TECHNISCHE UNIVERSITÄT MÜNCHEN

Institut für Informatik

Diplomarbeit

Entwicklung integrierter Lösungskonzepte zu ausgewählten Fragestellungen des X.25-Managements

Bearbeiter: Thomas Dirsch

Aufgabensteller: Prof. Dr. Heinz-Gerd Hegering

Betreuer: Dr. Sebastian Abeck
Anja Schuhknecht

Abgabedatum: 15. August 1995

Erklärung

Ich versichere, daß ich diese Diplomarbeit selbständig verfaßt und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

München, den 15. August 1995 .....

Inhaltsverzeichnis

1 EINLEITUNG 1

1.1 Problembeschreibung 1

1.2 Beschreibung eines Integrationsmodells für X.25-Management 2

1.3 Gliederung der Diplomarbeit 4

2 DAS X.25-NETZ UND SEINE EIGENSCHAFTEN 5

2.1 Allgemeines 5

2.1.1 Aufbau von paketvermittelnden Netzen 5

2.1.2 Paketvermittlung und virtuelle Verbindungen 6

2.1.3 X.121-Adressierung 7

2.1.4 X.25-Management 8

2.2 Komponenten in einem X.25-Netz 10

2.2.1 Switch 10

2.2.2 Packet Assembler / Disassembler (PAD) 11

2.2.3 Interworking Unit (IWU) 11

2.2.4 X.25-fähige Endgeräte 11

2.3 Das X.25-Schichtenmodell 11

2.3.1 Physical Layer 12

2.3.2 Link Layer 13

2.3.3 Packet Layer 15

3 ANFORDERUNGEN AN EIN MANAGEMENT-MODELL 17

3.1 Management-Aufgaben und -Ziele 17

3.1.1 Configuration Management 19

3.1.2 Fault Management 20
3.1.3 Accounting Management 23
3.1.4 Performance Management 24
3.1.5 Zusammenfassung 25

3.2 Beschreibungsanforderungen an das Management-Modell 25
3.2.1 Beschreibungssprache 25
3.2.2 Grundlagen des object oriented modeling 25

4 ENTWURF EINES INTEGRIERTEN X.25-MANAGEMENT-MODELLS 29

4.1 Plattformerweiterung durch ein X.25-Management-Modul 30
4.1.1 Relationen 33
4.1.2 Die Gruppe der Datenerfassungstypen 34
4.1.2.1 RS-232-like-Objektklasse 36
4.1.2.2 LAPB-Objektklasse 37
4.1.2.3 MLP-Objektklasse 39
4.1.2.4 PLP-Objektklasse 40
4.1.2.5 Interface-Objektklasse 42
4.1.2.6 Routing-Objektklasse 43
4.1.2.7 Peer-Objektklasse 43
4.1.2.8 Accounting-Objektklasse 44
4.1.3 Die Gruppe der Komponenten 44
4.1.3.1 Port-Objektklasse 45
4.1.3.2 RS-232-like-Port-Objektklasse 46
4.1.3.3 X.25-Port-Objektklasse / X.75-Port-Objektklasse 46
4.1.3.4 Module-Objektklasse 47
4.1.4 Die Gruppe der X.25-Geräte 48
4.1.4.1 X.25-Device-Objektklasse 48
4.1.4.2 Switch-Objektklasse 49
4.1.4.3 IWU-Objektklasse 50
4.1.4.4 STE-Objektklasse 50
4.1.4.5 PAD-Objektklasse 51
4.1.4.6 PAD-Terminal-Objektklasse 51
4.1.5 Die Gruppe der Verbindungstypen 52
4.1.5.1 Link-Objektklasse 53
4.1.5.2 RS-232-like-Link-Objektklasse 54
4.1.5.3 LAPB-Link-Objektklasse 54
4.1.5.4 PLP-Link-Objektklasse 55
4.1.6 Die Gruppe der Netzeinheiten 56
4.1.6.1 X.25-Net-Objektklasse 56
4.1.6.2 X.25-Subnet-Objektklasse 56

4.2 Applikationen 57
4.2.1 Fault Management 57
4.2.1.1 Objektdarstellung 57
4.2.1.2 Verwendung einer Datenbank 60
4.2.1.3 Management-Konzept 60
4.2.2 Accounting Management 62

5 WEGE DER DATENINTEGRATION 64
5.1 SNMP-Proxy-Agent 64
5.2 Beschreibung eines Schnittstellen-Moduls 65
5.2.1 Aufgaben 65
5.2.2 Aufbau 66

6 UMSETZUNG AUF DIE PLATTFORM SPECTRUM 70
6.1 Objektorientierte Elemente der Plattform 70
6.2 Realisierung des X.25-Management-Moduls 72
6.2.1 Benötigte Werkzeuge 72
6.2.2 Umsetzung der Relationen 72
6.2.3 Umsetzung der Objektklassen in model types 73
6.3 Realisierung von Applikationen 75

7 AUSBLICKE 77

ANHANG A MODELLIERUNGSBEISPIELE 78

ANHANG B RELATIONSREGELN 80

LITERATURVERZEICHNIS 81

ABKÜRZUNGEN 84

Abbildungsverzeichnis

Abbildung 1-1 Integration eines X.25-Netzes. 3
Abbildung 2-1 Schema eines paketvermittelnden Netzes. 5
Abbildung 2-2 Methoden des X.25-Managements. 9
Abbildung 2-3 Schichten des X.25-Protokolls. 12
Abbildung 2-4 Aufbau eines HDLC frame [THORPE, S.35]. 14
Abbildung 2-5 MLP-Protokoll. 14
Abbildung 2-6 Lokale Bedeutung von VCI. 15
Abbildung 3-1 End-to-End-Verbindung: (a) internet-Umgebung; (b) einfache X.25-Netz-Umgebung. 18
Abbildung 3-2 Zusammenhang zwischen Endgeräten und X.25-Netz: (a) Gesamtbild; (b) Verbindungsbeispiel. 22
Abbildung 4-1 Sektoren des X.25-Management-Modells. 29
Abbildung 4-2 Objektbeschreibung: (a) Vererbung; (b) Relation. 30
Abbildung 4-3 Gesamtdarstellung der Objektklassen. 31
Abbildung 4-4 Netzlandschaft. 32
Abbildung 4-5 Relationsarten mit Beispielen. 33
Abbildung 4-6 Gruppe der Datenerfassungsobjekte. 35
Abbildung 4-7 LAPB-Objekt. 39
Abbildung 4-8 MLP-Objekt. 39
Abbildung 4-9 PLP-Objekt. 42
Abbildung 4-10 Gruppe der Port-Objekte. 45
Abbildung 4-11 Port-Objekt. 45
Abbildung 4-12 RS-232-like-Port-Objekt: (a) Vererbung; (b) Relation. 46
Abbildung 4-13 X.25- und X.75-Port-Objekt: (a) Vererbung; (b) Relation. 47
Abbildung 4-14 Module-Objekt. 47
Abbildung 4-15 Gruppe der X.25-Device-Objekte. 48
Abbildung 4-16 X.25-Device-Objekt 49
Abbildung 4-17 Switch-Objekt: (a) Vererbung; (b) Relation. 50
Abbildung 4-18 IWU-Objekt: (a) Vererbung; (b) Relationen. 50
Abbildung 4-19 STE-Objekt. 51
Abbildung 4-20 PAD-Objekt: (a) Vererbung; (b) Relationen. 51
Abbildung 4-21 PAD-Terminal-Objekt 52
Abbildung 4-22 Gruppe der Link-Objekte. 52
Abbildung 4-23 RS-232-like-Link-Objekt: (a) Vererbung; (b) Relationen. 54
Abbildung 4-24 LAPB-Link-Objekt: (a) Vererbung; (b) Relationen. 55
Abbildung 4-25 PLP-Link-Objekt: (a) Vererbung; (b) Relationen. 55
Abbildung 4-26 X.25-Net-Objekt 56
Abbildung 4-27 X.25-Subnet-Objekt 57
Abbildung 4-28 virtual circuit: (a) Symboldarstellung; (b) Objektdarstellung. 59
Abbildung 4-29 Zusammenfassung der Accounting-Daten 62

Abbildung 5-1 Datenwege: (a) Befehle; (b) Abfragen; (c) Meldungen. 66
Abbildung 5-2 Grundstruktur des Schnittstellen-Moduls. 67
Abbildung 6-1 model type und inference handler. 71
Abbildung 6-2 Ableitungspunkte der neuen model types:
(a) Ableitung eines Datenerfassungstypen;
(b) Hauptvertreter der verschiedenen Gruppen. 75

Tabellenverzeichnis

Tabelle 2-1 Untergliederung der Physical Layer [THORPE, S.19]. 13
Tabelle 4-1 RS-232-like-Datenelemente. 37
Tabelle 4-2 Auszug der LAPB-Datenelemente. 38
Tabelle 4-3 Auszug der PLP-Datenelemente. 41
Tabelle 4-4 Auszug der Interface-Datenelemente. 43
Tabelle 4-5 Beispiel einer Routing-Tabelle. 43
Tabelle 4-6 Peer-Datenelemente. 44
Tabelle 4-7 Accounting-Datenelemente. 44
Tabelle 4-8 X.25-Device-Datenelemente. 48
Tabelle 4-9 PAD-Terminal-Datenelemente. 52
Tabelle 4-10 Link-Datenelemente. 53
Tabelle 4-11 RS-232-like-Link-Datenelemente. 54
Tabelle 4-12 LAPB-Link-Datenelemente. 55

1 Einleitung

1.1 Problembeschreibung

Mit der Entwicklung von Netzmanagement-Plattformen wurde der erste Schritt zur Zusammenfassung von Management-Aufgaben in einer gemeinsamen und einheitlichen Arbeitsumgebung unternommen. Zielvorstellung war es, sich von der Vielzahl der herstellereigenen Werkzeuge, mit denen ein Netzbetreiber konfrontiert wurde, zu lösen und ein heterogenes Netz zentral mit einem einzigen neuen 'Werkzeug', einer Management-Plattform, zu verwalten. Die Architektur einer Plattform ist derart ausgelegt, daß durch verschiedenartige Schnittstellen Netzdaten importiert und integriert werden können. Mit modularen Erweiterungen, die auf diesen Schnittstellen basieren, ist es möglich, neue Integrationsansätze und -methoden in der Plattform zu schaffen. Die Plattform selbst bietet Mechanismen, die die Lösung von Management-Aufgaben erleichtern bzw. übernehmen. Durch sie wird auch erreicht, ein Netz in seiner Gesamtheit zu erfassen und zu verwalten, da erst durch die Integration und Verfügbarkeit aller Netzdaten ein umfassendes Netzmanagement ermöglicht wird.

Dennoch steht vor der Realisierung einer derartigen Integrationsvorstellung zuerst das Entgegenkommen und die Einigung der Hersteller der Netzkomponenten, standardisierte Management-Informationen und Zugriffsmöglichkeiten anzubieten. Das Hauptproblem war zu Beginn der Integrationsphase die Heterogenität innerhalb der Netze und die damit verbundene Mannigfaltigkeit der Protokolle und Management-Daten. Die Lösung zeigte sich in der Entwicklung eines Standards in der TCP/IP-Welt. Unter dem Schlagwort SNMP (simple network management protocol [RFC1157]) setzte sich hier ein einfaches Management-Protokoll durch, das mittlerweile im LAN-Bereich die führende Rolle einnimmt. Es greift auf Management-Daten von Netzkomponenten zu, deren Informationsbestand ebenfalls einer Standardisierung unterworfen wurde (z.B. in Form der MIB-II [RFC1213]). Der Großteil der Netzkomponenten, die im LAN-Bereich eingesetzt werden, ist heute mit SNMP-Agenten ausgerüstet. Aufgrund dieses Standards ist eine Integration in eine Netzmanagement-Plattform über deren Kommunikationsschnittstelle einfach zu verwirklichen.

Das Einsatzgebiet von Plattformen beschränkt sich zur Zeit hauptsächlich auf den LAN-Bereich. Netze von Großunternehmen bewegen sich jedoch nicht nur in dieser Größenordnung. Durch die Einbindung von WAN-Techniken werden Teilnetze verbunden, deren geographische Entfernung beliebig weit sein kann. Eine der derzeit am weitesten verbreiteten Technologien auf diesem Sektor ist das paketvermittelnde Netz, dessen Zugang auf der Empfehlung X.25 der ITU beruht. Die Bemühungen, auch hier integrierte Lösungen für das Management zu finden, sind momentan noch im Anfangsstadium. Es dominieren immer noch isolierte, herstellerspezifische Werkzeuge, die mit proprietären Protokollen ihre nicht-standardisierten Datenbestände verwalten. Im Gegensatz dazu bieten manche Hersteller mittlerweile SNMP-Erweiterungen an, die aber z.T. nur oberflächliche und eingeschränkte Management-Möglichkeiten bieten. Der Trend bei den neuentwickelten X.25-Komponenten geht aber weiterhin zu einheitlichen Management-Ansätzen, um ein integriertes Management realisieren zu können.

Dennoch beruhen die derzeitigen Versuche noch auf dem Standpunkt des isolierten Managements der Netztypen, d.h. LAN- und WAN-Bereich werden als getrennt zu managende Netzsegmente behandelt. Beim Versuch, netzübergreifende Anschauungsweisen und Verwaltungsmethoden auf Plattformebene zu finden, stößt der Netzbetreiber momentan noch an Grenzen, da aus den bereits erwähnten Gründen derartige Integrationsmodelle bis jetzt noch sehr zaghaft entworfen worden sind.

Ein erster Ansatz wurde durch eine separate Integration geschaffen, d.h. es wurden innerhalb der Plattform die Bereiche LAN und WAN getrennt betrachtet und verwaltet. Die Verbundenheit der Netztechniken, die im Sinne des integrierten Netzmanagements eigentlich eine Einheit bilden sollten, wird dabei jedoch nur an den Randbereichen und Übergängen angedeutet. Übergreifendes Management, d.h. beispielsweise die kontinuierliche Überwachung von End-to-End-Verbindungen mit allen Zwischenstationen, die von einem LAN über ein WAN zu einem anderen LAN reichen, ist jedoch nicht möglich. Ein Beispiel für eine Teilintegration eines X.25-Netzes ist in [ICS] zu sehen. Es wurde dort die Überwachung von X.25-Links auf einer Plattform realisiert.

Das Ziel der Diplomarbeit soll die Entwicklung eines Lösungskonzepts zur Integration eines X.25-Netzes sein, um die soeben geschilderte Problematik zu beheben. Hierbei wird ein Management-Modell entworfen, das ein X.25-Netz in geeigneter Form beschreibt. Die Modellierung orientiert sich einerseits an physikalischen Netzelementen, andererseits werden auch Verbindungen, die in Zusammenhang mit den verschiedenen X.25-Schichten stehen, abgebildet. Das Modell berücksichtigt sowohl die Erfüllung von spezifischen X.25-Management-Aufgaben als auch Lösungen für Management-Bereiche, die unter einem netzübergreifenden Aspekt stehen. Es sollen neue Ansätze und Konzepte gefunden werden, die in der Lage sind, netzübergreifendes Management darzustellen und die Management-Möglichkeiten eines Netzbetreibers erweitern. Der Schwerpunkt liegt dabei in der Betrachtung des X.25-Sektors, den es in bestehende Management-Konzepte der LAN-Welt zu integrieren gilt. Eine Behandlung der Grundlagen der LAN-Welt erfolgt in dieser Arbeit nicht, da für integriertes Management in einer TCP/IP-Umgebung bereits ausreichend Literatur existiert und entsprechende Konzepte entwickelt wurden.

1.2 Beschreibung eines Integrationsmodells für X.25-Management

Mit der Integration eines X.25-Netzes in eine Plattformumgebung und der daraus resultierenden integrierten Sichtweise des Gesamtnetzes treten unweigerlich einige Probleme auf, die sich aus der unterschiedlichen Grundkonzeption der LAN- und WAN-Welt ergeben. Beide Netzarten haben als Aufgabenschwerpunkt den möglichst schnellen und zuverlässigen Datentransport. Da sie sich in ihrer räumlichen Ausdehnung unterscheiden, haben sie jedoch eine andere Grundstruktur. Während im lokalen Bereich z.B. auf ein gemeinsames, geteiltes Medium zugegriffen wird, werden Weitverkehrsnetze punktweise aufgebaut und verkabelt. Häufig sind WANs homogen, d.h. alle Geräte stammen vom gleichen Hersteller. In LANs hingegen trifft man vorwiegend auf Geräte unterschiedlichster Hersteller, die sich in ihren Management-Methoden und -Möglichkeiten unterscheiden. Zusätzlich werden in beiden Welten jeweils angepaßte Übertragungsformate und -protokolle verwendet, die sich in ihrer Technologie völlig voneinander abheben. Es stehen sich hier die Technik der verbindungsorientierten, paketvermittelnden und die der datagram-orientierten Datenübertragung gegenüber.

Aus den geschilderten Gründen ist es deshalb auch nicht notwendig, in allen Bereichen des Netzmanagements netzübergreifende Management-Konzepte zu entwerfen. Es sollen vielmehr nur zu einigen Problemstellungen, bei denen eine netzübergreifende Betrachtung sinnvoll ist, Lösungsansätze gefunden werden. Für den Entwurf des Modells bedeutet das, sowohl Aspekte zur einfachen Abbildung des X.25-Netzes auf die Plattformebene zu berücksichtigen als auch neue Möglichkeiten aufzuzeigen, die sich durch eine Einbettung des Modells in die existierende Netzwelt ergeben.

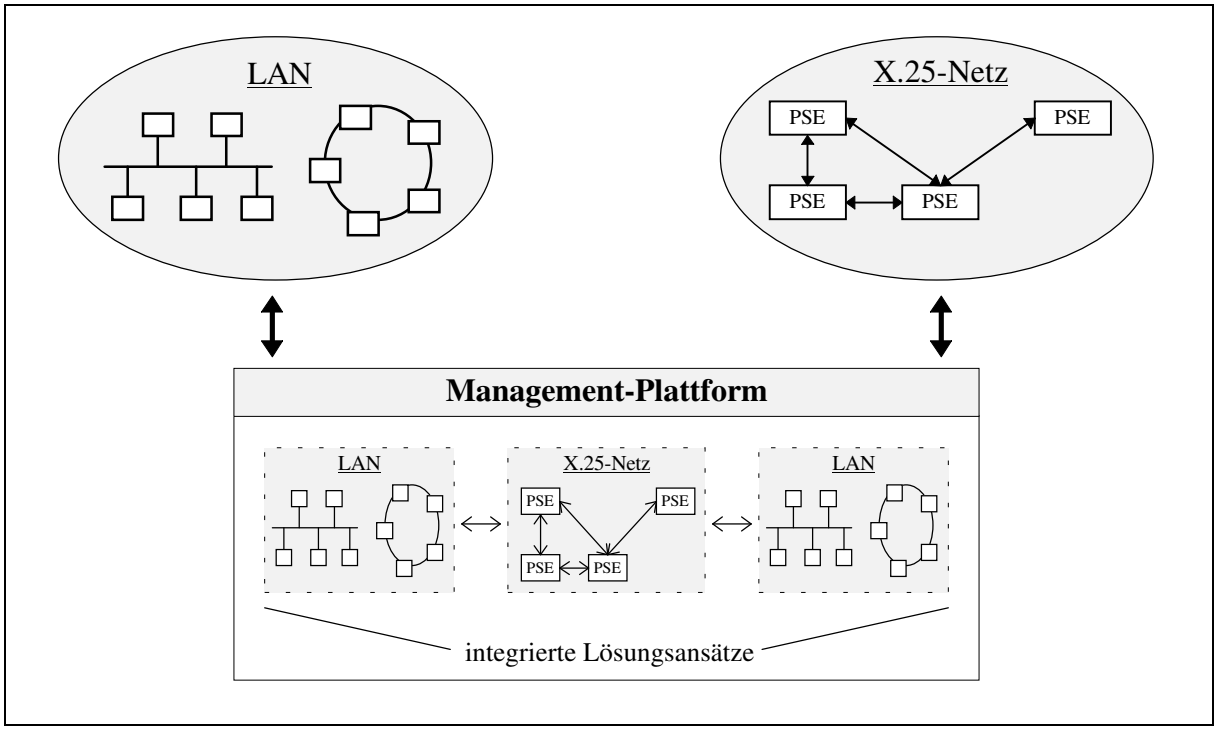


Abbildung 1-1 Integration eines X.25-Netzes.

Außer dem Bereich des Configuration Management, das sich hauptsächlich mit Vorgängen innerhalb einer konkreten Netztechnologie befaßt, werden in dieser Arbeit für die netzübergreifende Sicht die Aufgabenfelder des Fault und Accounting Management näher untersucht.

Es steht hier vor allem die Betrachtung der ‘End-to-End-Beziehung’ im Vordergrund. Dabei sollen Störungen in netzübergreifenden Kommunikationsverbindungen verfolgt und die sich daraus ergebenden Konsequenzen für alle beteiligten Netzteilnehmer verdeutlicht werden.

Eine Veranschaulichung für einen netzübergreifenden Ansatz kann im Fault Management gezeigt werden. Sowohl in der LAN- als auch WAN-Welt wird unter diesem Management-Begriff hauptsächlich die Überwachung von Komponenten verstanden. Eine Störung eines Koppelelements, z.B. Router oder Switch, kann zu Ausfällen von ganzen Netzsektoren führen. Betrachtet man die Störung nur innerhalb eines Netzes, kann unter Umständen die ausgefallene Komponente lokalisiert werden. Erst durch eine netzübergreifende Sichtweise ist es jedoch möglich, Auswirkungen des Störfalls auf das gesamte Netz zu untersuchen.

Ein immer mehr an Bedeutung gewinnender Bereich ist das Accounting Management. Die Feststellung und Zuordnung von entstehenden Kosten sowohl im netzinternen als auch im

netzübergreifenden Verkehr spielen für Betreiber von Großnetzen eine bedeutende Rolle. Sie bilden auch die Grundlage für Planung und Dimensionierung von Netzerweiterungen.

1.3 Gliederung der Diplomarbeit

Im folgenden wird eine Kurzübersicht über den weiteren Aufbau der Diplomarbeit gegeben.

Das Kapitel 2 „Das X.25-Netz und seine Eigenschaften“ erklärt die theoretischen Grundlagen von X.25-Netzen zum Verständnis der späteren Modellumsetzung. Es werden die Funktionsweise und der Aufbau von paketvermittelnden Netzen behandelt. Eine wichtige Rolle spielen hierbei die Art der Datenübertragung und in diesem Zusammenhang die virtuelle Verbindung. Weitere Themenbereiche sind die Vorstellung von derzeit angewandten Management-Formen und die Beschreibung von X.25-Netzkomponenten mit ihren Funktionalitäten. Mit einer Darstellung von Aufgaben und Funktionen der Schichten des X.25-Protokolls wird dieser Abschnitt abgeschlossen.

Nach den Grundlagen von X.25-Netzen werden in Kapitel 3 die Anforderungen konkretisiert, die von einem Management-Modell erfüllt werden müssen. Es werden unter anderem folgende Fragestellungen bearbeitet:

- Welche Bereiche des Managements müssen beachtet werden ?
- Wo gibt es Ansatzpunkte für netzübergreifendes Management ?
- Wie und womit soll das Modell beschrieben werden ?

Kapitel 4 enthält schließlich den Entwurf eines integrierten Management-Modells für X.25-Netze, dessen Anforderungen zuvor erläutert wurden. Es gliedert sich in die Teilbereiche der Erstellung einer Netzbeschreibung für ein X.25-Netz und der Entwicklung von Konzepten für Management-Applikationen, die auf dieser Beschreibung beruhen.

Mit den Möglichkeiten, wie die Daten des X.25-Netzes für das Management-Modell in die Plattform übertragen werden können, beschäftigt sich Kapitel 5. Es wird zum einen ein SNMP-Proxy-Agent betrachtet und zum anderen ein Eigenentwurf eines Schnittstellen-Moduls zur Datenintegration vorgestellt.

In Kapitel 6 „Umsetzung auf die Plattform SPECTRUM“ wird versucht, das entworfene Management-Modell auf die Plattform SPECTRUM umzusetzen. Es werden die von der Plattform bereitgestellten Möglichkeiten und Werkzeuge untersucht, um schließlich deren Eignung für die Modellumsetzung zu bestimmen. Unter anderem werden die objektorientierten Elemente der Plattform und SPECTRUM’s Model Type Hierarchy behandelt. Sie sind für die Realisierung und Einordnung der neuen Objektklassen wichtig, die im Management-Modell definiert wurden.

Zum Abschluß werden in Kapitel 7 „Ausblicke“ mögliche Ansätze für weitergehende Arbeiten behandelt.

2 Das X.25-Netz und seine Eigenschaften

2.1 Allgemeines

X.25-Netze gehören zur Gruppe der paketvermittelnden Netze. In den folgenden Unterkapiteln werden der allgemeine Aufbau, grundlegende Funktionsweisen, Methoden der Adressierung im öffentlichen Bereich und die zur Zeit praktizierten Management-Formen dieses Netztyps vorgestellt.

2.1.1 Aufbau von paketvermittelnden Netzen

Wide area networks (WAN) teilen sich nach der Art ihrer Vermittlungstechnik in die zwei Haupttypen leitungsvermittelnd (circuit-switched) und paket-vermittelnd (packet-switched) auf. Bei der Technik der Leitungsvermittlung wird zwischen zwei Netzteilnehmern eine physikalische Verbindung geschaltet, die von diesen exklusiv genutzt werden kann. Bekanntestes Beispiel hierfür ist das öffentliche Telefonnetz. Im Gegensatz dazu werden physikalische Verbindungen eines paketvermittelnden Netzes von mehreren Teilnehmern gleichzeitig verwendet. Ermöglicht wird diese Eigenschaft durch die Zerstückelung der zu übertragenden Daten in kleine Datenpakete, die mit einer Zieladresse versehen zu einem Empfänger verschickt werden.

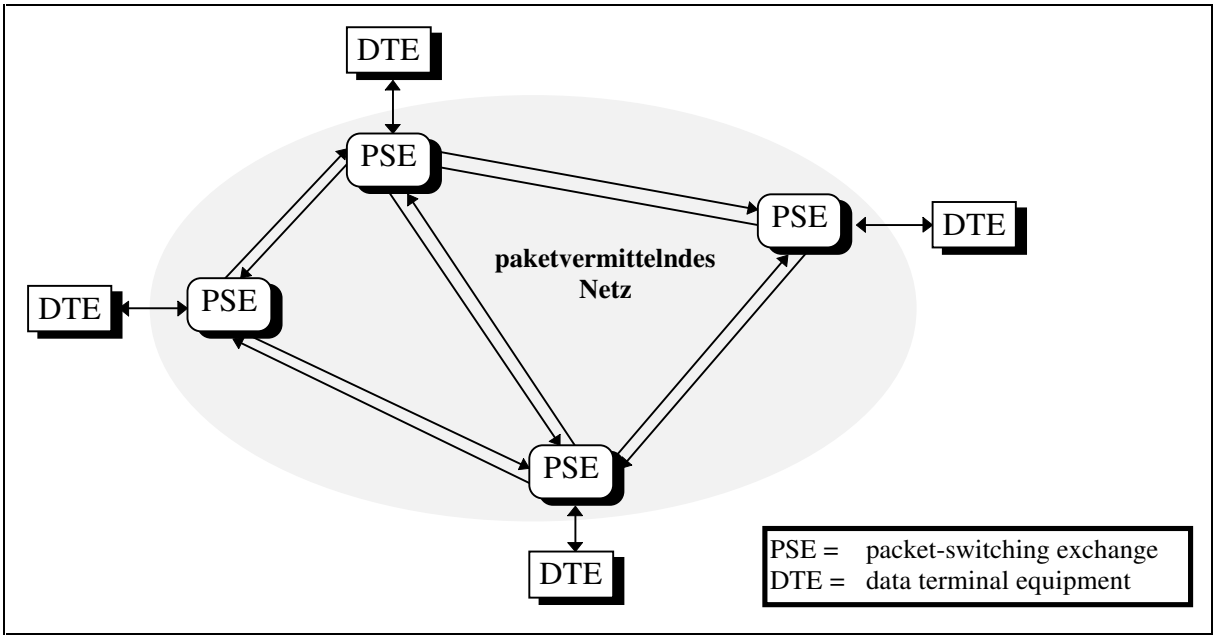


Abbildung 2-1 Schema eines paketvermittelnden Netzes.

In Abbildung 2-1 wird ein Beispiel für den Aufbau eines reinen paketvermittelnden Netzes gezeigt. Im Netz befinden sich Knotenpunkte, die alle durch physikalische Verbindungen mitein-

ander verknüpft sind. Da in der Regel keine besondere Ordnung bei der Verknüpfung vorliegt (z.B. jeder Knotenpunkt ist beliebig mit anderen Knotenpunkten verbunden), spricht man auch von einem vermaschten Netz. Die Hauptelemente befinden sich an den Knotenpunkten, die zugleich auch die Zugangspunkte zum Netz darstellen. Sie werden packet-switching exchange (PSE; switch) genannt und haben die Aufgabe, mit den Endeinheiten zu kommunizieren und Datenpakete an andere Knotenpunkte weiterzuleiten. Zu den Endeinheiten (data terminal equipment; DTE) werden alle Endgeräte gezählt, die das Zugangsprotokoll zum paketvermittelnden Netz beherrschen und somit eine eindeutige Endadresse besitzen. Diese können zum einen eigenständige Rechner, zum anderen aber auch Übersetzerkomponenten sein. Letztere ermöglichen, beispielsweise in Form eines packet assembler / disassembler (PAD), den Anschluß von einfachen zeichenorientierten Terminals oder als interworking unit (IWU) den Zugang von LAN-Segmenten an ein Paketnetz.

Außer paketvermittelnden Netzen, die gemäß dem oben genannten Schema aufgebaut sind, gibt es auch Mischformen. Hierbei werden Verbindungen zwischen Switches erst bei Bedarf z.B. über das Telefonnetz aufgebaut. Weitere Sonderfälle ergeben sich durch den Einsatz von Multiplexern, um Bandbreiten von physikalischen Verbindungen vollständig zu nutzen.

Unter den paketvermittelnden Netzen wird abhängig vom Betreiber nochmals zwischen öffentlichen und privaten Netzen unterschieden. Ein Beispiel für ein öffentliches Netz ist Datex-P, das von der Telekom unterhalten wird und für jedermann zugänglich ist. Größere Institute und Unternehmen besitzen jedoch ihre eigenen privaten Paketnetze. Die Switches, die Eigentum dieser Organisationen sind, sind mit gemieteten Telekom-Leitungen oder selbst verlegten Leitungen verbunden, soweit es sich um ein echtes Weitverkehrsnetz handelt. Beispiel hierfür sind die Netze der Deutschen Bahn AG oder der Energiekonzerne. Der Einsatz der X.25-Technologie ist jedoch auch im lokalen Bereich, z.B. zum Aufbau eines backbone, möglich.

2.1.2 Paketvermittlung und virtuelle Verbindungen

Der Vorgang der Paketvermittlung beginnt an einem Zugangspunkt, an dem Datenpakete mit einer Zieladresse abgeschickt werden. Der lokale Switch, der sich an diesem Zugangspunkt befindet, ermittelt den Empfänger des Paketes und entscheidet dann anhand einer Routing-Tabelle, auf welchem Ausgang (Link) er das Paket weiterleiten muß. In der Tabelle befinden sich Einträge mit Zieladressen von Endgeräten und die dazugehörigen Angaben, auf welchem Ausgang diese erreicht werden können. Der Weg eines Datenpaketes kann sich so über mehrere Switches erstrecken, bis letztendlich der Switch erreicht ist, an dem das Zielendgerät angeschlossen ist.

Ein paketorientiertes Netz bietet in der Regel zwei Arten von Diensten an. Mit einem datagram-Service besteht die Möglichkeit, eine einteilige Kurznachricht an einen Empfänger zu senden. Im allgemeinen wird jedoch das zweite Dienstangebot verwendet, wodurch einem Paketnetz auch die charakterisierende Bezeichnung eines verbindungsorientierten Netzes verliehen wird. Bei dieser Methode wird vor einer Datenübertragung zuerst eine virtuelle Verbindung (virtual circuit) zwischen den beteiligten Endgeräten aufgebaut. Der Ablauf eines Verbindungsaufbaus soll im folgenden in seinen Einzelschritten dargelegt werden.

Als Ausgangssituation wird angenommen, daß ein Endgerät A Daten zu einem anderen Endgerät B übertragen möchte. Der erste Schritt von Endgerät A besteht in der Übermittlung eines

call-request-packet an seinen lokalen Switch, mit dem es verbunden ist. Dieses beinhaltet die Zieladresse des Endgeräts B, einen eindeutigen virtual circuit identifier (VCI) und bestimmte Vorgaben (facilities). Unter dem Begriff facilities versteht man Bedingungswerte, wie sich die aufzubauende Verbindung verhalten und welche Eigenschaften sie haben soll. Dabei kann es sich beispielsweise um Geschwindigkeits-, Kostenvorgaben oder ähnliche Parameter handeln. In ihnen sind auch quality of service Parameter (QoS) enthalten.

Der Switch entscheidet anhand seiner Routing-Tabelle, der Zieladresse und den gewünschten facilities, an welchen Ausgangsport er das call-request-packet weiterleitet. Bei der Weiterleitung vergibt er außerdem einen neuen VCI für den nächsten Streckenabschnitt. Diese VCI-Umsetzung wird in sogenannten Link-Tabellen festgehalten. Auf diese Weise durchwandert das call-request-packet mehrere Switches, bis es an den gelangt, an dessen Port das entsprechende Endgerät B angeschlossen ist.

Das Endgerät B entscheidet, ob es den Verbindungswunsch akzeptiert. Wird die Verbindung angenommen, wird eine Bestätigung in Form eines confirm-packet an den Absender zurückgeschickt. Sobald jener dieses packet erhalten hat, gilt die virtuelle Verbindung als hergestellt. Die zu übermittelnden Daten werden in kleine Pakete zerstückelt und an den lokalen Switch übertragen, wobei nur noch Bezug auf die verwendeten VCIs genommen werden muß. Beim Empfänger werden die Pakete wieder zusammengesetzt, um die Daten in ihrer ursprünglichen Form wiederzuerhalten. Sind alle Datenpakete übertragen, wird die virtuelle Verbindung aufgelöst und die VCIs freigegeben.

Eine virtuelle Verbindung, die nach diesem Schema abläuft, wird switched virtual circuit genannt, da vor jeder Datenübertragung ein Verbindungsaufbau notwendig ist. Ein anderer Weg besteht darin, eine Verbindung einmal aufzubauen und dann bestehen zu lassen (permanent virtual circuit). Der Vorteil liegt in der Zeitersparnis durch den Wegfall der Prozedur der Verbindungsherstellung. Nachteilig ist jedoch die feste Belegung von Ressourcen, d.h. der VCIs auf der gesamten Verbindungsstrecke.

Endgeräte haben nicht nur die Möglichkeit, eine einzige virtuelle Verbindung zu unterhalten. Mit einer einfachen physikalischen Verbindung zwischen Endgerät und Switch wird dem Benutzer ermöglicht, mit verschiedenen anderen Endgeräten, die an dieses paketvermittelnde Netz angeschlossen sind, Daten auszutauschen. Durch mehrere virtual circuits, die über eine einzige physikalische Leitung geschaltet werden, kann der Benutzer somit verschiedene Verbindungen zugleich betreiben.

