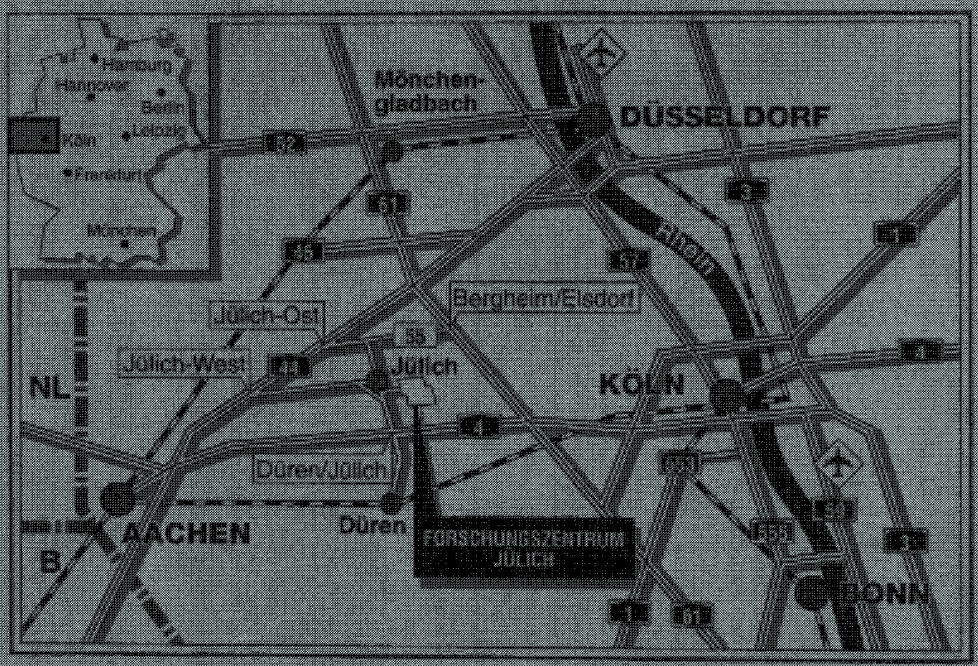
FORSCHUNGSZENTRUM JÜLICH GmbH
KFA

Zentralinstitut für Angewandte Mathematik

Einsatz von ATM
Möglichkeiten und heutige Grenzen

Adam Lukosek



Berichte des Forschungszentrums Jülich ; 3279
ISSN 0944-2952
Zentralinstitut für Angewandte Mathematik Jül-3279

Zu beziehen durch: Forschungszentrum Jülich GmbH · Zentralbibliothek
D-52425 Jülich · Bundesrepublik Deutschland
Telefon: 02461/61-6102 · Telefax: 02461/61-6103 · Telex: 833556-70 kfa d

Kurzfassung

Entscheidend für den Einsatz von ATM im lokalen Bereich ist die Bereitstellung von Konzepten, die einerseits die Weiterbenutzung bestehender Netze und Protokolle erlauben, andererseits aber im ATM-Umgebung auch die Nutzung der ATM-spezifischen Vorteile ermöglichen. Bei der Integration von ATM-Netzwerken in die herkömmlichen LAN-Umgebungen haben zur Zeit zwei Methoden eine praktische Bedeutung: *Classical*-IP der IETF (*Internet Engineering Task Force*) und die LAN-Emulation des ATM-Forums.

In der vorliegenden Arbeit werden beide Ansätze vorgestellt und analysiert, und es wird auf die Einschränkungen, die sie mit sich bringen, eingegangen. Dabei werden auch Weiterentwicklungen sowie zukünftige Konzepte vorgestellt, an denen bei der IETF und beim ATM-Forum aktuell gearbeitet wird, wie *Next Hop Resolution Protocol* (NHRP) oder *Multiprotocol Over ATM* (MPOA).

Außerdem wird anhand der im lokalen ATM-Testbett des ZAM durchgeführten Messungen das Verhalten von TCP(UDP)/IP über ATM untersucht.

Abstract

Concepts permitting further utilization of existing networks and protocols as well as taking advantage of new ATM features within the ATM environment are crucial for the deployment of ATM in local area networks. By now two methods gained practical significance in integrating ATM networks into traditional LAN environments: *Classical IP over ATM* of IETF (Internet Engineering Task Force) and *LAN Emulation* of the ATM Forum.

In this work both approaches are described and analyzed, and the restrictions they still encounter are considered. Proposed advancements as well as future concepts prepared by the IETF and the ATM Forum are presented, especially the *Next Hop Resolution Protocol* (NHRP) and *Multiprotocol Over ATM* (MPOA).

In addition, the behavior of TCP(UDP)/IP over ATM is explored based on experiments and measurements carried out at the local ATM testbed of ZAM.

Inhaltsverzeichnis

Kapitel 1

Einleitung

Jahrelang waren fast ausschließlich numerische und alphanumerische Informationen Gegenstand der Datenkommunikation. Die explosionsartige Zunahme der Leistungsfähigkeit moderner Prozessoren, der Preisverfall für Speichermedien sowie gravierende Fortschritte in Übertragungstechnik haben dazu geführt, daß zunehmend Informationen in Form von Graphik, Ton- und Bildsequenzen übertragen und gespeichert werden können.

Ein wichtiger Schritt der Integration unterschiedlicher Dienste erfolgte mit der Einführung des diensteintegrierenden digitalen Fernmeldenetzes (ISDN). Das ISDN basiert auf leitungsvermittelten 64-kbps-Kanälen, die für unterschiedliche Anwendungen genutzt werden können.

Das heutige Schmalband-ISDN ist aber noch kein wirklich universelles Netz. Die verfügbare Bandbreite ist für viele Anwendungen und auch für zukünftige Datenvolumina nicht ausreichend. Um multimediale Anwendungen verschiedenster Art sowie Übertragung von Sprache und Daten effizient betreiben zu können, ist die Weiterentwicklung des bestehenden Schmalband-ISDN zu einem universell nutzbaren Breitband-ISDN eine Notwendigkeit.

Als Vermittlungs- und Multiplextechnik für das Breitband-ISDN wurde von ITU-T (früher CCITT) im Jahre 1988 der *Asynchronous Transfer Mode* (ATM) festgelegt. ATM ist eine ganz neue Netztechnik, die allen bisherigen Netzwerktechnologien überlegen ist, und folgende wichtige Eigenschaften aufweist:

- ATM ist eine Switching-Technologie. Alle Geräte sind sternförmig an eine Vermittlungseinrichtung angeschlossen und müssen sich nicht – wie bei herkömmlichen *Shared-medium*-Systemen – die auf dem gemeinsam genutzten Medium verfügbare Bandbreite teilen.
- Die ATM-Dateneinheit ist eine 53 Bytes lange Zelle mit einer festen Struktur, was eine schnelle hardware-gesteuerte Bearbeitung erlaubt.
- ATM ist universell einsetzbar und stellt eine einheitliche Technik für verschiedene Dienste zur Verfügung.
- ATM verspricht Zeittransparenz und Skalierbarkeit.
- ATM erlaubt eine dynamische Zuordnung der Bandbreite, bis zum derzeit spezifizierten Maximum von 622 Mbps je Anschluß.

ATM gilt als die universelle Hochgeschwindigkeits-Netztechnologie der Zukunft. Die Standardisierung von ATM ist jedoch noch längst nicht abgeschlossen.

Obwohl die ATM-Technik ihren Ursprung im Bereich öffentlicher Netze hat, ist sie auch für den Einsatz im privaten und lokalen Bereich geeignet.

Die Verfügbarkeit internationaler Standards ist für öffentliche Netze von essentieller Bedeutung. Im lokalen Bereich wirken sich evtl. fehlende Standards nicht so gravierend aus, wenn statt dessen proprietäre Lösungen, wie sie von führenden Herstellern angeboten werden, eingesetzt werden.

Um den Einsatz von ATM im lokalen Bereich schnell voranzutreiben, gründete im Jahr 1991 eine Gruppe verschiedener Firmen das ATM-Forum. Das ATM-Forum ist kein Standardisierungsgremium in eigentlichem Sinn, sondern erarbeitet Spezifikationen, um die Interoperabilität zwischen ATM-Komponenten unterschiedlicher Anbieter sicherzustellen. Da sich die Hersteller an diese Spezifikationen halten, stellen sie eine Form von Industriestandards dar.

Das ATM-Forum hat schnell erkannt, daß für die Verbreitung von ATM die Bereitstellung von Konzepten entscheidend ist, die die Weiterbenutzung bestehender Netze, Protokolle und Dienste erlauben. Um die Integration von ATM-Netzen in herkömmliche LAN-Umgebungen zu erleichtern, erstellte das ATM-Forum eine Spezifikation, die es ermöglicht, LANs über ATM zu emulieren.

Neben dem ATM-Forum startete auch IETF (*Internet Engineering Task Force*) frühzeitig Aktivitäten, um ihrerseits Vorschläge zu erarbeiten, die eine Realisierung von IP-Netzen über ATM ermöglichen.

Das Ziel der vorliegenden Arbeit ist es, die heutigen Möglichkeiten für den Einsatz von ATM im lokalen Bereich, zu beschreiben. Die verwendeten Ansätze werden vorgestellt und analysiert. Dabei wird, auf die heute noch existierenden Einschränkungen eingegangen, und es werden mögliche Lösungen sowie zukünftige Konzepte vorgestellt.

Im Anschluß an diese Einleitung werden zunächst die Architektur, die Eigenschaften sowie die Konzepte der ATM-Technik beschrieben. In den zwei folgenden Kapiteln werden die beiden für den Einsatz von ATM im lokalen Bereich wichtigsten Ansätze vorgestellt. Zunächst wird in Kapitel 3 diskutiert, wie mit dem *Classical*-IP-Modell der IETF IP-Netze über ATM realisiert werden können. Dabei werden Aspekte wie die Verpackung der IP-Pakete, die Adreßauflösung und die Verbindungsverwaltung betrachtet. Kapitel 4 befaßt sich dann mit der LAN-*Emulation* des ATM-Forums und zeigt Techniken, die benutzt wurden, um ein ATM-Netz wie ein herkömmliches LAN aussehen zu lassen. Das Verhalten des heute meist benutzten Protokolls, TCP/IP, in ATM-Netzen wird in Kapitel 5 genauer untersucht. Dazu werden nach einer Erläuterung der Probleme mit TCP bei Hochgeschwindigkeitsnetzen, die im KFA-ATM-Testbett durchgeführten Messungen präsentiert, die kritischen Parameter herausgearbeitet und daraus Rückschlüsse auf optimale Konfigurationen abgeleitet. Im Anschluß daran wird kurz auf die Probleme in einem überlasteten ATM-Netz und auf mögliche Lösungen eingegangen. Während sich das folgende Kapitel 6 mit den Erweiterungen des *Classical*-IP-Modells, insbesondere NHRP (*Next Hop Resolution Protocol*) und MARS (*Multicast Address Resolution Server*), befaßt, beschäftigt sich Kapitel 7 mit dem Multiprotokoll-über-ATM-Problem und stellt die zwei verschiedenen, derzeit beim ATM-Forum diskutierten Lösungsvorschläge vor. In Kapitel 8 werden schließlich die heute verfügbaren Konzepte für den Einsatz von ATM mit ihren Möglichkeiten und Einschränkungen zusammengefaßt, und in einem Ausblick wird auf die erwarteten Erweiterungen dieser Konzepte sowie auf neue, zukünftige Ansätze und die daraus resultierenden Folgerungen eingegangen. Im Anhang wird der aktuelle Status der wichtigsten ATM-Standards und Spezifikationen angegeben.

Kapitel 2

Der Asynchrone Transfer Modus

2.1 ATM-Technik

Ein ATM-Netz besteht im allgemeinen aus den ATM-Vermittlungsknoten (*Switches*), die über die Breitbandübertragungswege (ATM-Netzelemente) verbunden sind und zusammen ein vermaschtes Netz bilden. Wie man in Bild 2.1 sieht, unterscheidet man in ATM-Netzen prinzipiell zwei Arten von Schnittstellen:

- UNI (*User Network Interface*): definiert den Anschluß eines Endgerätes an einen *Switch*
- NNI (*Network Node Interface*): Schnittstelle zwischen zwei ATM-Knoten

Ein ATM-Netz kann sowohl privat als auch öffentlich (B-ISDN) sein. Deshalb kommen die UNI und NNI in zwei Formen vor. Die Schnittstellen zwischen den öffentlichen ATM-Einrichtungen werden von ITU-T, die Schnittstellen im privaten Bereich dagegen vom ATM-Forum spezifiziert. Ein privater und ein öffentlicher ATM-*Switch* kommunizieren aber nicht über ein NNI sondern über ein Public-UNI, weil sie keine NNI-Informationen austauschen. Das Public-UNI

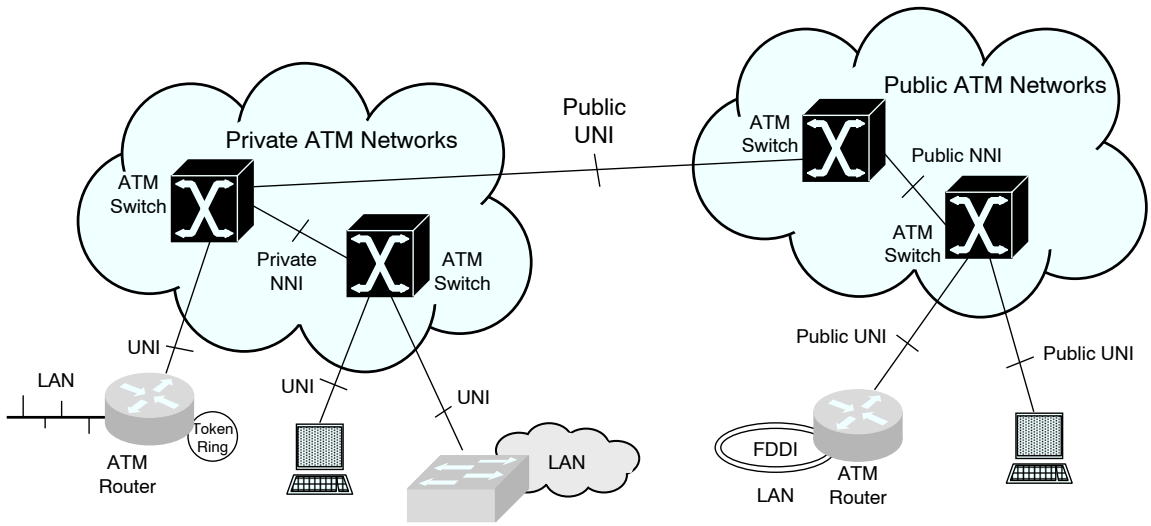


Bild 2.1: ATM Netz

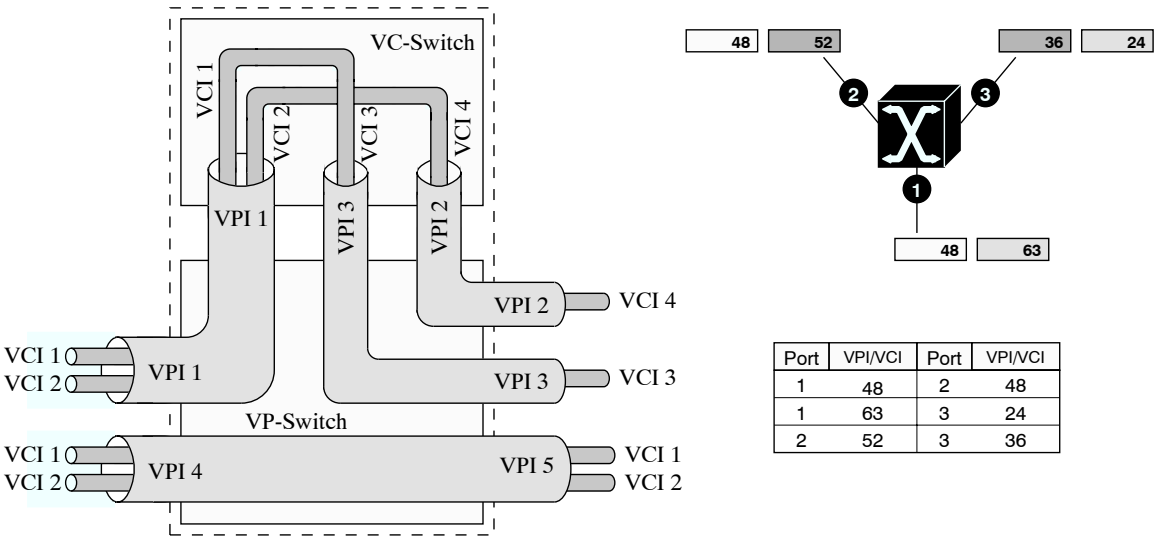


Bild 2.2: Vermittlung von ATM-Zellen

dient generell zur Anbindung eines privaten ATM-Systems – sei es ein Endgerät oder ein gesamtes privates ATM-Netz – an das öffentliche ATM-Netz.

Das ATM-Verfahren ist verbindungsorientiert. Das bedeutet, daß für den Austausch der Daten eine virtuelle Verbindung aufgebaut werden muß. Es gibt zwei Arten von virtuellen Verbindungen: die durch *Virtual Channel Identifier* (VCI) identifizierten virtuellen Kanäle und die durch *Virtual Path Identifier* (VPI) identifizierten virtuellen Pfade. Pfadverbindungen sind dabei Kanalverbindungen übergeordnet. In einem virtuellen Pfad werden mehrere virtuelle Kanäle zusammengefaßt. Die zugehörigen Zellen werden als einheitlicher Zellstrom behandelt und gemeinsam transportiert.

Die Vermittlung von ATM-Zellen erfolgt in einem ATM-*Switch* durch die Auswertung der VPI- und VCI-Werte. Die Arbeitsweise eines ATM-*Switches* ist relativ einfach: die an einem Port ankommenden ATM-Zellen mit bestimmten VPI- und VCI-Werten (die also einer virtuellen Verbindung zugeordnet sind) werden zu einem bestimmten Ausgangs-Port weitertransportiert und mit neuen (i.a. anderen) VPI- und VCI-Werten versehen, die die gleiche virtuelle Verbindung auf dem nächsten Übertragungsabschnitt identifizieren. Die entsprechenden Werte werden für jede Verbindung beim Verbindungsaufbau festgelegt und in lokalen Tabellen abgespeichert.

Grundsätzlich unterscheidet man zwischen reiner VP-Vermittlung (VP-*Switch*) und VC-Vermittlung (VC-*Switch*). In ATM-Pfadvermittlungen erfolgt die Vermittlung nur auf Basis der VPI-Werte, die VCI-Werte bleiben unverändert. In ATM-Kanalvermittlungen werden sowohl die VPI- als auch VCI-Werte ausgewertet und entsprechend umgesetzt. In ATM-Pfadvermittlungen, die daher weniger komplex aufgebaut sind, können mehrere Kanäle des gleichen Pfades schnell durch das Netz geschaltet werden.
Es gibt zwei Typen von ATM-Verbindungen:

- *Permanent Virtual Connections* (PVC)
  Die Festverbindungen werden vom Netzbetreiber manuell oder über Netzmanagementnachrichten dauerhaft eingerichtet.

- *Switched Virtual Connections* (SVC)
  Der Auf- und Abbau von Wählverbindungen erfolgt dynamisch mit Hilfe von Signalisierungsprotokollen.

Das ATM-Prinzip ermöglicht sowohl Punkt-zu-Punkt- als auch Punkt-zu-Mehrpunkt-Verbindungen. Dagegen werden von ATM *Multicasting* und *Broadcasting*, die man aus der *Shared-Medium*-LAN-Umgebung kennt, nicht unterstützt.

2.2 Das B-ISDN Protokoll-Referenzmodell

In modernen Kommunikationssystemen wird das Gesamtprotokoll mit seinen verschiedenen Protokollfunktionen hierarchisch gegliedert und in sogenannte Schichten unterteilt. Jede Schicht löst dabei die ihr zugeordneten Aufgaben und kommuniziert mit den Nachbarnschichten über definierte Schnittstellen. Die Funktionen der Schichten und das Zusammenspiel zwischen den einzelnen Schichten werden in einem Protokoll-Referenzmodell festgelegt. Als bekanntes Beispiel kann hier das ISO/OSI Referenzmodell dienen, das aus 7 Schichten besteht.

Das Protokoll-Referenzmodell des Breitband-ISDN zeigt dagegen eine von dem OSI-Referenzmodell etwas abweichende Architektur. Es wird ebenfalls in mehrere Schichten unterteilt und besteht in diesem Fall aus vier Schichten. Wie bereits vom Schmalband-ISDN bekannt ist, kennt das Protokollmodell außerdem noch drei Ebenen, die die vier Schichten miteinander verbinden: die Benutzerebene (*User Plane*), die Steuerebene (*Control Plane*) und die Managementebene (*Management Plane*). Die Unterteilung in die drei Ebenen spiegelt die Trennung der Nutzdatenübertragung von den Management- und Signalisierungsfunktionen wieder.

Bild 2.3 zeigt die in der Empfehlung I.321 von ITU-T [1] standardisierte Architektur des Breitband-ISDN Protokoll-Referenzmodells.

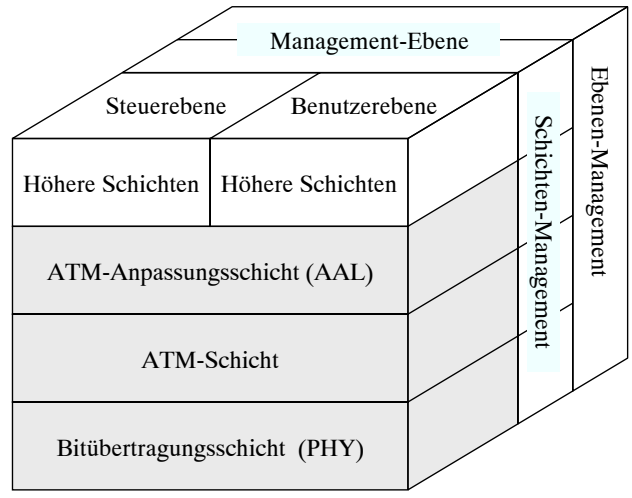


Bild 2.3: Protokoll-Referenzmodell des B-ISDN

Innerhalb der Benutzerebene werden alle Protokollfunktionen festgelegt, die für den Transfer von Benutzerdaten über alle Schnittstellen hinweg verantwortlich sind. Für den Auf- und Abbau sowie die Überwachung von Verbindungen ist die Steuerebene zuständig, die sämtliche Signalisierungsprotokolle enthält. Die Verwaltung der einzelnen Schichten erfolgt schließlich in der Managementebene, die sich über Steuerungs- und Benutzerebene erstreckt. Sie ist weiter in Funktionen für das Ebenen-Management und in Funktionen für das Schichten-Management unterteilt. Die Schichten-Management-Funktionen koordinieren die einzelnen Schichtenaufgaben und sind daher entsprechend strukturiert. Sie sind auch für die OAM-Informationsflüsse (*Operation And Maintenance*) zuständig. Dieser wird für Überwachungsaufgaben verwendet und mit Hilfe spezieller OAM-Zellen realisiert. Das

Bild 2.4: Funktionalität der Schichten des Protokoll-Referenzmodells

Ebenen-Management, das nicht weiter strukturiert ist, verwaltet die gemeinsamen Daten und koordiniert die Funktionen und Abläufe zwischen der Managementebene und den beiden anderen Ebenen.

In der ATM-Spezifikation werden vor allem die drei unteren Schichten definiert. Sie werden von der Benutzer- und der Steuerebene gemeinsam benutzt. Alle höheren Protokollschichten sind von der jeweiligen Anwendung abhängig.

In Bild 2.4 findet man eine Übersicht über die Funktionen der einzelnen Schichten sowie deren Unterteilung in weitere Teilschichten.

2.3 Die Bitübertragungsschicht (Physical Layer, PHY)

Die Aufgabe der Bitübertragungsschicht, die in der Empfehlung I.432 von ITU-T [5] spezifiziert wird, ist die transparente Übertragung der ATM-Zellen auf dem physikalischen Übertragungsmedium. Dazu gehört auch die Anpassung des Zellstroms an das jeweilige Übertragungssystem.

Aus diesem Grund ist die Bitübertragungsschicht weiter in zwei Teilschichten (*Sublayer*) unterteilt: in eine vom physikalischen Medium abhängige Teilschicht PM (*Physical Medium*) und in eine Übertragungsanpassungsteilschicht TC (*Transmission Convergence*), die die Eigenschaften des verwendeten Übertragungssystems vor der ATM-Schicht verbirgt.

Gegenstand der PM-Teilschicht ist vor allem die zum Einsatz kommende Übertragungshardware mit ihren physikalischen Eigenschaften. Außerdem werden Funktionen zur Leitungscodierung, die Mechanismen zur Bitsynchronisation und, sofern notwendig, die elektrisch-optische Umwandlung definiert. Die PM-Teilschicht stellt die eigentliche Bitübertragungsfunktion bereit und entspricht somit der Bitübertragungsschicht des OSI-Referenzmodells.

Innerhalb der zweiten Teilschicht, der Übertragungsanpassungteilschicht (TC), findet die Abbildung der ATM-Zellen in für das verwendete Übertragungssystem geeignete Dateneinheiten statt. Für die Übertragungsanpassung werden drei Methoden spezifiziert:

- Direkte zellorientierte Übertragung (*Cell Based*)
  Die ATM-Zellen werden direkt auf die Leitung geschickt. Es wird nur ein reiner ATM-Zellstrom übertragen. Es entsteht kein zusätzlicher *Overhead*. Im Gegensatz zu den beiden anderen Methoden, bei denen die Überwachungs- und Management-Informationen schon in den Übertragungsrahmen enthalten sind, muß hier nach jeweils 26 normalen Zellen eine OAM-Zelle (*Operation and Maintenance*) gesendet werden.

- Zellenanpassung auf Übertragungsrahmen eines bestehenden Übertragungssystems
  Die ATM-Zellen werden in die Übertragungsrahmen des jeweiligen Transportsystems eingebettet. Die ATM-Zellen werden dabei zwar bytesynchron mit den Bytes des Übertragungsrahmens, nicht jedoch rahmensynchron übertragen. Als Transportsystem kommen dabei sowohl die PDH- (*Plesiochrone Digitale Hierarchie*) als auch die SDH-Netzwerke (*Synchrone Digitale Hierarchie*) in Frage, wobei die letzten als der hauptsächliche Transportmechanismus für ATM-Zellen im Weitverkehrsbereich vorgesehen sind.

- Übertragung mittels PLCP-Rahmen (*Physical Layer Convergence Procedure*)
  Die ATM-Zellen werden in PLCP-Rahmen transportiert. PLCP findet in *Metropolitan Area Networks* Anwendung und wurde definiert, um die dort benutzten Datenpakete auf den PDH-Leitungen übertragen zu können.

Für den Einsatz von ATM sind verschiedene Geschwindigkeiten definiert worden. Die wichtigsten dabei sind 155,52 Mbps, 622 Mbps, 100 Mbps, 32 Mbps und 25 Mbps.

Die TC-Teilschicht hat darüberhinaus, unabhängig vom verwendeten Übertragungssystem, folgende funktionale Aufgaben:

- Entkopplung der Zellrate von der Bandbreite des Übertragungsmediums
  Zur Anpassung der Zellrate der ATM-Schicht auf die Bandbreite des Übertragungskanals werden *Idle*-Zellen eingefügt bzw. wieder entfernt. Diese *Idle*-Zellen werden mit einem speziellen *Header* versehen und werden auf Empfängerseite nicht an die ATM-Schicht weitergeleitet.

- Fehlerschutz des *Headers*
  Für den Fehlerschutz des *Headers* wird eine Prüfsumme errechnet und im HEC-Feld (*Header Error Control*) abgelegt.

- Zellsynchronisation (*Cell Delineation*)
  Es wird ein HEC-Zellsynchronisationsalgorithmmus benutzt, der mit Hilfe des HEC-Feldes und seiner Überprüfung versucht, den Beginn einer Zelle zu erkennen. Um diese Erkennung über das HEC-Muster zu verbessern, wird der Zellinhalt (*Payload*) der ATM-Zellen verschlüsselt (*scrambled*) übertragen. Die Codierung erfolgt jeweils mit einem entsprechenden Generatorpolynom.

- Sicherstellung des Zelltransports mittels OAM-Maßnahmen

2.4 Die ATM-Schicht

Die ATM-Schicht ist für das Vermitteln und Multiplexen von ATM-Zellen zuständig. Sie ist somit der Kern der ATM-Spezifikation. Die ATM-Schicht ist sowohl vom jeweiligen Dienst als auch vom physikalischen Übertragungsmedium unabhängig (!). Die Funktionen der ATM-Schicht sind ausschließlich dem *Header* zugeordnet. Das Informationsfeld wird auf der ATM-Schicht nicht bearbeitet.
Zu den wichtigsten Aufgaben der ATM-Schicht gehören:

- Erzeugung bzw. Entfernung des *Headers* (*Cell Header Generation/Extraction*)
  Es wird für die von der AAL-Schicht übergebenen Informationsfelder ein *Header* erzeugt, allerdings ohne HEC-Inhalt, der erst von der Bitübertragungsschicht berechnet wird.

- Umsetzen der Adreßfelder (*Cell VPI/VCI Translation*)
  Beim Vermitteln der ATM-Zellen in den Netzknoten werden die VCI- und VPI-Werte der ankommenden Zellen in neue VCIs bzw. VPIs übersetzt, die die gleiche virtuelle Verbindung auf dem nächsten Übertragungsabschnitt identifizieren.

- Multiplexen und Demultiplexen der ATM-Verbindungen
  Die zu übertragenden Zellen von verschiedenen Verbindungen werden in einen Zellstrom gemultiplext. Auf der Empfängerseite wird dieser Zellstrom wieder in die einzelnen Verbindungen aufgespalten.

- Generische Flußkontrolle (*Generic Flow Control, GFC*)
  Sie ist für Benutzer-Netz-Schnittstellen (UNI) mit einer Punkt-zu-Mehrpunkt Konfiguration vorgesehen, bei denen mehrere Teilnehmer ein gemeinsames Übertragungsmedium benutzen.

- Bearbeiten von Management-Funktionen
  Das Fehlermanagement und die Überwachung von Verbindungen werden durch den OAM-Informationsfluß (*Operation And Maintenance*) unterstützt.

Die ATM-Zelle
Nach verschiedenen Diskussionen und Vorschlägen wurde die Länge der ATM-Zelle auf 53 Bytes festgelegt, wovon 5 Bytes dem Zellkopf (*Header*) und 48 Bytes dem Informationsfeld (*Payload*) zugeordnet sind. Der Aufbau des *Headers* ist in der ITU-T Empfehlung I.361 [3] detailliert beschrieben. In Abhängigkeit davon, an welcher Netzschnittstelle sich die Zelle befindet, gibt es zwei Arten von ATM-Zellen: eine UNI-Zelle und eine NNI-Zelle. Sie unterscheiden sich durch das Flußkontrollfeld (GFC), das nur am UNI vorhanden ist. In Bild 2.5 findet man die Struktur des *Headers* in der UNI- und NNI-Zelle.

Generic Flow Control (GFC)
Dieses Feld ist für die Regelung von Zugriffs- und Übertragungsrechten in lokalen ATM-Netzen vorgesehen, bei denen mehrere Endgeräte an einer Schnittstelle angeschlossen sind.

Virtual Channel Identifier (VCI), Virtual Path Identifier (VPI)
Die Kennzeichnung für eine virtuelle Verbindung besteht aus einer Kennung für den virtuellen Kanal (*Virtual Channel Identifier, VCI*) und einer Kennung für den virtuellen

Pfad (*Virtual Path Identifier, VPI*). Die VCI- und VPI-Werte sind keine Adressen und nicht netzweit eindeutig.

Payload Type (PT)

Das PT-Feld dient zur Nutzlastidentifikation. Mit dessen Hilfe kann zwischen den Benutzerzellen und Zellen mit netzinternen Daten unterschieden werden:

PT	Bedeutung
000	Benutzerzelle, keine Überlast, AAL-5 SDU-Type=0
001	Benutzerzelle, keine Überlast, AAL-5 SDU-Type=1
010	Benutzerzelle, Überlast, AAL-5 SDU-Type=0
011	Benutzerzelle, Überlast, AAL-5 SDU-Type=1
100	Segment-OAM-Zelle
101	Ende-zu-Ende-OAM-Zelle
110	Reserviert für Lastmanagement
111	Reserviert

Cell Loss Priority (CLP)

Das CLP-Bit kennzeichnet die Zellen, die bei einer Überlastsituation des Netzes bevorzugt verworfen werden können.

Header Error Control (HEC)

Das HEC-Feld dient der Fehlersicherung des gesamten *Headers*. Der dabei verwendete Code erlaubt die Korrektur von Einbit-Fehlern sowie die Erkennung bestimmter Mehrbit-Fehler. Dieses Feld wird auch zur Erkennung des Zellanfangs benutzt.

Weiterhin gibt es einige reservierte *Header*-Belegungen (im wesentlichen reservierte VPI-/VCI-Werte), die zur Kennzeichnung von vordefinierten Zelltypen dienen. Dazu gehören z.B. die *Idle*-Zellen und die Management-Zellen (OAM) der physikalischen Schicht.

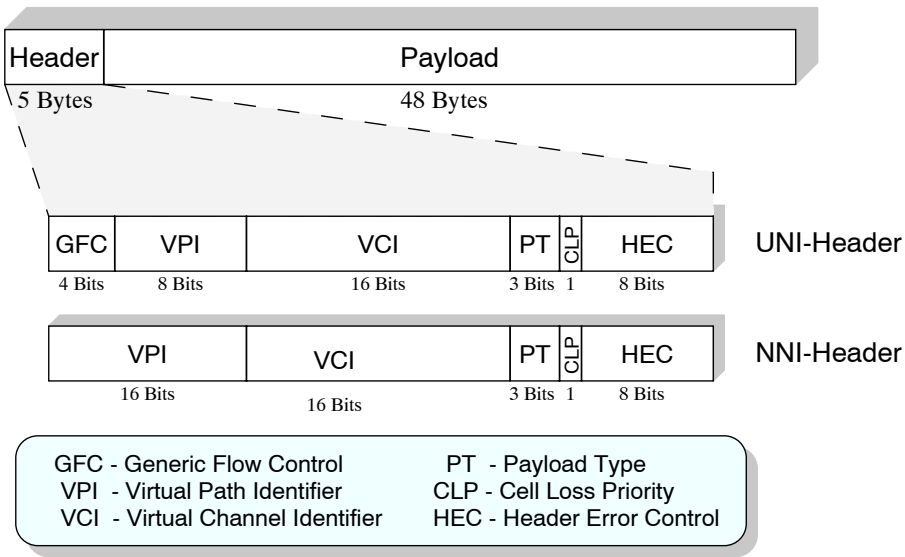


Bild 2.5: Struktur der ATM-Zelle

2.5 Die ATM-Anpassungsschicht (ATM Adaption Layer, AAL)

Die ATM-Anpassungsschicht hat die Aufgabe, die Protokoll-Dateneinheiten der höheren Schichten auf das Informationsfeld der ATM-Zellen abzubilden. Außerdem werden von der Anpassungsschicht die dazugehörigen Funktionen der Steuer- und der Managementebene zur Verfügung gestellt. Die einzelnen Funktionen der AAL-Schicht sind anwendungsorientiert und vom verwendeten Dienst abhängig.

Um die Anzahl der verschiedenen Anpassungsprotokolle zu begrenzen, wurde von ITU-T in der Empfehlung I.326 [2] für die AAL-Schicht eine Unterteilung in vier Dienstklassen mit unterschiedlichen Eigenschaften vorgeschlagen. Diese Unterteilung basiert auf folgenden Kriterien:

- Zeitbeziehung zwischen Sender und Empfänger
- Bitrate (konstant/variabel)
- Verbindungsprinzip (verbindungsorientiert/verbindungslos)

Dienst / Parameter	Klasse A	Klasse B	Klasse C	Klasse D
Zeitbeziehung	vorhanden (isochroner Verkehr)		nicht vorhanden (asynchroner Verkehr)	
Bitrate	konstant	variabel		
Verbindungstyp	verbindungsorientiert			verbindungslos
Beispiel	Circuit Emulation	Video (komprimiert)	Datentransfer	
AAL Typ	AAL1	AAL2	AAL3/4 / AAL 5	

Bild 2.6: Dienstklassen und AAL-Typen

Diese Dienstklassen werden von verschiedenen AAL-Protokolltypen abgedeckt, die in der ITU-T Empfehlung I.363 [4] ausführlich beschrieben werden. Es sind vier AAL-Typen definiert:

- AAL 1: isochroner Verkehr mit konstanten Bitraten (Klasse A)
- AAL 2: isochroner Verkehr mit variablen Bitraten (Klasse B)
- AAL 3/4: asynchrone verbindungsorientierte und verbindungslose Übertragung von Datenpaketen (Klassen C und D)
- AAL 5: vereinfachte Version von AAL 3/4 (SEAL, *Simple and Efficient Adaption Layer*)

Von der Funktionalität her wird die AAL-Schicht in zwei Teilschichten unterteilt:

- Konvergenzteilschicht (*Convergence Sublayer, CS*)
  Hier sind die anwendungsspezifischen Anpassungsfunktionen zusammengefaßt, welche die Funktionalität der ATM-Schicht entsprechend erweitern. Hier findet unter anderen der Ausgleich der Verzögerungsschwankungen, die Behandlung von Zellverlusten, das Zurückgewinnen der Senderfrequenz, die Fehlerkorrektur der Nutzdaten und die Flußkontrolle in anwendungsgerechte Weise statt.

- Segmentierungs-/Reassemblierungsteilschicht (*Segmentation and Reassembly Sublayer, SAR*)
  Innerhalb der SAR-Teilschicht findet die Zerlegung der von der CS-Teilschicht übernommenen Benutzer-Dateneinheiten, die je nach Anwendung eine variable Länge haben können, in eine Folge von ATM-Zellen statt. Auf der Empfängerseite werden dagegen die ursprünglichen Dateneinheiten wieder aus den ATM-Zellen zusamengesetzt und an die CS-Teilschicht weitergegeben. Für die Realisierung der AAL-internen Funktionen kann dabei ein Teil des Informationsfeldes der ATM-Zelle benötigt werden, der je nach Protokolltyp ein Byte (AAL 1) bis vier Bytes (AAL 3/4) groß sein kann.

Um das Zusammenspiel der verschiedenen Schichten besser verstehen zu können, wird in Bild 2.7 gezeigt, wie die Dateneinheiten untereinander ausgetauscht werden. SDU (*Service Data Unit*) bedeutet dabei die Dateneinheit, die von der darüber liegenden Schicht übergeben wird. SAR-SDU ist z.B. die Dateneinheit, die von der CS-Teilschicht an die SAR-Teilschicht geliefert wird. PDU (*Protocol Data Unit*) bedeutet dagegen die Dateneinheit, die von der Schicht selbst erzeugt wird (u.a. aus den SDU-Daten) und an die darunter liegende Schicht weitergeleitet wird, wo sie im Informationsfeld dieser Schicht transportiert wird. Anders gesagt, bezeichnet SAR-PDU die Protokoll-Dateneinheit, die zwischen den SAR-Teilschichten der beiden kommunizierenden Partner ausgetauscht wird.

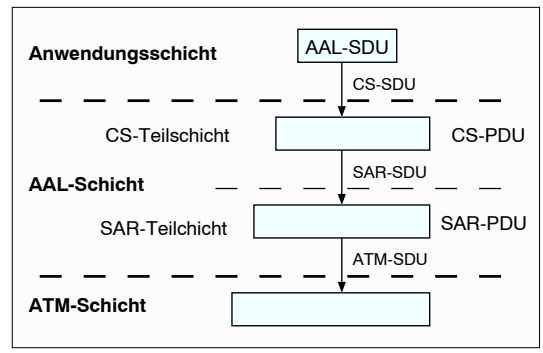


Bild 2.7: Datenfluß

2.5.1 AAL-Typ 1

AAL 1 realisiert die Dienste der Klasse A. Er bietet die Möglichkeit, verbindungsorientierte Anwendungen mit konstanten Bitraten über das ATM-Netzwerk synchron zu transportieren. Darunter versteht man solche Anwendungen, die einen kontiniuerlichen digitalen Datenstrom fester Rate benötigen, wie z.B. klassische Videoanwendungen, Sprachübertragung mit PCM-Codierung sowie die Leitungsemulation.

Weil es sich hier um isochronen Verkehr handelt, müssen neben den Daten auch die damit verbundenen Taktinformationen übertragen werden. Das kann z.B. auf der Basis des SRTS-Prinzips (*Synchronous Residual Time Stamp*) geschehen. Bei dem SRTS-Verfahren wird der Taktunterschied zwischen der Sender- und Netzfrequenz gemessen, und es wird lediglich die Frequenzabweichung des Sendetaktes von ihrem Nominalwert übertragen. Es ist dabei erforderlich, daß sowohl der Sender als auch der Empfänger den gleichen Netztakt besitzen.

Bei der Übertragung der Daten entstehen unterschiedliche Verzögerungen, die entsprechend gehandhabt werden müssen, wenn die Anwendung es verlangt. Diese variablen Verzögerungen der einzelnen Zellen werden mittels eines Pufferspeichers auf der Empfängerseite kompensiert.

Weiterhin ist es bei der Übertragung wichtig, daß die Sequenz einer Folge von Zellen vollständig erhalten bleibt. Um Zellverluste oder die Übertragung fehlerhafter Zellen erkennen zu können, werden die Zellen mit einer Folgenummer versehen. Verlorene oder fehlerhaft übertragene Zellen werden aber aus Zeitgründen nicht wiederholt.

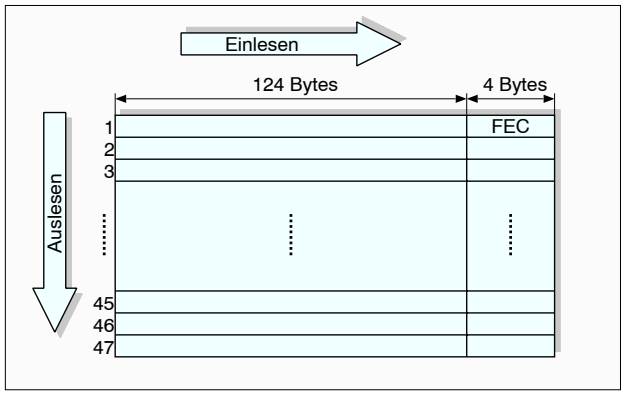


Bild 2.8: AAL-1 Korrekturverfahren

Wenn für eine Anwendung (z.B. Videoübertragung) die Bitfehlerrate oder die Zellverlustrate im Netz unakzeptabel hoch ist, muß eine Fehlerkorrektur durchgeführt werden. Innerhalb der AAL-1 Schicht wird dafür eine Kombination aus einem FEC-Verfahren (*Forward Error Correction*) und einer Byte-Verschachtelung benutzt. Dafür wird eine 128x47-Bytes große Matrix zeilenweise aufgebaut. Für jede Zeile werden dabei jeweils 124 Datenbytes eingelesen, die dann mit einem 4 Bytes langen FEC-Code versehen werden. Für die Übertragung wird die Matrix spaltenweise ausgelesen. Die dabei resultierenden 47-Bytes Datenblöcke entsprechen größenmäßig den von der SAR-Teilschicht erwarteten Dateneinheiten und werden in der SAR-PDU zum Empfänger transportiert. Für die FEC-Korrektur wird hier ein Reed-Solomon-Code benutzt. Damit ist es möglich, in einem 128 Bytes langen Block zwei fehlerhafte bzw. bei Zellverlusten (wenn die Position bekannt ist) vier verlorene Bytes zu korrigieren.

Wie in Bild 2.9 dargestellt, erhält die Konvergenzteilschicht (CS) des AAL 1 von der ihr übergeordneten Anwendungsschicht Informationseinheiten (AAL-SDUs), deren Länge von der Anwendung abhängt und z.B. bei Leitungsemulation 1 Bit beträgt, bei Video- oder Audioanwendungen dagegen 1 Byte. Diese in regelmäßigen Abständen kommenden Dateneinheiten werden gesammelt, bis ein 47-Bytes langer Datenblock aufgefüllt werden kann. Dieser wird dann von der SAR-Teilschicht mit einem 1 Byte langen *Header* versehen, in den unter anderem die Sequenznummer eingetragen wird. Die so gebildeten SAR-PDUs, die jetzt schon 48 Bytes lang sind, werden direkt an die darunterliegende ATM-Schicht weitergeleitet, wo sie im Informationsfeld einer ATM-Zelle übertragen werden.
Der von der SAR-Teilschicht erzeugte SAR-PDU-*Header* enthält folgende Informationen:

Convergence Sublayer Identification (CSI)
: Das CSI-Bit wird innerhalb der CS-Teilschicht für verschiedene Funktionen benutzt, und zwar für die Übertragung der Taktinformationen vom Sender zum Empfänger (beim SRTS-Verfahren) sowie für Strukturinformationen bei der Übertragung von strukturierten Daten.

Sequence Number (SN)
: Hier wird die aktuelle Folgenummer (modulo 8) der jeweiligen Zelle eingetragen.

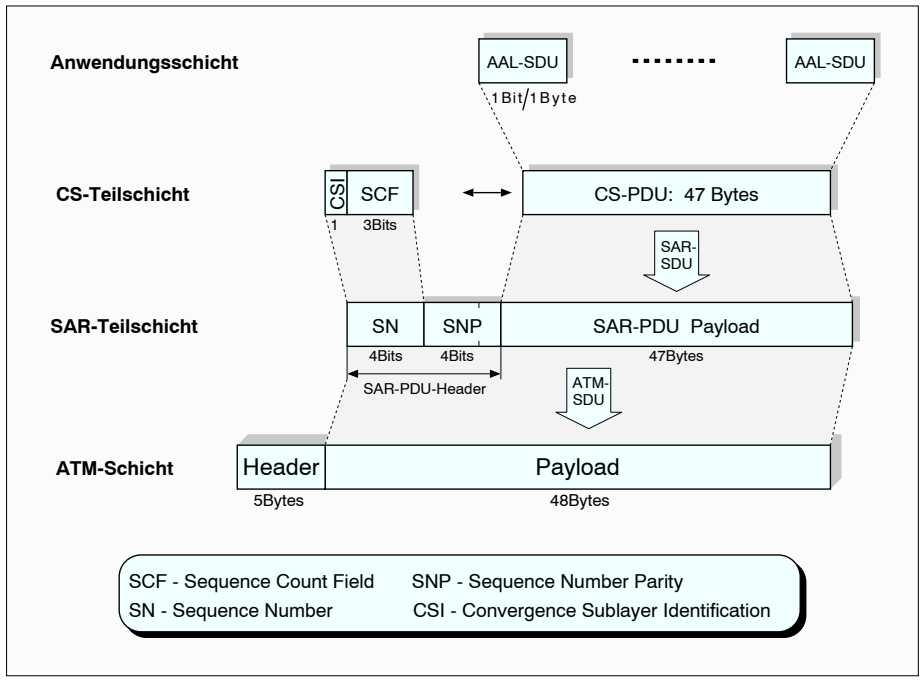


Bild 2.9: Die Struktur von AAL-Typ 1

Sequence Number Parity (SNP)
: Das SNP-Feld dient dem Schutz des SN-Feldes. Es besteht aus einer CRC-Prüfsumme (3 Bits) und einem Paritätsbit, mit dem das gesamte Byte zusätzlich gesichert wird.

2.5.2 AAL-Typ 2

AAL 2 entspricht den Protokollen der Dienstklasse B. Er ist für verbindungsorientierte Anwendungen mit variablen Bitraten (VBR) vorgesehen, wobei es sich hier auch wie bei AAL 1 um isochronen Verkehr handelt. Darunter fallen vor allem Video- und Audioanwendungen, die intern mit Kompressionsalgorithmen arbeiten.

