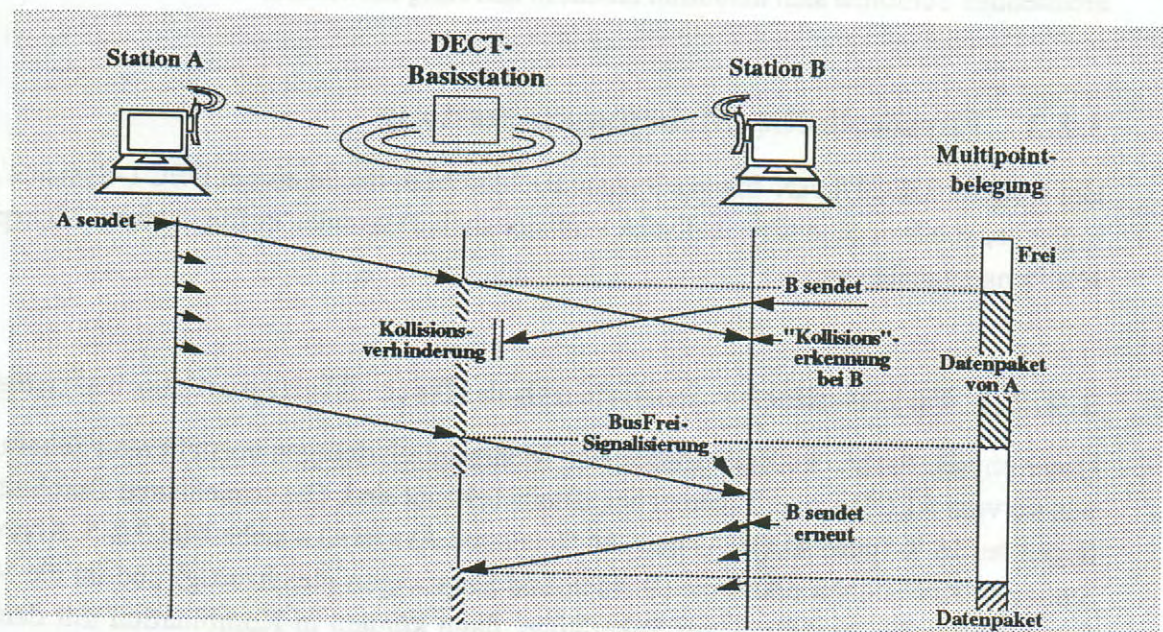


Bild 3.2-2: Kollisionsverhindernde Mehrpunktschaltung

Der 1. Sendeversuch von Station B führt durch eine Kollisionsverhinderungsmaßnahme in der Mehrpunktschaltung zu keiner Kollision mit dem Datenpaket von Station A. Für die DECT/LAN-Teilnehmer selbst ist dieser Vorgang transparent. So „glaubt“ z.B. Teilnehmer B bei seinem 1. Sendeversuch an eine Kollision, obwohl das Datenpaket von Station A kollisionsfrei übermittelt wird<sup>1</sup>.

Wegen der paketfilternden Arbeitsweise werden die paketorientierten Mehrpunktschaltungen nachfolgend als „Paketfilter“ bezeichnet.

### 3.2.2.3 Paketfilter- und Paketvermittlungsprinzipien

Die bisher eingesetzten paketorientierten Datenvermittlungsknoten arbeiten auf der ISO/OSI-Schicht 2 oder darüber (Bridge, Router oder Gateway) und benötigen zur Realisierung ihrer Aufgaben viel Intelligenz (Hard- und Software). Der Paketfilter ist funktional gesehen zwischen Schicht 1 und 2 anzusiedeln und kommt, wie später noch gezeigt wird, ohne aufwendige Intelligenz aus. Trotzdem läßt sich damit bei gleichzeitig sehr einfachen Mediumszugriffsverfahren das Durchsatz- und Verzögerungsverhalten eines DECT/LAN-Netzes wesentlich verbessern<sup>2</sup>.

Bild 3.2-3 verdeutlicht die unterschiedliche Arbeitsweise eines Paketfilters und eines Paketvermittlers. Beiden Varianten gemeinsam ist, daß sie, wie ihr Name schon andeutet, paketorientiert arbeiten, d.h. die Operationen beider Varianten beziehen sich auf komplette Pakete<sup>3</sup>.

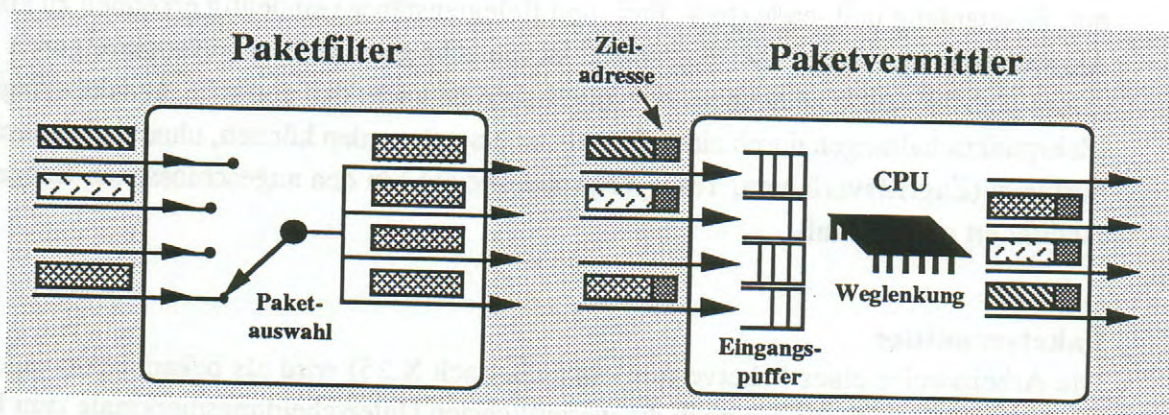


Bild 3.2-3: Paketfilter und Paketvermittler

- 1 Es versteht sich von selbst, daß Station B bei der „Kollisionserkennung“ das nicht kollidierte Datenpaket von Station A natürlich nicht ignoriert, sondern empfängt und weiterverarbeitet.
- 2 siehe Abschnitt „Durchsatzabschätzung für kollisionsverhindernde Mehrpunktschaltungen“
- 3 Das bedeutet natürlich nicht, daß intern nicht auch einzelne Bytes verarbeitet werden!



### **Paketfilter**

Die Aufgabe des Paketfilters ist es, aus dem parallel ankommenden Paketstrom jeweils ein Paket auszuwählen (ohne dabei den Paketinhalt zu interpretieren) und das ausgewählte Paket als Echo an alle Mehrpunktteilnehmer zurückzusenden. Wie bei den signalüberlagernden Mehrpunktschaltungen vergleichen die Teilnehmer ihre Sende- und Echodaten und können so feststellen, daß eventuell andere Informationen zurückkommen, als sie selbst ausgesandt haben. Nachdem von  $n$  am Paketfilter parallel ankommenden Paketen  $n-1$  verworfen werden, sieht es für  $n-1$  Endgeräte so aus, als hätten sich Signalverfälschungen ergeben, tatsächlich aber ist das Medium kollisions- bzw. signalüberlagerungsfrei, da sich ein Paket gegenüber allen anderen durchsetzt.

Trotz der paketerorientierten Arbeitsweise haben die Paketfilter, bedingt durch die Art ihrer Realisierung, i.d.R. keine Möglichkeit, mehrere Bytes oder gar Pakete zwischenzuspeichern. Jedes ankommende Byte muß entweder sofort weitergeleitet oder verworfen werden. Die Entscheidung, ob ein Byte weitergeleitet oder verworfen wird, hängt nicht vom Byte-Inhalt ab, sondern davon, ob das gesamte Paket weitergeleitet oder verworfen wird. Es werden also stets entweder alle Zeichen eines Pakets weitergeleitet oder verworfen. Verworfen Pakete werden vom Absender als Kollisionen erkannt und erneut übertragen.

Am Paketfilteransatz zu bemängeln ist, daß der Empfangskanal eines jeden Mehrpunktteilnehmers mit den gleichen Paketdaten „geflutet“ wird und dadurch nicht optimal, d.h. selektiv, genutzt wird. Der große Vorteil dieses Ansatzes ist, daß zur Erfüllung der reinen Paketfilterfunktion keine aufwendige Logik erforderlich ist, da der Paketfilter nichts über das verwendete Kommunikationsprotokoll oder den Paketinhalt wissen muß. Für ihn wichtig ist nur, Paketanfang und -ende (bzw. Frei- und Belegzustände) eindeutig erkennen zu können. Als weiterer wichtiger Paketfiltervorteil ist die sehr geringe Paketverzögerungszeit (kein „store-and-forward“) zu nennen. Hervorzuheben ist auch, daß einfache signalüberlagernde Mehrpunktschaltungen durch einen Paketfilter ersetzt werden können, ohne daß dadurch die Software (Zugriffsverfahren, Transportprotokolle etc.) in den angeschlossenen Endgeräten modifiziert werden muß.

### **Paketvermittler**

Die Arbeitsweise eines Paketvermittlers (z.B. nach X.25) wird als bekannt vorausgesetzt, weshalb an dieser Stelle nur noch die wesentlichsten Unterscheidungsmerkmale zum Paketfilter hervorgehoben werden sollen.

Auch der Paketvermittler wählt aus dem parallel ankommenden, nun aber gepufferten Paketstrom jeweils ein Paket aus und sendet dieses Paket, vermöge der jedem Paket beigefügten Protokollinformationen, selektiv an den oder die adressierten Mehrpunktteilnehmer. Die wichtigsten Unterscheidungsmerkmale zum Paketfilter sind, daß beim Paketvermittler

parallel ankommende Pakete nicht verworfen<sup>1</sup>, sondern im Empfangspuffer bis zur Weiterverarbeitung zwischengespeichert werden und daß zur selektiven Empfangskanalbehandlung eine Paketinterpretation und ein Protokollverständnis erforderlich ist. Diese Arbeitsweise erfordert einen erheblichen Soft- und Hardwareaufwand. Der große Vorteil des Paketvermittlers ist, daß neben zusätzlichen Managementfunktionen die zur Verfügung stehende Übertragungsrate effizienter genutzt werden kann.

## Paketfilterrealisierung

Bild 3.2-4 verdeutlicht die Arbeitsweise eines kollisionsverhindernden Paketfilters.

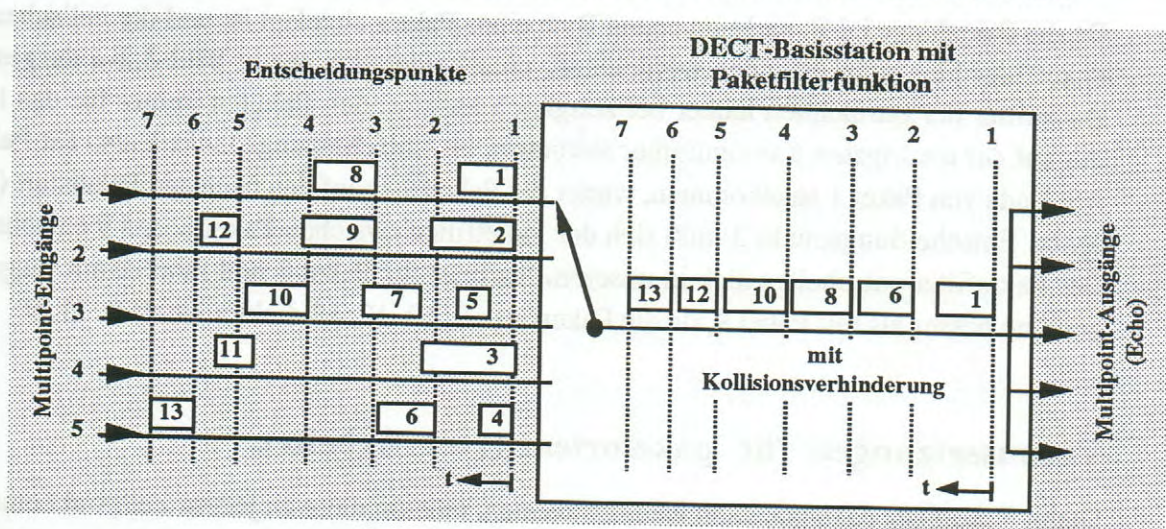


Bild 3.2-4: DECT-Basisstation mit kollisionsverhinderndem Paketfilter

Die Kollisionsverhinderung im zentralen Koppel- oder Echobildungspunkt (in der DECT-Basisstation). Bezüglich der Namensgebung ist zu beachten, daß Kollisionsverhinderung nicht bedeutet, daß jedes Datenpaket ohne Kollision durch den Paketfilter geschleust wird, sondern daß, bevor es zu einer Signalüberlagerung (Paketkollision) und dadurch zu einer Zerstörung aller an der Kollision beteiligten Pakete kommen kann, der Paketfilter einem der parallel eintreffenden Pakete die „Vorfahrt erteilt“ (als Echo an alle Mehrpunktteilnehmer transferiert) und alle anderen Pakete aus dem „Verkehr zieht“ (verwirft).

Der kollisionsverhindernde Paketfilter wählt bei Paketen mit versetzten Paketanfängen stets das zeitlich erste Paket zur Übertragung aus und verwirft alle zeitlich parallel einlaufenden Pakete. Bei gleichzeitig anstehenden Paketen, wie z.B. im Entscheidungspunkt 1, wählt der Paketfilter ein Paket aus und überträgt dieses im Echokanal an alle Mehrpunktteilnehmer. Die Paketauswahl kann sich dabei unterschiedlicher Methoden bedienen, z.B.:

<sup>1</sup> nur bei Bufferoverflow

- das Paket einer bestimmten Kanalnummer setzt sich durch (statische Priorisierung, z.B. Server gegenüber Clients bevorzugen)
- rotierende Priorisierung (Fairneßoptimierung, jeder Teilnehmer soll die gleiche Chancen beim Mediumszugriff haben)
- das längste Paket wird ausgewählt (einfacher Versuch der Effizienzoptimierung)
- „ein Paketfilter mit hellseherischen Fähigkeiten könnte vorausschauend bis in alle Unendlichkeit die optimalste Paketkette bestimmen und entsprechend seine Wahl treffen (in unserem Fall 3, 7, 8, 10, 12, 13 usw.)“

Da die Paketlänge i.d.R. nicht im ersten Byte eines Pakets abgelegt ist und der hellseherische Paketfilter hier nicht erfunden werden soll, entscheidet sich der in **Bild 3.2-4** dargestellte Paketfilter der Einfachheit halber bei zeitgleich anstehenden Paketen immer für das Paket, das auf der niedrigsten Kanalnummer ankommt, im Entscheidungspunkt 1 also für Paket 1. Am Ende von Paket 1 angekommen, wartet der Paketfilter auf den nächsten Paketstart (Paket 6). Im Entscheidungspunkt 3 muß sich der Paketfilter zwischen Paket 8 und 9 entscheiden. Der Paketfilter entscheidet sich in unserem Beispiel für Paket 8 und fährt damit insgesamt gesehen besser als mit Paket 9, da die Paketkette 8, 10, 12, 13 dichter ist als 9, 11, 9.