2.1.3 X.121-Adressierung

Zur Identifizierung der Endgeräte, die an ein öffentliches paketvermittelndes Netz angeschlossen sind, ist eine einheitliche Festlegung der Geräteadressen erforderlich. Um insbesondere einen internationalen Netzverbund realisieren zu können, muß diese Normung von allen öffentlichen Netzbetreibern eingehalten werden. Isolierte Netze, die nur für unternehmensspezifische Zwecke genutzt werden und nicht in Kontakt mit öffentlichen Netzen stehen, sind von dieser Festlegung natürlich ausgenommen.

Unter der Bezeichnung X.121 wurde von der ITU eine Empfehlung eines internationalen Nummerierungsplanes für öffentliche Datennetze veröffentlicht [CCITT_BB3]. Entsprechend

dieser Vorgabe besteht eine Endgeräte-Adresse (international data number) aus einer fünf bis vierzehn-stelligen Zahlenfolge der Ziffern 0-9, deren Aufbau sich nach bestimmten Vorgaben und Kriterien zu richten hat.

Die Adresse besteht aus zwei Teilen. Der erste Teil enthält in codierter Form das Land und das darin betriebene öffentliche Paketnetz, zu dem die Adresse gehört. Mit dem zweiten Teil wird schließlich das Endgerät selbst innerhalb dieses Netzes identifiziert. Die Adresse setzt sich nach folgendem Schema zusammen:

International data number =

= data network identification code (DNIC) + network terminal number (NTN)

oder = data country code (DCC) + National Number (NN)

Der DNIC ist ein vier-stelliger Code. Er ergibt sich einerseits aus einer drei-stelligen Länderkennung (DCC), die von der ITU vergeben wird, und andererseits aus einer Ziffer, die für ein in diesem Land betriebenes Netz steht. Zur Identifizierung eines Geräts in diesem Netz dient die NTN, die eine Länge von bis zu zehn Stellen haben darf. Gibt es innerhalb eines Landes bereits ein eindeutiges integriertes Nummerierungsschema der dortigen Netzbetreiber, so kann sich die Adresse auch nur aus DCC und einer NN zusammensetzen, wobei letztere dann aus bis zu elf Stellen bestehen kann.

2.1.4 X.25-Management

Mit dem Management eines X.25-Netz wird zur Zeit hauptsächlich die Konfiguration und Überwachung der Switches in Verbindung gebracht. Im allgemeinen werden dabei drei verschiedene Methoden unterschieden:

- out-band management
- in-band management
- Mischformen

Das einfache out-band management zeichnet sich durch den exklusiven Zugriff auf Management-Daten über einen separaten Management-Port eines Switches aus. Konfigurationseinstellungen und Auszüge aus den Daten können dort zumeist durch den Anschluß eines Terminals vorgenommen werden.

Eine weitaus komfortablere Methode stellen Switches mit in-band management zur Verfügung. Der Zugriff auf die Management-Daten erfolgt über den vorhandenen Netzverbund. Mit proprietären Management-Werkzeugen, die über herstellerspezifische Protokolle mit den Switches kommunizieren, werden Konfiguration und Verwaltung des Netzes vollzogen. Voraussetzung hierfür ist jedoch ein homogenes Netz, d.h. ein Netz, dessen Switches alle vom gleichen Hersteller stammen. Eine Weiterentwicklung wurde durch die Schaffung von Proxy-Agenten erreicht. Mit diesen können alle Management-Daten eines X.25-Netzes durch ein standardisiertes Management-Protokoll (z.B. SNMP) konfiguriert und abgerufen werden. Der Kontakt des Agenten zum X.25-Netz erfolgt jedoch weiterhin über ein Protokoll des Switch-Herstellers. Die Verfügbarkeit der Daten über Proxy-Agenten ermöglicht die Entwicklung von integrierten

Management-Lösungen auf Management-Plattformen, die in der Regel mit den standardisierten Protokollen arbeiten.

Eine weitere Management-Form besteht in der Mischung der beiden vorgestellten Methoden. Hierbei wird auf einen Management-Port eines Switches zugegriffen und über remote controling, d.h. wieder mittels eines herstellerspezifischen Protokolls, die Steuerung und Verwaltung der übrigen Switches erreicht. Zumeist stellen Switches auch Kombinationen dieser Management-Methoden zu Verfügung. In Abbildung 2-2 werden die drei Methoden nochmals verdeutlicht.

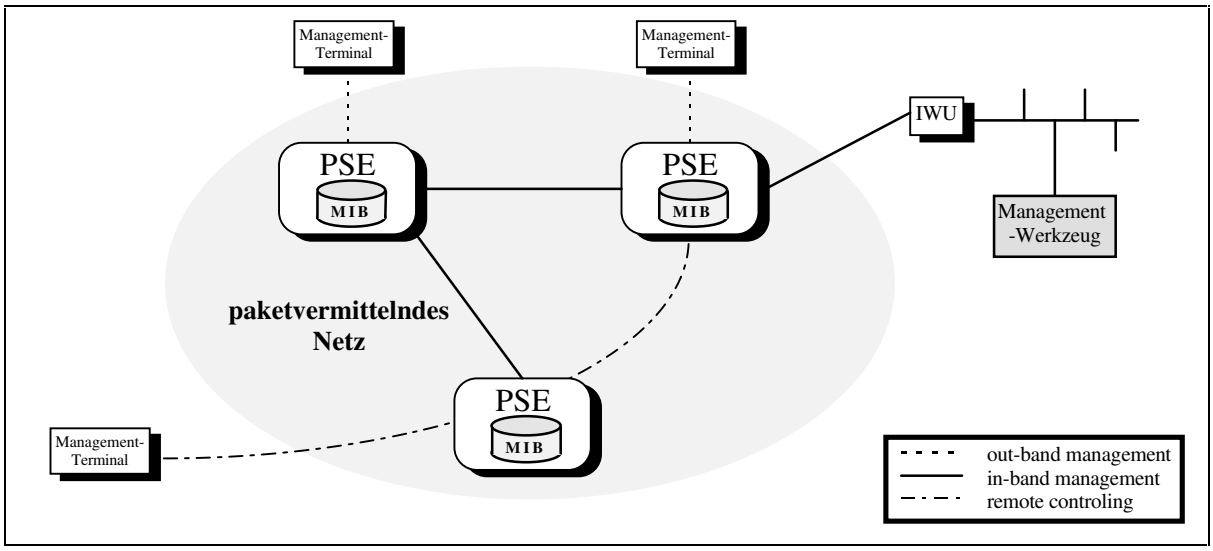


Abbildung 2-2 Methoden des X.25-Managements.

Neben der allgemeinen Verfahrensweise des Managements, bei der verschiedene Parameter gesetzt und Werte überwacht werden, bieten die meisten Switches einen eigenständigen Mechanismus zur Benachrichtigung bei aufgetretenen Störfällen an. Hierbei werden selbständig vom Switch Meldungen an dessen Management-System gesendet. Je nach Art der verwendeten Management-Methode kann es sich um einen speziellen Management-Port des Switches mit Drucker- oder Terminalanschluß oder ein externes Management-Werkzeug handeln. Die Meldungen beziehen sich auf Fehler und Störungen beim Verbindungsaufbau von einzelnen Links oder auf Ereignisse, die Änderungen des Systemzustandes betreffen.

Die Daten, die innerhalb eines Switches verwaltet werden, erstrecken sich im allgemeinen auf folgende Themengebiete:

- Konfiguration des Systems und der angeschlossenen Links
- Routing- und Link-Tabellen
- Daten von speziellen Erweiterungssteckkarten (z.B. PAD-, Gateway-Steckkarte)
- Management-Daten:
  In diesen Bereich fallen Status- und Statistikdaten, die einzelne Links oder virtuelle Verbindungen betreffen. Teilweise werden auch Daten für das Accounting Management bereitgestellt.

In der Regel sind die Angaben und Bezeichnungen der Daten innerhalb dieser Gebiete herstellerspezifisch. Zur Erfüllung von Management-Aufgaben ist deshalb gewöhnlich eine komplette Neueinarbeitung in das Management-System des jeweiligen Herstellers erforderlich.

2.2 Komponenten in einem X.25-Netz

In einem X.25-Netz gibt es spezifische Komponenten, mit denen zum einen das Netz aufgebaut und zum anderen der Zugang generell erst ermöglicht wird. Die Switches repräsentieren die Knotenpunkte eines paketvermittelnden Netzes. Die Endeinheiten PAD, IWU und X.25-fähige Endgeräte können direkt an dieses Netz angeschlossen werden. Die folgenden Unterkapitel beschäftigen sich mit dem Aufbau und der Funktionsweise dieser Komponenten.

2.2.1 Switch

Die Switches sind die Hauptkomponenten eines paketvermittelnden Netzes. Sie sind mit physikalischen Leitungen verbunden und bilden eine eigenständige Netzstruktur. Ihre Aufgabe liegt in der Weiterschaltung von ankommenden Paketen an die betreffenden nächsten Switches oder Endeinheiten und in der Verwaltung der virtuellen Verbindungen, auf denen diese Pakete übertragen werden.

Der Aufbau dieser Geräte ergibt sich gewöhnlich aus einzelnen Steckmodulen mit unterschiedliche Anschlußarten, die in einem zentralen Schrankgehäuse individuell nach den Bedürfnissen und Anforderungen des Netzbetreibers zusammengestellt werden können. Externe Endeinheiten werden über X.25-Ports angeschlossen, d.h. über Schnittstellen, auf denen das Zugangsprotokoll X.25 verwendet wird. Da ein Switch den Zugangspunkt zum Netz darstellt, werden von den Herstellern auch Module angeboten, auf denen bereits Endeinheiten, wie z.B. PAD oder Gateway, realisiert sind. Zusätzlich zu diesen multifunktionalen Einheiten werden jedoch auch einfache Kompaktgeräte verwendet, die mit einer festen Anzahl von Link-Anschlüssen ausgestattet sind und nur eine Basisfunktionalität zur Verfügung stellen.

Zur Erfüllung seiner Aufgaben werden vom Switch verschiedene Tabellen verwaltet. Anhand einer Routing-Tabelle wird die Entscheidung über die Wegewahl eines Datenpaketes getroffen. In ihr sind Adressen bzw. Adressmuster von allen angeschlossenen Teilnehmern des Netzes und die dafür zuwählenden Switch-Ausgänge gespeichert. Zusätzliche Kriterien, die ebenfalls in der Routing-Entscheidung berücksichtigt werden können, sind die Angabe von QoS-Parametern. Die Verwaltung der virtuellen Verbindungen geschieht in Form von Link-Tabellen, die für jeden einzelnen Link eines Switches geführt werden müssen. Sie enthalten Angaben über die bereits erwähnten VCIs. Für jeden Link wird ein Eintrag über einen ankommenden VCI und dazu ein Verweis auf den gewählten Ausgangs-Link mit einem neuem VCI erstellt. Genauere Angaben über Verwaltung und Anwendung der Tabellen finden sich in [HALSALL].

2.2.2 Packet Assembler / Disassembler (PAD)

Um zeichenorientierten Terminals oder generell Geräten, die über eine serielle Schnittstelle verfügen, den Zugang zum Paketnetz zu ermöglichen, wird ein Zwischenelement, das packet assembler / disassembler (PAD), eingesetzt. Dieses Gerät ist eine Art Protokoll-Konverter, an den mehrere Daten-Terminals angeschlossen werden können. Da diese nur einfache Zeichen übertragen und somit das X.25 Protokoll nicht beherrschen, übernimmt das PAD für sie diese Aufgabe. Es verpackt die einzelnen Zeichen in X.25-gerechte Pakete und verwaltet alle virtuellen Verbindungen dieser Terminals.

Das PAD übernimmt gleichzeitig auch die Aufgabe der Verteilung der virtuellen Verbindungen auf seine Ports. Ähnlich wie auch der Switch benötigt es hierfür eine Routing-Tabelle, um Adressen oder Namensbezeichnungen dem richtigen Port zuzuordnen.

2.2.3 Interworking Unit (IWU)

Diese Komponente ermöglicht den Anschluß eines LAN an das Paketnetz. Die geläufigere Bezeichnung für diese Einheit ist auch Gateway. Da an diesem Element LAN- und WAN-Welt mit ihren verschiedenen Adressierungsarten, Übertragungsprotokollen und -methoden aufeinander treffen, müssen hier entsprechende Umsetzungen vorgenommen werden. Datenübertragung zwischen LANs über ein WAN wird auf folgende Weise durchgeführt.

Erreicht ein frame eines LAN, dessen Adresse an ein über ein WAN erreichbares Ziel-LAN gerichtet ist, sein lokales IWU, so wird dort anhand einer Umsetzungstabelle die WAN-Adresse des remote IWU bestimmt. Danach wird zu diesem eine Verbindung aufgebaut und der frame durch encapsulation übertragen. Am remote IWU wird der frame wieder entpackt, zusammengesetzt und an seine Zieladresse weitergeschickt. Für den Netzbenutzer ist dieser Vorgang transparent, d.h. die Zwischenschaltung des WAN wird außer durch eine eventuelle Geschwindigkeitsminderung nicht wahrgenommen.

2.2.4 X.25-fähige Endgeräte

Unter die Bezeichnung X.25-fähige Endgeräte fallen alle Komponenten, die über einen eigenen X.25-Port verfügen und somit kein weiteres Zwischenelement zum Zugang an das X.25-Netz benötigen.

2.3 Das X.25-Schichtenmodell

Hinter dem Begriff ‘X.25’ verbirgt sich eine ITU-Empfehlung [CCITT_BB2], die den Zugang bzw. die Schnittstelle zu einem paketvermittelnden Netz beschreibt. Es werden die Versionen der Jahre 1980 und 1984 unterschieden, die jedoch nur geringfügig voneinander abweichen. Das X.25-Protokoll umfaßt drei Schichten, die im wesentlichen den ersten drei Schichten des ISO-OSI-Schichtenmodells entsprechen (siehe Abbildung 2-3). Es sei hier nochmals darauf hingewiesen, daß X.25 nur den Zugang zu einem paketvermittelnden Netz beschreibt und sich nicht mit Vorgängen innerhalb dieses Netzes befaßt. Es hat nur lokale Signifikanz.

Eine zu X.25 verwandte Protokollart wird zur Verbindung von Subnetzen eingesetzt. Das X.75-Protokoll verfügt über den gleichen geschichteten Aufbau. Die dritte X.75-Schicht, PLP, ist jedoch einfacher gestaltet und benötigt weniger Pakettypen als die X.25-PLP-Implementierung.

In den nächsten Unterkapiteln werden Inhalt und Aufgaben der einzelnen X.25-Schichten aufgezeigt. Im Zusammenhang mit den Protokollen werden die Begriffe data terminal equipment (DTE) und data circuit terminating equipment (DCE) verwendet. Sie stehen für die beiden Einheiten, zwischen denen das X.25-Protokoll eingesetzt wird. Insbesondere für die Abläufe der Link Layer wird diese eindeutige Zuordnung benötigt, da die dortigen frame-Prozeduren auf diesen Bezeichnungen in ihrem Adressierungsschema beruhen. In der Regel wird mit DCE der Switch und mit DTE das Endgerät bzw. die Endeinheit bezeichnet.

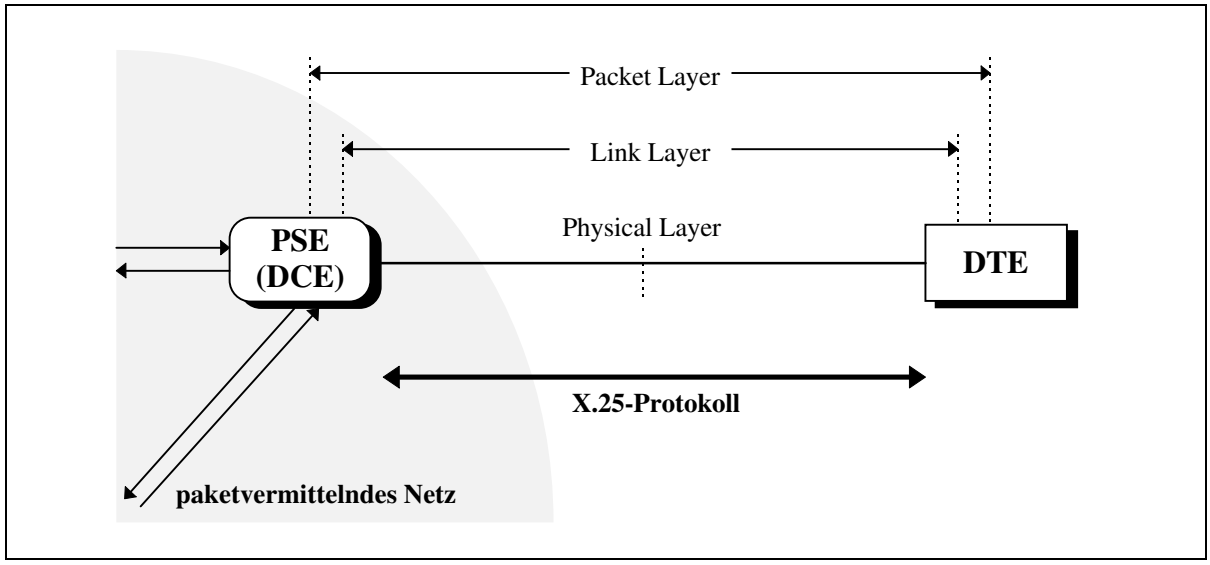


Abbildung 2-3 Schichten des X.25-Protokolls.

Auf Schicht 1, der Physical Layer, wird die physikalische Schnittstelle zwischen DTE und DCE beschrieben. Ausschlaggebend hierfür ist die ITU-Empfehlung X.21 und X.21 *bis*.

Die zweite Ebene von X.25 bildet ein HDLC-Protokoll (high-level data link control), das die Aufgabe der sicheren Datenübertragung übernimmt. Die Bezeichnung hierfür ist LAPB (link access procedures balanced), ein Protokoll, das auf der Link Layer für die Kontrolle und Steuerung der Kommunikationsverbindung zuständig ist.

Das packet layer protocol (PLP) stellt die dritte Schicht von X.25 dar. Seine Aufgaben sind die Kontrolle der virtuellen Verbindungen und der Pakettransport.

2.3.1 Physical Layer

Mit der untersten Schicht von X.25 wird die Schnittstelle zwischen DTE und DCE festgelegt. Es handelt sich hier hauptsächlich um Angaben über Steckverbinderarten mit deren Pinbelegungen, einzuhaltende Spannungswerte und vorgeschriebene Signalfolgen.

Für die Beschreibung der physikalischen Schnittstelle müssen folgende vier Bereiche abgedeckt werden:

- Funktionale Beschreibung
  (die Funktion der einzelnen Pins in einem Steckverbinder)
- Prozedurale Beschreibung
  (Ablauf und Bedeutung der einzelnen Signale)
- Elektrische Beschreibung
  (z.B. Spannungswerte, Geschwindigkeit, Signalverlauf)
- Mechanische Beschreibung
  (Art und Bauform der Steckverbindung)

Von ITU wurden dazu zwei Standards veröffentlicht, X.21 und X.21*bis*. In X.21 wird die digitale Schnittstelle zwischen DTE und DCE beschrieben, in X.21*bis* hingegen die analoge Version. Letztere Empfehlung war von ITU zu Beginn eigentlich nur als Übergangsstandard gedacht. Wegen der großen Verbreitung von V.28 (RS-232), der, wie in der nachfolgenden Tabelle zu sehen ist, einen Bestandteil von X.21*bis* bildet, wird dieser Standard aber zur Zeit immer noch am häufigsten verwendet.

	X.21		**X.21*bis***			
Data rate [bits/s]	<= 9600	> 9600	<=9600		>= 48 K	
Electrical	X.26 (RS-423)	X.27 (RS-422)	V.28 (RS-232)	X.26 (RS-423)	V.35	X.27 (RS-422)
Mechanical	4903, 15 pin	4903, 15 pin	2110, 25 pin	4902, 37 pin	2593, 34 pin	4902, 37 pin

Tabelle 2-1 Untergliederung der Physical Layer [THORPE, S.19].

2.3.2 Link Layer

Die Aufgabe der Link Layer ist es, einen zuverlässigen und fehlerfreien Datentransport auf der von der Physical Layer bereitgestellten Verbindung zu ermöglichen. Sie erfüllt hierbei folgende Funktionen:

- Datentransport über die bereitgestellte Verbindung
- Synchronisation von DTE und DCE
- Fehlerkontrolle

Durch das Protokoll der Packet Layer, die im folgenden Kapitel behandelt wird, werden die zu übertragenden Daten in kleine Datenpakete gleicher Länge (packets) zerstückelt. Sie beziehen sich jeweils auf einen virtual circuit. Für die Link Layer spielt der virtual circuit, zu dem ein packet gehört, keine Rolle. Innerhalb der Link Layer werden an ein packet noch zusätzliche Kontrollinformationen gehängt. Das resultierende Datenpaket auf dieser Schicht heißt frame,

worauf auch die ebenso geläufige Bezeichnung frame level für diese X.25-Schicht zurückzuführen ist.

Der Aufbau eines frame, sowie alle Vorgehensweisen zur Flußsteuerung, Fehlerkontrolle und zum Verbindungsaufbau basieren auf einer speziellen Version des bit-orientierten HDLC Protokolls im asynchronous balanced mode (ABM). Es wird hauptsächlich für vollduplex Verbindungen zwischen zwei Geräten eingesetzt. Im angegebenen Modus sind beide angeschlossenen Stationen gleichberechtigt und können somit sowohl Kommandos und als auch Antworten senden. Unter der Bezeichnung 'link access procedures balanced' (LAPB) veröffentlichte die ITU diese Empfehlung, die den Datenaustausch zwischen DTE und DCE bei X.25 festlegt.

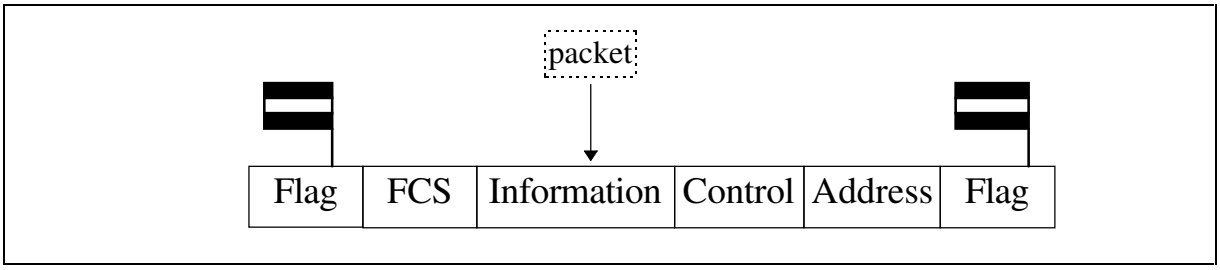


Abbildung 2-4 Aufbau eines HDLC frame [THORPE, S.35].

Zur Abgrenzung eines frame werden sogenannte flags verwendet, wie in Abbildung 2-4 zu sehen ist. Sie bestehen aus einer eindeutigen Bitfolge (01111110), die den Anfang und das Ende eines frame kennzeichnen. Die Fehlerkontrolle erfolgt mit dem CRC-Verfahren (cyclic redundance check). Mit einem Generatorpolynom wird hierbei eine frame check sequence (FCS) berechnet.

Im information-field befindet sich ein packet der Packet Layer, falls es sich um einen information frame und nicht um einen frame zur Steuerung der Schicht-2-Verbindung handelt. Die Unterscheidung zwischen verschiedenen frames und die Flußsteuerung sind Aufgabe des control fields. In diesem Feld stecken Informationen über die Art des frames (frames mit Datenpaketen oder frames, die Steuerzwecken dienen) und eine Numerierung, die auf einem sliding window mechanism beruht. Das address field gibt u.a. die Herkunft des frame an (DTE bzw. DCE).

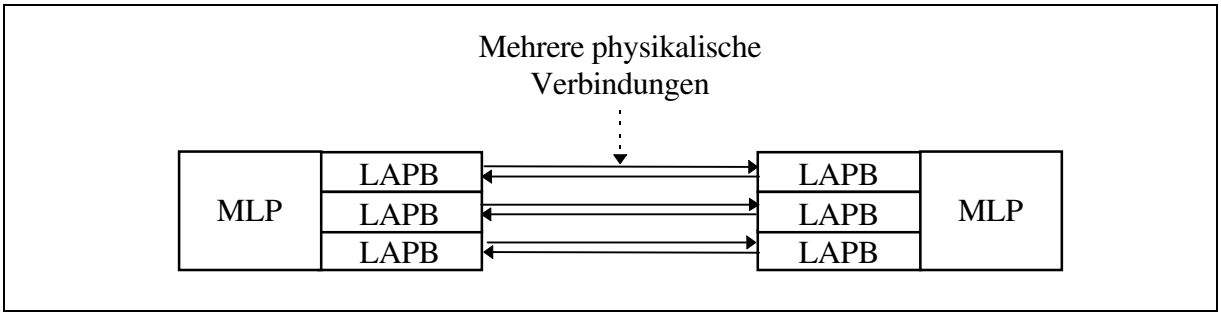


Abbildung 2-5 MLP-Protokoll.

LAPB kontrolliert die Übertragung von frames auf genau einem Link und wird deshalb auch single link procedure (SLP) genannt. Es besteht jedoch die Möglichkeit, auf der Link Layer von X.25 eine Erweiterung von LAPB, multilink procedure (MLP), einzusetzen. Abbildung 2-5 zeigt, daß diese Protokollform mehrere LAPB-Einheiten zugleich verwaltet und einsetzt. Sinn und Zweck dieser Vorgehensweise ist die Erhöhung des Datendurchsatzes und der Zuver-

lässigkeit der Verbindung, da bei Ausfall eines Links automatisch auf andere ausgewichen werden kann. Für die darüberliegende Packet Layer ist die Verwendung der MLP transparent.

2.3.3 Packet Layer

Das zentrale Element des X.25 Protokolls bildet die Packet Layer, die für die Verwaltung der virtual circuits / calls verantwortlich ist. Die Aufgaben dieser Schicht sind im einzelnen:

- Errichten und Löschen von virtual calls
- Management von permanenten virtual circuits
- Bereitstellung von Prozeduren zum Pakettransport
- Flußkontrolle
- Wiederherstellung und -aufbau bei Fehlerzuständen

Wie bereits zuvor beschrieben, werden die Kommunikationsverbindungen auf der Packet Layer zwischen Endeinheiten virtual circuits genannt, da es sich nur um dynamisch aufgebaute und nicht um echte physikalische Verbindungen handelt. Es wird zusätzlich noch zwischen switched und permanent virtual circuits (SVC und PVC) unterschieden. Während ein SVC für jede Datenkommunikation neu aufgebaut werden muß und nach deren Beendigung wieder gelöscht wird, bleibt der PVC ständig zwischen zwei DTEs bestehen. Bei letzterem entfällt der Prozeß des ständigen Verbindungsaufbaus, falls Daten übertragen werden sollen. Dem Geschwindigkeitsvorteil muß aber die permanente Reservierung der vorhandenen Ressourcen gegenübergestellt werden.

Zur Identifikation der einzelnen virtual circuits und Zuordnung der packets werden logische Kanalnummern (virtual circuit identifier - VCI) verwendet. Bei jedem Aufbau einer Schicht-3-Verbindung wird dem neuen virtual circuit eine eindeutige Nummer zugewiesen. Zu beachten ist hierbei, daß die Eindeutigkeit nur zwischen der jeweiligen DTE-PSE-Verbindungsstrecke (oder PSE-PSE, falls das X.25- oder X.75-Protokoll verwendet wird) gilt, also nur lokale Bedeutung besitzt. Innerhalb des X.25-Netzes und auf seiten der Ziel-DTE gelten wiederum andere VCIs, die in deren Umgebung eindeutig sind. Durch Tabellen in den PSEs werden die Kanalnummern für die jeweils nächste Strecke neu zugeordnet.

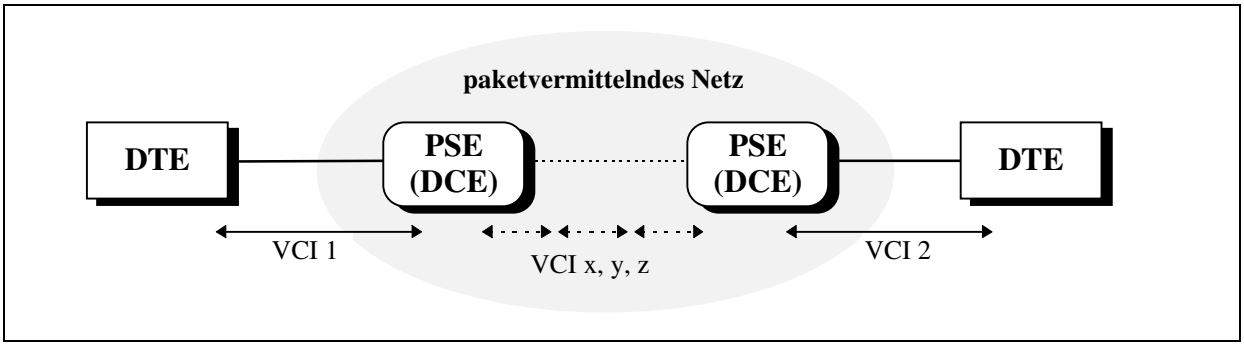


Abbildung 2-6 Lokale Bedeutung von VCI.

In der Packet Layer gibt es verschiedene Paketarten, die bestimmte Aufgaben erfüllen:

- Pakete für den Aufbau und die Beendigung einer Schicht-3-Verbindung
- Pakete für die Datenübermittlung
- Pakete zur Flußsteuerung, zum Zurücksetzen und Neustarten von virtual circuits
- Diagnosepakete (optional, d.h. wird nicht von allen Netzen unterstützt)

Die Flußsteuerung wird auch in dieser Schicht von einem sliding window mechanism übernommen.

Zum Verbindungsaufbau muß eine Zieladresse angegeben werden, zu der ein virtueller Kanal aufgebaut werden soll. Allen Teilnehmern an einem paketvermittelnden Netz sind eindeutige Adressen zugeordnet. Der Aufbau dieser Adresse ist in der ITU X.121-Empfehlung für öffentliche Netze festgelegt und besteht aus einem 14-stelligen Zahlencode.

Eine weitere Möglichkeit der Packet Layer von X.25 bietet die Angabe von facilities für virtuelle Verbindungen. Sie beinhalten unter anderem auch QoS-Parameter, die beispielsweise für Priorität und Kostenabrechnung der Verbindung relevant sind.

3 Anforderungen an ein Management-Modell

3.1 Management-Aufgaben und -Ziele

In diesem Kapitel sollen die Aufgaben und Ziele des Managements konkretisiert werden, die mit der Integration des X.25-Netzes in eine Plattformumgebung verfolgt werden. Im Gegensatz zur isolierten Sichtweise von LAN- und WAN-Welt, bei der nur Management-Aufgaben betrachtet werden, die sich innerhalb dieser Welten ergeben, werden hier auch zusätzliche Möglichkeiten einer netzübergreifenden Betrachtung aufgezeigt, die sich durch die X.25-Einbindung bieten.

Nach dem Systems Management Framework [ISO7498-4] der International Standards Organization (ISO) gliedern sich die Anforderungen des Open Systems Interconnection (OSI) Managements in folgende funktionelle Bereiche:

a) Fault Management

b) Accounting Management

c) Configuration Management

d) Performance Management

e) Security Management

Im Rahmen dieser Arbeit werden alle Bereiche außer dem Security Management betrachtet. Auf letzteren wird nicht weiter eingegangen, da er im Zusammenhang mit X.25-Netzen eine noch zu geringe Rolle spielt und dementsprechende Mechanismen nicht implementiert sind. Mit dem Management-Modell wird der Grundstock gelegt, um eine weitere Entwicklung von spezifischen Management-Anwendungen zu ermöglichen. Zuerst soll jedoch verdeutlicht werden, worin die Management-Grundlage in LAN- und WAN-Welt liegt.

In beiden Welten bieten die Netze ihre Transportleistung an. Dennoch verfügen die Netzarten über völlig unterschiedliche Strukturen und Grundkonzepte. Die LAN-Welt verwendet in der Regel datagram-Protokolle, die auf einem geteilten broadcast-Medium mit speziellen Zugriffsverfahren übertragen werden. Die Aufgabe des Netzmanagements ist die Koordination und Instandhaltung der Leistungsfähigkeit und Verfügbarkeit dieser Netzressource. Zwar zählt auch die Verwaltung von Dienstangeboten (z.B. in Form von Software oder Rechenleistung) und Peripheriegeräten zum Aufgabengebiet, da diese Leistungen jedoch alle auf der Nutzung der Transportkapazität des Netzes beruhen, wird als Schwerpunkt die Aufrechterhaltung des Datentransports genommen.

Bei der Betrachtung der WAN-Seite trifft man beim Aufbau auf eine Point-to-Point-Struktur, d.h. es gibt Knotenelemente, die miteinander abschnittsweise verbunden sind. Es ist zusätzlich notwendig, zwischen verschiedenen Netztypen zu differenzieren. Ein X.25-Netz kann einer-

seits als Weitverkehrsnetz eingesetzt werden, d.h. zum Überwinden von großen Entfernungen. Andererseits ist auch eine Nutzung der X.25-Technologie zum Aufbau eines backbone im lokalen Bereich oder für ein Rechnernetz möglich, das nur auf diese Technik setzt und über entsprechende Endgeräte verfügt.

Die Datenübertragungsrate bei WAN-Technologien ist in der Regel wegen der Gefahr von Störungen zumeist um einiges niedriger als im lokalen Bereich. Speziell bei der Entwicklung der X.25-Technologie stand der Gedanke der sicheren Übertragung über fehleranfällige Leitungen im Vordergrund. Es wurden deshalb auf den verschiedenen Schichtebenen Mechanismen eingesetzt, welche die Datensicherheit garantieren sollten. Das Dienstangebot des Netzes ist auf diese Übertragungssicherheit ausgerichtet und bietet mit seinem verbindungsorientierten Protokoll hohe Zuverlässigkeit. Das Management-Ziel eines WAN entspricht demjenigen des LAN. Auch hier liegt der Schwerpunkt in der maximalen Leistungsfähigkeit und Verfügbarkeit der gegebenen Ressource, der Transportleistung.

Obwohl sich beide Netztypen grundlegend im Aufbau ihrer Ressourcen unterscheiden, unterliegen beide den oben genannten Management-Bereichen, deren Funktionen bis jetzt hauptsächlich isoliert erfüllt werden. In manchen dieser Bereiche ist es aber an sich notwendig, Management-Aufgaben übergreifend, d.h. aus Sicht einer End-to-End-Beziehung zu betrachten. Die Bedeutung des Begriffes ‘End-to-End’ kann verschiedenartig interpretiert werden. Hierzu ist es erforderlich, innerhalb von einzelnen Management-Aufgaben zu differenzieren, welche Granularität benötigt wird. So kann beispielsweise für das Netzmanagement die Information über den Datenverkehr zwischen bestimmten WAN-Endpunkten wichtig sein, Angaben über die realen Endgeräte, die sich eventuell in einem LAN befinden und für den Datenverkehr verantwortlich sind, hingegen unerheblich sein. Im Accounting Management können wiederum gerade diese Angaben für eine direkte Kostenzuordnung ausschlaggebend sein.