Für diese Dienstklasse gibt es kein Vorbild. Sie wurde bisher von keiner Netztechnologie unterstützt und es gibt keine Erfahrungen, auf denen man aufbauen konnte. Auch die ATM-Spezifikation dieser Dienstklasse ist noch nicht weit fortgeschritten. Auf die fertigen Standards wird man noch einige Zeit warten müssen, insbesondere weil sich die Forschungsarbeiten und Pilotversuche zunächst auf die Anwendungen konzentrieren, die auf den Anpassungsschichten AAL 1 und AAL 5 basieren. Viele Ansätze und Vorschläge und die damit verbundenen Probleme sind von ganz neuer Art. Die Lösungen dafür müssen noch ausgearbeitet und untersucht werden. Viele interessante Fragen sind noch offen:

- Wie kann man die Verkehrsparameter bei solchem Verkehr spezifizieren (Maximalrate, Minimalrate, Durchschnittsrate, *Burst*-Dauer, *Burst*-Häufigkeit ...) ?
- Wie werden die Verkehrsparameter (QoS) überwacht und eingehalten?
- Was passiert bei Überschreitung der ausgehandelten Verkehrsparameter?

2.5.3 AAL-Typ 3/4

AAL 3/4 ist für die verbindungsorientierte (Klasse C) und für die nicht-verbindungsorientierte (Klasse D) Übertragung von Datenpaketen (asynchroner Verkehr) zuständig. Es werden sowohl Punkt-zu-Punkt- als auch Punkt-zu-Mehrpunkt-Verbindungen unterstützt.

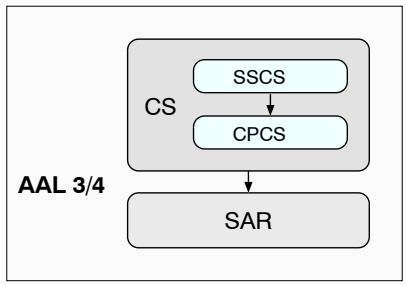


Bild 2.10: Die AAL 3/4

Aufgrund der immer stärkeren Verwendung des AAL 5 für verbindungsorientierte Dienste, findet AAL 3/4 hauptsächlich Anwendung in der Übertragung verbindungsloser Dienste, die auf dem SMDS/CBDS[1]- oder *Frame Relay* Protokoll basieren. Außer der Aufteilung der AAL 3/4 Schicht in die SAR- und CS-Teilschichten wird die Konvergenzteilschicht (CS) noch weiter in zwei Teilschichten unterteilt. Sie besteht aus einem gemeinsamen Teil (CPCS, *Common Part Convergence Sublayer*) und einem darüber liegenden dienstspezifischen Teil (SSCS, *Service Specific Convergence Sublayer*). Letzterer kann allerdings entfallen, wie das z.B. beim SMDC/CBDS-Dienst geschieht, der direkt auf dem CPCS aufsetzt.
Grundsätzlich unterscheidet AAL 3/4 zwei verschiedene Betriebsarten:

- *Message Mode*
  Im *Message*-Modus erhält die Anpassungsschicht von der darüberliegenden Anwendungsschicht eine Dateneinheit (AAL-SDU) in genau einer sogenannten AAL-IDU (*Interface Data Unit*). Mehrere, kurze AAL-SDUs fester Länge werden dabei innerhalb einer CS-PDU transportiert (*Blocking/deblocking*-Funktion wird aktiviert). AAL-SDUs variabler Länge werden dagegen in einer oder mehreren CS-PDUs (*Segmentation/Reassembly*-Funktion) übertragen (vgl. Abb. 2.11a) und 2.11b)).

- *Streaming Mode*
  Im *Streaming*-Modus werden die Dateneinheiten (AAL-SDUs) an die Anpassungsschicht in einer oder mehreren AAL-IDUs übergeben, wobei die Übergabe allerdings zeitlich unabhängig voneinander geschehen kann. Auch hier werden die AAL-IDUs in einer oder in mehreren CS-PDUs übertragen, wobei bei aktiver *Pipeline*-Funktion einzelne CS-PDUs bereits gesendet werden können, bevor die AAL-SDU vollständig empfangen wurde (vgl. Abb. 2.11c) und 2.11d)).

Für beide Betriebsarten ist sowohl eine 'garantierte' (*assured*) als auch eine 'nicht garantierte' (*non-assured*) Übertragung definiert. Bei der 'garantierten' Übertragung, die vor allem für Punkt-zu-Punkt Verbindungen gedacht ist, werden fehlerhafte oder verlorene AAL-SDUs nochmals übertragen. Es werden auch Funktionen zur Flußkontrolle bereitgestellt. Bei der 'nicht garantierten' Übertragung wird dagegen keinerlei Fehlerkorrektur durchgeführt. Die Übertragung innerhalb der CPCS-Teilschicht erfolgt immer 'nicht garantiert'. Die Funktionen für 'garantierte' Übertragung werden erst von der jeweiligen SSCS-Teilschicht erbracht (wie z.B. bei SSCOP: *Service Specific Connection-Oriented Protocol*).

Wie in Bild 2.12 dargestellt, bilden die zu übertragenden Daten variabler Länge (1-65535 Bytes), die zunächst mit PAD-Bytes aufgefüllt werden, das Informationsfeld einer CPCS-

[1] SMDS: *Switched Multi-megabit Data Service*, CBDS: *Connectionless Broadband Data Service*

PDU. Dieses *Payload*-Feld wird von einem *Header* und einem *Trailer* eingerahmt, wobei die einzelnen Felder dieser CPCS-PDU folgende Bedeutung haben:

Common Part Indicator (CPI)
: Hier werden die Einheiten der Werte in den Feldern *BA-Size* und *Length* spezifiziert. Bisher ist nur die Belegung '0000 0000' für die Einheit 'Byte' definiert.

Begin-End-Tag (BE-tag)
: Mit Hilfe des *BE-tag*-Feldes kann man das Ende eines CPCS-PDU-*Headers* bzw. den Beginn eines CPCS-PDU-*Trailers* erkennen. In einer CPCS-PDU besitzen beide Felder den gleichen Wert, in zwei aufeinanderfolgenden PDUs müssen sie sich jedoch unterscheiden.

Buffer Allocation Size Indicator (BA-size)
: In diesem Feld wird dem Empfänger mitgeteilt, wieviel Pufferspeicher für den Empfang der betreffenden CPCS-PDU reserviert werden muß. Es sind Werte bis 65535 möglich, die Einheit wird im CPI-Feld spezifiziert.

Padding Bytes (PAD)
: Mit PAD-Bytes wird die Länge des Informationsfeldes auf ganzzahlige Vielfache von 32 Bits ergänzt, was eine hardware-orientierte Bearbeitung erleichtert.

Alignment Field (AL)
: Dieses Ausgleichsfeld dient zur Ergänzung des CPCS-*Trailers* auf genau 4 Bytes.

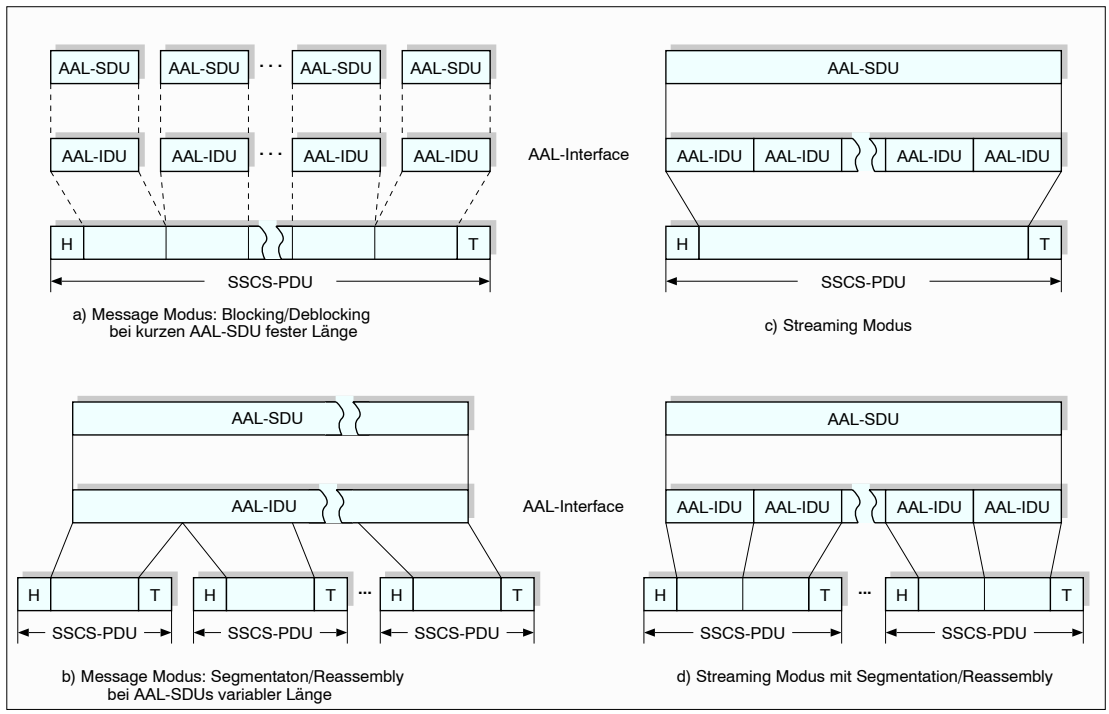


Bild 2.11: Betriebsarten von AAL-Type 3/4

Length

In diese zwei Bytes wird die Länge der CPCS-*Payload*-Feldes eingetragen. Die Einheit wird ebenfalls im CPI-Feld spezifiziert.

Von der SAR-Teilschicht wird dann die übergebene CPCS-PDU in 44 Bytes lange Datenblöcke aufgespalten. Diese werden mit einem weiteren *Header* und *Trailer* versehen und als 48 Bytes lange SAR-PDU an die ATM-Schicht weitergeleitet. Die Felder des SAR-PDU-*Headers* und SAR-PDU-*Trailers* haben folgende Bedeutung:

Sequence Number (SN)

Mit Hilfe dieses Feldes werden die SAR-PDUs modulo 16 durchnumeriert. Durch Auswertung dieses Feldes kann der Empfänger den Verlust oder die Übertragung einer nicht zur betreffenden Sequenz gehörigen SAR-PDU feststellen.

Length Indication (LI)

Das Längenfeld gibt an, wieviel Bytes des SAR-*Payload*-Feldes belegt sind. Eine EOM-SAR-PDU mit dem LI-Wert von 63 kennzeichnet eine sogenannte Abbruch-PDU (Abbruch der Übertragung einer CS-PDU).

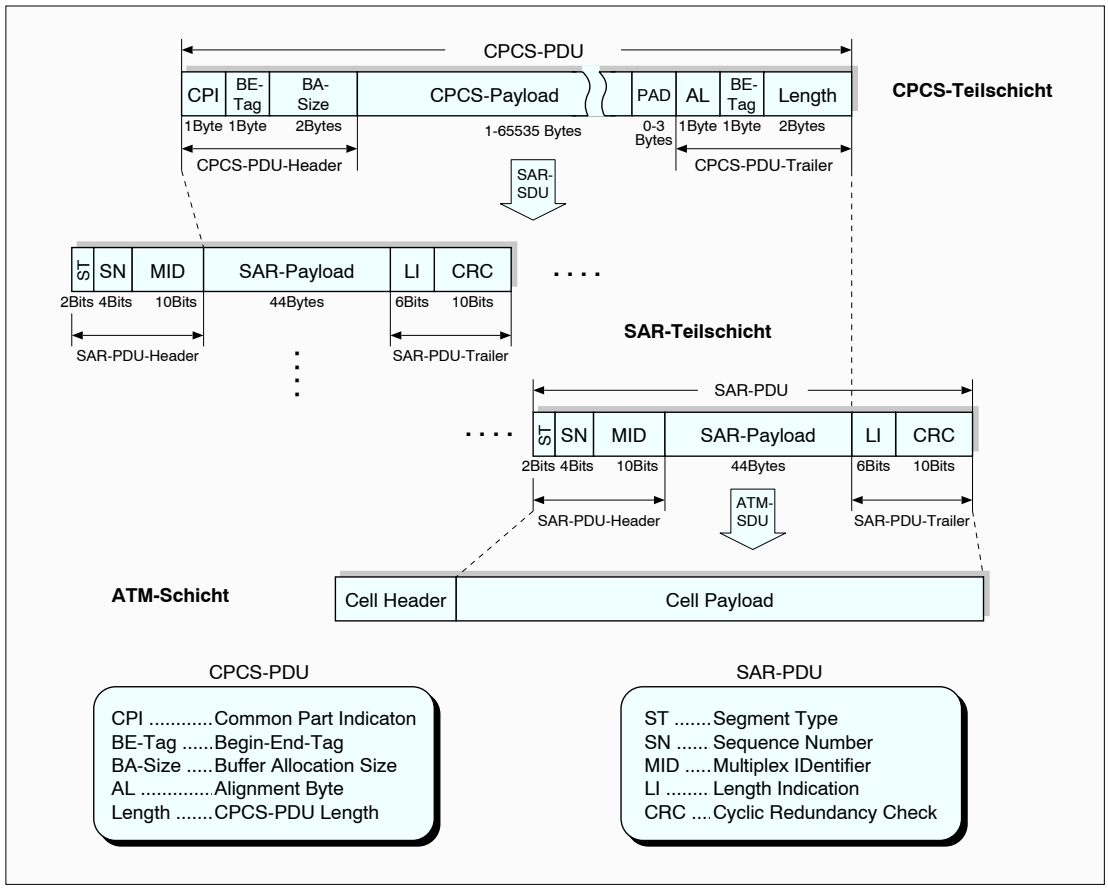


Bild 2.12: Die Struktur von AAL-Type 3/4

Multiplex IDentifier (MID)
: Dieses Feld bietet die Möglichkeit, mehrere CPCS-Verbindungen über eine virtuelle ATM-Verbindung zu multiplexen.

Segment Type (ST)
: Das Segmenttyp-Identifikationsfeld gibt an, um welchen Teil einer CPCS-PDU sich bei der betreffenden SAR-PDU handelt:

Segmenttyp	Codierung	Gültige Werte für LI-Feld
BOM (*Begin of Message*)	10	44
COM (*Continuation of Message*)	00	44
EOM (*End of Message*)	01	4...44, 63 (=Abbruch)
SSM (*Single Segment Message*)	11	8...44

Cyclic Redundancy Check (CRC)
: Mit diesem Feld wird die gesamte SAR-PDU durch eine CRC-10-Prüfsumme gesichert.

2.5.4 AAL-Type 5

AAL-Type 5 ist eine stark vereinfachte Version des AAL 3/4-Protokolls, die für die verbindungsorientierte (Klasse C) Übertragung von Datenpaketen geeignet ist. Die beiden Betriebs-

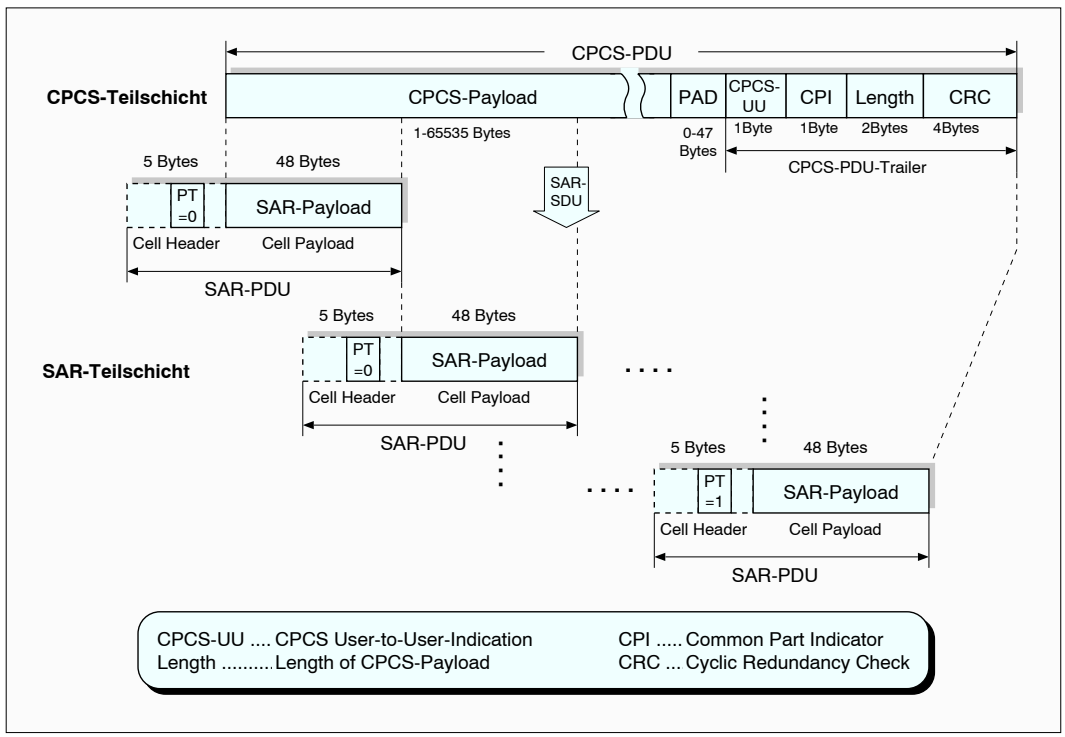


Bild 2.13: Die Struktur von AAL-Type 5

arten *Message*-Modus und *Streaming*-Modus sowie 'garantierte' und 'nicht garantierte' Übertragung werden in AAL 5 auf die gleiche Weise wie bei AAL 3/4 unterstützt. Was allerdings von AAL 5 im Unterschied zu AAL 3/4 nicht unterstützt wird, ist das Multiplexen verschiedener AAL-Verbindungen innerhalb einer ATM-Verbindung. Weiterhin besteht die Konvergenzteilschicht des AAL 5 ebenso wie bei AAL 3/4 aus einer gemeinsamen CPCS-Teilschicht und aus einer SSCS-Teilschicht, die jedoch nur dann vorhanden ist, wenn anwendungsspezifische Anpassungen notwendig sind.

Wie in Bild 2.13 gezeigt wird, werden die von der Anwendung übergebenen Datenpakete variabler Länge (1-65535 Bytes) innerhalb der CS-Teilschicht zuerst durch PAD-Bytes aufgefüllt und mit einem *Trailer* versehen. Im Gegensatz zu AAL 3/4 besitzt eine CPCS-PDU keinen *Header*. Die so gebildeten CPCS-PDUs (Konvergenzteilschicht-PDUs) werden dann von der SAR-Teilschicht in 48 Bytes lange Datenblöcke zerlegt, die dann direkt im Informationsfeld der ATM-Zelle ohne weiteren *Overhead* übertragen werden. Um das Ende einer CPCS-PDU erkennen zu können, wird zum Zwecke der Identifikation das PT-Feld des Zell-*Headers* "mißbraucht". Falls es sich um das Ende der CPCS-PDU handelt, wird es auf 1 gesetzt, beim Beginn bzw. der Fortsetzung einer CPCS-PDU auf 0.
Die Felder der CPCS-PDU haben folgende Bedeutung:

Padding Bytes (PAD)
: Mit 0 bis 47 Füllbytes wird die CPCS-PDU auf ein ganzzahliges Vielfaches von 48 Bytes aufgefüllt, damit nach der Segmentierung keine teilgefüllten ATM-Zellen entstehen.

CPCS-User-to-User-Identification (CPCS-UU)
: Dieses Feld ist für die Übertragung von Benutzerinformationen bestimmt.

Common Part Indicator (CPI)
: Die Funktion dieses Feldes ist bisher noch nicht festgelegt.

Length
: Das Längenfeld gibt die Länge der CPCS-*Payload* an, und kann Werte von 0 bis 65535 annehmen. Eine CPCS-PDU mit der Längenangabe 0 wird *Abort*-PDU genannt und führt zum Abbruch der laufenden CPCS-SDU-Übertragung.

Cyclic Redundancy Check (CRC)
: Hier wird die CRC-32-Prüfsumme übertragen, durch die die CPCS-PDU geschützt wird.

2.6 Signalisierung

In ATM-Netzen werden die Signalisierungsinformationen in einem eigenem virtuellen Kanal getrennt von der Nutzinformationen übertragen. Diese Signalisierungsverbindungen müssen wie jede andere virtuelle Verbindung zuerst aufgebaut werden. Dafür wird die sogenannte Meta-Signalisierung verwendet, die auf einer reservierten VPI/VCI-Belegung basiert.

Für die Signalisierungskanäle wird *Signaling ATM Adaption Layer* (SAAL) verwendet, der auf AAL 3/4 oder AAL 5 basiert. Weil die Signalisierungsprotokolle aber eine sichere Übertragung benötigen, wird für die anwendungsspezifische Teilschicht (SSCS) innerhalb der Konvergenzteilschicht (CS) ein spezielles SSCOP-Protokoll (*Service Specific Connection Oriented Protocol*) verwendet.

An den verschiedenen Schnittstellen in einem ATM-Netzwerk gibt es auch verschiedene Signalisierungsprotokolle. Für die öffentlichen Netze werden die Protokolle von ITU-T spezifiziert. Das Protokoll am UNI wurde aus dem Signalisierungsprotokoll Q.931 für Schmalband-ISDN weiterentwickelt und ist in der ITU-T Empfehlung Q.2931 [6] beschrieben. Das Signalisierungsprotokoll am NNI (B-ISUP[2]) entstand aus dem ISDN-ISUP (Q.761-Q.764) und ist in den Empfehlungen Q.2761-Q.2764 beschrieben.

Für die privaten Netze ist das ATM-Forum zuständig. Der aktuelle Standard für die UNI-Signalisierung wird im ATM-Forum UNI 3.1 [34] beschrieben. Diese Spezifikation basiert auf der Q.2931 Empfehlung. Für die NNI-Schnittstelle bemüht sich das ATM-Forum derzeit, einen Standard zu definieren, der mit P-NNI-Phase-1 bezeichnet wird und auch das Signalisierungsprotokoll an dieser Schnittstelle enthalten soll. Als Zwischenlösung wurde vom ATM-Forum ein IISP-Protokoll [35] (*Interim Inter-Switch Signaling Protocol*) definiert. IISP, das auch P-NNI-Phase-0 genannt wird, ermöglicht auf der Basis von UNI 3.1 die Signalisierung zwischen *Switches* zu realisieren.

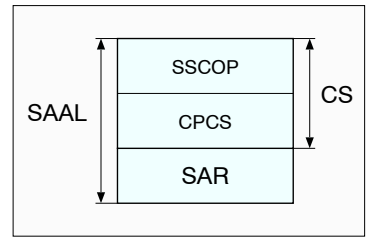


Bild 2.14: Signaling AAL

[2] B-ISUP: B-ISDN User Part

Kapitel 3

Classical IP over ATM

Einer der ersten Vorschläge, wie man IP über ATM verwenden kann, wurde unter der Regie der *Internet Engineering Task Force* (IETF) entworfen. Es handelt sich dabei um das *Classical-IP*-Modell, das von der *'IP over ATM'*-Arbeitsgruppe der IETF entwickelt wurde. Wie bei allen Netzwerkschicht-Protokollen, die über ATM transportiert werden sollen, mußten zwei Aspekte betrachtet werden:

- *Packet Encapsulation:*
  Wie sollen die verschiedene IP-Pakete verpackt und danach in den ATM-Zellen übertragen werden?
- *Address Resolution:*
  Wie können die Netzwerkschichtadressen (IP-Adressen) nach ATM-Adressen aufgelöst werden, und wie sieht die Netzstruktur aus?
- Verbindungsverwaltung:
  Wie soll ein verbindungsloses IP-Protokoll über ein verbindungsorientiertes ATM-Netz transportiert werden; wie werden die Verbindungen auf- und abgebaut?

Die Ergebnisse wurden von der IETF in folgenden zwei wichtigen RFCs (*Request for Comments*) veröffentlicht:

- **RFC 1483**: *'Multiprotocol Encapsulation over ATM'* [12]
  Dieser im Juli 1993 publizierte RFC beschreibt, wie Datenpakete verschiedener Protokolle, insbesondere IP-Pakete, über ATM verschickt werden können.
- **RFC 1577**: *'Classical IP and ARP over ATM'* [11]
  In diesem RFC, der im Januar 1994 folgte, wird gezeigt, wie IP-Netze über ATM realisiert sein sollen. Zur Adreßauflösung wird ein ATMARP-Server benutzt, und es wird mit einer an die ATM-Verhältnisse angepaßten Version des klassischen *Address Resolution Protocol* (ARP) gearbeitet.

Die beiden RFCs stellen eine einfache, standardisierte Grundlage für die Implementierungen dar, die eine direkte Nutzung von IP über ATM erlauben. Die Grundidee ist dabei, den IP-Stationen zu ermöglichen, ein ATM-Netz als Transportmedium zu verwenden. Der in den RFCs vorgestellte Ansatz ist protokolltransparent. Anwendungen, die auf IP aufsetzen, können dabei ohne Modifikationen über ATM weiter benutzt werden.

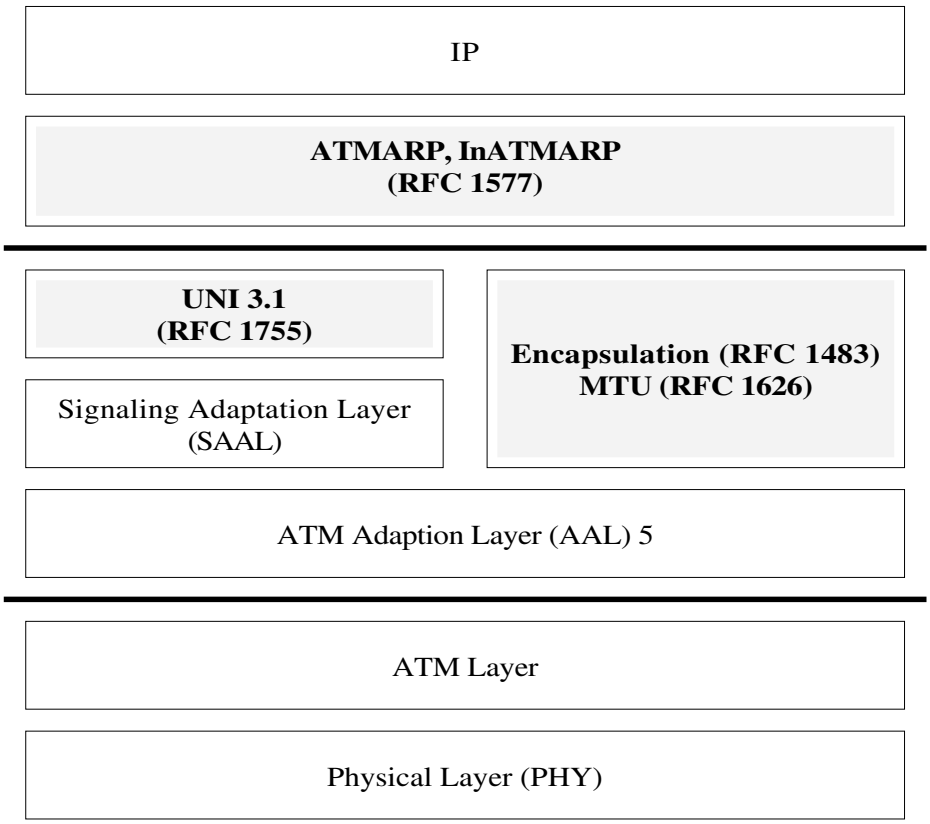


Bild 3.1: IP über ATM

Das *Classical-IP*-Modell wurde später in weiteren RFCs verfeinert und erweitert:

- **RFC 1626**: *'Default IP MTU for use over ATM AAL 5'* [10]
  Der im Mai 1994 veröffentlichte RFC definiert die maximalen Paketgrößen in ATM-Netzen.
- **RFC 1755**: *'ATM Signaling Support for IP over ATM'* [9]
  Dieser RFC wurde im Februar 1995 veröffentlicht und beschreibt die Verwendung der ATM-Forum-Signalisierung (UNI 3.1) in *Classical-IP*-Netzen.

Bild 3.1 stellt die Protokollarchitektur für *Classical IP* über ATM dar und zeigt, wo die einzelnen RFCs verwendet werden.

An vielen Aspekten der Zusammenarbeit von IP und ATM wird allerdings noch gearbeitet. Einen guten Überblick über die aktuellen Aktivitäten der IETF in diesen Bereich findet man im *Informational* **RFC 1932** '*IP over ATM: A Framework Document*' [8].

3.1 LLC/SNAP-Verpackung der IP-Pakete gemäß RFC 1483

RFC 1483 [12] mit dem Titel *'Multiprotocol Encapsulation over ATM Adaptation Layer 5'* definiert, wie Datenpakete (PDUs) verschiedener Protokolle, insbesondere IP-Pakete, in ATM-Zellen transportiert werden können. Als Anpassungsprotokolltyp wurde AAL-Typ 5 gewählt, der im Vergleich zu AAL 3/4 weniger *Overhead* produziert, effizienter implementiert werden kann, und von der Funktionalität völlig ausreichend ist. Die Datenpakete der verschiedenen Protokolle werden im *Payload*-Feld der gemeinsamen Konvergenzteilschicht (CPCS) von AAL 5 übertragen. Die anwendungsspezifische SSCS-Teilschicht bleibt in diesem Fall leer.

Für den Transport mehrerer Protokolle über ATM existieren zwei prinzipielle Möglichkeiten (siehe Bild 3.2):

- *VC-Based Multiplexing*
  Für jedes Protokoll wird eine eigene ATM-Verbindung benutzt. Die Methode benötigt keinen zusätzlichen *Overhead*, und das zu transportierende Protokoll wird implizit durch die ATM-Verbindung identifiziert. Sie ist für Netzwerkumgebungen interessant, bei denen man eine große Anzahl von virtuellen ATM-Verbindungen (SVCs) schnell und effizient verwalten kann. Dies wird zunächst in privaten ATM-Netzen der Fall sein. Die Methode wird u.a. bei der LAN-Emulation angewendet.

- *LLC Encapsulation*
  Die unterschiedlichen Protokolle werden über die gleiche ATM-Verbindung gemultiplext. Die Datenpakete der verschiedenen Protokolle werden dabei mit einem *Header* versehen, der es erlaubt, das Paket dem richtigen Protokoll zuzuordnen. Die Methode findet vor allem dann Anwendung, wenn das Schalten von Verbindungen kompliziert ist oder im Netz nur PVCs unterstützt werden.

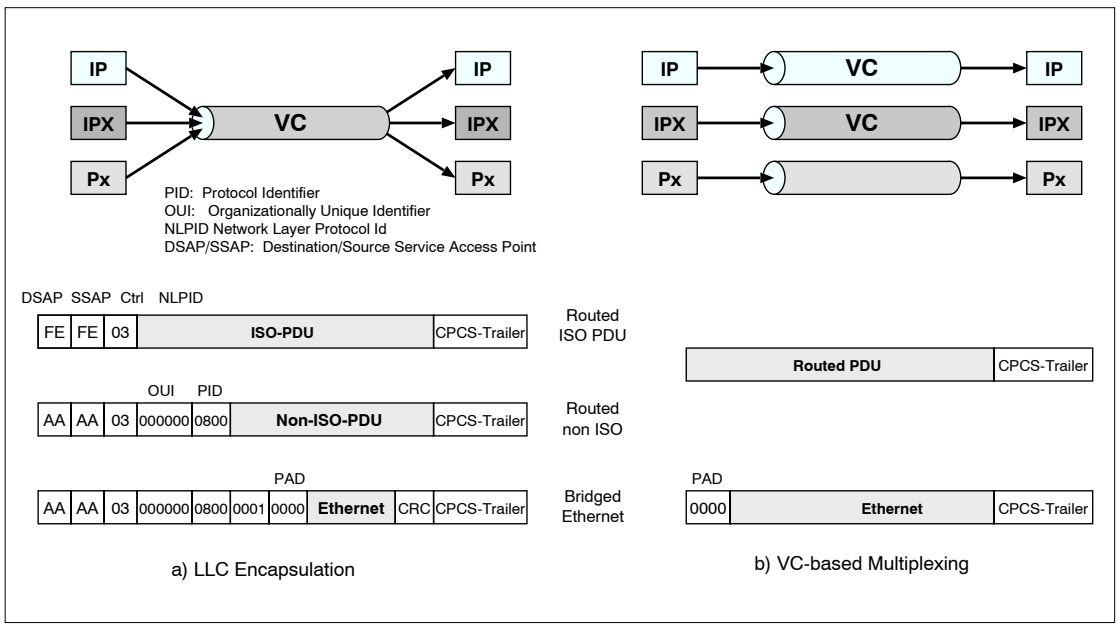


Bild 3.2: Verpackung der Pakete gemäß RFC 1483

Die Auswahl der Methode erfolgt bei PVCs durch Vereinbarungen während der Konfigurationsphase und bei SVCs durch Absprache während der Signalisierung beim Verbindungsaufbau (siehe dazu Kap. 3.4 auf Seite 41). Während bei LLC-*Encapsulation* der Protokolltyp aufgrund des LLC-*Headers* erkannt wird, muß er bei VC-basiertem Multiplexing explizit einer ATM-Verbindung zugeordnet werden. Das kann wieder entweder manuell konfiguriert werden oder geschieht automatisch während der Signalisierung.

Bei *Classical IP* kommt die zweite Methode zum Einsatz. Es wird ein LLC-*Header* (*Logical Link Control*) gemäß IEEE 802.2, gefolgt von einem SNAP-*Header* (*SubNetwork Attachment Point*) gemäß IEEE 802.1a, verwendet, so wie das in Bild 3.3 dargestellt ist. Die Methode ist daher für alle Protokolle geeignet, die den LLC-*Header* verwenden, wie z.B. Ethernet, Token Ring, DQDB sowie auch FDDI.

Der LLC-*Header* besteht aus 3 Bytes und erlaubt unter anderem, zwischen ISO- und Nicht-ISO-Protokollen zu unterscheiden. Für die IP-Pakete wird er mit dem Wert 0xAA-AA-03 belegt, was auf die Existenz des SNAP-*Headers* hinweist, in dem ein Nicht-ISO-Protokoll identifiziert werden soll. Der SNAP-*Header* liegt direkt hinter dem LLC-*Header* und besteht aus dem 3 Bytes langen *Organizationally Unique Identifier* (OUI) und dem 2 Bytes langen *Protocol IDentifier* (PID). Der Wert 0x00-00-00 des OUI gibt an, daß der PID einen Ethernet-Typ enthält. Insbesondere hat der Ethernet-Typ für die normale Internet PDU den Wert 0x08-00 (siehe RFC-1340 [13]).

In RFC 1483 wurde auch auf eine andere Möglichkeit hingewiesen, wie man IP-Datenpakete mit dem LLC-*Header* verpacken kann. Wird der LLC-*Header* mit dem Wert 0xFE-FE-03 belegt, bedeutet das, daß ein 'geroutetes' ISO Protokoll benutzt wird. Ein geroutetes ISO-Protokoll wird nicht über den SNAP-*Header*, der in diesem Fall entfällt, sondern über ein NLPID-Byte (*Network Layer Protocol IDentifier*) identifiziert, das ein Teil der PDU ist. Die Werte für NLPID werden von ISO und ITU verwaltet (siehe [7]). Es ist möglich, die IP-Datenpakete auf diese Weise zu verpacken, denn obwohl IP kein ISO-Protokoll ist, wurde für IP auch ein NLPID-Wert (0xCC) reserviert.

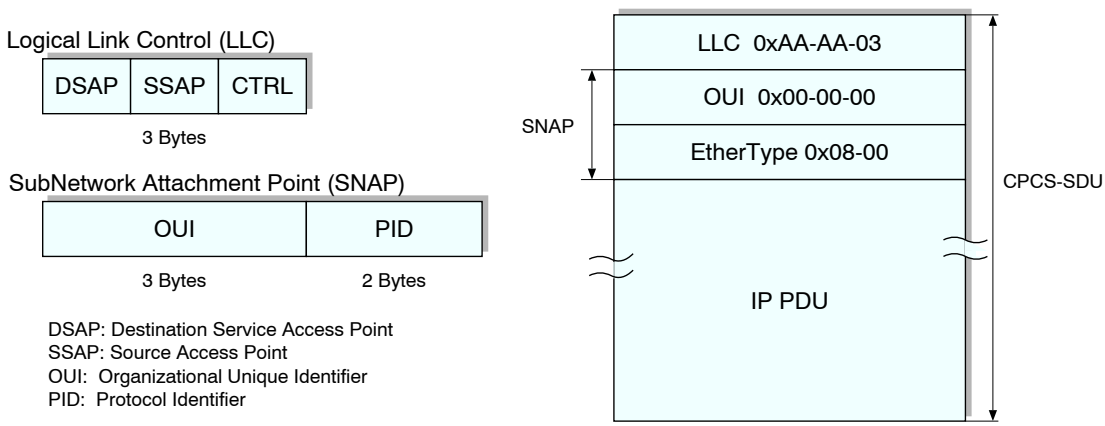


Bild 3.3: Struktur des CPCS-PDU *Payload*-Feldes für Internet PDU

3.2 Adreßauflösung nach RFC 1577 (ARP über ATM)

Will man IP-Datenpakete über ATM transportieren, braucht man einen Mechanismus, um die IP-Adressen nach ATM-Adressen auflösen zu können – ähnlich wie die Umsetzung von IP-Adressen nach IEEE MAC-Adressen in klassischen LANs.

Erhält ein Router mit einem ATM-*Interface* ein Paket über ein LAN-Interface, konsultiert er zuerst seine Routing-Tabelle, um festzustellen, an welchen Port und an welchen Nachbar-Router das Paket weitergeschickt werden soll. Soll das Paket über ein ATM-*Interface* geschickt werden, so überprüft der Router seine lokale Adreßtabelle. Er sucht entweder nach der richtigen ATM-Adresse oder nach den entsprechenden VCI/VPI-Werten, falls eine direkte PVC-Verbindung zu dem betreffenden Ziel existiert.

Es können statische Tabellen benutzt werden, die manuell konfiguriert wurden. In dieser Form ist der Verwaltungsaufwand jedoch sehr groß, das ganze System unflexibel und nicht gut skalierbar. In RFC 1577 [11] wird beschrieben, wie eine automatische (dynamische) Auflösung von IP-Adressen in einem ATM-Netz über AAL 5 realisiert werden kann.

Für das *Classical*-IP werden innerhalb des ATM-Netzes logische IP-Subnetze (*Logical IP Subnetwork*, LIS) definiert. Wie bei den klassischen IP-Subnetzen (etwa auf Ethernet), besteht ein LIS aus einer Reihe von IP-Knoten, wie Workstations oder Router, die alle die gleiche IP-Netznummer und Subnetzmaske besitzen und zum gleichen IP-Subnetz gehören. Ein LIS arbeitet unabhängig von jedem anderen LIS auf dem gleichen ATM-Netz. Alle Endsysteme innerhalb des gleichen LIS kommunizieren direkt über ATM-Verbindungen. Andere Systeme, die in anderen LIS liegen, sowie auch die traditionellen Netze, werden über einen IP-Router erreicht. Das bedeutet, daß zwei IP-Knoten, die zu verschiedenen IP-Subnetzen gehören, immer über einen Router kommunizieren müssen, auch wenn man zwischen ihnen direkt eine virtuelle ATM-Verbindung aufbauen könnte. Ein Router wird als Endsystem betrachtet und wird als Knoten in mehr als einem LIS konfiguriert.

In den klassischen *Shared-Medium*-LAN-Umgebungen erfolgt die Adreßauflösung der IP-Adressen nach IEEE MAC-Adressen mit dem *Address Resolution Protocol* (ARP) [21]. Will

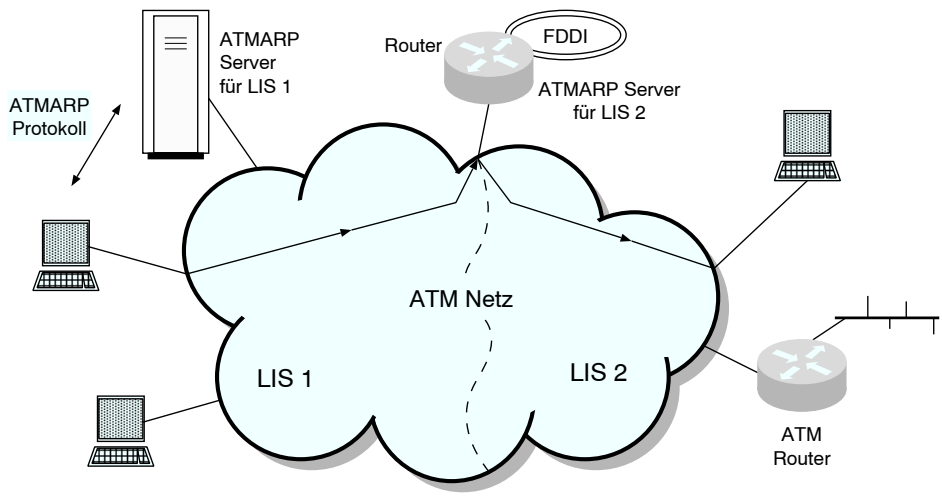


Bild 3.4: *Classical*-IP Modell: Netzübergang mittels Router realisiert

ein Rechner mit einem anderen kommunizieren, kennt aber nur dessen IP-Adresse, dann verschickt er eine ARP-*Broadcast*-Nachricht mit der IP-Adresse, zu der die MAC-Adresse gesucht wird. Derjenige Rechner, der diese IP-Adresse als seine erkannt hat, antwortet auf die Nachricht und übermittelt seine MAC-Adresse. Für die umgekehrte Umsetzung wird das *Inverse Address Resolution Protocol* (InARP) [15] verwendet. Um eine IP-Adresse zu einer MAC-Adresse zu erfahren, sendet ein Rechner einen InARP-*Request* per *Broadcast*, und als Antwort erhält er von einem oder mehreren Servern die gesuchte IP-Adresse.

In einem nicht *broadcast*-fähigen Netz wie ATM, können die klassischen Protokolle auf die oben beschriebene Weise nicht funktionieren. In RFC 1577 *'Classical IP and ARP over ATM'* wird gezeigt, wie sie an die Gegebenheiten des ATM-Netzes angepaßt werden können. Das ARP wurde zum ATMARP und das InARP zum InATMARP weiterentwickelt. Zur Unterstützung der gegenseitigen Abbildung der ATM- und IP-Adressen wird anstelle einer *Broadcast*-Verteilung der ARP-Nachrichten ein ATMARP-Server verwendet. In einem LIS gibt es genau einen solchen Server, der sich in einer Workstation, einem Router oder einem ATM-*Switch* realisieren läßt. Am Anfang registrieren alle Clients ihre IP-Adresse beim ATMARP-Server. Ein Client schickt dann einen ATMARP-*Request* zum ATMARP-Server, falls er eine ATM-Adresse zu einer bestimmten IP-Adresse benötigt.

Funktionsweise des ATMARP-Mechanismus
In einem logischen ATM-IP-Subnetz erfolgt die Adreßauflösung mit dem *ATM Address Resolution Protocol* (ATMARP: wie lautet die ATM-Adresse zu einer gegebenen IP-Adresse?) und dem *Inverse ATM Address Resolution Protocol* (InATMARP: wie lautet die IP-Adresse zu einer gegebenen ATM-Adresse?). Sie müssen von allen IP-Stationen entsprechend unterstützt werden. Wann und auf welche Weise sie benutzt werden, hängt davon ab, ob PVCs (*Permanent Virtual Connections*) oder SVCs (*Switched Virtual Connections*) benutzt werden.

Sowohl der ATMARP-Server als auch die Clients bewahren die im Laufe der Zeit gesammelten Adreßinformationen in ihren lokalen ATMARP-Tabellen auf. Die Einträge in der Tabelle werden mit einer *Timeout*-Information versehen. In der ATMARP-Tabelle des Clients sind die Einträge nur maximal 15 Minuten, in der Server-ATMARP-Tabelle dagegen mindestens 20 Minuten gültig. Sie müssen in regelmäßigen Zeitabständen verifiziert werden. Mit der Tabelle werden auch die exisitierenden, virtuellen Verbindungen (VCs) verknüpft, die jeweils dem entsprechenden Eintrag in der Tabelle zuzuordnen sind.

SVC-Umgebung. In einer SVC-Umgebung muß von jeder Station sowohl ATMARP als auch InATMARP unterstützt werden. Beim Ermitteln von Adressen hilft der ATMARP-Server. Er ist für die korrekte Beantwortung der ATMARP-*Requests* von allen IP-Knoten des LIS verantwortlich.

Für den korrekten Betrieb müssen in jedem Client innerhalb des LIS folgende Parameter konfiguriert werden: seine eigene ATM-und IP-Adresse, die Subnetzmaske und die ATM-Adresse des ATMARP-Servers. Diese Parameter müssen entweder bei jedem Client manuell konfiguriert werden oder können automatisch mittels ILMI[1] übermittelt werden.

Der Server selbst baut keine Verbindungen auf. Er wartet auf die Clients, die nach dem Neustart – sobald ihre ATM-Schnittstelle aktiviert wird – versuchen, sich beim ihm zu regi-

[1] ILMI (*Interim Local Management Interface*) ist ein Bestandteil der ATM-Forum UNI 3.1. Es benutzt das *Simple Network Management Protocol* (SNMP) zur Verwaltung und Überwachung der UNI-Schnittstelle. Im Rahmen des SNMP können zwischen *Switch* und Endsystem elementare Informationen ausgetauscht werden, wie z.B. die ATM-Adresse. ILMI benutzt einen festen, dafür reservierten virtuellen Kanal (VPI=0, VCI=16).

strieren. Dazu baut ein Client eine ATM-Verbindung zum Server auf und übergibt ihm dabei seine ATM-Adresse.

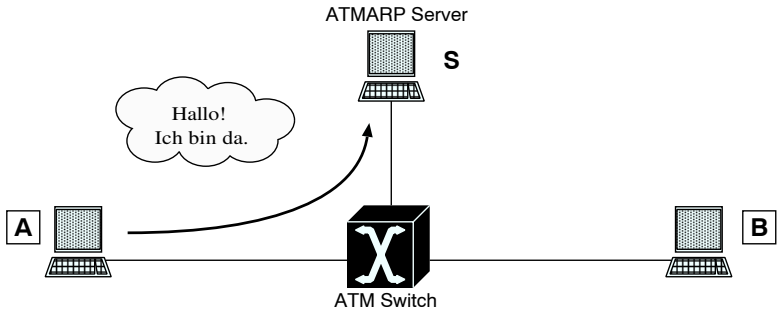


Bild 3.5: Verbindung zum ATMARP-Server

Sobald der Server eine Verbindung von einem neuen LIS-Client entdeckt, schickt er ihm einen InATMARP-*Request*, um seine IP-Adresse zu erfahren.

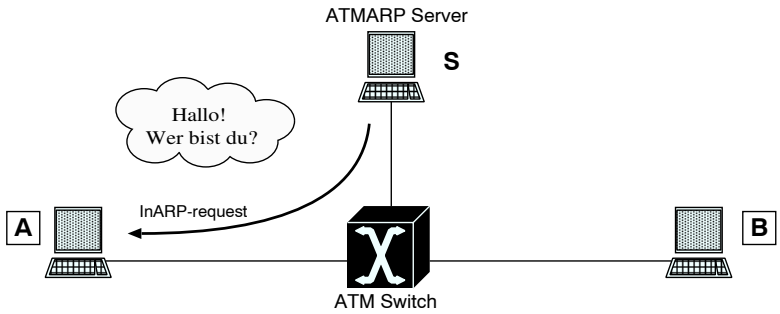


Bild 3.6: ATMARP-Server schickt einen InATMARP-*Request*

Der Client antwortet darauf mit einem InATMARP-*Reply* und teilt dem Server seine eigene IP-Adresse mit. Erhält der Server die neuen Informationen, dann aktualisiert er seine lokale Tabelle.

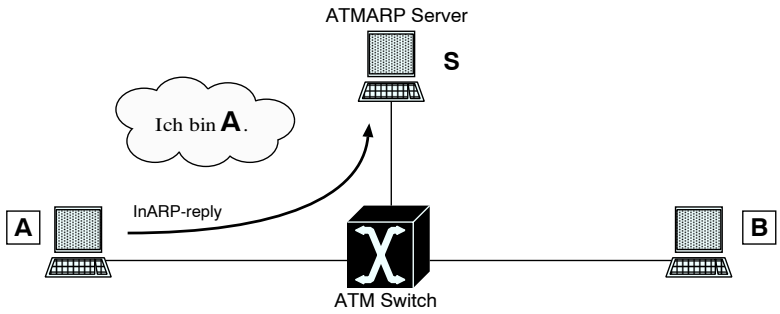


Bild 3.7: Der Client schickt einen InATMARP-*Reply*

Falls jetzt A mit B kommunizieren will, aber nur dessen IP-Adresse kennt, dann fragt er den Server mit einem ATMARP-*Request* nach der ATM-Adresse von B.