### Voraussetzungen für paketorientiertes Arbeiten

Damit Paketfilter paketorientiert arbeiten können, müssen diese zu jedem Zeitpunkt eindeutig entscheiden können, ob auf dem Kanal Daten übertragen werden, oder ob sich der Kanal im Freizustand (Idle) befindet. Um diese Entscheidung eindeutig treffen zu können, muß sichergestellt sein, daß der Freizustand des Kanals niemals in einem Datenpaket vorkommt.

Erfüllen kann man diese Bedingung z.B. dadurch, daß der IDLE-Zustand durch sieben<sup>1</sup> (oder mehr) Eins-Bits in Folge angezeigt wird und im Datenpaket dieser Zustand mit Hilfe des Bitstuffing-Mechanismus ausgeschlossen wird.

#### 3.2.2.4 Durchsatzabschätzung für kollisionsverhindernde Mehrpunktschaltungen (Paketfilter)

Zum Abschluß dieses Kapitels soll die durchsatzsteigernde Wirkung der kollisionsverhindernden Mehrpunktschaltungen bei Anwendung der Zugriffsverfahren ALOHA, nonpersistent CSMA und nonpersistent CSMA/CD aufgezeigt werden. In Kombination mit kollisi-

<sup>1</sup> Die Anzahl  $\geq 7$  ist deshalb wichtig, weil ein zum Datenpaket gehöriges Flag sechs Einsen in Folge aufweist. Um einen Idle-Zustand bei einer Byte-orientierten Arbeitsweise sicher erkennen zu können, müssen mindestens 13 Eins-Bits in Folge gesendet werden.

onsbehafteten Mehrpunktschaltungen ergeben sich nach [HamRei] folgende Durchsatzgleichungen:

$$\text{pure ALOHA:} \quad D = G e^{-2G} \quad (3.2-1)$$

$$\text{nonpersistent CSMA:} \quad D = \frac{G e^{-aG}}{G(1 + 2a) + e^{-aG}} \quad (3.2-2)^1$$

$$\text{nonpersistent CSMA/CD:} \quad D = \frac{G e^{-aG}}{G e^{-aG} + 2aG(1 - e^{-aG}) + (2 - e^{-aG})} \quad (3.2-3)^2$$

Kombiniert man die ALOHA-, CSMA- und CSMA/CD-Zugriffsverfahren mit kollisionsverhindernden Mehrpunktschaltungen, so benötigt man zur Durchsatzabschätzung den mittleren zeitlichen Abstand zwischen dem Ende einer laufenden und dem Anfang der darauffolgenden Paketübertragung, denn genau während dieser Zeit überträgt der Kanal keine Nutzdaten. Bei einer poissonverteilten Paketankunftsrate errechnet sich der Erwartungswert der Paketzwischenankunftszeit zu:

$$E = \frac{\text{Bezugszeiteinheit (Paketübertragungsdauer, Slot)}}{G} = \frac{1}{G}$$

Der Durchsatz für die ALOHA-, CSMA- und CSMA/CD-Zugriffsverfahren in Kombination mit kollisionsverhindernden Mehrpunktschaltungen berechnet sich damit wie folgt:

$$D = \frac{1}{1 + E} = \frac{1}{1 + \frac{1}{G}} = \frac{G}{G+1} \quad (3.2-4)$$

Auf den ersten Blick überraschend ist, daß sich in Kombination mit kollisionsverhindernden Mehrpunktschaltungen für alle drei Zugriffsverfahren die gleiche Durchsatzgleichung ergibt, d.h. daß die Hinzunahme der „Carrier Sense“- und „Collision Detection“-Mechanismen keine zusätzliche Durchsatzsteigerung ergibt. Verständlich wird dies, wenn man bedenkt, daß die Aufgabe der „Carrier Sense“- und „Collision Detection“-Mechanismen in der Kollisionsvermeidung liegt, doch diese Eigenschaft nicht zum Tragen kommt, da die Kollisionsvermeidung (Kollisionsverhinderung) bereits in den kollisionsverhindernden Mehrpunktschaltungen realisiert wird. Die Durchsatzgleichung (3.2-4) erhält man auch, wenn in den Gleichungen (3.2-2) und (3.2-3) von einer ideal schnellen Kollisionserkennung ausgegangen wird, d.h. der Konfliktparameter  $a$  gegen Null geht.

1 Konfliktparameter  $a = \frac{\text{max. Signallaufzeit}}{\text{Nachrichtenübertragungszeit}}$

2 Unter Vernachlässigung der Jam-Zeit, d.h. Jam-Zeit = 0.

Wie **Bild 3.2-5** verdeutlicht, liegt die Durchsatzkurve der kollisionsverhindernden Mehrpunktschaltung im gesamten Lastbereich oberhalb der Durchsatzkurven der signalüberlagernden Mehrpunktschaltungen und strebt mit steigender Last  $G$  asymptotisch gegen den Wert 1 (bzw. 100 %).

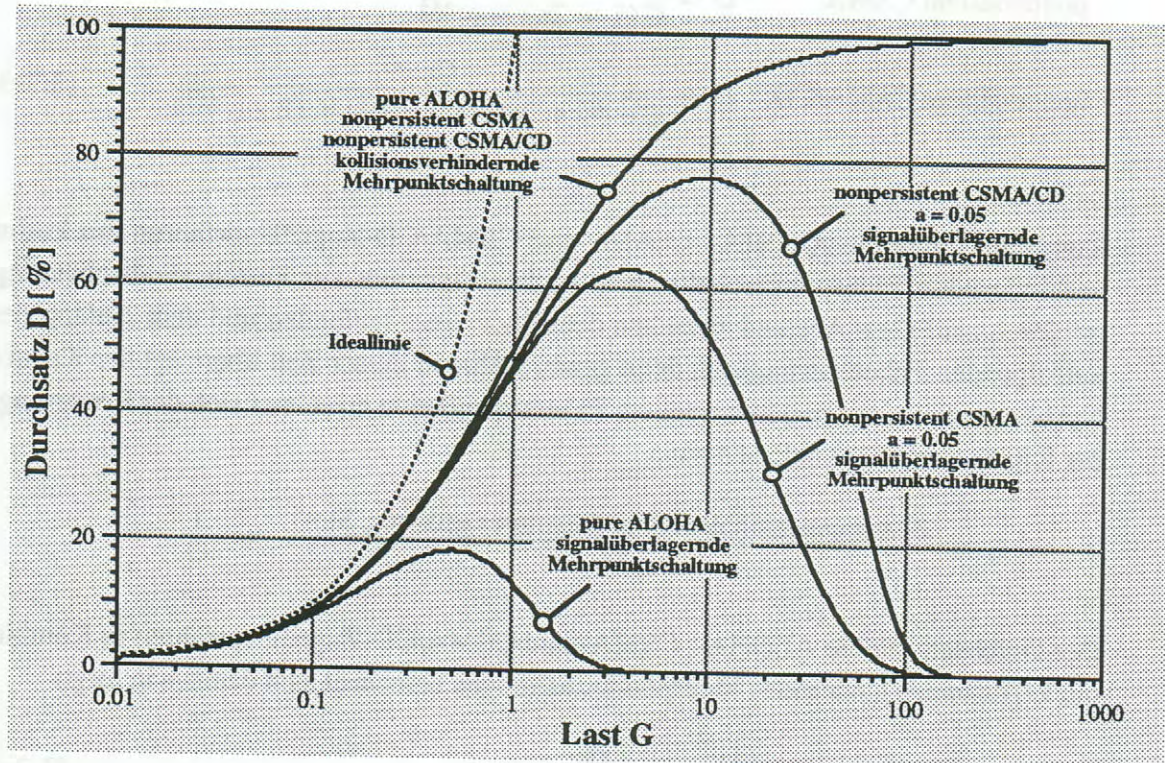


Bild 3.2-5: Durchsatzvergleich zwischen kollisionsbehafteten und kollisionsverhindernden Mehrpunktschaltungen

Kombiniert man die kollisionsverhindernde Mehrpunktschaltung mit der vom Ethernet her bekannten 1-persistenten CSMA/CD-Zugriffsmethode und setzt dabei das Backoff-Intervall zu Null<sup>1</sup>, so erhält man die in **Bild 3.2-5** dargestellte ideale Durchsatzkurve<sup>2</sup>.

<sup>1</sup> d.h. nach einer Kollision begeben sich die an der Kollision beteiligten Stationen nicht in den Backoff-Zustand, sondern suchen sofort erneut nach dem nächsten Mediumsfreizustand.

<sup>2</sup> Unter Vernachlässigung der Signallaufzeit.

## Kapitel 4

# Konzepte zur Realisierung der DECT/LAN-Netzwerksoftware

In diesem Kapitel wird der Versuch unternommen, aufbauend auf einer beliebigen DECT-Systemtopologie (Point-to-Point oder Multipoint), einen DECT/LAN-Treiber zu erstellen, der sich harmonisch in die Software-Architektur eines modernen Computers integrieren läßt, und volle Software-Kompatibilität verspricht.

### 4.1 Vorüberlegungen

Zunächst untersuchen wir, an welche Stelle die zu entwickelnde DECT/LAN-Netzwerksoftware in das Schichtenmodell eines modernen Computers einzubinden ist (welche Schichten zu substituieren sind) und welche Schnittstellen dabei einzuhalten sind.

#### 4.1.1 Die Einbindung der DECT/LAN-Software in das Funktionsschichtenmodell eines modernen Computers

Bild 4.1-1 zeigt das Funktionsschichtenmodell eines modernen Computers. Die unterste Schicht bildet dabei die Hardware, die über Gerätetreiber (Device Driver) mit dem Betriebssystem des Computers verknüpft ist. Die Aufgabe der Gerätetreiber ist es, die angeschlossenen Hardware-Einrichtungen zu bedienen und für die Computeranwender nutzbar zu machen. Dazu vergeben Gerätetreiber Hardware-Namen<sup>1</sup>, über die die Anwendungsprogramme die Hardware-Komponenten ansprechen (nutzen) können. I/O-Anforderungen der Applikationsprogramme werden i.d.R. nicht direkt an die Gerätetreiber oder die Hardware gestellt, sondern über das Betriebssystem an die Gerätetreiber weitergereicht<sup>2</sup>. Der Grund dafür liegt darin, daß Gerätetreiber nur „primitive“ Funktionen wie z.B. das Einlesen einzelner Platten-

<sup>1</sup> z.B. Plattenlaufwerke A:, B:, C:, Druckerausgänge LPT1:, LPT2: oder die seriellen Ports COM1: und COM2:

<sup>2</sup> Bei MS-DOS besteht für den Programmierer die Möglichkeit, an der Betriebssystemschnittstellen „vorbei“ zu programmieren und direkt auf die Hardware-Komponenten zuzugreifen.

sektoren ausführen, während Anwendungsprogramme nicht mit Plattensektoren operieren wollen, sondern z.B. eine ganze Datei Namens „C:DEMO.TXT“ zur Bearbeitung anfordern. Das Dateisystem und das BIOS (Basic Input Output System) – beides Bestandteile des Betriebssystems – wandeln die „High-Level“-Aufrufe der Applikationsprogramme um in treibergerechte „Low-Level“-Aufrufe.

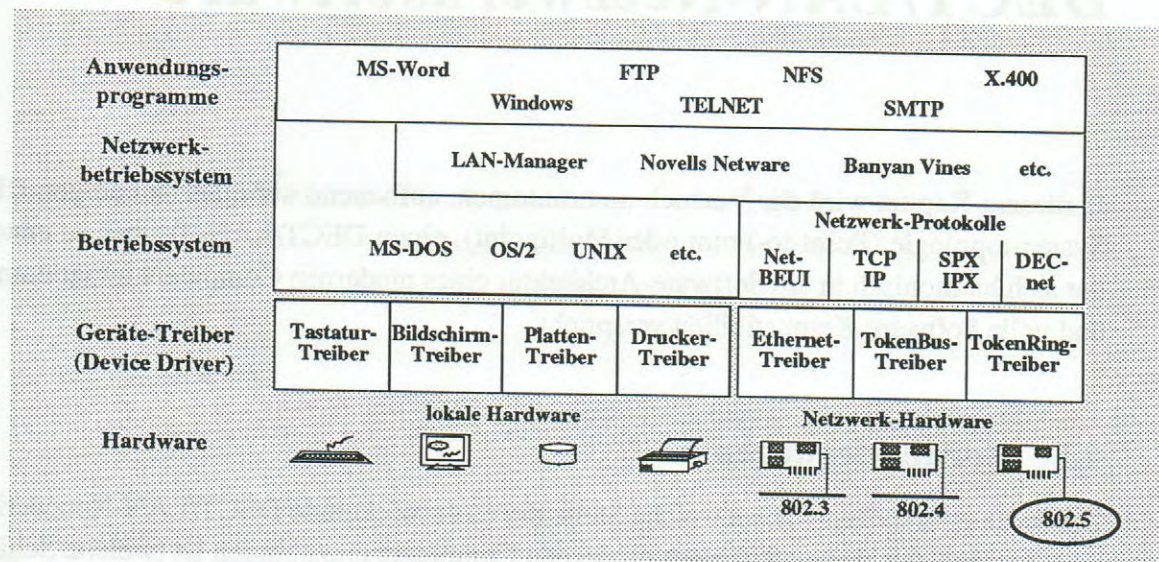


Bild 4.1-1: Funktionsschichtenmodell eines PCs

### Das Netzwerkbetriebssystem

Die zentrale Aufgabe der in den PCs ablaufenden Netzwerkbetriebssystem-Shell (NBS-Shell) ist die Umlenkung (Redirection) der I/O-Anforderungen, wenn anstatt auf eine lokale, auf eine entfernte Ressource zugegriffen wird. Im Beispiel von **Bild 4.1-2** wird der lokale Plattenzugriff „C:“ vom Redirector direkt an das lokale Betriebssystem und von dort an den Platten-Treiber weitergereicht, während der Zugriff auf das Netzlaufwerk „F:“ auf den File-Server umgelenkt wird<sup>1</sup>. Für die Anwendung selbst ist die Funktionsumlenkung nicht sichtbar. Diese Transparenz ist außerordentlich wichtig, da Applikationen unabhängig davon sein sollen, ob lokale oder entfernte Ressourcen benutzt werden. Es gibt allerdings auch Anwendungen, die an der Funktionsumlenkung vorbei direkt auf Funktionen des lokalen Betriebssystems, des Gerätetreibers oder direkt auf die Hardware zugreifen. Zu diesen sogenannten „nicht netzwerkfähigen“ Programmen zählen zahlreiche MS-DOS-Kommandos wie z.B. Chkdsk, Diskcopy, Format, Join, Sys oder Plattentestprogramme wie z.B. die von Norton

<sup>1</sup> Auf den Netzwerk-Servern läuft das eigentliche Netzwerkbetriebssystem.

Utilities oder PCTools. Die meisten PC-Applikationen tragen jedoch inzwischen das Prädikat „netzwerkfähig“.