In dieser Arbeit wird unter einer End-to-End-Beziehung eine Verbindung zum Datenaustausch zwischen Endgeräten verstanden. Der Ausdruck ‘Endgerät’ wird in einem X.25-Netz bereits mit X.121-adressierbaren Endpunkten assoziiert. Es kann sich hierbei um ein ‘echtes’ Endgerät (z.B. ein X.25-Host oder PAD) handeln oder aber auch um ein Verbindungselement, das den Übergangspunkt zu einem LAN bildet und eine entsprechende Adressumsetzung vornimmt.

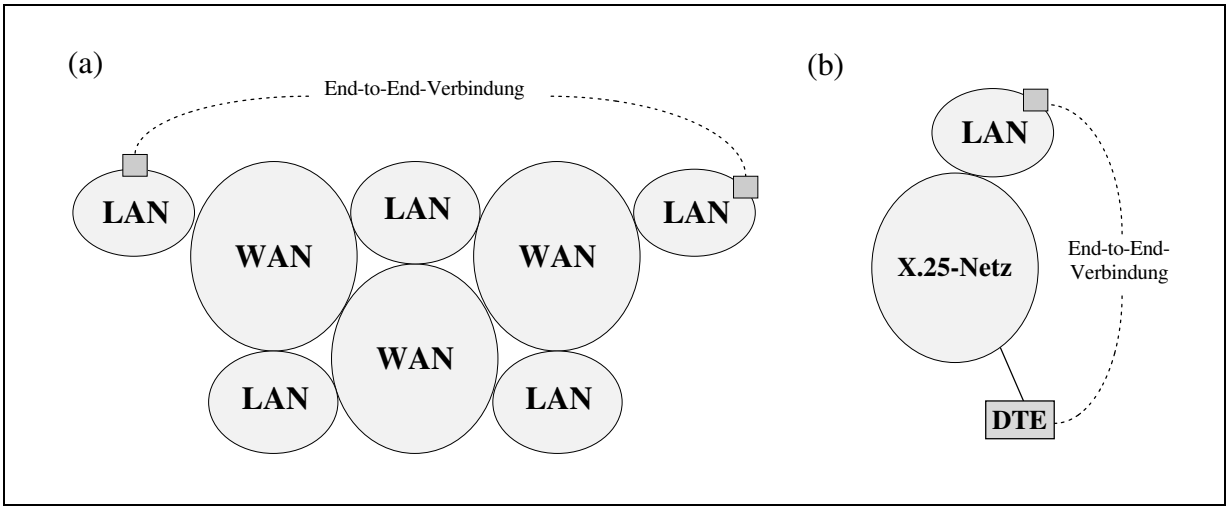


Abbildung 3-1 End-to-End-Verbindung: (a) internet-Umgebung; (b) einfache X.25-Netz-Umgebung.

Derartige Beziehungen können von einer einfachen Kombination LAN-WAN-LAN bis hin zu komplexen Verbindungsstrukturen reichen, die sich nicht nur über ein Verbindungsnetz sondern eine ganze Kette von Netzen erstrecken. Im allgemeinen wird das Arbeiten über mehrere Netze mit internetworking bezeichnet. Wie in Abbildung 3-1(a) zu sehen ist, werden manche Netze dabei nur als Durchgang benutzt, um letztendlich den Zielpunkt zu erreichen. Der einfachere Fall ergibt sich, falls die Verbindung nur über ein X.25-Netz vermittelt werden muß. In Abbildung 3-1(b) ist eine End-to-End-Verbindung von einem LAN-Endgerät zu einem X.25-Endgerät dargestellt.

Im weiteren werden nun, differenziert nach den einzelnen Bereichen, Management-Aspekte behandelt, die sich durch die X.25-Integration ergeben.

3.1.1 Configuration Management

Das Aufgabengebiet des Configuration Management ist breit gefächert. Es beinhaltet nach [ISO7498-4] folgende Funktionen:

a) das Setzen von Parametern, die Standardabläufe des Systems kontrollieren.

b) die Assoziierung von Namen mit managed objects.

c) die Initialisierung und Abschaltung von managed objects.

d) das Sammeln von Informationen über den momentanen Systemzustand.

e) das Erhalten von Meldungen über signifikante Änderungen des Sytemzustandes.

f) das Ändern der Systemkonfiguration.

Für die Konfiguration eines X.25-Netzes ist dessen Abbildung auf eine Management-Plattform erforderlich. Eine derartige Abbildung muß Elemente beinhalten, die einerseits die Sammlung und Auswertung von Netzdaten gestattet. Andererseits werden Elemente benötigt, die managed objects direkt repräsentieren (z.B. ein Switch), um deren Konfiguration für den Netzbetreiber anschaulicher zu machen und leichter in der Realität einordnen zu können. Die Informationsgrundlagen für diese Elemente sollten Standards entsprechen, damit eine direkte Integration der Daten gewährleistet ist. Hierzu kann auf Standard-MIBs (management information base) zurückgegriffen werden, die aus dem IAB (Internet Activities Board) stammen. Die Elemente, für deren Datenbestand keine Standard-MIB vorhanden ist, sollten weitestgehend einfach und allgemeingültig definiert werden, um eine zu starke Spezialisierung und eine damit verbundene eingeschränkte Verwendbarkeit zu vermeiden.

Bereits bei der Modellierung des X.25-Netzes sollten Mechanismen zur Vereinfachung von Konfigurationsaufgaben berücksichtigt werden. Durch eine Überordnung und Zusammenfassung von Netzelementen für das Setzen von Konfigurationsparametern kann dadurch dem Netzbetreiber eine Hilfe angeboten werden, um mögliche Konfigurationsfehler gleich im Vorfeld nicht entstehen zu lassen bzw. um mehrere Arbeitsschritte in einem Vorgang zu integrieren. Vorstellbar wäre hier ein Mechanismus, der das Verhalten einer Verbindungsstrecke zwischen zwei Switches kontrolliert und damit die Konfiguration von zwei Ports gleichzeitig beeinflußt.

Zusätzliche Elemente werden für die Einordnung in das bestehende Gesamtnetz benötigt. Hierunter fallen auch strukturierende Elemente, die das zu integrierende X.25-Netz in Teilsegmente untergliedern.

Die für die Abbildung des X.25-Netzes notwendigen Elemente müssen daher folgendermaßen gruppiert werden:

- Elemente für die Datenerfassung und für alle Datenbereiche eines X.25-Netzes.
- Elemente zur Beschreibung von Komponenten.
  Hierunter fallen die für X.25-Geräte ausschlaggebenden Ports.
- Elemente, die X.25-Geräte selbst darstellen.
- Elemente für die Erfassung der verschiedenen Verbindungsarten zwischen den X.25-Geräten.
- Elemente zur Strukturierung und Einordnung des X.25-Netzes.

3.1.2 Fault Management

Das Fault Management umfaßt die Aufspürung, Eingrenzung und Korrektur von anormalen Zuständen in einer OSI Umgebung. Hierunter sind nach [ISO7498-4] im wesentlichen folgende Funktionen zu verstehen:

a) das Erhalten und Untersuchen von Fehlermeldungen.

b) das Reagieren auf aufgespürte Fehlerzustände.

c) das Ausfindigmachen und Identifizieren von Störungen.

d) die Durchführung von Diagnosetests.

e) die Korrektur der Störung.

In einem X.25-Netz entstehen im allgemeinen Störungen durch den Ausfall von Verbindungen zwischen Switches. Ursachen dafür können Übertragungsfehler im Zusammenhang mit störanfälligen Datenleitungen, eine zeitweilige Überlastung oder ein Komplettausfall von Komponenten sein. Zur Erfüllung der Management-Aufgabe ist ein Mechanismus erforderlich, der diese Verbindungsausfälle erfaßt, weitermeldet und aufzeichnet. Desweiteren muß durch ihn eine Lokalisation der Störung möglich sein, um durch genauere Untersuchungen deren Ursache zu finden und sie gegebenenfalls zu beheben. Es handelt sich hierbei also um eine Verbindungs- und Komponentenüberwachung, wie sie z.T. bereits im LAN-Bereich durchgeführt wird.

Eine netzübergreifende Betrachtung im Bereich des Fault Managements stellt wesentlich komplexere Anforderungen. Ausgehend von einer einfachen End-to-End-Verbindung, die von einem LAN über ein WAN zu einem anderen LAN- oder WAN-Endgerät reicht, wären verschiedene Situationen vorstellbar.

- Ausfall eines Koppelelements in den betroffenen LAN-Segmenten (z.B. Hub, Bridge, Router, Steckkarte eines Routers).
- Ausfall eines Endgeräts in einer End-to-End-Verbindung.
- Ausfall eines Koppelelements innerhalb des WAN (z.B. Switch, Port eines Switches) und damit der Verlust einer Teilverbindungsstrecke.
- Probleme mit einzelnen Links (erhöhte Störanfälligkeit).

Eine Applikation des übergreifenden Fault Managements muß demnach über einen Mechanismus zum Aufspüren derartiger Ausfälle im gesamten Netzbereich verfügen. Es muß dabei festgestellt werden, welches Netzelement die Störung verursacht hat und in welchem Netzabschnitt der Fehler aufgetreten ist. Eine zusätzliche Voraussetzung ist die Berechtigung, diese Informationen überhaupt zu beziehen, falls die betroffenen Netze nicht nur einem Netzbetreiber allein unterstehen.

Da das Einsatzgebiet von Management-Plattformen zur Zeit in erster Linie der LAN-Bereich ist, wird davon ausgegangen, daß dort bereits entsprechende Melde- und Aufspürvorrichtungen vorhanden sind. Es genügt also, sich im weiteren dem X.25-Bereich zu widmen.

Im Falle einer X.25-Umgebung ist es notwendig, gezielte Informationen über die Wegewahl und den Aufbau eines X.25-Netzes zu besitzen. Ausschlaggebend sind hierfür zum einen die Routing-Tabellen, mit denen die Wegewahl getroffen wird. Durch sie wird festgelegt, auf welchem Weg eine Endeinheit erreicht werden kann. Zum anderen muß die Grobstrukturierung des Gesamtnetzes bekannt sein, d.h. der Zusammenhang zwischen den Endeinheiten des X.25-Netzes und dem X.25-Netz selbst mit seinen Knotenpunkten. Der Sachverhalt soll in Abbildung 3-2(a) verdeutlicht werden.

Der Ausfall einer Verbindungsstrecke z.B. zwischen Switch A und Switch B muß nicht unbedingt einen Ausfall der End-to-End-Verbindung zwischen LAN A und LAN B bedeuten. Bei entsprechender Konfiguration kann der Datenverkehr auf die anderen Switches umgeleitet und eine Beeinflussung der End-to-End-Beziehung vermieden werden. Der Ausfall hätte in diesem Beispiel nur lokale Bedeutung für das X.25-Netz. Im Unterschied dazu hätte der Ausfall des Großrechners sowohl lokale als auch netzübergreifende Auswirkungen. Lokal würde diesbezüglich heißen, daß der gesamte Endsektor in sich nicht einsatzfähig ist und mit den angeschlossenen Terminals nicht mehr gearbeitet werden kann. In einer netzübergreifenden Sichtweise bedeutet es jedoch auch, daß keine der anderen Endeinheiten auf den Großrechner mehr zugreifen kann. Jede End-to-End-Verbindung mit Teilnehmern aus der Großrechner-Endeinheit ist somit gestört.

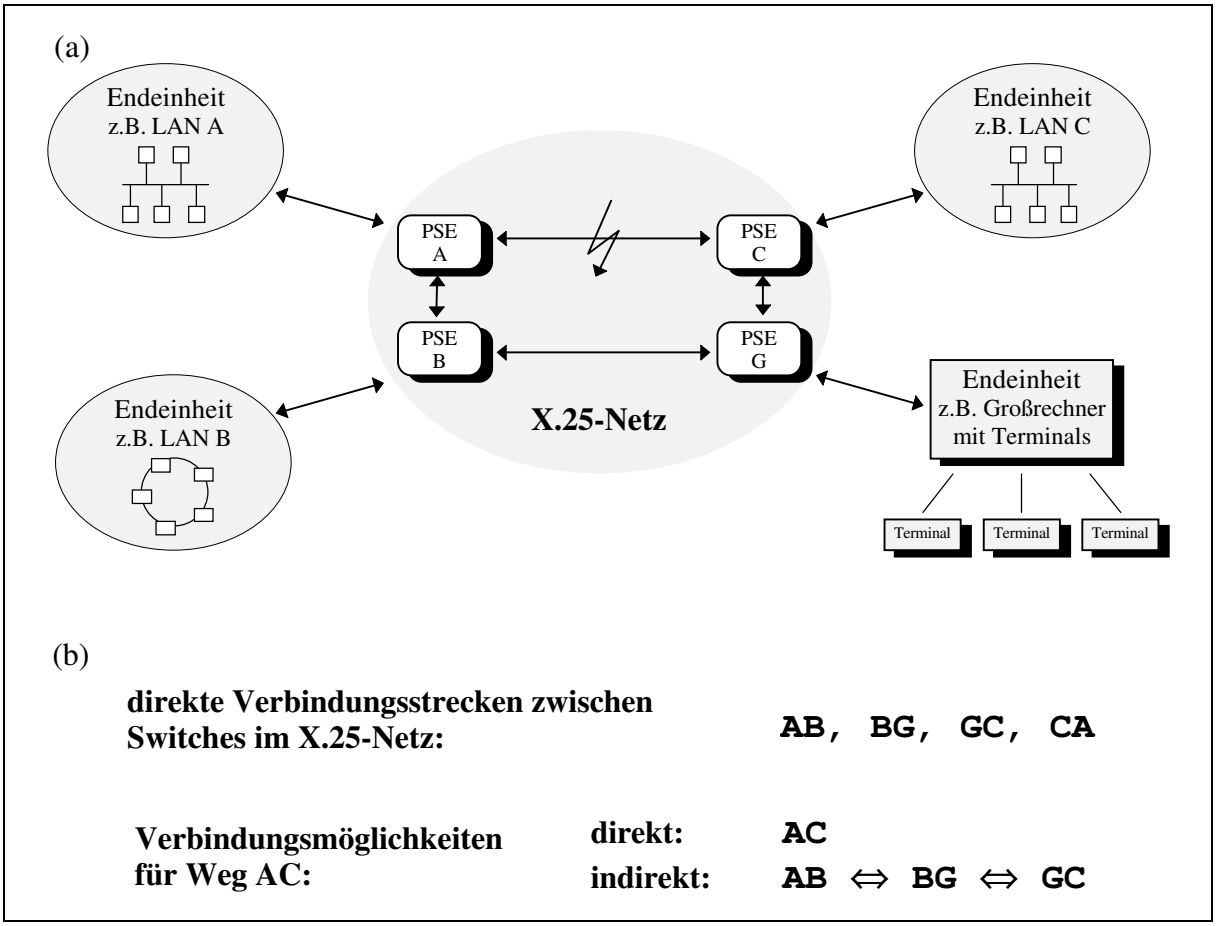


Abbildung 3-2 Zusammenhang zwischen Endgeräten und X.25-Netz: (a) Gesamtbild; (b) Verbindungsbeispiel.

Um eine derartige Schlußfolgerung treffen zu können, muß einem Management-System bekannt sein, wie Datenübertragungswege mit und in einem X.25-Netz zusammenhängen. Hierfür ist eine Netzbeschreibung erforderlich, die alle Verbindungsmöglichkeiten zwischen Knotenpunkten aufzeigt und über das Vorhandensein von Alternativen Aufschluß gibt.

Eine mögliche Beschreibungsmethode zeigt Abbildung 3-2(b). Durch das Erfassen aller Verbindungsstrecken zwischen den Switches in einem X.25-Netz wird eine einfache Netzbeschreibung gewonnen. In diesem Beispiel existieren nur vier Verbindungsstrecken. Bei der Suche nach Möglichkeiten, um von Switch A nach Switch C zu gelangen, wird gewöhnlich zuerst der direkte Weg über die Strecke AC gewählt. Ist diese Strecke jedoch zeitweise nicht verfügbar, so bietet sich noch ein weiterer indirekter Weg über die Switches B und G an, der sich durch transitive Zusammensetzung der Teilstrecken ergibt.

Für eine netzübergreifende End-to-End-Sichtweise ist außerdem eine Beschreibung der Endeinheiten mit ihren Anknüpfungspunkten an das X.25-Netz erforderlich. Hieraus muß zu erkennen sein, welche Endeinheiten mit welchen Knotenpunkten verbunden sind. Eine derartige Aussage wäre z.B.:

LAN A ist mit Switch A des X.25-Netzes verbunden.

Im Hinblick auf das Endgerät und den genauen Verbindungsweg kann diese Aussage natürlich weiter präzisiert werden:

IP-Host a.b.c.d des LAN A ist über den Router X mit Switch A des X.25-Netzes verbunden.

Auf diese Weise ergibt sich bereits eine nachvollziehbare Kette von Koppelelementen, die für die Kontrolle eines Abschnitts einer Verbindung relevant ist. Durch die Zusammensetzung von entsprechenden Beschreibungsketten aus den anderen Netzteilen können schließlich Aussagen über den Zustand einer End-to-End-Verbindung gemacht werden. Ein Ausfall eines Elements in dieser endgültigen Kette läßt Rückschlüsse auf mögliche Auswirkungen in der umschriebenen End-to-End-Beziehung zu.

3.1.3 Accounting Management

Mit Accounting Management wird die Bereitstellung einer Abrechnungsmöglichkeit für die Benutzung von Netzressourcen bezeichnet. Darunter fallen gemäß [ISO7498-4] folgende Funktionen:

a) die Information der Nutzer über aufgetretene Kosten oder verbrauchte Ressourcen.

b) die Möglichkeit zum Setzen von Abrechnungslimits und Tarifplänen.

c) die Möglichkeit der Zusammenfassung von Kosten, die in Zusammenhang mit einer Leistungserbringung stehen.

Die Ressource eines X.25-Netzes ist seine Transportleistung, die den Teilnehmern angeboten wird. Im Gegensatz zu einem circuit-switched network, bei dem Kosten anhand der Dauer der alleinigen Nutzung von Leitungen berechnet werden, werden bei einem packet-switched network Verbindungsleitungen in der Regel gemeinsam von mehreren Teilnehmern gleichzeitig genutzt. Die Kostenfeststellung kann sich somit aus der Menge der übertragenen Daten und den angeforderten Verbindungseigenschaften (facilities; QoS) ergeben.

Um die Nutzung des X.25-Netzes ermitteln zu können, müssen an den Zugangspunkten zum Netz Erfassungspunkte vorhanden sein, die den laufenden Datenverkehr festhalten. Da die Erfassung an den Zugangspunkten stattfindet, wird hier im allgemeinen nicht nach Endgeräten unterschieden, falls es sich bei der angeschlossene Endeinheit um ein LAN handelt. Die Switches unterstützen bereits die Sammlung von Accounting-Daten. In einer Tabelle werden dabei Informationen zu einer begrenzten Anzahl der zuletzt registrierten Verbindungen festgehalten. Darunter fallen zum Beispiel bei einem Switch der Firma Netcomm [NETCOMM_IG1]:

- die Adressen von Sender und Empfänger des virtual call.
- die Start- und End-Zeit der virtuellen Verbindung (somit auch die Dauer).
- die Anzahl der ausgetauschten Datenpakete.
- die Anzahl der ausgetauschten Bytes.
- eventuelle Gründe für einen Verbindungsabbruch oder Diagnosewerte.

Um eine weitere Aufgliederung nach Endgeräten zu erreichen, müssen innerhalb des LAN (z.B. am dortigen interworking unit) zusätzliche Messungen erfolgen. Ein netzübergreifender Aspekt des Accounting Management läßt sich durch die Verknüpfung der Daten der X.25-Erfassungspunkte mit denen, die im LAN vorliegen, gewinnen. Ein mögliches zukünftiges Accounting-Modell findet sich bei [SCHWERD], worin durch eine Zusammenführung von Abrechnungsdaten einzelner Dienste eine Kostenzuordnung auf den konkreten Verursacher ermöglicht wird.

Sollte sich die End-to-End-Verbindung über verschiedene Netze erstrecken, so kann eine Aufspaltung der angefallenen Kosten auf die einzelnen Netze sinnvoll sein. Unter Umständen kann auf diese Weise kostenintensiven Netzen ausgewichen oder deren Nutzung durch eingeschränkte Verbindungseigenschaften verbilligt werden.

Voraussetzung für ein derartiges Management im X.25-Netz ist die Verfügbarkeit über Mechanismen zur Messung des Datenverkehrs an den Zugangspunkten. Um eine weitere Aufteilung nach Endgeräten oder bestimmten Nutzern vorzunehmen, müssen ähnliche Verfahren in den jeweiligen Netzsegmenten vorhanden sein.

3.1.4 Performance Management

Performance Management ermöglicht die Auswertung des Verhaltens von Netzressourcen und der Effektivität der Kommunikationsaktivitäten. Es enthält nach [ISO7498-4] folgende Funktionen:

a) das Sammeln von statistischen Informationen.

b) das Unterhalten und Untersuchen von aufgezeichneten Systemstatusberichten.

c) das Feststellen der System-Performance unter natürlichen und künstlichen Bedingungen.

d) das Ändern von Systemarbeitszuständen, um Performance Management Aktivitäten durchzuführen.

In einem X.25-Netz sind die Auslastung der Links zwischen Switches und die Performance einer virtuellen Verbindung zwischen zwei Endeinheiten von Bedeutung. Entscheidend für die Durchsatzleistung eines Links ist dessen Bandbreite, d.h. seine maximale Übertragungsrate. Die Übertragungsleistung einer virtuellen Verbindung ergibt sich aus den Einzelleistungen der verwendeten Links. Ein weiterer Faktor, der sich auf die Leistungsfähigkeit auswirkt, ist eine hohe Übertragungsfehlerrate, die auf störanfällige Verbindungsleitungen zurückzuführen ist.

Große Performance-Schwankungen sollten mit dem Management-Modell feststellbar sein und eine Mitteilung an enstprechende Management-Abteilungen zur Folge haben. Wünschenswert wären Funktionen, die automatische Konfigurationsänderungen vornehmen, falls kritische Situationen im Netz entstehen (z.B. Änderung von Routing-Tabellen bei Überlastung von Teilstrecken).

3.1.5 Zusammenfassung

Für das integrierte Management von X.25-Netzen wurden die Bereiche Configuration, Fault, Accounting und Performance Management näher untersucht. Hierbei wurden sowohl Aspekte betrachtet, die ein X.25-Netz intern betreffen, als auch Überlegungen für netzübergreifende Sichtweisen behandelt. Der Schwerpunkt liegt zu Beginn im Configuration Management, da eine geeignete Netzbeschreibung für eine Abbildung auf eine Plattform entworfen werden muß. In dieser Beschreibung müssen bereits notwendige Mechanismen zur Erfüllung der Aufgaben der anderen Bereiche vorgesehen sein. Die weiteren Bereiche setzen auf den Daten dieser Abbildung auf und können als Applikationen realisiert werden. Obwohl in diesem Fall nur Fault und Accounting Management als wichtigste Punkte erfaßt wurden, ist auch auf den anderen Management-Gebieten eine ähnliche Verfahrensweise denkbar.

3.2 Beschreibungsanforderungen an das Management-Modell

3.2.1 Beschreibungssprache

Für die Entwicklung des Management-Modells ist eine geeignete Beschreibungssprache erforderlich. Mit ihr müssen Daten und Abläufe des Netzes nachgebildet und nachvollzogen werden können. Da viele verschiedene Teilbereiche mit unterschiedlichen Sinnzusammenhängen und Sachverhalten erfaßt werden müssen, ist es notwendig, eine Aufteilung des Gesamtsystems in kleine für sich unabhängige Elemente vorzunehmen. Über diese können einfache, allgemeingültige Aussagen bezüglich ihrer Struktur und ihres Verhaltens gemacht werden. Aufbauend auf diese Teilbeschreibungen können so komplexe Systeme mit ihren Zusammenhängen entwickelt werden.

Anforderungen an die Beschreibungssprache:

- Unterstützung der Bildung von unabhängigen Elementen mit spezifischen Eigenschaften und Verhaltensweisen.
- Möglichkeit zur Entwicklung von komplexen Systemen durch logische Verknüpfungen zwischen Elementen.

Die Forderungen werden von der Methode des object oriented modeling erfüllt. Es bietet eine Rahmenumgebung, in der große und komplexe Systeme anhand ihres Aufbaus und Verhaltens beschrieben werden können. Für die Abbildung des zu integrierenden X.25-Netzes soll diese Beschreibungssprache angewendet werden.

3.2.2 Grundlagen des object oriented modeling

Dieses Kapitel beschreibt die Grundideen des object oriented modeling nach [BAPAT]. Im Mittelpunkt steht das Objekt, das als Synonym für die zu beschreibende Sache gilt. Die Begrif-

fe, die für das Verständnis dieser Modellierungs- und Beschreibungsmethode wichtig sind, werden im folgenden in kurzer Form vorgestellt.

Abstraction

Mit Abstraction wird der Weg bezeichnet, sich nur auf die relevanten Details einer Problemstellung oder Objektbeschreibung zu konzentrieren. Hierbei wird eine Einteilung der Objekteigenschaften nach ihrer Wichtigkeit und Allgemeingültigkeit vorgenommen. Unter die Eigenschaften eines Objekts fallen seine Struktur und sein Verhalten. Wichtig für den Beschreibungsprozess ist die Beschränkung auf Aussagen, wodurch sich ein Objekt von anderen unterscheidet und was seine spezifischen Eigenarten oder Merkmale darstellen. Im allgemeinen stehen die getroffenen Aussagen dann für eine ganze Gattung von ähnlichen Objekten, die durch diese ausgezeichnet werden und sich damit von anderen abgrenzen. Die wesentlichen Charaktermerkmale eines Objekts werden durch seine Attribute und Funktionen bestimmt.

Encapsulation

Die Einkapselung eines Objekts beinhaltet die Abgrenzung und das Verbergen seiner inneren Struktur. Relevant sind nur die äußeren Verhaltensweisen eines Objekts, d.h. die Schnittstelle zu seiner Umgebung. Der Vorteil dieser Methode liegt in der Flexibilität. Objekte können in ihrer inneren Struktur geändert werden, ohne daß sich Konsequenzen auf andere Objekte ergeben, die in einer Beziehung zu ihnen stehen.

Classification

Objekte mit ähnlichen Eigenschaften werden einer Objektklasse zugeordnet. Ausschlaggebend für eine derartige Gruppierung können gemeinsame Eigenschaften im Verhalten, der Struktur oder aus beiden zugleich sein. Die Kriterien, nach denen eine Klassifizierung vorgenommen werden kann, sind jedoch nicht eindeutig festgelegt. Sie richten sich nach den Anwendungsabsichten und -zielen der zu entwerfenden Modellumgebung.

Ein Objekt ist ein Repräsentant seiner Klasse. Alle Repräsentanten einer Klasse besitzen die gleichen Eigenschaften. Attribute beschreiben die Struktur eines Objekts, d.h. sie stellen dessen Datenelemente dar. Sie werden mit einem Datentyp definiert, durch den die Art und die möglichen Werte des Attributinhalts vorgegeben werden. Im Gegensatz dazu wird ein Verhalten mit Objektfunktionen realisiert. Diese benötigen in der Regel Übergabeargumente und liefern als Ergebnis Ausgabeparameter.

Beispiel:

Objektklasse Switch:

mögliche Attribute:	Portanzahl
	Standort
mögliche Funktion:	Pakete vermitteln

Inheritance and Extension

Objektklassen können wiederum in übergeordneten Klassen zusammengefaßt werden. Dies geschieht durch Abstraktion von Klassen auf gemeinsame Eigenschaften. Einer neuen übergeordneten Klasse werden die entsprechend gleichen Eigenschaften zugeordnet. Betrachtet man diese Zusammenfassung umgekehrt, so kann auch gesagt werden, daß aus übergeordneten Basisklassen spezifischere Objektklassen mit neuen zusätzlichen Eigenschaften abgeleitet werden. Man spricht in diesem Zusammenhang auch von einer Vererbung der Eigenschaften der Basis-

klasse an die abgeleitete Klasse. Werden alle Objektklassen einer Modellbeschreibung nach und nach gruppiert, so entsteht eine Objektklassen-Hierarchie. Sie besteht aus einer Ursprungsklasse, von der alle weiteren Klassen abgeleitet sind. An den Enden dieser Ableitungshierarchie finden sich in der Regel die spezifischen Klassen.

Beispiel:

Basisklasse:	Switch
abgeleitete Klassen:	Datus-Switch, Netcomm-Switch

Instantiation

Eine konkretes Objekt einer Klasse wird Instanz oder Repräsentant dieser Klasse genannt. Es hat eine eindeutige Identität, mit der es sich von anderen Instanzen unterscheidet. Man unterteilt in konkrete und abstrakte Objektklassen. Konkrete Klassen können direkte Instanzen besitzen. Abstrakte Klassen hingegen werden hauptsächlich aus Modellierungsgründen eingeführt und können nicht instantiiert werden.

Aggregation

Mit der Methode der Aggregation wird die Möglichkeit gegeben, durch Zusammensetzung von einfachen Objekten komplexe Objektkonstrukte zu erzeugen. Hierzu werden Relationsaussagen zwischen Objekten getroffen, die einen Zusammenhang oder Sachverhalt beschreiben. Eine Relation besteht aus zwei Objekten, die mit einem definierten Relationsausdruck verbunden sind. Man spricht in diesem Zusammenhang auch von der linken bzw. rechten Seite einer Relationsaussage.

Beispiel:

	Modul A	*ist Komponente von*	Switch X
oder	Switch X	*ist verbunden mit*	Switch Y

In diesem Beispiel sind *'ist Komponente von'* und *'ist verbunden mit'* Relationsrealisierungen zu konkreten Objekten. Bevor jedoch derartige Aussagen gemacht werden können, müssen eindeutige Regeln aufgestellt werden, welche Relationen zwischen Repräsentanten von Objektklassen erlaubt sind. Sie sind notwendig, um die Plausibilität der aufzustellenden Aussage zu überprüfen. Diese Regeln sind Bestandteil einer aggregation hierarchy. In diesem Fall müßten also die Regeln

	Klasse Modul	*ist Komponente von*	Klasse Switch
und	Klasse Switch	*ist verbunden mit*	Klasse Switch

existieren. Die Regeln werden an abgeleitete Klassen vererbt, d.h. wird von einer Objektklasse eine neue abgeleitet, so werden gleichzeitig alle geltenden Regeln der alten Klasse auf die neue übertragen.

Beispiel:

(1) Klasse Modul *ist Komponente von* Klasse Switch

durch die Ableitung der Klasse X.25-Modul von der Klasse Modul entsteht die neue Regel

(2) Klasse X.25-Modul *ist Komponente von* Klasse Switch

Eine zusätzliche Verfeinerung der Regeln kann durch die Vorgabe eines Realisierungsintervalls vorgenommen werden. Mit ihm wird festgelegt, wieviele Relationsaussagen zu einer Instanz gemacht werden dürfen:

- [n] := genau n Relationen
- [-n] := bis zu n Relationen
- [n-] := mindestens n Relationen
- [n-m] := mindestens n und maximal m Relationen

Zur Vereinfachung im späteren Modellentwurf wird dieses Intervall nur für eine Seite des Relationsausdrucks angegeben, d.h. es gilt für die Objektklasse auf der linken Seite.

Beispiel:

Klasse X.25-Modul *enthält [1-4]* Klasse X.25-Port

Diese Regel bedeutet, daß jede Instanz der Objektklasse X.25-Modul eine bis vier *'enthält'*-Relationen zu X.25-Port-Objekten besitzen darf.

4 Entwurf eines integrierten X.25-Management-Modells

In diesem Kapitel wird ein Modell für das Management von X.25-Netzen mit einer Plattform vorgestellt. Voraussetzung für eine spätere Umsetzung des Modells ist ein objektorientierter Ansatz in der Plattformarchitektur, d.h. es müssen entsprechende objektorientierte Modellierungsmöglichkeiten zur Verfügung stehen. Desweiteren werden Plattformschnittstellen zum Daten-Import- und Export benötigt, um speziellen X.25-Management-Applikationen Zugriff auf die gesammelten Informationen über das X.25-Netz zu ermöglichen.

Im X.25-Management-Modell werden zwei Sektoren unterschieden. Der erste Sektor umfaßt den Entwurf und die Umsetzung der X.25-Netzwelt auf die Plattformebene. Hierzu werden verschiedene Objektklassen definiert, die einerseits durch Vererbung und andererseits durch Relation zueinander die Grundlage für eine Abbildung des zu integrierenden X.25-Netzes darstellen. Es werden zum einen Klassen unterschieden, die Objekte der realen Welt repräsentieren, wie z.B. der Switch mit einzelnen Ports. Zum anderen werden auch Objektklassen eingeführt, die abstrakte Begriffe verkörpern oder als Gruppierungsmittel dienen, wie z.B. der Link als Verbindungsglied zwischen zwei Ports oder das X.25-Subnet. Letztere Klassen zeichnen sich vor allem durch ihre Fähigkeit aus, andere Objekte zu kontrollieren und teilweise auch zu modifizieren. Alle diese neu eingeführten Erweiterungen der Plattform sind Bestandteil des X.25-Management-Moduls. Sie beinhalten und verwalten alle Daten, die in einem X.25-Netz anfallen.

Auf der Datengrundlage des X.25-Management-Moduls setzen zusätzliche X.25-Management-Applikationen auf. Sie sind Bestandteil des zweiten Sektors und eröffnen dem Netzbetreiber neue Möglichkeiten, die ihm durch die herkömmliche isolierte Betrachtungsweise nicht zugänglich sind. Ihr Zuständigkeitsbereich umfaßt einzelne Management-Gebiete mit speziellen Aufgaben und Anforderungen, z.B. die Verwaltung von virtuellen Verbindungen. Zur Erfüllung ihrer Aufgabe benötigen sie die erfaßten Daten des X.25-Netzes, an die sie über eine Schnittstelle der Plattform gelangen.

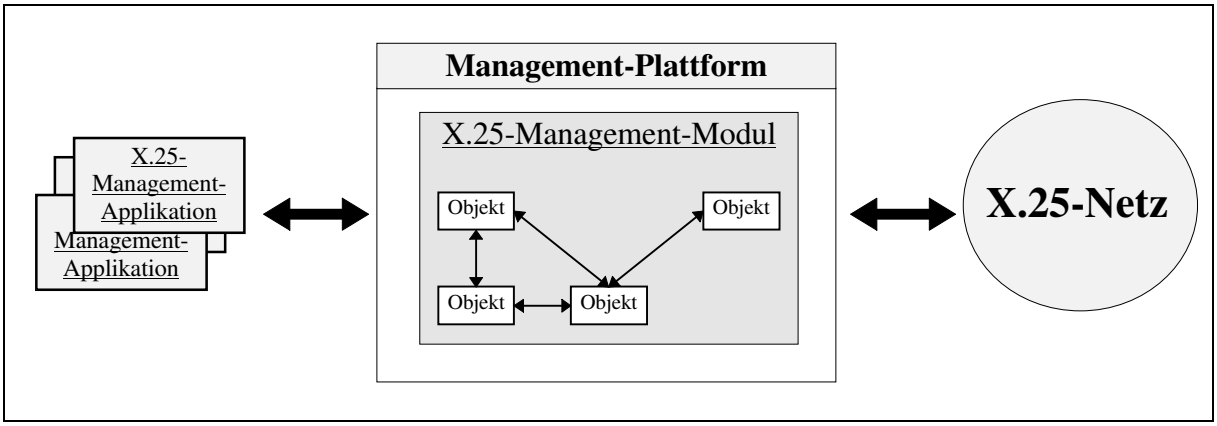


Abbildung 4-1 Sektoren des X.25-Management-Modells.