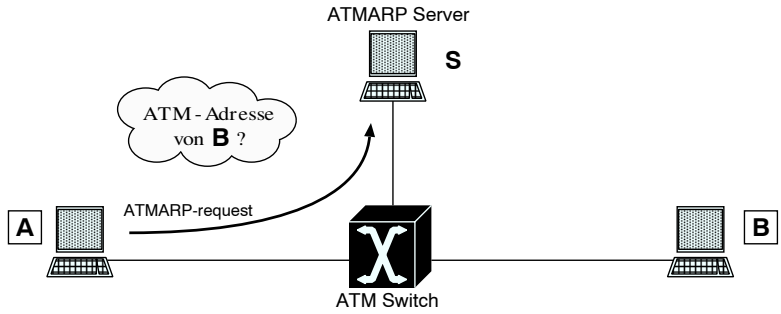


Bild 3.8: Der Client schickt einen ATMARP-*Request*

Wurde B früher korrekt registriert und gibt es einen Eintrag für B in der ATMARP-Tabelle des Servers, dann antwortet er mit einem ATMARP-*Reply*, in dem die gewünschte ATM-Adresse von B übermittelt wird.

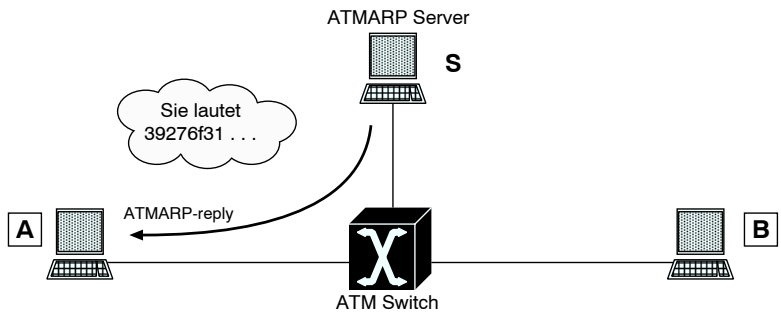


Bild 3.9: Der Server schickt einen ATMARP-*Reply*

Der Client speichert die erhaltene Information in seiner lokalen Tabelle ab und baut mittels UNI-Signalisierung eine ATM-Verbindung zu B auf (siehe Kapitel 3.4 auf Seite 36).

Ein Eintrag in der lokalen ATMARP-Tabelle des Clients ist nur maximal 15 Minuten gültig und muß regelmäßig verifiziert werden. Falls dabei zu dem Eintrag keine Verbindung existiert, wird er gelöscht. Andernfalls schickt der Client ein ATMARP-*Request* zum Server und setzt die *Timeout*-Information beim Empfang von ATMARP-*Reply* zurück.

Unterhält ein Client keine Verbindung mehr[2] zum ATMARP-Server, dann muß er mindestens alle 20 Minuten seine Angaben beim Server erneuern. Dazu baut er eine Verbindung zum Server auf und tauscht mit ihm die InATMARP-Nachrichten aus.

PVC-Umgebung. Alle IP-Knoten, die mit PVCs arbeiten, müssen das InATMARP unterstützen. Jedem PVC muß die IP-Adresse des Systems am anderen Ende der Verbindung

[2] Im späteren RFC1755 [9] wurde auf die Benutzung eines *inactivity timers* hingewiesen, um Verbindungen abbauen zu können, die eine bestimmte Zeitlang nicht mehr benutzt wurden. Ein *inactivity timer* muß z.B. bei Endsystemen implementiert werden, die an einem ATM-Public-UNI arbeiten.

zugeordnet werden. Um das zu bewerkstelligen, fragt der Client mit einem InATMARP-*Request* die andere Station nach ihrer IP-Adresse und trägt die Information in seiner lokalen ATMARP-Tabelle ein. Kommt von einer anderen Station ein InATMARP-*Request*, dann wird er vom Client mit einem InATMARP-*Reply* beantwortet. In einem LIS, in dem ausschließlich PVCs benutzt werden, ist kein ATMARP-Server notwendig, denn wie oben beschrieben, tauschen die IP-Stationen die nötigen Informationen selbst untereinander aus. Für den korrekten Betrieb ist es nicht unbedingt notwendig, daß die ATM-Adressen bekannt sind. Eine Abbildung der IP-Adressen auf die existierenden PVCs ist in diesem Fall ausreichend. Die entsprechenden Felder der InATMARP-Nachrichten, die die Länge der ATM-Adresse angeben, müssen dann auf Null gesetzt werden.

Wie in der SVC-Umgebung, müssen auch bei Festverbindungen die Einträge in der lokalen ATMARP-Tabelle des Clients in regelmäßigen Abständen auf ihre Korrektheit überprüft werden. Der Client verifiziert dafür mit einem InATMARP-*Request* die IP-Adresse des Systems am anderen Ende der Verbindung und aktualisiert entsprechend den Tabelleneintrag.

Arbeitsweise eines ATMARP-Servers

Die Verbindungen zum ATMARP-Server werden während der Initialisierungsphase von der Client-Seite aus aufgebaut. Ist der Verbindungsaufbau erfolgreich abgeschlossen und benutzt die neue *Virtual Connection* (VC) die LLC/SNAP-*Encapsulation*, so schickt der Server einen InATMARP-*Request*, um die IP-Adresse des neuen Client zu erfahren. Der Client antwortet darauf mit einem InATMARP-*Reply*, auf den der Server auf folgende Weise reagiert:

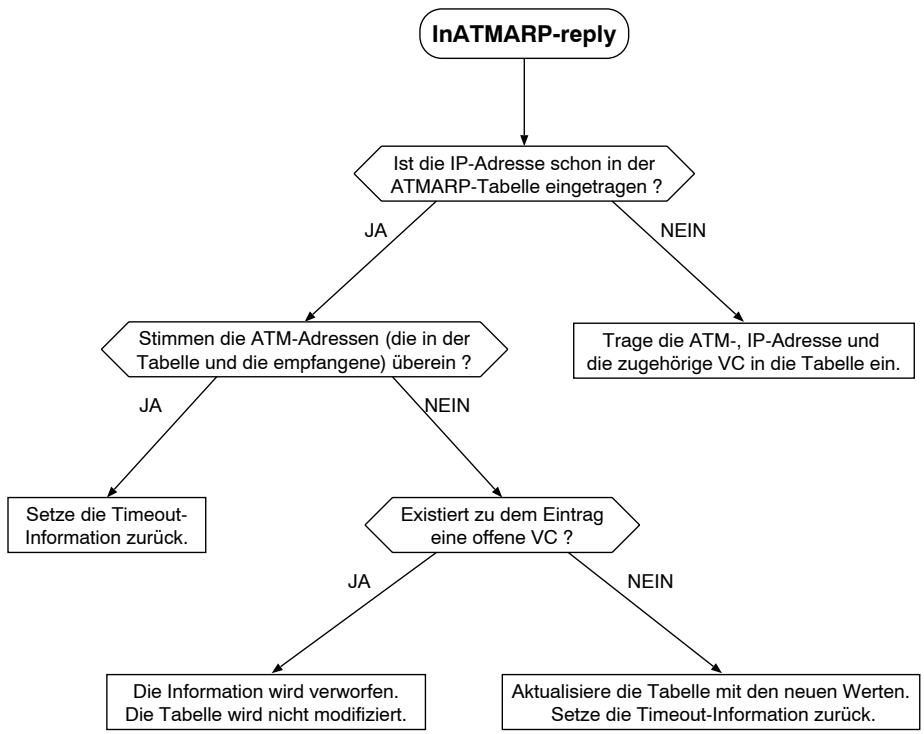


Bild 3.10: Reaktion des Servers auf einen InATMARP-*Reply*

Empängt später der Server einen ATMARP-*Request* von einem Client, der eine ATM-Adresse zu einer bestimmten IP-Adresse braucht, dann reagiert er wie in Bild 3.11 dargestellt. Er schickt entweder einen ATMARP-*Reply* oder ein ATMARP-NAK (einen negativen ATMARP-*Reply*). Die ATMARP-NAK-Antwort ist eine Erweiterung in dem ATMARP-Protokoll und verbessert die Stabilität des ATMARP-Server-Mechanismus. Sie hilft dem Client zu unterscheiden, ob der Server abgestürzt ist oder der entsprechende Eintrag in der ATMARP-Tabelle nicht vorhanden ist. Auch die mit unterbrochenen Linien gekennzeichneten Aktionen verbessern die Robustheit des ATMARP-Mechanismus und können bei Bedarf implementiert werden.

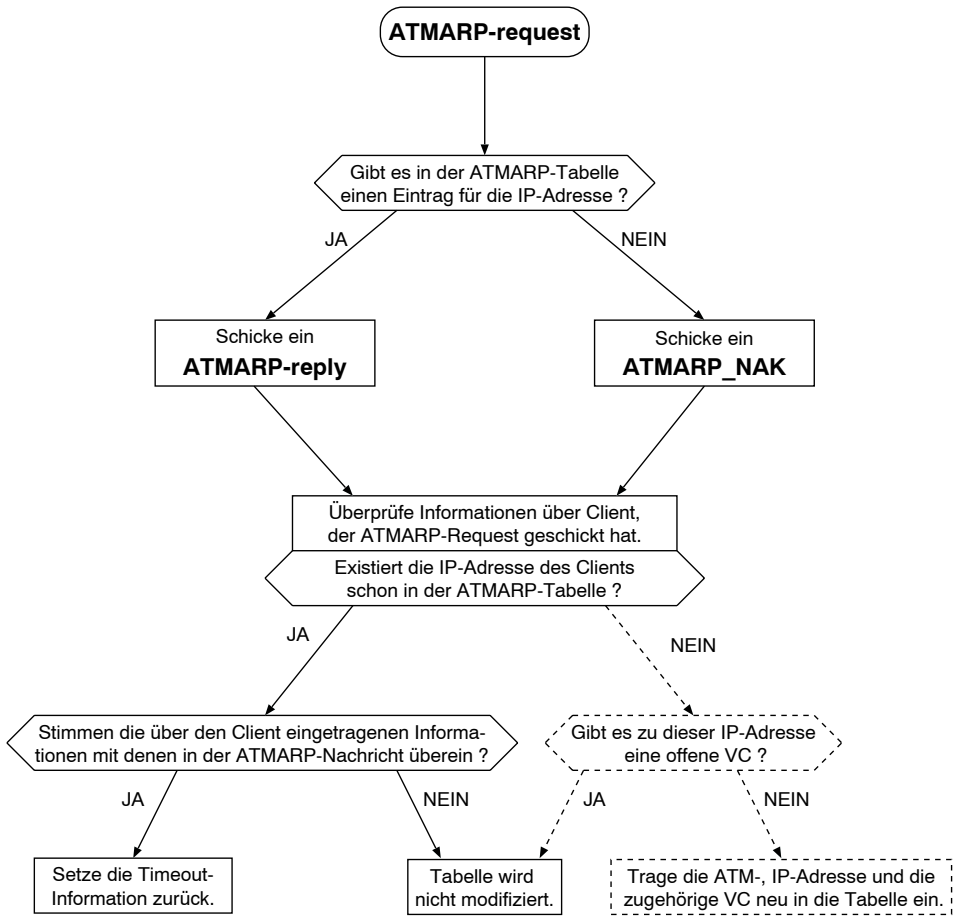


Bild 3.11: Reaktion des Servers auf einen ATMARP-*Request*

Ein Eintrag in der ATMARP-Tabelle des Servers ist mindestens 20 Minuten gültig. Bevor er ungültig wird, muß er verifiziert und die *Timeout*-Information aktualisiert werden. Der Server überprüft, ob der entsprechende Client noch erreichbar ist, indem er ihm ein InATMARP-*Request* schickt. Antwortet der Client mit einem InATMARP-*Reply*, wird die *Timeout*-Information zurückgesetzt und der Eintrag nicht gelöscht. Existiert dagegen keine Verbindung zum Client, wird der entsprechende Eintrag gelöscht. Empfängt der Server einen

ATMARP-*Request* von einem Client, dann wird die *Timeout*-Information ebenfalls zurückgesetzt, wenn die über den Client empfangenen Informationen mit denen in der Tabelle übereinstimmen (vgl. Bild 3.11).

Format und Verpackung der ATMARP-/InATMARP-Pakete

Das in RFC 1577 definierte Format der ATMARP- und InATMARP-Pakete ist in Bild 3.12 dargestellt. Die Felder: ar$hrd (Hardwaretyp), ar$pro (Protokolltyp) und ar$op (Operationscode) haben das gleiche Format und stehen an der gleichen Position wie in den ARP-/InARP-Paketen. Im Unterschied zu ARP wird hier ein zusätzlicher Operationscode für ARP-NAK benutzt. Für die ATM-Forum-Adressen wird der Wert des Hardwaretyps in RFC 1340 [13] festgelegt.

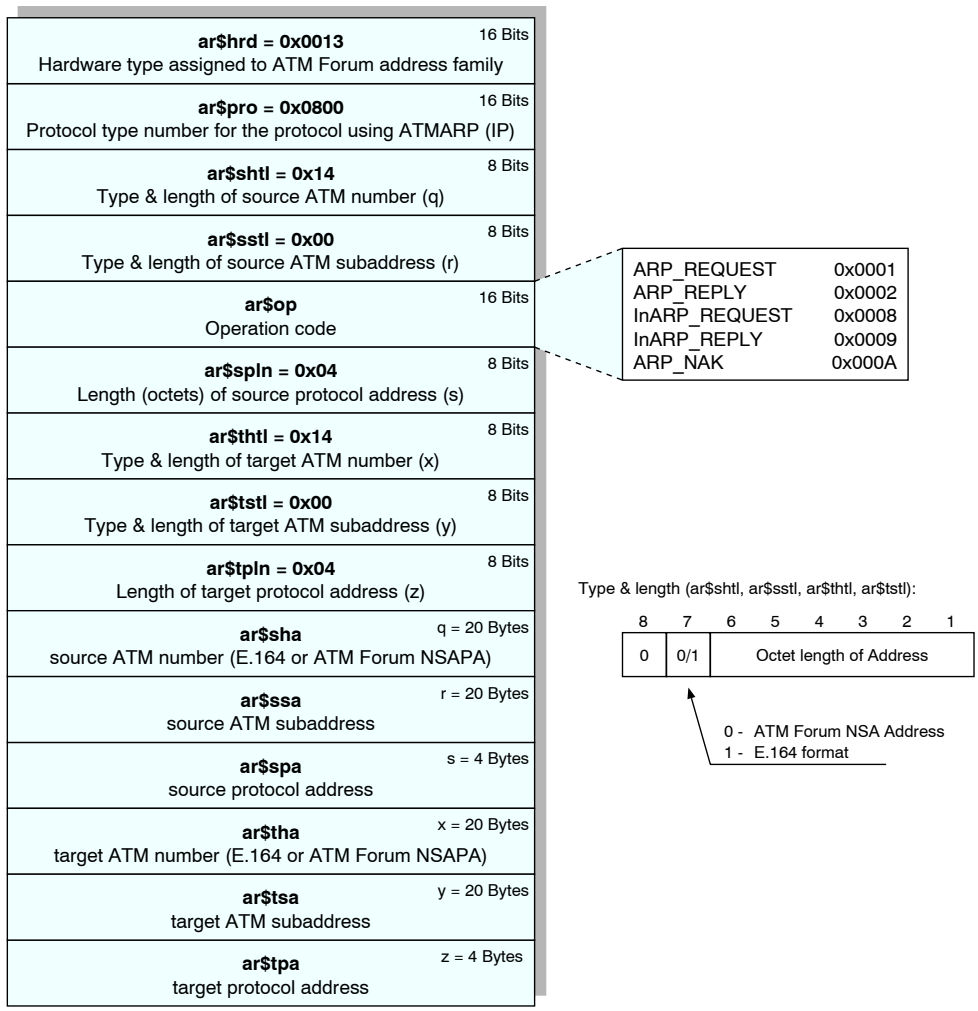


Bild 3.12: Format der ATMARP- und InATMARP-Pakete

Die ATM-Adreßinformation, die in den Signalisierungsnachrichten zur Identifizierung der Sendestation benutzt wird, besteht aus zwei Teilen: '*Calling Party Number Information Ele-*

ment' und *'Calling Party Subaddress Information Element'* (siehe in Kap. 3.4 auf Seite 42). Diese *Information Elements* (IE) sollten direkt in Beziehung zu den Feldern ar$sha (*source ATM number*) und ar$ssa (*source ATM subaddress*) stehen. Weiterhin werden in der UNI-Spezifikation zwei weitere IEs für die Zieladresse definiert: *'Called Party Number'* und *'Called Party Subaddress'*. Diese sollten den Feldern ar$tha (*target ATM number*) und ar$tha (*target ATM subaddress*) entsprechen.

Es gibt grundsätzlich zwei ATM-Adreßformate: E.164 für öffentliche und das NSAP-Format[3] für private Netze (mehr dazu weiter unten im Abschnitt 'ATM Adressen'). Für die Kombination aus ATM-*Number* und ATM-*Subaddress* definiert das ATM-Forum drei Strukturen, die diese zwei Formate unterstützen:

	ATM *Number*	ATM *Subaddress*
Struktur 1	ATM Forum NSAPA	null
Struktur 2	E.164	null
Struktur 3	E.164	ATM Forum NSAPA

IP-Stationen benutzen zum Registrieren ihrer ATM-Adresse beim ATMARP-Server die Struktur, die ihrem ATM-Netzwerkanschluß entspricht. Wird für die ATM-Adresse die Struktur 1 oder 2 verwendet, dann muß in den ATMARP-/InATMARP-Nachrichten die Länge der ATM-Subadressen auf Null gesetzt werden. Die Felder ar$ssa und ar$tsa sind in diesem Fall nicht vorhanden.

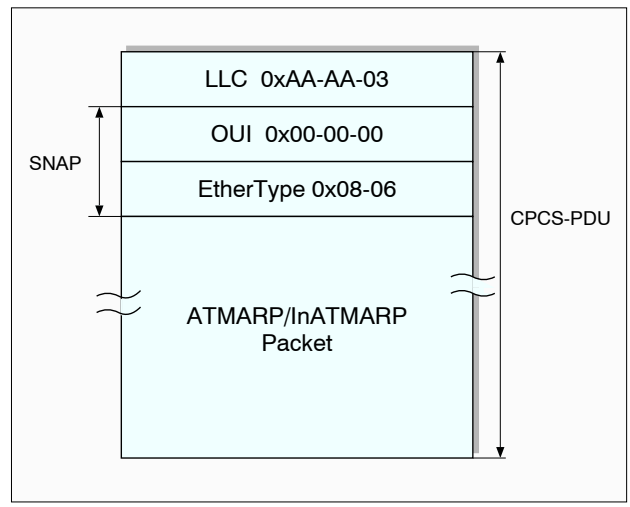


Bild 3.13: Verpackte ATMARP-Nachrichten

ATMARP-/InATMARP-Nachrichten werden in AAL5-PDUs mit dem LLC/SNAP-*Header* verpackt, so wie es Bild 3.13 zeigt. Der Ethernet-Type hat in diesem Fall, wie in RFC 1340 [13] definiert, den Wert 0x08-06. Die LLC/SNAP-Verpackung der ATMARP-/InATMARP-Nahrichten stimmt mit der in RFC 1483 [12] spezifizierten *'Multiprotocol Encapsulation over ATM Adaptation Layer 5'* überein (siehe in Kap. 3.1 auf Seite 24).

ATM Adressen

In den öffentlichen ATM-Netzen werden E.164 Adressen verwendet, deren Format von ITU-T definiert ist. E.164 spezifiziert das Nummernsystem, das innerhalb von ISDN zum Einsatz kommt. In ATM-Netzen wird das internationale Nummernformat benutzt. Diese Nummern können bis zu 15 Ziffern enthalten.

Für private ATM-Netzwerke werden die Adreßformate vom ATM-Forum definiert und basieren auf der Syntax von OSI-NSAP-Adressen. ATM-NSAP-Adressen sind 20 Bytes lang.

Es werden drei Formate für NSAP-Adressen definiert (siehe Bild 3.14). Alle drei bestehen prinzipiell aus drei Elementen: *Authority and Format Identifier* (AFI), *Initial Domain Identi-*

[3] NSAP: *Network Service Access Point (OSI)*

fier (IDI) und *Domain Specific Part* (DSP). Die drei Formate werden aufgrund der Belegung der Felder AFI und IDI unterschieden:

- DCC-Adreßformat
  IDI enthält den *Data Country Code* (DCC), der das Land kennzeichnet, in dem die Adresse registriert ist.

- IDC-Adreßformat
  IDI enthält den *International Code Designator* (ICD), durch den internationale Organisationen identifiziert und somit einheitlich adressiert werden können.

- *NSAP Encoded E.164* Format
  In diesem Fall enthält IDI eine E.164 Nummer.

Die NSAP-Adressen bestehen also aus einem 13 Bytes langen Netzwerk-Präfix und einem 7 Bytes langen Endsystemteil. Letzterer besteht aus einem 6 Bytes großen *End System Identifier* (ESI) – häufig eine MAC-Adresse – und einem 1 Byte großen Selektor (SEL). Der Selektor wird z.B. für die Adressierung mehrerer virtueller Schnittstellen verwendet, die wiederum z.B. in einem Router gebraucht werden, falls er als Station in mehreren LIS konfiguriert wird.

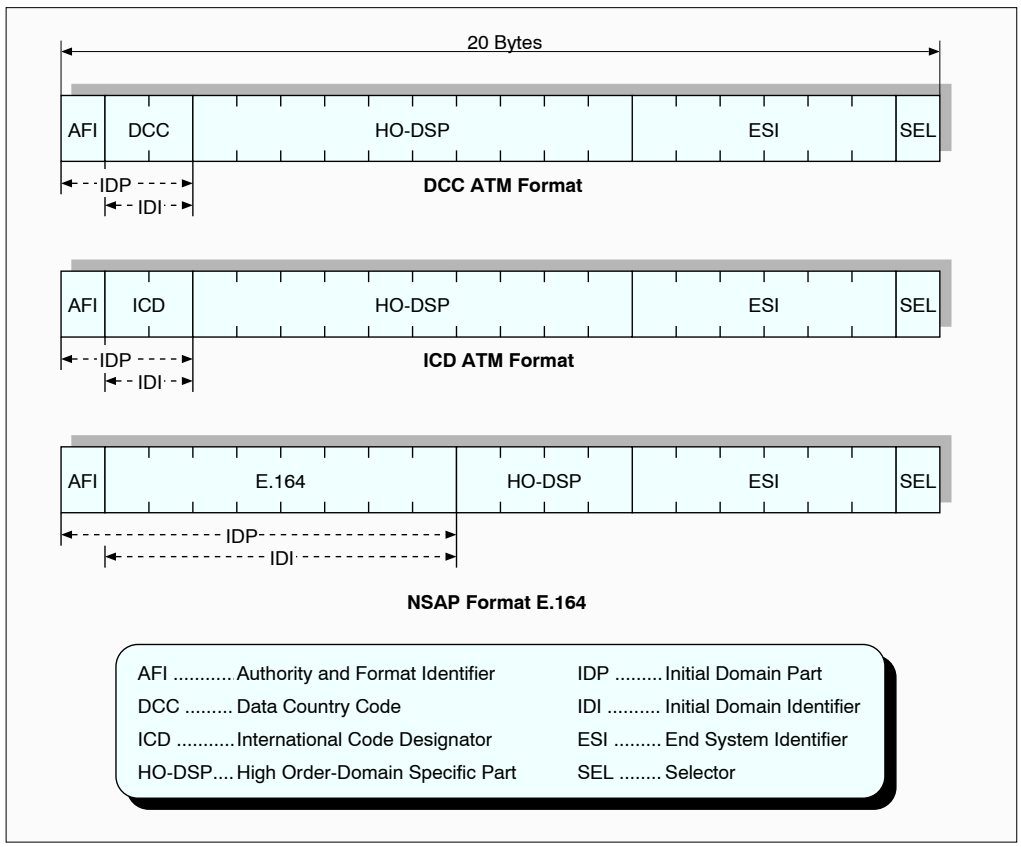


Bild 3.14: Die Adressformate in privaten ATM-Netzen

3.3 RFC 1626: Maximale Paketgrößen für IP über AAL 5

Erfahrungen mit verschiedenen Applikationen, die auf TCP/IP basieren, zeigen, daß sie effizienter arbeiten, falls das *Internet Protocol* (IP) mit großen maximalen Paketgrößen (*Maximum Transmission Unit, MTU)* arbeitet. Fragmentierung der IP-Pakete ist erfahrungsgemäß unerwünscht. Router arbeiten in der Regel mit größeren Paketen viel schneller, denn die Bearbeitungszeit ist weitgehend unabhängig von der Größe der IP-Pakete. Es gibt also einige gute Gründe, um die MTU-Größe für IP über ATM entsprechend groß zu wählen.

In RFC 1209 wird für IP über SMDS eine MTU-Größe von 9180 Bytes definiert. Es sprach nichts dagegen, die *Default*-MTU-Größe in IP-Netzen über ATM auf denselben Wert festzulegen, was die gemeinsame Kommunikation erleichtert und die Notwendigkeit der Fragmentierung von IP-Paketen reduziert.

Deshalb wird in RFC 1626 *'Default IP MTU for use over ATM AAL5'* [10] spezifiziert, daß die MTU für IP-Pakete, die in ATM-Netzen über AAL 5 transportiert werden, standardmäßig 9180 Bytes groß sein soll. Zusammen mit dem 8 Bytes großen LLC/SNAP-*Header* werden also in einer CPCS-*Payload* über AAL5 9188 Bytes transportiert. Die MTU-Größe kann in der Regel auch anders konfiguriert werden. Dann muß allerdings von jeder IP-Station in *Classical*-IP-Subnetzen die gleiche MTU-Größe verwendet werden. Alle Implementierungen sollen aber wenigsten die *Default*-MTU-Größe unterstützen.

PVC. In Applikationen, die nur PVCs unterstützen, wird die ATM-Signalisierung nicht implementiert. Es soll als Standardeinstellung die MTU-Größe von 9180 Bytes benutzt werden, es sei denn, beide Kommunikationspartner haben sich auf eine andere MTU-Größe geeinigt.

SVC. In einer SVC-Umgebung müssen die Applikationen fähig sein, über die AAL CPCS-SDU-Größe (und somit auch die MTU-Größe) mit Hilfe der ATM-Signalisierung zu verhandeln. Dafür wird das Informationselement '*AAL Parameter*' in den Signalisierungsnachrichten verwendet (siehe Kap. 3.4 auf Seite 41). Wie das Informationselement zu verwenden ist und wie die Stationen darauf reagieren sollen, wird in RFC 1626 [10] detailliert beschrieben.

Weiterhin wird in RFC 1626 auch auf die Verwendung des '*IP Path MTU Discovery*'-Verfahrens hingewiesen, das in RFC 1191 [16] definiert ist und zum Ziel hat, die Fragmentierung der IP-Pakete im Netz zu vermeiden (siehe auch Kap. 5.1.2 auf Seite 69).

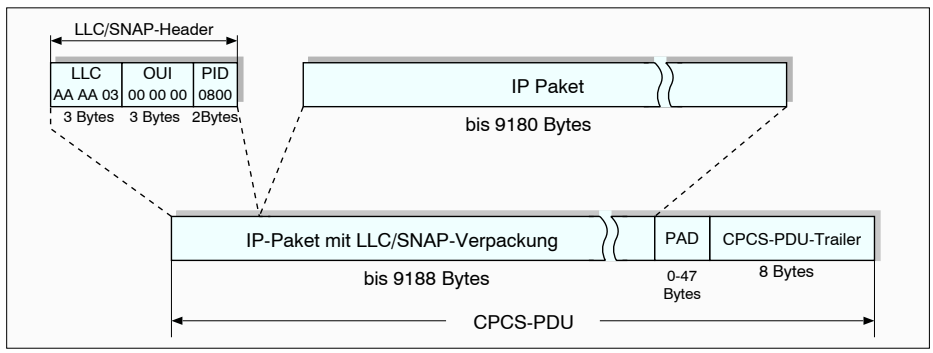


Bild 3.15: Default-MTU-Größe

3.4 Verwendung der ATM-Forum-Signalisierung in Classical-IP-Netzen gemäß RFC 1755

In einer SVC-Umgebung werden die ATM-Verbindungen (*Virtual Channel Connections*, VCCs) nach Bedarf dynamisch auf- und abgebaut. Zu diesem Zweck wird das ATM-Signalisierungsprotokoll verwendet, das zwischen den ATM-Endgeräten und dem ATM-Netz operiert, so wie in der ATM-Forum UNI-Spezifikation [34] definiert.

RFC 1755 *'ATM Signaling Support for IP over ATM'* [9] regelt die Verwendung der UNI-Signalisierung in *Classical*-IP-Netzen. Insbesondere werden Informationen beschrieben, die für die Unterstützung von *Classical*-IP über ATM von Bedeutung sind und während der Signalisierung ausgetauscht werden müssen.

In *Classical*-IP-Netzen über ATM werden während der Signalisierung folgende UNI-Nachrichten verwendet:

Verbindungsaufbau	Verbindungsabbau
SETUP	RELEASE
CALL PROCEEDING	RELEASE COMPLETE
CONNECT	
CONNECT ACKNOWLEDGE	

Verbindungsaufbau. Will Station A mit B kommunizieren und besitzt bereits deren ATM-Adresse, dann versucht sie zunächst mit Hilfe der Signalisierungsnachrichten eine Verbindung zu B aufzubauen. Dazu schickt A zuerst eine SETUP-Nachricht, in der die Übertragungsanforderungen beschrieben sind, die für diese Verbindung verlangt werden. Die Werte für den *Virtual Path Identifier* (VPI) und den *Virtual Channel Identifier* (VCI) werden jedoch vom Netzwerk ausgewählt.

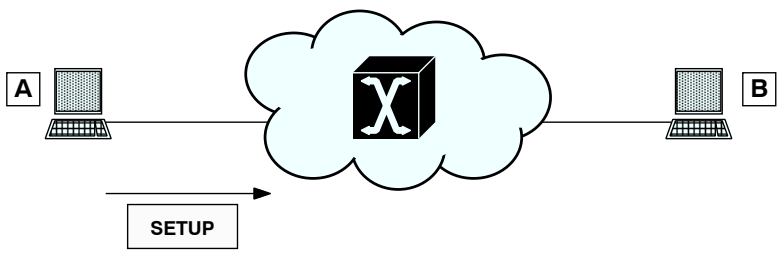


Bild 3.16: Verbindungsaufbau: Sendestation schickt SETUP.

Kann das Netzwerk die angeforderten Dienste oder Verkehrsparameter nicht zur Verfügung stellen, oder die Empfängerstation nicht erreicht werden, so schickt es eine RELEASE-COMPLETE-Nachricht an A und bricht damit den Verbindungsaufbau ab.

Kann die SETUP-Nachricht jedoch vom Netzwerk akzeptiert werden, so wird sie an B weitergeleitet und der Senderstation A kann die Fortsetzung des Verbindungsaufbaus optional mit CALL PROCEEDING mitgeteilt werden.

Die vom Netzwerk an B geschickte SETUP-Nachricht enthält zusätzlich die VPI-/VCI-Werte, die für diese Verbindung von B benutzt werden sollen und nicht verhandelbar sind.

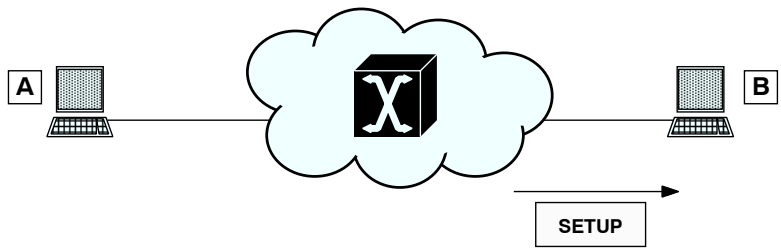


Bild 3.17: Verbindungsaufbau: Netzwerk leitet SETUP an B weiter.

Können die für die Verbindung verlangten Anforderungen – soweit sie B betreffen – erfüllt werden, antwortet B mit einer CONNECT-Nachricht, andernfalls wird der Verbindungsaufbau mit RELEASE-COMPLETE abgelehnt. Auch hier kann optional zuerst CALL-PROCEEDING als erste Antwort auf die SETUP-Nachricht geschickt werden. In der CONNECT-Nachricht beschreiben bestimmte Parameter, daß die gewünschten Übertragungsparameter akzeptiert werden, oder daß sie noch ausgehandelt werden müssen.

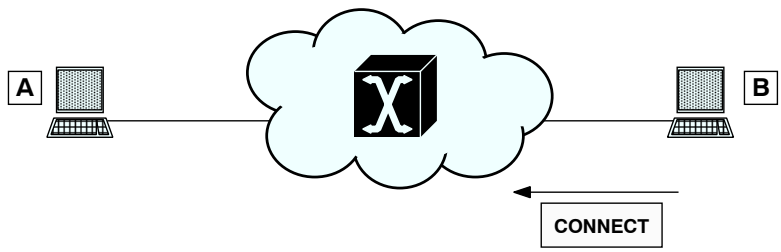


Bild 3.18: Verbindungsaufbau: Empfängerstation schickt ein CONNECT.

Das Netzwerk bestätigt den Empfang der CONNECT-Nachricht mit CONNECT ACKNOWLEDGE und schickt gleichzeitig die CONNECT-Nachricht an die Senderstation A weiter.

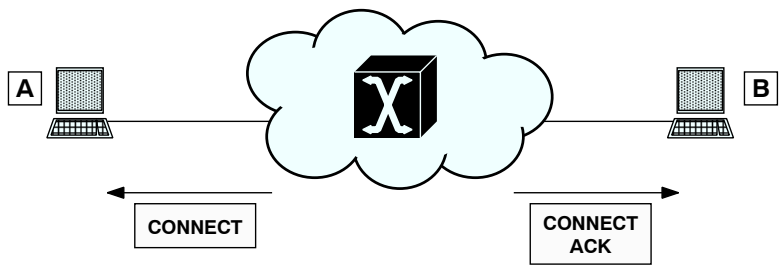


Bild 3.19: Verbindungsaufbau: Netzwerk bestätigt CONNECT und schickt es an A weiter.

Empfängt Station A die CONNECT-Nachricht vom Netzwerk, dann quittiert sie die Nachricht ebenfalls mit CONNECT ACKNOWLEDGE. Damit ist der Verbindungsaufbau abgeschlossen und die Verbindung aktiviert. A und B können jetzt Daten austauschen.

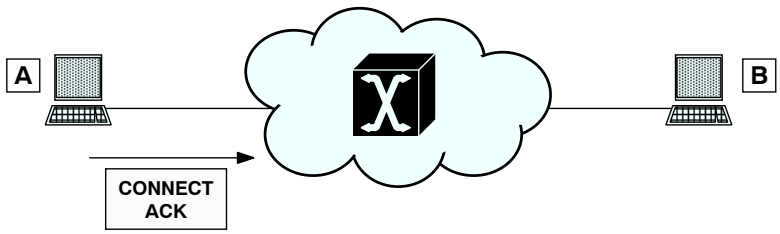


Bild 3.20: Verbindungsaufbau: A bestätigt CONNECT.

Bild 3.21 zeigt den kompletten Nachrichtenaustausch beim Verbindungsaufbau. Die CALL-PROCEEDING-Nachricht ist optional und signalisiert, daß der Verbindungsaufbau initialisiert wurde und keine Informationen mehr für den Verbindungsaufbau benötigt werden. Sie wird als erste Antwort auf die SETUP-Nachricht geschickt, und ihr muß anschließend eine CONNECT- oder RELEASE-Nachricht folgen.

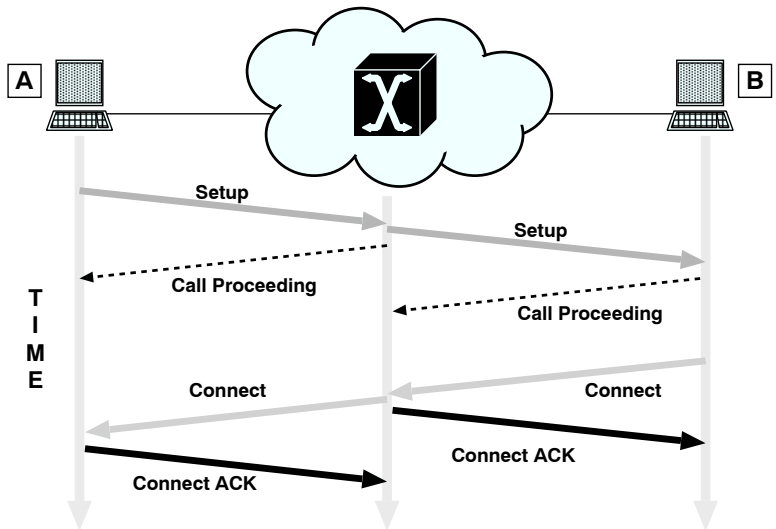


Bild 3.21: Verbindungsaufbau

Verbindungabbau. Will Station A die Verbindung abbauen, dann schickt sie B eine RELEASE-Nachricht.

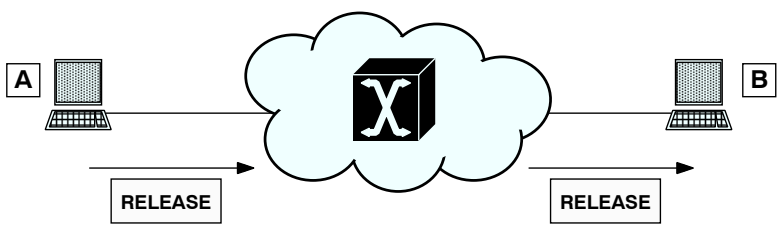


Bild 3.22: Verbindungsabbau: RELEASE

Weil die RELEASE-COMPLETE-Nachricht lokale Bedeutung hat, kann der *Switch* sie direkt zu A schicken, nachdem er die RELEASE-Nachricht von A empfangen hat. Die Verbindung zwischen A und *Switch* wird somit abgebaut.

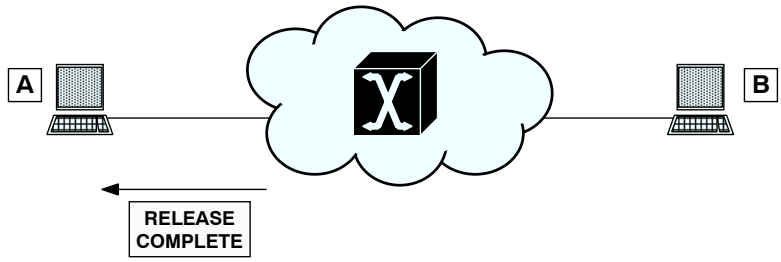


Bild 3.23: Verbindungsabbau: RELEASE COMPLETE

Nach dem Empfang der RELEASE-Nachricht schickt B schließlich eine RELEASE-COMPLETE-Nachricht zum Netzwerk, um die Verbindung zwischen ihnen abzubauen.

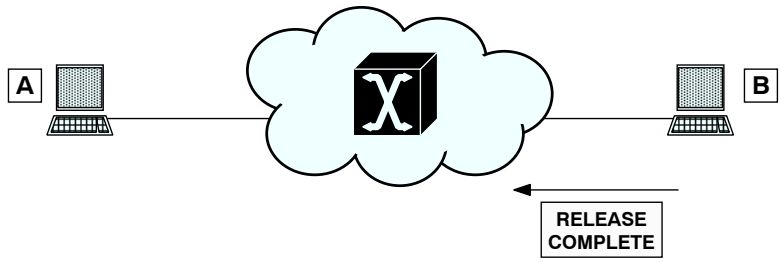


Bild 3.24: Verbindungsabbau: RELEASE COMPLETE

Das folgende Bild zeigt den kompletten Nachrichtenaustausch beim Verbindungsabbau.

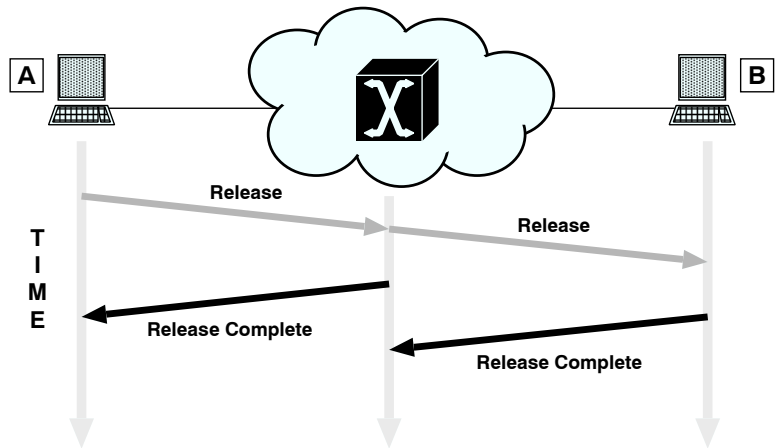


Bild 3.25: Verbindungsabbau

Signalisierungsnachrichten bestehen aus einer Reihe von Informationselementen (*Information Elements*, IEs) variabler Länge. Es wird zwischen Pflicht-IEs (*mandatory IEs*) und optionalen IEs unterschieden. Die IEs sind in Bytes strukturiert, die wiederum in Felder unterteilt sein können. Verschiedene Nachrichtentypen benutzen je nach Bedarf spezifische Informationselemente, um ihre jeweilige Funktion zu realisieren. Der Inhalt der IEs sowie die Verwendung der optionalen IEs hängt von der Art der Verbindung ab. Beim Verbindungsaufbau unterscheiden sich die Signalisierungsnachrichten, z.B. danach, ob die Verbindung für eine Videoanwendung mit konstanter Bitrate über AAL 1 oder für IP über AAL 5 benutzt werden soll.

Zunächst beinhaltet jede Signalisierungsnachricht die vier folgenden Pflicht-Informationselemente (wie in Bild 3.26), die jeweils am Anfang der Nachricht stehen und für alle Nachrichtentypen gleich sind:

- *Protocol discriminator*
- *Call reference*
- *Message type*
- *Message length*

Diesen vier Pflicht-Informationselementen folgen bei verschiedenen Nachrichtentypen jeweils spezifische IEs.

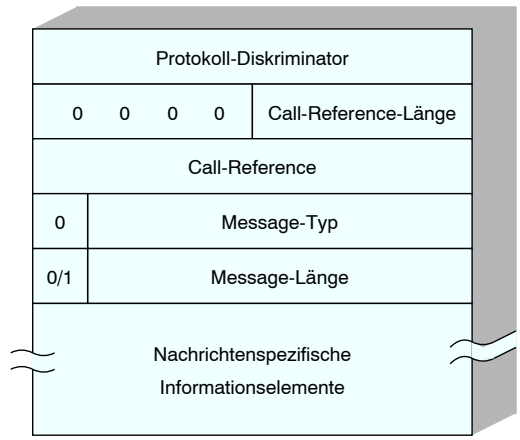


Bild 3.26: UNI-*message*

SETUP

In *Classical*-IP-Netzen werden Verbindungen mit Hilfe der SETUP-Nachricht aufgebaut, die folgende IEs enthält:

Mandatory:

- *AAL Parameter*
- *ATM Traffic Descriptor*
- *Broadband Bearer Capability*
- *Broadband Low Layer Information*
- *QoS Parameter*
- *Called Party Number*
- *Calling Party Number*
- *Connection Identifier*[4]

Optional:

- *Called Party Subaddress*
- *Calling Party Subaddress*
- *Transit Network Selection*

[4] Muß und soll nur in der Network→User Richtung in der SETUP-Nachricht enthalten sein.

Wie man sieht, werden hier die *AAL Parameter*, *Broadband Low Layer Information* und *Calling Party Number* IEs als *mandatory* definiert, im Gegensatz zur UNI-Spezifikation, bei der sie in der SETUP-Nachricht optional sind. Diese IEs enthalten Informationen, die für die *Classical*-IP-Unterstützung von Bedeutung sind, wie z.B. die *Encapsulation*-Methode oder die MTU-Größe.

AAL Parameter

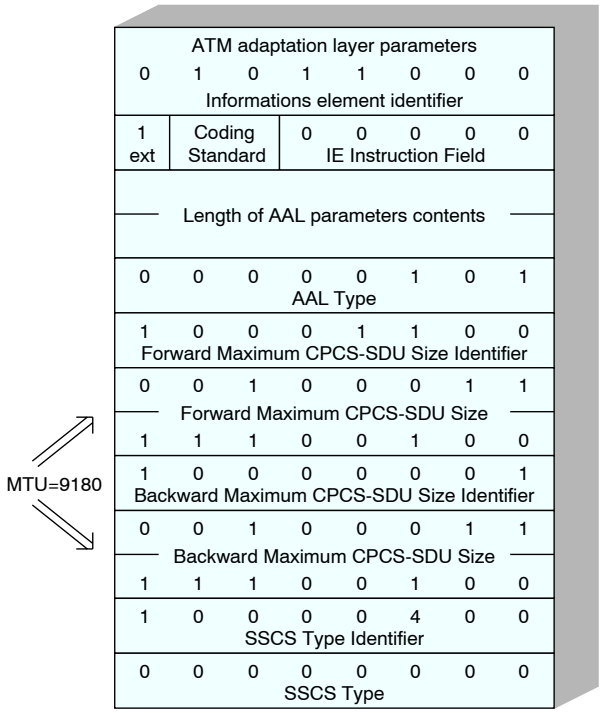


Bild 3.27: AAL Parameter

Das 'AAL Parameter'-IE enthält Informationen über den angefordeten Typ der Anpassungsschicht. Für den Transport von IP-Paketen über ATM wird AAL 5 gewählt, so wie es in RFC 1483 und RFC 1577 definiert ist.

Die Codierung und Verwendung der Felder, die sich auf *Forward and Backward Maximum CPCS-SDU Size* beziehen, wird in RFC1626 [10] beschrieben, weil sie direkt in Beziehung zu der gewünschten MTU-Größe stehen. Es sind Werte bis zu 65535 möglich, obwohl die *Default*-CPCS-SDU-Größe für IP-Netze über ATM auf 9188 Bytes festgelegt ist (8 Bytes LLC/SNAP-*Header* + 9180 Bytes MTU).

Weiterhin wird für das *Classical*-IP keine SSCS-Teilschicht verwendet. 'SSCS-*Type*'-Feld soll also nicht vorhanden sein oder mit Null belegt werden.

Broadband Low Layer Information

Mit Hilfe der B-LLI IEs wird die Verpackungsmethode für die IP-Pakete gewählt. RFC1577 spezifiziert die LLC/SNAP-Verpackung der IP-Pakete als die Standard-Verpackungsmethode in *Classical*-IP-Netzen über ATM. Sie muß von allen Stationen unterstützt werden. Die LLC-Verpackung soll im B-LLI Informationselement im Feld *Layer*-2-Benutzerinformationen wie in Bild 3.28 identifiziert werden. Benutzerinformationen für den *Layer* 3 sollen im B-LLI IE nicht enthalten sein. Es ist die Aufgabe des LLC-*Layers*, auf der Basis des SNAP-*Headers* zu erkennen, zu welchem Protokoll das Paket gehört, um es entsprechend weiterzuleiten.