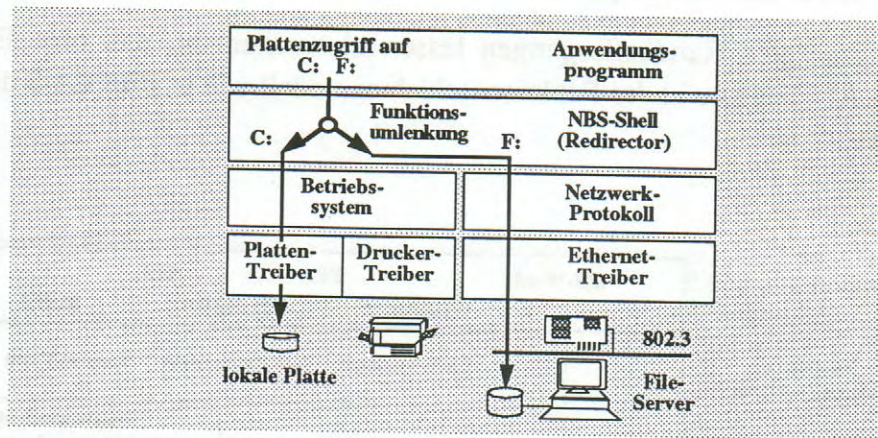


Bild 4.1-2: Funktionsumlenkung des Netzwerkbetriebssystems

### Die Bedeutung separater Gerätetreiber

Die offene Architektur moderner Computersysteme erfordert, daß die System-Software und die Applikationsprogramme unabhängig von der Hardware sein muß, auf der sie operiert. Dazu werden, wie bereits dargestellt, hardware-spezifische Informationen in lad- und konfigurierbare Gerätetreiber ausgelagert und gegenüber dem Betriebssystem mittels Schnittstellen abgegrenzt. Würde diese klare Abgrenzung nicht erfolgen, müßte der Betriebssystemkern bei jeder Hardware-Änderung individuell angepaßt werden. So aber ist es nur nötig, den Hardware-Treiber auszutauschen.

Es gilt festzuhalten, daß Device-Treiber i.d.R. die einzige Schnittstelle zur System-Hardware sind und I/O-Operationen der Anwendungsprogramme über das Betriebssystem an die Gerätetreiber weitergereicht werden.

### Die DECT/LAN-Software realisiert als Netzwerktreiber

Neue Entwicklungen stoßen i.d.R. nur dann auf Marktakzeptanz, wenn vorhandene Investitionen geschützt werden und es eine breite Palette von Anwendungen für die neue Entwicklung gibt. Das Ziel einer angestrebten DECT/LAN-Integration muß deshalb sein, alle netzwerkfähigen Applikationen modifikationsfrei zu unterstützen. Für den DECT/LAN-Nutzer muß es, mit Ausnahme der Performance, völlig transparent sein, ob seine Textverarbeitung oder Datenbankanwendung über ein DECT/LAN- oder ein Ethernet-Netz betrieben wird. Aber nicht nur die vielen Applikationen, auch die in der Praxis zahlreich vorhandenen Kommunikationsprotokolle und Netzwerkarchitekturen müssen unterstützt werden. Es ist



auch wichtig, daß sich die DECT/LAN-Software bezüglich der Installation nicht von anderen Netzwerktreibern unterscheidet. Dies bedeutet, daß der DECT/LAN-Treiber wie ein gewöhnlicher Netzwerktreiber in das Betriebssystem bzw. die Systemkonfigurationsdatei (bei MS-DOS- oder OS/2-Systemen z.B. „config.sys“) eingebunden wird.

Die aufgezeigten Randbedingungen lassen klar erkennen, daß eine Einbindung des DECT/LAN-Treibers in das Funktionsschichtenmodell wie in **Bild 4.1-3** dargestellt zu erfolgen hat.

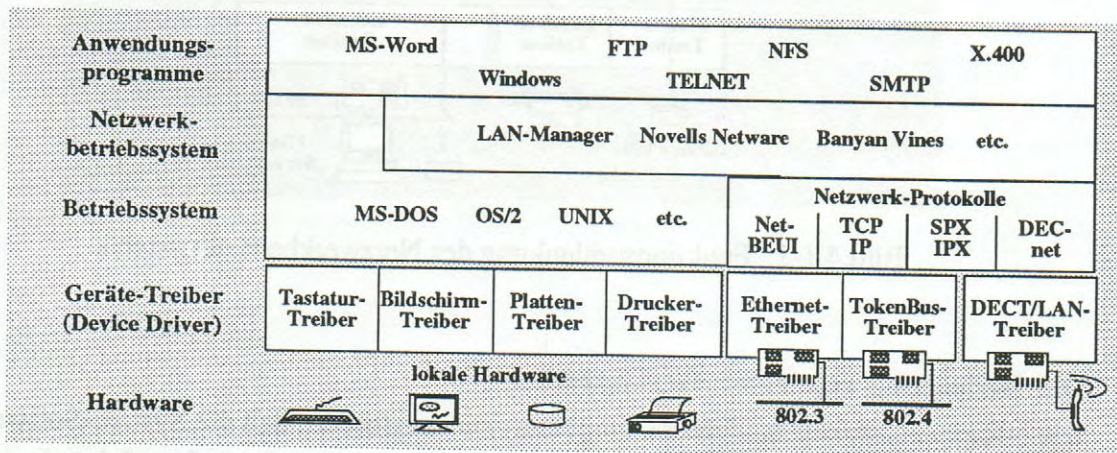


Bild 4.1-3: Einbindung des DECT/LAN-Treibers

## 4.1.2 Netzwerktreiber und Treiberschnittstellen

Im letzten Abschnitt wurde festgestellt, daß sich der zu entwickelnde DECT/LAN-Treiber von einem klassischen Netzwerktreiber nicht unterscheiden sollte, weshalb wir in diesem Abschnitt der Frage nachgehen, wie die klassischen PC-Netzwerktreiber strukturiert und welche Schnittstellen einzuhalten sind.

Wie **Bild 4.1-4** verdeutlicht, lassen sich PC-Netzwerktreiber in Protokoll-, MAC- und sogenannte „All-In-One“-Treiber unterteilen.

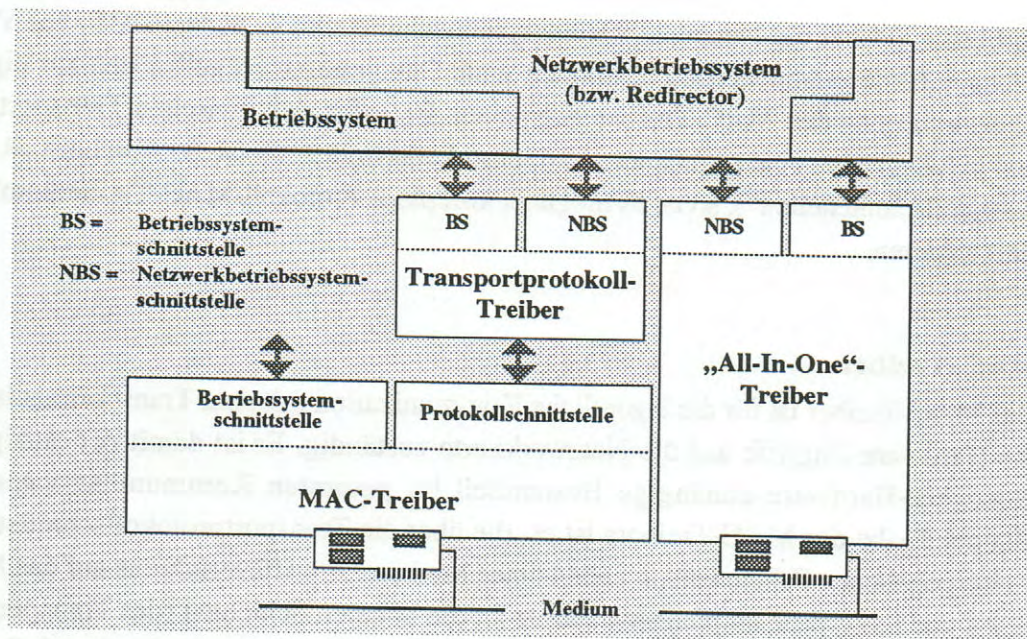


Bild 4.1-4: PC-Netzwerktreiber

### „All-In-One“-Treiber

„All-In-One“-Treiber realisieren sowohl das Transportprotokoll als auch den Mediumszugriff in einem einzigen Treiber-Modul und stellen dem Netzwerkbetriebssystem ihre Dienste über eine Netzwerkbetriebssystem-schnittstelle (z.B. NetBIOS) zur Verfügung. Zum Einsatz kommen „All-In-One“-Treiber vor allem dann, wenn die Wirts-CPU mittels aktiver Netzwerkkarten entlastet werden soll und nur ein Transportprotokoll im Endgerät betrieben wird. Indem „All-In-One“-Treiber speziell auf die Netzwerkkarte zugeschnitten und ohne unnötigen Schnittstellen-Overhead (Funktionsaufrufe, Parameterübergaben etc.) realisiert werden können, sparen sie Speicherplatz und erhöhen den Durchsatz. Ein Nachteil derartiger Treiber ist, daß diese i.d.R. nur ein bestimmtes Protokoll und nicht die Vielzahl der in der Praxis vorhandenen Transportprotokolle unterstützen.

Um Transportprotokolle und Adapterkarten flexibel miteinander kombinieren zu können und eine leichtere Anpassung der Netzwerk-Hardware (der Adapterkarten) an die Zielmaschinen zu erreichen, wird, wie in **Bild 4.1-4** dargestellt, die Kommunikationssoftware in der Praxis häufig in zwei separate Treibermodule aufgeteilt:

- hardware-unabhängige Transportprotokolltreiber
- hardware-abhängige MAC-Treiber

### Transportprotokoll-Treiber

Der Transportprotokolltreiber realisiert die hardware-unabhängigen Transportfunktionen und stellt dem Netzwerkbetriebssystem seine Dienste (z.B. das Öffnen einer Kommunikations-

verbindung) über die Netzwerkbetriebssystemschnittstelle (z.B. NetBIOS) zur Verfügung. Dienstanforderungen werden in oftmals viele Datagramme aufgelöst und zur eigentlichen Übertragung an den MAC-Treiber übergeben. Befindet sich zwischen Transportprotokoll- und MAC-Treiber eine sogenannte Multiprotokollschnittstelle (siehe Abschnitt „4.1.3 Multiprotokollschnittstellen“), so ist es möglich, komplexe Protokoll/MAC-Treiberkombinationen zu realisieren.

### MAC-Treiber

Der MAC-Treiber ist für die eigentliche Kommunikation mit dem Transportmedium und für die Hardware-Zugriffe auf die Netzwerkkarte zuständig. Er ist damit der einzige von der Netzwerk-Hardware abhängige Bestandteil im gesamten Kommunikationsprozeß. Die Hauptaufgabe des MAC-Treibers ist es, die über die Transportprotokollschnittstelle an ihn weitergegebenen Pakete entsprechend eines Mediumszugriffsverfahrens auf das Medium zu legen und fehlerfreie empfangene Pakete an die höheren Schichten (das Transportprotokoll) weiterzuleiten. Da der MAC-Treiber auf „primitive“ Basisfunktionen beschränkt bleiben kann (TransmitPacket, ReceivePacket etc.), läßt sich auch seine Größe und sein Entwicklungsaufwand relativ gering halten.

### Treiberschnittstellen

Wie in **Bild 4.1-4** dargestellt, verfügt jeder PC-Netzwerktreiber über mehrere Schnittstellen:

- Netzwerkschnittstellen:

Über die Netzwerkschnittstelle(n) ist jeder Netzwerktreiber mit über- bzw. untergeordneten Netzwerkkomponenten verbunden. Über diese Schnittstellen fordern bzw. bieten die Netzwerktreiber ihre Kommunikationsdienste an. Während die Transportprotokolltreiber über zwei Netzwerkschnittstellen verfügen, sind „All-In-One“- und MAC-Treiber mit nur jeweils einer Netzwerkschnittstelle ausgestattet.

- Betriebssystemschnittstelle:

Die zweite Schnittstelle eines jeden PC-Netzwerktreibers ist die Betriebssystemschnittstelle. Über diese Schnittstelle wird jeder Treiber beim Boot-Vorgang initialisiert und mit dem Betriebssystem verbunden. Anders als bei Drucker- oder Plattentreibern jedoch, werden die Dienstanforderungen (z.B. TransmitPacket) an einen Netzwerktreiber nicht über die Betriebssystemschnittstelle, sondern über die Netzwerkschnittstelle herangetragen. Aus diesen Gründen beschränkt sich die Betriebssystemschnittstelle eines Netzwerktreibers auf die unverzichtbaren Funktionen wie z.B. das beim Boot-Vorgang aufgerufene „Init“ und kann entsprechend klein gehalten werden.

### 4.1.3 Multiprotokollschnittstellen

Ziel der Multiprotokollschnittstellen ist es, daß „jedes“ Transportprotokoll über „jede“ Netzwerkkarte betrieben werden kann und auch komplexe Protokoll/MAC-Kombinationen möglich sind. Wie **Bild 4.1-4** verdeutlicht, können über Multiprotokollschnittstellen mehrere Transportprotokolle parallel auf die Dienste einer Netzwerkkarte zugreifen oder ein Transportprotokoll mehrere Netzwerkkarten parallel bedienen.

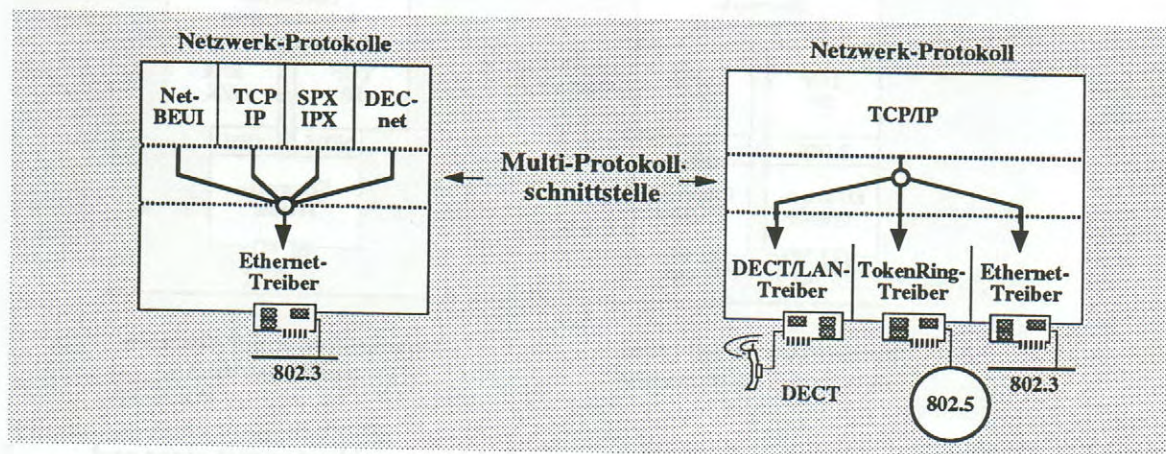


Bild 4.1-4: Komplexe Protokoll/MAC-Kombinationen über Multiprotokollschnittstellen

In der Praxis durchgesetzt haben sich die Multiprotokollschnittstellen

- NDIS (Network Driver Interface Spezifikation) der Firmen Microsoft und 3Com - [NDIS]
- ODI (Open Data-Link Interface) der Firmen Novell und Apple - [ODI]
- PD (Packet Driver Specification) der Clarkson University - [PD]

Realisiert wird der Multiprotokollbetrieb z.B. dadurch, daß sobald ein Receive-Paket im MAC-Treiber ansteht, dieser der Reihe nach versucht, das Paket an die übergeordneten Transportprotokolle weiterzugeben, bis schließlich ein Protokoll das Paket als das seine erkennt und akzeptiert (NDIS-Variante). Andere Implementierungsverfahren sehen vor, daß die Protokolltreiber dem MAC-Treiber mitteilen, welcher Pakettyp an sie weitergeleitet werden soll. Der MAC-Treiber muß dann vor der gezielten Weiterleitung das Typfeld eines jeden Pakets interpretieren (PD-Variante).