4.1 Plattformerweiterung durch ein X.25-Management-Modul

Die Beschreibung des X.25-Management-Moduls erfolgt durch Objektklassen und Relationen zwischen deren Instanzen. Die Objektklassen beinhalten im allgemeinen Datenelemente und können darüber hinaus auch über intelligente Mechanismen verfügen, die ihnen ihre charakteristische Verhaltensweise verleihen. In jeder objektorientierten Plattformumgebung sollte eine Realisierung des X.25-Management-Moduls mit Hilfe dieser Beschreibung möglich sein.

Abbildung 4-2 zeigt die graphische Beschreibungsweise von Vererbung und Relation, die im folgenden verwendet wird. Eine Vererbung wird durch einen Pfeil von der Basis- zur abgeleiteten Objektklasse ausgedrückt. Die Rahmen der Basisklassen sind nicht grau hinterlegt. Eine Relation zwischen zwei Objektinstanzen wird durch ein Hexagon gekennzeichnet, in dem die Relationsbezeichnung enthalten ist und ein Intervall mit der Angabe, wieviele derartige Relationen aufgestellt werden dürfen.

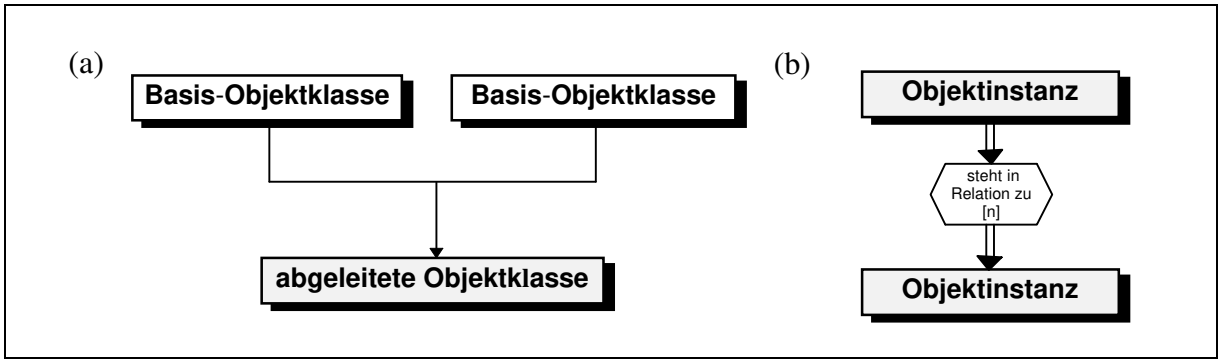


Abbildung 4-2 Objektbeschreibung: (a) Vererbung; (b) Relation.

Die benötigten Objektklassen für die Realisierung des X.25-Management-Moduls lassen sich in verschiedenen, logischen Gruppen zusammenfassen. Mit ihnen kann ein X.25-Netz vollständig mit seinen Geräten und Komponenten modelliert werden. Eine Gesamtübersicht der Objektklassen ist in Abbildung 4-3 dargestellt

Die Grundlage des Moduls bildet die Gruppe der Datenerfassungstypen. Sie beinhaltet alle Objektarten, die die verschiedenen Datenquellen innerhalb eines X.25-Netzes abdecken. Hauptsächlich werden hierzu einzelne Protokollschichten und Geräte-Funktionalitäten in Objektklassen umgesetzt, um deren Eigenschaften später durch Vererbung oder Verbindung mit Relationen an andere Klassen weiterzugeben. Sie bilden unter anderem die Basis für die Gruppe der Komponenten.

In der Komponentengruppe befinden sich Porttypen, die von X.25-Geräten verwendet werden. Abhängig von der Art des Protokolls, das auf einem Interface-Port verwendet wird, oder des Verwendungszwecks sind in dieser Gruppe vier verschiedene Portklassen enthalten. In der Regel sind ein oder mehrere Ports auf einzelnen Steckmodulen montiert, die dann in die jeweiligen Geräte eingesetzt werden. Diese Module-Klasse ist ein weiterer Bestandteil der Komponentengruppe.

Die nächste Hauptgruppe sind schließlich die X.25-Geräte. Darunter sind generell alle Einheiten mit einer X.121-Adresse zu verstehen. Die X.25-Geräteklasse dient als Grundlage für die spezifischen X.25-Einheiten, z.B. Switch oder PAD.

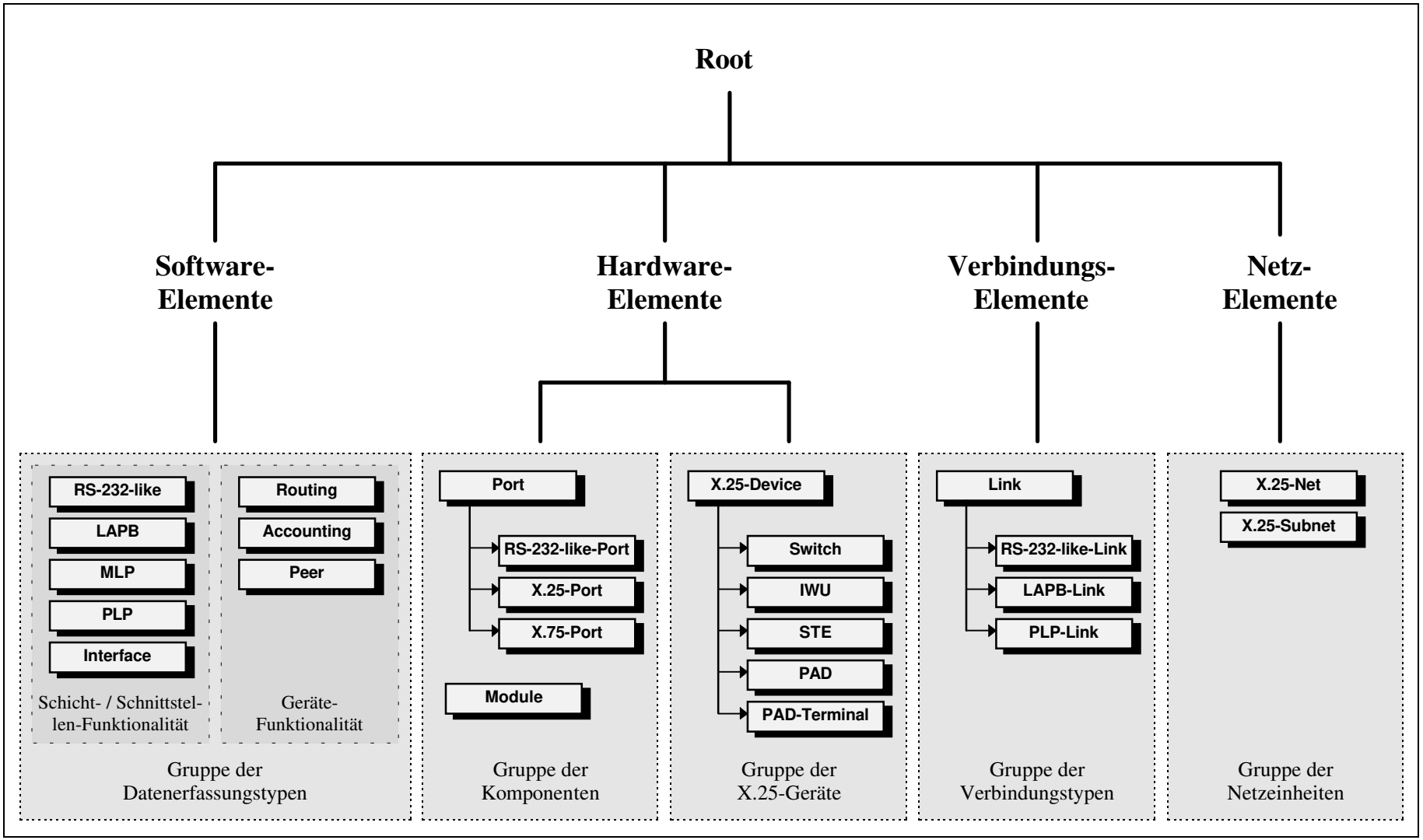


Abbildung 4-3 Gesamtdarstellung der Objektklassen.

Um den Begriff der Verbindung zwischen Geräten darzustellen, wird die Gruppe der Verbindungstypen benötigt. Für alle Verbindungen zwischen den einzelnen Porttypen mit ihren Datenerfassungsobjekten werden hier spezifische Link-Objektklassen definiert.

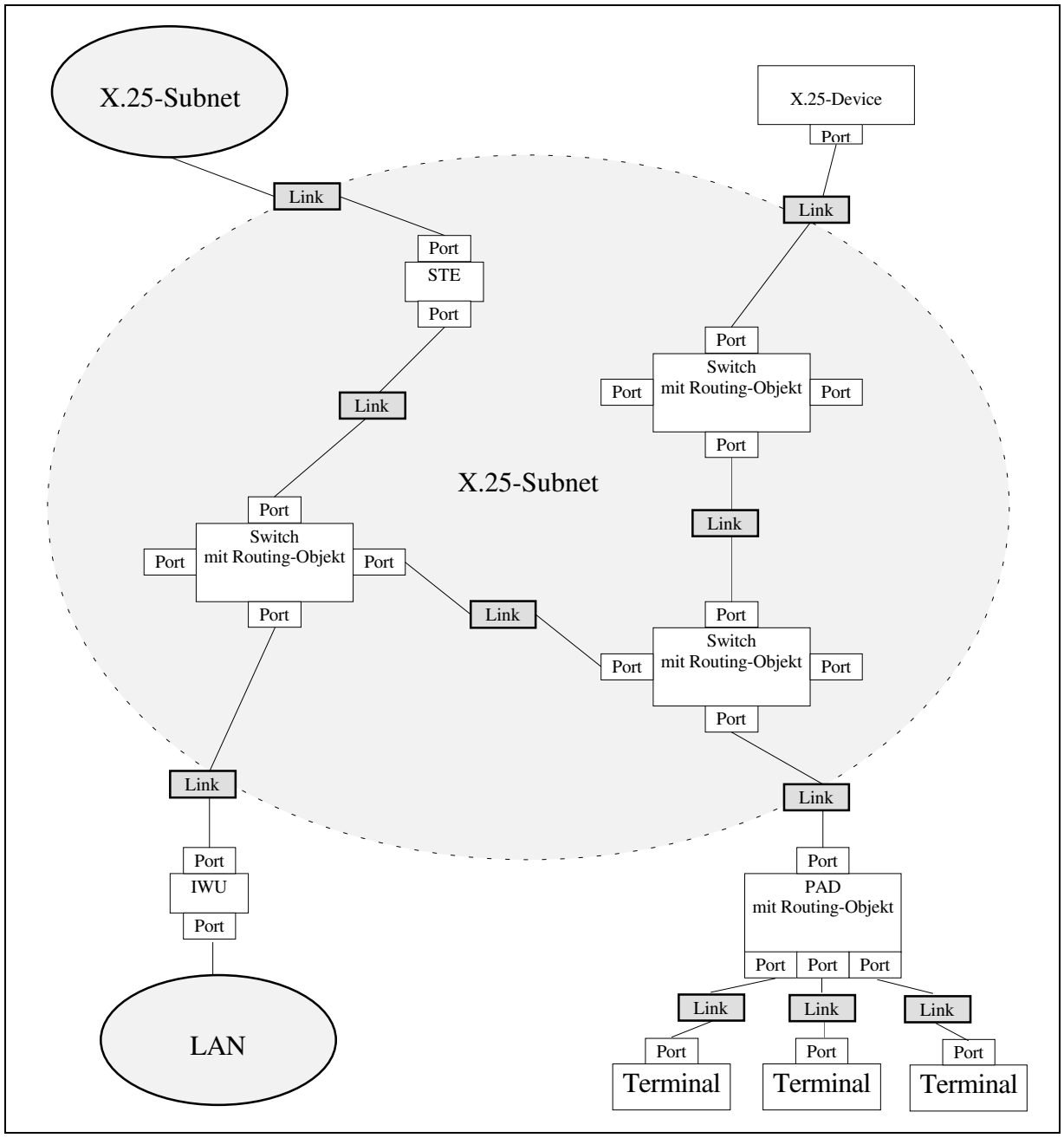


Abbildung 4-4 Netzlandschaft.

Letztendlich ist es auch notwendig, das gesamte X.25-Netz im Zusammenhang zu beschreiben. Diese Gruppe ist wichtig, um das WAN-Netz mit den verschiedenen anderen Netzen, mit denen es einen Netzverbund bildet, korrekt einzugliedern und die Bildung von Subnetzen zu ermöglichen.

In Abbildung 4-4 werden die Zusammenhänge der verschiedenen Objektklassen und Begriffe verdeutlicht. Datenerfassungsobjekte sind dort im einzelnen nicht aufgeführt, abgesehen von

den Routing-Objekten, auf die bei den Switch- und PAD-Einheiten hingewiesen wird. Mit den Port-Objekten sind jedoch Protokoll-Objekte verbunden, auch wenn diese Tatsache in der Abbildung aus Gründen der Übersichtlichkeit nicht hervortritt.

4.1.1 Relationen

Die Objektklassen sind, isoliert betrachtet, abgekapselte Elemente ohne Zusammenhang. Durch Relationen können aus einzelnen Objekten komplexe Objektkonstrukte mit unterschiedlichen Beziehungen und Verbindungen hergestellt werden. Im X.25-Management-Modul werden insgesamt fünf Relationsarten verwendet, die Verhältnisse und Abhängigkeiten ausdrücken sollen. Sie sind die verbindenden Elemente und stehen für Sinnzusammenhänge zwischen den Objekten. Abbildung 4-5 zeigt die verschiedenen Relationsarten mit Beispielen für ihre Einsatzbereiche.

Applikationen und Management-Mechanismen der Plattform greifen auf das aufgebaute Objektkonstrukt der Netzlandschaft zu. Durch die Verknüpfungen ergeben sich für sie vielfältige Zugriffsmöglichkeiten. Es kann nach bestimmten Gruppen von Relationen oder nach Objekten gesucht werden, die durch einen Relationsausdruck verknüpft sind. Unter anderem werden auch aus diesem Grund unterschiedliche Relationsarten eingeführt, um Suchvorgänge nach ganz konkreten Zusammenhängen durchführen zu können.

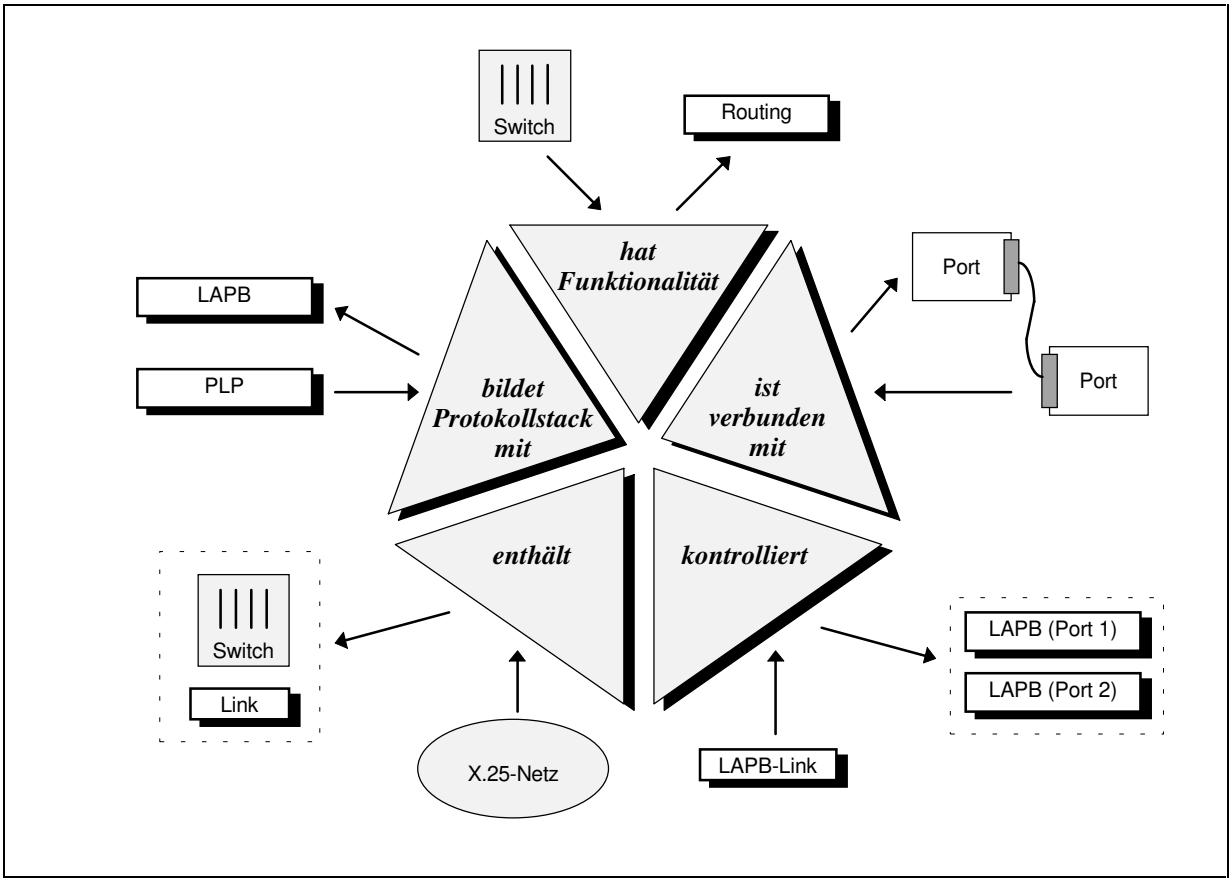


Abbildung 4-5 Relationsarten mit Beispielen.

Beschreibung der einzelnen Relationsarten:

- „bildet Protokollstack mit“:
  Die Relationsart steht für zusammenhängende Protokollschichten. Mit ihr werden Objektklassen verbunden, die einzelne Protokollschichten repräsentieren. In der Regel gibt es für jede Instanz einer Schicht nur einen Relationsausdruck, der die Verbindung zu einem Objekt der nächsten Protokollschicht herstellt. Ein Sonderfall zeigt sich im Management-Modul bei der MLP-Objektklasse. Sie darf Relationen zu mehreren Instanzen haben.

- „hat Funktionalität“:
  Physikalische Einheiten haben bestimmte Funktionalitäten. Sie geben Auskunft über die Funktionen, die ein Gerät erfüllt, oder welche Eigenschaften es besitzt. Mit einem Relationsausdruck wird die Verbindung zu einem Datenobjekt hergestellt, das für diese Funktionalität benötigt wird. Im Modul wird diese Relation für Funktionalitäten von X.25-Geräten eingesetzt und für Daten, die ein physikalischer Port verwaltet.

- „ist verbunden mit“:
  Mit dieser Relation werden existierende Verbindungen zwischen Einheiten ausgedrückt, die in der Regel durch Leitungen entstehen. Zusammenhängende Ports können damit realisiert werden.

- „kontrolliert“:
  Ein Link-Objekt kontrolliert den Verbindungsweg zwischen zwei Ports. Generell kann diese Relation überall dort eingesetzt werden, wo Objekte anderen Objekten übergeordnet sind und Einfluß auf deren Datenelemente haben.

- „enthält“:
  Die Beziehung bedeutet, daß eine Einheit eine andere Einheit entweder physikalisch oder logisch enthält. Ein X.25-Geräte-Objekt hat derartige Relationen mit Modulen. Ein Objekt für Netzeinheiten hingegen beinhaltet damit alle seine zugehörigen X.25-Geräte- und Link-Objekte.

Eine Übersicht von allen aufgestellten Relationsregeln, die im X.25-Management-Modul aufgestellt werden, ist in Anhang B zu finden.

4.1.2 Die Gruppe der Datenerfassungstypen

Innerhalb des X.25-Netzes befinden sich konfigurierbare und statistische Datenbereiche, die verwaltet und ausgewertet werden können. Sie stehen zum einen in Zusammenhang mit den verschiedenen Protokollen, die auf den jeweiligen Schnittstellen verwendet werden, zum anderen fallen sie durch Realisierung bestimmter Funktionalitäten in X.25-Geräten an. In Abbildung 4-6 sind die einzelnen Objektklassen zur Datenerfassung zu sehen, die in dieser Gruppe definiert werden. Soweit wie möglich wird bei der Definition der Objektklassen auf Standard-MIB-Beschreibungen zurückgegriffen, die in Form von RFC-Dokumenten (request for comments) des IAB vorliegen.

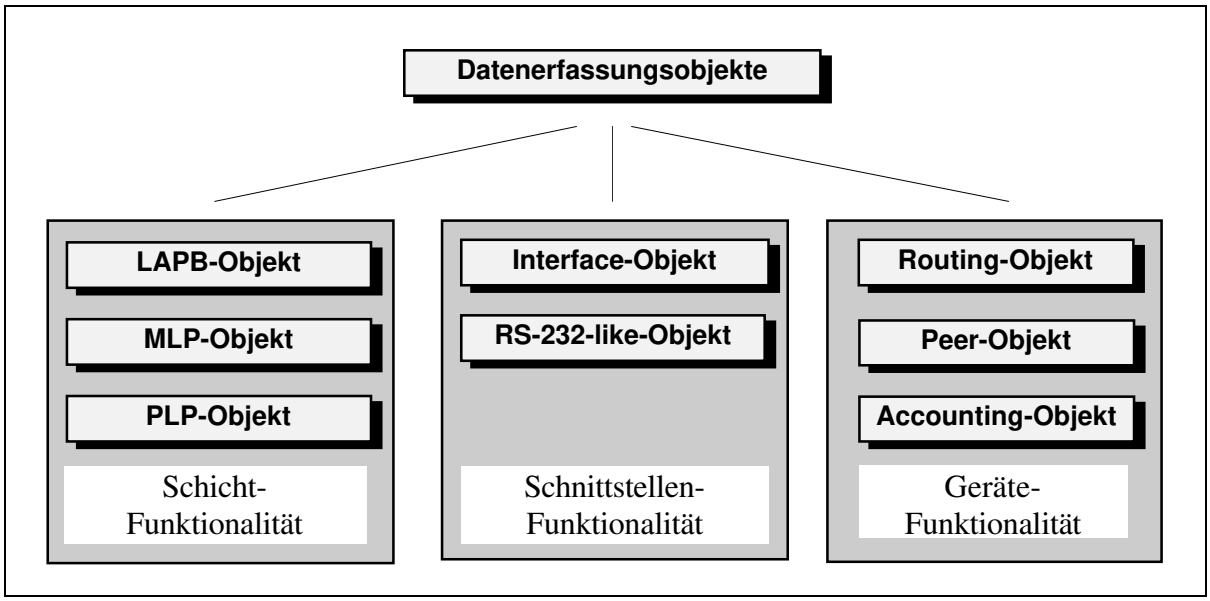


Abbildung 4-6 Gruppe der Datenerfassungsobjekte.

Die wichtigste Informationsquelle ist das X.25-Protokoll selbst. Die Schichten dieses Protokolls können konkret durch Verwendung von drei Standard-MIB-Beschreibungen in Objektklassen umgesetzt werden. Bei den Objektklassen, die auf dieser Basis beschrieben werden, handelt es sich im einzelnen um die PLP- (Packet Layer Protocol), LAPB- (Link Access Procedure Balanced) und RS-232-like-Klasse. Letztere besitzt sowohl Schicht- als auch Schnittstellen-Funktionalität. Da sie aber für sich allein bereits die Datengrundlage für eine serielle Schnittstelle enthält, wird sie in Abbildung 4-6 der Schnittstellengruppe zugeordnet.

Eine ähnliche Verfahrensweise bietet sich ebenso beim X.75-Protokoll an, das als Schnittstellenprotokoll zwischen X.25-Subnetzen dient. Da sich das X.75- und X.25-Protokoll nur geringfügig unterscheiden, genügt es, auf die entsprechenden X.25-Klassen zurückzugreifen.

Anstatt eines LAPB-Protokolls bei X.25 und X.75, besteht bei der Verwendung einer zusätzlichen Unterschicht die Möglichkeit, je nach Art der Verkabelung mehrere Schicht-2-Verbindungen einzusetzen. Diese Protokollart wird MLP (Multilink Procedure) genannt und verwaltet untergeordnete LAPB-Verbindungen. Eine entsprechende Objektklasse findet sich auch in der Datenerfassungsgruppe.

Eine allgemeine Datenobjektklasse für Schnittstellen ohne Berücksichtigung der darauf genutzten Protokolle steht mit der Interface-Klasse zur Verfügung. Unter anderem wird es für den Anschluß eines LAN (Local Area Network) verwendet werden. Es gibt zwar verschiedene LAN-Grundtypen, wie z.B. Ethernet, Token Ring usw., dennoch reicht eine Klasse für die Datenerfassung aus, da nur grundlegende Schnittstellendaten von Bedeutung sind. Auch hier wird wieder auf einen Teil einer Standard-MIB-Beschreibung zurückgegriffen.

Für die Routing-Funktionalität des Switches wird eine Objektklasse definiert, mit der die Wegewahlentscheidung des jeweiligen Switches innerhalb der Modellwelt nachvollzogen werden kann. Weitere Klassen mit speziellen Funktionalitäten werden für den Bereich des Accounting und für die Adressumsetzung in einem IWU benötigt.

4.1.2.1 RS-232-like-Objektklasse

Über die meisten Schnittstellen und Ports im X.25-Netz werden Daten seriell übertragen. Die RS-232-like-Objektklasse bildet die Grundlage für eine Datenerfassung, die mit dieser synchronen oder asynchronen Übertragungsart zusammenhängt. Zugleich repräsentiert sie auch die erste Schicht des X.25- und X.75-Protokolls.

Die Daten der Objektklasse gliedern sich in folgende Bereiche:

- Konfigurationsdaten für die serielle Übertragung
  (z.B. Geschwindigkeit, Datenbits, Stopbits).
- Informationen über die Zustände von einzelnen Signalleitungen.
- Statistikdaten über aufgetretene Fehlerzustände.

Die Datenelemente der Objektklasse werden aus der MIB-Beschreibung „Definitions of Managed Objects for RS-232-like Hardware Devices“ [RFC1317] entnommen. In dieser Beschreibung wird ein RS-232-like MIB-Modul definiert, das eine Reihe von seriellen physikalischen Verbindungen mit einer ähnlichen Menge von Kontrollsignalen unterstützt. Die wichtigsten darin enthaltenen Verbindungen sind RS-232, RS-422, RS-423 und V.35. Das RS-232-like MIB-Modul ist vorgesehen als Bestandteil einer MIB eines Systems, das über derartige Schnittstellen verfügt. Es besteht aus fünf Tabellen und einem Eintrag, der die Anzahl der in diesem MIB-Teilbaum verwalteten Schnittstellen enthält. Die Zuordnung der einzelnen Tabelleneinträge auf die betreffende Schnittstelle erfolgt über Indizes.

Die Hauptbestandteile des RS-232-like MIB-Moduls sind im einzelnen:

- rs232Number:
  Objekt, das die Anzahl der Ports in den Tabellen enthält.
- rs232PortTable:
  Die Einträge in dieser Tabelle beschreiben den Status und die Parameter der einzelnen Ports.
- rs232AsyncPortTable:
  Eine Liste von Einträgen der asynchronen Ports mit deren Zuständen und Parametern.
- rs232SyncPortTable:
  Eine Liste von Einträgen der synchronen Ports mit deren Zuständen und Parametern.
- rs232InSigTable:
  Die Tabelle erfaßt die Eingangs-Kontroll-Signale eines Ports.
- rs232OutSigTable:
  Die Tabelle erfaßt die Ausgangs-Kontroll-Signale eines Ports.

Folgende Tabelle zeigt einen Ausschnitt der wichtigsten Datenelemente der RS-232-like-Objektklasse:

Datenelement	Bemerkung [1]
rs232PortType	type of port (e.g. rs232, rs422)
rs232PortInSpeed	port's input speed in bits per second
rs232PortOutSpeed	port's output speed in bits per second
rs232AsyncPortBits	number of bits in a character
rs232AsyncPortStopBits	number of stop bits
rs232AsyncPortParity	parity bit
rs232AsyncPortParityErrs	number of characters with a parity error
rs232AsyncPortFramingErrs	number of characters with a framing error
rs232SyncPortFrameCheckErrs	number of frames with an invalid frame check sequence
rs232SyncPortTransmitUnderrunErrs	number of frames that failed to be transmitted (reason: time)
rs232SyncPortReceiveUnderrunErrs	number of frames that failed to be received (reason: time)
rs232SyncPortInterruptedFrames	number of frames that failed to be received (reason: loss of modem signals)
rs232SyncPortAbortedFrames	number of frames aborted (reason: abort sequence)

Tabelle 4-1 RS-232-like-Datenelemente.

Eine Instanz einer RS-232-like-Objektklasse verwaltet genau einen Port und verfügt somit auch nur über jeweils einen Eintrag der verschiedenen Tabellen des RS-232-like MIB-Moduls.

4.1.2.2 LAPB-Objektklasse

Das LAPB-Objekt repräsentiert die anfallenden Daten des Schicht-2-Protokolls von X.25. Es ist zuständig für die Datenerfassung und Kontrolle eines Links zwischen zwei X.25-Schnittstellen. Die Daten des Objekts lassen sich anhand ihres Aufgabenbereichs in folgender Weise gruppieren:

- Konfigurationsdaten, wie z.B. frame- und windowsize, Timer-Einstellungen für die Steuerung der Schicht-2-Abläufe.
- Statistikdaten, wie z.B. rejected frames, timeouts, frames sent, frames received.

Alle diese Datenaspekte werden innerhalb der MIB-Beschreibung „SNMP MIB Extension for X.25 LAPB“ [RFC1381] ausführlich behandelt. Mit Hilfe des dort beschriebenen LAPB-MIB-Moduls kann somit eine entsprechende Objektklasse konstruiert werden.

[1] für eine detailiertere Beschreibung siehe [RFC1317]

Die Hauptbestandteile des LAPB-MIB-Moduls sind vier Tabellen. Ähnlich wie in der RS-232-like MIB-Beschreibung beinhaltet jedes X.25-Gerät, das über mindestens einen X.25-Port verfügt, in seiner System-MIB ein LAPB-MIB-Modul. Jeder Eintrag in dessen Tabellen bezieht sich dann über Indizes auf genau eine Schnittstelle. Eine Instanz einer LAPB-Objektklasse des Management-Moduls bezieht sich auf einen Link und verwaltet deshalb jeweils einen Eintrag aus den verschiedenen Tabellen.

Das LAPB-MIB-Modul enthält folgende Elemente:

- lapbAdmnTable:
  Diese Tabelle enthält alle Konfigurationsmöglichkeiten für das LAPB-Protokoll, die verändert werden können. Eine Veränderung dieser Einstellungen kann unter Umständen keine sofortige Beeinflussung des Protokollablaufs zur Folge haben, da manche Werte erst bei einem Reset der Schnittstelle übernommen werden. Die momentan wirklich aktiven Werte sind in der lapbOperTable ersichtlich.

- lapbOperTable:
  Die Tabelle entspricht fast völlig der lapbAdmnTable. Sie stellt die augenblicklich aktuelle Laufzeit-Konfiguration dar. Da die Tabelle zur Kontrolle der gerade aktiven Parameter dient, sind ihre enthaltenen Objekte schreibgeschützt.

- lapbFlowTable:
  Hier werden Statistik- und Statusdaten gespeichert.

- lapbXidTable:
  Definition von Werten für XID-Bearbeitung (exchange identification) zum Austausch von Parametern des Link-Aufbaus.

- lapbProtocolVersion:
  MIB-Teilbaum für Protokollversion.

Einen Auszug der wichtigsten Datenelemente der Tabelleneinträge zeigt folgende Übersicht:

Datenelement	Bemerkung[1]
lapbAdmnStationType	station type of interface (dte or dce)
lapbAdmnTransmitN1FrameSize	max. framesize to transmit
lapbAdmnReceiveN1FrameSize	max. framesize to receive
lapbAdmnTransmitKWindowSize	sliding window mechanism
lapbAdmnReceiveKWindowSize	sliding window mechanism
lapbFlowCurrentMode	current condition of conversation
lapbFlowBusyDefers	counter for busy conditions of remote device
lapbFlowRejOutPkts	rejected frames sent
lapbFlowRejInPkts	rejected frames received
lapbFlowFrmrSent	infofield of the last rejected frame sent
lapbFlowFrmrReceived	infofield of the last rejected frame received

Tabelle 4-2 Auszug der LAPB-Datenelemente.

[1] für eine detailierte Beschreibung siehe [RFC1381]

Das LAPB-Objekt steht gewöhnlich mit einem PLP-Objekt in Relation. Eine Sonderstellung nimmt hier jedoch das MLP-Protokoll ein, da es mehrere LAPB-Objekte gleichzeitig verwaltet (siehe MLP-Objektklassen-Beschreibung). Ein LAPB-Objekt bildet mit einem RS-232-like-Objekt einen Teil eines Protokollstacks.

Relationsregeln der Objektklasse:

LAPB *bildet Protokollstack mit [1]* RS-232-like

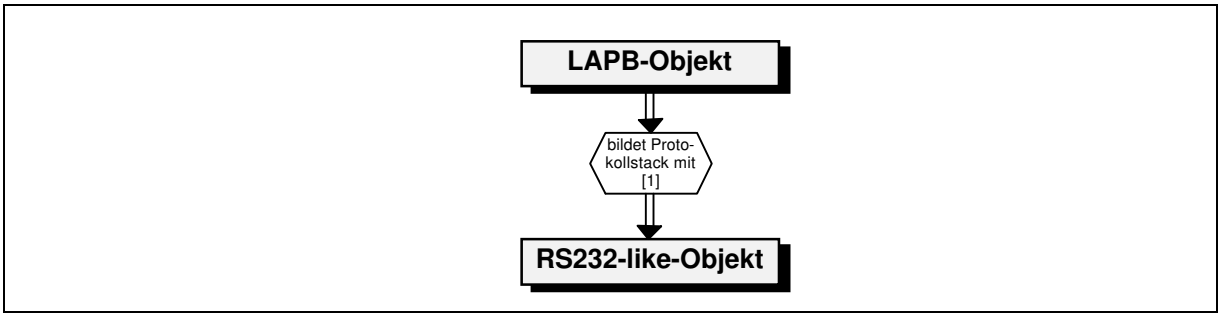


Abbildung 4-7 LAPB-Objekt.