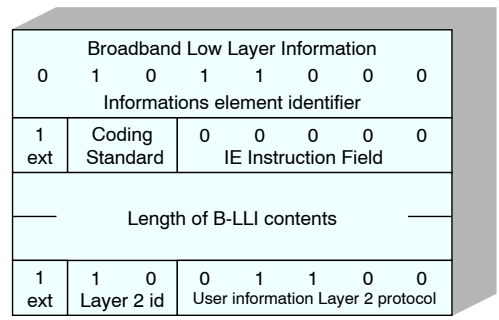


Bild 3.28: B-LLI

ATM Traffic Descriptor

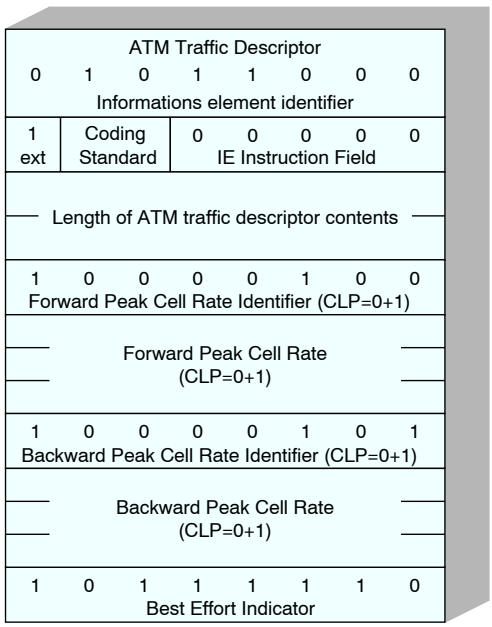


Bild 3.29: *ATM Traffic Descriptor*

Mit dem Informationselement '*ATM Traffic Descriptor*' werden die maximalen Zellübertragungsraten angegeben. Diese Information wird benutzt, um die entsprechenden Betriebsmittel (Bandbreite, Puffer) im Netz zuzuteilen.

Das *ATM-Traffic-Descriptor*-IE wird von *Bradband-Bearer-Capability*-IE und *Quality-of-Service-Parameter*-IE begleitet. Alle drei IEs zusammen spezifizieren die zu verwendende Verkehrs- und Überlast-Steuerung. Die verschiedenen, möglichen Kombinationen dieser drei IEs sind in RFC1755 [9] detailliert beschrieben.

Das IP-Verhalten entspricht am ehesten der *Best-Effort*-Eigenschaft. Falls sie im Netz verfügbar ist, soll sie mit dem *ATM-Traffic-Descriptor*-IE angefordet werden, der den '*Best Effort Indicator*' enthält und die Felder *Forward peak cell rate* (CLP=0+1) und *Backward peak cell rate* (CLP=0+1) spezifiziert.

ATM Adreßinformationen

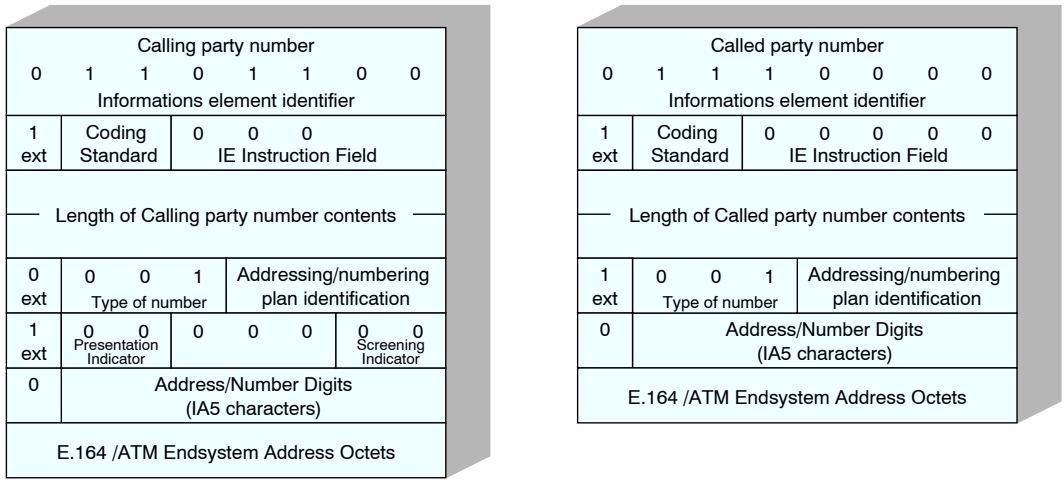


Bild 3.30: ATM Adreßinformationen

In den Informationselementen '*Calling Party Number*' und '*Called Party Number*' sowie unter Umständen auch in '*Calling Party Subaddress*' und '*Called Party Subaddress*' werden die Sende- und Empfängerstationen identifiziert. Mehr zu den ATM-Adressen siehe in Kap. 3.2 auf Seite 33.

Quality of Service

Mittels des Informationselements '*Quality of Service*' wird die Auswahl einer QoS-Klasse getroffen. Die QoS-Klasse ohne Leistungsparameter (*Class* 0) ist die einzige QoS-Klasse, die von allen Netzen unterstützt wird, und die einzige, die erlaubt ist, wenn der *Best Effort Service* benutzt werden soll. Die anderen QoS-Klassen für verbindungsorientierten Datentransfer (*Class* 3) und für verbindungslosen Datentransfer (*Class* 4) können nach der Absprache mit dem Netzbetreiber ebenfalls benutzt werden. Die verschiedenen möglichen Kombinationen der QoS-Parameter mit *ATM Traffic Descriptor* und *Broadband Bearer Capability* findet man in RFC1755[9].

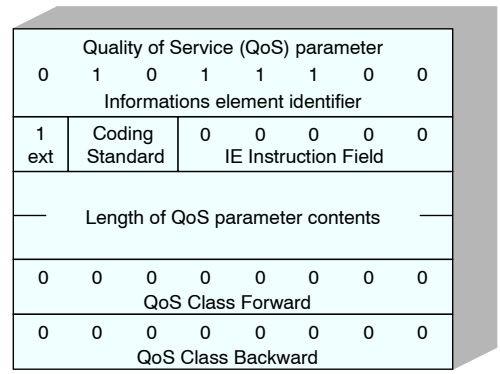


Bild 3.31: ***Quality of Service***

Broadband Bearer Capability

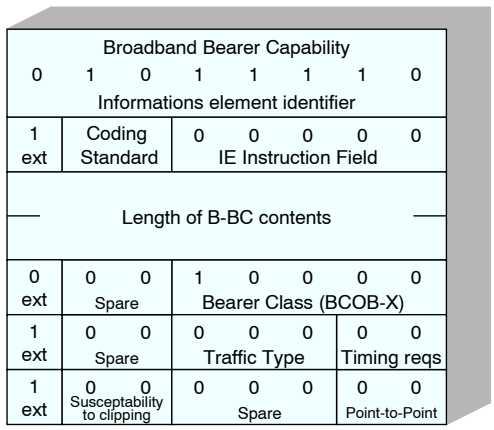


Bild 3.32: ***Broadband Bearer Capability***

In dem Informationselement *'Broadband Bearer Capability'* wird der zu benutzende Netzwerkdienst spezifiziert. Für Multiprotokoll-Verbindungen kann der *Broadband Connection Oriented Bearer Service Type X* (BCOB-X) oder *Type C* (BCOB-C) verwendet werden. Soll BCOB-X verwendet werden und wird *Best Effort* im *ATM-Traffic-Descriptor*-Informationselement identifiziert, so zeigt Bild 3.32 das Format und die Codierung des *Broadband-Bearer-Capability*-IE.

Die verschiedenen *Broadband-Connection-Oriented-Bearer*-Klassen werden in der ITU-T Empfehlung F.811 spezifiziert.

Welcher Netzwerkdienst mit welchen Übertragungsparametern (die in *ATM-Traffic-Descriptor*- und QoS-IEs spezifiziert werden) benutzt werden kann, wird in RFC1755 [9] genau besprochen.

3.5 Classical IP: Einschränkungen und offene Probleme

Classical-IP ist ein einfacher und robuster Ansatz. Er erlaubte als einer der ersten eine direkte Nutzung von IP über ATM und damit die Integration von ATM-Netzwerken in die bestehenden LAN-Umgebungen. *Classical*-IP wird von fast allen Herstellern unterstützt und ermöglicht als standardisiertes Verfahren, in einer ATM-Umgebung Komponenten unterschiedlicher Hersteller zu verwenden. Dies stellt einen Durchbruch in der Nutzung von ATM dar. Trotz der großen Akzeptanz und guten Chancen auf Verbreitung bringt das *Classical*-IP-Modell heute noch viele Einschränkungen mit sich, auf die im folgenden näher eingegangen wird:

- IP-*Broadcasting*
  Das *Classical*-IP-Modell unterstützt keinen *Broadcast*-Mechanismus. Es wird keine Abbildung von IP-*Broadcast*-Adressen auf einen ATM-*Broadcast*-Dienst definiert. Die Vorschläge der IETF zu diesem Thema werden im aktuelen '*IP Broadcast over ATM Networks*'-Draft [27] veröffentlicht.

- IP-*Multicasting*
  In dem *Classical*-IP-Modell wird auch kein IP-*Multicasting* unterstützt. Es wird keine Abbildung von IP-*Multicast*-Adressen auf einen ATM-*Multicast*-Dienst spezifiziert. Die aktuellen Aktivitäten der IETF in diesem Bereich wurden im '*IP Multicasting over ATM*'-Draft [28] vorgestellt.

- Das *Classical*-IP-Modell weist ein *Single-Point-Of-Failure*-Problem auf. Der Ausfall des ATMARP-Servers führt zum Zusammenbruch des Netzes. Mit diesem Problem beschäftigt sich die '*IP over ATM*'-Arbeitsgruppe der IETF. Im aktuellen '*Classical IP and ARP over ATM*'-Draft [25], das die RFCs 1577 und 1626 ablösen soll, werden Lösungsansätze vorgeschlagen, die eine Implementierung von verteilten bzw. redundanten ATMARP-Servern erlauben.

- Wie erwähnt, braucht jeder Client in einem LIS für den Betrieb u.a. die eigene ATM-Adresse und die ATM-Adresse des ATMARP-Servers, der für die Auflösung der IP-Adressen innerhalb dieses LIS zuständig ist. Diese Parameter müssen bei jedem Client manuell konfiguriert werden, was zu einem hohen Konfigurations- und Änderungsaufwand führt (wenn sich z.B. die ATM-Adresse des ATMARP-Servers ändert).
  Abhilfe schafft die ILMI-Spezifikation, die Bestandteil der ATM-Forum UNI 3.1 ist und SNMP zur Verwaltung und Überwachung der UNI-Schnittstelle benutzt. Wenn ILMI implementiert ist, was nicht bei allen Herstellern der Fall ist, kann es vom Client benutzt werden, um vom *Switch* zu erfahren, wie das eigene Netzwerk-Präfix lautet. Bis jetzt wird aber keine Möglichkeit spezifiziert, die es einem Client erlaubt, im Rahmen des ILMI die ATM-Adresse des ATMARP-Servers zu erfahren.
  Auf die möglichen Lösungen wird im aktuellen Internet-Draft '*Classical IP and ARP over ATM*' [25] hingewiesen.

- Eine große mit *Classical*-IP verbundene Einschränkung ist die Beibehaltung der klassischen IP-Subnetz-Architektur. Endsysteme im gleichen Subnetz können direkt kommunizieren. Endsysteme in unterschiedlichen Subnetzen können dagegen nur über einen IP-ATM-Router erreicht werden. Es können keine direkten ATM-Verbindungen über Subnetzgrenzen hinweg etabliert werden. Die Verwendung von Routern führt dabei zu

unnötigem Protokolloverhead und unerwünschten Verzögerungen. Die Zeitransparenz und Skalierbarkeit gehen verloren.
Zwei Ansätze sollen hier in Zukunft Abhilfe schaffen: das '*Next Hop Resolution Protocol*' (NHRP) [29] der IETF und das '*Multi-Protocol over ATM*' (MPOA) [40] des ATM-Forums. Es sollen Verfahren entwickelt werden, die den netzübergreifenden Aufbau von direkten ATM-Verbindungen gestatten. Beide Vorschläge sollen in den nächsten Kapiteln näher erläutert werden.

- Im *Classical*-IP-Modell wird versucht, die Komplexität des ATM-Netzes gegenüber den höheren Schichten zu verbergen. Auf Dauer ist das nicht akzeptabel, weil die Anwendungen dadurch gehindert werden, von den Vorteilen des ATM-Netzes (wie z.B. QoS-Parametern) Gebrauch zu machen.
  Einige dieser Einschränkungen sollen mit dem '*Resource ReSerVation Protocol*' (RSVP) [30] überwunden werden, an dem innerhalb der IETF gearbeitet wird. Mit dem RSVP soll den IP-Anwendungen die Möglichkeit gegeben werden, die spezifischen ATM-Eigenschaften, wie Bandbreitenresevierung, verwenden zu können.

Einen guten Überblick über die Aktivitäten der IETF findet man im *Informational* RFC 1932 '*IP over ATM: A Framework Document*' [8].

Kapitel 4

LAN-Emulation des ATM-Forums

Das Herstellerkonsortium ATM-Forum, das die Interessen der Anwender und Hersteller von ATM-Komponenten im lokalem Bereich vertritt, hat schnell erkannt, daß für die Verbreitung von ATM die Bereitstellung von Möglichkeiten entscheidend ist, die die Weiterbenutzung bestehender Netze und Protokolle erlauben. Deshalb begann das ATM-Forum im Januar 1993, an einer Spezifikation für LAN-Emulation (LANE) zu arbeiten. Die Version 1.0 der '*LAN Emulation over ATM*'-Spezifikation wurde schließlich im Januar 1995 veröffentlicht.

Ein *Local Area Network* (LAN) ist ein lokal begrenztes, privates Netz. Die an einem LAN angeschlossenen Stationen teilen sich das gemeinsame Übertragungsmedium und somit auch die Bandbreite. Die Kommunikation innerhalb klassischer LANs findet verbindungslos statt und basiert auf der Rundsendung aller Nachrichten (*Broadcasting*). Alle angeschlossenen Stationen hören zu und nehmen das gesendete Datenpaket an, falls sie die eigene oder eine *Broad-/Multicast*-Adresse erkennen.
In einem LAN-Protokoll wird die Schicht 2 in zwei Teile unterteilt:

- *Medium Access Control* (MAC)
  Die MAC-Teilschicht regelt den Zugriff auf das gemeinsam genutzte Übertragungsmedium. Hier findet auch die Auswertung der physikalischen Adressen der einzelnen Stationen statt. Die MAC-Teilschicht ist vom verwendeten LAN-Typ abhängig.

- *Logical Link Control* (LLC)
  Die LLC-Teilschicht stellt die eigentlichen Funktionen der Sicherungsschicht bereit. Ihre Hauptaufgabe besteht darin, eine zuverlässige, effiziente Übertragung zwischen zwei direkt benachbarten Stationen zu gewährleisten. Sie arbeitet unabhängig von der darunterliegenden MAC-Teilschicht. Die LLC-Teilschicht wird meistens als Software-Schnittstelle realisiert. Die bekanntesten sind NDIS (*Network Driver Interface Specification*), ODI (*Open Datalink Interface*) und DLPI (*Data Link Provider Interface*).

Die Verbindungen zwischen einzelnen LAN-Knoten werden durch unterschiedliche Übertragungsmedien realisiert. Der Zugriff auf das Übertragungsmedium wird bei diesen verschiedenen LAN-Typen auch unterschiedlich geregelt. Die beiden wichtigsten Zugriffsverfahren sind:

- das Kollisions-Verfahren (z.B. Ethernet: CSMA/CD)

- Zugriff nach erhaltener Sende-Berechtigung (z.B. Token Ring)

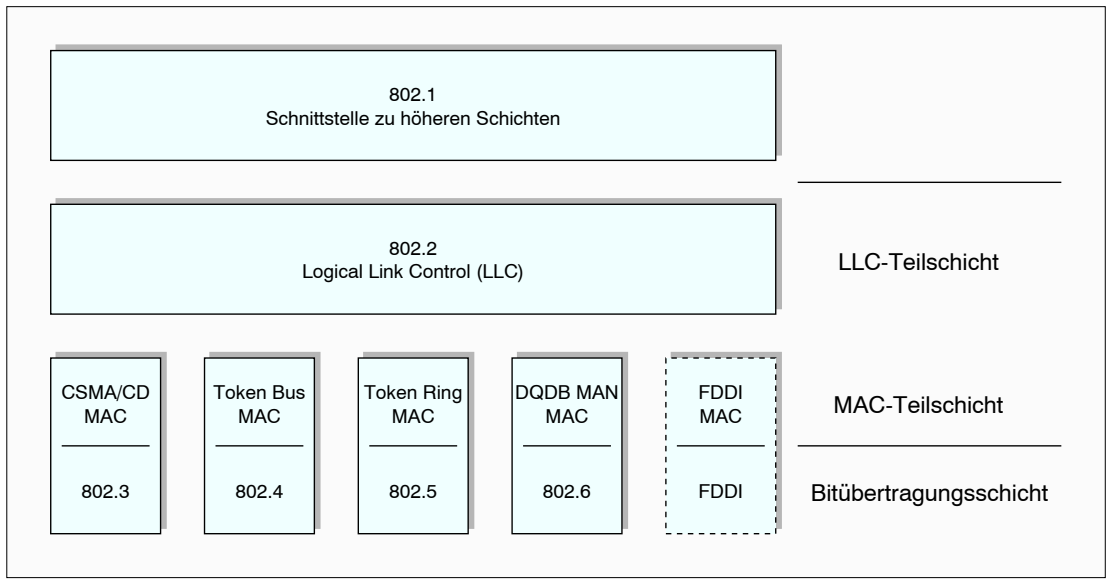


Bild 4.1: IEEE 802-LAN-Festlegungen

Die unterschiedlichen LAN-Konzepte verschiedener Hersteller wurden vom *Institute of Electrical and Electronics Engineers* (IEEE) standardisiert und sind in den IEEE-802-Spezifikationen beschrieben. Eine Übersicht über die LAN-Standards gibt Bild 4.1.

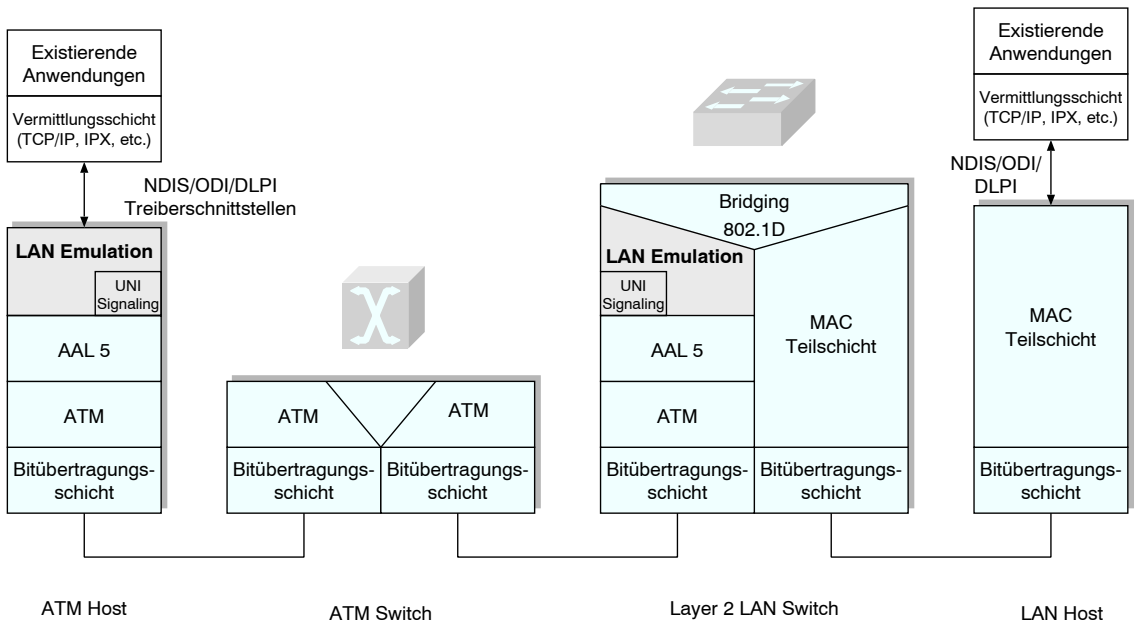


Bild 4.2: LAN-Emulation

4.1 Prinzip der LAN-Emulation

LAN-Emulation ist ein Verfahren, das die einfache Nutzung von ATM durch viele unterschiedliche Protokolle (wie z.B. IP, IPX, AppleTalk) erlaubt, die dabei ohne Anpassungen über das ATM-Netz transportiert werden können. LANE spezifiziert, wie klassische LANs wie Ethernet und Token Ring auf einem ATM-Netz nachgebildet werden können, indem man IEEE-802.3- und IEEE-802.5-LANs auf dem MAC-*Layer* emuliert. In Bild 4.2 wird die Protokollarchitektur der LAN-Emulation dargestellt.

LANE ist protokolltransparent und läßt das ATM-Netz gegenüber den standardisierten Schnittstellen (wie z.B. NDIS, ODI oder DLPI) wie ein Standard-LAN aussehen.

Herkömmlichen Anwendungen wird vorgespiegelt, daß sie über ein konventionelles LAN arbeiten. Es werden die typischen LAN-Dienste wie *Broad-/Multicasting* sowie verbindungsloser Datentransport unterstützt. Die LAN-Anwendungen müssen auf die gewohnte Funktionalität nicht verzichten und können ohne Modifikation über das ATM-Netz weiter benutzt werden.

Die LANE-Spezifikation definiert die Funktionsweise eines einzelnen emulierten LAN. Ein solches *Emulated LAN* (ELAN) kann als ein Segment eines herkömlichen LAN betrachtet werden, wie z.B. eines Ethernet/IEEE 802.3 oder eines IEEE 802.5 (Token-Ring) LAN. Ein ELAN kann mit anderen LANs sowohl über Router als auch über Brücken verbunden werden. Bei Verwendung von Brücken wird sowohl *Transparent Bridging* als auch *Source Route Bridging* unterstützt.

Komponenten
Da sich die Grundeigenschaften von ATM-Netzen und klassischen LANs stark unterscheiden, basiert die LAN-Emulation auf dem Client/Server-Prinzip, bei dem die benötigte Funktionalität emuliert wird. Jedes emulierte LAN (ELAN) besteht also aus einer Reihe von *LAN Emulation Clients* und einem einzelnen *LAN Emulation Service*, der wiederum aus *LAN Emulation Server*, *Broadcast and Unknown Server* und *LAN Emulation Configuration Server* besteht (siehe Bild 4.3):

- *LAN Emulation Client* (LEC, LE-Client)
  LEC ist eine Einheit in Endsystemen des ATM-Netzes, die für den Transfer der Daten, Adreßauflösung sowie andere Kontrollfunktionen zuständig ist. LEC muß in jedem System eines emulierten LAN implementiert werden, sei es Workstation, Router oder LAN-*Switch*. Er wird bei den Endgeräten meistens als Teil der Treibersoftware der ATM-Karte implementiert. Bei Routern oder LAN-*Switches* wird für jedes ELAN, an das sie angeschlossen sind, jeweils eine Instanz des LEC implementiert. Die Hauptaufgabe des LE-Clients besteht darin, den höheren Protokollen die MAC-Schnittstelle zur Verfügung zu stellen. Der LEC wird durch eine 20-Bytes-ATM-Adresse identifiziert und durch eine oder mehrere 6-Bytes-MAC-Adressen charakterisiert, die durch diese ATM-Adresse erreichbar sind.

- *LAN Emulation Server* (LES, LE-Server)
  LES ist die Kontrollstation des emulierten LAN und koordiniert dessen Funktionen. Zu den Hauptaufgaben des LES gehört die Unterstützung der Clients (LECs) bei der Auflösung der Ethernet- oder Token-Ring-MAC-Adressen nach ATM-Adressen (mittels eines *LAN Emulation Address Resolution Protocols*, LE_ARP). Ein LES wird durch seine ATM-Adresse identifiziert. Pro emuliertem LAN darf es nur ein LES geben. Ein LES hat Verbindungen zu allen Engeräten (LECs).

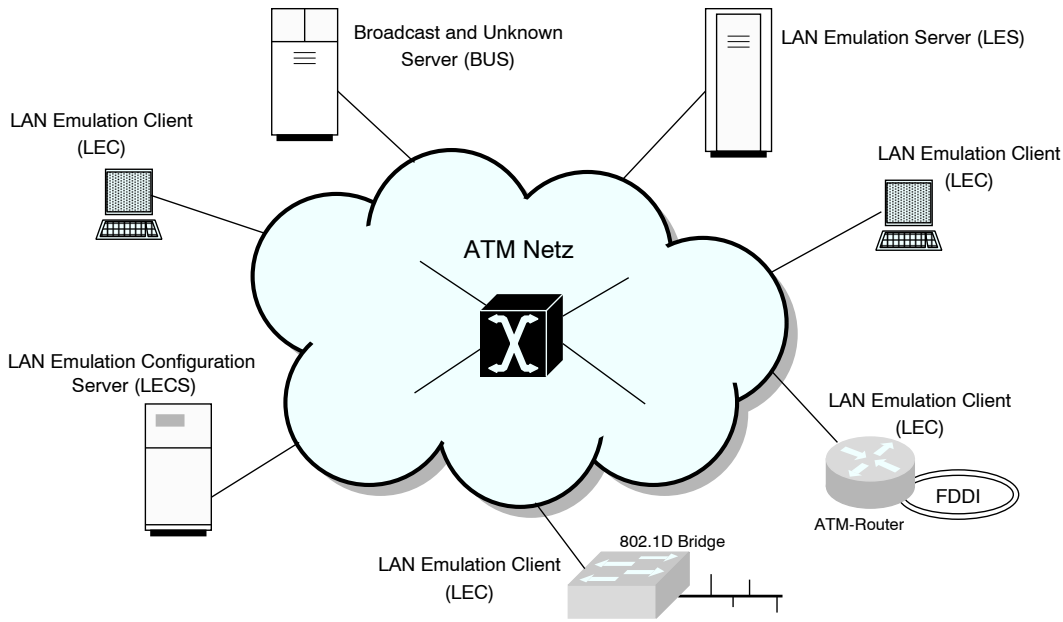


Bild 4.3: Ein emuliertes LAN (ELAN)

- *Broadcast and Unknown Server* (BUS)
  Dem *Broadcast and Unknown Server* kommt eine besondere Bedeutung in einem emulierten LAN zu, weil er die Eigenschaften des klassischen LAN – wie z.B. Verteilen von *Broad-/Multicast*-Paketen – emuliert, die in einem reinen ATM-Netz nicht ohne weiteres unterstützt werden. Außer der Verteilung von *Broad-/Multicast*-Nachrichten ist der BUS für die Zustellung von Paketen verantwortlich, die von einem LEC gesendet wurden, der noch keine direkte ATM-Verbindung zur Zielstation hat. Der BUS wird durch seine ATM-Adresse identifiziert und im LES mit der *Broadcast*-MAC-Adresse (alle Bits 1) verknüpft. Der BUS unterhält genau so wie LES Verbindungen zu allen Endsystemen (LECs) eines emulierten LAN.

- *LAN Emulation Configuration Server* (LECS)
  LECS verwaltet die Konfigurationsinformationen für ein ATM-Netz und unterstützt die Clients (LECs) bei der automatischen Konfiguration. Während der Initialisierungsphase ordnet LECS jeden Client einem bestimmten ELAN zu, indem er ihm die ATM-Adresse des zuständigen LES übergibt. Innerhalb einer Verwaltungsdomäne auf einem ATM-Netz existiert nur ein LECS, der durch seine ATM-Adresse identifiziert wird und für alle emulierten LANs in dieser Domäne zuständig ist.

Auf einem physikalischen ATM-Netz können mehrere emulierte LANs (ELANs) definiert werden. Jedes ELAN arbeitet dann unabhängig von den anderen. Die ATM-Endsysteme – sowohl Clients als auch Server – können dabei als Mitglied in mehreren ELANs definiert werden.

Verbindungstypen

In einem emulierten LAN wird für den korrekten Betrieb eine Reihe von ATM-Verbindungen

(*Virtual Channel Connections, VCCs*) benötigt. LE-Clients unterhalten separate Verbindungen (VCCs) für den Transport von Daten und Kontrollinformationen.
Es werden folgenden Kontrollverbindungen benutzt (siehe Bild 4.4):

- *Configuration Direct VCC*
  Dies ist eine bidirektionale Punkt-zu-Punkt Verbindung, die ein Client zum LECS aufbaut, um von ihm die Konfigurationsinformationen zu erhalten.

- *Control Direct VCC*
  Dies ist eine bidirektionale Punkt-zu-Punkt Verbindung, die vom LEC zum LE-Server während der Initialisierungsphase aufgebaut wird. Sie wird für das Senden von Kontrollinformationen benutzt.

- *Control Distribute VCC*
  Dies ist eine optionale Punkt-zu-Punkt oder Punkt-zu-Mehrpunkt Verbindung, die ein LE-Server zu den LE-Clients während der Initialisierungsphase aufbaut und dann zum Verteilen von Kontrollinformationen benutzt.

Für den Datentransport werden dagegen folgende Verbindungen benutzt (siehe Bild 4.5):

- *Data Direct VCC*
  Dies ist eine bidirektionale Punkt-zu-Punkt Verbindung, die zwischen zwei Clients (LECs) aufgebaut wird, die untereinander Daten austauschen wollen.

- *Multicast Send VCC*
  Dies ist eine bidirektionale Punkt-zu-Punkt Verbindung, die ein LEC zum BUS aufbaut. Über diese Verbindung schickt ein LEC *Multicast*-Pakete, die dann vom BUS entsprechend weiter verteilt werden sollen.

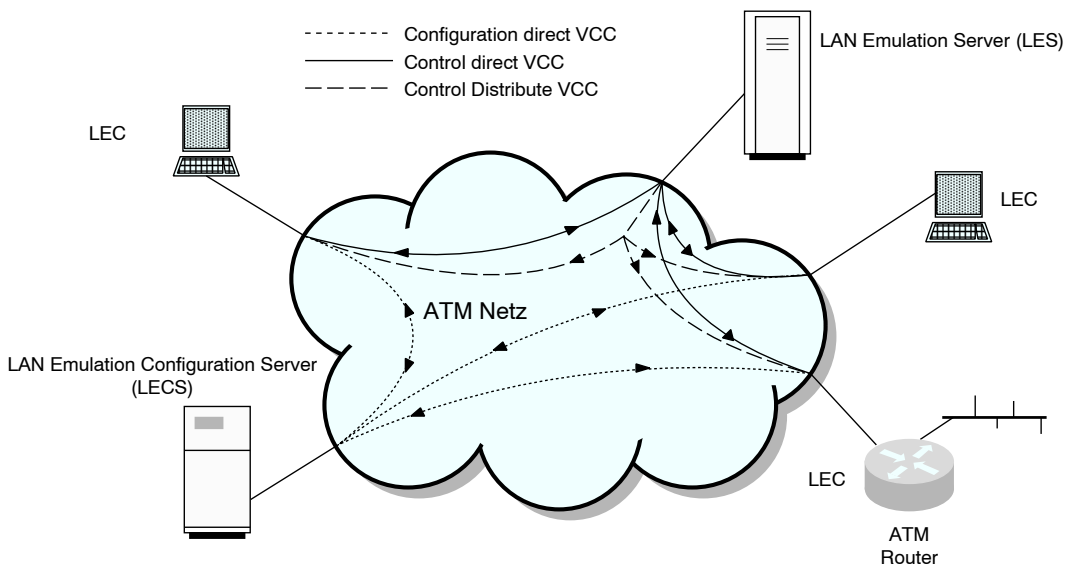


Bild 4.4: LANE: Kontrollverbindungen

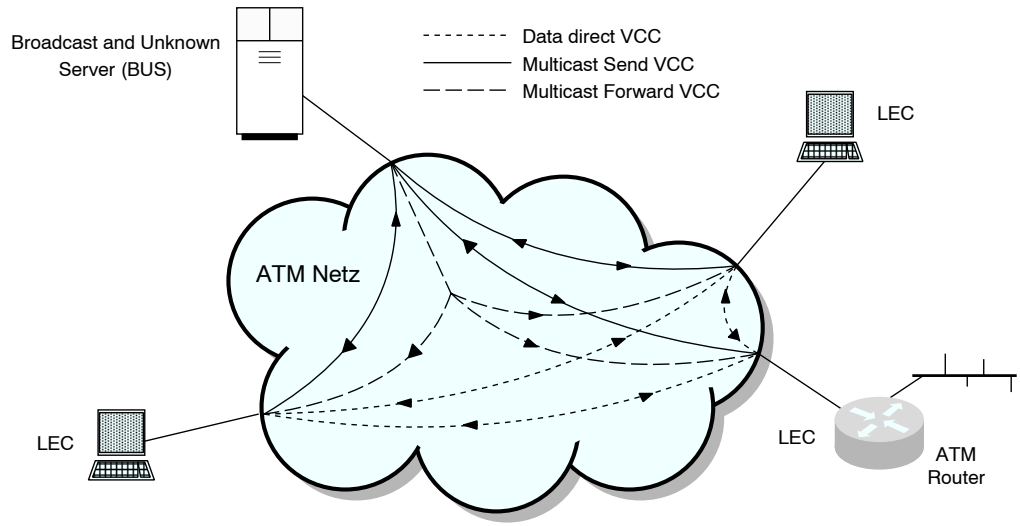


Bild 4.5: LANE: Verbindungen für den Datentransfer

- *Multicast Forward VCC*
  Dies ist eine Punkt-zu-Punkt oder Punkt-zu-Mehrpunkt Verbindung, die der BUS zu den Clients (LECs) aufbaut und dann zur Verteilung von Paketen benutzt.

Arbeitsweise des LANE-Protokolls

Damit ein LE-Client an der LAN-Emulation teilnehmen kann, muß eine Initialisierung durchgeführt werden, die normalerweise keine manuelle Konfiguration bei den Clients erfordert.

Ein Client wird zunächst in ein definierten Anfangszustand versetzt, indem einige Variablen und Parameter initialisiert werden. Was und wie das genau geschieht, hängt von der Implementierung ab.

Danach versucht ein Client in der LECS-Verbindungsphase eine Kontrollverbindung (*Configuration Direct VCC*) zu seinem *LAN Emulation Configuration Server* (LECS) aufzubauen.

Wenn die Verbindung aufgebaut ist, teilt der Client in der Konfigurationsphase dem LECS seine ATM-Adresse mit und erhält von ihm einige Informationen, die er benötigt, um an der LAN-Emulation teilzunehmen. Dazu gehört vor allem die ATM-Adresse des zuständigen *LAN Emulation Server* (LES).

Hat der Client alle nötigen Informationen vom LECS erhalten, so baut er während der *Join*-Phase eine Kontrollverbindung (*Control Direct VCC*) zum LES auf und meldet sich bei ihm an. Der LES identifiziert den Client und weist ihm einen *LEC Identifier* (LECID) zu, der den Client eindeutig innerhalb des emulierten LAN identifiziert. Um die Kontrolldaten zum Client zu senden, benutzt der LES entweder die *Control-Direct-VCC* oder fügt optional den neuen Client an seine Punkt-zu-Mehrpunkt Verbindung (*Control Distribute VCC*) an, die er generell zum Verteilen von Kontrollinformationen benutzt.

Danach kann der Client in der *Initial-Registration*-Phase alle übrigen MAC-Adressen beim Server (LES) registrieren lassen.

Am Ende der Initialisierungsphase baut der Client eine Verbindung (*Multicast Send VCC*) zum *Broadcast and Unknown Server* (BUS) auf, um *Multicast*-Pakete verschicken zu können.

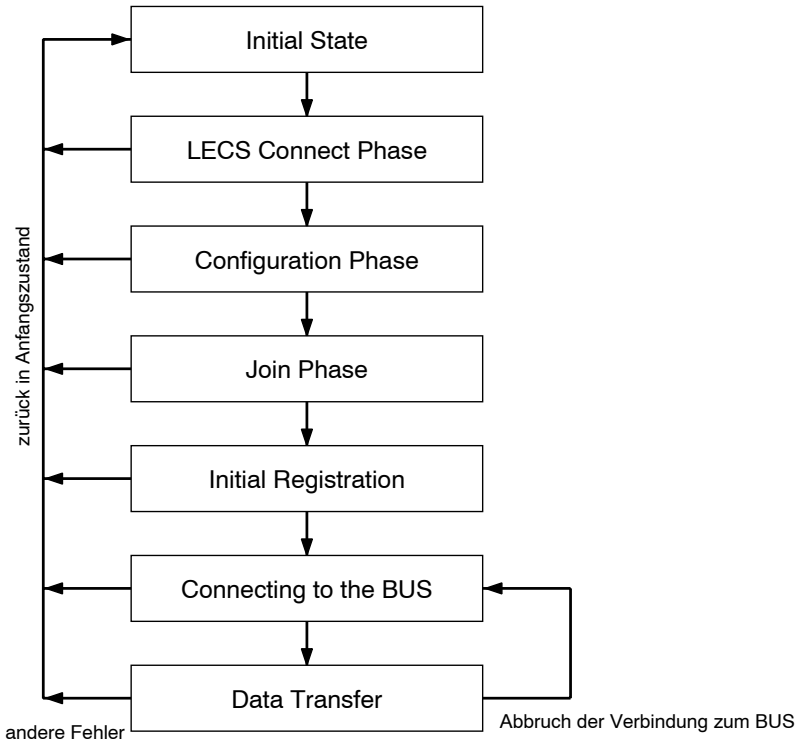


Bild 4.6: Verschiedene Phasen der der LAN-Emulation

Damit der Client auch die *Broadcast und Multicast*-Pakete empfangen kann, wird vom BUS eine Verbindung (*Multicast Forward VCC*) zum Client aufgebaut.

Wurde ein Client in ein emuliertes LAN erfolgreich integriert, kann er Daten mit anderen Clients austauschen. Der Datentransfer zwischen zwei Clients erfolgt im ELAN normalerweise über direkte Punkt-zu-Punkt Verbindungen (*Data Direct VCC*).

Falls eine solche Verbindung zur Zielstation nicht existiert, versucht der Client sie nach einer Adreßauflösung aufzubauen. Bei der Auflösung der MAC-Adresse nach der ATM-Adresse wird der Client vom LE-Server unterstützt.

Solange keine direkte Verbindung zur Zielstation existiert, kann ein Client die Datenpakete zum BUS senden, der sie dann über eine *Multicast-Forward*-Verbindung an alle Clients weiterleitet. *Broadcast*- und *Multicast*-Pakete werden prinzipiell via BUS verschickt.

LANE-Kontrollnachrichten

Das LANE-Protokoll benutzt verschiedene Arten von Kontrollnachrichten, deren Format in Bild 4.7 dargestellt ist. Die meisten Kontrollnachrichten sind 108 Bytes lang. Eine Ausnahme bilden hier zum einen die *Ready_Query*- und *Ready_Ind*-Nachrichten, die 6 Bytes lang sind und nur die ersten vier Felder enthalten (*Marker*, *Protocol*, *Version* und *Op-Code*). Zum anderen können die LE_*Configure*-Nachrichten einige zusätzliche, mit *Type/Length/Value* codierte Parameter (T-L-V) enthalten. Abhängig vom Typ der Kontrollnachricht werden manche Felder nicht benutzt, die dann meistens mit Nullen belegt sind. Die in Bild 4.7 grau dargestellten Felder werden z.B. nur während der *Configuration*- und *Join*-Phase verwendet.

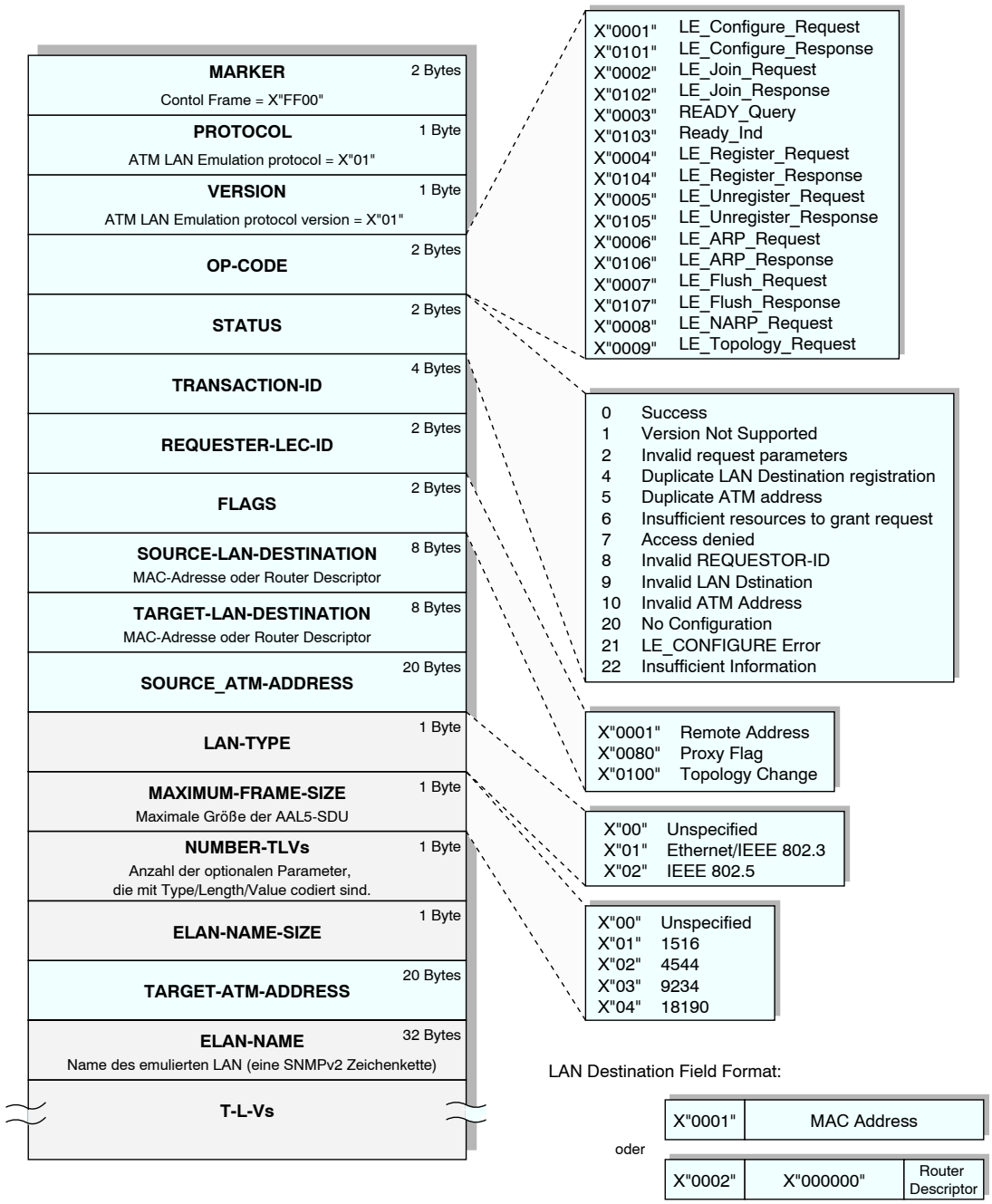


Bild 4.7: Control Frame Format

4.2 LANE-Initialisierungsprotokolle

Die Initialisierung eines LE-Clients wird in einen Anfangszustand (*Initial State*) und fünf Phasen aufgeteilt. Diese fünf Phasen sind *LECS Connect Phase*, *Configuration Phase*, *Join Phase*, *Initial Registration* und *BUS Connect Phase*. Sie müssen[1] nacheinander ausgeführt werden, wie es in Bild 4.6 dargestellt. Nach der *BUS-Connect*-Phase ist der Client betriebsbereit und kann Daten mit anderen Clients austauschen. Kann eine Phase während der Initialisierung nicht erfolgreich abgeschlossen werden oder tritt im normalen Betriebszustand ein ungewöhnlicher Fehler auf, dann muß der Client in den Anfangszustand (*Initial State*) zurückversetzt und die Managementschicht informiert werden.

4.2.1 Initial State

Der Anfangszustand wird in der LANE-Spezifikation nicht genau definiert und hängt von der konkreten Implementierung ab. Es wird lediglich eine Reihe bestimmter Variablen und Parameter spezifiziert, die bei der LAN-Emulation verwendet werden. Für den Client sind das insgesamt 28 (C1 bis C28) und bei den Servern 8 (S1 bis S8) Variablen, von denen die wichtigsten in der folgenden Tabelle aufgelistet sind.

C1	**LE Client's ATM Address.** Dies ist die eigene ATM-Adresse des Clients.
C2	**LAN Type.** Dies ist der Typ des emulierten LAN, an das der Client angeschlossen ist oder wird. Als Typ kann Ethernet/IEEE 802.3, IEEE 802.5 (Token-Ring) oder 'nicht-spezifiziert' (nur bis zur *Join*-Phase) angegeben werden. Er darf nicht geändert werden, ohne den Client zurück in den Anfangszustand zu versetzen.
C3	**Maximum Data Frame Size.** Dies ist die maximale AAL5-SDU-Größe, die der Client für die Verbindungen benutzt. Sie kann zuerst 'nicht-spezifiziert' sein, muß aber spätestens bis zum erfolgreichen Abschluß der *Join*-Phase auf einen der folgenden Werte gesetzt werden: 1516, 4544, 9234 oder 18190 Bytes.
C4	**Proxy.** Idenifiziert, ob der Client für einige (*Remote*)-MAC-Adressen die Rolle des Vertreters spielt.
C5	**ELAN Name.** Dies ist eine 32 Bytes lange SNMPv2-Zeichenkette und wird z.B. beim Netzwerkmanagement für benutzerfreundliche Identifikation des emulierten LAN benutzt.

[1] Einige der Initialisierungsphasen können bei manchen Implementierungen weggelassen werden, falls die Clients z.B. manuell konfiguriert werden.

C6	**Local Unicast MAC Address(es).** Jeder Client wird mit einer oder mehreren MAC-Adressen konfiguriert. Alle Adressen müssen von ihm beim Server (LES) registriert werden. In einem emulierten LAN müssen alle Clients verschiedene MAC-Adressen aufweisen.
C9	**LE Server ATM Address.** Dies ist die ATM-Adresse des zuständigen *LAN Emulation Server* (LES). Sie wird in der Konfigurationsphase festgestellt und beim Etablieren der *Control-Direct*-Verbindung benutzt.
C12	**VCC Time-out Period.** Findet innerhalb dieser Zeit kein Datenaustausch über eine *Data-Direct*-Verbindung statt, wird sie abgebaut.
C14	**LE Client Identifier.** Jeder Client erhält vom LES während der *Join*-Phase einen *LAN Emulation Client Identifier* (LECID), der ihn innerhalb des ELAN eindeutig identifiziert. Er wird z.B. benutzt, um das Echo der eigenen *Multicast*-Pakete zu unterdrücken.
C15	**LE Client Multicast MAC Address(es).** Jeder Client besitzt eine Liste von *Multicast*-Adressen und empfängt alle Pakete die für diese Adressen bestimmt sind. Die *Broadcast*-MAC-Adresse sollte in der Liste enthalten sein.
C16	**LE_ARP Cache.** Dies ist eine lokale Adreßtabelle, in der der Client die Abbildungen der MAC-Adressen auf die ATM-Adressen aufbewahrt.
C27	**Remote Unicast MAC Address(es).** Dies sind die MAC-Adressen, die nicht beim Server (LES) registriert sind, für die aber der *Proxy*-Client den Vertreter spielt und auf die LE_ARP-*Requests* antwortet.

S1	**LE Server's ATM Addresses.**
S2	**LAN Type.** Spezifiziert den Typ des emulierten LAN (entweder auf Ethernet/IEEE 802.3 oder IEEE 80.5 gesetzt).
S3	**Maximum Data Frame Size.** Dies ist die maximale AAL5-SDU-Größe für Pakete, die der LE-*Service* ohne Fragmentieren handhaben kann. Sie beträgt 1516, 4544, 9234 oder 18190 Bytes.
S6	**Broadcast and Unknown Server's Address(es).** Es muß mindestens eine ATM-Adresse für den BUS bekannt sein. Der BUS kann auch mehrere ATM-Adressen besitzen und diese können sogar während des Betriebs hinzugefügt werden.

4.2.2 LECS Connect Phase

In der Regel wird bei den Clients keine manuelle Konfiguration benötigt. Um eine automatische Konfiguration zu ermöglichen, versucht der Client zunächst eine Kontrollverbindung (*Configuration Direct VCC*) zu seinem *LAN Emulation Configuration Server* (LECS) aufzubauen. Falls der Client eine dafür vorkonfigurierte SVC (*Switched Virtual Connection*) oder PVC (*Permanent Virtual Connection*) benutzt, braucht die LECS-*Connect*-Phase nicht ausgeführt zu werden.