Das gesteckte Ziel der Multiprotokollschnittstellen (die volle Kombinierbarkeit „aller“ Netzwerkkarten mit „allen“ Transportprotokollen) scheitert in der Praxis oftmals daran, daß ein Transportprotokoll z.B. nur die PD-Schnittstelle unterstützt, während ein MAC-Treiber z.B. nur die NDIS- oder ODI-Schnittstelle realisiert. Will man über eine Netzwerkkarte zwei

Transportprotokolle mit unterschiedlicher Multiprotokollschnittstelle betreiben, so müssen in einem Endgerät zwei Karten und zwei sich nur in der Multiprotokollschnittstelle unterschiedlich verhaltende MAC-Treiber installiert werden, wobei beide Netzwerkkarten am selben Medium angeschlossen sind – siehe **Bild 4.1-6**.

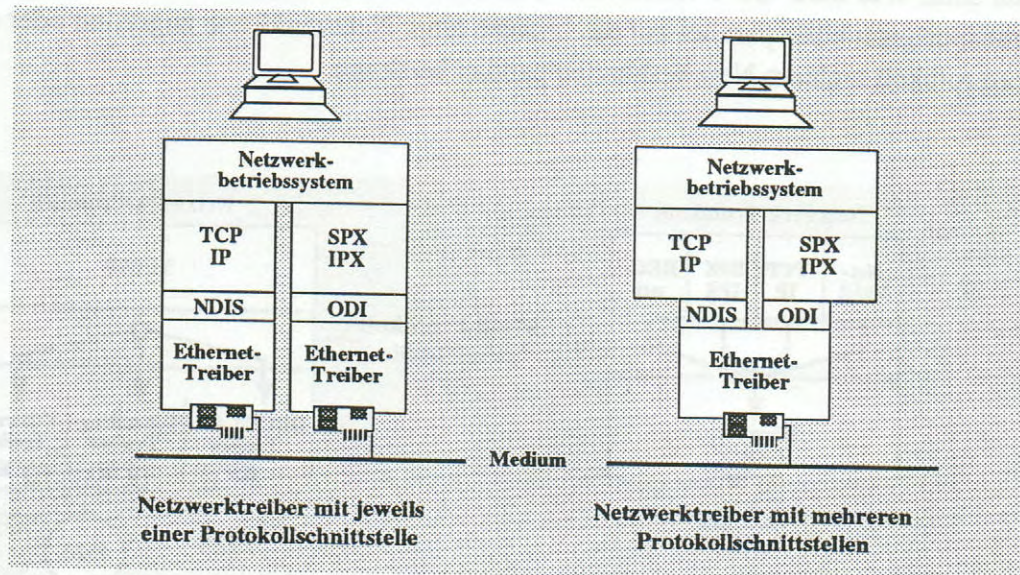


Bild 4.1-6: Multiprotokollbetrieb bei Netzwerkkarten mit einer und mehreren Multiprotokollschnittstellen

Um im Rahmen der angestrebten DECT/LAN-Integrationslösung volle Kombinierbarkeit erreichen zu können, werden wir darum bemüht sein, möglichst mehrere Multiprotokollschnittstellen in einem DECT/LAN-Netzwerktreiber parallel zu unterstützen – siehe **Bild 4.1-6** und Abschnitt „4.2.1 Die Architektur des Connection-Managers“.

#### 4.1.4 Die Mehrkanalkommunikation

Immer mehr Netzwerkknoten werden mit mehr als nur einer Kommunikationskarte ausgestattet, sei es,

- weil verschiedenartige Kommunikationsdienste oder Netze dies erfordern. So wird z.B. der LAN-Zugang über eine Ethernet- oder Token-Ring-Karte, der Zugang zum Datex-L oder Datex-P über eine X.21 bzw. X.24-Karte und der Zugang zum ISDN über eine ISDN-Karte realisiert.

- weil spezielle Netzwerkfunktionalitäten wahrzunehmen sind. So verfügen Brücken, Router, Gateways, Netzwerkmanagementstationen mit Out-Band-Signalisierung etc. immer über mehrere Kommunikationskarten.
- weil immer mehr portable PCs zum Einsatz kommen. So wird beispielsweise der portable PC eines Außendienstmitarbeiters im Büro über eine Ethernet-Karte an das Unternehmensnetz angeschlossen, beim Kunden dagegen erhält er Zugang zum Unternehmensnetz über das Telefonnetz (Modem), über ISDN oder über Mobilfunknetze (z.B. Modacom, GSM oder DECT).

Die bisherigen LAN-Konzepte sehen vor, daß für jede Netzwerkkarte ein eigener Treiber geladen werden muß. Bedenkt man, daß ein Treiber bei MS-DOS Systemen nicht erst bei Bedarf, sondern bereits beim Boot-Vorgang in das Betriebssystem eingebunden wird<sup>1</sup> und Speicherplatz belegt, und jeder Treiber mit redundanten Programmteilen wie Betriebssystem- und Protokollschnittstelle, Pufferverwaltung etc. ausgestattet ist, so wird klar, daß ein neues, bedarfsorientiertes und benutzerfreundliches Mehrkanalkommunikationskonzept wünschenswert ist. Die zu entwickelnde DECT/LAN-Software soll dieser Forderung gerecht werden.



<sup>1</sup> egal ob der Treiber dann tatsächlich benutzt wird oder nicht, denn eine Umkonfiguration eines PCs erfordert vom Benutzer Spezialkenntnisse (z.B. Modifikationen in der Systemdatei „config.sys“ mit anschließendem Restart)

## 4.2 Connection-Manager – ein Konzept zur Integration klassischer LAN- und DECT/LAN-Treiber

Bevor wir auf die wesentlichen Aspekte einer exemplarischen DECT/LAN-Treiberrealisierung zu sprechen kommen, soll in diesem Abschnitt ein dem DECT/LAN-Treiber übergeordnetes Konzept zur Integration der klassischen LAN- und DECT/LAN-Treibern vorgestellt werden. Das Ziel dieses Konzeptes (des Connection-Managers) ist es, die Basis des Netzbetriebs (den LAN- und DECT/LAN-Netzanschluß) so flexibel und wirtschaftlich wie möglich zu machen und die Bedienbarkeit (Administrierbarkeit) selbst einem unerfahrenen Anwender zu ermöglichen.

### 4.2.1 Die Architektur des Connection-Managers

Wie **Bild 4.2-1** verdeutlicht, gruppieren sich bei der Connection-Manager-Architektur um die zentralen Verarbeitungsfunktionen die Betriebssystem- und (Multi-) Protokollschnittstelle(n), die Zusatzfunktionen und die Kartentreiber.

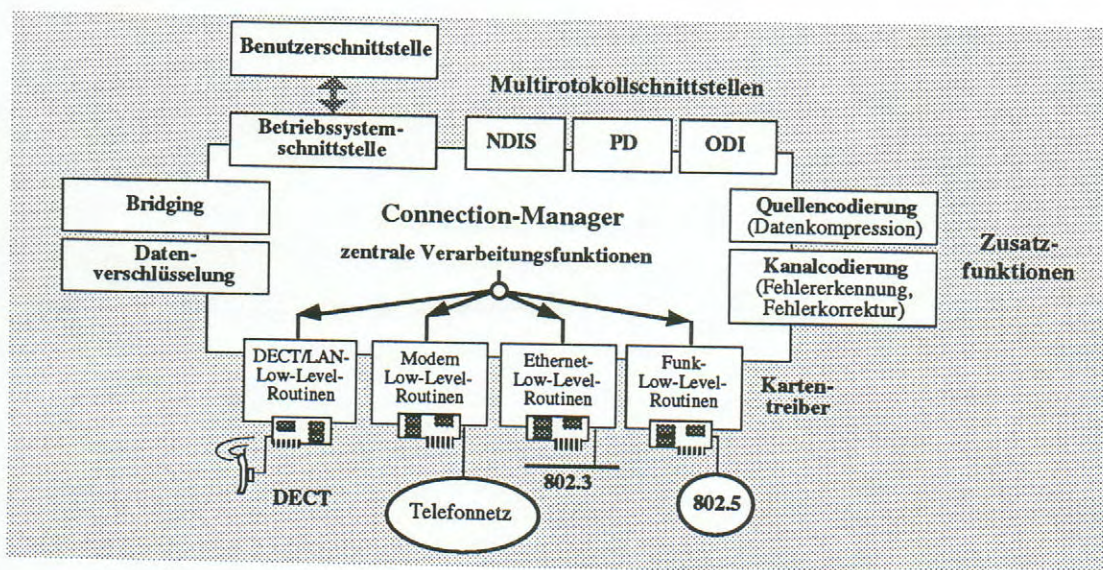


Bild 4.2-1: Connection-Manager-Architektur

Der Connection-Manager realisiert für alle ihm zugeordneten Adapterkarten die Betriebssystem-schnittstelle, die (Multi-) Protokollschnittstelle(n), Verwaltungsfunktionen (z.B. Paketpufferung) und Zusatzfunktionen, also die sogenannten „High-Level“-Funktionen. Die Kartentreiber enthalten nur „Low-Level“-Routinen, die direkt auf die Netzwerk-Hardware zugreifen und nach außen hin nicht sichtbar sind. Dazu zählen z.B. die Karteninitialisierung, die Realisierung des Mediumzugriffsverfahrens, die Paketübertragung und die Interrupt-

Bearbeitung. Über eine betriebssystemkompatible Benutzerschnittstelle können die Anwender im laufenden Betrieb Konfigurationen<sup>1</sup> vornehmen und verbindungsrelevante Informationen<sup>2</sup> abfragen.

### 4.2.2 Connection-Manager-Zusatzfunktionen

In den folgenden Abschnitten werden nur zwei der vielen möglichen Connection-Manager-Zusatzfunktionen beschrieben.

#### Bridging- und „Remote-Access“-Funktionalität

Durch Erweiterung des Connection-Managers um eine Bridging-Funktionalität (siehe Bild 4.2-3) kann dieser als leistungsfähiges, protokolltransparentes, kostengünstiges und leicht zu bedienendes Koppellement zur Integration von Remote Workgroups und Remote Workstations in ein Corporate Network fungieren.

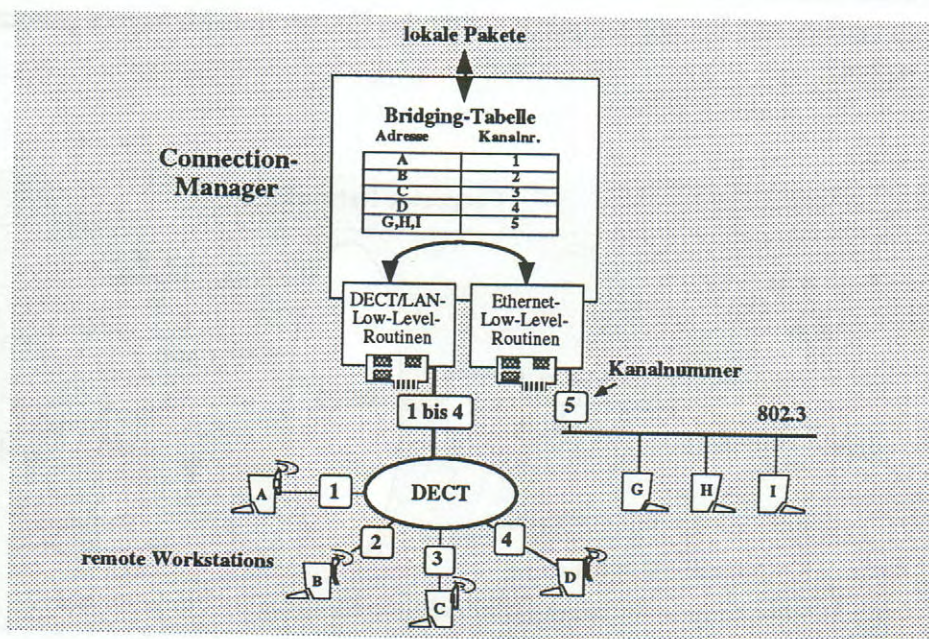


Bild 4.2-3: Die Bridging- und „Remote-Access“-Funktionalität

Da die Bridging-Funktionalität für alle dem Connection-Manager zugeordneten Netzwerkkarten nutzbar ist, ist es natürlich auch möglich, den Connection-Manager z.B. als DECT/LAN-Ethernet-Koppellement einzusetzen. Indem die Netzwerktreiber automatisch

<sup>1</sup> z.B. Deaktivierung einer Ethernet-Karte und Aktivierung einer Wählverbindung in ein PBX/LAN-Netz, Abbau einer ISDN-Wählverbindung nach einer Minute Inaktivität, automatische Aktivierung einer Wählverbindung in ein PBX/LAN-Netz bei Ausfall des Ethernet-Netzes etc.

<sup>2</sup> z.B. aufgelaufene Verbindungsgebühren im ISDN, Verkehrsstatistiken etc.

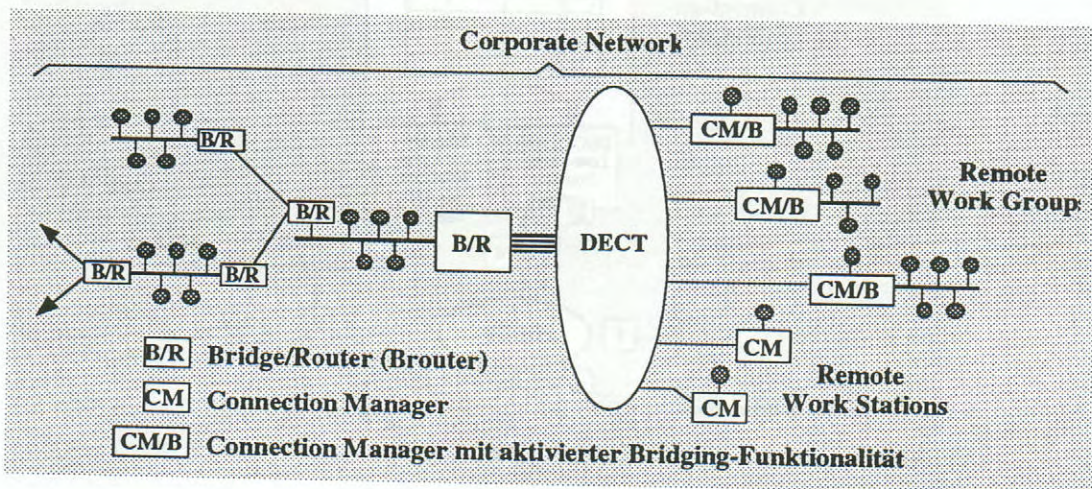


erkennen, ob sie an eine Multipoint- oder eine Point-to-Point-Verbindung angeschlossen sind, kann der Connection-Manager sowohl als lokale als auch als remote-Bridge eingesetzt werden.