4.1.2.3 MLP-Objektklasse

Die MLP-Objektklasse verwaltet mehrere LAPB-Objekte. Sie wird für die Einordnung der Schicht-2-Erweiterung MLP (Multi Link Procedure) verwendet, mit dem Links über mehrere physikalische Verbindungen gleichzeitig betrieben werden können. Das Schicht-2-Protokoll setzt sich in diesem Fall also aus MLP und mehreren LAPBs zusammen, die mit entsprechend vielen Schicht-1-Einheiten verbunden sind. Bei Ausfall eines Links kann immer noch auf die anderen Verbindungen zurückgegriffen werden, so daß erst bei einer Störung von allen verwalteten Verbindungen die Schicht-2 der jeweiligen Schnittstelle nicht mehr funktionsfähig ist.

Für die Objektwelt bedeutet das, daß ein MLP-Objekt in Relation zu mindestens zwei LAPB-Objekten steht (siehe Abbildung 4-8). Der Status eines MLP-Objekts leitet sich aus den Zuständen der abhängigen Objekte ab. Bei der Erzeugung einer Instanz der MLP-Klasse müssen gleichzeitig die benötigten LAPB-Instanzen geschaffen werden.

Relationsregeln der Objektklasse:

MLP *bildet Protokollstack mit [2-]* LAPB

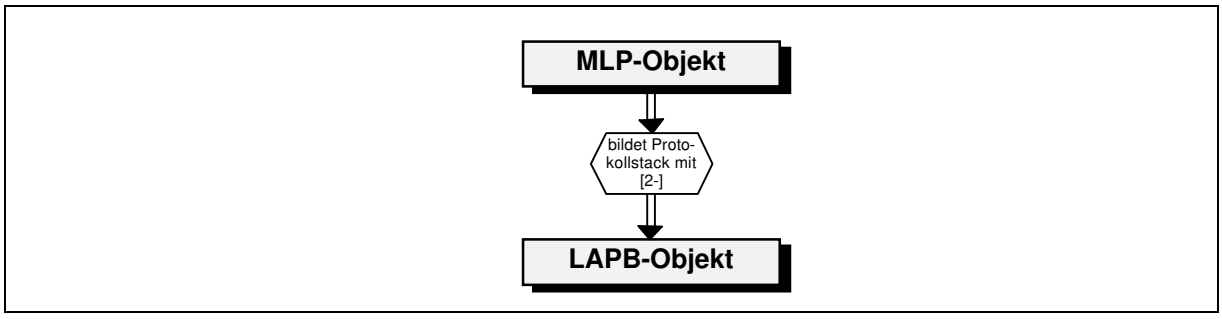


Abbildung 4-8 MLP-Objekt.

4.1.2.4 PLP-Objektklasse

Das packet layer protocol bildet die dritte Schicht von X.25. Ebenso wie für LAPB wird auch hier eine separate Objektklasse für die Erfassung der Daten dieser Schicht definiert. Das Aufgabengebiet umfaßt hauptsächlich die Verwaltung der virtuellen Verbindungen. Es ergeben sich somit wieder einzelne Datengruppen innerhalb des Objekts:

- Konfigurationsdaten, wie z.B. windowsize, Timer-Einstellungen, circuit-Verwaltung.
- Statistikdaten, wie z.B. timeouts, übertragene Paketarten.
- circuit-table.
- facilities der aktuellen calls.

Auch für diese Objektklasse kann auf eine vorhandene MIB-Beschreibung „SNMP MIB Extension for the X.25 Packet Layer“ [RFC1382] zurückgegriffen werden.

In der MIB-Beschreibung zeigt sich ein ähnlicher Aufbau wie zuvor in der LAPB-RFC. Das dort definierte PLP-MIB-Modul besteht hauptsächlich aus sieben Tabellen, deren Einträge sich jeweils über Index-Referenzen auf die jeweiligen Schnittstellen eines Geräts beziehen. Dennoch tritt hier ein Sonderfall auf, da eine Instanz einer PLP-Objektklasse eine im vornherein nicht bekannte Anzahl von virtual calls mit deren Parametern verwaltet. Im Gegensatz zu Instanzen von LAPB- und RS-232-like-Klassen, die nur jeweils Einträge von Tabellen verwalten, müssen für ein PLP-Objekt Tabellen angelegt werden. In der folgenden Aufzählung wird deshalb unterschieden in Tabellen, von denen eine PLP-Instanz jeweils einen Eintrag erhält, und Tabellen, die als eigenständige Datenelemente vollständig in das PLP-Objekt übernommen werden müssen.

Die Hauptbestandteile des PLP-MIB-Moduls sind im einzelnen:

- x25AdmnTable:
  Konfigurationsmöglichkeiten des PLP-Protokolls.
- x25OperTable:
  Ähnlicher Aufbau wie die x25AdmnTable mit der aktuellen Laufzeit-Konfiguration.
- x25StatTable:
  Statistikdaten der gesamten PLP-Schicht.
- x25ChannelTable:
  Einstellungen, die die Nummernvergabe der logischen Kanäle betreffen.

Vollständige Übernahme:

- x25CircuitTable:
  Tabelle mit Informationen zu verschiedenen logischen Verbindungen, die zur Zeit auf dieser Schnittstelle aktiv sind.
- x25ClearedCircuitTable:
  Daten von älteren logischen Verbindungen.

- x25CallParmTable:
  Jeder Eintrag dieser Tabelle beschreibt eine logische Verbindung anhand seiner facilities. Er ist durch einem Index mit einem Eintrag der x25CircuitTable verknüpft.

Ein Auszug der wichtigsten Datenelemente:

Datenelement	Bemerkung[1]
x25AdmnInterfaceMode	dte or dce
x25AdmnMaxActiveCircuits	maximum number of supported circuits
x25AdmnCallTimer	call timer in milliseconds
x25AdmnResetTimer	reset timer in milliseconds
x25AdmnClearTimer	clear timer in milliseconds
x25StatInCalls	counter for incoming calls
x25StatInDataPackets	data packets received
x25StatOutDataPackets	data packets sent
x25StatCallTimeouts	number of times the call timer expired
x25StatResetTimeouts	number of times the reset timer expired
x25StatClearTimeouts	number of times the clear timer expired
x25ChannelLIC	lowest incoming channel
x25ChannelHIC	highest incoming channel
x25ChannelLOC	lowest outgoing channel
x25ChannelHOC	highest outgoing channel
x25CircuitStatus	current status of the circuit
x25CircuitEstablishTime	time when channel was associated with this circuit
x25CircuitInOctets	user data delivered to upper layer
x25CircuitOutOctets	user data sent for this circuit
x25CircuitCalledDteAddress	called address
x25CircuitCallingDteAddress	calling address
x25ClearedCircuitTimeEstablished	time when circuit was established
x25ClearedCircuitTimeCleared	time when circuit was cleared
x25ClearedCircuitClearingCause	clearing cause
x25ClearedCircuitInPdus	number of PDUs received on the circuit
x25ClearedCircuitOutPdus	number of PDUs transmitted on the circuit
x25ClearedCircuitCalledAddress	called address
x25ClearedCircuiteCallingAddress	calling address
x25CallParmInPacketSize	maximum receive packet size
x25CallParmOutPacketSize	maximum transmit packet size
x25CallParmInWindowSize	receive window size
x25CallParmOutWindowSize	transmit window size
x25CallParmAcceptReverseCharging	accept or refuse charges
x25CallParmFastSelect	use of fast select facility

Tabelle 4-3 Auszug der PLP-Datenelemente.

[1] für eine detailiertere Beschreibung siehe [RFC1382]

Obwohl diese Objektklasse eigentlich für X.25 definiert wurde, kann sie auch auf die X.75-Schnittstelle ausgedehnt werden. Die Schicht-3 von X.75 besteht aus einer vereinfachten Form des X.25-PLP, so daß eine Verwendung der beschriebenen PLP-Objektklasse auch für die X.75-Schnittstelle möglich ist.

Da das packet layer protocol entweder auf das einfache LAPB-Protokoll oder das Mehrfach-Protokoll MLP aufsetzt, müssen beide Varianten auch beim Relationsaufbau berücksichtigt werden, wie bei den gegebenen Relationsmöglichkeiten in Abbildung 4-9 zu sehen ist.

Relationsregeln der Objektklasse:

PLP *bildet Protokollstack mit [1]* LAPB
PLP *bildet Protokollstack mit [1]* MLP

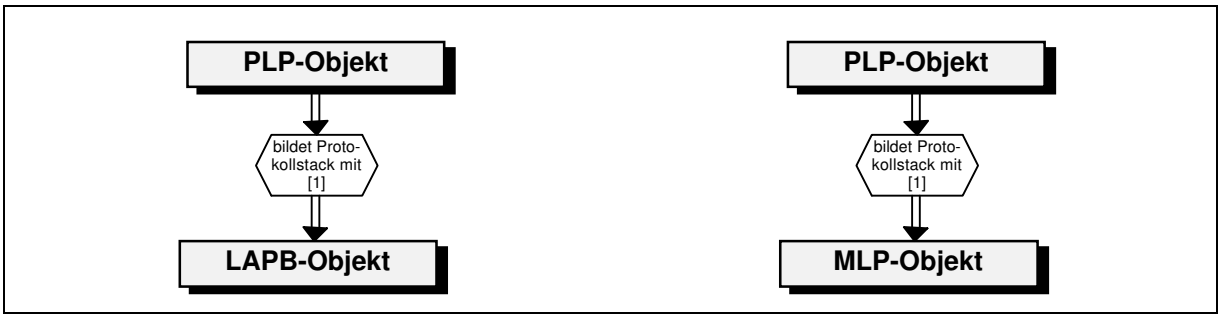


Abbildung 4-9 PLP-Objekt.

4.1.2.5 Interface-Objektklasse

Das Interface-Objekt bietet die Grundlage für die Daten, die bei einer beliebigen Schnittstelle anfallen. Es enthält einfachste Schnittstellendaten und berücksichtigt keine Daten für spezielle Protokolle, die auf dieser Schnittstelle verwendet werden.

Um eine standardisierte Datenerfassung beizubehalten, wird die MIB-Beschreibung „Management Information Base for Network Management of TCP/IP-based internets: MIB-II“ [RFC1213] verwendet. In ihr wird in der Interface-Gruppe eine allgemeine Schnittstelle mit ihren wichtigsten Datenelementen beschrieben.

Ein Auszug aus den Datenelementen der Interface-Objektklasse:

Datenelement	Bemerkung[1]
ifType	type of interface
ifSpeed	bandwidth in bits per second
ifAdminStatus	desired state of interface
ifOperStatus	current operational state of interface
ifInOctets	octets received from the interface
ifOutOctets	octets transmitted out of the interface

[1] für eine detailiertere Beschreibung siehe [RFC1213]

ifInErrors	number of inbound packets with errrors
ifOutErrors	number of outbound packets with errrors

Tabelle 4-4 Auszug der Interface-Datenelemente.

4.1.2.6 Routing-Objektklasse

Anhand des Routing-Objekts kann in der Modellwelt nachvollzogen werden, an welchen Port ein Datenpaket geleitet wird. Es soll vor allem die Routing-Tabelle eines Switches repräsentieren, aber auch für andere Zwecke eingesetzt werden können, bei denen anhand eines Kriteriums eine Wegewahlentscheidung getroffen werden muß. Es wird deshalb so einfach wie möglich gehalten, um eine flexible Verwendung zu ermöglichen.

Hauptdatenelement dieser Objektklasse ist eine Tabelle mit zwei Spalten, die zum einen das Routing-Kriterium enthalten und zum anderen den Port, der aufgrund dieses Kriteriums gewählt werden soll.

Folgende Tabelle stellt eine mögliche, vereinfachte Routing-Tabelle eines Switches dar.

Routing-Kriterium (Switch-Adresse, Facilities, QoS)	Port
458900101031x	2
4589001010302, NoReverseCharging	3
4589001010302	2
xxxxxxxxxxxxxx	1

Tabelle 4-5 Beispiel einer Routing-Tabelle.

4.1.2.7 Peer-Objektklasse

In einem IWU findet eine Adressumsetzung statt. Um Ziele, die in einem Remote-LAN liegen erreichen zu können, wird zuerst die X.121-Adresse des dortigen IWU benötigt. Die Peer-Objektklasse beinhaltet eine derartige Umsetzungstabelle.

Die Datenelemente stammen aus der MIB-Beschreibung „SNMP MIB extension for Multiprotocol Interconnect over X.25“ [RFC1461]. Es wird jedoch nur die benötigte Teiltabelle miox-PeerTable mit den Adressbeschreibungen entnommen.

Sie enthält folgende Datenelemente:

Datenelement	Bemerkung[1]
mioxPeerIndex	index value
mioxPeerStatus	status of this entry
mioxPeerMaxCircuits	maximum number of X.25 circuits allowed
mioxPeerIfIndex	index for the X.25 interface to use
mioxPeerConnectSeconds	time, a call to this peer was active

[1] für eine detailiertere Beschreibung siehe [RFC1461]

mioxPeerX25CallParamId	index to call parameters
mioxPeerEnAddr	encapsulation address of the remote host mapped by this table entry
mioxPeerX121Address	X.25 address of the remote host mapped by this table entry
mioxPeerX25CircuitId	index to circuit table
mioxPeerDescr	identification information about the peer

Tabelle 4-6 Peer-Datenelemente.

4.1.2.8 Accounting-Objektklasse

Zur Erfassung von Accounting-Daten wird die Accounting-Objektklasse eingeführt. Da es keinerlei Standards in Form einer MIB-Beschreibung gibt und dieser Management-Bereich von Switches auch nur teilweise unterstützt wird, sind die Datenelemente der Klasse sehr einfach gehalten. Sie wurden aus der Account Message eines Netcomm-Switches [NETCOMM_IG1] entnommen.

Instanzen dieser Klasse bestehen aus einer Tabelle, deren Einträge sich auf jeweils eine virtuelle Verbindung beziehen. Die Accounting-Objektklasse besitzt zusätzlich zwei Funktionen. Zum einen müssen die Accounting-Daten eines Switches erfaßt werden. Hierzu bieten sich die Möglichkeiten an, periodisch in bestimmten Zeitintervallen die Switches abzufragen oder vom Switch gesendete Nachrichten über Verbindungsaufbau und -ende auszuwerten. Zum anderen dürfen die gesammelten Daten nicht überhand nehmen. Hierfür soll eine automatische Speicherung der Daten in regelmäßigen Zeit- oder Mengenintervallen sorgen.

Datenelemente eines Tabelleneintrags der Accounting-Objektklasse:

Datenelement	Bemerkung
CalledAddress	called address
CallingAddress	calling address
CallStartTime	time when call was started
CallEndTime	time when call was cleared
PacketsIn	packets received
PacketsOut	packets sent
BytesIn	bytes received
BytesOut	bytes sent
ClearingCause	clearing cause
ClearingDiagnostic	contents of diagnostic field

Tabelle 4-7 Accounting-Datenelemente.

4.1.3 Die Gruppe der Komponenten

Die Komponentengruppe besteht großteils aus diversen physikalischen Ports, die in Verbindung mit den Eigenschaften der einzelnen Protokolle stehen. Diese Protokolleigenschaften wurden bereits in der Gruppe der Datenerfassungstypen konkretisiert. Die verschiedenen Port-

arten ergeben sich durch Ableitung von einer Basis-Portklasse und Relation zu diesen bereitgestellten Objektklassen.

Der serielle RS-232-like-Port dient als Grundlage für Schnittstellen mit serieller Übertragung. Er wird von PAD-Einheiten und Terminals benötigt.

X.25-Port und X.75-Port sind die Kernstücke dieser Gruppe. Sie stehen in Relation zu einem vollständigen X.25-Protokollstack, der über ein PLP-Objekt verbunden wird. Diese Ports stellen die Hauptschnittstellen für einen Switch dar.

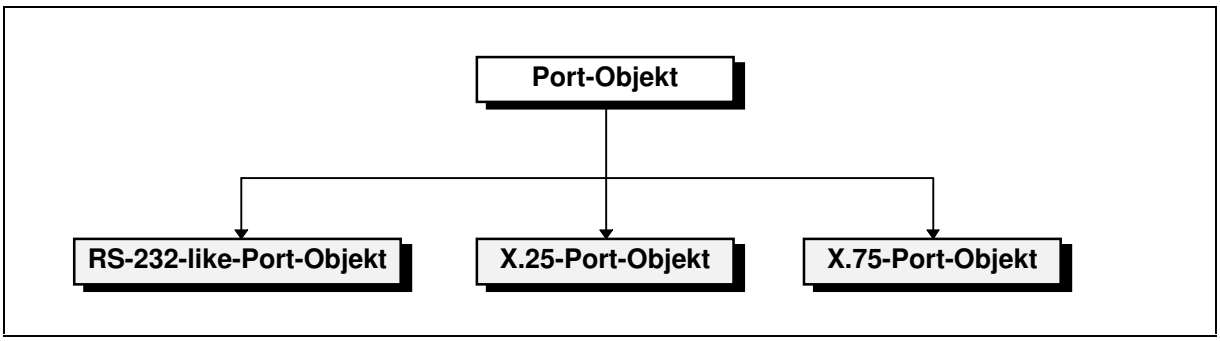


Abbildung 4-10 Gruppe der Port-Objekte.

Außer den Porttypen befindet sich noch eine weitere Objektklasse in dieser Gruppe. Die einzelnen Ports werden nur selten direkt in Geräte eingebaut, sondern aus Gründen der Erweiterbarkeit auf Steckplatten modular in diese eingesetzt. Um diese Modularität darstellen zu können, wird eine Module-Objektklasse eingeführt, die diese Eigenschaft verkörpern soll. Durch Verbindung mit einem Module-Objekt können mehrere Ports zusammengefaßt werden.

4.1.3.1 Port-Objektklasse

Die Port-Objektklasse bildet die Grundlage für alle Porttypen dieser Objektgruppe. Sie wird von der Interface-Objektklasse abgeleitet, die die Datenelemente eines Standard-Interfaces aus der MIB-II beinhaltet. Weitere speziellere Portarten können von dieser Klasse abgeleitet und mit den jeweils benötigten Protokoll-Objekten verbunden werden.

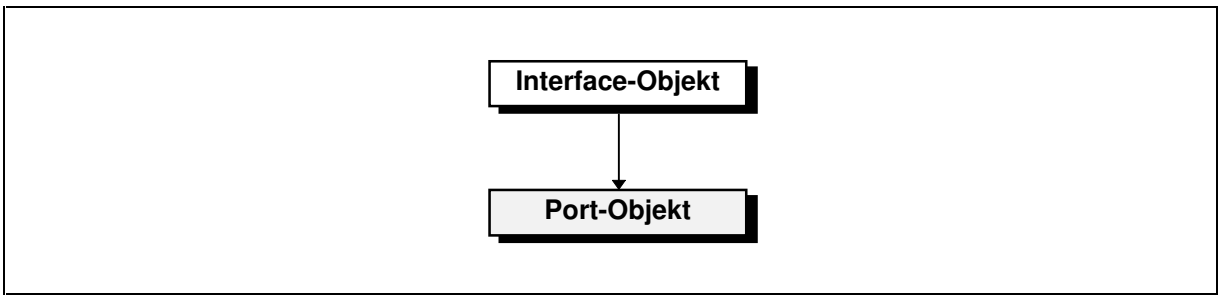


Abbildung 4-11 Port-Objekt.

4.1.3.2 RS-232-like-Port-Objektklasse

Mit der RS-232-like-Port-Objektklasse wird ein serieller Port definiert. Er wird von der Port-Objektklasse abgeleitet und übernimmt durch eine Relation die Funktionalität eines RS-232-like-Objekts. Er wird für die Verbindung von Terminals und PAD-Einheiten eingesetzt. Zum Ausdruck dieser Portverbindung existiert ebenfalls eine entsprechende Relation.

Relationsregeln der Objektklasse:

RS-232-like-Port	*hat Funktionalität [1]*	RS-232-like
RS-232-like-Port	*ist verbunden mit [1]*	RS-232-like-Port

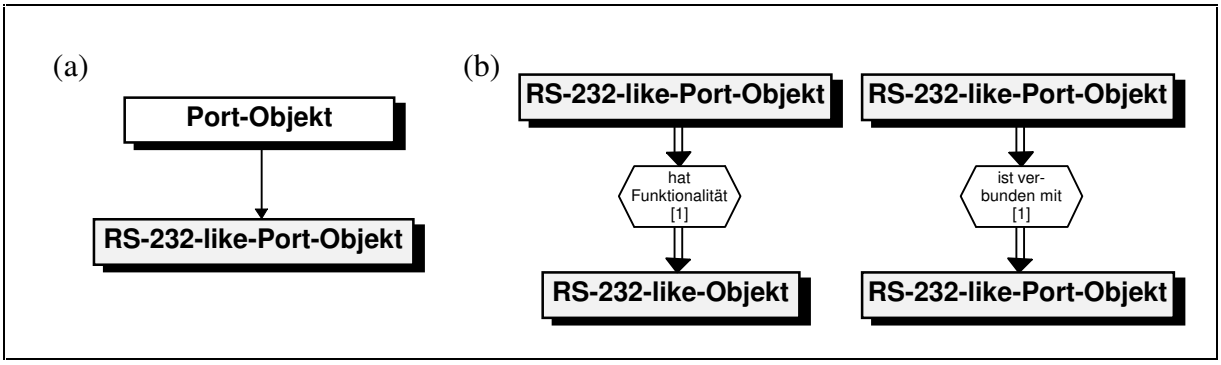


Abbildung 4-12 RS-232-like-Port-Objekt: (a) Vererbung; (b) Relation.

4.1.3.3 X.25-Port-Objektklasse / X.75-Port-Objektklasse

X.25- und X.75-Port-Objekte repräsentieren physikalisch vorhandene Interfaces. Ihre spezifischen Eigenschaften, die sie von anderen Portarten unterscheidet, wird ihnen durch die Relation zu einem PLP-Objekts verliehen. Über das PLP-Objekt sind weitere Protokoll-Objekte angebunden, die zusammen einen vollständigen X.25-Protokollstack bilden. Obwohl X.25-und X.75-Port nach dem gleichem Schema aufgebaut sind, werden zwei separate Objektklassen definiert. Der Grund für dieses Vorgehen liegt in der Sichtweise eines Netzbetreibers, der beim Management eventuell aus Einordnungs- oder Gewichtungsgründen eine Differenzierung der Ports vorzieht. Beide Portarten unterhalten ebenfalls eine Relation zu dem Port, mit dem sie verbunden sind.

Relationsregeln der Objektklassen:

X.25-Port	*hat Funktionalität [1]*	PLP
X.25-Port	*ist verbunden mit [1]*	X.25-Port
X.75-Port	*hat Funktionalität [1]*	PLP
X.75-Port	*ist verbunden mit [1]*	X.75-Port

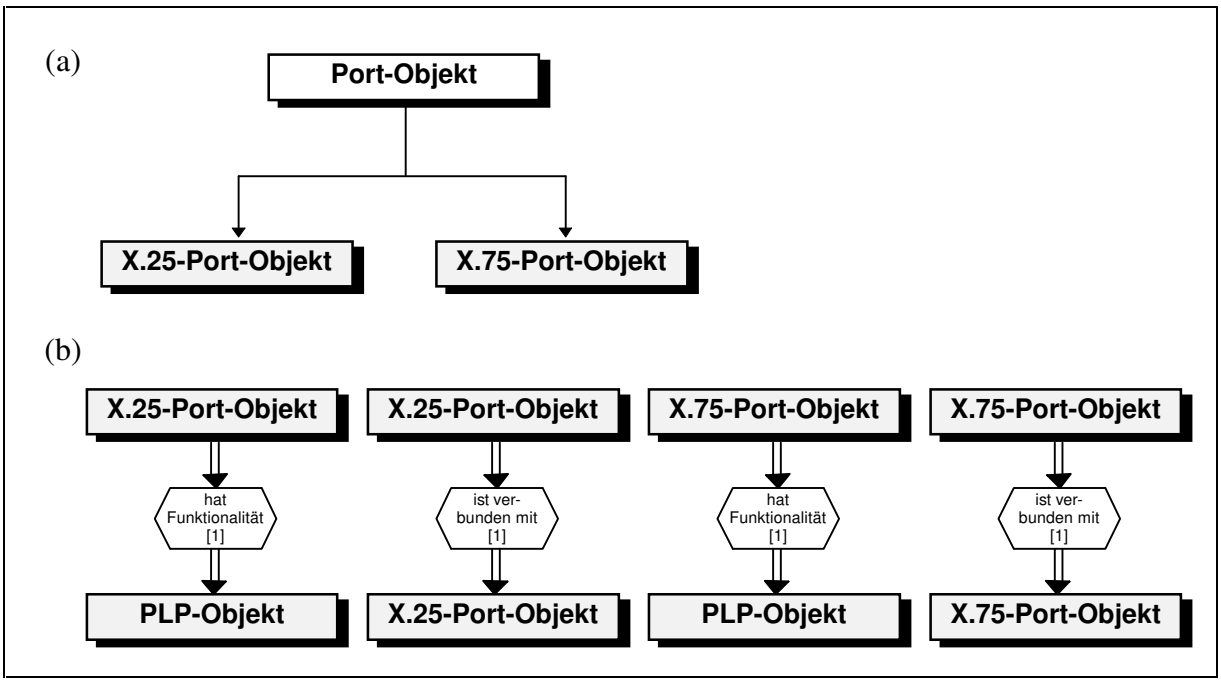


Abbildung 4-13 X.25- und X.75-Port-Objekt: (a) Vererbung; (b) Relation.

4.1.3.4 Module-Objektklasse

Eine Objektklasse, die hauptsächlich der Übersichtlichkeit und der leichteren Übertragung der realen Welt auf die Modellwelt dient, ist das Modul. Ein Modul enthält mehrere Ports. Im allgemeinen ist es Bestandteil eines Geräts, das modular erweitert werden und verschiedene Arten von Modulen beinhalten kann. Der Status eines Module-Objekts ergibt sich aus den Status-Feldern der verbundenen Port-Objekte.

Durch die Verbindung von Modulen und bestimmten Ports können Modularten generiert werden, die denjenigen in der realen Welt entsprechen. Beispielsweise läßt sich mit einem Module-Objekt und PAD-Port-Objekten ein Basistyp für eine PAD-Module-Karte eines Switches erzeugen.

Relationsregeln der Objektklasse:

Module *enthält [1-]* Port

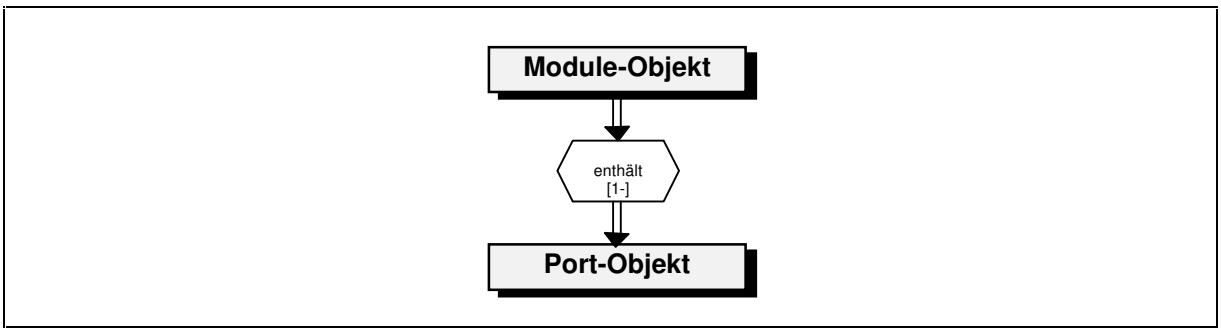


Abbildung 4-14 Module-Objekt.

4.1.4 Die Gruppe der X.25-Geräte

Mit einem X.25-Netz werden im allgemeinen ganz bestimmte Geräte in Verbindung gebracht. Sie stellen zumeist physikalisch abgegrenzte Einheiten dar. In der Regel bestehen diese aus einzelnen Modulen, über die ihnen Schnittstellen zur Verfügung stehen. Die Elemente dieser Gruppe definieren Objektklassen, die diese Geräte beschreiben sollen.

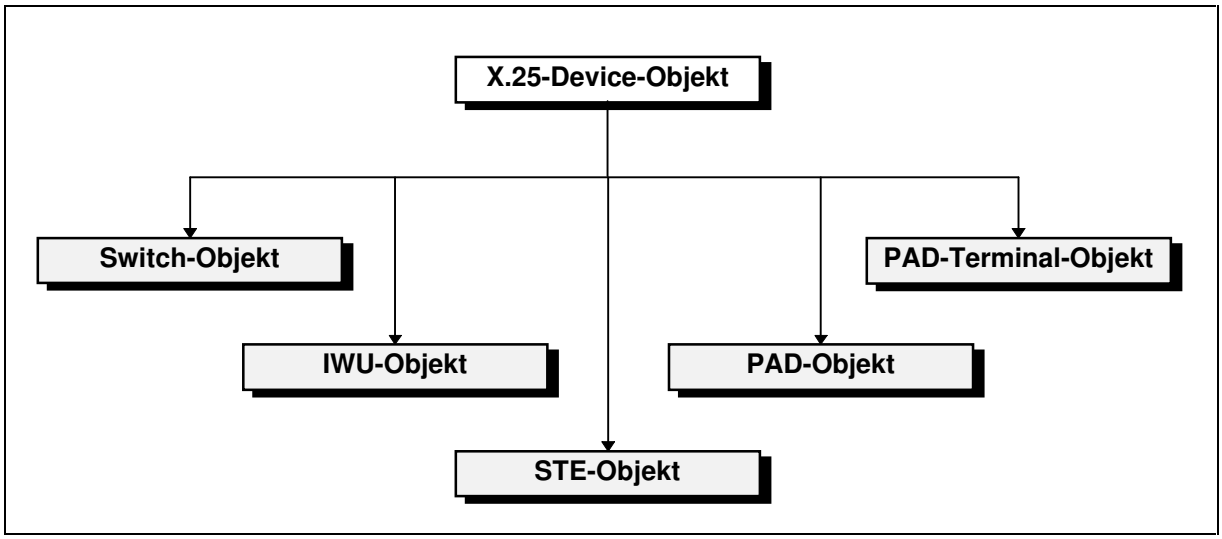


Abbildung 4-15 Gruppe der X.25-Device-Objekte.

Eine Objektklasse, die die Grundlage für weitere Typen bildet, ist das X.25-Device. Wichtigste Eigenschaft ist die Adressierungsmöglichkeit mit einer X.121-Adresse. Von ihr können Switch, IWU, STE, PAD und zeichenorientierte Terminals abgeleitet werden, die an PADs angeschlossen sind. Auch jede Art von X.25-fähigen Endsystemen (DTE) kann diese Klasse verwenden.

4.1.4.1 X.25-Device-Objektklasse

Die Basis für alle Geräte im X.25-Netz ist die X.25-Device-Objektklasse. Ihre grundlegende Eigenschaft ist die Adressierung und Identifizierbarkeit mit einer X.121-Adresse. Von ihr werden alle weiteren Elemente dieser Objektklassengruppe abgeleitet.

Die eindeutige X.121-Adresse eines X.25-Device-Objekts ist das Kriterium für eine korrekte Zuordnungs- und Kontaktaufnahmemöglichkeit innerhalb der Plattform. Sie ist vergleichbar mit einer IP-Adresse in der TCP/IP-Welt. Durch Ableitung von der X.25-Device-Objektklasse können beliebig spezielle X.25-Systemeinheiten kreiert werden. Eine markante Fähigkeit dieser Objektklasse ist die Durchführung eines Pollings über ihre X.121-Adresse zur Funktionsüberwachung.

Datenelement	Bemerkung
Status	ergibt sich aus den Statusfeldern der in Beziehung stehenden Module-Objekten
X.121-Address	Adresse des Geräts

Tabelle 4-8 X.25-Device-Datenelemente.

In Abbildung 4-16 ist die Relation eines X.25-Device-Objekts zu mindestens einem Module-Objekt zu sehen. Diese Eigenschaft vererbt sich auch auf abgeleitete Objekte. Alle abgeleiteten Objektklassen müssen für die Ausstattung mit bestimmten Ports entsprechende Module definieren, die dann über diese Relation verbunden werden können. Das X.25-Device-Objekt selbst hat noch keine direkte Beziehung zu Datenerfassungsobjekten mit Geräte-Funktionalität. Durch diese Eigenschaft zeichnen sich erst abgeleitete Objekte wie z.B. der Switch aus.

Relationsregeln der Objektklasse:

X.25-Device *enthält [1-]* Module

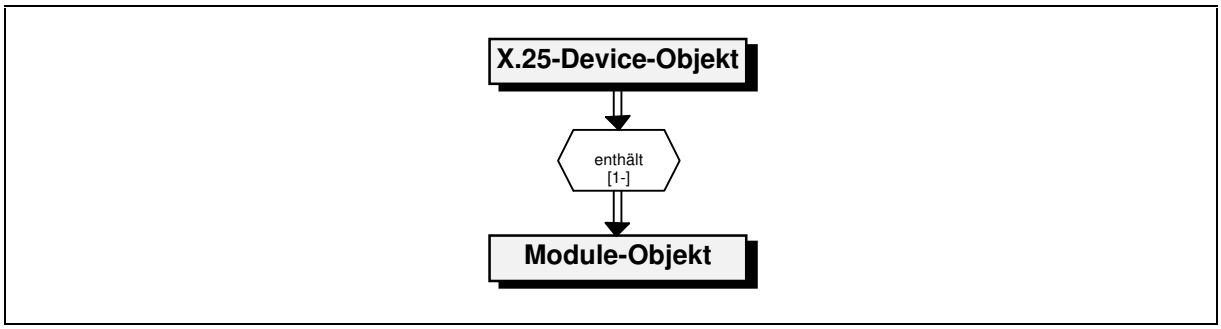


Abbildung 4-16 X.25-Device-Objekt.

4.1.4.2 Switch-Objektklasse

Die Hauptaufgabe eines Switches besteht darin, ankommende Datenpakete auf die korrekten Links weiterzuleiten. Damit ist seine grundlegende Funktionalität beschrieben. Im allgemeinen verfügt der Switch über eine gewisse Anzahl von Ports, an die dann anhand einer Routing-Tabelle die ankommenden Datenpakete entsprechend weitergeschaltet werden. In der Modellwelt wird diese Funktionalität von einem Routing-Objekt übernommen, das in Relation zu einem Switch-Objekt steht. Da ein Switch den Zugangspunkt zum X.25-Netz darstellt, bildet er auch einen Erfassungspunkt für Accounting-Daten. Zusätzlich wird ihm deshalb in Form eines Accounting-Objekts die Möglichkeit zur Erfassung von Acounting-Daten verliehen. Die Relation zu diesem Objekt ist optional, weil nicht davon ausgegangen werden kann, daß jeder Switch mit dieser Funktionalität ausgestattet ist.

Der Aufbau eines Switches kann von einer einfachen Einheit, die nur über grundlegende Module mit X.25-Ports verfügt, bis zur komplexen Multifunktionseinheit reichen. Im letzteren Fall müssen entsprechende Module-Objekte zusammengestellt werden, welche die einzelnen Funktionalitäten abbilden. In Anhang A wird ein Beispiel für den modellierten Aufbau eines Switches gezeigt.