Ansonsten kann ein LE-Client den LE-*Configuration*-Server auf verschiedene Weise erreichen, und zwar in folgender Reihenfolge:

1. Zunächst versucht der Client die ATM-Adresse des LECS über ILMI zu erfahren.

2. Kann die Adresse nicht über ILMI ermittelt werden, benutzt der Client eine vordefinierte ('*well-known*') ATM-Adresse (X"47007900000000000000000000-00A03E000001-00"), um die Verbindung zu LECS aufzubauen.

3. Gelingt ihm auch das nicht, so wird eine feste, reservierte PVC (VPI=0, VCI=17) für die *Configuration-Direct*-Verbindung benutzt.

4.2.3 Configuration Phase

In der Konfigurationsphase wird der LE-Client einem emulierten LAN (ELAN) zugeordnet und für die *Join*-Phase vorbereitet. Die einzelnen Clients können dabei unabhängig von der physikalischen Lage verschiedenen emulierten LANs zugeordnet werden. Ein LE-Client erfährt in dieser Phase Konfigurationsparameter, zu denen vor allem die ATM-Adresse des LE-Servers gehört, und die ihm erlauben, sich an das emulierte LAN anzuschließen.

Während der Konfigurationsphase werden über die *Configuration-Direct*-Verbindung zwei Kontrollnachrichtentypen ausgetauscht:

- **LE_Configuration_Request**: Dieser wird vom LE-Client an den *LE Configuration Server* geschickt, um von ihm die Konfigurationsinformationen zu erhalten.

- **LE_Configuration_Response**: Dieser wird vom LECS als Antwort auf den LE_*Configuration_Request* geschickt und übermittelt dem Client die nötigen Konfigurationsparameter.

Zunächst wird ein LE_*Configuration_Request* vom LE-Client zum LECS geschickt. In dieser Nachricht muß vor allem die eigene ATM-Adresse des Clients (C1) im SOURCE-ATM-ADDRESS-Feld (siehe Bild 4.7) übertragen werden, damit der LECS den Client identifizieren kann.

Auf der Basis der eigenen Konfigurationsinformationen und der in LE_*Configuration_Request* gelieferten Informationen über den Client, kann der *LE Configuration Server* den LE-Client einem emulierten LAN zuordnen, indem er ihm die ATM-Adresse des zuständigen LE-Servers, zusammen mit anderen Konfigurationsparametern, im LE_*Configuration_Response* übermittelt.

Der LE-Client kopiert daraufhin die in dieser Nachricht erhaltenen Parameter wie LAN-Type, MAXIMUM-FRAME-SIZE, TARGET-ATM-ADDRESS und ELAN-NAME (siehe Bild 4.7) in seine entsprechenden Variablen C2, C3, C9 (ATM-Adresse des LE-Servers) und C5.

Nach dem erfolgreichen Austausch der Konfigurationsparameter kann die *Configuration-Direct*-Verbindung optional abgebaut werden.

Soll der Client die statischen, manuell konfigurierten Parameter verwenden, dann braucht die *Configuration-Phase* nicht durchgeführt zu werden, und es kann auf den LECS verzichten werden. Der LE-Client benutzt dann eine vorkonfigurierte SVC oder PVC zum LE-Server.

4.2.4 Join Phase

Während der *Join*-Phase baut der LE-Client eine *Control-Direct*-Verbindung zum LE-Server auf, dessen ATM-Adresse er während der *Configuration*-Phase erhalten hat. Dann führen Client und Server einen Abgleich ihrer Daten durch, und der LE-Client erfährt die engültigen Werte für seine Parameter, die er für den korrekten Betrieb an dem emulierten LAN benötigt. Gleichzeitig kann der Client in der *Join*-Phase implizit eine seiner MAC-Adressen (aus C6) beim LE-Server registrieren lassen.

Das *Join*-Protokoll benutzt zwei Typen von Kontrollnachrichten:

- **LE_Join_Request**: Dieser wird vom LE-Client zum LE-Server geschickt, um sich bei ihm anzumelden und an der LAN-Emulation teilnehmen zu können.
- **LE_Join_Response**: Mit dieser Nachricht bestätigt der LE-Server den LE_*Join_Request* des Clients und übermittelt ihm die notwendigen Parameter, oder er lehnt den *Join* ab.

Nachdem die *Control-Direct*-Verbindung aufgebaut worden ist, schickt der LE-Client zunächst einen LE_*Join_Request* zum LE-Server. Diese Nachricht muß folgende Variablen des Clients enthalten: C2 (*LAN-Type*), C3 (*Maximum-Data-Frame-Size*), C4 (*Proxy*), C5 (*ELAN-Name*) sowie die eigene ATM-Adresse des Clients (C1). Zusätzlich kann sie auch eine der MAC-Adressen des Clients (aus C6) enthalten, die zusammen mit der SOURCE-ATM-ADDRESS als Paar beim LE-Server registriert werden soll.

Empfängt der LE-Server den LE_*Join_Request*, dann entscheidet er, ob er ihn akzeptiert oder ablehnt.

Soll ein LE_*Join_Request* abgelehnt werden, antwortet der LE-Server mit einem LE_*Join_Response*, der im STATUS-Feld (siehe Bild 4.7) einen Hinweis auf den Grund der Ablehnung enthält.

Falls der LE_*Join_Request* dagegen akzeptiert wird, kann der LE-Server optional eine *Control-Distribute*-Verbindung zum Client aufbauen. Meistens geschieht das, indem der Server den Client an seine Punkt-zu-Mehrpunkt-Verbindung anschließt, die er zum Verteilen von Kontrollinformationen benutzt. Anschließend wird vom Server die erfolgreiche Integration des Clients in das emulierte LAN durch ein LE_*Join_Response* (STATUS-Feld = *'Success'* bestätigt. In dieser Nachricht wird, außer den endgültigen Werten für die Variablen C2, C3 und C5, zusätzlich dem Client sein *LE-Client IDentifier* (LECID) übergeben, der ihn in einem emulierten LAN eindeutig identifiziert soll. Akzeptiert der Client die Werte, wird die *Join*-Phase erfolgreich abgeschlossen.

4.2.5 Initial Registration

Im Anschluß an die *Join*-Phase kann der LE-Client seine zusätzlichen MAC-Adressen[2] (aus der Variable C6) beim Server registrieren lassen. Jede MAC-Adresse wird jeweils einzeln in Verbindung mit der entsprechenden ATM-Adresse als Paar beim Server registriert.

[2] und im Falle eines emulierten Token-Ring-LAN auch die *Route Descriptors*

Zur Registrierung werden folgende Nachrichtentypen verwendet:

- **LE_Register_Request**: Dieser wird vom LE-Client zum LE-Server geschickt, um ein Paar, bestehend aus MAC- und ATM-Adresse, beim Server registrieren zu lassen.
- **LE_Register_Response**: Mit dieser Nachricht wird ein LE_*Register_Request* vom Server bestätigt oder abgelehnt.
- **LE_Unregister_Request**: Der LE-Client schickt diese Nachricht, um ein registriertes Paar beim Server zu löschen.
- **LE_Unregister_Response**: Dieser wird vom Server als Antwort auf LE_*Unregister_Request* zurückgeschickt.

Ein Client, der nur eine MAC-Adresse besitzt, kann die *Registration*-Phase überspringen, falls er diese Adresse schon während der *Join*-Phase beim Server registriert hat.

Bevor der LE-Client den '*Operational*'-Zustand erreicht, müssen alle lokalen MAC-Adressen aus der Variable C6 beim Server registriert werden. Im Gegensatz dazu dürfen die *Remote*-MAC-Adressen in der Variable C27 nicht beim Server registriert werden.

Um eine MAC-Adresse registrieren zu lassen, schickt der LE-Client ein *LE_Register_Request* über die *Control-Direct*-Verbindung zum Server. Das zu registrierende Paar aus MAC- und ATM-Adresse wird dem Server in den Feldern SOURCE-LAN-DESTINATION und SOURCE-ATM-ADDRESS (siehe Bild 4.7) übergeben.

Der Server antwortet auf *LE_Register_Request* mit einem *LE_Register_Response*, den er über die *Control-Direct*- oder *Control-Distribute*-Verbindung schickt. Mit Hilfe des STATUS-Feldes informiert der Server den Client, ob die Registrierung erfolgreich war.

Falls ein Adressen-Paar (MAC- und ATM-Adresse) beim Server gelöscht werden soll, schickt der Client einen *LE_Unregister_Request* zum Server und erhält als Antwort einen LE_*Unregister_Response*.

4.2.6 BUS Connect Phase

Am Ende der Initialisierungsphase baut der LE-Client schließlich eine Datenverbindung (*Multicast Send VCC*) zum BUS (*Broadcast and Unknown Server*) auf. Über diese Verbindung schickt der Client alle *Multicast*-Pakete zum BUS, der sie dann weiter verteilt.

Bevor der Client diese Verbindung aufbauen kann, braucht er die ATM-Adresse des BUS. Um diese zu erfahren, benutzt er das *LAN Emulation Address Resolution Protocol* (LE_ARP). Der LE-Client schickt einen LE_ARP_*Request* über die *Control-Direct*-Verbindung zum LE-Server, um die *Broadcast*-MAC-Adresse des BUS (alle Bits 1) nach der ATM-Adresse aufzulösen. Der LE-Server antwortet darauf mit einem LE_ARP_*Response*, in dem er dem Client die ATM-Adresse des BUS (S6) übermittelt.

Damit der Client auch die *Broadcast*- und *Multicast*-Pakete empfangen kann, wird vom BUS eine Verbindung (*Multicast Forward VCC*) zum Client aufgebaut. Dies ist entweder eine neue Punkt-zu-Punkt Verbindung oder der BUS schließt den Client an seine Punkt-zu-Mehrpunkt Verbindung an.

Wird die Verbindung zum BUS abgebrochen, kann der Client versuchen, diese neu aufzubauen, indem er die BUS-*Connect*-Phase wiederholt.

In Bild 4.8 wird der komplette Nachrichtenaustausch dargestellt, der zwischen dem LE-Client und den LAN-Emulation-*Service*-Komponenten während der Initialisierungsphase stattfindet.

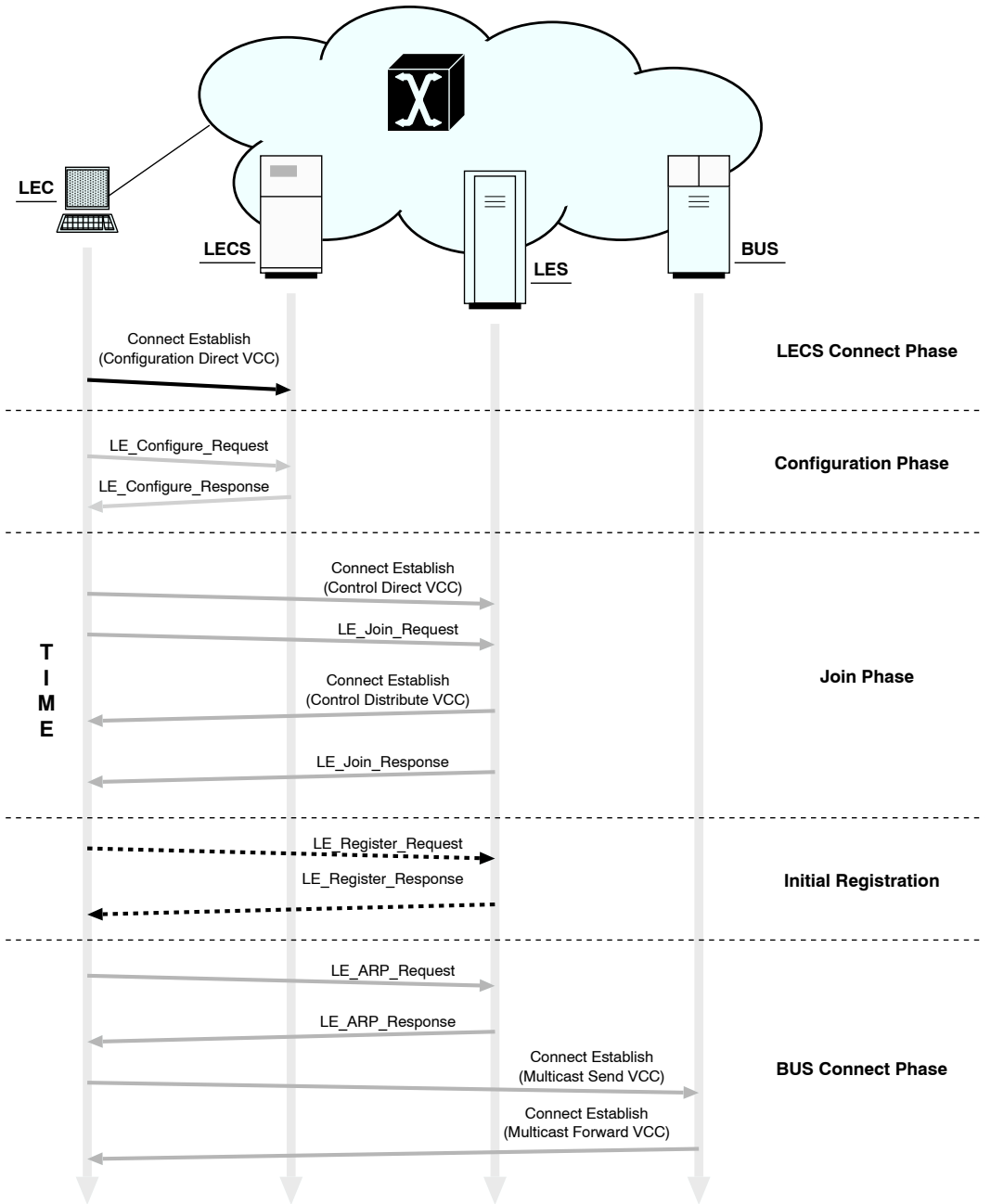


Bild 4.8: Initialisierung

4.3 Datentransfer

4.3.1 Verpackung der Pakete

Beim Transfer von Ethernet-Paketen wird sowohl die LLC-Verpackung nach IEEE 802.3 (siehe RFC 1042 [19]) als auch die häufiger verwendete DIX[3]-Ethernet-Verpackung unterstützt. Das Format der Datenpakete, das für die Ethernet-Pakete bei der LAN-Emulation verwendet wird, ist in Bild 4.9 dargestellt. Einerseits wird das FCS-Feld (*Frame Check Sequence*) weggelassen, das bei ATM überflüssig wäre, denn die Fehlerkorrektur wird schon mittels CRC-Verfahren innerhalb der AAL5-Schicht durchgeführt. Andererseits wird aber zusätzlich ein 2 Bytes langer LE-*Header* am Anfang des Pakets hinzugefügt. Er enthält normalerweise den LECID (*LAN Emulation Client IDentifier*), der in einem emulierten LAN (ELAN) eindeutig ist und zum schnellen Filtern von Paketen benutzt wird. Die minimale Länge der AAL5-SDU, die bei der LAN-Emulation zur Übertragung der Ethernet-Pakete verwendet wird, beträgt 62 Bytes.

Weil in den emulierten LANs bei einer reinen ATM-Umgebung längere Pakete als in klassischen LANs transportiert werden können, muß der Client bei der Codierung des *Type/Length*-Feldes folgende Regeln beachten:

- Soll ein DIX-Ethernet-EtherType-Paket geschickt werden, wird das EtherType-Feld im *Type/Length*-Feld übertragen. Die Daten, die hinter dem EtherType-Feld stehen, folgen ebenfalls direkt dem *Type/Length*-Feld.
- Bei LLC-verpackten Paketen, deren Länge – inklusive LLC-*Header*, aber ohne FCS-Feld – kleiner als 1536 Bytes (X"0600") ist, enthält das *Type/Length*-Feld die Länge des Pakets, gefolgt direkt vom LLC-Feld.

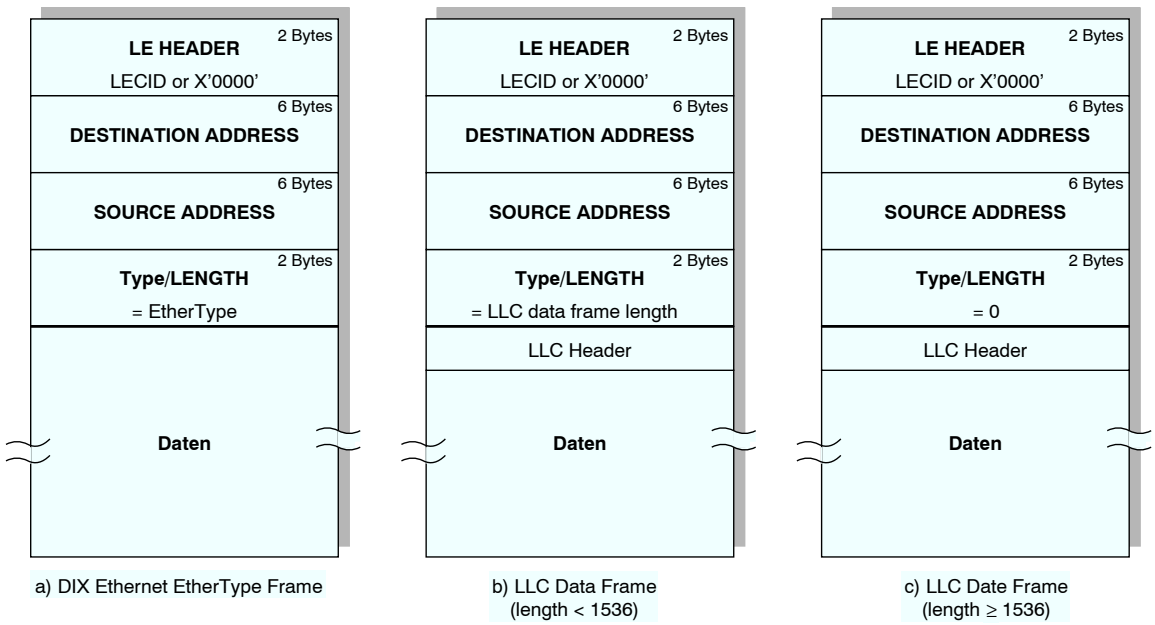


Bild 4.9: Verpackung der Ethernet-Pakete

[3] DIX wurde von DEC, Intel und Xerox entwickelt

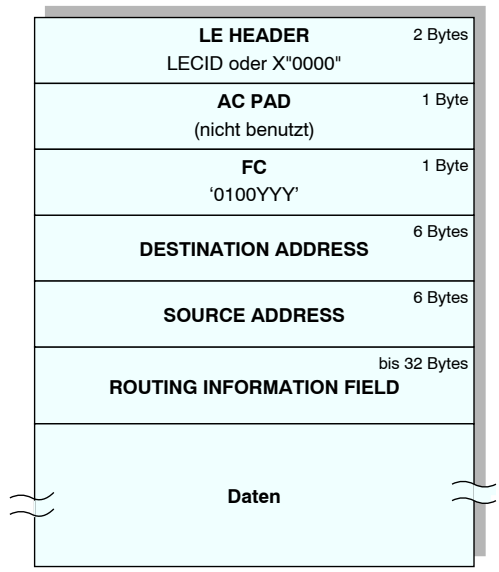


Bild 4.10: Verpackung der IEEE-802.5-Pakete

- Bei längeren Paketen (gleich oder größer als 1536 Bytes) wird das *Type/Length*-Feld auf Null gesetzt. Das LLC-Feld folgt wieder direkt dem *Type/Length*-Feld.

Beim Decodieren prüft der LE-Client den Wert des *Type/Length*-Feldes. Ist er größer oder gleich 1536 Bytes, handelt sich um ein DIX-Ethernet-EtherType-Paket, sonst ist es ein IEEE-802.3-LLC-Paket.

Das Format für die zweite Art von Paketen – die Token-Ring-Pakete – wird in Bild 4.10 gezeigt. Die minimale Länge der LANE-AAL5-SDU beträgt bei Token-Ring-Paketen 16 Bytes. Der LE-*Header* enthält wieder den LECID, und das FCS-Feld wird weggelassen. Das AC-PAD-Byte wird nicht benutzt und kann beliebige Werte enthalten. Weil nur LLC-Pakete unterstützt werden, besitzt das FC-Feld das Format '01000YYY', wobei 'YYY' die Priorität bezeichnet, die von der sendenden LLC-Teilschicht gesetzt wird.

4.3.2 Datenaustausch

Der normale Datentransfer zwischen zwei Clients in einem emulierten LAN findet über direkte Punkt-zu-Punkt-Verbindungen (*Data Direct VCC*) statt. Datenpakete mit einer *Broadcast*- oder *Multicast*-Adresse werden dagegen prinzipiell über die *Multicast-Send*-Verbindung zum BUS geschickt. Dieser verteilt sie über seine *Multicast-Forward*-Verbindung an alle Clients, die wie bei klassischen LANs die für sie bestimmten Pakete herausfiltern können.

Hat ein LE-Client Daten für einen einzelnen Client, zu dem noch keine direkte Verbindung existiert, dann versucht er eine aufzubauen. Kennt der Client aber nur die MAC-Adresse des Zielsystems, dann muß diese nach der ATM-Adresse aufgelöst werden. Beim Ermitteln der ATM-Adresse für eine vorgegebene MAC-Adresse wird das LAN *Emulation Address Resolution Protocol* (LE_ARP) verwendet. Dabei wird der Client vom LE-Server unterstützt.

Der LE-Client schickt zunächst einen LE_ARP_*Request* mit der MAC-Adresse, zu der die ATM-Adresse gesucht wird, an den LE-Server. Gleichzeitig kann er, um die Verbindungs-

aufbauphase zur überbrücken und Verzögerungen möglichst gering zu halten, anfangen, die Daten zum BUS zu senden, der sie über seine *Multicast-Forward*-Verbindung an alle Clients weiterleitet.

Kann die MAC-Adresse aufgelöst werden, bekommt der Client als Antwort einen LE_ARP-*Response*, in dem ihm die gewünschte ATM-Adresse übermittelt wird. Nach dem Empfang dieser Nachricht kann die direkte Verbindung zum Ziel-Client aufgebaut werden.

Bevor der Client aber über diese neue Verbindung Daten zum anderen Client senden darf, muß sichergestellt werden, daß alle Datenpakete, die vorher zum BUS geschickt wurden, den Ziel-Client schon erreicht haben, damit die Reihenfolge der Pakete erhalten bleibt.

Der Client verschickt dazu über den BUS ein LE_*Flush_Request*. Die angesprochene Station antwortet darauf mit einem LE_*Flush_Response* und bestätigt damit, daß die vor dem LE_*Flush_Request* gesendeten Pakete empfangen wurden. Erst dann (oder wenn eine vordefinierte Zeit abgelaufen ist) kann die neue Verbindung zum Transfer von Daten benutzt werden.

Empfängt der Client aber kein LE_ARP_*Response*[4], dann verschickt er die Pakete weiter via BUS an alle Clients, insbesondere an alle *Proxy*-Clients, von denen sie im Falle einer *Transparent Bridge* an alle Ports (bzw. nur an den entsprechenden Port) weitergeleitet werden. Gleichzeitig muß der Client aber regelmäßig den LE_ARP_*Request* wiederholen.

4.3.3 LE Address Resolution Protocol (LE_ARP)

Das LE_ARP-Protokoll dient in einem emulierten LAN zur Adreßauflösung der MAC-Adressen nach ATM-Adressen.

Im Rahmen des LE_ARP-Protokolls werden vier Nachrichtentypen ausgetauscht:

- **LE_ARP_Request**: Dieser wird vom LE-Client an den LE-Server geschickt, um eine ATM-Adresse zu einer gegebenen MAC-Adresse zu erfahren.
- **LE_ARP_Response**: Dieser wird vom LE-Server oder Client als Antwort auf LE_ARP-*Request* geschickt und enthält die gesuchte ATM-Adresse.
- **LE_NARP_Request**: Mit dieser Nachricht teilt der LE-Client mit, daß sich die ATM-Adresse zu einer *Remote*-MAC-Adresse geändert hat.
- **LE_TOPOLOGY_Request**: Dieser wird vom Client geschickt, um Änderungen in der Netztopologie anzukündigen.

Die LE-Clients schicken die LE_ARP-Nachrichten über die *Control-Direct*-Verbindung zum LE-Server. Vom LE-Server werden sie dagegen entweder über die *Control-Direct*- oder *Control-Distribute*-Verbindung zu den Clients geschickt.

Benötigt also ein LE-Client eine ATM-Adresse zur einer gegebenen MAC-Adresse, dann schickt er einen LE_ARP_*Request* an den LE-Server. Das Verhalten des Servers hängt von der Implementierung ab; in der LANE-Spezifikation sind verschiedene Realisierungen erlaubt. Ein einfacher LE-Server leitet den LE_ARP_*Request* via Broadcast über seine *Control-Direct*-Verbindung an alle Clients weiter. Bei intelligenteren Implementierungen nutzt der Server seine Informationen über die registrierten Clients. Einerseits kann er selbst auf einen

[4] Das kann passieren, wenn eine MAC-Adresse nur über eine IEEE 802.1D *Transparent Bridge* erreichbar ist, die aber keine Informationen über die Ziel-MAC-Adresse besitzt.

LE_ARP_*Request* mit einem LE_ARP_*Response* antworten, falls die MAC-Adresse bei ihm registriert wurde und die ATM-Adresse ihm bekannt ist. Andererseits kann er den LE_ARP_*Request* nur an das entsprechende Zielsystem weiterleiten, auf das sich die MAC-Adresse in der LE_ARP-Anfrage bezieht.

Empfängt der LE-Server aber einen LE_ARP_*Request* mit einer nicht registrierten MAC-Adresse, dann leitet er ihn entweder an alle oder nur an die als *Proxy* registrierten Clients weiter.

Empfängt ein Client einen LE_ARP_*Request*, der sich auf eine MAC-Adresse bezieht, für die er zuständig ist (also eine aus seinen Variablen C6 oder C27), dann muß er mit einem LE_ARP_*Response* antworten.

Ist der Client ein *Proxy*-Client und handelt es sich bei diesem *Request* um eine *Remote*-MAC-Adresse (eine in der Variable C27), muß das *Remote-Address*-Flag in dieser Nachricht gesetzt werden. Bei den lokalen MAC-Adressen (C6) bleibt sie dagegen gelöscht.

In einem herkömmlichen mit *Bridges* gekoppelten LAN-System, könnte es vorkommen, daß Pakete zirkulieren und sich Schleifen bilden. Um diese unerwünschten Schleifen zu verhindern, wird in den *Transparent Bridges* ein Schleifenunterdrückungsprotokoll implementiert, das unter der Bezeichnung *Spanning-Tree-Algorithmus* bekannt und im Standard IEEE 802.1D festgelegt ist. Wird ein solcher Algorithmus verwendet, werden redundante *Bridges* (bzw. nur die einzelnen Ports) logisch vom restlichen Netz abgetrennt, was dazu führt, daß sich die logische Topologie des Netzes ändert.

Um die LE-Clients auch über solche Änderungen in der Netztopologie zu informieren, werden in einem emulierten LAN LE_*Topology_Requests* verschickt. Empfängt z.B ein LE-*Proxy*-Client, der in einem als IEEE 802.1D *Transparent Bridge* arbeitenden LAN-*Switch* implementiert ist, eine *Configuration*-BPDU (*Bridge Protocol Data Unit*), dann sendet er eine LE_*Topology_Change*-Nachrict an den LE-Server, der diese dann an alle Clients weiterleitet.

Ein LE-Client kann auch unaufgefordert einen LE_NARP_*Request* senden, um andere Clients über Änderungen in der Adreßzuordnung zu informieren. Mit einem LE_NARP_*Request* teilt ein Client mit, daß eine MAC-Adresse, die früher durch eine entfernte ATM-Adresse vertreten war, jetzt über ihn erreichbar ist.

Diese Nachricht schickt der Client zum Server, von dem sie dann an alle Clients verschickt wird. Sie kann dann von diesen wiederum als ein Hinweis auf eventuelle Änderungen in ihren Adreßtabellen angesehen werden.

Ein LE_NARP_*Request* darf aber nicht gesendet werden, falls das *Topology-Flag* beim Client gesetzt ist, was bedeutet, daß gerade eine allgemeinere Änderung in der Netzwerktopologie stattfindet.

Weiterhin besitzt jeder LE-Client eine *Cache*-Tabelle (C16) für die Abbildung von MAC-Adressen auf ATM-Adressen. In dieser *Cache*-Tabelle werden Informationen gehalten, die der Client durch die empfangenen LE_ARP_*Responses* und LE_NARP_*Requests* erhält. Zusätzlich kann der Client Adreßinformationen gewinnen, indem er die Senderadressen in den über eine *Data-Direct*-Verbindung ankommenden Paketen überprüft.

Wenn der Client ein Paket für eine bestimmte MAC-Adresse zu senden hat, dann konsultiert er zuerst die *Cache*-Tabelle und sucht die dazugehörige ATM-Adresse. Erst wenn sie nicht bekannt ist, schickt er einen LE_ARP_*Request* an den LE-Server.

Für die Einträge in der *Cache*-Tabelle wird ein *Timeout*-Mechanismus verwendet. Nach Ablauf einer bestimmten Zeit wird ein Eintrag ungültig. Durch die begrenzte Dauer der Gültigkeit wird die Information automatisch an Veränderungen im Netz angepaßt.

Bevor ein Eintrag ungültig wird, versucht der Client ihn zu verifizieren (erneuern). Das

geschieht, wie beim Erzeugen eines Eintrags, entweder durch Empfang der LE_ARP_*Responses* und LE_NARP_*Requests*, die sich auf diesen Eintrag beziehen, oder durch das Beobachten der Senderadresse in den *Data-Direct*-Paketen.

Ein Eintrag in der *Cache*-Tabelle verliert normalerweise seine Gültigkeit, falls seit der letzten Verifizierung die sog. '*Aging-Time*' vergangen ist. Wird aber im Netz durch den LE_*Topology_Request* eine Änderung der Netztopologie angekündigt, wird als *Timeout* für bestimmte Einträge die kürzere sog. '*Forward-Delay-Time*' benutzt. Dabei handelt sich um die Einträge, die sich auf *Remote*-MAC-Adressen beziehen, und um die Einträge, die durch das Beobachten der Senderadresse in den *Data-Direct*-Paketen entstanden sind. Dadurch kann ein Client schneller auf Änderungen im Netz reagieren.

4.3.4 Flush Protocol

Beim normalen Betrieb kann ein LE-Client in verschiedenen Zeitabschnitten Daten entweder über den BUS oder über die *Data-Direct*-Verbindung zu einem anderen Client schicken. Das *Flush*-Protokoll erlaubt dem sendenden Client, zu vermeiden, daß die Pakete beim Empfänger in der falschen Reihenfolge ankommen, was leicht passieren kann, wenn zwischen den beiden Möglichkeiten gewechselt wird.

Während der '*Flush*'-Phase werden zwei Nachrichtentypen ausgetauscht:

- **LE_Flush_Request**: Dieser wird vom Client entlang des bisherigen Weges (über die *Data-Direckt-* oder *Multicast-Send*-Verbindung) geschickt, um sicher zu stellen, daß alle über diesen Weg gesendeten Pakete ihr Ziel erreicht haben.

- **LE_Flush_Response**: Dieser wird als Antwort auf den LE_*Flush_Request* über die *Control-Direct-* und *Control-Distribute*-Verbindung geschickt.

Wenn ein LE-Client vom alten Weg auf einen neuen Weg umschalten will, schickt er entlang des alten Weges einen LE_*Flush_Request*. Das passiert z.B. wenn er eine neue *Data-Direct*-Verbindung zu dem anderen Client aufgebaut hat, und die Dienste des BUS nicht mehr in Anspruch zu nehmen braucht.

Empfängt der andere Client den LE_*Flush_Request*, bedeutet dies, daß die vor dem LE_*Flush_Request* gesendeten Pakete auch angekommen sind (die Pakete auf einem Weg können nicht überholt werden). Er bestätigt dies dem sendenden Client, indem er ihm einen LE_*Flush_Response* über die Kontrollverbindung zurückschickt.

Nach Empfang des LE_*Flush_Responses* (oder auch nach einem *Timeout*), kann der Sender sicher sein, daß der alte Weg frei von Datenpaketen ist, und er kann die Daten über den neuen Weg schicken.

'*Flush*'-Nachrichten werden nur zwischen den LE-Clients ausgetauscht und werden nicht über die *Bridges* in klassische LANs weitergeleitet.

4.4 Verbindungsmanagement

Der Auf- und Abbau der Verbindungen erfolgt bei der LAN-Emulation ähnlich wie in den *Classical*-IP-Netzen (siehe Kapitel 3.4 auf Seite 36). In der SETUP-Nachricht werden die gleichen *Information Elements* (IEs) verwendet, in denen AAL 5, *Best Effort* und *QoS Class 0* identifiziert wird.

Bei der LAN-Emulation werden fünf Arten[5] von Verbindungen benutzt. Der Typ der Verbindung wird während des Verbindungsaufbaus im B-LLI-Informationselement (*Broadband Low Layer Information*) identifiziert.

Um Resourcen nicht unnötig zu verschwenden, wird bei den Clients eine sog. *VCC-Time-out-Period* definiert. Wird innerhalb dieser Zeit eine *Data-Direct*-Verbindung nicht benutzt, dann wird sie abgebaut.

Weiterhin findet bei den *Data-Direct*-Verbindungen nach dem Verbindungsaufbau ein zusätzlicher Kontrollnachrichtenaustausch statt, damit die gerufene Station (*Called Party*) sicher sein kann, daß die rufende Station (*Calling Party*) zum Empfang von Daten bereit ist.

In Bild 4.11 wird der Austausch der UNI-Signalisierungs- und LANE-Kontrollnachrichten gezeigt, der beim Aufbau einer *Data-Direct*-Verbindung stattfindet.

Nach dem Empfang der SETUP-Nachricht schickt die gerufene Station eine CONNECT-Nachricht, sobald sie bereit ist, Daten zu empfangen. Das Netzwerk bestätigt diese Nachricht mit CONNECT ACKNOWLEDGE. Weil letztere aber lokale Bedeutung hat, kann die gerufene Station durch ihren Empfang noch nicht sicher sein, ob die rufende Station die CONNECT-Nachricht erhalten hat.

Deshalb wartet die gerufene Station mit dem Senden von Daten solange, bis sie ein *Ready_Ind* empfängt. Empfängt sie innerhalb der sog. *Connection-Completion-Time* kein *Ready_Ind* (falls sie z.B. verloren geht), dann kann sie trotzdem anfangen, Daten zu senden. Zusätzlich kann sie aber ein *Ready_Query* schicken, auf den die rufende Station mit *Ready_Ind* zu antworten hat.

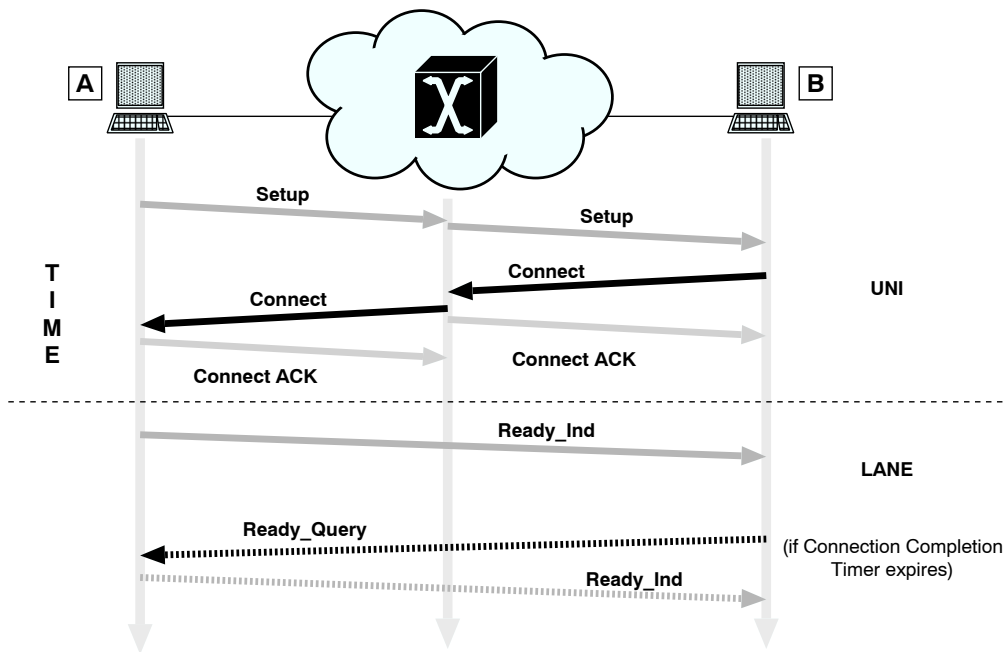


Bild 4.11: Aufbau einer *Data-Direct*-Verbindung

[5] Es handelt sich dabei um Kontrollverbindungen, *Data Direct* für IEEE 802.3, *Data Direct* für IEEE 802.5, *Multicast*-Verbindungen für IEEE 802.3 und *Multicast*-Verbindungen für IEEE 802.5.

4.5 LAN-Emulation: Einschränkungen und offene Probleme

Die LAN-Emulation spielt eine wichtige Rolle bei der Integration von herkömmlichen Netzen und ATM-Netzen. Man sollte aber die Einschränkungen nicht vergessen, die sie in der Version 1.0 noch mit sich bringt. Auf die wesentlichen wird im folgenden näher eingegangen:

- Die LANE-Spezifikation erlaubt nur, Ethernet- und Token-Ring-LAN-Segmente zu emulieren. FDDI-LANs sind nur indirekt und umständlich über Ethernet oder Token-Ring emulierbar. Die FDDI-Pakete müssen dabei zuerst in Ethernet- oder Token-Ring-Pakete konvertiert werden, was verarbeitungs- und zeitaufwendig ist. Das bedeutet, daß eine Kommunikation z.B. zwischen einem emulierten LAN und FDDI nur über Router möglich ist. Andere LAN-Techniken wie z.B. Fast-Ethernet können ebenfalls nicht über ATM emuliert werden.

- Die LANE-Spezifikation definiert nur, wie ein einzelnes ELAN funktioniert. Sie beschreibt nicht, wie mehrere ELANs auf einem ATM-Netz zusammen arbeiten könnten. Ein ELAN arbeitet unabhängig von jedem anderen ELAN auf dem gleichen ATM-Netz. Für jedes ELAN benötigt man einen separaten LE-Server und *Broadcast and Unknown Server*. Es wird nicht darauf eingegangen, wie sie untereinander kommunizieren könnten, um Informationen über Clients auszutauschen.

- Daraus folgt, daß zwei auf einem ATM-Netz emulierte LANs nicht direkt miteinander kommunizieren können. Zwischen zwei Clients in unterschiedlichen ELANs kann man also keine direkte ATM-Verbindung aufbauen. Sie müssen über einen Router komunizieren, der als Client in mehreren ELANs konfiguriert ist. Bei Verwendung von Routern enstehen aber unnötige Verzögerungen, und man verliert wichtige ATM-Eigenschaften wie Skalierbarkeit und Zeittransparenz.

- Ebensowenig ist eine direkte Kommunikation zwischen zwei unterschiedlichen ELAN-Typen möglich. LAN-Emulation löst also nicht das *'Mixed-media-bridging'*-Problem auf. Es ist kein Topologiewechsel zwischen zwei verschiedenen LAN-Techniken möglich.

- LAN-Emulation weist genauso wie das Classical-IP-Modell ein *Single-Point-Of-Failure*-Problem auf. Die wichtigsten Komponenten eines emulierten LAN, nähmlich LECS, LE-Server und BUS, sind im Netz nicht redundant auslegbar. Der Ausfall von LES oder BUS kann das ganze Netz lahmlegen.

  Das ATM-Forum arbeitet zur Zeit an der LANE-Spezifikation Phase 2, die u.a. beschreiben soll, wie in einem emulierten LAN mehrere LE-Servers, BUSs und LECSs implementiert werden können, und wie sie untereinander kommunizieren.

- Bei der LAN-Emulation werden die physikalischen LAN-Segmente auf der MAC-Ebene emuliert, die LANE-Funktionen orientieren sich an der Ebene 2. Dadurch werden die typischen *Layer*-2-Einschränkungen beibehalten und der Ansatz skaliert schlecht; er ist eher für kleinere Netzwerke geeignet. Es werden *Broadcast*-Domänen gebildet, und man hat bei LANE wegen der *Broadcast*-Problematik mit ähnlichen Beschränkungen wie bei klassichen LANs zu kämpfen.

- Weil der ganze *Broadcast*-Verkehr über den BUS erfolgt, der nur einmal im Netz verfügbar ist, kann es schnell zu Engpässen kommen. Erst in der LANE-Spezifikation Phase 2 soll

die Implementierung von verteilten *Broadcast and Unknown Servers* möglich sein, die sogar wegen der besseren Skalierbakeit eine Hierarchie bilden können.

- Im Vergleich zum *Classical*-IP-Modell muß bei der LAN-Emulation eine Netzwerkadresse (IP-Adresse) zuerst nach einer MAC-Adresse und erst dann nach einer ATM-Adresse aufgelöst werden, was einen zusätzlichen Ballast durch das größere *Broadcast*-Aufkommen mit sich bringt.
- LAN-Emulation läßt – wie das *Classical*-IP-Modell – das ATM-Netz für höhere Schichten so aussehen wie ein Standard-LAN. Dies macht es den Anwendungen unmöglich, ATM-spezifischen Vorteile zu nutzen.

Kapitel 5

Leistungsbewertung von TCP(UDP)/IP über ATM

5.1 Probleme mit TCP bei Hochgeschwindigkeitsnetzen

TCP (*Transmission Control Protocol*) [22] hat sich als ein gutes Transportprotokoll erwiesen, das zuverlässig über verschiedene Medien, mit verschiedenen Verzögerungen und bei verschiedenen Geschwindigkeiten arbeitet. Mit der Einführung von Glasfasern und den sehr großen Übertragungsraten wurde TCP mit neuen Problemen konfrontiert, für die ursprünglich keine Lösungen entwickelt wurden.

5.1.1 Probleme mit den *long-fat-pipes*

TCP bietet einen zuverlässigen, verbindungsorientierten End-to-end-Transportdienst, der zur Flußsteuerung ein variables Schiebefenster (*Sliding-Window*) verwendet [62].

Schickt eine Station eine Nachricht, dann muß sie nicht jedes Mal auf eine Empfangsbestätigung warten. Sie kann direkt weitere Daten an die andere Station schicken, und zwar darf sie so viele Bytes ohne Bestätigung senden, wie in dem Fenster freigegeben sind.

Das Fenster wird dadurch immer kleiner. Es wird erst wieder vergrößert, wenn durch eine Quittierung vom Empfänger einige Bytes freigegen werden. In den Quittierungen teilt der Empfänger dem Sender jeweils mit, wieviele Bytes er noch aufnehmen kann (wie groß momentan sein Empfangsfenster ist).

Die Größe des *Sliding-Window* hat bei Hochgeschwindigkeitsstrecken einen großen Einfluß auf die *Performance*. Die *Performance* von TCP hängt nämlich in erster Linie weniger von der Übertragungsrate selbst ab, sondern vielmehr vom sog. *Bandwidth-Delay*-Produkt, das die Menge der Daten angibt, die in einer Kommunikationsleitung unterwegs (und somit in dieser gespeichert) sein können. Es wird als das Produkt der *Round-Trip*-Verzögerungen, die durch Verbreitungszeit und Warteschlangen entlang des ganzes Weges zum Empfänger und zurück entstehen, mit der für die Verbindung verfügbaren Bandbreite definiert, und beschreibt die Menge der Daten, die eine gegebene Leitung füllen kann.

Das *Bandwidth-Delay*-Produkt gibt somit an, wie viele unbestätigte Bytes ein Sender handhaben muß, wenn er die Leitung immer mit Daten gefüllt halten und damit einen kontinuerlichen Datenfluß gewährleisten will.

Die optimale *Performance* erreicht man, wenn ein Fenster benutzt wird, dessen Größe

mindestens gleich dem *Bandwidth-delay*-Produkt ist.

Wenn ein kleineres Fenster benutzt wird, muß der Sender, nachdem er alle Daten aus diesem Puffer geschickt hat, mit dem weiteren Senden warten, bis eine Bestätigung empfangen wird. Durch das Warten wird der Sender daran gehindert, die verfügbare Bandbreite voll auszunutzen.

Probleme mit der TCP-*Performance* entstehen dann, wenn man mit sog. *long-fat-pipes* zu tun hat, für die das *Bandwidth-Delay*-Produkt groß ist. Um gute *Performance* zu erreichen, muß man sehr große Puffer benutzen. Das Problem liegt dann darin, daß in dem TCP-*Header* nur ein 16 Bits langes Feld zum Anzeigen der Fenstergröße verwendet wird. Dadurch ist die Größe des Fensters, das benutzt werden kann, auf maximal $2^{16} = 64$ KBytes begrentzt.

Wenn wir aber eine 622 Mbit/s–Verbindung betrachten (z.B. zwischen KFA und GMD), bei der aufgrund der geografischen Entfernung Verzögerungen von 2 ms denkbar sind, dann ist das *Bandwidth-delay*-Produkt ca. 130 kBytes groß. Um die Bandbreite voll ausnutzen zu können, würde man also Puffer benötigen, deren Größe 64 KBytes übersteigt.

Um die Probleme mit *long-fat-pipes* zu umgehen, werden in RFC1323 [14] einige Erweiterungen für TCP definiert. Mit einer sog. *Window-Scale*-Option wird die Benutzung von größeren Fenstern als 64 KBytes erlaubt. Diese Option definiert einen Skalierungsfaktor, mit dem der Wert für die Fenstergröße aus dem TCP-*Header* multipliziert wird, um die tatsächliche Fenstergröße zu erhalten.

5.1.2 MTU-/MSS-Problematik

Außer der Fenstergröße hat auch die Länge der IP-Pakete einen gravierenden Einfluß auf die *Performance*. Die Bearbeitungszeit ist beim Sender und Empfänger sowie bei den dazwischenliegenden Routern weitgehend unabhängig von der Größe der Pakete. Bei einem gegebenen Datenblock müssen bei Verwendung von kleineren Paketgrößen mehrere Pakete verschickt werden. Das verursacht einen größeren Protokoll-*Overhead* und führt zu einer größeren Belastung des Systems.

TCP teilt für die Übertragung den Datenstrom in Datensegmente auf. Die maximale Größe dieser Segmente (MSS, *Maximum Segment Size*) wird beim Verbindungsaufbau ausgehandelt (siehe [20]). Für die Kommunikation innerhalb des lokalen Netzes wird die MSS auf der Basis der für das benutzte *Interface* maximal zulässigen Paketlänge (MTU, *Maximum Transfer Unit*) bestimmt. TCP berechnet die MSS, indem von der MTU die Länge der IP- und TCP-*Header* abgezogen wird ($MSS = MTU - 20 - 20$). Für die Verbindungen mit Rechnern außerhalb des lokalen Netzes wird dagegen eine im Betriebssystem eingestellte *Default*-MSS genommen. Diese ist meistens nur 536 Bytes groß (gemäß [17]), um zu verhindern, daß die Pakete im Netz durch Router fragmentiert werden müssen.

Eine Lösung des MSS-Problems stellt die im RFC 1191 [16] vorgestellte Technik der *'Path MTU Discovery'* dar, die eine dynamische Anpassung der MSS(MTU)-Größe an die Eigenschaften des Datenpfades erlaubt. Es wird für den Pfad zum Empfänger die größtmögliche Paketgröße (sog. *Path*-MTU) herausgefunden, die von allen betreffenden *Links* ohne Fragmentierung unterstützt wird.