Wie **Bild 4.2-3** ebenfalls verdeutlicht, ist es nicht nur möglich, den Weg der Pakete zwischen LANs zu lenken, sondern auch, daß sich externe Teilnehmer z.B. über DECT einwählen und über die Bridging-Funktionalität des Connection-Managers Zugang zum Ethernet erhalten, der Connection-Manager also als sogenannter „Remote-Access“-Knoten fungiert.

Es ist zu beachten, daß trotz der Bridging- und „Remote-Access“-Funktionalität die Workstation- oder Server-Funktionalität der Station erhalten bleibt, da an die Station adressierte Pakete (lokale Pakete) an die übergeordneten Schichten weitergereicht werden.

Beim Zusammenschluß vieler Remote Workgroups in einem zentralen Punkt<sup>1</sup>, lassen sich die Vorteile einer kombinierten Bridge/Router-Einheit (Brouter) nutzen, wenn in der Zentrale ein Brouter installiert wird<sup>2</sup>, während in den Remote Workgroups<sup>3</sup>, die kostengünstige und leicht zu bedienende Bridging-Funktionalität des Connection-Managers genutzt wird – siehe **Bild 4.2-4**.



**Bild 4.2-4:** Kombiniertes Connection-Manager und Brouter Einsatz

Diesen Abschnitt zusammenfassend kann man feststellen, daß der Connection-Manager mit integrierter Bridging- und „Remote-Access“-Funktionalität folgende Vorteile mit sich bringt:

- 1 z.B. im zentralen Rechenzentrum eines Unternehmens
- 2 Der Brouter ist i.d.R. wesentlich teurer und kompliziert zu bedienenden als eine Bridge.
- 3 also dort, wo i.d.R. keine speziellen Netzwerkkennnisse vorhanden sind

- sehr leistungsfähig (n-mal 64 kBit/s evtl. mit Datenkompression)
- flexibel und einfach in der Installation und Bedienung („Plug, Play and Forget“)
- leichte Erweiterung einer Remote Workstation zu einer Remote Workgroup
- leichte Integration einer Remote Workstation oder Remote Workgroup in ein Corporate Network
- kostengünstig in der Anschaffung und im Betrieb
- durch Kombination mit einem zentralen Brouter lassen sich dezentral die Vorteile einer Bridge und über das gesamte Netzwerk gesehen die Vorteile eines Brouters nutzen.

### Multi-LAN-Funktionalität

Wie **Bild 4.2-5** verdeutlicht, kann die Bridging-Tabelle des Connection-Managers auch dazu verwendet werden, die von den höheren Netzwerkschichten erzeugten Datenpakete in die richtigen Netzwerkkanäle (LANs) zu lenken und so einen Multi-LAN-Knoten zu realisieren. Für die übergeordneten Schichten bleibt die Paketlenkungsfunktion des Connection-Managers transparent. Broadcast-Pakete (z.B. der Server gibt seine Netzwerkressourcen bekannt) werden entweder in alle oder nur in bestimmte LANs weitergeleitet, Pakete mit einer Einzelzieladresse nur in das sich aus der Bridging-Tabelle ergebende Ziel-LAN.

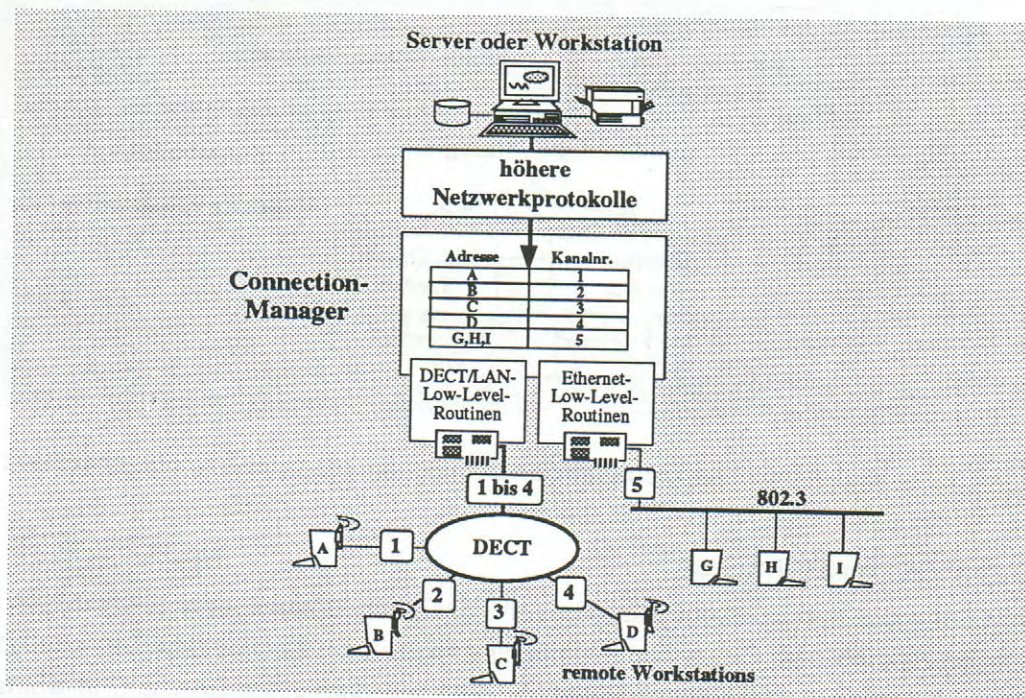


Bild 4.2-5: Der Connection-Manager als Basis eines Multi-LAN-Knotens

Auf diese Weise ist es z.B. möglich, daß eine Workstation mit permanentem Anschluß an ein lokales Ethernet-Netz über eine zusätzlich installierte DECT/LAN-Karte bedarfsweise eine Verbindung in ein anderes Netz aufbaut, um dort Datenbankrecherchen durchzuführen, ohne die Verbindung zum lokalen Ethernet und damit zu Netzwerkdiensten wie E-Mail, Terminal-Sessions etc. zu verlieren.

### 4.2.3 Exemplarische Implementierung des Connection-Managers

Zur Demonstration der Machbarkeit und Beurteilung der Leistungsfähigkeit ist es nicht erforderlich, den vollen Funktionsumfang des in Bild 4.2-1 dargestellten Connection-Manager-Konzeptes zu implementieren. Aus diesem Grund wurde der Connection-Manager im Rahmen dieser Arbeit nur mit den in Bild 4.2-6 dargestellten und nachfolgend beschriebenen Funktionen realisiert.

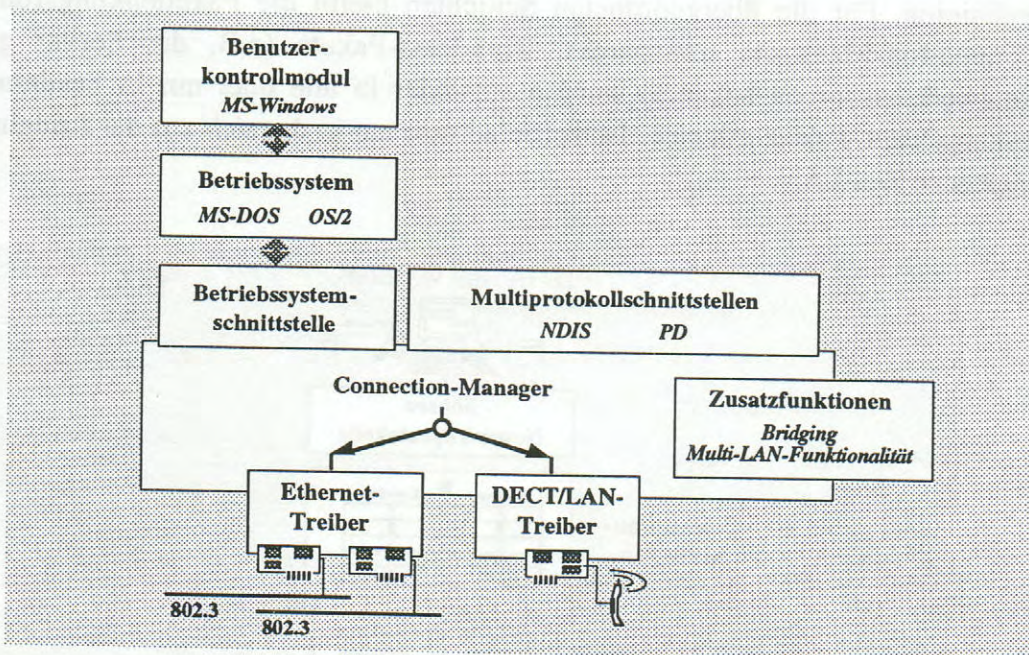


Bild 4.2-6: Die Elemente der Connection-Manager-Prototypenversion

#### Benutzerkontrollmodul

Mit Hilfe des Benutzerkontrollmoduls soll der Netzwerkanwender sehr einfach und im laufenden Betrieb dem Connection-Manager zugeordnete Netzwerktreiber konfigurieren und Zusatzfunktionen aktivieren können. In der exemplarischen Connection-Manager-Implementierung wurde zu Demonstrationszwecken eine unter MS-Windows ablauffähige Wähloberfläche realisiert, über die sich der Anwender durch Aktivierung eines Icons oder durch händische Rufnummerneingabe in ein bestimmtes DECT/LAN-Netz einwählen und die Ver-

bindung dahin abrechnen kann. Hervorzuheben ist, daß dabei die Verbindungssteuerungsfunktionen des DECT/LAN-Netzwerktreibers nicht direkt vom Benutzerkontrollmodul angesprochen werden<sup>1</sup>, sondern daß zwischen Connection-Manager und Benutzerkontrollmodul die bei MS-DOS, OS/2 und UNIX vorhandene IOCtrl-Schnittstelle benutzt wird.

### **Betriebssystemschnittstellen**

Als Netzwerkbetriebssystem sollte die Connection-Manager-Prototypenversion den LAN-Manager, Novell's Netware und TCP/IP unterstützen und als Computerbetriebssystem MS-DOS und OS/2. Eine Implementierung unter UNIX wurde ebenfalls angedacht, bisher aber nicht realisiert. Über die Gestaltung der Treiberschnittstellen zu MS-DOS, OS/2 und UNIX, soll an dieser Stelle nichts ausgesagt werden, siehe hierzu z.B. [Klein], [Iaco] oder [Bass]. Als Schnittstelle zum Kontrollfeld des Benutzers kam die IOCtrl-Schnittstelle der Betriebssysteme MS-DOS, OS/2 und Unix zum Einsatz – siehe z.B. [Tisch], [Altha], [Dunc] oder [Egan].

### **Multiprotokollschnittstellen**

Wie in **Bild 4.2-6** dargestellt, wurde die Connection-Manager-Prototypenversion mit den Multiprotokollschnittstellen NDIS (siehe [NDIS]) und PD (siehe [PD] und [Doup]) ausgestattet, wobei (bei Bedarf) beide Multiprotokollschnittstellen parallel betrieben werden können. Der Integration der ODI-Schnittstelle (siehe [ODI]) standen bisher die hohen Kosten<sup>2</sup> für die Offenlegung der ODI-Spezifikation im Wege. Seit kurzem ist auch diese Spezifikation gebührenfrei erhältlich.

### **Zusatzfunktionen**

Die Vielzahl der möglichen Connection-Manager-Zusatzfunktionen wurde in der Prototypenversion reduziert auf die Bridging- und Multi-LAN-Funktionalität. Zur Realisierung beider Funktionalitäten war es erforderlich, eine dynamische Bridging-Tabelle aufzubauen, in der „alle“ an den LAN-Segmenten angeschlossenen Teilnehmer mit deren 6-Byte langen Adressen verzeichnet werden.

Realisiert wurden die Connection-Manager-Zusatzfunktionen der Einfachheit halber durch Codeintegration in den Connection-Manager-Treiber. In späteren Produktversionen könnten die Zusatzfunktionen auch in eigenen Treibermodulen realisiert werden. Diese Treibermodule können dann bei Bedarf geladen werden. Die Kommunikation zwischen den Treiber-

---

<sup>1</sup> Viele Betriebssysteme lassen einen direkten Zugriff nicht zu.

<sup>2</sup> \$ 7500

modulen erfolgt über die IOCTL-Schnittstelle. **Bild 4.2-7** verdeutlicht dieses Lösungskonzept und zeigt einen Ausschnitt aus der Konfigurationsdatei „config.sys“.

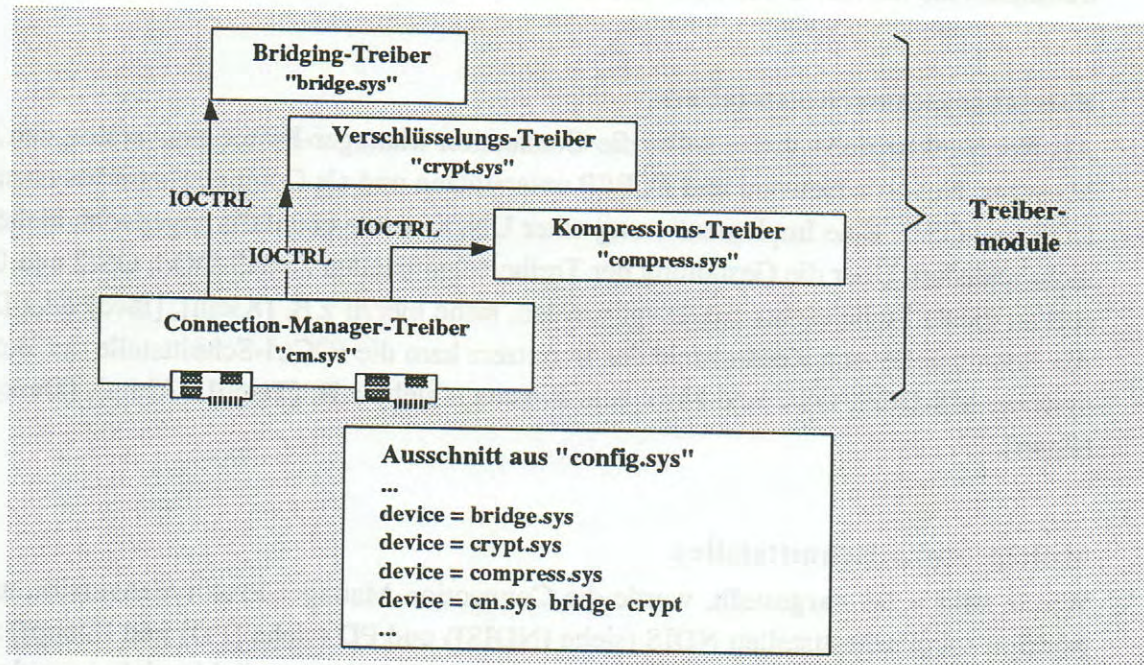


Bild 4.2-7: Realisierung der Connection-Manager-Zusatzfunktionen in Treibermodulen

### Zentrale Connection-Manager-Aufgaben

Zu den zentralen Aufgaben des Connection-Managers gehört es, den vertikalen (von den übergeordneten Protokollen an die Netzwerkkarten und umgekehrt) und horizontalen (von Netzwerkkarte zu Netzwerkkarte) Paketstrom zu lenken, zu adaptieren und bei Bedarf zu puffern.