Relationsregeln der Objektklasse:

Switch	*hat Funktionalität [1]*	Routing
Switch	*hat Funktionalität [0-1]*	Accounting

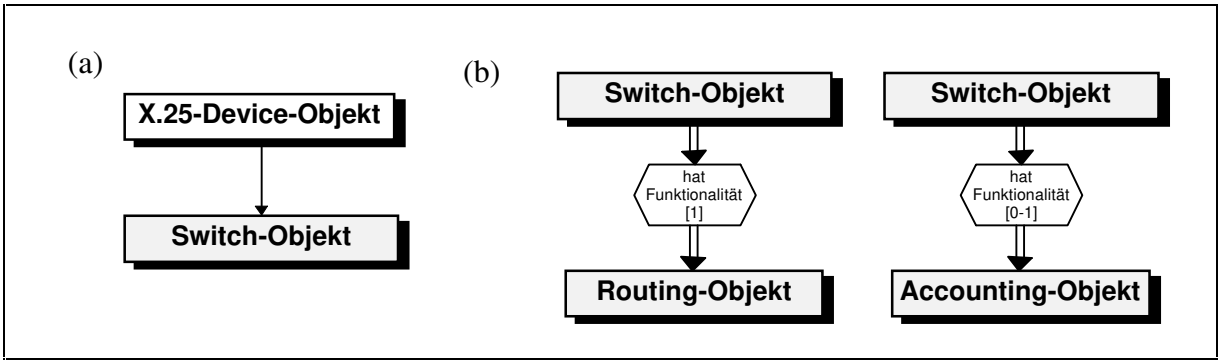


Abbildung 4-17 Switch-Objekt: (a) Vererbung; (b) Relation.

4.1.4.3 IWU-Objektklasse

Den Übergangspunkt vom LAN zum WAN bildet das interworking unit (IWU). Demzufolge verfügt es über Module mit Ports zu LAN-Einheiten und einen X.25-Port zum Anschluß der beiden Netzarten. Für die LAN-Seite werden einfache Port-Objekte verwendet.

Zur Verleihung seiner Geräte-Funktionalität steht es in Relation zu einem Peer-Objekt, das eine Adressumsetzungstabelle beinhaltet. Die Ports werden über entsprechende Module-Objekte eingebunden.

Relationsregeln der Objektklasse:

IWU *hat Funktionalität [1]* Peer

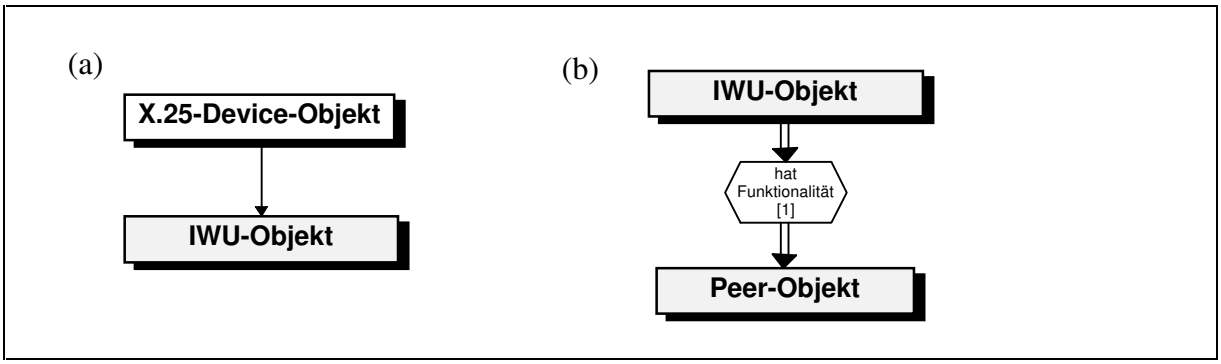


Abbildung 4-18 IWU-Objekt: (a) Vererbung; (b) Relationen.

4.1.4.4 STE-Objektklasse

Durch eine STE(signalling terminal exchange)-Einheit ist es möglich, verschiedene X.25-Netze zu verbinden. Jedes X.25-Subnetz enthält eine eigene STE-Einheit mit einer eindeutigen X.121-Adresse. Je nach Produktpalette und Realisierung der einzelnen Hersteller sind verschiedene Arten von STE-Einheiten denkbar. Zum einen wäre als Integrationslösung die Realisierung in einem Switch durch ein STE-Modul möglich, das durch ein Module-Objekt mit einem X.75-Port definiert wird. In diesem Fall wird kein STE-Objekt benötigt. Als separate Einheit hingegen werden zwei Ports auf entsprechenden Module-Objekten benötigt. Ein X.75-Port

stellt dabei die Verbindung zum anderen X.25-Subnetz dar. Der Kontakt zu einem Switch des eigenen Subnetzes wird z.B. durch einen X.25-Port realisiert.

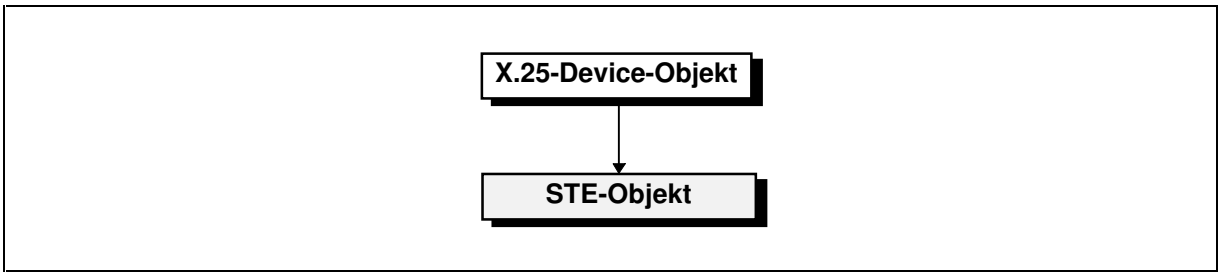


Abbildung 4-19 STE-Objekt.

4.1.4.5 PAD-Objektklasse

Wie auch der Switch benötigt diese Einheit ein Objekt, das seine Wegewahl-Funktionalität ausdrückt. Auch wenn gewöhnlich bei einem PAD nicht von Routing gesprochen wird, erfüllt das Routing-Objekt die entsprechende Aufgabe. Es verteilt in diesem Fall Datenpakete auf die verschiedenen PAD-Ports bzw. leitet diese über einen X.25-Port weiter an einen Switch. Ein PAD-Objekt hat somit mindestens einen X.25-Port und mehrere PAD-Ports, die mit entsprechenden Module-Objekten dem PAD-Objekt untergeordnet werden müssen. Desweiteren verfügt es über das genannte Routing-Objekt.

Relationsregeln der Objektklasse:

PAD *hat Funktionalität [1]* Routing

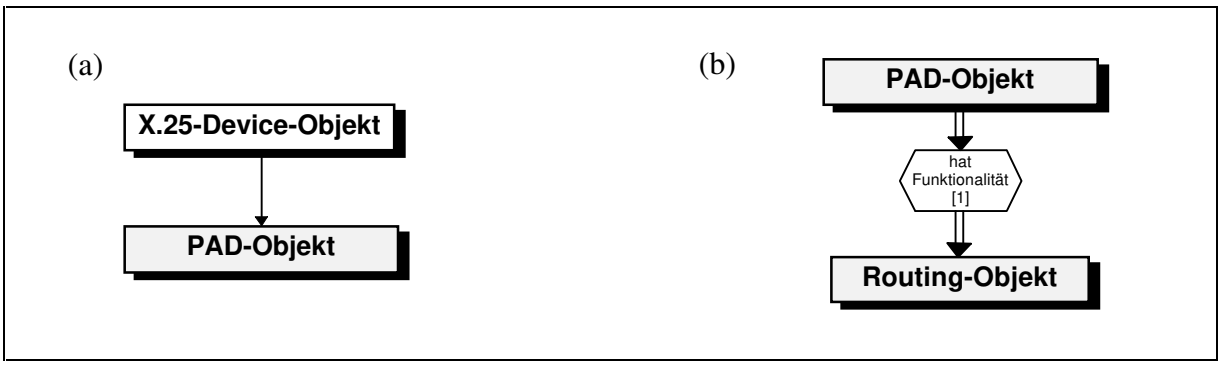


Abbildung 4-20 PAD-Objekt: (a) Vererbung; (b) Relationen.

4.1.4.6 PAD-Terminal-Objektklasse

Das PAD-Terminal-Objekt repräsentiert ein zeichenorientiertes Terminal. Ebenso kann jedoch auch ein Rechner, der ein Terminal emuliert, als PAD-Terminal definiert werden. Es besitzt mindestens einen seriellen Port, der über ein RS-232-like-Link-Objekt mit einem geeigneten Port einer PAD-Einheit verbunden ist. Die Objektklasse enthält ein Access-Datenelement, mit dem das Terminal ein- und abgeschalten werden kann, d.h. es ist möglich, Terminal-Zugänge zu sperren.

Datenelement	Bemerkung
Access	Ein- und Abschalten des Terminal-Zugangs

Tabelle 4-9 PAD-Terminal-Datenelemente.

Die einzige Relation, über die eine Instanz dieser Klasse verfügt, ist die Beziehung zu einem Module-Objekt mit RS-232-like-Port-Objekten.

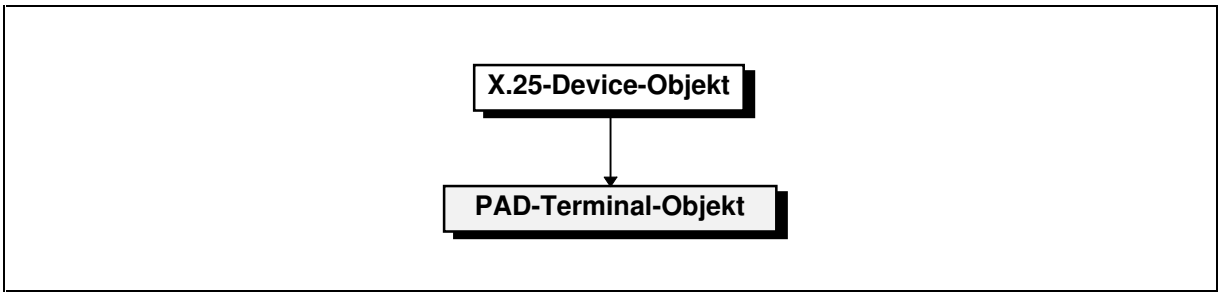


Abbildung 4-21 PAD-Terminal-Objekt.

4.1.5 Die Gruppe der Verbindungstypen

In dieser Gruppe werden alle Arten von Objektklassen definiert, die benötigt werden, um Beziehungen zwischen Datenerfassungsobjekten zweier verbundener Porteinheiten auszudrücken. Mit ihnen können die Funktionsfähigkeit von Verbindungen auf verschiedenen Schichtebenen überprüft und auch einfache Konfigurationsaufgaben durchgeführt werden. Aspekte des Performance Management werden ebenfalls berücksichtigt. Die Basis bildet eine abstrakte Link-Klasse, deren Eigenschaften an die anderen Typen weitergegeben wird.

Von dieser Link-Klasse werden die spezifischen Klassen für die Verbindung der einzelnen Schichtprotokolle abgeleitet. Auf diese Weise werden die RS-232-like-, LAPB- und PLP-Link-Objektklasse eingeführt. Sie verwalten jeweils zwei Datenerfassungsobjekte einer physikalischen Verbindung zweier Ports. In Anhang B wird ein Modellierungsbeispiel für einen Link zwischen zwei X.25-Ports gezeigt.

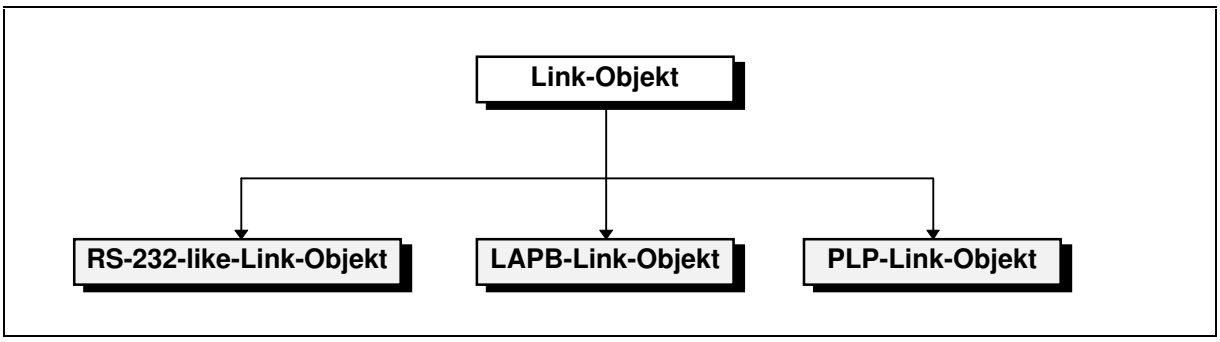


Abbildung 4-22 Gruppe der Link-Objekte.

4.1.5.1 Link-Objektklasse

Das abstrakte Link-Objekt symbolisiert eine Verbindung zwischen zwei gleichen, noch nicht festgelegten Datenerfassungsobjekten, die in Relation zu Port-Objekten stehen. Seine Aufgabe besteht darin, deren Daten zu überwachen und diese auch teilweise zu konfigurieren. Mit Link-Objekten ist es möglich, schichtweise nach Protokollebenen zwei verbundene Port-Objekte zu kontrollieren. Es erfüllt ebenfalls Aufgaben aus dem Bereich des Performance Management durch die Feststellung von Auslastung und Durchsatzleistung eines Links. 'Abstrakt' bedeutet in diesem Fall, daß diese Klasse noch nicht instantiierbar ist.

Um die Funktionsfähigkeit eines Links in Erfahrung zu bringen, stehen zwei Funktionsmechanismen zur Verfügung:

- Polling der Switches und Abfrage der entsprechenden Zustandsfelder der Schicht-Objekte in regelmäßigen Zeitabständen
- Auswertung von automatisch gesendeten Event-Nachrichten der Switches

Der Nachteil des Polling liegt in der Erzeugung zusätzlicher Netzlast (bei in-band management), je nachdem, wie groß das Abfrageintervall gewählt wird. Zusätzlich können kurze Ausfälle, die innerhalb eines Intervalls auftreten, nicht registriert werden. Zuverlässiger ist hier die Event-Auswertung. Es werden Nachrichten nur bei konkreten Störungen gesendet, wodurch die Belastung des Netzes niedrig gehalten und dennoch alle Ausfälle erfaßt werden. Gefahr besteht nur darin, daß gesendete Event-Nachrichten eventuell im Netz verloren gehen.

Der Zustand des Links ergibt sich aus den Zuständen der jeweiligen Datenerfassungsobjekte. Mit dem Link-Objekt vereinfacht sich außer der Überwachung vor allem auch die Konfiguration eines Links. Es bietet die Möglichkeit, allgemeine Konfigurationsdaten zu verändern, die sich dann auf die Einstellungen der Datenerfassungsobjekte auswirken und somit letztendlich auf den Port bzw. Link.

Da ein Link-Objekt indirekt über die Datenerfassungsobjekte mit Port-Objekten in Beziehung steht, können deren Datenelemente für eine schicht-unabhängige Performance-Messung verwendet werden. Es bieten sich hier zum einen als Grundlage die Interface-Bandbreite ifSpeed und zum anderen die Durchsatzwerte ifInOctets bzw. ifOutOctets an. Durch regelmäßiges Abfragen dieser Werte kann die Auslastung eines Ports und damit der Verbindung berechnet werden. Eine andere Möglichkeit eröffnet sich mit der Verwendung von ifInErrors bzw. ifOutErrors zur Bestimmung der durchschnittlichen Fehlerrate.

Die Objektklasse enthält folgende Datenelemente:

Datenelement	Bemerkung
Status	ergibt sich aus Statusfeldern der beiden Datenerfassungsobjekte
DataRate	Durchsatzrate des Links
ErrorRate	Fehlerrate des Links

Tabelle 4-10 Link-Datenelemente.

4.1.5.2 RS-232-like-Link-Objektklasse

Durch eine direkte Ableitung von der Link-Objektklasse entsteht die RS-232-like-Link-Objektklasse. Sie kontrolliert eine Verbindung zwischen RS-232-like-Objekten zweier Ports.

Relationsregeln der Objektklasse:

RS-232-like-Link *kontrolliert [2]* RS-232-like

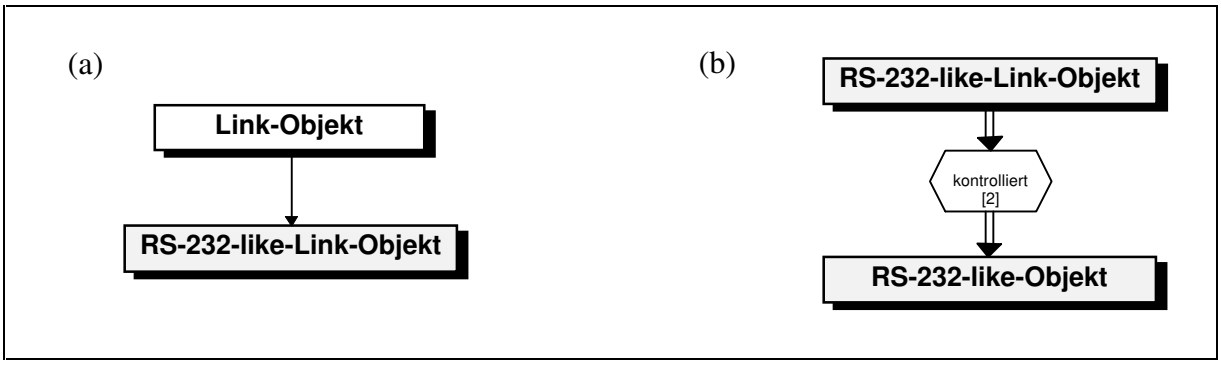


Abbildung 4-23 RS-232-like-Link-Objekt: (a) Vererbung; (b) Relationen.

Ihre Datenelemente beziehen sich auf entsprechende Datenelemente der verwalteten RS-232-like-Objekte. Eine Veränderung dieser Werte zieht automatisch auch eine Veränderung der Einstellungen der verbundenen Objekte nach sich.

Datenelement	Bemerkung
rs232Speed	speed in bits per second
rs232AsyncBits	number of bits in a character
rs232AsyncStopBits	number of stop bits
rs232Parity	parity bit

Tabelle 4-11 RS-232-like-Link-Datenelemente.

4.1.5.3 LAPB-Link-Objektklasse

Ein LAPB-Link-Objekt symbolisiert eine Schicht-2-Verbindung zwischen LAPB-Objekten zweier Ports. Die Objektklasse wird von der abstrakten Link-Objektklasse abgeleitet. Sie kontrolliert LAPB-Objekte zweier Ports.

Relationsregeln der Objektklasse:

LAPB-Link *kontrolliert [2]* LAPB

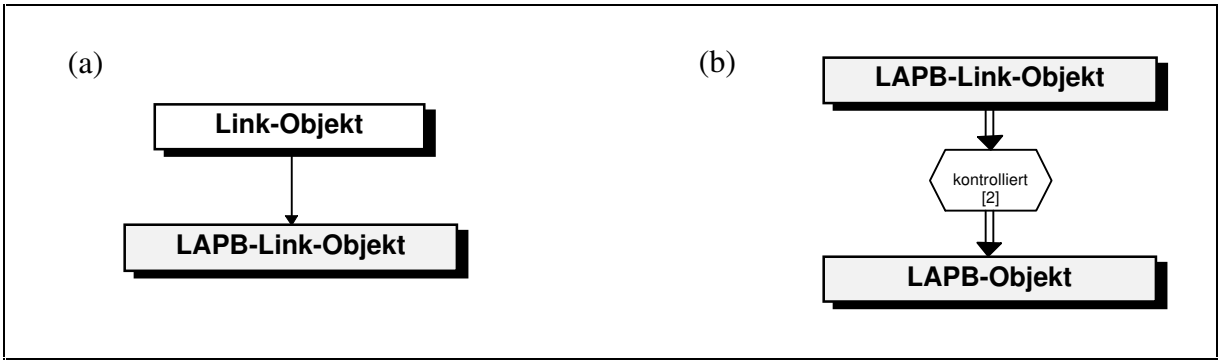


Abbildung 4-24 LAPB-Link-Objekt: (a) Vererbung; (b) Relationen.

Der Datenbereich der Objektklassen umfaßt LAPB-spezifische Elemente. Für sie gilt wieder die gleiche Eigenschaft wie bereits zuvor beim RS-232-like-Objekt. Eine Veränderung dieser Werte wirkt sich auf die entsprechenden Datenelemente der LAPB-Objekte aus, die mit diesem Link-Objekt durch Relation verbunden sind.

Datenelement	Bemerkung
lapbKWindowSize	sliding window mechanism
lapbN1FrameSize	max. framesize

Tabelle 4-12 LAPB-Link-Datenelemente.

4.1.5.4 PLP-Link-Objektklasse

Mit der PLP-Link-Objektklasse wird die Gruppe der Verbindungstypen vervollständigt. Mit ihr kann eine Schicht-3-Verbindung zwischen zwei PLP-Objekten aufgebaut werden. Die Objektklasse erbt die Eigenschaften der Link-Objektklasse und steht in Relation zu zwei PLP-Objekten.

Relationsregeln der Objektklasse:

PLP-Link *kontrolliert [2]* PLP

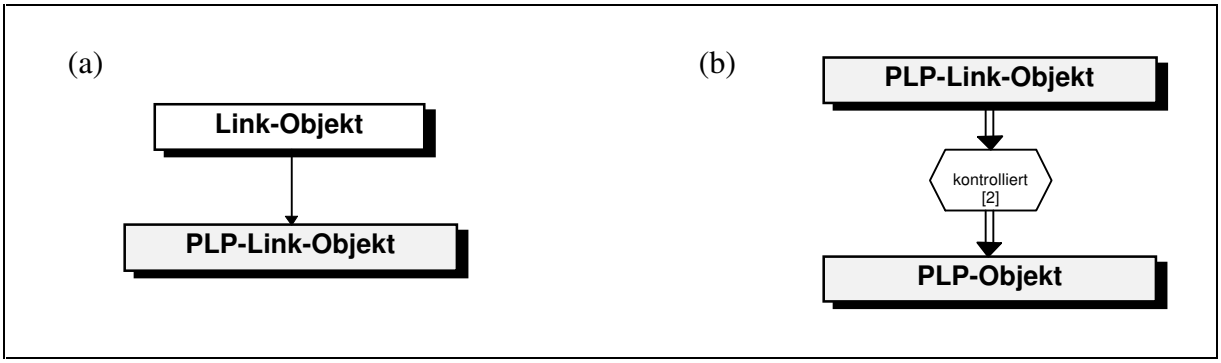


Abbildung 4-25 PLP-Link-Objekt: (a) Vererbung; (b) Relationen.

Die Objektklasse wird nur zur Vollständigkeit und für eine eventuelle spätere Erweiterung im Modul definiert, da keine Datenelemente vorhanden sind, für die eine gemeinsame Konfiguration und Überwachung von zwei PLP-Objekten sinnvoll erscheint.

4.1.6 Die Gruppe der Netzeinheiten

Um das X.25-Netz in seiner Gesamtheit zu beschreiben und in geeigneter Weise in Relation zu anderen Netztypen zu bringen, wird die Objektklasse X.25-Net eingeführt. Innerhalb eines X.25-Netzes ist es zur besseren logischen Strukturierung ebenfalls möglich, untergeordnete Netze zu bilden. Hierzu wird die Objektklasse X.25-Subnet definiert.

4.1.6.1 X.25-Net-Objektklasse

Die Einordnung eines X.25-Netzes in die Netzumgebung der Management-Plattform geschieht durch ein X.25-Net-Objekt. Es umfaßt alle X.25-Device- und Link-Objekte eines X.25-Netzes. Zur weiteren Strukturierung können Subnetze aufgenommen werden. Ein X.25-Net-Objekt steht zu allen diesen Objekten in direkter Beziehung und enthält keine zusätzlichen speziellen Datenelemente. Obwohl die Möglichkeit gegeben ist, vollständige Netzlandschaften ohne jegliche Subnetze zu erstellen, sollte es primär als Gruppierungsobjekt von Subnetzen eingesetzt werden. Es hat den höchsten Rang in der Gliederungs-Hierarchie, d.h. es ist nicht möglich, X.25-Net-Objekte nochmals zusammenzufassen. Hierfür müssen X.25-Subnet-Objekte geeignet angelegt werden.

Relationsregeln der Objektklasse:

X.25-Net	*enthält [0-]*	X.25-Subnet
X.25-Net	*enthält [0-]*	Link
X.25-Net	*enthält [0-]*	X.25-Device

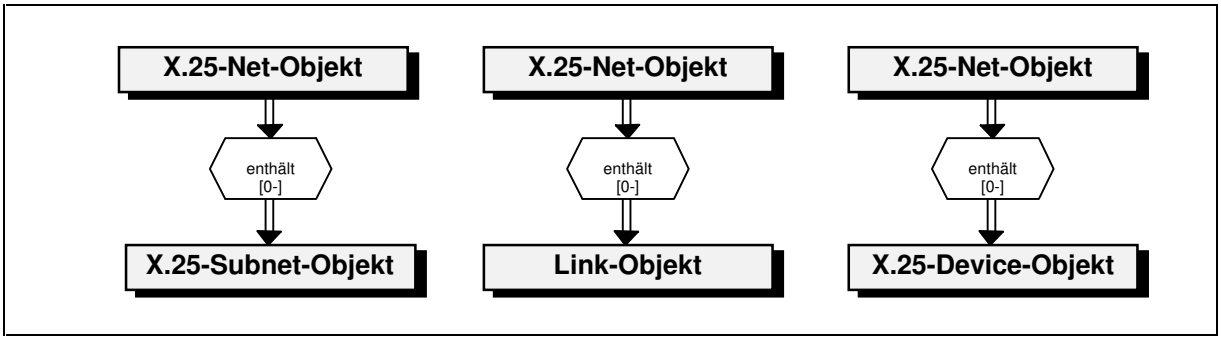


Abbildung 4-26 X.25-Net-Objekt.

4.1.6.2 X.25-Subnet-Objektklasse

Mit einem X.25-Subnet-Objekt können X.25-Device- und Link-Objekte eines X.25-Netzes zusammengefaßt werden. Genau wie das X.25-Net-Objekt ist es durch Relation mit diesen Objekten verknüpft. Es dient der sinnvollen Untergliederung und Strukturierung von komplexen X.25-Netzen.

Relationsregeln der Objektklasse:

X.25-Subnet	*enthält [0-]*	Link
X.25-Subnet	*enthält [0-]*	X.25-Device

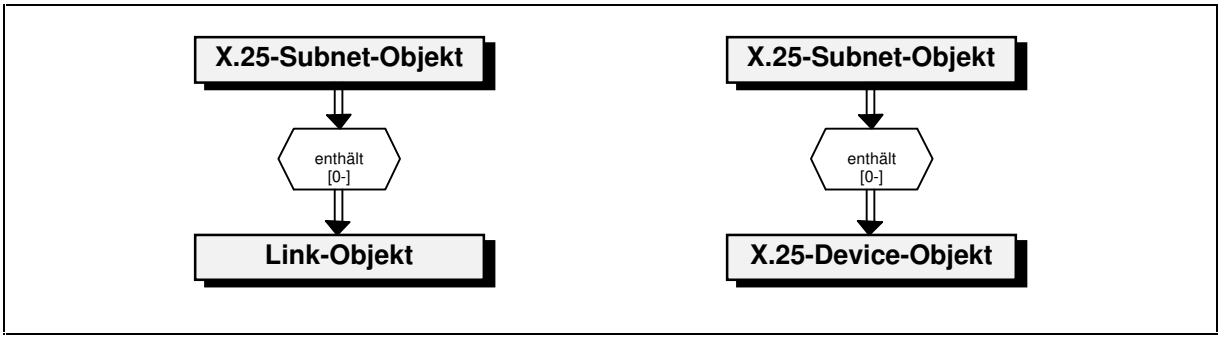


Abbildung 4-27 X.25-Subnet-Objekt.

4.2 Applikationen

In den folgenden Unterkapiteln werden Applikationen für das Fault und Accounting Management in einem integrierten X.25-Netz beschrieben. Zu jeder Applikation werden mögliche Aufgabengebiete vorgestellt und Lösungsansätze in der zuvor beschriebenen Objektwelt aufgezeigt.

4.2.1 Fault Management

Innerhalb des X.25-Netzes liegt die Hauptaufgabe des Fault Management in der Überwachung der Links zwischen den Switches und der Kontrolle der virtuellen Verbindungen zwischen den Endgeräten des Netzes. Die verbindungsorientierte Übertragung ist eines der wichtigsten Charaktermerkmale eines X.25-Netzes. Im Gegensatz zu lokalen Netzen muß vor der eigentlichen Datenübertragung zwischen den beteiligten X.25-Endgeräten eine virtuelle Verbindung (virtual circuit) aufgebaut werden. Das Management dieser virtual circuits ist der Schwerpunkt für die Applikation, die in diesem Kapitel beschrieben wird. Sie soll dem Netzbetreiber ermöglichen, die verschiedenen Aufgaben, die bei deren Verwaltung anfallen, zu vereinfachen. Wie in Kapitel 3 „Anforderungen an ein Management-Modell“ beschrieben, wird auch der netzübergreifende Aspekt berücksichtigt.

Bevor jedoch damit begonnen wird, ein Management-Konzept für eine derartige Applikation zu entwerfen, wird zunächst die Darstellung der virtuellen Verbindung in der Objektwelt des X.25-Management-Moduls verdeutlicht.

4.2.1.1 Objektdarstellung

Wie in Kapitel 2.1.2 geschildert erstreckt sich eine virtuelle Verbindung zumeist über mehrere Knotenpunkte und Links. In Abbildung 4-28(a) wird diese Vorgehensweise exemplarisch anhand eines Ausschnitts der in Abbildung 4-4 verwendeten Netzlandschaft verdeutlicht. Ausgehend von einem X.25-Endgerät gelangt man über dessen Port zum Port des ersten Switches. Der richtige Ausgangsport des Switches ergibt sich aus seiner Routing-Tabelle. In diesem Fall läuft der Weg über einen weiteren Switch, an dem schließlich eine PAD-Einheit mit dem gesuchten Zielterminal angeschlossen ist. In diesem Beispiel wurde vorausgesetzt, daß auch die

Verbindungen zwischen Switches auf dem X.25-Protokoll beruhen und somit durch X.25-Port-Objekte darstellbar sind.

Eine andere Ansicht findet sich in Abbildung 4-28(b) wieder. Hierbei wird die gleiche virtuelle Verbindung mit den zur Verfügung stehenden Objekten und Relationen beschrieben. Es bildet sich zwischen den Endpunkten eine ähnliche zusammenhängende Folge wie zuvor bei der symbolischen Darstellung. Durch die Dreiteilung der Link-Objekte ist eine Untersuchung oder Verfolgung einer virtuellen Verbindung auf verschiedenen Schichtebenen möglich. Je nach Sichtweise und Anforderung kann somit ein Weg durch ein X.25-Netz auf verschiedenen Protokollebenen beschrieben werden.

Der erste Weg ergibt sich durch die Verknüpfung der Port-Objekte mit der „ist verbunden mit“-Relation. Ohne Berücksichtigung der beteiligten Protokollarten kann hiermit vom Anfangs- zum Endpunkt einer virtuellen Verbindung gelangt werden. Die andere Möglichkeit besteht in der Verfolgung der Link-Objekte auf den verschiedenen Protokollebenen. Die schichtweise Verknüpfung der Datenerfassungsobjekte der Ports erlaubt eine durchgehende Betrachtung der virtuellen Verbindung im X.25-Netz. Durch diese Eigenschaft der Netzbeschreibung ist es mit entsprechenden Applikationen möglich, eine nach Ebenen gestaffelte Fehlersuche durchzuführen.

Bereits mit der Kontrolle der einzelnen Link-Objekte des Management-Moduls kann eine wichtige Aufgabe des Fault Management erfüllt werden. Durch geeignete graphische Repräsentation und Meldemechanismen ist eine Überwachung der einzelnen Links möglich. Ausfälle können aufgespürt und deren Ursachen auf den verschiedenen Schichten gesucht werden.

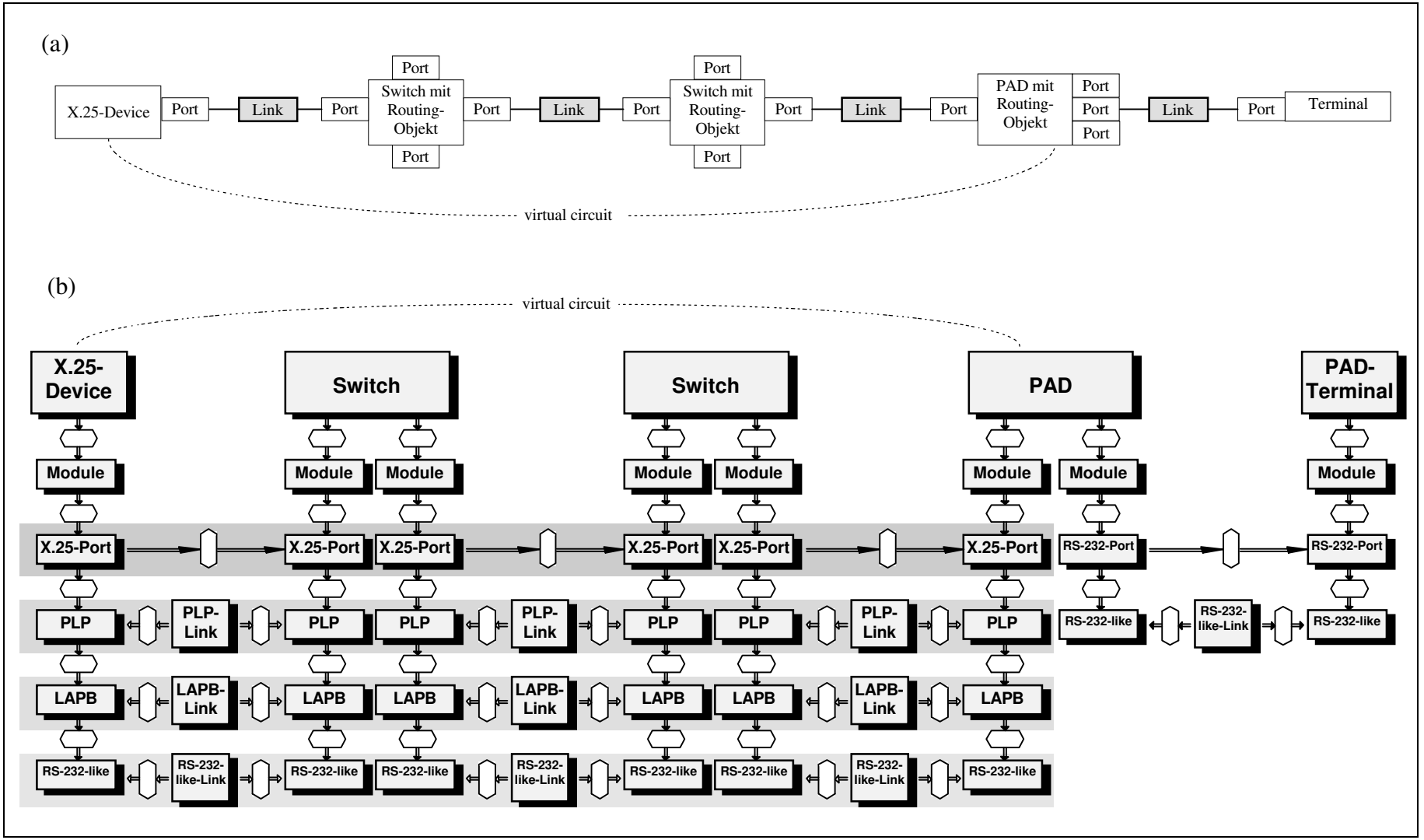


Abbildung 4-28 virtual circuit: (a) Symboldarstellung; (b) Objektdarstellung.