Um die *Path*-MTU dynamisch feststellen zu können, wird eine Technik eingesetzt, die das DF-Bit (*Don't Fragment*) im IP-*Header* [24] benutzt. Die dabei verwendete Idee basiert auf folgenden Prinzip. Der Sender benutzt als *Path*-MTU zuerst die bekannte MTU-Größe für den ersten *Hop* zum Router und schickt alle Pakete mit gesetztem DF-Bit. Wenn ein auf dem Weg zum Empfänger liegender Router feststellt, daß das Paket zu groß ist, um

es ohne Fragmentierung weiterzubefördern, dann verwirft er es und schickt eine ICMP[1]-*Destination-Unreachable*-Nachricht [23] zum Sender zurück, in der angegeben wird, daß eine Fragmentierung notwendig wäre, diese aber durch das gesetzte DF-Bit verhindert wird.

Empfängt der Sender diese Nachricht, die auch *'Datagram-Too-Big'*-Nachricht genannt wird, dann reduziert er entsprechend die verwendete *Path*-MTU.

Die Prozedur wird beendet, wenn eine *Path*-MTU bestimmt wurde, die klein genug ist, um die Pakete ohne Fragmentierung zum Empfänger transportieren zu können. Natürlich kann der Sender die Bestimmumg der *Path*-MTU auch früher beenden und die Pakete mit nicht gesetzten DF-Bit senden, wenn unter bestimmten Umständen die Fragmentierung der Pakete nicht unbedingt verhindert werden muß. Normalerweise schickt der Sender aber die Pakete weiterhin mit gesetztem DF-Bit, um erkennen zu können, wenn sich der Weg zum Empfänger ändert und die neue *Path*-MTU kleiner geworden ist. Er kann dann versuchen, die *Path*-MTU für den neuen Weg anzupassen.

Unglücklicherweise liefert die *Datagram-Too-Big*-Nachricht standardmäßig nicht die MTU-Größe für den *Link*, für den das Paket zu groß war, so daß der Sender nicht genau weiß, um wieviel er die *Path*-MTU reduzieren soll. Um dieses Problem zu überwinden, wird in RFC 1191 vorgeschlagen, ein zur Zeit nicht benutztes Feld in der *Datagram-Too-Big*-Nachricht zu verwenden, um dem Sender die MTU-Größe des betreffenden *Link* mitzuteilen. Bei den Routern ist das die einzige Änderung, die für die Unterstützung des *Path-MTU-Discovery*-Mechanismus notwendig ist.

Die *Path*-MTU kann sich mit der Zeit verändern, falls sich der Weg zum Empfänger ändert. Eine Reduzierung der *Path*-MTU kann der Sender daran erkennen, daß er eine *Datagram-Too-Big*-Nachricht empfängt. Um eine mögliche Vergrößerung der *Path*-MTU aber zu entdecken, muß der Sender regelmäßig versuchen, mit einer seinerseits vergrößerten *Path*-MTU (und gesetztem DF-Bit) zu senden. Das führt meistens dazu, daß das Paket verworfen, und eine *Datagram-Too-Big*-Nachricht empfangen wird, weil sich die *Path*-MTU für den Weg zum Empfänger nicht so oft ändert.

5.1.3 Tuning von UNIX-Workstations.

Bereits die ersten Durchsatzmessungen bestätigten, daß die bei den Workstations *default*-mäßig eingestellten TCP/UDP/IP-Parameter nicht optimal sind und die Transferleistung erheblich einschränken. Die Änderungen dieser Parameter haben einen gravierenden Einfluß auf die erreichbare *Performance*. Um optimale Durchsatzraten zu erzielen, müssen die Workstations einem *Tuning* unterzogen werden. Dabei spielt vor allem die Größe der Systempuffer eine sehr wichtige Rolle. Mit der Vergrößerung dieser Puffer läßt sich die *Performance* u.U. um über 100% steigern. Einen großen Einfluß auf die *Performance* hat auch die *Default*-MSS-Größe, die die Paketlänge bei der Kommunikation mit Rechnern außerhalb des lokalen Netzes begrenzt. Unterstützen die Workstations den *Path-MTU-Discovery*-Mechanismus nicht, kann man, um bessere *Performance* zu erreichen, die *Default*-MSS erhöhen.

Die o.g Anpassungen können bei SUN-Workstations mit dem `ndd`-Kommando und bei IBM-Rechener mit dem `no` Programm vorgenommen werden. Bei DEC-Machinen benutzt man dagegen den *Kernel-debugger* `dbx`.

[1] IMPC – Internet Control Message Protocol

5.2 Messungen im KFA-Testbett

5.2.1 Test-Umbgebung

Das ZAM verfügt derzeit über ein lokales ATM-Netz mit zwei ATM-*Switches* und einigen mit ATM-Adaptern ausgestatteten Workstations. Dieses lokale ATM-Netz wird mit Hilfe eines Cisco-7000-Routers (zam002), der mit zwei ATM-Karten ausgerüstet ist, über ein FDDI-Netz und einen weiteren Cisco-7000-Router (zam047) an den KFA-FDDI-*Backbone* angeschlossen. Die genaue Topologie des lokales ATM-Netzes kann man Bild 5.1 entnehmen.

Alle Geräte im Testbett unterstützen sowohl PVCs als auch SVCs. Für den Auf- und Abbau der Wählverbindungen (SVCs) wird die UNI-3.0-Signalisierung verwendet.

Alle Workstations unterstützen das *Classical*-IP-Modell und können in einer gemischten Umgebung sowohl mit PVCs als auch SVCs arbeiten. Einige davon, wie z.B. SUN-Maschinen, können außer der normalen Client-Funktionalität auch die Rolle des ATMARP-Servers übernehmen.

Für die Tests mit der LAN-Emulation standen der Cisco-7000-Router und lediglich zwei SUN-Workstations zur Verfügung, die als einzige im Testbett mit der neuen Softwareversion die LAN-Emulation unterstützen können.

Auf dem physikalischen ATM-Netz wurden zwei IP-Subnetze definiert. Einerseits wurde ein logisches IP-Subnetz (LIS) mit allen Workstations und dem Cisco-Router für die *Classical*-IP-Umgebung definiert. Andererseits wurde ein emuliertes LAN (ELAN) für die LAN-

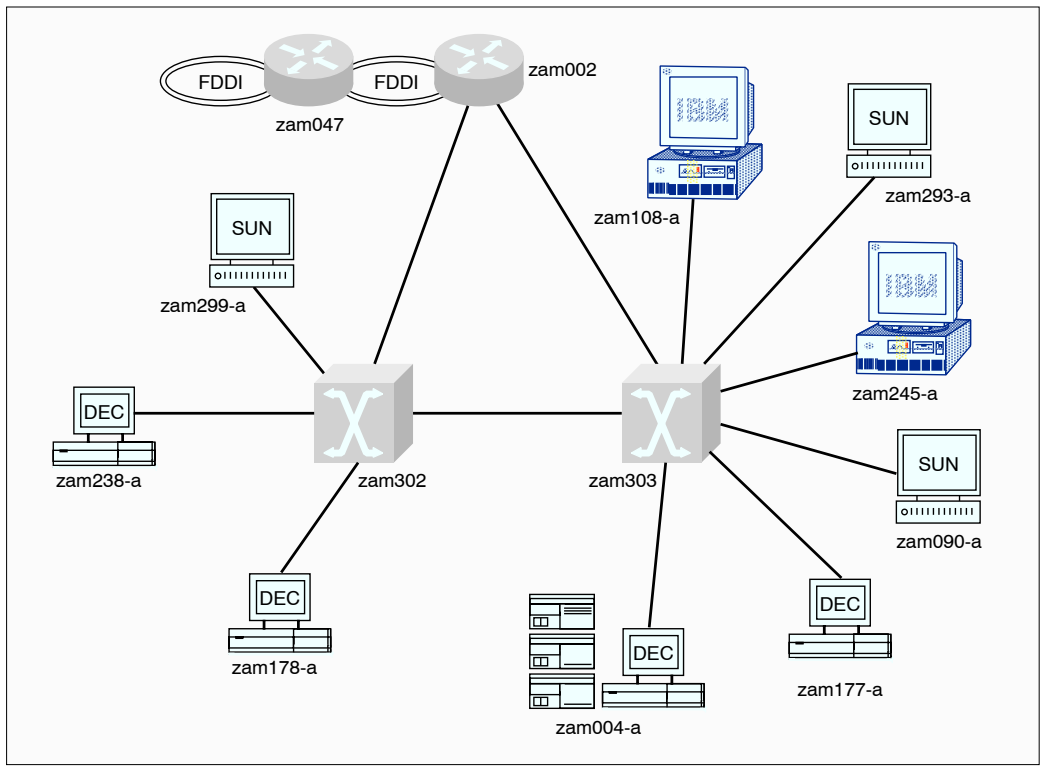


Bild 5.1: KFA – ATM Testbett

Emulation aus den zwei SUN-Workstations und dem Cisco-Router gebildet.

Beide IP-Subnetze kommunizieren sowohl miteinander als auch mit traditionellen Netzen über den Cisco-Router, für den im lokalen ATM-Netz auch einige andere Aufgaben hinzukommen. Er übernimmt die Rolle des ATMARP-Servers in der *Classical*-IP-Umgebung, und außerdem wurde er als LECS, LES und BUS für das emulierte LAN eingesetzt.

Alle Workstations (außer zam238) verfügen neben der ATM-Karte zusätzlich über eine normale 10Mbps-Ethernet-Anbindung ans KFAnet.

Außerdem verfügt ZAM über einen ATM-Analyzer der Firma Telenex, der sich bei der Problemsuche als sehr hilfreich erwiesen hat. Dieser ermöglicht, den ATM-Verkehr auf verschiedenen Protokollebenen zu beobachten, eine Endstation zu emulieren sowie verschiedenen Verkehrsstatistiken zu erstellen. Er besitzt *Interfaces* für OC-3s als auch für E3-Leitungen. Ohne ihn wäre die Konfiguration sowie das Ausprobieren verschiedener Optionen erheblich erschwert oder gar unmöglich geworden.

Die im Test verwendeten Systeme haben folgende Eigenschaften:

- **zam302, zam303**
  Cisco LightStream 100
  UNI 3.0/3.1, ILMI, IISP.

- **zam002**
  Cisco 7000 mit zwei ATM-Karten für SONET OC-3c (155.52 Mbps).
  Einsetzbar als ATMARP-Server, LECS, LES und BUS.

- **zam299, zam090**
  SPARCstation 20 MP Model 502, zwei 50MHz SuperSPARC CPUs, 32 MB, Solaris 2.4
  SunATM-155 SBus Adapter mit SONET OC-3c Ports, Hardwareversion 1.0,
  Softwareversion 2.0(1.0[2])

- **zam293**
  SPARCstation 5
  SunATM-155 SBus Adapter mit SONET OC-3c Ports, Hardwareversion 1.0,
  Softwareversion 2.0

- **zam177, zam178**
  DEC–Alpha 3000/300
  155.52 Mbps ATMworks 750 TURBOchannel-Adapter (SONET OC-3c)

- **zam238**
  DEC–AlphaStation 200 4/100
  155.52 Mbps ATMworks 350 PCI-Adapter (SONET OC-3c)

- **zam004**
  DEC–AlphaServer 1000 4/266
  155.52 Mbps ATMworks 350 PCI-Adapter (SONET OC-3c)

- **zam108, zam245**
  IBM RS/6000-41T
  100 Mbps Turoways 100 TAXI-Adapter (4B/5B)

[2] Bei einigen Tests wurde bei der zam090 noch die ältere Version 1.0 installiert.

- **ATM-Analyzer**
  INTERVIEW 8750 ATM EXPRESS

5.2.2 Testsoftware

Für die Leistungsmessungen im ATM-Testbett wurde das frei verfügbare `Netperf` [57] verwendet. `Netperf` wurde von Hewlett-Packard entwickelt und basiert auf dem Client/Server-Modell. Möglich sind Messungen des Durchsatzes sowie der Antwortzeiten (*request/response*) für verschiedene Protokolle, u.a. auch für TCP und UDP.

Bei den Messungen erlaubt `Netperf` verschiedene Parameter zu definieren, wie z.B. bei TCP die Größe der *Socket*-Puffer (sowohl beim Sender als auch beim Empfänger), die Nachrichtengröße, die Menge der zu übertragenden Daten oder die Dauer des Tests sowie die Größe des Konfidenzintervalls. Dabei können die Parameter, die den Empfänger (den Server) betreffen, von der Senderseite (Client) aus geändert werden. Dadurch lassen sich die Messungen durch entsprechende Scripts leicht automatisieren.

Wenn der `Netperf`-Client gestartet wird, baut er zunächst eine Kontrollverbindung zum `Netperf`-Server auf der entfernten Maschine auf. Diese Verbindung wird zur Übertragung von Konfigurationsinformationen und Ergebnissen von und zu der anderen Maschine benutzt. Es werden die TCP-Parameter übermittelt und danach die *Socket*-Optionen (vor allem Größe der Puffer) jeweils mit Hilfe von `setsocopt`() auf beiden Maschinen gesetzt. Anschließend baut `Netperf` zu dem Server eine zweite, für den jeweilgen Test passende Verbindung auf, die für die tatsächlichen Messungen benutzt wird.

Aus den zahlreichen Testmöglichkeiten wurden drei Arten von Tests gewählt: TCP-*Stream*-Test, UDP-*Stream*-Test und TCP-*Request/Response*-Test.

5.2.3 Theoretische Durchsatzgrenzen

Die Übertragungsrate eines OC-3-Kanals beträgt 155.52 Mbps. Durch die Übertragung der ATM-Zellen in den SONET-Übertragungsrahmen sowie durch den gesamten Protokoll-*Overhead* ist die effektive Bandbreite, die bei Verwendung von TCP/IP für die Übertragung der Nutzdaten zur Verfügung steht, etwas geringer.

In der *Classical*-IP-Umgebung, bei einer MTU-Größe von 9180 Bytes, beträgt die maximale Nachrichtenlänge (MSS, *Maximum Segment Size*) 9140 Bytes, und der *Overhead* bildet sich wie folgt (siehe Bild 5.2):

ATM *Layer Overhead*	9.4%
AAL 5 *Overhead*	0.3%
LLC/SNAP-Verpackung (RFC1483)	0.09%
IP *Header*	0.2%
TCP *Header*	0.2%

Zur Übertragung einer 9140 Bytes langen Nachricht werden also 192 ATM-Zellen verwendet, und es werden insgesamt $192 * 53 = 10176$ Bytes transportiert, was einen *Overhead* von 10.2% darstellt.

Dazu kommt noch der SONET-Transport und SONET-*Path-Overhead*, so daß von den 155.52 Mbps bei TCP/IP nur ungefähr 134 Mbps nutzbar sind.

Bei der LAN-Emulation kommt noch ein zusätzlicher *Overhead* von 16 Bytes dazu, der durch die Verpackung der Ethernet-Pakete (siehe Bild 4.9) entsteht.

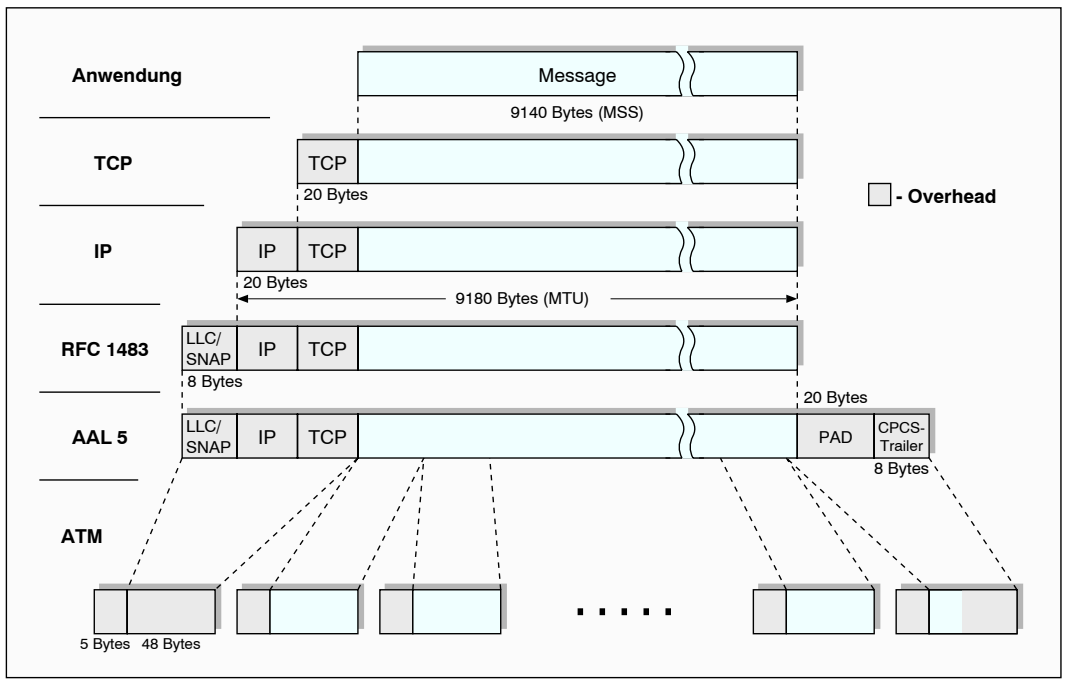


Bild 5.2: Protokoll-*Overhead*

5.2.4 Meßergebnisse

Die TCP-*Performance* hängt von vielen Parametern ab. Dabei handelt sich vor allem, wie schon oben angedeutet, um die Größe der Puffer, die Größe der transportierten Pakete sowie um die gesamte Nachrichtenlänge der zu übertragenden Daten. Bei den Puffern wird aber noch weiter zwischen den *Socket*-Puffern und den Systempuffern[3] unterschieden. Um die Abhängigkeiten und den Einfluß der Parameter auf die Kommunikationsleistung zu untersuchen, wurde eine Reihe von Tests durchgeführt, wobei jeweils ein oder zwei Parameter geändert wurden.

Zuerst wurde der Durchsatz in Abhängigkeit von der Größe der *Socket*-Puffer und der Nachrichtenlänge der zu übertragenden Daten gemessen. Die *Socket*-Puffer der sendenden und der empfangenden Maschine waren bei jeder Messung gleich groß. Für den Test kamen zwei gleichartige schon einem *Tuning* unterzogene SUN-Maschinen zum Einsatz, bei denen die TCP-Systempuffer mit dem `ndd`-Kommando auf 64 kBytes eingestellt wurden. Die gemessenen Werte sind in Bild 5.3 dargestellt und zeigen, daß die besten Durchsatzraten erst bei größerer Nachrichtenlänge sowie bei großen *Socket*-Puffern erreicht werden.

Anschließend wurde der Einfluß der zwei Pufferarten – der TCP-Systempuffer und der *Socket*-Puffer – auf die *Performance* genauer untersucht. Dazu wurden mit den gleichen SUN-Maschinen zwei Meßreihen durchgeführt.

Bild 5.4 zeigt die Ergebnisse der ersten Messungen, bei denen zunächst die Größen der TCP-Systempuffer jeweils mit dem `ndd`-Kommando variiert wurden. Die *Socket*-Puffer hingegen waren bei beiden Maschinen stets gleich und wurden auf 64 kBytes gesetzt. Die gewählte Nachrichtengröße betrug bei allen Messungen 9136 Bytes, so daß für jeden Datenblock nur

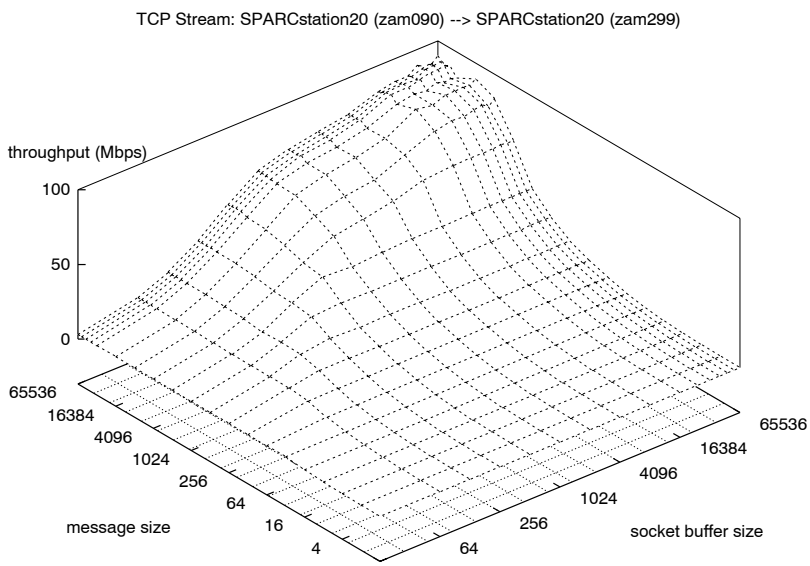


Bild 5.3: Durchsatzrate SPARCstation20 zur SPARCstation20 (receive/send buffer = 65535)

[3] Bei den SUNs können diese mit dem `ndd`-Kommando geändert werden.

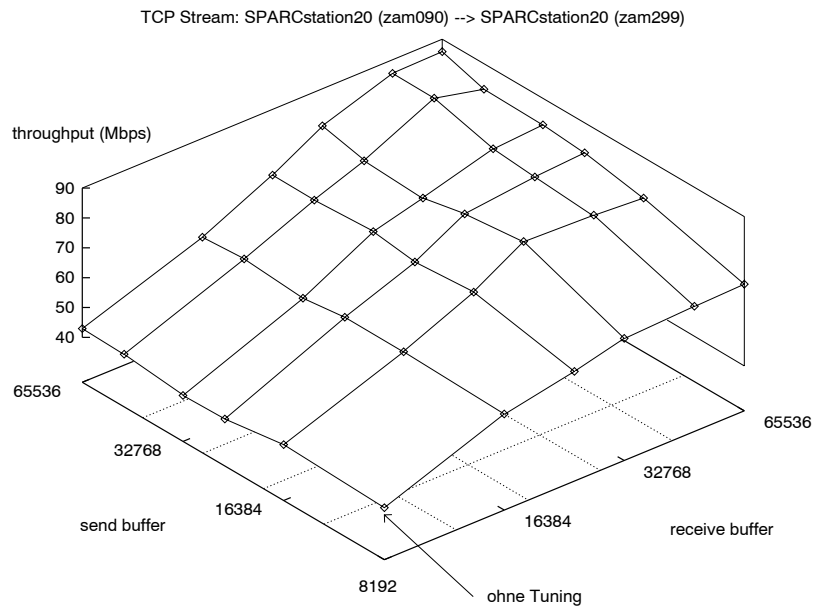


Bild 5.4: Durchsatzrate bei verschiedenen Größen der Systempuffer. (receive/send socket buffer=65536, message length=9136)

ein Paket geschickt werden mußte. Es läßt sich hier deutlich erkennen, welch bedeutenden Einfluß die Größe der TCP-Systempuffer auf den Durchsatz hat. Bei einer *Default*-Größe von 8192 bytes, sowohl beim Sender als auch beim Empfänger, ergab sich eine Datenrate von nur 42 Mbps. Im Gegensatz dazu wurde bei den größtmöglichen Systempuffern von 64 kBytes auch eine größte Durchsatzrate von 85 Mbps erreicht, was eine Steigerung der *Performance* um mehr als 100% darstellt. Außerdem kann man beobachten, daß die Größe des Empfangspuffers eine viel größere Wirkung auf die *Performance* hat als die Größe des Sendepuffers. Der Durchsatz steigt viel schneller mit wachsenden Empfangspuffer.

Der Einfluß der zweiten Pufferart (der *Socket*-Puffer) auf die *Performance* ist in Bild 5.5 dargestellt. Hier waren die TCP-Systempuffer auf beiden Maschinen gleich, und stattdessen wurden die *Socket*-Puffer mit `Netperf` und `setsocopt`() geändert. Die Systempuffer wurden auf den maximalen (und wie die vorherigen Messungen zeigen auch bestmöglichen) Wert von 64 kBytes gesetzt. Die Nachrichtenlänge betrug bei allen Messungen wieder 9136 Bytes. Die Meßergebnisse bestätigen die Vermutungen, daß für die *Performance* praktisch nur die Größe der *Socket*-Empfangspuffer eine Rolle spielt und der Einfluß der Größe der *Socket*-Sendepuffer in diesem Fall kaum spürbar ist. Die Größe des *Sliding-Window* (siehe Kap. 5.1.1 auf Seite 68) wird vor allem durch die Größe der *Socket*-Empfangspuffer beeinflußt. Bei kleinerem Puffer wird das Fenster zu klein, um gute Resultate zu erzielen.

Wie die zwei gerade beschriebenen Messungen zeigen, hängt der Durchsatz weniger von der Größe der Sendepuffer (*Socket*- oder Systempuffer), sondern vor allem von der Größe der Empfangspuffer ab. Es stellt sich die Frage, wie sich der Durchsatz ändert, wenn man beide Empfangspuffer bei fester Größe der Senderpuffer gleichzeitig variiert. Um dies zu untersuchen, wurde ein weiterer Test durchgeführt, dessen Ergebnisse in Bild 5.6 dargestellt sind.

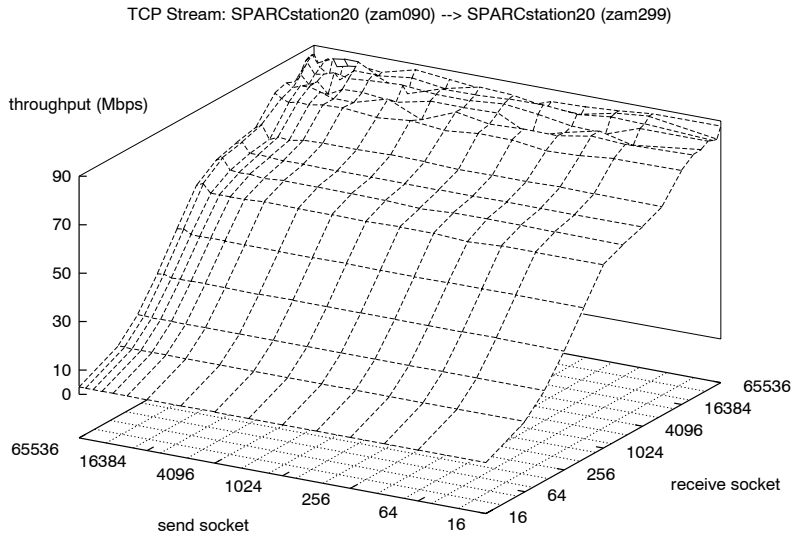


Bild 5.5: Durchsatzrate bei verschiedenen Größen der *Socket*-Puffer. (receive/send buffer=65536, message length=9136)

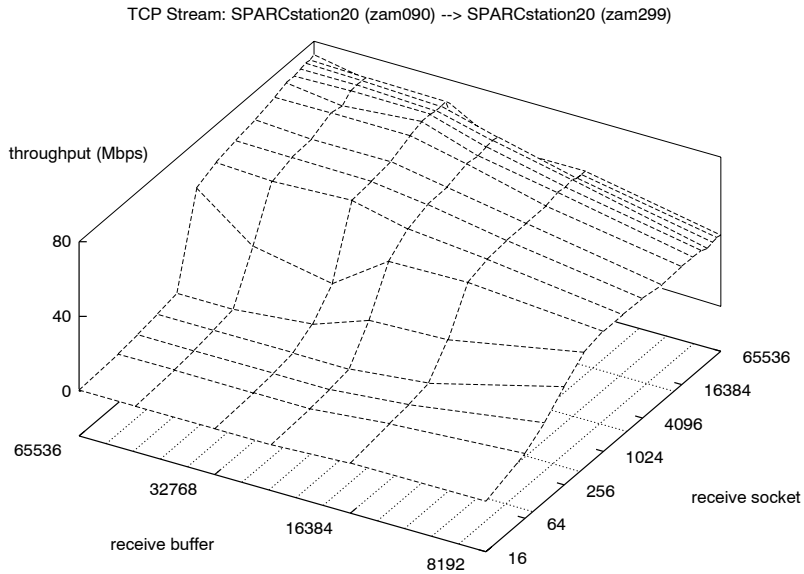


Bild 5.6: Durchsatzrate bei verschiedenen Empfangspuffern. (send buffers=65536, message length=9136)

Bei den Messungen waren sowohl der *Socket*- als auch der System-Sendepuffer auf 32 kBytes eingestellt, und die Länge der gesendeten Nachrichten betrug wieder 9136 Bytes.

Außer der Größe der Puffer hat auch die Nachrichtenlänge der zu übertragenden Daten einen großen Einfluß auf die *Performance*, was schon aus Bild 5.3 ersichtlich ist. Aus diesem Grund wurden bei den nächsten Messungen die Durchsatzraten in Abhängigkeit von der Nachrichtenlänge untersucht. Bild 5.7 zeigt die zunächst zwischen den beiden SUN-Maschinen gemessenen Werte. Die Messungen wurden jeweils bei drei verschiedenen Größen der Systempuffer durchgeführt, zuerst mit *default*-Werten (8KB), dann mit maximaler Größe (64KB) und schließlich noch mit einer Größe von 32KB. Zum Vergleich ist in das Bild auch der erreichbare Durchsatz über 10-Mbps-Ethernet aufgetragen. Die *Socket*-Puffer waren während aller Messungen gleich groß und wurden auf den maximalen Wert von 64 kBytes gesetzt.

Hier wird noch einmal der Unterschied in der *Performance* zwischen Maschinen mit und ohne *Tuning* deutlich. Während das *Tuning* bei kleineren Nachrichtenlängen nicht so spürbar ist, verdoppelt sich die *Performance* bei größeren Längen der Nachrichten. Außerdem kann man den kontinuierlichen Anstieg des Durchsatzes bis zu einer bestimmten Nachrichtengröße beobachten. Danach bleibt der Durchsatz auf ungefähr gleichem Niveau, schwankt jedoch ein wenig von Nachrichtengröße zu Nachrichtengröße.

Um die Schwankungen genauer zu untersuchen, wurden im nächsten Test Durchsatzraten für Nachrichten gemessen, deren Länge im Bereich von 3500 bis 10000 Bytes jeweils in kleinen Schritten (4 Bytes) anstieg. Wie man Bild 5.8 entnehmen kann, steigt der Durchsatz kontinuierlich an, solange die Länge der zu übertragenen Nachricht die maximale Segmentgröße nicht übersteigt, welche in diesem Fall 9140 Bytes beträgt. Dies läßt sich dadurch erklären, daß der Bearbeitungsaufwand in den Stationen weitgehend unabhängig von der Größe des Pa-

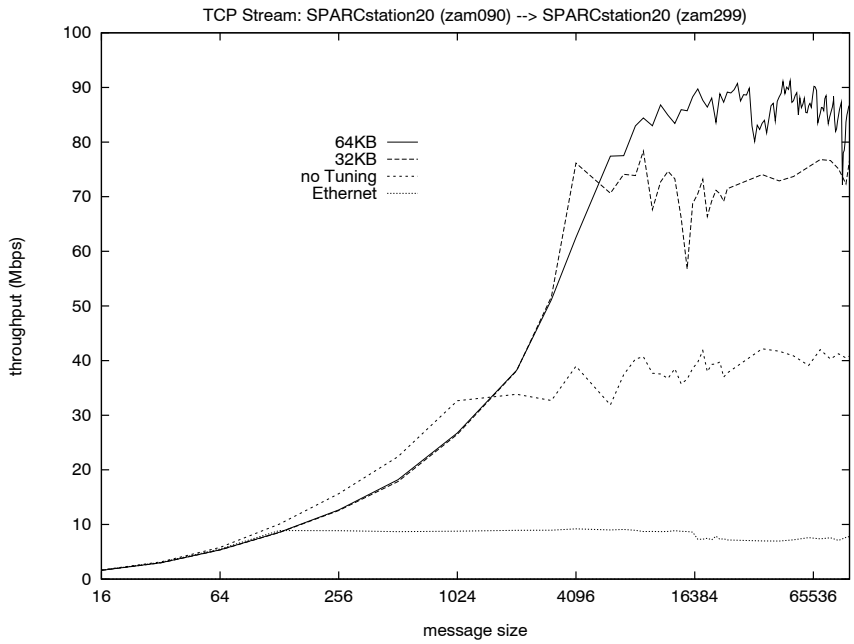


Bild 5.7: Einfluß des *Tuning* auf den Durchsatz zwischen zam090 und zam299. (socket buffer = 65536)

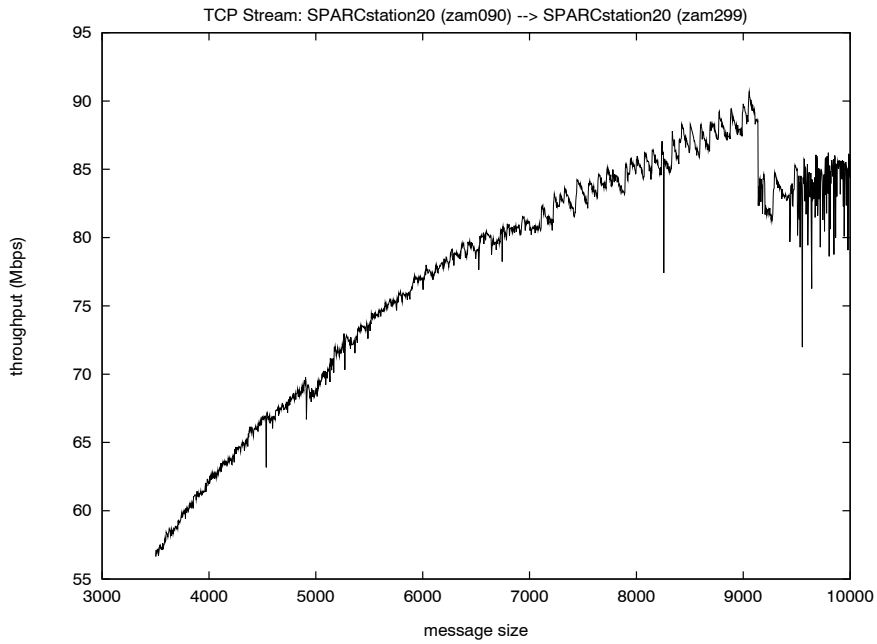


Bild 5.8: Kontinuierlicher Anstieg der *Performance* in Detail

ketes ist. Die *Performance* steigt also an, solange nur ein Paket für jede Nachricht geschickt werden muß. Müssen zwei Pakete für jede Nachricht gesendet werden, wird der Kommunikationsaufwand größer und die *Performance* fällt um ca. 7.5 % ab. Bild 5.8 zeigt auch sehr gut den Verlauf der Anstiegskurve, der Sägezähnen ähnelt. Dieses Phänomen, das auch schon in [59] beobachtet wurde, läßt sich auf die Implementierungsdetails bei der SUN-Maschine zurückführen.

Vergleichbare Ergebnisse mit den oben beschriebenen ergaben sich auch bei ähnlichen Messungen mit den anderen Maschinen im Testbett. Dabei stellte sich allerdings die starke Abhängigkeit des erreichbaren Durchsatzes von der Leistungsfähigkeit der verwendeten Systeme heraus.

Um die Kommunikationsleistung der verschiedenen Maschinen im ATM-Testbett zu vergleichen, wurden in den folgenden Tests jeweils unterschiedliche Maschinen auf der Sender- und Empfängerseite eingesetzt.

In Bild 5.9 wird zunächst die Sendeleistung der Maschinen verglichen. Als Empfänger wurde die stärkste Maschine, die zam004 (AlphaServer), eingesetzt, um die Zellverluste weitgehend zu vermeiden und die Sender dadurch nicht aufzuhalten.

Bild 5.10 vergleicht dagegen die gleichen Maschinen in ihrer Empfangsleistung. Bei allen Messungen wurde ein relativ starker Sender, und zwar eine SPARC-Maschine (zam090), eingesetzt, die jeweils zusammen mit verschiedenen Empfängern getestet wurde.

Als nächstes wurden zwei Maschinen getestet, die über einen Router kommunizieren, um dessen Einfluß sowie den Einfluß der *Default*-MSS-Größe auf die *Performance* zu untersuchen. Dazu wurde die Konfiguration des ATM-Testbetts kurzzeitig geändert. Es wurde auf demselben physikalischen ATM-Netz ein zweites *Classical*-IP-Subnetz gebildet. Es bestand nur aus einer SUN-Maschine (zam299), die über den Router (zam002) mit dem ursprünglichen *Classical*-IP-Subnetz kommuniziert. Beide *Classical*-IP-Subnetze wurden über verschiedene ATM-

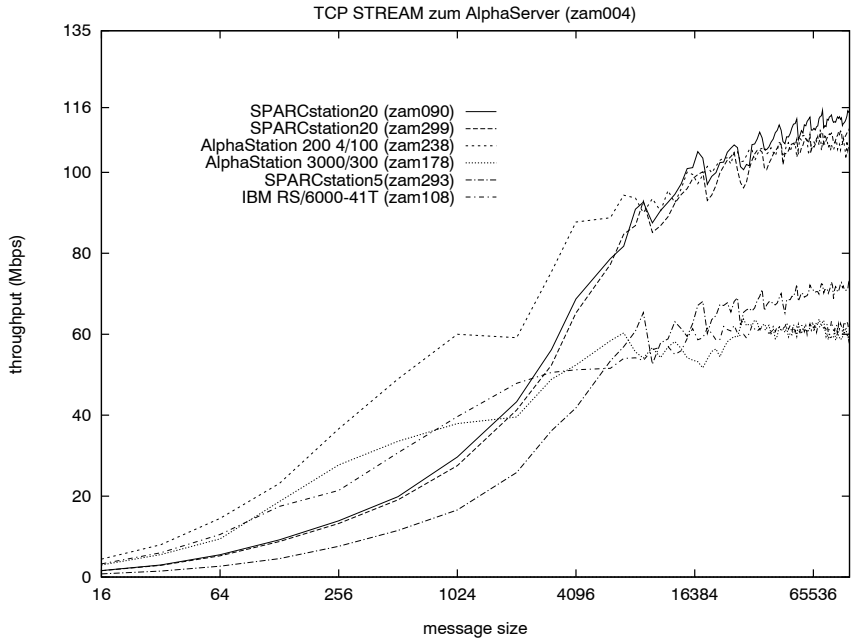


Bild 5.9: Vergleich der Sendeleistung der verschiedenen Maschinen. (receive/send buffer = 65535, socket buffers = 65536)

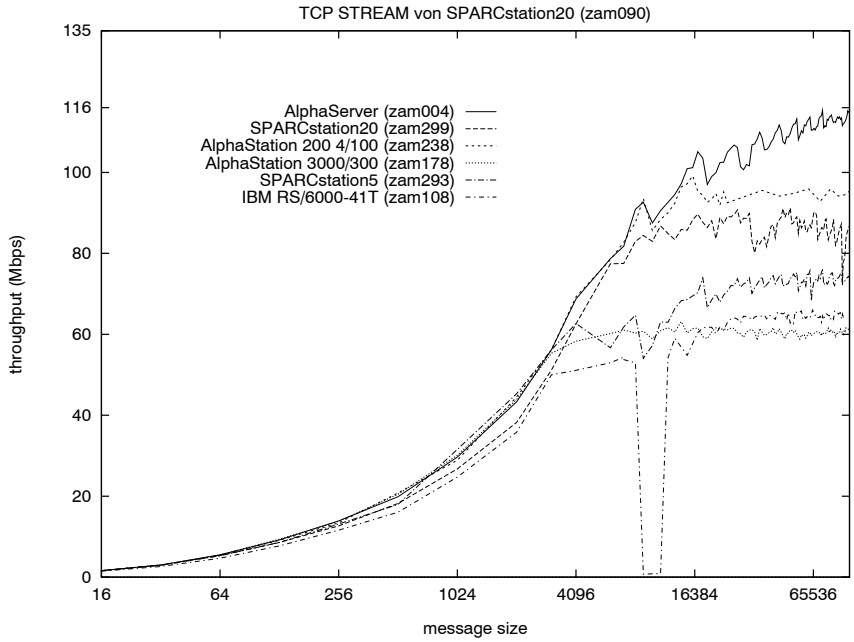


Bild 5.10: Vergleich der Empfangsleistung der verschiedenen Maschinen. (receive/send buffer=65536, socket buffers = 65536)

Interfaces an den Router angeschlossen. Es wurde zwischen einer SUN und dem AlphaServer (zam004) bei zwei verschiedenen MSS-Größen gemessen. Die dabei erzielten Durchsatzraten sind in Bild 5.11 dargestellt, in dem zum Vergleich noch die Durchsatzrate aufgetragen ist, die zwischen diesen Maschinen bei direkter Verbindung (ohne Router) gemessen wurde.

Neben den Tests in den *Classical*-IP-Subnetzen wurde auch die *Performance* zwischen den beiden SUN-Maschinen in der LAN-Emulation-Umgebung untersucht. In Bild 5.12 sind die Durchsatzraten zusammengefaßt, die zwischen den beiden SUNs bei verschiedensten Konfigurationen gemessen wurden. Man sieht, daß u.a. aufgrund der MTU-Größe von 1500 Bytes in dem emulierten LAN, nur Durchsatzraten bis etwa 22 Mbps erreichbar sind, was ca. 70 % unter der *Performance* liegt, die zwischen diesen Maschinen in der *Classical*-IP-Umgebung gemessen wurde.

Zum Schluß wurden noch zwei andere Testarten durchgeführt, und zwar UDP-*Stream*-Test und TCP-*Request/Response*-Test. Für die Messung der UDP-*Performance* wurde von der zam299 (SPARC20) zur zam178 (Alpha3000/300) gesendet. Es wurde jeweils die Übertragungsrate beim Sender und beim Empfänger gemessen. Alle Ergebnisse sind noch zusammen mit den zwischen diesen Maschinen erzielten TCP-Durchsatzraten in Bild 5.13 aufgetragen.

Man sieht, daß zuerst beide Durchsatzraten gleich sind und kontinuierlich ansteigen, weil keine Zell-/Paketverluste auftreten. Ab einer bestimmten Übertragungsrate ist jedoch der Empfänger nicht leistungsfähig genug, um die ankommenden Pakete schnell genug zu bearbeiten, und beginnt, Zellen zu verwerfen. Die Empfängerrate fällt dadurch beinahe linear mit wachsender Paketgröße. Die Senderate steigt jedoch weiter und nimmt erst wieder ab, wenn die UDP-Nachrichtenlänge die MSS-Größe (9140 Bytes), d.h die Paketgröße, übersteigt. Sie fällt dabei sogar um 40% ab und liegt damit im Bereich, in dem noch keine Verluste

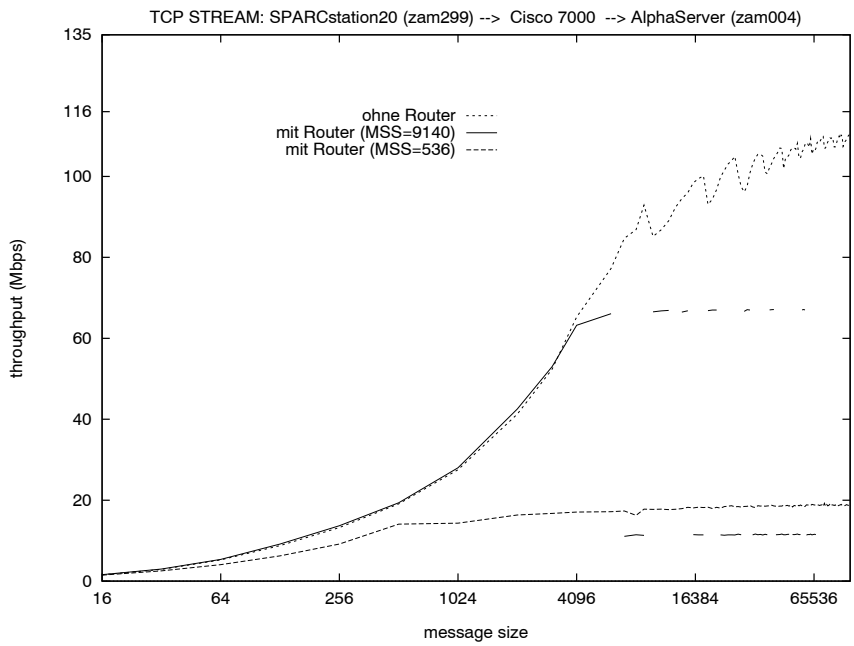


Bild 5.11: Durchsatzraten bei der Kommunikation zwischen zwei *Classical*-IP-Subnetzen (receive/send buffer=65536, socket buffers=65536)

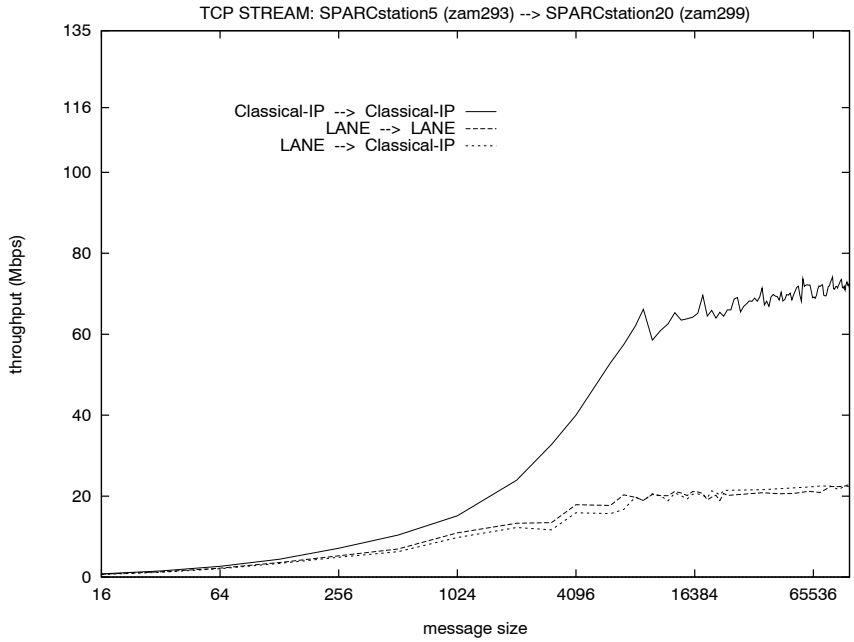


Bild 5.12: ***Performance*** **in der** ***Classical*****-IP- und LAN-Emulation-Umgebung. (receive/send buffer=65536)**

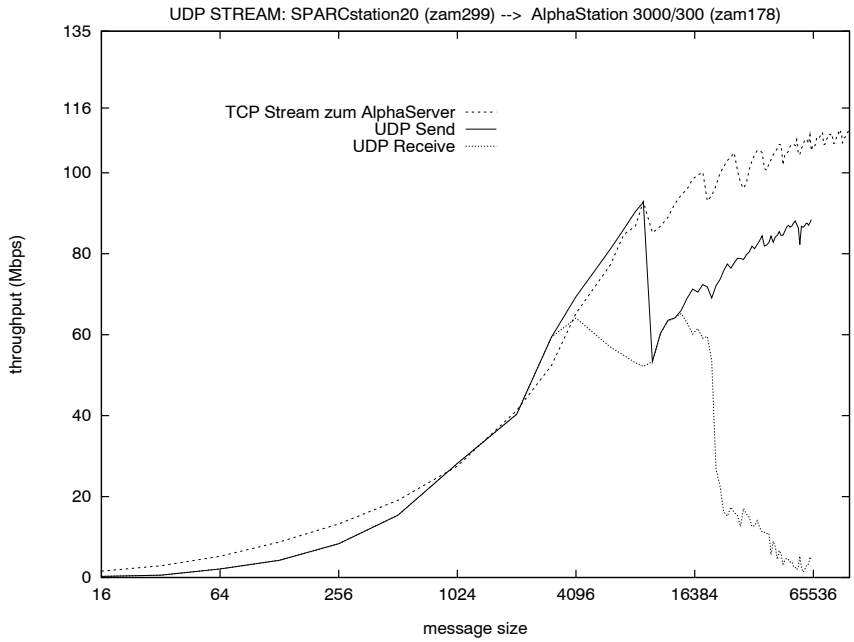


Bild 5.13: UDP-*Performance* (receive/send buffer=65536)

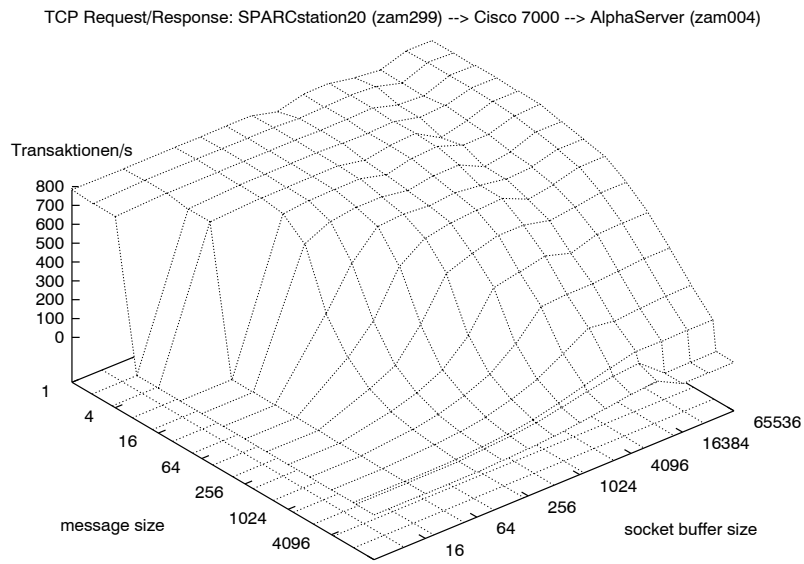


Bild 5.14: TCP-Antwortverhalten

beim Empfänger auftreten. Dadurch steigen beide Durchsatzraten mit wachsender Nachrichtenlänge noch einmal kurz zusammen an. Wird die Nachricht aber länger, dann sinkt die Empfängerrate wieder und fällt schließlich drastisch ab, sobald für eine UDP-Nachricht drei Pakete geschickt werden müssen. Wächst die Nachrichtenlänge noch weiter, dann steigt auch die Wahrscheinlichkeit, daß aufgrund eines einzigen Zellverlustes die ganze UDP-Nachricht verlorengeht, und der Durchsatz beim Empfänger bricht zusammen.