Zur Paketlenkung bedient sich der Connection-Manager der Paketlenkungsinformation der Bridging-Tabelle. Wird für eine Adresse kein Ziel-LAN gefunden oder handelt es sich bei dem vorliegenden Paket um ein Broadcast- oder Multicast-Paket, so wird das Paket nach dem Flooding-Verfahren in alle Netzsegmente geleitet.

Die von den höheren Protokollen an den Connection-Manager zur Übertragung übergebenen Pakete werden i.d.R. nicht als eine zusammenhängende Byte-Kette übergeben, sondern z.B. in der in **Bild 4.2-8** dargestellten Struktur, weshalb der Connection-Manager die Paketeile in der richtigen Reihenfolge auf eine durchgehende Byte-Kette kopiert, mit der logischen Kanalnummer versieht und dann an die Kartentreiber weiterreicht.

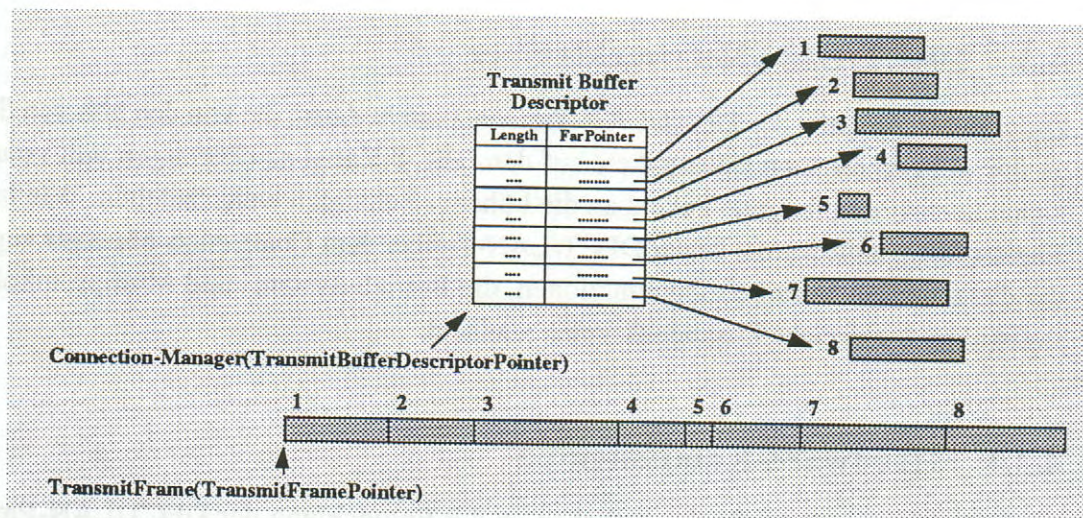


Bild 4.2-8: Zusammenkopieren der Paketeile zu einer durchgehenden Byte-Kette

Der Connection-Manager darf an die Kartentreiber immer nur dann ein Paket weiterreichen, wenn diese bereit dazu sind. Kommen mehr Pakete an als die Kartentreiber verarbeiten können, so sollte der Connection-Manager diese Pakete aus Performance-Gründen kurzzeitig zwischenspeichern können. Gleiches gilt auch für die Paketübergabe an die höheren Protokolle, wenn diese signalisieren, daß sie im Moment keine Pakete annehmen können.

### Ethernet-Treiber

Wie in **Bild 4.2-8** dargestellt, unterstützt der dem Connection-Manager zugeordnete Ethernet-Treiber zwei Netzwerkkarten. Als Ethernet-Adapter kamen NI4210 Karten der Firma Interlan zum Einsatz. Der Assembler-Quellcode für diese Kartentreiber wurde von Interlan erworben, den eigenen Bedürfnissen angepaßt (z.B. Modulstruktur, OS/2-Ablauffähigkeit etc.) und in den Connection-Manager integriert. Dabei wurde besonderes Augenmerk darauf gelegt, daß der Interlan-Ethernet-Treiber ohne großen Aufwand durch andere Netzwerktreiber substituierbar ist.

### DECT/LAN-Treiber

Als letzter „Baustein“ der exemplarischen Connection-Manager-Implementierung fehlt nun noch der DECT/LAN-Treiber. Wegen seiner besonderen Bedeutung werden wesentliche Elemente dieses Treibers im nächsten Kapitel ausführlicher besprochen.

### 4.3 DECT/LAN-Netzwerktreiber

Die zentrale Aufgabe des DECT/LAN-Treibers besteht darin, die von den höheren Schichten an ihn weitergereichten Datenpakete auf den DECT-Kanal zu übertragen und vom DECT-Kanal korrekt empfangene Pakete an die höheren Schichten weiterzuleiten. Wie im letzten Kapitel festgestellt wurde, ist der DECT/LAN-Treiber dem Connection-Manager zugeordnet, der die zentralen Verarbeitungsfunktionen, die Betriebssystem- und Multiprotokollschnittstelle(n) und zahlreiche Zusatzfunktionen realisiert.

#### 4.3.1 Hardware-Komponenten

Bild 4.3-1 verdeutlicht die wesentlichen an der Entwicklung der DECT/LAN-Prototypenversion beteiligten Hardware-Komponenten.

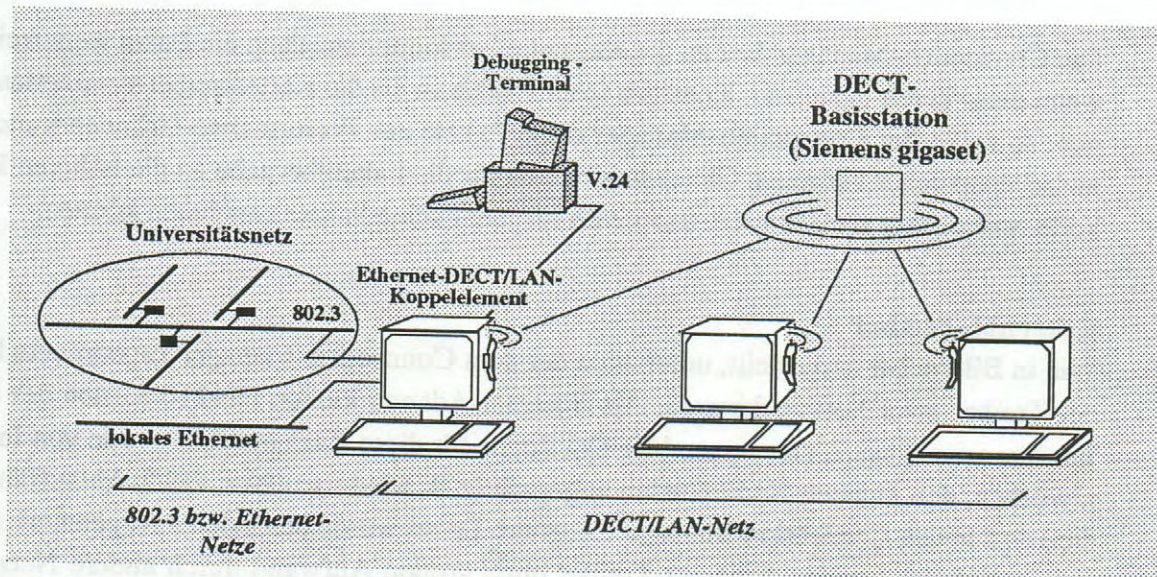


Bild 4.3-1: Hardware-Komponenten der DECT/LAN-Prototypenversion

#### DECT-Basisstation

Im Zentrum des DECT/LAN-Betriebs steht das Gigaset, das DECT-System der Firma Siemens.

#### Netzwerkkarten

Da bis heute keine DECT-Karten zur Verfügung stehen, wurden in den Endgeräten (PCs) synchrone Netzwerkkarten eingesetzt. Für den DECT/LAN-Anschluß von XTs kam die X.21-Karte mit der Bezeichnung DU31 (siehe [DU31]) und für den Anschluß von ATs die

modernere DU78-Karte (siehe [DU78] und [IOBOARD]) zum Einsatz. Alle Boards sind mit einem Serial Communications Controller (SCC) ausgestattet – siehe [SCC] - und über einen externen Takt (geliefert vom modifizierten Gigaset) steuerbar.

Zur Verbindungssteuerung (automatischer Wählvorgang) zwischen PC und Handset wird eine V.24-Schnittstelle verwendet.

### **Debugging-Hilfen**

Da MS-DOS „Non-Reentrant“ ist und der MS-DOS-Debugger DEBUG selbst MS-DOS-Funktionen für seine I/O-Operationen benutzt, kann dieses Programm nicht zur Fehlersuche in Geräte-Treibern benutzt werden. Darüber hinaus müssen Geräte-Treiber bereits beim Boot-Vorgang getestet werden können, zu einer Zeit also, zu der die Applikation DEBUG nicht zur Verfügung steht. Um Treiber dennoch testen zu können, wird vor dem zu testenden Geräte-Treiber ein sogenannter Debug-Treiber geladen. Dieser Debug-Treiber reagiert auf den Interrupt-Aufruf „int 3“ und übergibt daraufhin einem externen Debug-Programm die Kontrolle. Wie in **Bild 4.3-1** dargestellt, läuft das externe Debug-Programm auf einem über die V.24-Schnittstelle an den zu testenden PC angeschlossenen Computer (Debugging-Terminal).

Zur optischen Kontrolle sehr hilfreich waren an die Transmit- und Receive-Leitungen der synchronen Datenschnittstelle angebrachte LEDs. Damit konnten die Leitungszustände auch optisch überwacht werden. Sehr einfach konnte damit z.B. die Mediumsauslastung abgeschätzt werden.

Zur akustischen Kontrolle wurde an der zu überwachenden Programmstelle eine Sound-Routine eingebunden, die den PPI<sup>1</sup> 8255-Baustein des PCs nutzt. Damit konnte z.B. sehr einfach der Paketempfang, ein Pufferüberlauf oder der Systemabsturz in einer Interrupt-Service-Routine<sup>2</sup> angezeigt werden.

### **4.3.2 Modulstruktur**

**Bild 4.3-2** verdeutlicht die Modulstruktur des exemplarischen DECT/LAN-Treibers. Um eine schnelle Portierung der Treiber-Software auf andere Hardware-Plattformen (Adapterkarten, Computermodelle) zu ermöglichen, wurden sämtliche Hardware-Zugriffe in den „Low-Level“-Modulen durchgeführt.

---

<sup>1</sup> Programmable Peripheral Interface

<sup>2</sup> Sirenenwirkung: SoundOn() beim Eintritt in die ISR, SoundOff() beim Austritt.



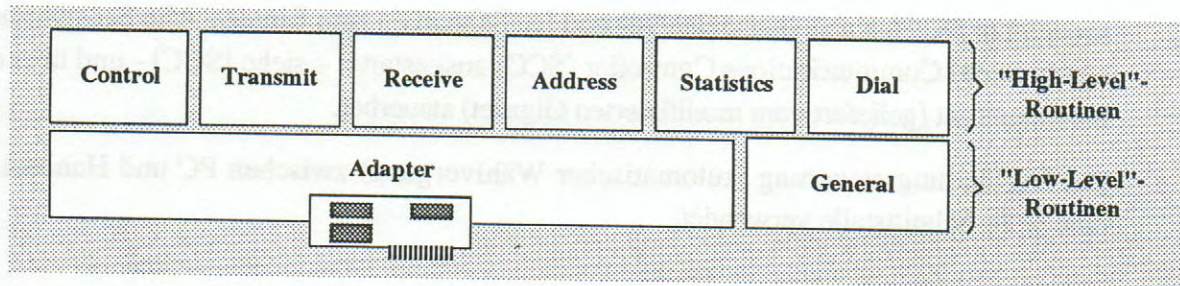


Bild 4.3-2: Funktionale Modul-Struktur des DECT/LAN-Treibers

In den „High-Level“-Modulen erfolgt die Initialisierung und Steuerung des Netzwerktreibers (Control-Modul), die Realisierung des Mediumszugriffsverfahrens (bei Multipoint-Betrieb) und Übertragung der Datenpakete (Transmit-Modul), der Paketempfang und die Übergabe der korrekt empfangenen Datenpakete an die höheren Schichten (Receive-Modul), die Einstellung und Überprüfung der lokalen Adressen (Address-Modul), die Fortschreibung der Statistikdaten (Statistics-Modul) und die im Gegensatz zu klassischen LAN-Treibern in einem DECT/LAN-Netz erforderliche Verbindungssteuerung (Dial-Modul). Auf die Beschreibung der Prozeduren eines jeden Moduls wird an dieser Stelle verzichtet.

## 4.4 DECT/LAN-Prototypenkonfigurationen

Realisiert und der Öffentlichkeit vorgestellt wurde die DECT/LAN-Integrationslösung in den beiden Versionen reiner DECT/LAN-Treiber und Connection-Manager mit integriertem DECT/LAN- und Ethernet-Treiber. Beide Versionen werden nachfolgend kurz beschrieben.

### 4.4.1 DECT/LAN-Treiberversion

Bild 4.4-1 verdeutlicht die Inhalte der reinen DECT/LAN-Netzwerktreiberversion. Implementiert wurde dabei der DECT/LAN-Treiber mit einer synchronen X.21-Interface-Karte, die NDIS-Multiprotokollschnittstelle und eine unter MS-Windows ablauffähige Benutzerschnittstelle. Die Anwahl der Zielstation erfolgte manuell über das Bedienfeld des Handsets. Als Betriebssystem kam MS-DOS und OS/2 und als Netzwerkbetriebssystem der LAN-Manager zum Einsatz. Ein Koppellement zu anderen Netzen wurde nicht realisiert.

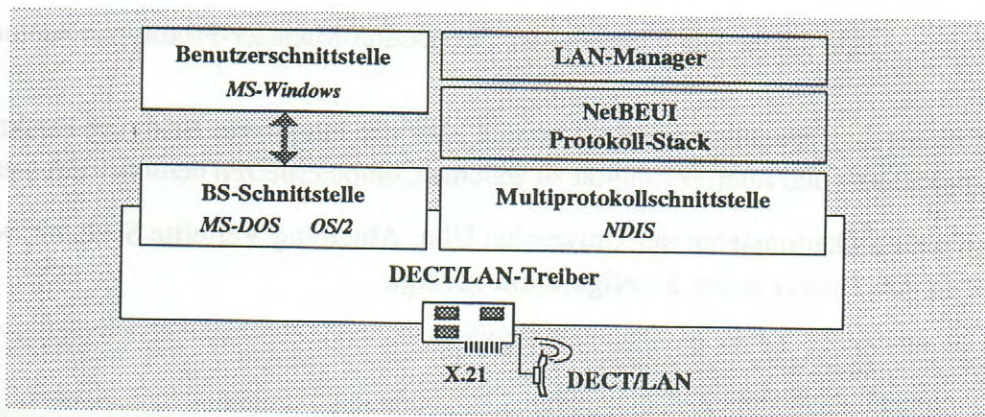


Bild 4.4-1: DECT/LAN-Treiberversion

Mit dieser Version konnte die Machbarkeit der DECT/LAN-Integrationslösung und die Kompatibilität dieser Lösung zu allen gängigen PC-Applikationen demonstriert werden. Darüber hinaus war es möglich, die Leistungsfähigkeit eines 32 kBit/s-LANs an praktischen Beispielen zu beurteilen.