4.2.1.2 Verwendung einer Datenbank

Um die Zusammenhänge zwischen Endeinheiten und einem X.25-Netz zu erfassen, wie sie in Abbildung 3-2 gezeigt wurden, eignet sich die Verwendung einer Datenbank. Im folgenden wird davon ausgegangen, daß die Routing-Tabellen so konfiguriert sind, daß keine Einschränkungen für die Wegewahl der Verbindungen der Endeinheiten auftreten. Das bedeutet unter anderem, daß ein Verbindungsaufbau zwischen zwei Endeinheiten über alle verfügbaren Links möglich ist. Beschränkungen von Teilstrecken auf bestimmte Endgeräte sollten daher nicht gelten. In diesem Fall würde die Gestaltung der Datenbank weitaus komplexer, da die Sonderkonfigurationen aller Routing-Tabellen berücksichtigt werden müßten.

Für die Beschreibung der Zusammenhänge sind zwei Tabellen notwendig:

- Erfassung aller Links zwischen Switches:
  Diese Einträge können durch Zufügen von spezifischen Eigenschaften einer Verbindung erweitert werden (z.B. Übertragungsgeschwindigkeit).
- Erfassung aller Anbindungen von Endeinheiten an die Zugangspunkte (Switches), z.B. in der Form: LAN A - Switch A; Großrechner 1 - Switch G.

Mit einer Abfrage aus der Kombination dieser Tabellen ist es möglich, Konsequenzen in Erfahrung zu bringen, die sich aus dem Ausfall eines Links auf Endeinheiten ergeben. Hierzu ist ein geeigneter Algorithmus erforderlich, der für alle Endeinheiten prüft, ob der Ausfall Auswirkungen auf eine End-to-End-Verbindung hat oder ob durch den Einsatz von transitiven Wegewahlmöglichkeiten ein backup gefunden werden kann.

4.2.1.3 Management-Konzept

Nach der Behandlung der Grundlagen folgt nun eine Aufstellung der Management-Möglichkeiten und -Aufgaben, die sich mit der Verwaltung von Links und virtuellen Verbindungen ergeben.

- **Generelle Überwachung von Links zur Kontrolle von netzübergreifenden Konsequenzen auf Endeinheiten:**
  Außer der direkten Kontrolle von einzelnen Links, die durch die Link-Objekte im X.25-Management-Modul bereits gegeben ist, können Verbindungen im Hinblick auf netzübergreifende Auswirkungen überwacht werden. Bei einem Ausfall eines bestimmten Link kann mit den vorhandenen Datenbanktabellen ermittelt werden, ob es sich um einen überbrückbaren Ausfall innerhalb des X.25-Netzes handelt oder sich die Störung auf Endeinheiten auswirkt. Im letzteren Fall werden die betroffenen Endeinheiten und deren gestörte End-to-End-Beziehung ausfindig gemacht.
- **Überwachung einer virtuellen Verbindung zwischen zwei Endeinheiten auf verschiedenen Protokollebenen:**
  Diese Aufgabe kann unter zwei verschiedenen Aspekten betrachtet werden. Zum einen soll es möglich sein, daß eine Verbindung zwischen zwei Punkten überwacht werden soll, also eine bestehende virtuelle Verbindung (z.B. permanent VC). Zum anderen

kann auch die generelle Überwachung einer Verbindungsstrecke zwischen zwei Standorten / Endeinheiten erwünscht sein, ohne daß eine bestimmte virtuelle Verbindung existiert. Für die Erfüllung dieser Aufgabe spielen diese Betrachtungsweisen jedoch keine Rolle, wenn von der Darstellung der virtuellen Verbindung in der Objektwelt des X.25-Management-Moduls ausgegangen wird (siehe Abbildung 4-28(b)). Beide Möglichkeiten werden in der Objektwelt auf die gleiche Art und Weise behandelt.

Betrachtet man die Zustände alle Link-Objekte, über die sich eine virtuelle Verbindung von X.25-Device A zu X.25-Device B erstreckt, so ergibt sich dadurch der Gesamtzustand der gesamten Verbindungsstrecke. Sonderfälle treten auf, falls die Wegewahl innerhalb eines Switches nicht eindeutig ist. Auf diese Weise wird im allgemeinen ein backup realisiert, d.h. eine Ausweichmöglichkeit, um auf einem weiteren Weg das Endgerät mit der Zieladresse zu erreichen, wenn auf dem Hauptweg eine Störung aufgetreten ist.

Der erste Schritt für die Applikation besteht darin, die erforderlichen Link-Objekte zu finden, die zwischen den Endpunkten der Verbindung liegen. Die Endpunkte werden eindeutig durch zwei X.121-Adressen identifiziert. Beginnend mit der Adresse, von der der Verbindungswunsch ausgeht, müssen schrittweise die einzelnen Knotenpunkte bis zum Erreichen der Endadresse durchwandert und dabei gefundene Link-Objekte in eine Überwachungsliste eingetragen werden.

Um alle erforderlichen Link-Objekte festzustellen, kann auf die angelegten Datenbanktabellen zurückgegriffen werden. Mit ihnen können alle Wege und mögliche Alternativen zwischen den Endeinheiten ermittelt werden. Erst nachdem alle Link-Objekte bekannt sind und die Überwachungsliste vollständig ist, kann die eigentliche Aufgabe der Verbindungsüberwachung erfüllt werden. Beim Ausfallen eines Links wird überprüft, ob sich dieser in der Liste befindet und dadurch eventuell eine Störung der virtuellen Verbindungsstrecke nach sich gezogen wird.

Abgesehen von der direkten Kontrolle einer virtuellen Verbindung kann diese Management-Aufgabe auch als Integrationslösung für den X.25-Bereich in einer netzübergreifenden End-to-End-Verbindung gesehen werden. Die Überwachung des virtual circuit entspricht dann einem Teil aus der verketteten Gesamtverbindung, die sich aus verschiedenen LAN- und WAN-Abschnitten ergeben kann.

- **Fehlersuche bei Störungen im Vebindungsaufbau / Überprüfen der Wegewahl eines virtual call durch ein X.25-Netz:**
  Außer durch Link-Ausfälle verursachte Störungen können Fehler durch Falsch-Konfiguration von Routing-Tabellen auftreten. Eine Möglichkeit, um die Wegewahl eines virtual call beim Verbindungsaufbau nachzuvollziehen, ist die schrittweise Überprüfung der Tabellen der Switches, die an der Verbindungsvermittlung beteiligt sind. Mit den Objekten des Management-Moduls kann diese Aufgabe automatisch ausgeführt und Unstimmigkeiten aufgezeigt werden. Hierfür werden nur die X.121-Adressen der Endeinheiten benötigt, zwischen denen überprüft werden soll, und eventuell für diese Verbindung notwendige facilities, falls auch diese in den Routing-Tabellen berücksichtigt werden.

4.2.2 Accounting Management

Das Ziel des Accounting Management ist die Zuordnung von Kosten an die Verursacher des Netzverkehrs. Prinzipiell können hierbei zwei Methoden unterschieden werden. Bei der statischen Abrechnung wird nur der Netzzugang abgerechnet, d.h. die in Wirklichkeit verursachte Netzlast wird nicht beachtet. Die dynamische Methode orientiert sich an Messungen, mit denen die wirkliche Nutzung des Netzes festgestellt wird. In diesem Kapitel wird auf diese Art der Kostenzuordnung eingegangen.

Die Daten über die Nutzung des X.25-Netzes werden anhand der aufgezeichneten virtuellen Verbindungen ermittelt, die mit den Accounting-Objekten an den Zugangspunkten gesammelt werden.

Es stehen folgende Daten zur Verfügung:

- Daten über die Netznutzer durch die Adressen der bei der virtuellen Verbindung beteiligten Endeinheiten.
- Daten über die Nutzungsdauer durch die Angabe von Start- und Endzeit einer Verbindung.
- Daten über das Nutzungsvolumen durch die Erfassung der übertragenen Datenpakete und Bytes.

Für den Netzbetreiber ist eine Zusammenfassung dieser Daten notwendig, die in der Regel in periodischen Zeitabständen (z.B. jeden Tag, Woche, Monat) durchgeführt werden muß. Das dafür verwendete Kriterium sind die Kostenverursacher, in diesem Fall die Netzteilnehmer (siehe Abbildung 4-29).

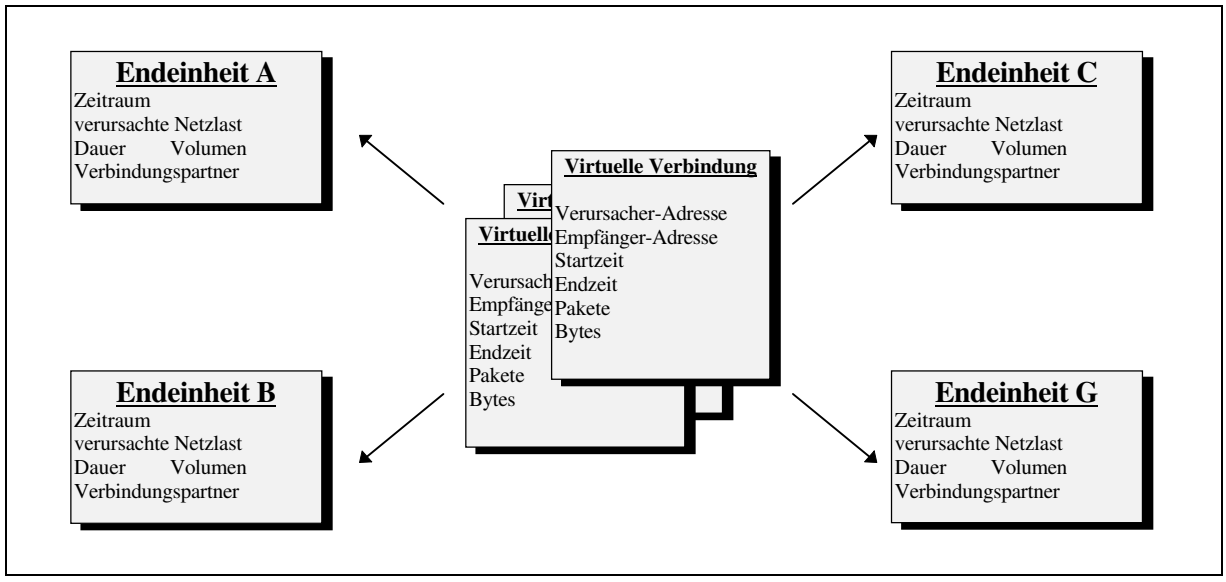


Abbildung 4-29 Zusammenfassung der Accounting-Daten.

Die Daten über die Netznutzer bestehen aus der Adresse des Verursachers, der den virtuellen call zum Verbindungsaufbau abgesetzt hat, und der Adresse der zu erreichenden Gegenstelle. Bei der Zuordnung, wer die Rolle des Kostenverursachers übernimmt, gibt es verschiedene

Möglichkeiten. Einerseits kann der Absender des virtual call als alleiniger Kostenverursacher festgesetzt werden. Sinnvoll wäre dieser Ansatz z.B. bei einem Anbieter einer Online-Datenbank, der an ein Paketnetz angeschlossen ist. Der gesamte Netzverkehr, der durch Anfragen an diese Datenbank verursacht wurde, wird dann beim Kunden abgerechnet. Andererseits können die Nutzungskosten auch auf beide Endeinheiten der virtuellen Verbindung aufgeteilt werden.

Die Einheit der Abrechnung kann sich nach folgenden Kriterien richten:

- Nutzungsdauer:
  Hierfür wird für alle Einträge aus Start- und Endzeit die Dauer der virtuellen Verbindung berechnet und schließlich addiert. Man erhält die Gesamtnutzungsdauer, in der die jeweilige Endeinheit während des vorher festgelegten Zeitraums das X.25-Netz in Anspruch genommen hat.
- Nutzungsvolumen:
  Die übertragenen Pakete oder Bytes der Einträge werden addiert. Das Ergebnis ist das gesamte Datenvolumen, das von einer Endeinheit über das X.25-Netz übertragen wurde.

Wie auch zuvor bei der Applikation des Fault Management bietet sich hier die Verwendung einer Datenbank zur Sortierung und Verwaltung der Daten an. Die gesammelten Daten der Accounting-Objekte müssen hierfür in eine Datenbanktabelle geschrieben werden. Der Vorteil dieser Methode liegt in der Mächtigkeit einer Datenbankumgebung, in der Datengruppierungen nach bestimmten Kriterien auf leichte Weise vollzogen werden können. Zusätzlich steht mit der Datenbanksprache SQL eine standardisierte Schnittstelle zur Abfrage und Weiterverarbeitung der Daten durch andere Management-Anwendungen zur Verfügung.

5 Wege der Datenintegration

Der Weg, um die Daten eines X.25-Netzes integrieren zu können, ist abhängig von den zur Verfügung stehenden Management-Schnittstellen der X.25-Komponenten. Die einfachste Möglichkeit bietet sich beim in-band management mit einem Proxy-Agenten, der über ein standardisiertes Management-Protokoll angesprochen wird, das auch von der Plattform unterstützt wird.

Steht diese Möglichkeit nicht zur Verfügung, so muß mit einem eigens zu erstellenden Schnittstellen-Modul auf eine Schnittstelle des herstellerspezifischen Management-Werkzeugs zugegriffen werden. Hierzu muß ein separates Modul geschaffen werden, das sowohl die Daten aus dem X.25-Netz erfaßt und in geeigneter Form an eine Datenschnittstelle der Plattform weiterleitet, als auch Befehle und Anfragen der Plattform entsprechend dem X.25-Management-Werkzeug umsetzt. Dieses Modul muß im Gegensatz zum Modul für die Plattform für jeden X.25-Hersteller neu erzeugt werden, da die Ansteuerungsbefehle und -möglichkeiten der X.25-Komponenten in der Regel völlig unterschiedlich aufgebaut sind.

5.1 SNMP-Proxy-Agent

Da SNMP das zur Zeit am weitesten verbreitete Management-Protokoll ist, wird es auch von allen Plattformen unterstützt. Eine Datenintegration über diese Schnittstelle ist deshalb am einfachsten zu realisieren.

SNMP bietet folgende Möglichkeiten:

- Auslesen von Management-Information durch GetRequest- und GetNextRequest-PDU.
- Setzen von Management-Information durch SetRequest-PDU.
- Antworten des SNMP-Agenten auf Abfragen mit GetResponse-PDU.
- eigenständiges Senden des SNMP-Agent von Trap-Nachrichten.

Genauere Beschreibungen des Management-Protokolls und dessen Anwendung finden sich in [RFC1157] und [ROSE].

Ein SNMP-Proxy-Agent für ein X.25-Netz übernimmt die Aufgabe, die Management-Daten des gesamten Netzes in standardisierter Form zugänglich zu machen. Die wichtigste Eigenschaft wäre die Implementierung der MIB-II [RFC1213], mit der bereits eine Verwaltung aller Interfaces möglich ist. Die Datenerfassungstypen des X.25-Management-Moduls beruhen hauptsächlich auf Standard-MIB-Beschreibungen.

Die für die Objektklassen benötigten spezifischen X.25-MIBs sind in den Quellen

- [RFC1317] für die RS-232-like-Objektklasse,
- [RFC1381] für die LAPB-Objektklasse,
- [RFC1382] für die PLP-Objektklasse,
- und [RFC1356] für die Peer-Objektklasse enthalten.

Eine Implementierung dieser MIBs ist ebenso für den vollständigen Einsatz des X.25-Management-Moduls erforderlich. Für die weiteren Objektklassen zur Datenerfassung (Accounting-, Routing- und MLP-Objektklasse) existieren keine standardisierten Beschreibungen, so daß auf herstellerspezifische MIBs zurückgegriffen werden muß.

5.2 Beschreibung eines Schnittstellen-Moduls

Die folgen Unterkapitel beschäftigen sich mit dem Eigenentwurf eines Schnittstellen-Moduls. Voraussetzung für den Einsatz des Moduls ist eine homogene X.25-Umgebung, d.h. alle X.25-Geräte werden zentral von einem herstellerspezifischen Management-Werkzeug verwaltet. Sollten unterschiedliche Komponenten verschiedener Hersteller verwendet werden, so ist für jedes Management-Werkzeug ein passendes Schnittstellen-Modul zu realisieren.

Weitere Voraussetzung auf seiten der Plattform ist das Vorhandensein einer geeigneten programmierbaren Integrationsschnittstelle für die Anbindung des Moduls.

5.2.1 Aufgaben

Die Aufgabe des Schnittstellen-Moduls läßt sich einfach beschreiben. Es soll die Funktion des Übersetzers zwischen Plattform und herstellerspezifischem X.25-Management-Werkzeug übernehmen. Innerhalb dieser Kommunikationsbeziehung treten verschiedene Situationen auf, die sich aus den möglichen Datenbewegungen in diesem Modul ergeben:

- **von der Plattform ausgelöste Befehle:**
  In diesen Bereich fallen Befehle, die von der Plattform an das X.25-Netz abgesetzt werden und keine direkte Antwort von diesem zurückerwarten. Das Modul muß hierfür den Befehl umwandeln und an das X.25-Netz weitergeben. Beispiel hierfür wären Kommandos zur Konfiguration von Ports oder Geräten. Bezüglich des beschriebenen X.25-Management-Moduls handelt es sich um das Setzen und Verändern von Datenelementen eines Objekts, das auf die entsprechenden real verwalteten Elemente des Netzes übertragen werden muß.
- **von der Plattform ausgelöste Abfragen:**
  Die wichtigste und auch komplexeste Aufgabe stellt sich in diesem Bereich, da alle Daten, die von der Plattform benötigt und verwaltet werden, mit dieser Methode integriert werden müssen. Im Gegensatz zu den anderen Bereichen läuft hier der Datenweg zweimal durch das Schnittstellen-Modul. Zu Beginn wird von der Plattform eine Ab-

frage an das Modul gerichtet. Es kann sich dabei um immerwiederkehrendes Polling zur Geräte-, Komponenten- oder Link-Überwachung oder auch um einmaliges Anfordern von speziellen Daten, wie z.B. Routing-Tabellen, handeln, das in der Regel nur bei Bedarf geschieht. Das Modul übersetzt die Anfrage in eine Befehlsequenz, die vom X.25-Management-Werkzeug verstanden werden kann, und leitet sie an dieses weiter. Das X.25-Management-Werkzeug wiederum beantwortet die Anfrage und schickt die entsprechenden Daten an das Modul zurück. Im Modul werden die Daten ausgewertet und nach einer Umwandlung an eine Schnittstelle der Plattform übergeben.

- **vom X.25-Management-Werkzeug verursachte Meldungen:**
  Vom X.25-Netz werden im allgemeinen bestimmte Ereignisse automatisch erfaßt und gemeldet. Es handelt sich dabei um evt. Störfälle und Ausfallmeldungen des Netzes oder um Nachrichten, die über eingetretene Ereignisse, z.B. einem Verbindungsaufbau, informieren. Sie beeinflussen somit direkt Datenelemente der betroffenen Objekte. Der Datenweg läuft vom X.25-Management-Werkzeug zum Modul, in dem eine Auswertung und anschließende Übersetzung stattfindet, und letztendlich zur Plattform.

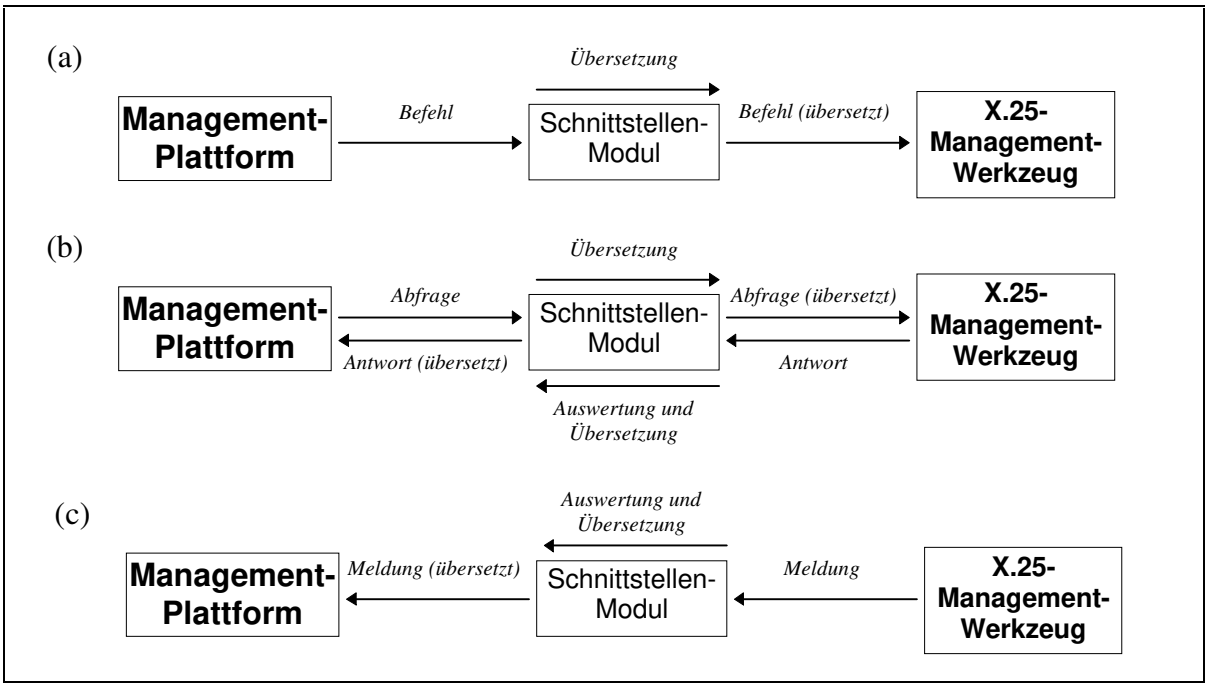


Abbildung 5-1 Datenwege: (a) Befehle; (b) Abfragen; (c) Meldungen.

5.2.2 Aufbau

Für den Aufbau des Schnittstellen-Moduls sind die Zugangsmöglichkeiten von Plattform und X.25-Management-Werkzeug ausschlaggebend. Grundsätzlich sind zwei Formen denkbar:

- textuelle Schnittstelle: der Zugang über ein command line interface (CLI) oder ein Terminal.

- programmierbare Schnittstelle: ein direkter Zugang über eine Funktionsbibliothek (API) einer Programmiersprache, die vom Hersteller angeboten wird.

Obwohl die Verwendung einer API mit Sicherheit am komfortabelsten und elegantesten scheint, steht sie im allgemeinen auf seiten des X.25-Management-Werkzeugs selten zur Verfügung. Im folgenden wird davon ausgegangen, daß jedoch zumindest die Plattform einen derartigen Zugang anbietet.

Die Grundstruktur des Moduls ist in Abbildung 5-2 zu sehen. Sie weist im wesentlichen drei Teilbereiche auf. Die Bereiche A und C stellen die Schnittstellen zu Plattform und X.25-Management-Werkzeug dar. Bei letzterem wird die Möglichkeit berücksichtigt, einen eventuell vorhandenen API-Zugang oder einen textuellen CLI-Zugang zu verwenden. Die CLI-Variante erscheint auf den ersten Blick umständlich und langwierig. Sie bietet jedoch durch Konfigurationsdateien und -schablonen mehr Aspekte zur Verwendung bei X.25-Netzen verschiedener Hersteller. Teilbereich B beinhaltet im wesentlichen Analyse-, Umsetzungs- und Verarbeitungs-Komponenten. Auch hier muß bezüglich der beiden Varianten unterschieden werden.

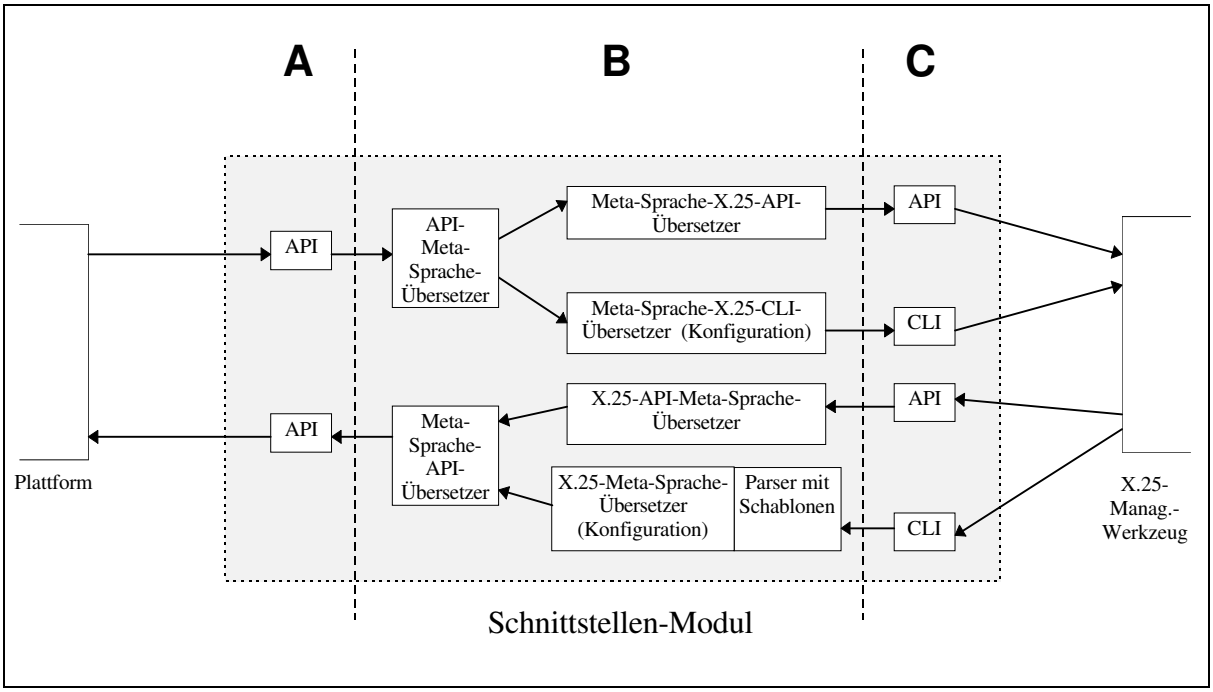


Abbildung 5-2 Grundstruktur des Schnittstellen-Moduls.

Die Komponenten der einzelnen Teilbereiche werden nun genauer beschrieben.

Teilbereich A:

- **API:**
  Für die Anbindung des Schnittstellen-Moduls an die Plattform wird eine Funktionsbibliothek der Plattform benötigt, die den Zugriff auf deren Datenbank und Ereignisverwaltung gestattet. Die API muß über ähnliche Möglichkeiten verfügen, wie sie auch bei der Manager-Agent-Beziehung im SNMP-Protokoll auftreten. An die Stelle des Managers tritt die Plattform und an die des Agenten das Modul. Ausschlaggebend sind hier

drei Situationen, die weitgehend den zuvor beschriebenen Aufgaben des Schnittstellen-Moduls entsprechen:

1. Manager vergibt Aufträge (bei SNMP z.B. das Setzen von MIB-Objekten)
2. Manager holt Informationen (bei SNMP z.B. das Auslesen einer Geräte-MIB)
3. Agent meldet ein bestimmtes Ereignis (bei SNMP die Trap-Steuerung)

Um das Modul mit verschiedenen Plattformen einsetzen zu können, müssen jeweils die benötigten APIs zur Verfügung stehen. Eine weitere Möglichkeit wäre die Implementierung eines echten Proxy-Agenten, der über das standardisierte SNMP-Protokoll anzusprechen ist.

Teilbereich B:

- **API-Meta-Sprache-/Meta-Sprache-API-Übersetzer:**
  Die möglichen Grundkommandos, die von der Plattform abgesetzt werden, sind im allgemeinen standardisiert. Wegen dieser Standardisierung ist es möglich, eine Meta-Sprache als Übergang innerhalb des Moduls zu verwenden, um die Teilbereiche unabhängiger und damit auch flexibler zu gestalten. Die Hauptbefehle der Sprache befassen sich mit dem Anfordern und Setzen von Daten und zusätzlichen Aktionen zur Ereignisüberwachung. Alle Befehle arbeiten mit einem Parametersatz. Die Kommandos der Plattform oder die Meldungen von seiten des X.25-Netzes werden in ein allgemeines Befehlsschema mit zugehörigem Parametersatz gepackt und vom Modul in dieser Form weiterverarbeitet. Die Umsetzung in eine Meta-Sprache gestattet insbesondere in der CLI-Variante des Teilbereichs C eine leichtere Portierbarkeit des Moduls auf verschiedene Plattform-APIs und X.25-Management-Werkzeuge.

- **Meta-Sprache-X.25-API/X.25-API-Meta-Sprache-Übersetzer (API-Variante):**
  Die beiden Komponenten übersetzen Meta-Kommandos in die entsprechenden API-Befehle des X.25-Management-Werkzeugs bzw. wandeln dessen Antworten und Meldungen in Meta-Sprachelemente um. Bei den Übersetzungsvorgängen ist zu beachten, daß Befehle, die eventuell von der Plattform-API direkt auf Meta-Ebene umgesetzt werden können, bei der nächsten Umsetzung in X.25-Werkzeug-Kommandos in eine nicht mehr einteilige Befehlssequenz verwandelt werden.

  Beispiel:

Plattform-API:	get_data(addrx.121, port3, data_element)	
Meta-Sprache:	Befehl:	REQUEST_DATA
	Variable:	X.121
		PORT
		DATA-ELEMENT
X.25-API:	remote control X.121	
	setport PORT	
	get_port_table	

- **Meta-Sprache-X.25-CLI-Übersetzer (CLI-Variante):**
  Erfolgt die Anbindung an das X.25-Management-Werkzeug über eine textuelle

Schnittstelle, so müssen die Meta-Sprache-Befehle in entsprechende Textkommandos umgesetzt werden. Für den Übersetzungsvorgang ist außer der Kenntnis des Wortschatzes und der Syntax des X.25-CLI der genaue Aufbau bzw. die Adressierung der im X.25-Management-Werkzeug verwalteten Tabellen notwendig, um an die relevanten Informationen zu gelangen, die den Datenelementen der X.25-Objekte entsprechen. Diese Vorschriften können in einer Konfigurationsdatei gespeichert werden. Auf diese Weise ist es möglich, nur durch das Austauschen von Konfigurationsdateien verschiedenartige X.25-Netze an das Modul zu binden.

- **X.25-Meta-Sprache-Übersetzer / Parser (CLI-Variante):**
  Alle textuellen Ausgaben des X.25-Management-Werkzeugs müssen vor der Weiterverarbeitung durch eine Parser-Einheit geschickt werden, um die Art der Nachricht und deren Inhalt erfassen zu können. Der Parser benötigt hierzu Schablonen mit genauen Formatanweisungen. Sie enthalten die Regeln für die Auswertung der Nachricht und beschreiben, an welchen Stellen Daten stehen und welche Bedeutung sie haben. Nach dem Weg durch die Parser-Einheit werden die ausgewerteten Daten in einen Meta-Befehl umgesetzt. Sowohl die Parser-Schablonen als auch die Vorschriften für den Umsetzungsvorgang können in einer separaten Konfigurationsdatei untergebracht werden.

Teilbereich C:

- **API:**
  Ähnlich wie bei der Verbindungsstelle zur Plattform muß auch die API des X.25-Management-Werkzeugs bestimmte Voraussetzungen erfüllen. Es liegt wieder eine Manager-Agent-Beziehung vor, mit dem Unterschied, daß diesmal das Modul an die Stelle des Managers tritt. Es werden Routinen zur Kommandoübergabe, Datenabfrage und Nachrichtenbehandlung benötigt.

- **CLI:**
  Bei dieser Variante werden alle Ein- und Ausgaben des X.25-Management-Werkzeugs, die ansonsten von einem menschlichen Bediener direkt an einem CLI bearbeitet werden würden, an das Schnittstellen-Modul umgeleitet. Hierzu müssen entsprechende Mechanismen vom CLI zur Verfügung stehen.

6 Umsetzung auf die Plattform SPECTRUM

Für die Umsetzung des X.25-Management-Modells auf die Plattform SPECTRUM wird in diesem Kapitel vorausgesetzt, daß ein geeigneter SNMP-Proxy-Agent zur Importierung der Netzdaten zur Verfügung steht. Desweiteren ist es notwendig, daß die eingesetzten Netzkomponenten alle notwendigen MIBs, die von den Objektklassen des Management-Moduls benötigt werden, bereitstellen.

In den folgenden Unterkapiteln werden nach einer kurzen Vorstellung der objektorientierten Elemente von SPECTRUM die notwendigen Schritte zur Realisierung des X.25-Management-Moduls und die dazu erforderlichen Hilfsmittel behandelt.

6.1 Objektorientierte Elemente der Plattform

Die Wissensbasis von SPECTRUM beruht auf der Definition von model types, relations und rules [SPEC_CG].

Ein model type ist eine Beschreibung einer Elementgruppe, die eine spezifische, abgekapselte Kombination von Attributen und Funktionen aufweist. Sie entspricht weitgehend einer Objektklasse im Sinne des object oriented modeling. Alle model types sind in der Model Type Hierarchy [SPEC_KB] erfaßt, die durch Vererbung von einfachen zu immer komplexeren und spezifischeren Typen aufgebaut wird. Ein model ist eine Instantiierung eines model type und kommt einem Objekt gleich. Die Instanzen eines model type verfügen über die gleichen Datenelemente und Funktionen.

Zum Aufbau von Beziehungen zwischen models werden verschiedene relations verwendet. Eine relation repräsentiert in SPECTRUM einen möglichen Sinnzusammenhang zwischen model types (z.B. relation „Contains). Es sind verbale Ausdrücke, mit denen Beziehungen zwischen models erstellt werden können. Durch Verknüpfung zweier models mit einer bestimmten relation wird eine association gebildet. Dieser Vorgang ist gleichbedeutend mit einer Instantiierung dieser relation.

Beispiel:

models:	Building A, Room B		
relation:	*contains*		
association:	Building A	*contains*	Room B

Um nur sinnvolle associations bilden zu können, werden für die einzelnen relations Vorschriften, sogenannte rules, definiert, die alle möglichen Sinnzusammenhänge zwischen model types festlegen. Eine association, die keiner rule entspricht, kann nicht instantiiert werden.

Beispiel:

rule für die relation *contains*:

model type Building	*contains*	model type Room

Das Verhalten eines model type wird durch intelligente Mechanismen beschrieben, die sich aus einer Sammlung von inference handlers zusammensetzen. Sie bestimmen dessen Reaktion auf Änderungen in seiner Umgebung. Es gibt unterschiedliche Einflußfaktoren, die eine Aktivierung eines inference handler verursachen können:

- Änderung eines Attributwertes eines model des jeweiligen model type.
- Neuanlage, Zerstörung oder Aktivierung eines model.
- Ereignisse oder Alarme, die in der Management-Umgebung von SPECTRUM auftreten.
- sonstige Veränderungen in der Netzlandschaft (z.B. Relationsänderungen).