Die Ergebnisse des letzten Tests, bei dem das Antwortverhalten untersucht wurde, sind in Bild 5.14 zu sehen. Es wurde die Anzahl der Transaktionen, die jeweils aus einem *Request* und einem *Response* bestehen, in Abhängigkeit von der Nachrichtenlänge und der Größe der *Socket*-Puffer pro Sekunde gemessen. Die besten Resultate wurden bei kleineren Nachrichtenlängen erzielt. Der maximale Wert von 800 Transaktionen/s läßt auf eine *Round-Trip-Time* von ca. 1,25 ms schließen.

5.3 ATM-Netz bei Überlast

Verschiedene Simulationen und analytische Studien zeigen, daß durch die Segmentierung der TCP-Pakete in ATM-Zellen die Wahrscheinlichkeit, daß durch einen Zellverlust ein Paket verloren geht, enorm ansteigt. Der Durchsatz nimmt drastisch mit wachsender Paketgröße ab, wenn bei Überlast die Zellen willkürlich und unkorreliert verworfen werden.

In *Classical*-IP-Netzen wird bei einer MTU-Größe von 9180 Bytes ein TCP-Paket in 192 Zellen zerlegt. Geht eine davon auf dem Weg zum Empfänger verloren, muß das ganze Paket neu übertragen werden. Bild 5.15 zeigt an einem Beispiel, wie solch eine Situation leicht zu einem Zusammenbruch des Durchsatzes führen kann. Es wird ein ATM-*Switch* gezeigt, über den mehrere TCP-Verbindungen abgewickelt werden. Die TCP-Pakete werden in den *Hosts* zerlegt und bilden beim *Switch* einen Zellstrom. Wenn der Puffer im *Switch* im Falle einer Überlastsituation überläuft, werden neu ankommende Zellen verworfen. Da die zu den verschiedenen TCP-Paketen gehörenden Zellen abwechselnd an dem *Switch* ankommen, kann es leicht passieren, daß dabei jeweils ein paar Zellen aus jedem Paket weggeworfen werden. Als Resultat kommen bei den Empfängern unvollständige Pakete an, bei denen unter Umständen nur eine oder wenige Zellen fehlen. Das hat zur Folge, daß alle Pakete wiederholt werden müssen. Durch die wiederholte Übertragung der Pakete erhöht sich die Verkehrslast und die Situation kann sich leicht wiederholen. Das kann, wie einige Berichte zeigen, bei der verwendung der UBR-Dienstklasse zu einem kompletten Zusammenbruch des Durchsatzes führen, bei dem kaum eine Kommunikation möglich ist.

5.3.1 Packet-Discard-Strategien

Beim Übertragen von TCP-Paketen über ATM führt der Verlust einer Zelle dazu, daß die restlichen, zu dem gleichem Paket gehörenden Zellen, beim Empfänger verworfen werden. Diese Zellen verbrauchen aber, bevor sie das Ziel erreichen, unnötig die wichtigen Resourcen im Netz wie Bandbreite und Pufferplatz, und können dadurch weitere Überlast verursachen. Es wäre von Vorteil, wenn die Zellen, die zu bereits zerstörten Paketen gehören und beim Ziel sowieso verworfen werden, schon im *Switch* vernichtet würden, und dadurch nicht unnötig das Netz belasten.

Romanov und Floyd haben in [44] zwei Verfahren vorgestellt und untersucht, die dieses Ziel verfolgen und den Durchsatz bei einer Überlast im Netz verbessern sollen:

- **Partial Packet Discard** (PPD)
  Nachdem im *Switch* eine Zelle verworfen werden muß, werden auch alle nachfolgenden Zellen vernichtet, die zu dem gleichen Paket gehören. Damit erreicht man aber nur bis zu einem gewissen Grad eine Steigerung des Durchsatzes, was vor allem dadurch beschränkt wird, daß der *Switch* erst dann anfängt, Zellen zu verwerfen, wenn der Puffer überläuft. Die erste verworfene Zelle kann zu einem Paket gehören, desen Zellen schon im Puffer gespeichert sind oder bereits über den überlasteten *Link* übertragen wurden. Solange diese Zellen erst beim Empfänger weggeworfen werden, wird daher immer noch ein Teil der Bandbreite und des Pufferplatzes verschwendet.

- **Early Packet Discard** (EPD)
  Der effektive Durchsatz von TCP über ATM kann noch mehr verbessert werden, indem im *Switch* ganze Pakete verworfen werden. Wird ein EPD-Verfahren verwendet, dann werden im *Switch* vollständige TCP-Pakete verworfen, falls die Gefahr besteht, daß der

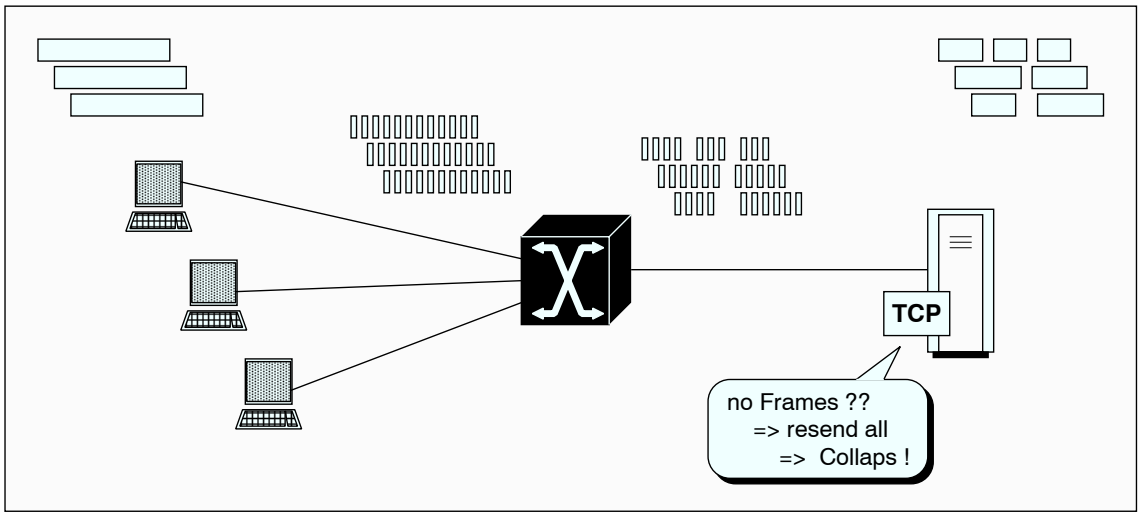


Bild 5.15: Zellverluste in einem überlasteten ATM-Netz

Puffer überläuft. Der EPD-Algorithmus verhindert direkt das Übertragen von nutzlosen Zellen über einen überlasteten *Link* und reduziert die gesamte Anzahl von zerstörten Paketen.

Wird für die Übertragung der Daten AAL 5 verwendet, wie das bei *Classical*-IP und LAN-Emulation der Fall ist, können beide Algorithmen recht einfach implementiert werden. Bei AAL 5 wird das gleichzeitige Multiplexen verschiedener AAL-Verbindungen innerhalb einer ATM-Verbindung nicht unterstützt, und die Paketgrenzen können leicht mit Hilfe der PT-Feldes im Zell-*Header* erkannt werden.

Sowohl PPD als auch EPD arbeiten auf der VC-Basis und werden jeweils, falls nötig, für die betreffende VC verwendet. Soll für eine ATM-Verbindung eines der beiden Verfahren verwendet werden, wird das in den Signalisierungsnachrichten beim Verbindungsaufbau angezeigt, wie dies im ATM-Forum-Dokument '*Signaling support for frame discard*' [46] beschrieben ist.

PPD und EPD können gleichzeitig auch mit anderen Verfahren zur Verkehrs- und Überlaststeuerung (wie z.B. ABR) verwendet werden. Eine Möglichkeit, bessere Verkehrskontrolle zu erreichen, bietet z.B. die Verwendung von EPD in Verbindung mit der RED-Strategie (*Random Early Detection*), die in [45] vorgestellt wurde. Beim RED-Verfahren wird die durchschnittliche Größe der Warteschlange im *Switch* berechnet, um eine angehende Überlastsituation zu entdecken und entsprechend darauf reagieren zu können.

Wird EPD zusammen mit RED verwendet, werden zwei Schwellen für die Warteschlange im *Switch* definiert, auf deren Überschreiten jeweils entsprechend reagiert wird. Wird die EPD-Schwelle überschritten, beginnt der *Switch*, alle ankommenden Pakete zu verwerfen. Wird aber zuerst die RED-Schwelle (für die durchschnittliche Warteschlangengröße) überschritten, werden vom *Switch* nur zufällig ausgewählte Pakete verworfen, und die Senderstation kann optional veranlaßt werden, ihre End-to-end Flußsteuerung zu benutzen, um die Last im Netz zu reduzieren.

5.3.2 Available Bit Rate (ABR)

Heutzutage kommt in den ATM-LANs die UBR-Dienstklasse mit *Best-Effort* zum Einsatz. Es werden dabei keine Maßnahmen zur Verkehrs- (*Traffic Control*) und Überlaststeuerung (*Congestion Control*) verwendet. Die im vorherigen Kapitel vorgestellten *Packet-Discard*-Strategien können die geschilderten Überlastprobleme in ATM-Netzen allerdings nur begrenzt entschärfen. Eine bessere Lösung des Problems kann erst mit der ABR-Dienstklasse erreicht werden.

Die in [47] definierte ABR stellt einige Maßnahmen zur Verkehrs- und Überlaststeuerung auf der ATM-Ebene zur Verfügung. Diese Maßnahmen werden ergriffen, um eine Überlastsituation zu verhindern und die erforderliche Dienstqualität sicherzustellen. Die ABR-Dienstklasse ist für Anwendungen geeignet, die keine besonderen Anforderungen an Bandbreite und Verzögerungen stellen, jedoch eine niedrige Zellverluste fordern. Die benötigte Senderate wird an die verfügbare Bandbreite angepaßt.

Zur Unterstützung der ABR-Dienstklasse in einem ATM-Netz wird eine spezielle Zelle, die sog. *Resource Management* (RM) Zelle benutzt mit der eine *'rate-based'* Verkehrssteuerung realisiert wird. Eine RM-Zelle wird durch die Belegung '110' des PT-Feldes (*Payload Type*) im ATM-*Header* identifiziert (siehe Kap. 2.4 auf Seite 11) und hat eine spezielle Struktur, wie in Bild 5.18 dargestellt. Mit Hilfe der RM-Zellen können Kontrollinformationen zwischen Sender, *Switches* und Empfänger ausgetauscht werden und dadurch kann eine End-to-End-Steuerung realisiert werden.

Die RM-Zellen werden vom Sender generiert. Sie wandern durch das Netz zuerst zum Empfänger und dann wieder zurück zum Sender. Dabei wird ihr Inhalt von den auf dem Weg liegenden Stationen entsprechend der im Netz verfügbaren Resourcen geändert. Der Fluß der RM-Zellen bildet eine Kontrollschleife (*Control loop*). Diese stellt einen *Feedback*-Mechanismus dar, der eine direkte Steuerung der Senderate erlaubt.

Eine ABR-Verbindung kann optional in mehrere ABR-Segmente unterteilt werden, die jeweils getrennt überwacht werden. Die Segmentierung der ABR-Kontrollschleife (*Control Loop*) wird durch die Implementierung von sog. *Virtual-Sources* und -*Destinations* erreicht, so wie das im Bild 5.17 gezeigt wird. Ein *Switching*-Element zwischen einzelnen ABR-Segmenten schließt jeweils eine Kontrollschleife und initialisiert gleichzeitig eine neue, indem es sich als

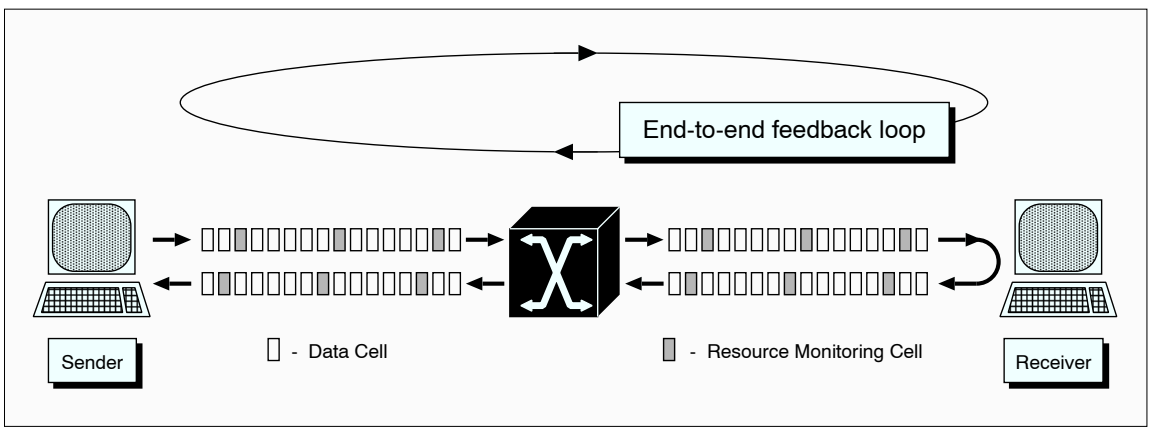


Bild 5.16: Funktionsweise des ABR-Mechanismus

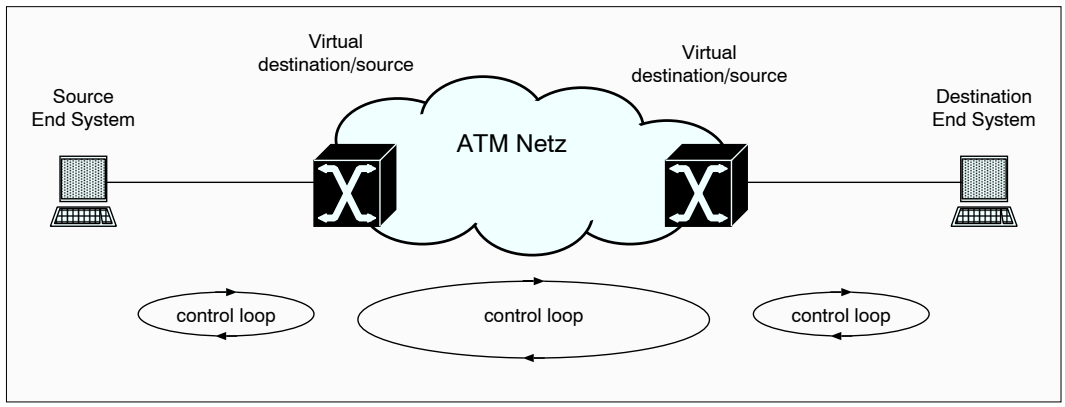


Bild 5.17: Segmentierung der ABR-Kontrollschleife

Ziel- bzw. als neue Quellstation verhält. Die Segmentierung der ABR-Kontrollschleife wird benutzt, um z.B. die Länge der Schleife und die damit verbundene Reaktionszeit des *Feadback*-Mechanismus zu reduzieren. Außerdem wird sie aus Verwaltungsgründen bei Übergängen zwischen privaten und öffentlichen Netzen verwendet.

Weiterhin erlaubt ABR, den schon im Rahmen der UNI-Spezifikation definierten EFCI-Mechanismus (*Explicit Forward Congestion Indication*), mit dem sich ebenfalls eine *Feedback*-Steuerung realisieren läßt, weiterzubenutzen.

Der EFCI-Mechanismus dient zur Mitteilung einer Überlastsituation an die nächsten Vermittlungsstationen. Wenn in einem *Switch* eine Überlastsituation festgestellt wird, wird das EFCI-Bit[4] im ATM-*Header* auf 1 gesetzt.

Soll für eine virtuelle ATM-Verbindung die ABR-Dienstklasse verwendet werden, dann werden zunächst beim Verbindungsaufbau bestimmte ABR-Verkehrsparameter vereinbart. Dabei handelt sich vor allem um die Spitzenzellrate (*Peak Cell Rate*, PCR) und die minimale Zellrate (*Minimum Cell Rate*, MCR), die vom Sender angefordert werden, vom Netz aber geändert werden können. Einige andere Parameter, die z.B. die dynamische Anpassung der Senderate charakterisieren (RIF, RDF, CDF)[5] oder die Anzahl der normalen Zellen pro RM-Zelle (Nrm) angeben, werden direkt vom Netz gewählt.

Wird die MCR von den *Switches* auf dem Weg zum Empfänger unterstützt und können die ABR-Verkehrsparameter ausgehandelt werden, wird der Verbindungsaufbau akzeptiert und der Sender kann mit dem Verschicken der Daten beginnen.

Die Zellen werden mit *Allowed Cell Rate* (ACR) gesendet. Die ACR wird zunächst auf *Initial Cell Rate* (ICR) gesetzt und bewegt sich immer zwischen *Minimum Cell Rate* (MCR) und *Peak Cell Rate* (PCR).

Außer den normalen Zellen schickt der Sender regelmäßig die RM-Zellen (*Resource Management*). Eine RM-Zelle wird jeweils nach (Nrm-1) normalen Zellen gesendet, ggf. auch öfter, falls die ACR relativ niedrig ist. Die RM-Zelle enthält u.a. in dem CCR-Feld (*Current Cell Rate*) die Zellrate, mit der aktuell gesendet wird (ACR), und in dem ER-Feld die gewünschte Senderate, die normalerweise gleich PCR ist (siehe Bild 5.18).

[4] Als EFCI-Bit wird das zweite Bit des *PT*-Feldes im ATM-*Header* betrachtet, wenn das erste Bit dieses Feldes mit 0 belegt ist.

[5] RIF – Rate Increase Factor, RDF – Rate Decrease Factor, CDF – Cutoff Decrease Factor

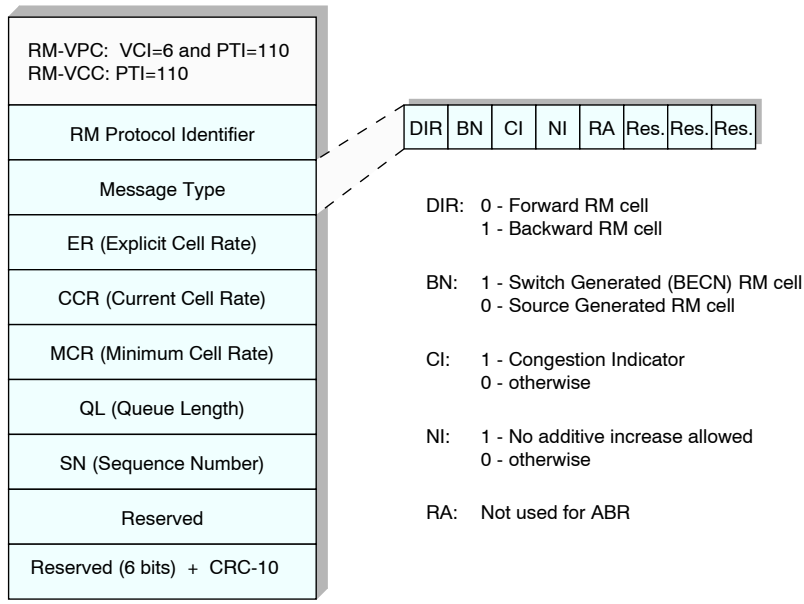


Bild 5.18: Resource Management Cell

Die RM-Zelle wandert durch das Netz und erreicht schließlich das Zielsystem. Dieses ändert das DIR-Bit von '*Forward*' auf '*Backward*' und falls erforderlich, auch einige andere Felder in der empfangenen RM-Zelle. Einerseits muß die Zielstation die ER reduzieren und/oder das CI- bzw. NI-Bit auf 1 setzen, falls sie überlastet ist und die erfordeliche Zellrate aus dem ER-Feld nicht einhalten kann. Andererseits beobachtet die Zielstation das PT-Feld im ATM-*Header* der empfangenen Zellen. Falls vor Empfang der RM-Zelle Zellen mit gesetztem EFCI-Bit empfangen wurden, was auf eine Überlastsituation im Netz hindeutet, setzt die Zielstation in der RM-Zelle das CI-Bit auf 1. Die so geänderte RM-Zelle wird schließlich zum Sender zurückgeschickt.

Die Änderung der RM-Zelle wird falls erfoderlich in erster Linie von *Switches* vorgenommen werden, die sich auf dem Weg zwischen Sender und Empfänger befinden. Sie überwachen die verfügbaren Resourcen im Netz und reagieren als erste auf eine Überlastsituation.

Für die ABR-Dienstklasse sind bei den *Switches* verschiedene Implementierungen erlaubt, die aber eine der folgenden Methoden zur Überlaststeuerung unterstützen müssen:

- EFCI–Markierung
  Die einfacheren *Switches* der früheren Generation unterstützen nur den EFCI-Mechanismus, der zur Mitteilung einer Überlastsituation an weitere Stationen dient. Stellt ein *Switch* eine Überlastsituation fest, wird das EFCI-Bit im Zellkopf gesetzt. Diese Information wird, wie schon oben angedeutet, von der Zielstation ausgewertet und durch den *Feedback*-Mechanismus in einer RM-Zelle zum Sender weitergereicht, der darauf entsprechend reagieren kann.

- *Relativ-Rate*-Markierung (RR)
  Im Falle einer Überlastsituation kann vom *Switch* das CI- oder NI-Bit in den *Forward*- und/oder *Backward*-RM-Zellen auf 1 gesetzt werden, um sicherzustellen, daß die Sen-

destation keinesfalls die Senderate erhöht.

- *Explicit-Rate*-Markierung (ER)
  Die intelligenteren *Switches* überprüfen, ob sie die erlaubte maximale Rate, die in der RM-Zelle im ER-Feld identifiziert ist, unterstützen können. Falls die ER zu hoch ist, wird sie entsprechend den verfügbaren Resourcen auf eine Rate reduziert, die aktuell vom *Switch* eingehalten werden kann.

Darüber hinaus kann ein *Switch* auch selbst RM-Zellen in Rückwärts-Richtung generieren. Das könnte z.B. passieren, wenn ein *Switch* eine RM-Zelle aus der Vorwärts-Richtung empfängt und feststellt, daß die darin identifizierte CCR-Rate größer als die ist, die er momentan einhalten kann. Durch die in Rückwärts-Richtung generierten RM-Zellen wird der Sender direkt (ohne Umweg über den Empfänger) über ein Überlastproblem informiert und kann dadurch schneller reagieren. In den vom *Switch* generierten RM-Zellen muß das DIR-Bit auf '*Backward*', das BN-Bit sowie entweder das CI- oder das NI-Bit auf 1 gesetzt werden.

Nachdem die RM-Zelle durch das Netz gewandert ist und dabei u.U. von den Stationen auf diesem Weg geändert wurde, kommt sie zum Sender zurück. Basierend auf den in dieser RM-Zelle empfangenen Informationen, paßt der Sender seine Sendeparameter an die im Netz vorhandenen Resourcen an.

Die Senderate darf natürlich nur dann vergrößert werden, wenn keine Überlastsituation im Netz festgestellt wurde. Das kann der Sender daran erkennen, daß in der empfangenen *Backward*-RM-Zelle die CI- und NI-Bits nicht gesetzt sind.

Der Sender erhöht die Senderate ACR (*Allowed Cell Rate*) stufenweise, was durch den während des Verbindungsaufbaus festgelegten RIF-Parameter (*Rate Increase Factor*) geregelt wird. Allerdings kann die ACR höchstens bis auf die in der RM-Zelle angegebene erlaubte maximale Rate ER erhöht werden und nie größer als die PCR sein.

In jedem anderen Fall, auch wenn der Sender den erwarteten Kontrollfluß der RM-Zellen nicht zurückempfängt, muß der Sender seine Rate ensprechend erniedrigen. Das Reduzieren der Senderate wird durch eine Reihe von weiteren ABR-Parametern geregelt, die ebenfalls beim Verbindungsaufbau festgelegt werden. Dazu gehört z.B. der RDF (*Rate Decrease Factor*) und der CDF (*Cutoff Decrease Factor*). Auf jeden Fall darf aber die reduzierte ACR nicht kleiner als die MCR sein.

Der oben beschriebene *Feedback*-Mechanismus liefert dem Sender Informationen über die im Netz verfügbare Bandbreite. Er kann dadurch die Zellen mit einer variablen, jedoch kontrolierten Rate senden, die jeweils automatisch an die im Netz vorhandenen Resourcen angepaßt wird. Dadurch werden Zellverluste vermieden, auf die z.B. paketorientierte Protokolle wie TCP besonders empfindlich reagieren.

Es wird erwartet, daß ABR zur den populärsten Dienstklassen in ATM-Netzen gehören wird, vor allem weil sie für den LAN-Verkehr besonders geeignet ist. Insbesondere, wenn Internet-Verkehr über ATM transportiert wird, wird der ABR-Dienst von vielen heutigen Anwendungen verwendet werden.

Bei der Einführung von ABR wird ein Hardware-Austausch nötig, vor allem bei den Endgeräten, die die RM-Zellen generieren, ändern und auswerten müssen. Eine Erleichterung bei der Umstellung auf ABR stellt die Tatsache dar, daß am Anfang nicht gleich alle *Switches* ausgewählt werden müssen. *Switches*, bei denen der EFCI-Mechanismus bereits implementiert wurde, können zunächst weiterbenutzt werden. Man muß dabei aber bemerken, daß sich das auf die Qualität der ABR-Steuerung negativ auswirkt.

Kapitel 6

Aktuelle Ansätze zur Erweiterung des Classical-IP

Wie schon am Ende des Kapitels 3 angedeutet, bringt das *Classical*-IP-Modell der IETF in der heutigen Form viele Einschränkungen mit sich und hat mit einigen Problemen zu kämpfen. Um diese Probleme in den Griff zu bekommen und die Einschränkungen zu überwinden, wird innerhalb der IETF an Erweiterungen und Verfeinerungen des *Classical*-IP-Modells gearbeitet. Die erarbeiteten Ergebnisse werden jeweils in den Arbeitsdokumenten der IETF, sog. *Internet-Drafts* veröffentlicht. Die wichtigsten davon sind im folgenden aufgelistet:

- RFC1577-Update: *'Classical IP and ARP over ATM.'* [25]
  Diese neue Spezifikation verfeinert das *Classical*-IP-Modell und soll die RFCs 1577 und 1626 ablösen. Die wichtigste Erweiterung stellt die Verwendung von mehreren, synchronisierten ATMARP-Servern dar, deren ATM-Adressen bei den Clients mit Hilfe der in [26] definierten *Management Information Base* (MIB) konfigurierbar sein sollen.
- *'NBMA Next Hop Resolution Protocol (NHRP).'* [29]
  Um die Verwendung von Routern bei der Kommunikation zwischen logischen IP-Subnetzen (LIS) zu vermeiden und die Anzahl der *Hops* zu minimieren, wird ein erweiterter Adreßauflösungs-Mechanismus verwendet. Er erlaubt Auflösung von Adressen unabhängig von ihrer Subnetzzugehörigkeit und ermöglicht damit den Aufbau von direkten ATM-Verbindungen über IP-Subnetzgrenzen hinaus.
- *'Support for Multicast over UNI 3.0/3.1 based ATM Networks.'* [28]
  Für die Unterstützung von Multicasting in ATM-Netzen wird ein *Multicast Address Resolution Server* (MARS) vorgeschlagen.
- *'IP Broadcast over ATM Networks.'* [27]
  Beschreibt, wie die für ATM-Netze vorgesehenen IP-*Multicast*-Dienste für die Unterstützung von IP-*Broadcasting* verwendet werden können.
- *'Resource ReSerVation Protocol (RSVP).'* [30, 31]
  Bandbreitenreservierung für IP-Anwendungen, insbesondere in ATM-Netzen.
- *'Definitions of Managed Objects for Classical IP and ARP Over ATM Using SMIv2.'*
  Für die Unterstützung von *Classical*-IP über ATM, so wie das in [25] vorgesehen, wird eine *Managemen Information Base* (MIB) definiert.

6.1 Next Hop Resolution Protocol (NHRP)

Eine der größten Einschränkungen, unter der heutzutage das *Classical*-IP-Modell leidet, ist die Beibehaltung der klassischen IP-Subnetz-Architektur. Werden auf einem physikalischen ATM-Netz mehrere LISs (*Logical IP Subnetworks*) definiert, können Endgeräte, die in verschiedenen LISs liegen, nur über Router kommunizieren, obwohl man im Prinzip zwischen ihnen eine direkte virtuelle ATM-Verbindung aufbauen könnte. Der Ansatz erlaubt somit keine '*cut-through*'-Wege, die die zwischenliegenden Router umgehen.

Einen Ansatz zur Überwindung dieser Einschränkung stellt das *Next Hop Resolution Protocol* (NHRP) dar, an dem innerhalb der '*Routing over Large Clouds*'-Arbeitsgruppe (ROLC) der IETF gearbeitet wird.

NHRP wurde für *Non-Broadcast, Multi-Access* (NBMA) Netze definiert. Dies sind Netze, so wie ATM, *Frame Relay* oder X.25, bei denen viele Geräte an demselben Netz angeschlossen sind, in denen aber kein einfacher *Broadcast*-Mechanismus unterstützt wird, wie das in gewöhnlichen LANs der Fall ist. Für die an ein NBMA-Netz angeschlossenen Stationen gibt es aber keine weitere physikalische oder administrative Einschränkung, die eine direkte Kommunikation zwischen einzelnen Stationen verhindern könnte.

Für das *Classical-IP*-Modell stellt NHRP eine Erweiterung bei der Adreßauflösung dar. Es erlaubt einer Station (*Host* oder Router), die an einem ATM-Netz angeschlossen ist, die IP- und ATM-Adresse des optimalen *Next-Hop* zu erfahren, der auf dem Weg zur Zielstation liegt. Liegt die Zielstation im gleichen ATM-Netz, wird sie selbst als *Next-Hop* angegeben. Andernfalls wird als *Next-Hop* ein Rand-Router im ATM-Netz zurückgegeben, durch den die Zielstation am besten erreichbar ist.

Anstelle des ATMARP-Servers wird für NHRP ein *Next Hop Server* (NHS) verwendet.

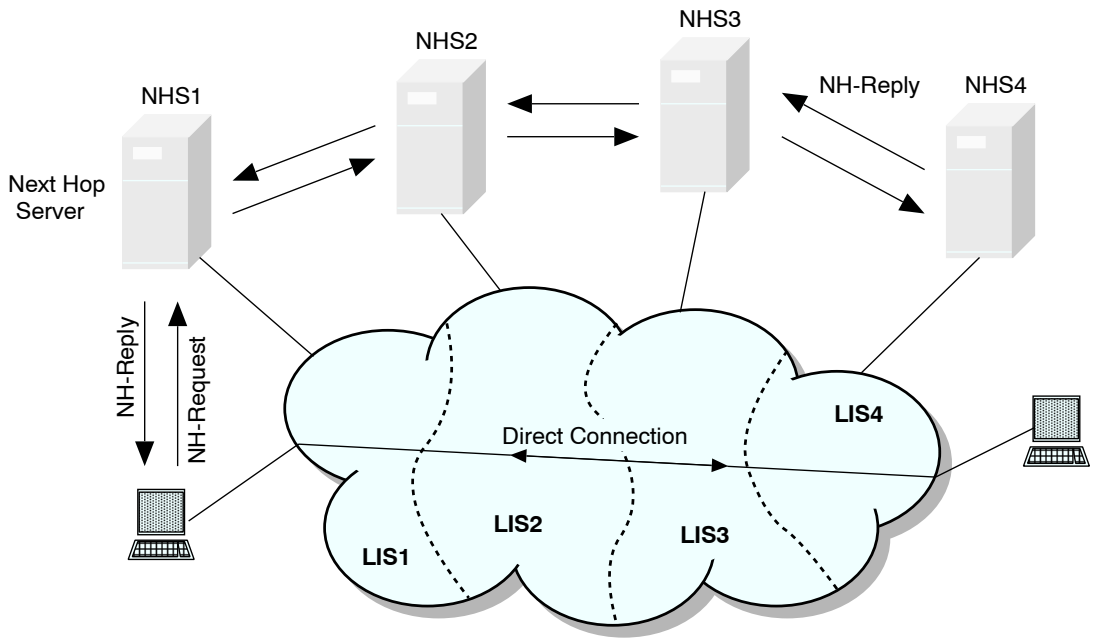


Bild 6.1: ***Next Hop Resolution Protocol***

Ein NHS ist jeweils für eine Gruppe von Clients zuständig und beantwortet direkt die NHRP-*Requests*, die sich auf diese beziehen. Die Clients in einer Gruppe werden mit der ATM-Adresse ihres zuständigen NHS (meistens der *Default*-Router) konfiguriert. Sie registrieren sich am Anfang bei ihm, indem sie ihm jeweils die eigene IP- und ATM-Adresse übermitteln. Ein NHS verwaltet diese Adreßabbildungen in seiner lokalen *Cache*-Tabelle. In dieser Tabelle speichert er auch Informationen, die er aus den NHRP-Paketen gewinnen kann, die ihn auf ihrem Weg passieren.

Ein NHS ist immer mit einer Routingeinheit gekoppelt. Es werden klassische Routing-Verfahren benutzt, und der Weg zum benachbarten NHS wird für ein NHRP-Paket basierend auf seiner IP-Zieladresse bestimmt.

Ein für eine Station zuständiger NHS muß also entlang der *Default-Route* liegen, die zu dieser Zielstation führt. In der Praxis bedeutet dies z.B., daß ein Router, der am Rand des ATM-Netzes liegt, als zuständiger NHS für all die Ziele fungiert, die außerhalb des ATM-Netzes liegen und durch ihn erreichbar sind.

Wenn eine Station ein Paket über das ATM-Netz schicken will und dazu eine IP-Adresse nach einer ATM-Adresse auflösen muß, generiert sie einen NHRP-*Request* und schickt ihn an den zuständigen NHS. Ein NHRP-*Request* wird, so wie alle anderen NHRP-Pakete, in einem IP-Paket geschickt.

Empfängt ein NHS einen NHRP-*Request*, dann überprüft er zunächst, ob die Zielstation, für die die *Next-Hop*-Adresse gesucht wird, in seinem Zuständigkeitsbereich liegt. Wurde die im NHRP-*Request* identifizierte IP-Zieladresse beim ihm registriert und gibt es dafür einen Eintrag in der lokalen Tabelle, dann antwortet der NHS mit einem NHRP-*Reply*, in dem die gesuchte ATM-Adresse des *Next-Hop* zusammen mit seiner IP-Adresse übermittelt wird. Der *Next-Hop* ist entweder die Zielstation selbst, sofern sie am ATM-Netz angeschlossen ist, oder der geeignete ATM-*Edge*-Router. Empfängt eine Station ein positives NHRP-*Reply*, dann baut sie eine ATM-Verbindung zum *Next-Hop* auf, und schickt die Daten an ihn weiter.

Kennt der NHS die gesucht ATM-Adresse für die Zielstation nicht, dann leitet er den NHRP-*Request* an einen anderen NHS weiter, und zwar an den, der als nächstes auf dem Weg zur Zielstation liegt. Der Weg zum Ziel wird dabei mittels der klassischen Routing-Prozedur bestimmt. Bei den nächsten NHSs wiederholt sich die Prozedur solange, bis ein NHS erreicht wird, der auf den NHRP-*Request* antworten kann (siehe Bild 6.1).

Kann keiner der NHSs im ATM-Netz den NHRP-*Request* auflösen, wird ein negatives NHRP-*Reply* (NAK) zurückgeschickt. Das kann z.B. passieren, wenn keine Routing-Informationen über den *Next-Hop* für die Zielstation vorhanden sind und der NHRP-*Request* nicht weitergeleitet werden kann.

Ein NHRP-*Reply* wird normalerweise über den gleichen Weg zurückgeschickt, der vom NHRP-*Request* benutzt wurde. Auf dem Weg zum *Requester* passiert ein NHR-*Reply* in umgekehrter Reihenfolge die gleiche Sequenz von NHSs. Die auf diesem Weg liegenden NHSs können aus dem NHRP-*Reply* lernen und die darin enthaltene *Next-Hop*-Information in ihren lokalen *Cache*-Tabellen abspeichern.

In einigen Fällen kann ein NHRP-*Reply* von einem NHS auch direkt zum *Request*-Initiator geschickt werden, falls eine direkte Kommunikation zwischen ihnen erlaubt und realisierbar ist. Dies reduziert die Antwortzeit, hindert jedoch die anderen NHSs, die ein NHRP-*Reply* im Normalfall passieren würde, daran, die in diesem *Reply* enthaltene Informationen in ihren lokalen *Cache*-Tabellen abzuspeichern.

Die in der *Cache*-Tabelle gespeicherte Information kann ein NHS später benutzen, falls er einen NHRP-*Request* empfängt, der sich auf die gleiche IP-Zieladresse bezieht. Er kann diesen

dann nämlich direkt beantworten und muß ihn nicht an den nächsten NHS weiterleiten. Das kann allerdings verhindert werden, falls sich der NHRP-*Requester* einen authorisierten NHRP-*Reply* wünscht. In diesem Fall dürfen die Informationen aus den *Cache*-Tabellen nicht benutzt werden. Ein NHRP-*Reply* wird erst von dem NHS geschickt, der direkt für die betreffende Zielstation zuständig ist (d.h. bei dem sich die Zielstation früher registriert hat).

Es hängt von der Implementierung ab, was die auf einen NHRP-*Reply* wartende Station mit den zu sendenden Paketen macht. Einerseits kann sie, solange eine direkte ATM-Verbindung zum *Next-Hop* nicht existiert, die Pakete entweder wegwerfen oder puffern. Andererseits kann die Station die Pakete direkt entlang des normalen Weges über Router zum Ziel schicken. Das reduziert die Verzögerungen und erlaubt den Paketen das Ziel zu erreichen, auch bevor der NHRP-*Request* aufgelöst und eine direkte ATM-Verbindung aufgebaut wird.

Weiterhin bietet das NHRP einige zusätzliche Eigenschaften. Eine wichtige Fähigkeit ist z.B. die Unterstützung der Adreß-Aggregation. Ein NHS liefert dann nicht nur die ATM-Adresse des *Next-Hop*, durch den eine gegebene IP-Adresse erreichbar ist, sondern zusätzlich auch die IP-Subnetzmaske, die mit dieser IP-Adresse verbunden ist. Diese Information wird in den *Cache*-Tabellen gespeichert und kann später bei Auflösung von NHRP-*Requests* benutzt werden, die sich auf IP-Adressen aus dem gleichen Subnetz beziehen. Das hat vor allem dann Vorteile, wenn einige IP-Adressen mit gleichem Adreß-Präfix durch denselben ATM-Rand-Router erreichbar sind. Jeder folgende NHRP-*Request* für eine dieser Adressen kann nämlich direkt aufgelöst werden und die ATM-Adresse des betreffenden Routers liefern.

Eine andere interessante Eigenschaft des NHRP ist die Möglichkeit des *Route-Recording* in den NHRP-Paketen. Bei dieser Option enthält die *Route-Recording*-Erweiterung des NHRP-Paketes die Adressen aller zwischenliegenden NHSs, die auf dem Weg zum Ziel und auf dem Rückweg zum *Requester* passiert wurden.

Diese Funktionalität kann z.B. verwendet werden, um Schleifen im Netzwerk zu entdecken. Außerdem kann sie hilfreich sein, falls eine direkte ATM-Verbindung (SVC) zum Empfänger aus irgendwelchen Gründen nicht aufgebaut werden kann oder darf. Kann eine Station nicht direkt mit einer IP-Zielstation kommunizieren, dann kann sie systematisch versuchen, eine ATM-Verbindung zu einem der NHS aus der *Route-Recording*-Liste aufzubauen, der dann als Zwischenpunkt zur Weiterbeförderung der Pakete zu dieser IP-Adresse dienen kann.

Außer den NHRP-*Request*- und NHRP-*Reply*-Nachrichten, die zur Adreßauflösung verwendet werden, definiert die NHRP-Spezifikation noch weitere Arten von NHRP-Nachrichten. Die NHRP-*Registration-Request*- und NHRP-*Registration-Reply*-Nachrichten werden bei der Registrierung der Clients beim NHRP-Server benutzt. Die NHRP-*Purge-Request*- und NHRP-*Purge-Reply*-Nachrichten werden dagegen verschickt, falls aufgrund irgendwelcher Änderungen im Netz die *Cache*-Information bei einer Station nicht mehr aktuell sein könnte und gelöscht werden soll. Schließlich wird die NHRP-*Error-Indication*-Nachricht zum Sender eines NHRP-*Request* geschickt, um ihn über eine Fehlersituation zu informieren.

Im Gegensatz zu heute verwendeten Ansätzen ermöglicht das NHRP also als erstes den Aufbau von direkten ATM-Verbindungen zwischen beliebigen Stationen, unabhängig von ihrer Zugehörigkeit zu Subnetzen. Der NHRP-Ansatz wird in Zukunft eine wichtige Rolle in ATM-Netzen spielen, besonders im Zusamenhang mit dem MPOA-Protokoll des ATM-Forums. Wie später im Kap. 7.1 beschrieben ist, soll bei MPOA ein ähnlicher und zum Teil erweiterter Mechanismus verwendet werden, um den Transport von Daten mit einem einzigen *Hop* über das gesamte ATM-Netzwerk zu ermöglichen.

6.2 IP-Multicasting über ATM

Seit einiger Zeit gewinnt das IP-*Multicasting* zunehmend an Bedeutung. Immer mehr neue Anwendungen und Protokolle benutzen es, um schnell Daten oder Informationen über das Netz zu versenden. Beim *Multicasting* werden die Daten von einer Station an eine Reihe ganz bestimmter Empfänger gleichzeitig verschickt. Diese Empfänger gehören einer definierten *Multicast*-Gruppe von Geräten an, die durch eine abstrakte IP-*Multicast*-Adresse identifiziert wird. Der IP-*Multicast*-Mechanismus ist in der Standard-IP-Spezifikation nicht enthalten und wurde erst in RFC 1112 [18] definiert.

Heute wird das *Multicasting* im *Classical*-IP-Protokoll noch nicht unterstützt, was schon längst als kritische Schwäche von RFC 1577 erkannt wurde, insbesondere im Vergleich zur LAN-Emulation.

Im aktuellen Internet-Draft *'Support for Multicast over UNI 3.0/3.1 based ATM Networks'* [28] werden Vorschläge der *'IP over ATM'*-Arbeitsgruppe der IETF beschrieben, wie *Multicasting* in Zukunft in *Classical*-IP-Netzen über ATM realisiert sein kann.

Für die Realisierung von *Multicastong* in ATM-basierten Netzen gibt es prinzipiell zwei Ansätze:

Multicast Server
: Beim Verschicken der *Multicast*-Pakete werden die Stationen von einem *Multicast*-Server unterstützt. Alle *Multicast*-Pakete werden von den Stationen zunächst an den zuständigen *Multicast*-Server geschickt, von dem sie dann weiter über eine Punkt-zu-Mehrpunkt-Verbindung an die Empfänger weitergeleitet werden. Für jede *Multicast*-Gruppe, die er bedient, unterhält der Server eine getrennte Punkt-zu-Mehrpunkt-Verbindung.

Verteilung durch direkte vermaschte Punkt-zu-Mehrpunkt-Verbindungen
: Jeder Sender baut zu allen Empfängern einer *Multicast*-Gruppe eine Punkt-zu-Mehrpunkt-Verbindung auf, über die er dann seine *Multicast*-Pakete verschickt. Auch hier gibt es bei jedem Sender für jede *Multicast*-Gruppe, an die er Daten sendet, eine getrennte Punkt-zu-Mehrpunkt-Verbindung.

Der Entwurf der IETF verfolgt beide Ansätze und benutzt zur Unterstützung von IP-*Multicasting* einen sog. *Multicast Address Resolution Server* (MARS). MARS kann analog zum ATMARP-Server gesehen werden. Er ist jeweils für die Auflösung von IP-*Multicast*-Adressen für eine Reihe von Knoten (sog. *Cluster*) zuständig. Alle Endsysteme innerhalb eines *Cluster* sind mit der ATM-Adresse des zuständigen MARS konfiguriert.

Auflösen von IP-*Multicast*-Adressen

Will eine Endstation Daten an eine bestimmte *Multicast*-Gruppe schicken, dann sendet sie zunächst an MARS einen MARS_*Request*, um die *Multicast*-Adresse dieser Gruppe aufzulösen.

Wenn keine Station sich für diese *Multicast*-Gruppe registriert hat, d.h. keine die Pakete für diese Adreßgruppe empfangen will, antwortet MARS mit einem MARS_NAK. Enthält dagegen die *Multicast*-Gruppe einige registrierte Stationen, hängt die Arbeitsweise des MARS davon ab, ob die betreffende *Multicast*-Gruppe für den Einsatz von *Multicast*-Servern oder für den Einsatz von direkten Punkt-zu-Mehrpunkt-Verbindungen konfiguriert ist.

Werden direkte Punkt-zu-Mehrpunkt-Verbindungen benutzt, schickt MARS als Antwort eine MARS_MULTI-Nachricht mit einer Liste der Stationen, die sich bei dieser *Multicast*-Gruppe angemeldet haben und Daten für diese IP-*Multicast*-Adresse empfangen wollen. Die

sendewillige Station baut daraufhin eine Punkt-zu-Mehrpunkt-Verbindung zu all diesen Stationen auf und kann dann ihre *Multicast*-Pakete über diese Verbindung versenden.

Soll dagegen ein *Multicast*-Server zum Einsatz kommen, antwortet MARS mit einer MARS_MULTI-Nachricht, die eine Liste mit ATM-Adressen von einem oder mehreren *Multicast*-Servern enthält, die für die Verteilung von Paketen für diese *Multicast*-Adresse zuständig sind. In diesem Fall baut der Sender ebenfalls eine Punkt-zu-Mehrpunkt-Verbindung zu den Servern auf, die er dann zum Verschicken seiner *Multicast*-Pakete benutzt.

Teilnahme an den *Multicast*-Gruppen

Ein anderer, komplexerer Teil des MARS-Protokolls beschäftigt sich mit der Frage, wie die o.g. Listen, die Liste der *Multicast*-Server und die Liste der Stationen, für jede *Multicast*-Adresse automatisch beim MARS erstellt werden kann. Das Problem liegt vor allem darin, wie der MARS erfährt, welche Stationen Pakete für die jeweilige IP-*Multicast*-Adresse empfangen wollen, d.h. wie sie sich bei den *Multicast*-Gruppen dynamisch an- und abmelden können.