### 4.4.2 Connection-Manager Version

Die Connection-Manager-Version umfaßte alle in **Bild 4.4-2** dargestellten Elemente.

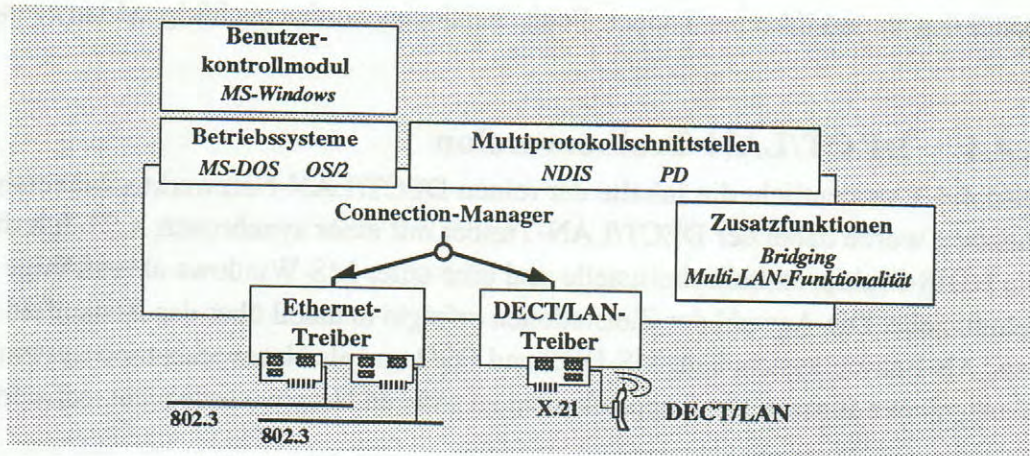


Bild 4.4-2: Die Elemente der Connection-Manager-Prototypenversion

Mit dieser Implementierungsstufe konnte über die integrierte Bridging-Funktionalität die Kompatibilität und Interoperabilität zu anderen Computernetzen demonstriert werden.

Im Kommunikationslabor der Universität Ulm, Abteilung Verteilte Systeme, wird z.Zt. die in **Bild 4.4-3** dargestellte Konfiguration gezeigt.

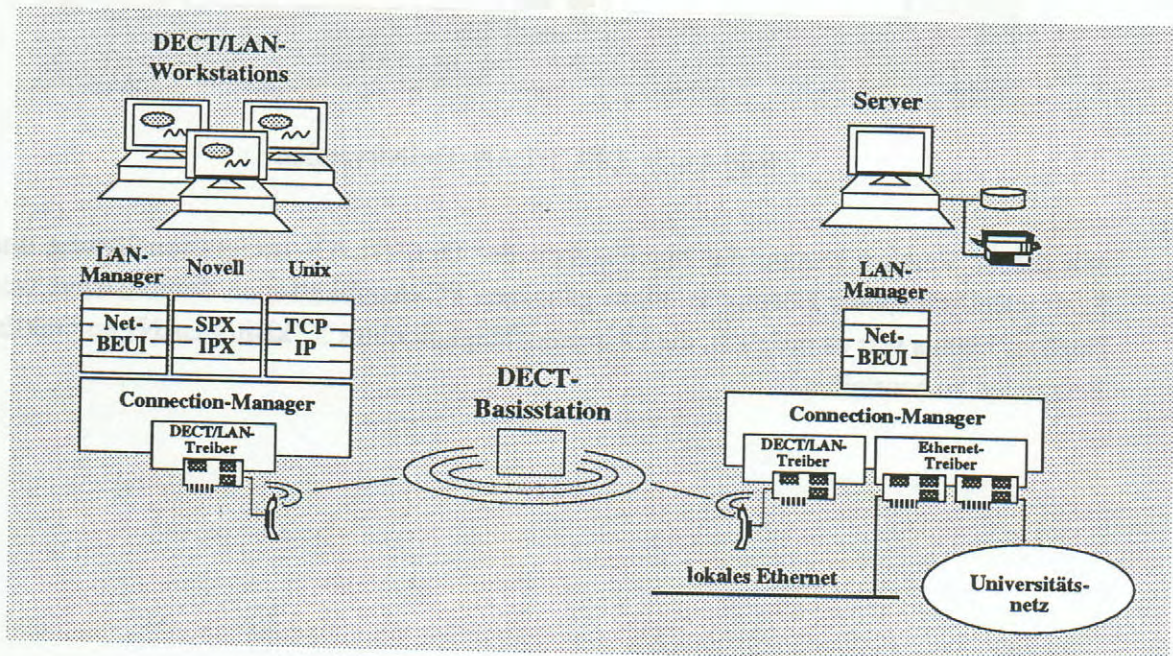


Bild 4.4-3: Demonstrationsszenario an der Universität Ulm, Abteilung Verteilte Systeme

Über zwei parallel betriebene Multiprotokollschnittstellen (NDIS und PD) werden in einer DECT/LAN-Workstation gleichzeitig drei verschiedene Protokollstapel (NetBEUI, SPX/IPX und TCP/IP) betrieben. Die NetBEUI-Pakete werden vom LAN-Manager-Server verarbeitet, während die TCP/IP und SPX/IPX-Pakete auf das Universitätsnetz oder das lokale Ethernet weitergeleitet werden. Indem von allen drei Netzsegmenten auf Dienste des Servers zugegriffen werden kann, demonstriert der Server bzw. der im Server ablaufende Connection-Manager seine Multi-LAN-Funktionalität. Die Anwahl der Zielstationen erfolgt automatisch über eine V.24 Verbindung zwischen PC und DECT-Handset.

Über zwei parallel betriebene Mehrprozessorkomplexe (MPCs und HP) werden in einer DECT-LAN-Weiterleitung gleichzeitig drei verschiedene Protokolle (MIB, SPANX und TCP/IP) betrieben. Die MIB-B-Karte wandelt vom LAN-Manager-Server empfangene, während die TCP/IP- und SPANX-Karte mit dem LAN-Manager oder dem lokalen Netzwerk weitergeleitet werden. Dabei werden alle die Netzsegmente auf Dienste der Server zugreifbar werden kann, dementsprechend der Server bzw. der im Server ablaufende Connection-Manager seine Multi-LAN-Funktionalität. Die Auswahl der Zielstationen erfolgt automatisch über eine V.24-Verbindung zwischen PC und DECT-Handset.

# Kapitel 5

## Zusammenfassung

### Ergebnisse

In den vorangegangenen Kapiteln wurden DECT/LAN-Integrationslösungen vorgestellt und diskutiert und ein DECT/LAN-Funktionsmuster erstellt. Die wichtigsten Ergebnisse lassen sich in folgenden Punkten zusammenfassen:

- Basis für die in diesem Bericht vorgestellten DECT/LAN-Integrationslösungen sind Punkt-zu-Punkt- und Mehrpunkt-Verbindungen.
- Die DECT/LAN-Netzwerktreiber in den Endgeräten unterscheiden sich bezüglich ihrer Software-Schnittstellen (zum Betriebssystem und zu den Transportprotokollen) nicht von klassischen Netzwerktreibern wie Ethernet oder Token-Ring. Daraus folgt, daß sind über DECT/LAN alle am Markt verfügbaren Netzwerkarchitekturen (z.B. Microsofts LAN-Manager, Novells Netware oder der TCP/IP-Protocolstack) und LAN-Applikationen ohne Modifikationen betreiben lassen, d.h. es ist volle Software-Kompatibilität gegeben<sup>1</sup>.
- Die Kopplung mehrerer DECT/LANs untereinander und mit klassischen LAN-Lösungen wie z.B. Ethernet oder Token-Ring ist möglich, d.h. die in diesem Bericht vorgestellten DECT/LAN-Integrationslösungen stellen keine Insellösung dar.

### Einsatzszenarien

Die möglichen DECT/LAN-Einsatzbereiche sind äußerst vielfältig und eine Aufzählung dieser würde den Rahmen dieses Berichtes bei weitem sprengen. Generell kann festgehalten werden, daß überall dort, wo ein Arbeitsplatz (Standort) über keinen LAN-Anschluß verfügt, dieser in ein DECT/LAN integriert werden kann.

Wenn man jedoch bedenkt, daß die „Standard“-Übertragungsrate im DECT z.Zt. nur 32 kBit/s pro Kanal beträgt, so wird deutlich, daß DECT/LAN ohne Kanalbündelung natürlich keine Substitution heutiger LANs darstellt, sondern eine wirtschaftliche Ergänzung zu herkömmlichen Vernetzungskonzepten, insbesondere dann, wenn es gilt Verkabelungspro-

---

<sup>1</sup> Die ISDN-Industrie mußte sich in der Vergangenheit zurecht den Vorwurf gefallen lassen, zu sehr auf Technik zu setzen und zu wenig in Anwendungen zu denken. Mit PBX/LAN jedoch wird „jede“ PC-Applikation auch zu einer ISDN-Applikation.

bleme zu überwinden. Die in diesem Bericht vorgestellten DECT/LAN-Integrationsansätze zielen schwerpunktmäßig auf die vielen mobilen „Computer-Arbeitsplätze“, die nicht im Einzugsbereich einer konventionellen LAN-Verkabelungsstruktur liegen, die keine hohen Ansprüche an das Durchsatzverhalten des Netzes stellen und für die deswegen eine konventionelle LAN-Verkabelung zu teuer oder nicht möglich ist.

## Schlußbemerkungen

Durch die Bereitstellung eines DECT/LAN-Funktionsmusters wurde die Realisierbarkeit unter Beweis gestellt und auch bei Vorträgen und Präsentationen erweckte der DECT/LAN-Prototyp großes Interesse. Es ist deshalb zu hoffen, daß die DECT-Technologie zukünftig nicht nur zur Sprachkommunikation sondern auch zur Übermittlung von Daten genutzt wird.

Aktuelle Entwicklungen zum Thema DECT/LAN-Integration können über unseren WWW-Server unter URL: <http://www-vs.informatik.uni-ulm.de> abgerufen werden. Gerne sind wir auch zu einem Gedankenaustausch oder zu Kooperationen bereit.

# Anhang

## Literaturverzeichnis

- [802.3] IEEE Standards for Local Area Networks: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications, IEEE Standards Board, New York
- [Achatz] Klaus Achatz: Leistungsanalyse für virtuelle Bustopologien, Diplomarbeit, Universität Augsburg, 1991
- [AdTo] Phillip M. Adams, Clovis L. Tondo: Writing DOS Device Drivers in C, Prentice Hall, ISBN 0-13-970864-2
- [Altha] Martin Althaus: Das PC Profibuch, Sybex, 1988
- [Bass] John L. Bass: Unix Device Drivers, Dr. Dobb's Journal, December 1984
- [BerGal] Dimitri Bertsekas, Robert Gallager: Data networks, Prentice Hall, 1992
- [Boell] Hans-Peter Boell: Lokale Netze – Momentane Möglichkeiten und zukünftige Entwicklung, McGraw-Hill, 1989
- [Boro] Petra Borowka: Brücken und Router, DATACOM, 1992
- [Brand] Don Brandly: Implementing Direct Memory Access (DMA) in C, The C Users Journal, May 1992
- [Bran] Lewis Brandscomb: Networks for the Nineties, IEEE Infocom, 1983
- [Cort] David E. Cortesi: OS/2-Programmierhandbuch, Markt&Technik, 1989
- [DECT] ETS 300 175-1 bis ETS 300 175-9, October 1992
- [DMA] 82C37A - CMOS High Performance Programmable DMA Controller, Harris Corporation, 1985
- [Doup] Joe R. Douppnik: Paket Drivers made simple, Utah State University
- [DU31] Pflichtenheft SAS-DUE Adapter I für XT-Bus, Nixdorf Computer AG
- [DU78] Pflichtenheft SAS-DUE Adapter II für AT-Bus, Nixdorf Computer AG
- [Dunc] Ray Duncan: Advanced OS/2 Programming, Microsoft Press, 1989
- [Egan] Janet Egan: Unix device drivers, Addison-Wesley, 1989



- [Esde] Erich Esders: Treiberimplementation in MS-DOS, mc 12/1987
- [Fromm] Ingrid Fromm: Anforderungen an private Netze der Bürokommunikation, Datacom 2/88
- [HamRei] Joseph Hammond, Peter J. P. O'Reilly: Performance Analysis of Local Computer Networks, Addison-Wesley, 1986
- [Iaco] Ed Iacobucci: Das OS/2-Buch, McGraw-Hill, 1989
- [IOBoard] Pflichtenheft AT-IOBOARD/ST506, Nixdorf Computer AG
- [Klein] Andy Klein: Writing MS-DOS Device Drivers in C, Dr. Dobb's Journal, September 1987
- [KrMiWi] J. Krantz, A. Mizell, R. Williams: OS/2 für Anwender und Systementwickler, Vieweg, 1988
- [LLC] IEEE Standards for Local Area Networks: Logical Link Control, IEEE Standards Board New York, 1984
- [Lupp] Alfred Lupper: Ein Transaktionsdienst über ISDN-Kanäle mit Protokollsubstitution auf der Link-Ebene, Diplomarbeit, Universität Augsburg, 1989
- [MUSAC] MUSAC (Multipoint Switching and Conferencing Unit) Data Sheet Rev. 1.0 12/89, Siemens AG
- [NDIS] Network Driver Interface Spezifikation, 3Com Corporation, Santa Clara California, Manual Part No. 6406-00
- [ODI] Open Data-Link Interface, Novell Inc., Provo 122 East 1700 South, UT 84606
- [OnDo] J. O. Onunga, R. W. Donaldson: Performance Analysis of CSMA with Priority Acknowledgements (CSMA/PA) on Noisy Data Networks with Finite User Population, IEEE Transactions on Communications, Vol. 39, No. 7, July 1991
- [Patent\_1] Gerhard Partsch, Helmut Partsch, Peter Schulthess, Stefan Traub: Verfahren zur Mehrpunktvermittlung von paketorientierten Informationen in Kommunikationssystemen, Aktenzeichen P4329319.0
- [Patent\_2] Gerhard Partsch, Peter Schulthess, Alfred Lupper, Klaus Achatz: Verfahren zum signallaufzeitkompensierten Freigeben einer Mehrpunktverbindung, Aktenzeichen P4331039.7
- [Patent\_3] Gerhard Partsch, Franz Pachner, Werner Schwarz, Dieter Laufkötter: Verfahren zum Erkennen von Belegungen einer Mehrpunktverbindung in einem Kommunikationssystem, Aktenzeichen P4329318.2