Rules und inference handlers werden bei Neuableitung eines model type vererbt. Eine Darstellung der Zusammenhänge zwischen model type und inference handler zeigt Abbildung 6-1.

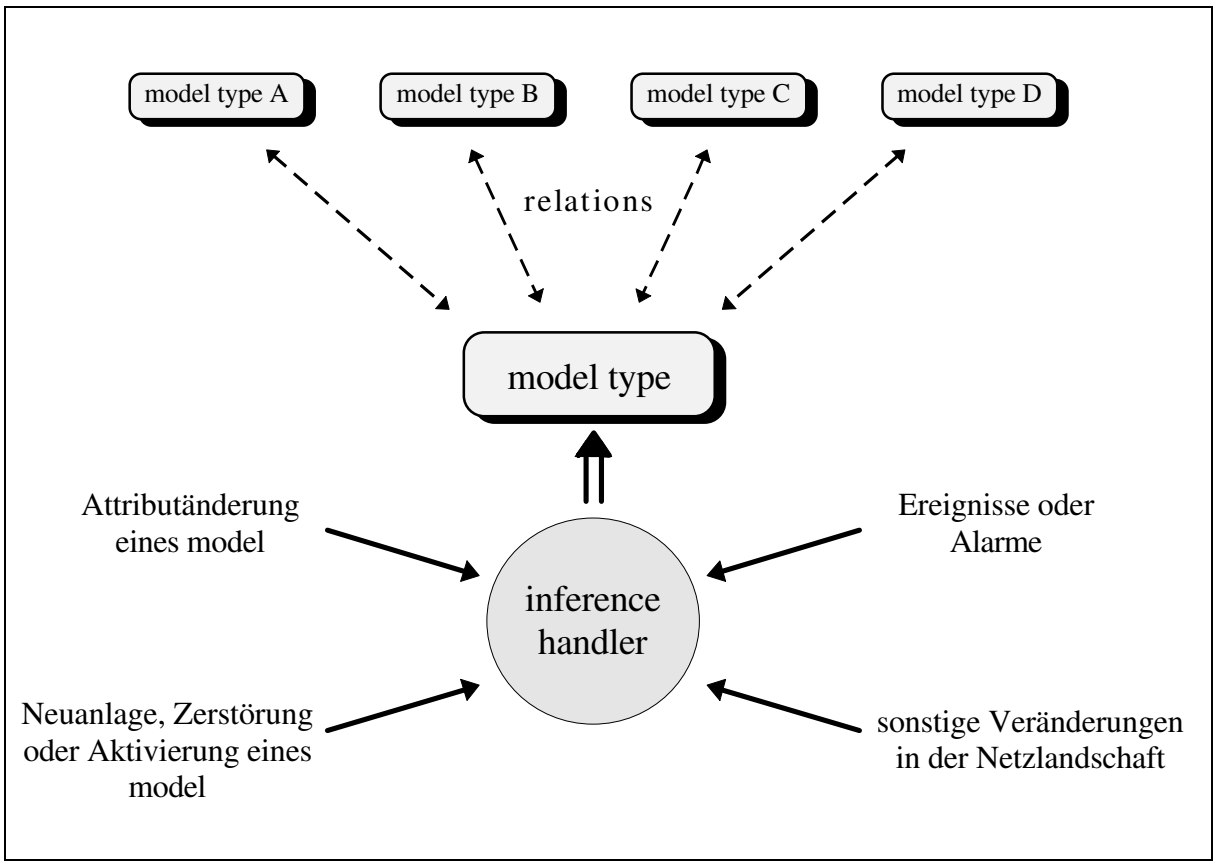


Abbildung 6-1 model type und inference handler.

Die Architektur der Wissensgrundlage von SPECTRUM entspricht den Anforderungen des object oriented modeling und eignet sich für eine Umsetzung des X.25-Management-Moduls bzw. des gesamten X.25-Management-Modells.

6.2 Realisierung des X.25-Management-Moduls

Durch die Realisierung des Moduls auf SNMP-Basis können bereits vorhandene Management-Mechanismen der Plattform eingesetzt werden. Insbesondere bei der Darstellung und automatischen Abfrage der zu managenden Objekte verringert sich der Aufwand für die Umsetzung. Dennoch muß in manchen Bereichen, z.B. bei bestimmten Objektklassen und deren Darstellungsweise, eine völlig neuer Weg bei der Implementierung gewählt werden, da für derartige Konzepte noch keine Vorlagen existieren.

Im folgenden werden die benötigten Werkzeuge zur Realisierung des Moduls aufgeführt und Hinweise zur Umsetzung der Objektklassen und Relationen gegeben. Für das Verständnis dieser Kapitel ist die Kenntnis der Model Type Hierarchy und der verschiedenen Relationsarten von SPECTRUM empfehlenswert [SPEC_KB].

6.2.1 Benötigte Werkzeuge

Zur Erweiterung der Wissensgrundlage von SPECTRUM ist die Einführung von neuen model types, relations und rules notwendig. Hierfür steht als graphisches Hilfsmittel der Model Type Editor [SPEC_MTE] zur Verfügung. Mit ihm können neue model types und relations bzw. rules definiert werden.

Der Model Type Editor orientiert sich bei der Neueinführung eines type an der bestehenden Model Type Hierarchy. Es muß deshalb zuvor bekannt sein, von welchen Punkten der neue type abgeleitet werden soll. Nach der Anlage können die geerbten Attribute angepaßt oder neue hinzugefügt werden. Der Model Type Editor bietet die Möglichkeit, MIB-Definitionen, die in ASN.1 (Abstract Syntax Notation One) vorliegen, direkt als Attribute für einen model type einzulesen. Alle notwendigen Schritte zur Schaffung von Relationen und Relationsregeln können ebenfalls mit dem Model Type Editor vollzogen werden.

Um die verschiedenen model types mit Funktionalität zu versehen, müssen inference handlers entworfen und implementiert werden. Hierfür wird die inference handler API [SPEC_IH] benötigt. Es handelt sich hierbei um eine C++ Klassenbibliothek, mit der die einzelnen handlers in die SPECTRUM-Umgebung integriert werden können.

6.2.2 Umsetzung der Relationen

SPECTRUM verfügt in der Grundausstattung über eine Gruppe von definierten Relationen. Eine Übersicht mit Erklärungen und Einsatzgebieten findet sich [SPEC_KB]. Die Relationsarten, die im X.25-Management-Modul benötigt werden, können z.T. direkt auf entsprechende Relationen von SPECTRUM umgesetzt werden.

Umsetzungshinweise zu den einzelnen Relationen:

- „hat Funktionalität“:
  Die „Manages“ Relation stellt eine Beziehung zwischen einem device model und einem application model auf. Zu letzterem zählen auch die umgesetzten Objektklassen der

Datenerfassungstypen. Sie entspricht somit der „hat Funktionalität“ Relation und kann an ihrer Stelle verwendet werden.

- „ist verbunden mit“:
  Die Relation „Connects_to“ drückt eine physikalische Verbindung zwischen zwei models aus. Sie bietet somit die gleichen Eigenschaften wie die „ist verbunden mit“ Beziehung.

- „enthält“:
  SPECTRUM unterscheidet beim Begriff des Enthaltenseins zwischen der Beziehung von einer Netzlandschaft zu ihren Geräten und der Beziehung von einem Gerät zu seinen Komponenten. Die Relation „Collects“ tritt in Verbindung mit topology models auf, die Informationen über ihre zugehörigen models sammeln. Mit „Contains“ hingegen wird die Zugehörigkeit einer Komponente zu einem Gerät oder von Ports zu einem Modul ausgedrückt.

  Im X.25-Management-Modul werden diese Unterteilungen hingegen nicht vorgenommen. Um mit den bereits bestehenden Modellierungsmodellen in SPECTRUM kompatibel zu bleiben, müssen entsprechende Unterscheidungen auch bei den umgesetzten Relationsregeln für die neuen model types beachtet werden.

- „bildet Protokollstack mit“:
  Diese Beziehung ist in SPECTRUM nicht definiert. Sie muß als neue Relation eingeführt werden.

- „kontrolliert“:
  Im X.25-Management-Modul wird diese Relation zur Verbindung der Link-Objekte mit Datenerfassungsobjekten eingesetzt. Da es in SPECTRUM keine Entsprechung für diese Beziehung gibt, muß eine neue Relation eingeführt werden.

6.2.3 Umsetzung der Objektklassen in model types

Da vorausgesetzt wird, daß die Netzdaten mit SNMP abgefragt werden können, bietet sich für die Übertragung der Objektklassen die Verwendung eines generischen SNMP model type an. In [SPEC_MM] werden vier ableitbare generic SNMP model types eingeführt:

- GnSNMPDev:
  Für die Erstellung von einfachen device models.

- GnSNMPMibDerPt:
  Dieser model type dient als Ableitungspunkt von types für neu zu importierende MIBs.

- GnSNMPAppDerPt:
  Bevor mit einem MIB model type gearbeitet werden kann, muß er zuvor in einen application model type eingebettet werden. Er übernimmt u.a. Aufgaben der Darstellung in verschiedenen SPECTRUM-views.

- GnSNMPSubDerPt:
  Dieser Ableitungspunkt wird für kleinere application models eingesetzt, die über eine

Relation einem GnSNMPAppDerPt untergeordnet sind. Unterschiede zeigen sich auch in den SPECTRUM-views.

Zusätzlich zu diesen model types stehen weitere generische types zur Verfügung, auf die gesondert hingewiesen wird. Die Objektklassen der verschiedenen Gruppen können in der Regel von einem bestehenden model type abgeleitet werden. Im folgenden werden die notwendigen Ableitungspunkte gruppenweise besprochen:

- Gruppe der Datenerfassungstypen:
  Der erste Schritt ist die Definition von MIB model types, die GnSNMPMibDerPt als Ableitungspunkt verwenden. Für jede Objektklasse, die auf einer MIB-Beschreibung basiert, wird ein eigener model type abgeleitet. Durch Importierung der MIB-Beschreibung in den model type werden die Datenelemente automatisch angelegt. Für die Objektklassen, denen keine standardisierte Beschreibung zugrunde liegt, muß auf herstellerspezifische MIB-Werte zurückgegriffen werden. Die Möglichkeiten richten sich nach dem MIB-Angebot des Proxy-Agenten.

  Im zweiten Schritt werden aus den zuvor definierten MIB model types einzelne application types erzeugt. Hierfür werden GnSNMPAppDerPt und GnSNMPSubDerPt als Ableitungspunkte eingesetzt. Abbildung 6-2(a) zeigt den Ableitungsvorgang für den LAPB model type. Application models werden durch die „Manages“ Relation mit einem device model verbunden. Eine Ausnahme zeigt sich bei der Umsetzung der Interface-Objektklasse, da sie als späterer Ableitungspunkt der Port-Objektklasse dient. Für sie ist es nicht erforderlich, einen application model type anzulegen.

- Gruppe der Komponenten:
  Für die Port-Objektklasse dient der in der Gruppe der Datenerfassungstypen eingeführte Interface model type als Ableitungspunkt. Die Module-Objektklasse setzt auf den generischen model type GnModule auf.

- Gruppe der X.25-Geräte:
  Das X.25-Device wird von GnSNMPDev abgeleitet. Alle GnSNMPDev models können durch die „Collects“ Relation mit einem topology model verknüpft werden.

- Gruppe der Verbindungstypen:
  Da derartige model types in SPECTRUM noch nicht vorhanden sind, muß in der Model Type Hierarchy eine neue Ableitung von EntityTypes geschaffen werden (siehe [SPEC_KB]).

- Gruppe der Netzeinheiten:
  Die Objektklassen dieser Gruppe gehören in den Bereich der topology model types. Der Ableitungspunkt für die X.25-topology types ist der model type WAN.

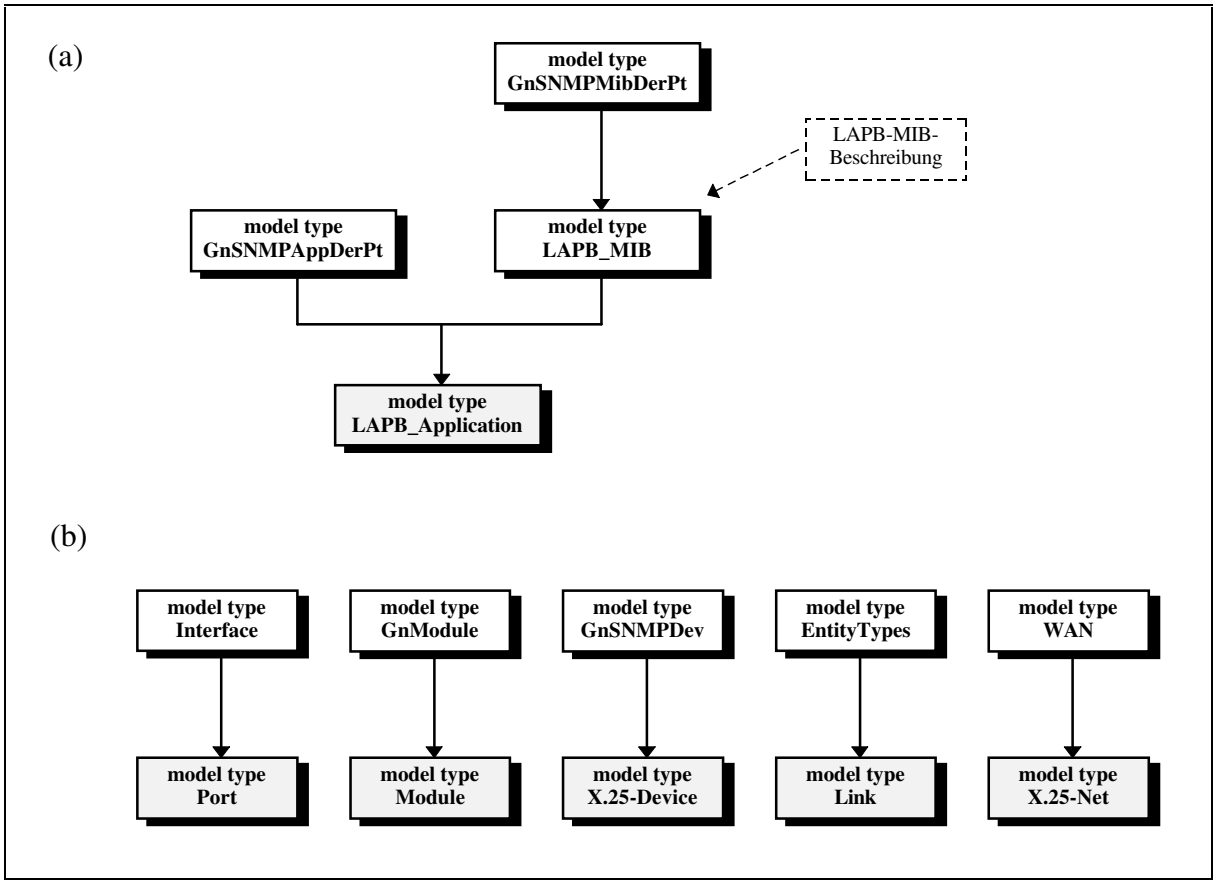


Abbildung 6-2 Ableitungspunkte der neuen model types: (a) Ableitung eines Datenerfassungstypen; (b) Hauptvertreter der verschiedenen Gruppen.

In Abbildung 6-2(b) sind die Hauptvertreter der einzelnen Objektklassengruppen mit ihren Ableitungspunkten aufgeführt. Nach der Definition aller notwendigen model types und der Einführung der neuen Relationen müssen rules aufgestellt werden. Sie entsprechen den Relationsregeln, die im X.25-Management-Modul festgelegt wurden.

6.3 Realisierung von Applikationen

Applikationen, die auf die Wissensbasis von SPECTRUM zugreifen, können auf zwei Arten realisiert werden:

a) Erweiterung des SpectroGRAPH[1] um zusätzliche views

b) Entwicklung von eigenständigen Applikationen

Für die Neuanlage eines view wird der GIB-Editor (Generic Information Block) eingesetzt [SPEC_GIB]. Mit ihm können generische views kreiert und angepaßt werden. Die views greifen in der Regel auf Attribute der models zu und beziehen sich auf bestimmte Relationsarten. Das Einsatzgebiet der Applikationen unter SpectroGRAPH ist deshalb sehr eingeschränkt.

[1] Visualisierungs-Tool der SPECTRUM-Plattform

Die zweite Möglichkeit besteht in der Entwicklung von eigenständigen Applikationen, die mit APIs auf die unter SPECTRUM verwaltete Netzlandschaft zugreifen. Der Vorteil dieser Vorgehensweise liegt in der gezielten Auswahl der benötigten Management-Informationen, die mit einem für die Applikation spezifischen Algorithmus verarbeitet werden können. Auf diese Weise können Applikationen erstellt werden, wie sie z.B. in Kapitel 4.2 beschrieben wurden. Mit entsprechenden Algorithmen können Darstellungen von Links, Protokollhierarchien oder virtuellen Verbindungen realisiert werden.

7 Ausblicke

In dieser Arbeit wurde ein Konzept für ein integriertes Management-Modell eines X.25-Netzes entworfen. Hierfür wurden sowohl netzinterne, als auch netzübergreifende Management-Aspekte untersucht.

In Hinblick auf eine End-to-End-Sichtweise wurde in Kapitel 3.1.2 eine Verbindung durch eine Verkettung von Koppelelementen einzelner Netzabschnitte beschrieben, wobei hier auf den X.25-Bereich eingegangen wurde. Ein Themengebiet für Folgearbeiten liegt in der Untersuchung einer geeigneten Netzbeschreibungssprache für den LAN-Bereich, die für eine vollständige Darstellung einer End-to-End-Beziehung notwendig ist.

Ansatzpunkte hierfür werden in [CITI] vorgestellt. Es werden dort erste Schritte in der Entwicklung einer allgemeinen Netzbeschreibungssprache unternommen. Mit einem Standardformat soll es möglich sein, Netztopologien exakt darzustellen und Management-Werkzeugen bzw. Plattformen eine standardisierte Informationsquelle zu bieten.

Mit einer derartigen Beschreibung können auch im WAN-Bereich einfache Mechanismen zur automatischen Übernahme von Netztopologien erstellt werden. Auf diese Weise würde sich eine Möglichkeit bieten, Topologiedaten eines X.25-Netzes zu importieren, die zur Generierung der entsprechenden Elemente des X.25-Management-Modul auf einer Plattform benötigt werden.

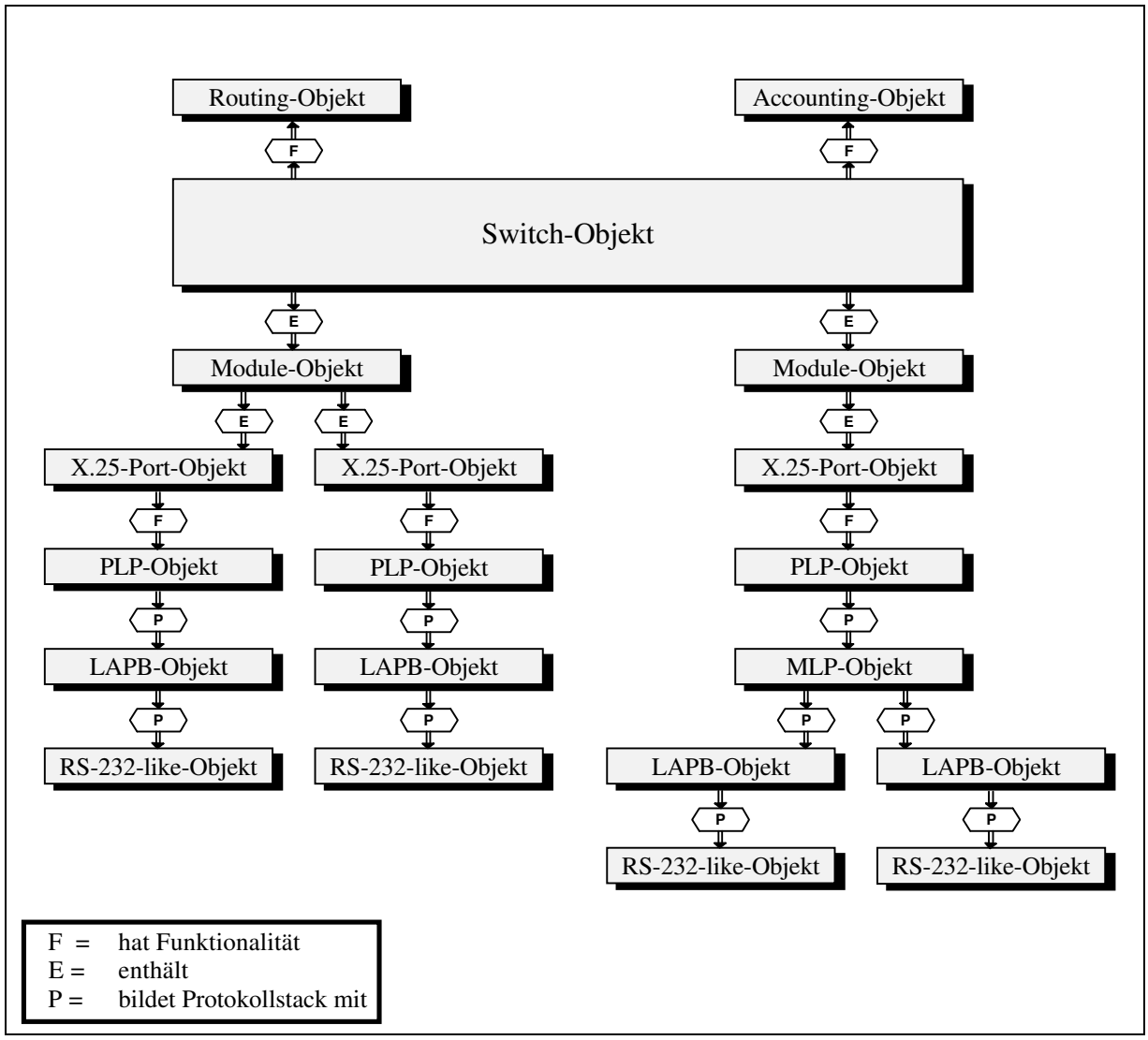
Anhang A Modellierungsbeispiele

Als Beispiel für die Anwendung des Modellierungsmodells wird hier ein Switch-Objekt und eine Verbindung von zwei X.25-Ports mit den in Kapitel 4 definierten Objektklassen und Relationen beschrieben.

Switch

Der Switch hat folgende Module:

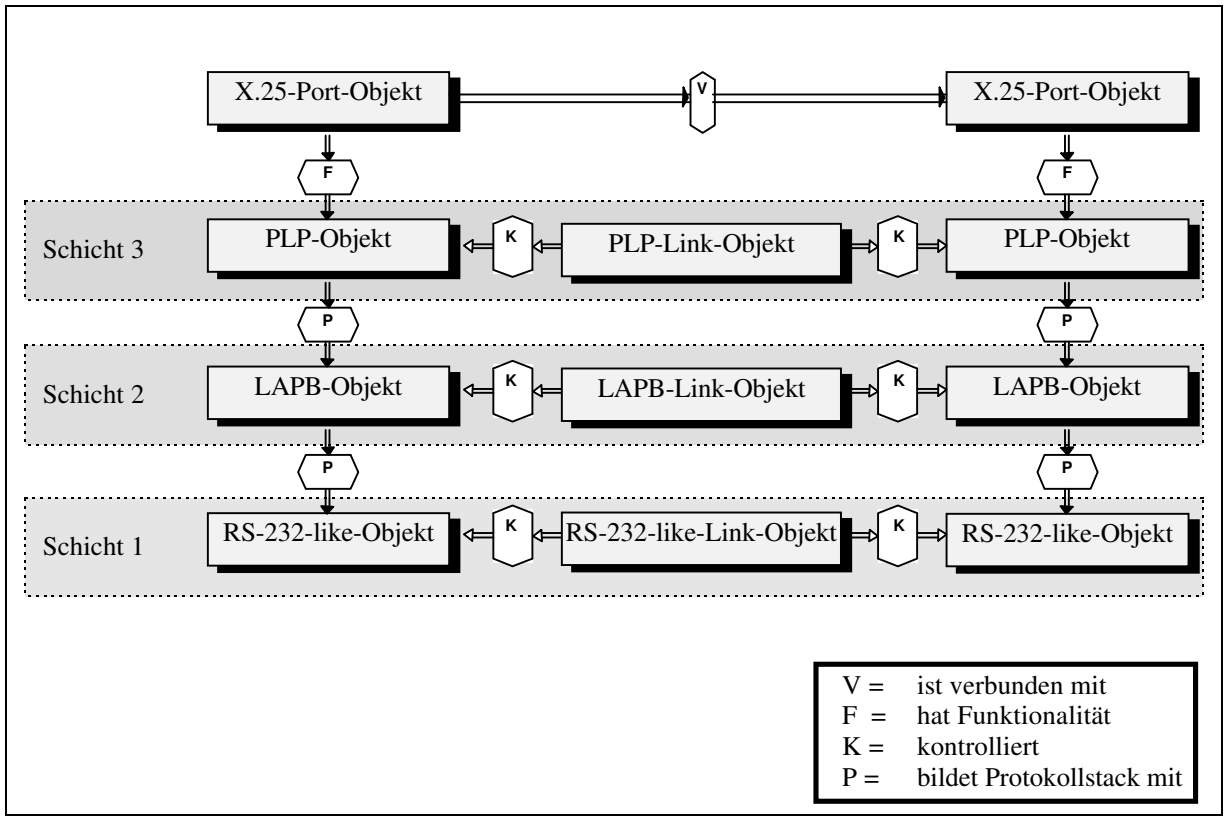
- X.25-Modul mit zwei X.25-Ports.
- X.25-Modul (MLP) mit einem X.25-Port, der auf zwei Links basiert.



Verbindung

Es wird eine Verbindung zwischen zwei X.25-Ports mit allen Link-Objekten der einzelnen Protokollschichten gezeigt.

Bei dieser typischen X.25-Verbindung kommen alle in der Verbindungsgruppe definierten Link-Objektarten zum Einsatz. Zusätzlich zu den Link-Objekten zwischen den Datenerfassungsobjekten sind die Port-Objekte selbst mit einer Relation verknüpft.



Anhang B Relationsregeln

Relationsregeln des X.25-Management-Moduls:

Objektklasse links	Relationsausdruck	Intervall	Objektklasse rechts
LAPB	*bildet Protokollstack mit*	[1]	RS-232-like
MLP	*bildet Protokollstack mit*	[2-]	LAPB
PLP	*bildet Protokollstack mit*	[1]	LAPB
PLP	*bildet Protokollstack mit*	[1]	MLP
RS-232-like-Port	*hat Funktionalität*	[1]	RS-232-like
RS-232-like-Port	*ist verbunden mit*	[1]	RS-232-like-Port
X.25-Port	*hat Funktionalität*	[1]	PLP
X.25-Port	*ist verbunden mit*	[1]	X.25-Port
X.75-Port	*hat Funktionalität*	[1]	PLP
X.75-Port	*ist verbunden mit*	[1]	X.75-Port
Module	*enthält*	[1-]	Port
X.25-Device	*enthält*	[1-]	Module
Switch	*hat Funktionalität*	[1]	Routing
Switch	*hat Funktionalität*	[0-1]	Accounting
IWU	*hat Funktionalität*	[1]	Peer
STE	*enthält*	[1]	X.25-Port
STE	*enthält*	[1]	X.75-Port
PAD	*hat Funktionalität*	[1]	Routing
RS-232-like-Link	*kontrolliert*	[2]	RS-232-like
LAPB-Link	*kontrolliert*	[2]	LAPB
PLP-Link	*kontrolliert*	[2]	PLP
X.25-Net	*enthält*	[0-]	X.25-Subnet
X.25-Net	*enthält*	[0-]	X.25-Device
X.25-Net	*enthält*	[0-]	Link
X.25-Subnet	*enthält*	[0-]	X.25-Device
X.25-Subnet	*enthält*	[0-]	Link

Literaturverzeichnis

[ABECK] Abeck, Sebastian: „Integrationstechniken im Netzmanagement“. Münchner Netzmanagement Team, TU München - Institut für Informatik.

[BAPAT] Bapat, Subodh: „Object-Oriented Networks - Models for Architecture, Operations, and Management“. PTR Prentice Hall, Englewood Cliffs, New Jersey 07632, 1994.

[CCITT_BB2] „BLUE BOOK - Volume VIII -Fascicle VIII.2: Data Communication Networks Transmission, Signalling and Switching, Network Aspects, Maintenance and Administrative Arrangements - Recommendations X.1-X.32“. CCITT Melbourne, November 1988.

[CCITT_BB3] „BLUE BOOK - Volume VIII -Fascicle VIII.3: Data Communication Networks Transmission, Signalling and Switching, Network Aspects, Maintenance and Administrative Arrangements - Recommendations X.40-X.181“. CCITT Melbourne, November 1988.

[CITI] cisco Systems, Inc.:„Common Internetworking Topology Initiative“.

[DATUS] „DATUS DATENPAKETVERMITTLUNGSANLAGEN DPV 5810 und MDPV 5810 - Operator Manual -“. DATUS elektronische Informationssysteme GmbH, Würselen, Ausgabe 09.89-2.

[HALSALL] Halsall, Fred: „Data Communications, Computer Networks and Open Systems“. Addison-Wesley, 1992.

[HEGER] Hegering, H.-G.; Abeck S.: „Integriertes Netz- und Systemmanagement“. Addison-Wesley, 1993.

[ICS] Anastos, Sotiris; Prusseit, Jürgen: „X.25-Integration Spectrum Management Module (Version 1.0) Technical Specification“. Document No.: 16.434.426.0. ICS - Intelligent Communication GmbH, München, 20.8.94.

[ISO7498-4] ISO, Information Processing Systems - Open Systems Interconnection - Basic Reference Model Part4: Management Framework, 1989.

[KRISHNAN] Estrin, Deborah; Zhang, Lixia: „Design Consideration for Usage Accounting and Feedback in Internetworks“. International Symposium on Integrated Network Management II, Krishnan & Zimmer (Editors), Elsevier Science Publishers B.V. (North-Holland), 1991.

[NETCOM_RG] „NMS3000 Reference Guide Volume 1 User Interface & Core Facilities“. Netcomm Limited, 1993.

[NETCOM_IG1] „Switch/PAD Installation Manager's Guide Volume 1 - Version 3.2". Netcomm Limited, 1992.

[NETCOM_IG2] „Switch/PAD Installation Manager's Guide Volume 2 - Version 3.2". Netcomm Limited, 1992.

[RFC1317] Stewart, B.: „Definitions of Managed Objects for RS-232-like Hardware Devices", RFC 1317. Xyplex, Inc., April 1992.

[RFC1213] McCloghrie, K.; Rose, M.: „Management Information Base for Network Management of TCP/IP-based internets: MIB-II", STD 17, RFC 1213. Performance Systems International, March 1991.

[RFC877] Korb, J.: „A Standard for the Transmission of IP Datagrams Over Public Data Networks", RFC 877. Purdue University, September 1983.

[RFC1356] Malis, A.; Robinson, D.; Ullmann, R.: „Multiprotocol Interconnect on X.25 and ISDN in the Packet Mode", RFC 1356. BBN Communications, Computervision System Integration, Process Software Corporation, August 1992.

[RFC1461] Throop, D.: „SNMP MIB extension for Multiprotocol Interconnect over X.25", RFC 1461. Data General Corporation, November 1992.

[RFC1381] Throop, D.: „SNMP MIB extension for LAPB", RFC 1381. Data General Corporation, November 1992.

[RFC1382] Throop, D.: „SNMP MIB extension for the Packet Layer of X.25", RFC 1382. Data General Corporation, November 1992.

[ROSE] Rose, M. T.: „The Simple Book: An Introduction to Management of TCP/IP-based Internets". Prentice Hall, Englewood Cliffs, New Jersey, 1991.

[SCHWERD] Schwerdtner, Stefan: „Ein verursacher- und verbrauchsbezogenes Abrechnungsmodell". Münchner Netzmanagement Team, TU München - Institut für Informatik.

[SPEC_BEG] „SPECTRUM® Basic Extensions Guide". Cabletron Systems, Inc., Rochester, NH, USA, June 1993.

[SPEC_CG] „SPECTRUM® Concepts Guide". Cabletron Systems, Inc., Rochester, NH, USA, August 1994.

[SPEC_GIB] „SPECTRUM® GIB Editor Guide". Cabletron Systems, Inc., Rochester, NH, USA, June 1993.

[SPEC_IH] „SPECTRUM® Inference Handler Application Programming Interface Developer's Guide". Cabletron Systems, Inc., Rochester, NH, USA, January 1993.

[SPEC_KB] „SPECTRUM® Knowledge-Base Guide“.
Cabletron Systems, Inc., Rochester, NH, USA, August 94.

[SPEC_MM] „SPECTRUM® Management Module Guide for Generic SNMP“.
Cabletron Systems, Inc., Rochester, NH, USA, September 94.

[SPEC_MTE] „SPECTRUM® Model Type Editor Guide“.
Cabletron Systems, Inc., Rochester, NH, USA, June 93.

[SPEC_SG] „SPECTRUM® Administration Lecture/Lab Student Guide 3.0“.
Cabletron Systems, Inc., Rochester, NH, USA, June 1994.

[STÖTT] Stöttinger, Klaus: „X.25 Datenpaketvermittlung“. Datacom, 1991.

[THORPE] Thorpe, Nicolas M.; Ross, Derek: „X.25 Made Easy“.
Prentice Hall, 1992

[VALTA] Valta, Robert: „Entwicklung einer Methodik zur Beschreibung von offenen Rechnernetzen als Grundlage für integriertes betreiberorientiertes Netzmanagement“. Doktorarbeit, TU München, 1990.

[VARLEY] Varley, Barry: „User Administration and Accounting“.
Sema Group Consulting, London, UK.

Abkürzungen

API	Application programming interface
CCITT	International Telegraph and Telephone Consultative Committee
CLI	Command line interface
CRC	Cyclic redundancy check
DCC	Data country code
DCE	Data circuit terminating equipment
DNIC	Data network identification code
DTE	Data terminal equipment
FCS	Frame check sequence
HDLC	High-level data link control
IAB	Internet activities board
IP	Internet protocol
ITU	International Telecommunications Union
IWU	Interworking unit
LAN	Local area network
LAPB	Link access procedure balanced
MIB	Management information base
MLP	Multilink procedure
NN	National number
NTN	Network terminal number
PAD	Packet assembler / disassembler
PDU	Protocol data unit
PLP	Packet layer protocol
PSE	Packet-switching exchange
PVC	Permanent virtual circuit
QoS	Quality of service
RFC	Request for comments
SNMP	Simple network management protocol
SLP	Single link procedure
STE	Signalling terminal exchange
SVC	Switched virtual circuit
TCP	Transmission control protocol
WAN	Wide area network

Entwicklung integrierter Lösungskonzepte zu ausgewählten Fragestellungen des X.25-Managements

Thomas Dirsch