- [Patent\_4] Gerhard Partsch: Verfahren zur Zugriffssteuerung in Mehrpunktverbindungen nach Kollisionen von Informationspaketen, Aktenzeichen P4331038.9
- [PD] Paket Driver Specification, Clarkson University, 675 Mass Ave, Cambridge, MA 02139
- [SCC] Zilog: Z8030/Z8530 SCC Serial Communications Controller, Technical Manual, 1983
- [SchFro1] Peter Schulthess, Konrad Froitzheim: Mehrpunktschaltungen in ISDN-Kommunikationsanlagen, ntz Bd. 41 (1988) Heft 9
- [SchFro2] Peter Schulthess, Konrad Froitzheim: Virtuelle Ringtopologien in ISDN-Kommunikationssystemen, ntz Bd. 43 (1990) Heft 9
- [Schick] Pietro Schicker: Datenübertragung und Rechnernetze, Teubner, 1983
- [Schind] Sigram Schindler: Aktiv gleich intelligent - passiv gleich dumm?, ISDNreport Nr. 1, Januar 1992
- [Schult] Peter Schulthess: Schaltungsanordnung für einen mehrpunktfähigen Raum-Zeit Koppelschaltkreis (Memory-Time-Switch), Patentschrift H04L 11/20 Akz P 3840 117.7, Anr 3756920
- [Schwa] Mischa Schwartz: Telecommunication Networks, Reading, Massachusetts, 1987
- [Tane] Andrew S. Tanenbaum: Computer Networks, Prentice Hall International, London, 1989
- [Tisch] Michael Tischer: PC intern, Data Becker, 1992
- [ToHu] F. A. Tobagi, V. B. Hunt: Performance Analysis of Carrier Sense Multiple Access with Collision Detection, Computer Networks 4, 1980
- [ToKl] F. A. Tobagi, L. Kleinrock: Packet Switching in Radio Channels, IEEE Transactions on Communications, December 1975
- [Turner] J. S. Turner: Design of a Broadcast Packet Switching Network, IEEE Transactions on Communications, Vo. 36 No. 6, pp. 734-743, June 1988
- [Ward] Robert Ward: Debugging C, Addison-Wesley, 1988
- [West] Raymond Westwater: Writing OS/2 Device Drivers, Addison-Wesley, ISBN 0-201-52234-9
- [Wybr] Dieter Wybranietz: Multicast-Kommunikation in verteilten Systemen, Springer Verlag Berlin Heidelberg 1990, ISBN 0-387-52551-3

- [X.20-32] CCITT: Data Communications Network Interfaces, Recommendations X.20-X.32, Red Book VIII.3, Genf, 1985
- [X.200] CCITT: Recommendation X.200, Reference Model of Open Systems Interconnection for CCITT Applications, Blue Book, Melbourne, 1988

[Schw] Misha Schwartz: Telecommunication Networks, Reading, Massachusetts, 1987

[Tan] Andrew S. Tanenbaum: Computer Networks, Prentice Hall International, London, 1989

[Tsch] Michael Tschöke: PC Netz, Data Becker, 1992

[Tob] E. A. Tobey, V. E. Heath: Performance Analysis of Carrier Sense Multiple Access with Collision Detection, Computer Networks 4, 1980

[Tob] E. A. Tobey, L. Kleinrock: Packet Switching in Radio Channels, IEEE Transactions on Communications, December 1975

[Tan] I. S. Turner: Design of a Broadcast Packet Switching Network, IEEE Transactions on Communications, Vol. 36 No. 6, pp. 754-763, June 1988

[Ward] Robert Ward: Debugging C, Addison-Wesley, 1988

[West] Raymond Westwick: Writing OS/2 Device Drivers, Addison-Wesley, ISBN 0-201-52234-9

[Wyd] Dietz Wyder: Praktische Kommunikation in verteilten Systemen, Springer Verlag Berlin Heidelberg 1990, ISBN 0-387-52251-3

[Schw2] Peter Schwinde, Konrad Fritzsche: Virtuelle Ringtopologien in ISDN-Kommunikationssystemen, Nr. 45 (1990) Heft 9

[Schw1] Peter Schwinde, Konrad Fritzsche: Mehrpunktanordnungen in ISDN-Kommunikationssystemen, Nr. 41 (1988) Heft 9

[Sch] Peter Schuster: Brandübertragung und Nachbarn, Technik, 1983

[Schind] Sigismund Schindler: Aktiv gleich intelligent - passiv gleich dumm, ISDN Forum, Nr. 1, Januar 1992

[Schul] Peter Schulze: Schaltungsanordnung für einen mehrprozessorigen Kram-Netz, Kopyschriften (Memory-Time-Switch), Patentschrift 1044 1120 A & B, 1940 117.7, Am 31.05.90

Liste der bisher erschienenen Ulmer Informatik-Berichte

Einige davon sind per FTP von [ftp.informatik.uni-ulm.de](ftp://ftp.informatik.uni-ulm.de) erhältlich

Die mit \* markierten Berichte sind vergriffen

List of technical reports published by the University of Ulm

Some of them are available by FTP from [ftp.informatik.uni-ulm.de](ftp://ftp.informatik.uni-ulm.de)

Reports marked with \* are out of print

- 91-01 *Ker-I Ko, P. Orponen, U. Schöning, O. Watanabe*  
Instance Complexity
- 91-02\* *K. Gladitz, H. Fassbender, H. Vogler*  
Compiler-Based Implementation of Syntax-Directed Functional Programming
- 91-03\* *Alfons Geser*  
Relative Termination
- 91-04\* *J. Köbler, U. Schöning, J. Toran*  
Graph Isomorphism is low for PP
- 91-05 *Johannes Köbler, Thomas Thierauf*  
Complexity Restricted Advice Functions
- 91-06\* *Uwe Schöning*  
Recent Highlights in Structural Complexity Theory
- 91-07\* *F. Green, J. Köbler, J. Toran*  
The Power of Middle Bit
- 91-08\* *V. Arvind, Y. Han, L. Hamachandra, J. Köbler, A. Lozano, M. Mundhenk, A. Ogiwara, U. Schöning, R. Silvestri, T. Thierauf*  
Reductions for Sets of Low Information Content
- 92-01\* *Vikraman Arvind, Johannes Köbler, Martin Mundhenk*  
On Bounded Truth-Table and Conjunctive Reductions to Sparse and Tally Sets
- 92-02\* *Thomas Noll, Heiko Vogler*  
Top-down Parsing with Simultaneous Evaluation of Noncircular Attribute Grammars
- 92-03 *Fakultät für Informatik*  
17. Workshop über Komplexitätstheorie, effiziente Algorithmen und Datenstrukturen
- 92-04\* *V. Arvind, J. Köbler, M. Mundhenk*  
Lowness and the Complexity of Sparse and Tally Descriptions
- 92-05\* *Johannes Köbler*  
Locating P/poly Optimally in the Extended Low Hierarchy
- 92-06\* *Armin Kühnemann, Heiko Vogler*  
Synthesized and inherited functions - a new computational model for syntax-directed semantics
- 92-07\* *Heinz Fassbender, Heiko Vogler*  
A Universal Unification Algorithm Based on Unification-Driven Leftmost Outermost Narrowing

- 92-08\* *Uwe Schöning*  
On Random Reductions from Sparse Sets to Tally Sets
- 92-09\* *Hermann von Hasseln, Laura Martignon*  
Consistency in Stochastic Network
- 92-10 *Michael Schmitt*  
A Slightly Improved Upper Bound on the Size of Weights Sufficient to Represent Any Linearly Separable Boolean Function
- 92-11 *Johannes Köbler, Seinosuke Toda*  
On the Power of Generalized MOD-Classes
- 92-12 *V. Arvind, J. Köbler, M. Mundhenk*  
Reliable Reductions, High Sets and Low Sets
- 92-13 *Alfons Geser*  
On a monotonic semantic path ordering
- 92-14\* *Joost Engelfriet, Heiko Vogler*  
The Translation Power of Top-Down Tree-To-Graph Transducers
- 93-01 *Alfred Lupper, Konrad Fritzscheim*  
AppleTalk Link Access Protocol basierend auf dem Abstract Personal Communications Manager
- 93-02 *M.H. Scholl, C. Laasch, C. Rich, H.-J. Schek, M. Tresch*  
The COCOON Object Model
- 93-03 *Thomas Thierauf, Seinosuke Toda, Osamu Watanabe*  
On Sets Bounded Truth-Table Reducible to P-selective Sets
- 93-04 *Jin-Yi Cai, Frederic Green, Thomas Thierauf*  
On the Correlation of Symmetric Functions
- 93-05 *K.Kuhn, M.Reichert, M. Nathe, T. Beuter, C. Heinlein, P. Dadam*  
A Conceptual Approach to an Open Hospital Information System
- 93-06 *Klaus Gaßner*  
Rechnerunterstützung für die konzeptuelle Modellierung
- 93-07 *Ullrich Keßler, Peter Dadam*  
Towards Customizable, Flexible Storage Structures for Complex Objects
- 94-01 *Michael Schmitt*  
On the Complexity of Consistency Problems for Neurons with Binary Weights
- 94-02 *Armin Kühnemann, Heiko Vogler*  
A Pumping Lemma for Output Languages of Attributed Tree Transducers
- 94-03 *Harry Buhrman, Jim Kadin, Thomas Thierauf*  
On Functions Computable with Nonadaptive Queries to NP
- 94-04 *Heinz Faßbender, Heiko Vogler, Andrea Wedel*  
Implementation of a Deterministic Partial E-Unification Algorithm for Macro Tree Transducers

- 94-05 *V. Arvind, J. Köbler, R. Schuler*  
On Helping and Interactive Proof Systems
- 94-06 *Christian Kalus, Peter Dadam*  
Incorporating record subtyping into a relational data model
- 94-07 *Markus Tresch, Marc H. Scholl*  
A Classification of Multi-Database Languages
- 94-08 *Friedrich von Henke, Harald Rueß*  
Arbeitstreffen Typtheorie: Zusammenfassung der Beiträge
- 94-09 *F.W. von Henke, A. Dold, H. Rueß, D. Schwier, M. Strecker*  
Construction and Deduction Methods for the Formal Development of Software
- 94-10 *Axel Dold*  
Formalisierung schematischer Algorithmen
- 94-11 *Johannes Köbler, Osamu Watanabe*  
New Collapse Consequences of NP Having Small Circuits
- 94-12 *Rainer Schuler*  
On Average Polynomial Time
- 94-13 *Rainer Schuler, Osamu Watanabe*  
Towards Average-Case Complexity Analysis of NP Optimization Problems
- 94-14 *Wolfram Schulte, Ton Vullings*  
Linking Reactive Software to the X-Window System
- 94-15 *Alfred Lupper*  
Namensverwaltung und Adressierung in Distributed Shared Memory-Systemen
- 94-16 *Robert Regn*  
Verteilte Unix-Betriebssysteme
- 94-17 *Helmuth Partsch*  
Again on Recognition and Parsing of Context-Free Grammars:  
Two Exercises in Transformational Programming
- 94-18 *Helmuth Partsch*  
Transformational Development of Data-Parallel Algorithms: an Example
- 95-01 *Oleg Verbitsky*  
On the Largest Common Subgraph Problem
- 95-02 *Uwe Schöning*  
Complexity of Presburger Arithmetic with Fixed Quantifier Dimension
- 95-03 *Harry Buhrman, Thomas Thierauf*  
The Complexity of Generating and Checking Proofs of Membership
- 95-04 *Rainer Schuler, Tomoyuki Yamakami*  
Structural Average Case Complexity

- 95-05 *Klaus Achatz, Wolfram Schulte*  
Architecture Independent Massive Parallelization of Divide-And-Conquer Algorithms
- 95-06 *Christoph Karg, Rainer Schuler*  
Structure in Average Case Complexity
- 95-07 *P. Dadam, K. Kuhn, M. Reichert, T. Beuter, M. Nathe*  
ADEPT: Ein integrierender Ansatz zur Entwicklung flexibler, zuverlässiger kooperierender Assistenzsysteme in klinischen Anwendungsumgebungen
- 95-08 *Jürgen Kehrer, Peter Schulthess*  
Aufbereitung von gescannten Röntgenbildern zur filmlosen Diagnostik
- 95-09 *Hans-Jörg Burtschick, Wolfgang Lindner*  
On Sets Turing Reducible to P-Selective Sets
- 95-10 *Boris Hartmann*  
Berücksichtigung lokaler Randbedingung bei globaler Zielloptimierung mit neuronalen Netzen am Beispiel Truck Backer-Upper
- 95-12 *Klaus Achatz, Wolfram Schulte*  
Massive Parallelization of Divide-and-Conquer Algorithms over Powerlists
- 95-13 *Andrea Mößle, Heiko Vogler*  
Efficient Call-by-value Evaluation Strategy of Primitive Recursive Program Schemes
- 95-14 *Axel Dold, Friedrich W. von Henke, Holger Pfeifer, Harald Rueß*  
A Generic Specification for Verifying Peephole Optimizations
- 96-01 *Ercüment Canver, Jan-Tecker Gayen, Adam Moik*  
Formale Entwicklung der Steuerungssoftware für eine elektrisch ortsbediente Weiche mit VSE
- 96-02 *Bernhard Nebel*  
Solving Hard Qualitative Temporal Reasoning Problems: Evaluating the Efficiency of Using the ORD-Horn Class
- 96-03 *Ton Vullinghs, Wolfram Schulte, Thilo Schwinn*  
An Introduction to TkGofer
- 96-04 *Thomas Beuter, Peter Dadam*  
Anwendungsspezifische Anforderungen an Workflow-Management-Systeme am Beispiel der Domäne Concurrent-Engineering
- 96-05 *Gerhard Schellhorn, Wolfgang Ahrendt*  
Verification of a Prolog Compiler - First Steps with KIV
- 96-06 *Manindra Agrawal, Thomas Thierauf*  
Satisfiability Problems
- 96-07 *Vikraman Arvind, Jacobo Torán*  
A nonadaptive NC Checker for Permutation Group Intersection
- 96-08 *David Cyrluk, Oliver Möller, Harald Rueß*  
An Efficient Decision Procedure for a Theory of Fix-Sized Bitvectors with Composition and Extraction

- 96-09 *Bernd Biechele, Dietmar Ernst, Frank Houdek, Joachim Schmid, Wolfram Schulte*  
Erfahrungen bei der Modellierung eingebetteter Systeme mit verschiedenen SA/RT-Ansätzen
- 96-10 *Falk Bartels, Axel Dold, Friedrich W. von Henke, Holger Pfeifer, Harald Rueß*  
Formalizing Fixed-Point Theory in PVS
- 96-11 *Axel Dold, Friedrich W. von Henke, Holger Pfeifer, Harald Rueß*  
Mechanized Semantics of Simple Imperative Programming Constructs
- 96-12 *Axel Dold, Friedrich W. von Henke, Holger Pfeifer, Harald Rueß*  
Generic Compilation Schemes for Simple Programming Constructs
- 96-13 *Klaus Achatz, Helmuth Partsch*  
From Descriptive Specifications to Operational ones: A Powerful Transformation Rule, its Applications and Variants
- 97-01 *Jochen Messner*  
Pattern Matching in Trace Monoids
- 97-02 *Wolfgang Lindner, Rainer Schuler*  
A Small Span Theorem within P
- 97-03 *Thomas Bauer, Peter Dadam*  
A Distributed Execution Environment for Large-Scale Workflow Management Systems with Subnets and Server Migration
- 97-04 *Christian Heinlein, Peter Dadam*  
Interaction Expressions - A Powerful Formalism for Describing Inter-Workflow Dependencies
- 97-05 *Vikraman Arvind, Johannes Köbler*  
On Pseudorandomness and Resource-Bounded Measure
- 97-06 *Gerhard Partsch*  
Punkt-zu-Punkt- und Mehrpunkt-basierende LAN-Integrationsstrategien für den digitalen Mobilfunkstandard DECT