In RFC 1112 wurde dafür ein *Internet Gateway Message Protocol* (IGMP) definiert. Außerdem wird eine IP-*Multicast*-Adresse 224.0.0.1 für eine *Multicast*-Gruppe reserviert, die alle *multicast*-fähigen *Hosts* in einem Subnetz enthält. Diese Gruppe wird z.B. von den Routern verwendet, um den Status der *Multicast*-Gruppen in einem Subnetz zu überwachen. Die Router schicken IGMP-*Queries* an diese Gruppe, die von den betreffenden *Multicast*-Stationen mit IGMP-*Report*-Nachrichten beantwortet werden.

Um diesen Anforderungen des RFC 1112 gerecht zu werden und die reservierte *Multicast*-Adresse 224.0.0.1 (alle *Hosts* im Subnetz) zu unterstützen, wird MARS als *Multicast*-Server für zwei *Multicast*-Gruppen eingesetzt. Für die eine Gruppe, zu der alle *Multicast*-Server gehören, benutzt MARS die *ServerControlVC*. Die zweite Gruppe, die alle Endsysteme (inklusive Router) im *Cluster* enthält, erreicht MARS über die *ClusterControlVC*.

Jede Station, die *Multicast*-Pakete an eine der *Multicast*-Gruppen senden will, muß sich bei MARS mit einem entsprechenden MARS_JOIN registrieren lassen. Das hat vor allem zur Folge, daß MARS den neuen Knoten an seine *ClusterControlVC* anschließt, um ihn später über Änderungen in den *Multicast*-Gruppen zu informieren.

Auch jeder *Multicast*-Server, der die *Multicast*-Pakete für eine bestimmte *Multicast*-Adresse verteilen will, muß sich zuerst bei MARS mit einem MARS_MSERV registrieren lassen. MARS benutzt die Registrierung, um für jede *Multicast*-Adresse eine Liste der zuständigen *Multicast*-Server zu erstellen. Die Liste enthält jeweils die ATM-Adressen eines oder mehrerer *Multicast*-Server, die das Verschicken von *Multicast*-Paketen für diese *Multicast*-Gruppe übernehmen. Die Liste wird von MARS als Antwort auf einen MARS_Request geschickt, der sich auf die betreffende Gruppe bezieht. Darüber hinaus schließt MARS jeden neu registrierten *Multicast*-Server an seine *ServerControlVC* an.

Ein *Multicast*-Server, der eine *Multicast*-Adresse bedient, muß wissen, welche Stationen sich für diese Gruppe registriert haben und Pakete für diese *Multicast*-Adresse empfangen wollen. Das erfährt er, indem er an MARS einen normalen MARS_*Request* schickt. MARS erkennt, daß der *Request* von einem *Multicast*-Server kommt, und antwortet mit der korrespondierenden Liste der Stationen, so daß der Server daraufhin seine Punkt-zu-Mehrpunkt-Verbindung aufbauen kann.

Will eine Station Daten für eine bestimmte *Multicast*-Adresse empfangen, muß sie sich zuerst bei MARS mit einem MARS_JOIN für diese *Multicast*-Gruppe registrieren lassen. Das neue Gruppenmitglied wird von MARS an die entsprechende Liste angehängt, die alle zu dieser

Gruppe gehörenden Stationen enthält. Kommt später bei MARS ein MARS_*Request* an, der sich auf diese Gruppe bezieht, wird er von MARS mit einem MARS_MULTI beantwortet, in dem diese Liste übermittelt wird.

Falls später eine Station eine *Multicast*-Gruppe verlassen will, schickt sie an MARS einen MARS_LEAVE. Das veranlaßt MARS, die Station aus der entsprechenden Gruppenliste zu entfernen.

Damit die bereits für die Unterstützung von *Multicast* existierenden aktiven Punkt-zu-Mehrpunkt-Verbindungen, sei es beim *Multicast*-Server oder bei den Endstationen, automatisch aktualisiert werden können, muß MARS jeden MARS_JOIN und MARS_LEAVE an die entsprechenden Stellen weiterleiten. Kommt eine direkte Punkt-zu-Mehrpunkt-Verbindung zum Einsatz, werden diese Nachrichten über die *ClusterControlVC* an alle Stationen im *Cluster* geschickt, um sie über Änderungen in der betreffenden *Multicast*-Gruppe zu informieren. Beim MARS_Join wird das neue Gruppenmitglied bei den angesprochenen Stationen jeweils an ihre Punkt-zu-Mehrpunkt-Verbindung angeschlossen. Bei MARS_LEAVE wird die betreffende Station aus diesen Verbindungen gelöscht.

Im Falle von *Multicast*-Servern, werden MARS_JOIN und MARS_LEAVE dagegen über die *ServerControlVC* an alle Server weitergeleitet. Das erlaubt den relevanten Servern, ihre Punkt-zu-Mehrpunkt-Verbindungen zu aktualisieren, indem sie die betreffende Station an diese Verbindungen anschließen oder die Verbindungen zu ihr abbauen.

Weitere Details, die sich mit der Realisierung von IP-*Multicasting* in *Classical*-IP-Netzen beschäftigen, sind in dem betreffenden *Internet-Draft* [28] beschrieben oder werden noch diskutiert.

Kapitel 7

Multiprotokoll-Routing über ATM

Die Integration von ATM-Netzen in herkömmliche LAN-Umgebungen ist heutzutage nach wie vor ein schwieriges Problem. Die derzeit verwendeten Verfahren, wie das *Classical*-IP-Modell der IETF und die LAN-Emulation des ATM-Forums, schaffen zwar dabei eine Abhilfe, sie verfolgen jedoch immer noch den klassischen Ansatz, bei dem zur Kommunikation zwischen den Subnetzen ein traditioneller Router verwendet wird. Die Frage, wie man LAN-Verkehr über einen ATM-Netz routet, bleibt offen.

Für das Routing von Netzwerkschichtprotokollen über ATM gibt es zwei unterschiedliche Ansätze:

Layered-Routing-Modell (auch *Overlay-Routing*-Modell genannt)
: Auf der ATM- und der Netzwerkschichtebene wird ein voneinander unabhängiges Routing durchgeführt. Das Modell erlaubt dadurch die Verwendung der vorhandenen Ansätze und ist einfach und allgemein realisierbar.

Integrated-Routing-Modell
: Das Routing auf den beiden Ebenen wird integriert. Es wird ein einziges Routing-Protokoll verwendet, sowohl für die zellbasierte ATM-Ebene als auch für die paketorientierte Netzwerkschicht.

Innerhalb des ATM-Forums werden derzeit zwei verschiedene Verfahren diskutiert, wobei je eines auf einem der o.g. Routing-Modelle basiert. Innerhalb der PNNI-Arbeitsgruppe (*Private Network Node Interface*), die sich mit dem Protokoll für die Kommunikation zwischen den ATM-*Switches* beschäftigt, wird an einem sog. I-PNNI (*Integrated PNNI*) gearbeitet, das auf dem *Integrated-Routing*-Modell basiert. Der zweite Vorschlag, der von der '*Multiprotocol over ATM*'-Arbeitsgruppe (MPOA) erarbeitet wurde, basiert dagegen auf dem ersten Modell, dem *Layered-Routing*-Modell.

Integrated-PNNI stellt eine Erweiterung des PNNI auf paketorientierte Netze dar. PNNI selbst ist das einzige standardisierte Routing-Protokoll auf der ATM-Ebene, das für den Einsatz in privaten ATM-Netzen definiert ist. Als LAN-spezifische Weiterentwicklung von PNNI kann I-PNNI somit generell als einziges Routing-Protokoll für paket- und zellbasierte Systeme verwendet werden. Dadurch können sowohl die nicht ATM-Netzsegmente als auch die Netzwerkschicht von den Vorteilen des PNNI, wie hierarchischem oder QoS-Routing, profitieren. Auf den fertigen I-PNNI-Standard wird man noch aber eine Weile warten müssen, denn den Prognosen zur Folge kann man ihn frühenstens 1998 erwarten.

7.1 Multiprotocol over ATM (MPOA)

Um zu verstehen, wie MPOA arbeitet, wird zuerst auf die Probleme eingegangen, die die derzeit verwendeten Ansätze – *Classical IP over ATM* und LAN *Emulation* – mit sich bringen und die mit MPOA gelöst weden sollen.

Die LAN-Emulation des ATM-Forums basiert auf einem *Bridging*-Ansatz und emuliert ein physikalisches Segment eines herkömmlichen LAN auf der MAC-Ebene. Dadurch werden in einem emulierten LAN (ELAN) die typischen *Layer*-2-Einschränkungen beibehalten, und man hat mit den gleichen Problemen zu kämpfen, wie bei klassischen – mit Brücken aufgebauten – Netzwerken. Mit den Funktionen der Ebene 2 läßt sich z.B. keine *Broadcast*-Lasttrennung realisieren, sowie kein Wechsel zwischen unterschiedlichen LAN-Techniken. Die Verbindungen zwischen virtuellen ELANs erfordern traditionelle Router, die unnötige Verzögerungen produzieren und bei denen es schnell zu Engpässen kommen kann. Das bedeutet auch gleichzeitig, daß keine direkten ATM-Verbindungen zwischen zwei Stationen möglich sind, die zu unterschiedlichen emulierten LANs gehören. Die aktuelle Version der LANE-Spezifikation erlaubt auch nicht die Verwendung der gleichen SVC für den Transport von Daten aus unterschiedlichen virtuellen ELANs. All das führt dazu, daß der Ansatz schlecht skaliert und nicht für den Aufbau von größeren Netzen geeignet ist.

LANE emuliert ein LAN auf der MAC-Ebene und schirmt damit ATM von den höheren Schichten, insbesondere vor der Netzwerkschicht, effektiv ab. Sie hindert dadurch gleichzeitig die Anwendungen daran, von den wichtigen ATM-Dienstmerkmalen wie QoS Gebrauch zu machen. Außerdem führt das zu einem zusätzlichen Protokoll-*Overhead*, insbesondere bei der Adreßauflösung (IP → MAC → ATM). Schließlich müssen in einem emulierten LAN alle Geräte die gleiche MTU-Größe benutzen. Bei einem ELAN mit ATM- und Ethernet-*Hosts* wird diese dadurch auf 1500 Bytes begrenzt, was einen großen negativen Einfluß auf die Performance hat.

Das von der IETF entwickelte *Classical*-IP-Modell zur Unterstützung von IP über ATM, kennt einige der o.g. Probleme nicht. Die IP-Netzwerkschicht wird unmittelbar auf ATM abgebildet und die IP-Subnetze müssen nicht die LANE-Software benutzen. Sie können direkt über ATM betrieben werden.

Zur Adreßauflösung wird beim *Classical*-IP-Modell der ATMARP-Server verwendet, der ähnlich wie der LE-Server arbeitet. Anstatt aber für die Auflösung von MAC-Adressen ist er für direktes Auflösen von IP- nach ATM-Adressen zuständig, was den Protokoll-*Overhead*, insbesondere beim der Verbindungsaufbau, reduziert.

Einige wichtige Einschränkungen sind jedoch auch hier vorhanden. Wie bei der LAN-Emulation braucht man traditionelle Router, um die *Classical*-IP-Subnetze zu verbinden. Die IP-Stationen aus unterschiedlichen LISs können nicht direkt miteinander kommunizieren. Das Problem soll mit dem im Kap. 6.1 beschriebenen NHRP-Protokoll gelöst werden, an dem innerhalb der IETF noch gearbeitet wird. Andere Einschränkungen, die sich auf *Multicasting* beziehen, sollen ebenfalls mit einem Ansatz der IETF überwunden werden, der die Verwendung von MARS (*Multicast Address Resolution Server*) vorschlägt.

In der *Classical*-IP-Umgebung lassen sich darüber hinaus die klassischen LAN-Segmente nur mit teuren ATM-Routern an das ATM-Netz anschließen. Außerdem ist das von der IETF entwickelte Modell, obwohl konzeptionell breiter angelegt, nur für IP definiert.

Die Frage, wie man allgemein ein beliebiges Netzwerkschichtprotokoll über ein ATM-Netz routet, bleibt immer noch offen und soll mit MPOA beantwortet werden.

Das Ziel von MPOA ist, den Transport unterschiedlicher Netzwerkschicht-Protokolle über

ATM auf eine effiziente, elegante und gut skalierbare Weise zu realisieren. Außer den direkt an ATM angeschlossenen Endsystemen, werden auch Endsysteme an traditionellen Netzen berücksichtigt, die dabei sowohl direkt mit ATM-Endgeräten als auch mit Endsystemen eines anderen klassischen LANs über ATM kommunizieren können.

Basierend auf den Subnetzstrukturen der Netzwerkschicht sollen virtuelle LANs (VLANs) definiert werden, die sich sogar über die Grenzen des ATM-Netzes hinweg erstrecken können. Die Zugehörigkeit der Systeme zu einem VLAN erfolgt unabhängig von ihrer physikalichen Lage. D.h. ein VLAN kann auch Endsysteme aus unterschiedlichen Ethernet-Segmenten an verschiedenen Standorten enthalten.

MPOA basiert auf verschiedenen Ansätzen und bereits verwendeten Lösungen. Dabei handelt sich sowohl um die schon o.g. Standards wie LAN-Emulation des ATM-Forums und *Classical*-IP der IETF sowie um die neueren Entwicklungen der IETF wie NHRP, MARS und RSVP.

MPOA soll als Kombination aller dieser Konzepte implementiert werden. Zum Teil erweitert es diese und definiert, wie sie optimal zusammen arbeiten können, um das gewünschte Ziel zu erreichen. Außerdem verfolgt MPOA einen weiteren Ansatz, der die Verwendung eines sogenannten Route-Servers zur Adreßabbildung in einer Multiprotokollumgebung vorsieht.

Die MPOA-Architektur enthält folgende Komponenten (siehe Bild 7.1):

- **Randgeräte** (*Edge Devices*)
  Ein Randgerät ist ein intelligenter *Switch*, der die klassischen LANs mit ATM-Netzen verbindet. Neben den ATM-Schnittstellen besitzt er einen odere mehrere Ports für traditionelle LAN-Segmente. Das Switching der Datenpakete erfolgt bei ihm nicht nur auf Basis von MAC-Adressen (wie bei den heutigen LAN-*Switches*) sondern auch auf Basis ihrer Netzwerkschichtadressen. Weil ein Randgerät sowohl einen Teil der *Layer*-2- als auch der *Layer*-3-Funktionalität besitzt, wird es auch als *Multilayer-Switch* bezeichnet. Obwohl er die *Layer*-3-Pakete weiterbefördern kann, wird er meistens nicht aktiv in die Ausführung der Routingprotokolle der Netzwerkschicht einbezogen.

- **ATM-*Hosts***
  Dies sind direkt an ATM angeschlossene Stationen, die einen ATM-Netzadapter besitzen, bei dem das MPOA-Protokoll als Teil der Treibersoftware implementiert wird. Er kann direkt über ATM mit anderen ATM-*Hosts* kommunizieren oder mit herkömmlichen LANs über Randgeräte, durch die diese erreichbar sind.

- **Route Server**
  Der Route-Server ist nicht ein physikalisches Gerät als solches, sondern viel mehr eine Sammlung von Funktionen (sog. *Route Server Functional Group*, RSFG), die es erlauben, die auf der Netzwerkschicht definierten virtuellen Subnetze auf ATM abbilden zu lassen. Der Route-Server erlernt und verwaltet die Informationen über die Netzwerktopologie. Er unterhält Adreßtabellen für die ATM- und MAC-Ebene, sowie die Netzwerkschicht und versorgt die MPOA-Clients (ATM-*Hosts*, Randgeräte) auf Anfrage mit diesen Informationen, die sie benötigen, um eine direkte virtuelle ATM-Verbindung zwischen beliebigen Endpunkten aufzubauen.

- **IASG** (*Internet Address Summarization Group*)
  Im Gegensatz zur LAN-Emulation, bei der die virtuellen Netze auf der Ebene 2 definiert sind, definiert MPOA virtuelle Subnetze (sog. IASG) basierend auf den Strukturen

der Netzwerkschicht. Ein IASG spezifiziert dabei einen Bereich der Netzwerkschicht-Adressen mit dazugehörigem Netzwerkschichtprotokoll, mit anderen Wörten identifiziert ein IASG z.B. IP und ein virtuelles IP-Subnetz.

Beim MPOA-Modell soll die *Intra*-IASG-Kommunikation die Rückwärtskompatibilität zur LAN-Emulation aufweisen. Das bedeutet, daß die heute verwendeten ATM-Adapter und LAN-*Switches*, die LANE unterstützen, mit den MPOA-Geräten werden kommunizieren können, falls sie in der gleichen *Internet Address Summarization Group* liegen (d.h. z.B. wenn alle Stationen zum gleichen IP-Subnetz gehören). Für die *Inter*-IASG-Kommunikation brauchen die LANE-Geräte allerdings einen Router oder ein *Gateway*, die sowohl MPOA als auch LANE unterstützen.

Das MPOA-Modell kann auch als 'virtueller Router' betrachtet werden, da die MPOA-Geräte an einem ATM-Netz gemeinsam die Funktionalität eines traditionellen Multiprotokoll-Bridge/Routers besitzen. Die herkömmlichen LAN-Segmente sind über Randgeräte angeschlossen und entsprechen somit den Schnittstellenkarten des Routers. Das ATM-Netzwerk kann als *Backplane* des Routers angesehen werden, die die Randgeräte miteinander verbindet. Der Route-Server selbst erfüllt schließlich die Funktionen des Steuerprozessors eines klassischen Routers.

Kommt ein Datenpaket aus einem Segment eines herkömmlichen LAN beim Randgerät an,

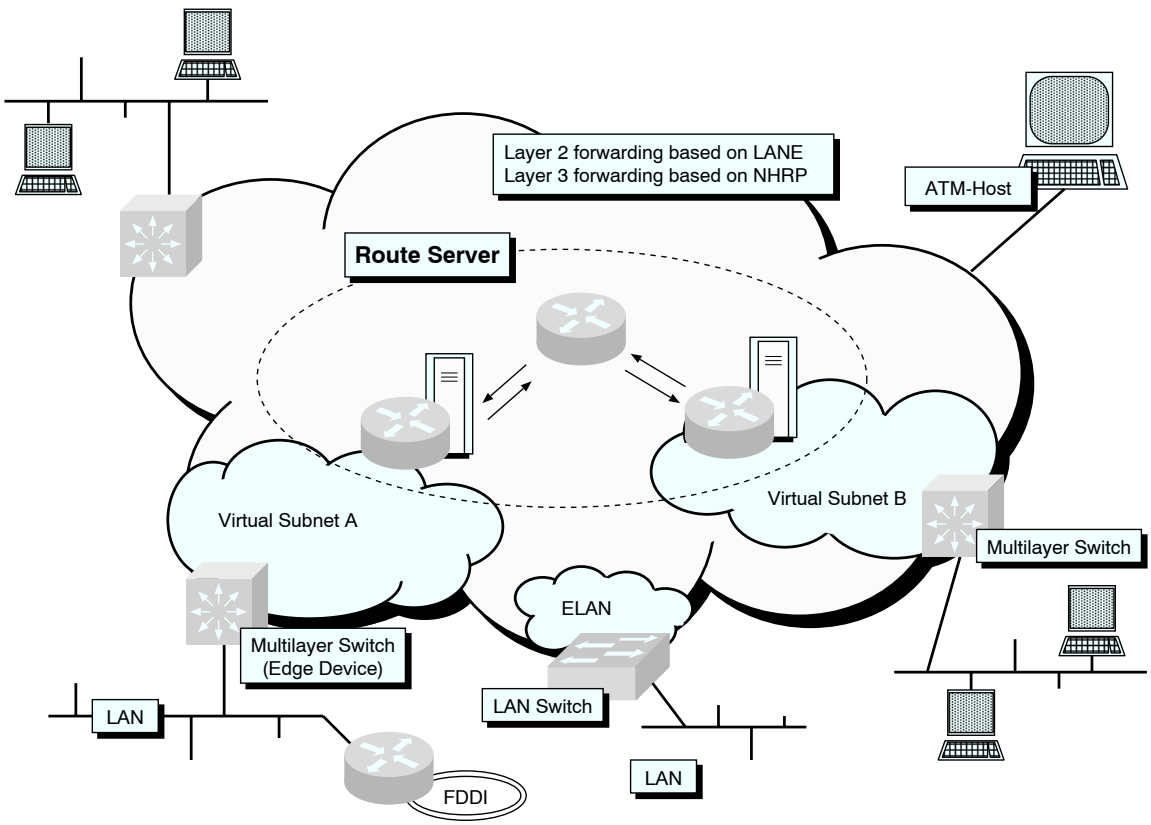


Bild 7.1: MPOA Architektur

muß dieses entscheiden, wie es weiterbefördert werden soll. Es überprüft die Zieladresse, um festzustellen, ob das Paket geroutet oder gebridged werden soll. Ist das Paket für eine Station innerhalb des gleichen virtuelles Subnetztes (IASG) bestimmt, wird es gebridged. Es wird ein LANE-ähnliches Verfahren benutzt, um die gesuchte ATM-Adresse zu erfahren und eine direkte virtuelle ATM-Verbindung aufzubauen. Muß das Paket jedoch geroutet werden, so wird beim Randgerät die Netzwerkschicht-Adresse, basierend auf Informationen vom Route-Server, direkt nach der ATM-Adresse aufgelöst und eine direkte virtuelle ATM-Verbindung zum entsprechenden Ziel aufgebaut.

Bei einem ATM-*Host* ist die Vorgehensweise einfacher. Dieser fragt den Route-Server sowohl bei der *Inter*- als auch bei der *Intra*-IASG-Kommunikation nach der passenden ATM-Adresse und baut dann eine direkte virtuelle ATM-Verbindung zum Ziel auf.

Die Adreßauflösung bei den Route-Servern basiert auf dem gleichem Prinzip wie beim NHRP-Protokoll. Kennt der lokale Route-Server die gesuchte ATM-Adresse nicht, dann leitet er die Adreßanfrage zum nächsten Route-Server weiter. Die Ziel-ATM-Adresse, die letztendlich vom Route-Server zurückgeliefert wird, ist dann entweder die Adresse des Ziel-*Host* selbst, falls dieser direkt an das ATM-Netz angeschlossen ist, oder die Adresse des entsprechenden Randgeräts, durch das das Paket weitergeleitet werden soll.

Um die Netztopologie zu erkennen, wird bei MPOA für jedes Netzwerkschicht-Protokoll ein eigenes unabhängiges, klassisches Routing-Protokoll verwendet, wie z.B. RIP (*Routing Information Protocol*), OSPF (*Open Shortest Path First*) für IP, NLSP für IPX und RTMP für AppleTalk. Auf der ATM-Ebene wird dagegen beim Aufbau der ATM-Verbindungen das PNNI-Routing-Protocol verwendet.

Die Einführung von MPOA wird die Migration zu skalierbaren ATM-Netzen erleichtern. MPOA bringt im Vergleich zu den derzeit verwendeten Ansätzen viele Vorteile. Zum einen können Daten zwischen zwei Stationen, unabhängig von ihrer Zugehörigkeit zu Subnetzen, über direkte ATM-Verbindungen mit einem einzigen *Hop* über das gesamte ATM-Netzwerk transportiert werden, was vor allem zu geringeren Verzögerungen führt. Zum anderen reduziert das direkte Abbilden der Netzwerkschichtprotokolle auf ATM den Protokoll-*Overhead* und ermöglicht den Anwendungen, von den ATM-spezifischen Merkmalen, wie z.B. QoS, Gebrauch zu machen. Schließlich erlaubt MPOA das Definieren und das effektive Verwalten von virtuellen LANs, die auf den Strukturen der *Layer*-3-Ebene basieren.

Kapitel 8

Zusammenfassung und Ausblick

Obwohl die ATM-Technik ihren Ursprung im Bereich der öffentlichen Netze hat, findet sie auch Akzeptanz und Verbreitung im privaten/lokalen Bereich, und eröffnet erstmals die Perspektive für eine einheitliche Netztechnologie in allen Bereichen und für alle Arten von Netzdiensten.

Während man anfgangs beim Aufbau eines lokalen ATM-Netzes auf proprietäre Lösungen angewiesen war, lassen sich inzwischen auf der Basis der Standards der *Internet Engineering Task Force* (IETF) und des ATM-Forums auch heterogene ATM-Netze aus Komponenten unterschiedlicher Firmen betreiben.

Eine Grundlage dafür stellt die vom ATM-Forum entwickelte UNI-3.0/3.1-Spezifikation dar, welche die Schnittstelle zwischen ATM-*Switch* und Endgerät beschreibt und die Interoperabilität von Komponenten verschiedener Hersteller beim Einsatz von Wählverbindungen gewährleistet. Ein Durchbruch in der Nutzung von ATM gelang aber erst mit dem *'Classical IP over ATM'* der IETF, einem einfachen Ansatz, der von fast allen Herstellern unterstützt wird. Danach folgte die Spezifikation der LAN-Emulation des ATM-Forums, die auch den Transport von anderen Protokollen als IP über ATM erlaubt und *Broad-/Multicasting* unterstützt.

Beide Verfahren erleichtern die Integration von ATM-Netzen in herkömmliche LAN-Umgebungen, bringen heute jedoch noch viele Einschränkungen und Probleme mit sich. Sie verfolgen beide immer noch den klassischen Ansatz, bei dem zur Kommunikation zwischen den Subnetzen ein traditioneller Router benötigt wird. Dabei entstehen unnötige Verzögerungen und eine wichtige ATM-Eigenschaft, die Zeittransparenz, geht verloren. Aus diesem Grund, wie auch wegen der zentralen Server und den damit verbundenen *Single-Point-Of-Failure*-Problemen, skalieren beide Ansätze schlecht und sind nicht für den Aufbau größerer Netze geeignet.

Jedes Modell hat im direkten Vergleich zum anderen seine Vor- und Nachteile. Obwohl die LAN-Emulation die größere Funktionalität aufweist und in der Konfiguration durch den Einsatz eines zentralen *LAN Emulation Configuration Server* (LECS) einfacher ist, stellt das *Classical*-IP-Modell zur Zeit eine deutlich stabilere und effizientere Lösung dar. Das Modell wird zukünftig noch interessanter, wenn die vorgesehenen Erweiterungen wie *Next Hop Resolution Protocol* (NHRP) und *Multicast Address Resolution Server* (MARS) standardisiert werden und zum Einsatz kommen. Während NHRP den Aufbau von direkten ATM-Verbindungen über IP-Subnetzgrenzen hinaus unterstützt und somit den Umweg über Router erspart, wird mit dem MARS den Anwendungen auch in den *Classical*-IP-Netzen *Multica-*

sting zur Verfügung gestellt. Sollen jedoch auch andere Protokolle als IP über ein ATM-Netz transportiert werden, bleibt keine Wahl, und man ist an die LAN-Emulation gebunden.

Heutzutage liegt der Schwerpunkt der Nutzung von ATM-Netzen im Transport von IP-basierten Protokollen, vor allem TCP. Wie sich aber während der Tests herausstellte, werden heutige Implementierungen von TCP oder UDP nicht ausreichend für den Einsatz in ATM-Netzen mit ihren sehr hohen Übertragungsgeschwindigkeiten vorbereitet. Die Protokolle weisen häufig Eigenschaften auf, die eine optimale Nutzung der hohen Bandbreiten erheblich erschweren, und werden mit neuen Problemen konfrontiert, für die ursprünglich, wie z.B. im Falle des TCP-Protokolls, keine Lösungen vorgesehen wurden.

Erweiterungen der klassischen Protokolle sind nötig, um sie über ein ATM-Netz effektiv transportieren zu können. Einige bereits in dieser Hinsicht entwickelten Ansätze, wie die *Window-Scale*-Option (Benutzung von größeren Fenstern als 64 KBytes) oder der *Path-MTU-Discovery*-Mechanismus (Ermitteln der maximaler Paketgröße, die von allen *Links* auf dem Pfad zum Empfänger ohne Fragmentierung unterstützt wird), sind noch nicht bei allen Herstellern implementiert. Auch die *Default*-Einstellungen der TCP/UDP-Parameter führen nicht zu optimalen Durchsatzraten. Um in den Genuß der hohen Übertragungsraten zu kommen, müssen bei den Workstations Anpassungen durchgeführt werden. Eine wichtige Rolle spielt dabei die Größe der Sende- und Empfangspuffer sowie die Verwendung von großen Paketlängen.

Nicht zuletzt hat auch die Art und die Leistungsfähigkeit der Endsysteme einen bedeutenden Einfluß auf die erreichbare *Performance*. Häufig stellt das Endgerät den Flaschenhals dar und nicht mehr das Netz. Die verwendeten 155-Mbps-Verbindungen zum Endgerät können von vielen Workstations nicht voll ausgelastet werden und sind daher zur Zeit mehr als ausreichend.

Im Bereich zwischen den *Switches* (wie z.B. bei *Backbone*-Leitungen) sind aber die größeren Übertragungsgeschwindigkeiten, wie 622 MBps, durchaus sinnvoll. Bislang werden diese aber nur von wenigen Herstellern unterstützt. Für noch größere Bandbreiten (Gigabitbereich) gibt es zur Zeit noch keine endgültigen Standards, und sie werden derzeit nur in Pilotnetzen in Testlabors eingesetzt.

Große Auswirkungen auf die *Performance* haben – wegen fehlender Mechanismen zur Verkehrs- und Überlaststeuerung auftretenden – Zellverluste. Die paketbasierten Protokolle wie IP reagieren darauf besonders empfindlich, denn bei Verlust nur einer einzigen Zelle muß ein ganzes Paket erneut übertragen werden. Dadurch wird Bandbreite verschwendet, und bei unglücklichen Konstellationen kann es sogar zu einem Zusammenbruch des Durchsatzes kommen. Teilweise Abhilfe schaffen hier *Packet-Discard*-Techniken; eine grundsätzliche Lösung des Problems kann jedoch erst mit der Ablösung der UBR-Dienstklasse (*Unspecified Bit Rate*) durch die vor kurzem standardisierte ABR-Dienstklasse (*Available Bit Rate*) erfolgen. ABR stellt Mechanismen zur Flußsteuerung bereit und erlaubt den Stationen, dynamisch ihre Senderate an die im Netz verfügbare Bandbreite anzupassen, so daß Zellverluste vermieden werden. Allerdings macht die Einführung von ABR einen Austausch der ATM-Interfaces und eine Aufrüstung der *Switches* erforderlich.

Nachdem das ATM-Forum mit der LANE-Spezifikation definiert hat, wie mit einem *Bridging*-Ansatz LAN-Verkehr über ein ATM-Netz transportiert werden kann, wird zur Zeit daran

gearbeitet, wie man verschiedene Netzwerkschichtprotokolle effizient, elegant und auf eine gut skalierbare Weise über ATM transportieren kann, d.h. wie man LAN-Verkehr über ein ATM-Netz routen kann. Bei dem dafür vorgesehenem *'Multi-Protocol Over ATM'* (MPOA) wird ein virtueller Route-Server eingesetzt, der Informationen über die Netztopologie erlernt und verwaltet. Er stellt den angeschlossenen Geräten Informationen über Adressen der ATM- und MAC-Ebene sowie der Netzwerkschicht bereit. Damit unterstützt er den Aufbau von direkten ATM-Verbindungen zwischen beliebigen Stationen am ATM-Netz. Die Segmente eines herkömmlichen LAN werden über *Multilayer-Switches* (Randgeräte) an das ATM-Netz angeschlossen. Außerdem soll MPOA Mechanismen zur Konfiguration und effektiven Verwaltung von komplexen und flexiblen virtuellen LANs bereitstellen. Diese VLANs sollen auf Basis von Layer-3-Informationen definiert werden und die logischen Netzwerkstrukturen von den physikalischen unabhängig machen.

Eine weiter verbesserte Lösung für das Multiprotokoll-über-ATM-Problem wird sich mit dem *Integrated*-PNNI ergeben. I-PNNI als LAN-spezifische Erweiterung des PNNI auf paketorientierte Netze, integriert das Routing auf der ATM- und auf der Netzwerkschicht, und kann als einheitliches Routing-Protokoll für paket- und zellbasierte Systeme verwendet werden. Die Verabschiedung des I-PNNI-Standards wird frühestens 1998 erwartet.

Solange noch keines der beiden Verfahren – MPOA oder I-PNNI – standardisiert ist und keine voll funktions- und interoperationsfähigen Implementierungen verfügbar sind, sollte man sich beim Aufbau größerer ATM-Netze auf einen bestimmten Hersteller beschränken. Daran werden auch die *Switches* der nächsten Generation, die die neuen Standards wie PNNI und ABR unterstützen werden, nicht viel ändern. Beim Konfigurieren, Überwachen und Verwalten eines größeren ATM-Netzes, das aus vielen LAN-ATM-*Switches* und virtuellen LANs aufgebaut ist, ist man zur Zeit auf proprietäre Lösungen angewiesen. Welche Ansätze und Modelle sich in diesem Bereich durchsetzen werden, bleibt abzuwarten.

Anhang A

Status der wichtigsten ATM-Standards und -Spezifikationen

A.1 ATM-Forum

Verabschiedete Spezifikationen

- **UNI 3.0** [33]
  Inhalt: Physikalische Schnittstellen, UNI-Signalisierung (Unterstützung von SVCs), ATM-Managementaspekte, ILMI.
  Veröffentlicht im September 1993.

- **UNI 3.1** [34]
  Inhalt: Anpassung der UNI 3.0 an den ITU-T-Standard Q.2931.
  Veröffentlicht im September 1994.

- **LANE Phase 1** [36]
  Inhalt: Emulation eines klassischen LAN auf einem ATM-Netz.
  Veröffentlicht im Januar 1995.

- **IISP** [35]
  Inhalt: Ein auf UNI 3.0/3.1 basierendes statisches NNI-Routingprotokoll.
  Veröffentlicht im Februar 1995.

- **P-NNI Phase 1** [39]
  Inhalt: Protokoll für die Kommunikation zwischen ATM-*Switches*.
  Unterstützung von dynamischen sowie hierarchischen ATM-Netzen, Routingfunktionen (auch QoS-Routing).
  Veröffentlicht im März 1996.

- **Traffic Management ver. 4.0** [47]
  Inhalt: Available Bit Rate (ABR), *Quality of Service* (QoS).
  Veröffentlicht im April 1996.

In Arbeit befindliche Spezifikationen

- **UNI 4.0**
  Inhalt: Erweiterung der UNI 3.1-Spezifikation wie z.B. Unterstützung von ABR und MPEG, neue QoS-Parameter.
  Verabschiedung: 8/96

- **LANE Phase 2**
  Inhalt: Erweiterungen der LANE 1.0. Verwendung mehrerer Server. Protokoll zur Kommunikation zwischen verschiedenen Servern.
  Verabschiedung: 4/97

- **MPOA** [40]
  Inhalt: Routing von unterschiedlichen Protokollen über ATM-Netz.
  Verabschiedung: voraussichtlich erstes Quartal '97.

- **I-PNNI**
  Inhalt: Erweiterung des PNNI auf paketbasierte Netze.
  Verabschiedung: frühestens 1998.

A.2 IETF

Verabschiedete Spezifikationen

- **RFC 1483** [12]
  Inhalt: Methoden zur Verpackung herkömmlicher Pakete in ATM-AAL5-Pakete.
  Veröffentlicht im Juli 1993.

- **RFC 1577** [11]
  Inhalt: Realisierung der IP-Netze über ATM; Netzstrukturen, Adreßauflösung.
  Veröffentlicht im Januar 1994.

- **RFC 1626** [10]
  Inhalt: Maximale Paketgrößen für IP über ATM.
  Veröffentlicht im Mai 1994.

- **RFC 1755** [9]
  Inhalt: Verwendung der ATM-Forum-Signalisierung UNI 3.0/3.1 in *Classical*-IP-Netzen.
  Veröffentlicht im Februar 1995.

- **RFC 1932** [8]
  Inhalt: IP über ATM: Überblick über die Aktivitäten der IETF.
  Veröffentlicht im April 1996.

In Arbeit befindliche Spezifikationen (*Internet-Drafts*[1])

- **RFC1577-Update** [25]
  Inhalt: Erweiterungen des *Classical*-IP-Modells. Unterstützung mehrerer ATMARP-Server. Konfiguration mit Hilfe der MIB.
  Gültig bis 7. Dezember 1996.

- **NHRP** [29]
  Inhalt: Unterstützung des Aufbaus von direkten ATM-Verbindungen über IP-Subnetzgrenzen hinaus.
  Gültig bis Dezember 1996.

- **RSVP** [30]
  Inhalt: Bandbreitenreservierung für IP-Anwendungen.
  Gültig bis November 1996.

- ***Managed Objects* für *Classical*-IP** [26]
  Inhalt: Definition der *Management Information Base* (MIB) zur Unterstützung von *Classical*-IP über ATM, so wie in [25] vorgesehen.
  Gültig bis September 1996.

- **Multicast Support in Classical IP** [28]
  Inhalt: Verwendung eines MARS zur Unterstützung von IP-*Multicasting* in ATM-Netzen.
  Gültig bis September 1996.

- ***IP Broadcast over ATM Networks*** [27]
  Inhalt: Verwendung der IP-*Multicast*-Dienste in ATM-Netzen zur Unterstützung des IP-*Broadcasting*.
  Gültig bis Oktober 1996.

[1] *Internet-Drafts* sind Arbeitsdokumente der IETF und bleiben nur sechs Monate gültig.

Abbildungsverzeichnis

Abkürzungsverzeichnis

AAL	ATM Adaption Layer
ABR	Available Bit Rate
ACR	Allowed Cell Rate
AL	Alignment Field
ARP	Address Resolution Protocol
ATM	Asynchronous Transfer Mode
ATMARP	ATM Address Resolution Protocol
B-ISDN	Broadband Integrated Services Digital Network
BCOB	Broadband Connection Oriented Bearer
BECN	Backward Explicit Congestion Notification
BUS	Broadcast and Unknown Server
CBDS	Connectionless Broadband Data Service
CCR	Current Cell Rate
CDF	Cutoff Decrease Factor
CI	Congestion Indicator
CLP	Cell Loss Priority
CPCS	Common Part Convergence Sublayer
CPI	Common Part Indicator
CRC	Cyclic Redundancy Check
CS	Convergence Sublayer
CSI	Convergence Sublayer Identification
CSMA/CD	Carrier Sense Multiple Access / Collision Detection

DIX	DEC/Intel/Xerox
DLPI	Data Link Provider Interface
DQDB	Distributed Queue Dual Bus
DSAP	Destination Service Access Point
EFCI	Explicit Forward Congestion Indication
ELAN	Emulated LAN
EPD	Early Packet Discard
ER	Explicit Rate
FCS	Frame Check Sequence
FDDI	Fibre Distributed Digital Interface
FEC	Forward Error Correction
GFC	Generic Flow Control
HEC	Header Error Control
IASG	Internet Address Summarization Group
ICR	Initial Cell Rate
IDU	Interface Data Unit
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IGMP	Internet Gateway Message Protocol
IISP	Interim Inter-Switch Signaling Protocol
ILMI	Interim Local Management Interface
InARP	Inverse Address Resolution Protocol
InATMARP	Inverse ATM Address Resolution Protocol
IPX	Internetwork Packet Exchange
ISO	International Organisation for Standardisation
ITU-T	International Telecommunication Union - Telecommunication Standardization Sector
LAN	Local Area Network
LANE	LAN Emulation

LEC	LAN Emulation Client
LECID	LAN Emulation Client IDentifier
LECS	LAN Emulation Configuration Server
LES	LAN Emulation Server
LE_ARP	LAN Emulation Address Resolution Protocol
LI	Length Indication
LIS	Logical IP Subnetwork
LLC	Logical Link Control
MAC	Medium Access Control
MARS	Multicast Address Resolution Server
MCR	Minimum Cell Rate
MIB	Management Information Base
MID	Multiplex IDentifier
MPOA	Multi-Protocol Over ATM
MSS	Maximum Segment Size
NBMA	Non-Broadcast Multiple-Access
NDIS	Network Driver Interface Specification
NHRP	Next Hop Resolution Protocol
NHS	Next Hop Server
NI	No Increase
NLPID	Network Layer Protocol IDentifier
NNI	Network Node Interface
NSAP	Network Service Access Point
OAM	Operation And Maintenance
ODI	Open Datalink Interface
OSI	Open Systems Interconnection
OSPF	Open Shortest Path First
OUI	Organizationally Unique Identifier
P-NNI	Private Network Node Interface

PAD	Padding Bytes
PCM	Pulse Code Modulation
PCR	Peak Cell Rate
PDH	Plesiochronous Digital Hierarchy
PDU	Protocol Data Unit
PID	Protocol IDentifier
PLCP	Physical Layer Convergence Procedure
PM	Physical Medium
PPD	Partial Packet Discard
PT	Payload Type
PVC	Permanent Virtual Connection
QoS	Quality of Service
RDF	Rate Decrease Factor
RED	Random Early Detection
RFC	Request For Comments
RIF	Rate Increase Factor
RIP	Routing Information Protocol
RM	Resource Management
ROLC	Routing Over Large Clouds
RSFG	Route Server Functional Group
RSVP	Resource ReSerVation Protocol
SAAL	Signaling ATM Adaption Layer
SAR	Segmentation And Reassembly
SDH	Synchronous Digital Hierarchy
SDU	Service Data Unit
SEAL	Simple and Efficient Adaption Layer
SMDS	Switched Multi-megabit Data Service
SMI	Structure of Management Information
SN	Sequence Number

SNAP	SubNetwork Attachment Point oder SubNetwork Access Protocol
SNMP	Simple Network Management Protocol
SNP	Sequence Number Parity
SRTS	Synchronous Residual Time Stamp
SSAP	Source Service Access Point
SSCOP	Service Specific Connection Oriented Protocol
SSCS	Service Specific Convergence Sublayer
ST	Segment Type
SVC	Switched Virtual Connection
TC	Transmission Convergence
TCP	Transmission Control Protocol
UBR	Unspecified Bit Rate
UDP	User Datagram Protocol
UNI	User Network Interface
VBR	Variable Bit Rate
VC	Virtual Connection
VCC	Virtual Channel Connection
VCI	Virtual Channel Identifier
VLAN	Virtual Local Area Network
VPI	Virtual Path Identifier

Literaturverzeichnis

[1] ITU-T, Recommendation I.321: *B-ISDN Protocol Reference Model and its Application.* Geneva, 1991.

[2] ITU-T, Recommendation I.326: *B-ISDN ATM Adaptation Layer (AAL) Functional Description.* Geneva, 1992.

[3] ITU-T, Recommendation I.361: *B-ISDN ATM Layer Specification.* Geneva, 1992.

[4] ITU-T, Recommendation I.363: *B-ISDN ATM Adaptation Layer (AAL) Specification.* Geneva, 1992.

[5] ITU-T, Recommendation I.432: *B-ISDN User Network Interface – Physical Layer Specification.* Geneva 1991.

[6] ITU-T, Recommendation Q.2931: *User Network Interface Layer 3 Specification for Basic Call Connection Control.* Geneva 1995.

[7] ISO/IEC TR 9557: *Protocol Identification in the Network Layer.* October 1990.

[8] RFC 1932: R. Cole, D. Shur, C. Villamizar. *IP over ATM: A framework Document.* April 1996.

[9] RFC 1755: M. Perez, F. Liaw, D. Grossman, A. Mankin, E. Hoffman & A. Malis. *ATM Signaling Support for IP over ATM.* February 1995.

[10] RFC 1626: R. Atkinson. *Default IP MTU for use over ATM AAL5.* May 1994.

[11] RFC 1577: M. Laubach. *Classical IP and ARP over ATM.* January 1994.

[12] RFC 1483: J. Heinanen *Multiprotocol Encapsulation over ATM Adaptation Layer 5.* Finland, July 1993.

[13] RFC 1340: J. Postel and J. Reynolds. *Assigned Numbers.* ISI, July 1992.

[14] RFC 1323: V. Jacobson, R. Braden and D. Borman. *TCP Extensions for High Performance.* May 1992.

[15] RFC 1293: T. Bradley and C. Brown. *Inverse Address Resolution Protocol.* January 1992.

[16] RFC 1191: J. MOGUL AND S. DEERING. *Path MTU Discovery.* November 1990.

[17] RFC 1122: R. BRADEN. *Requirements for Internet Hosts – Communication Layers.* October 1989.

[18] RFC 1112: S. DEERING. *Host Extensions for IP Multicasting.* August 1989.

[19] RFC 1042: J. POSTEL AND J. REYNOLDS. *Standard for the transmission of IP datagrams over IEEE 802 networks.* ISI, February 1988.

[20] RFC 879: J. POSTEL. *TCP maximum segment size and related topics.* 1983.

[21] RFC 826: D. PLUMMER. *Ethernet Address Resolution Protocol – or – Converting network protocol addresses to 48.bit Ethernet address for transmission on Ethernet hardware.* MIT, November 1982.

[22] RFC 793: J. POSTEL. *Transmission Control Protocol.* 1981.

[23] RFC 792: J. POSTEL. *Internet Control Message Protocol.* September 1981.

[24] RFC 791: J. POSTEL. *Internet Protocol.* 1981.

[25] INTERNET-DRAFT: M. LAUBACH, J. HALPERN. *Classical IP and ARP over ATM.* June 7, 1996.

[26] INTERNET-DRAFT: M. GREENE, J. LUCIANI, K. WHITE. *Definitions of Managed Objects for Classical IP and ARP Over ATM Using SMIv2.* February 1996.

[27] INTERNET-DRAFT: T. J. SMITH, G. ARMITAGE. *IP Broadcast over ATM Networks.* April 12, 1996.

[28] INTERNET-DRAFT: G. ARMITAGE. *Support for Multicast over UNI 3.0/3.1 based ATM Networks.* February 22, 1996.

[29] INTERNET-DRAFT: D. KATZ, D. PISCITELLO, B. COLE & J.V.LUCIANI. *NBMA Next Hop Resolution Protocol (NHRP).* June 1996.

[30] INTERNET-DRAFT: R. BRADEN, L. ZHANG, S. BERSON, S. HERZOG & S. JAMIN. *Resource ReSerVation Protocol (RSVP).* May 6, 1996.

[31] S. BERSON: *"Classical" RSVP and IP over ATM.* INET'96, April 1996.

[32] ATM-FORUM/96-0258: A. DEMIRTJIS, S. BERSON, B. EDWARDS, M. MAHER, B. BRADEN AND A. MANKIN. *RSVP and ATM Signalling.*

[33] ATM FORUM: *User-Network Interface Spezification version 3.0.* Prentice-Hall, September 1993.

[34] ATM FORUM: *User-Network Interface Specification version 3.1.* Foster City, September 1994.

[35] ATM FORUM: *Interim Inter-Switch Signaling Protocol.* Foster City, February 1995.

[36] ATM-FORUM: *LAN Emulation Over ATM. Version 1.0.* January, 1995.

[37] ATM-FORUM: *LAN Emulation over ATM, Version 1.0 Addendum.* December, 1995.

[38] ATM-FORUM: *LAN Emulation Client Management Specification, Version 1.0.* September 1995.

[39] ATM-FORUM: *Private Network-Network Interface Spezifikation Version 1.0 (PNNI 1.0).* March 1996.

[40] ATM-FORUM: C. BROWN. *Baseline Text for MPOA (atm95-0824r6).* February 26, 1996.

[41] E. ANDREWS: *MPOA Ties It All Together.* Data Communication, April 1996.

[42] L. KLEIN: *MultiProtocol Over ATM (MPOA).* ≪`http://www.gmd.de/People/Lothar.Klein/papers/MPOA/`≫.

[43] M. HEIN, T. VOLLMER: *Zukünftige Verfahren für LAN-ATM-Kommunikation.* LANline 6/1996, S.110.

[44] A. ROMANOV, S. FLOYD. *Dynamics of TCP Traffic over ATM Networks.* May 1995.

[45] D. FLOYD, V. JACOBSON. *Random Early Detection Gateways for Congestion Avoidance.* IEEE/ACM Translation on Networking, Vol. 1, pp. 397-413, August 1993.

[46] ATM-FORUM/95-0059: J.HEINANEN. *Signalling support for frame discard.* Februar 1995.

[47] ATM-FORUM: *Traffic Management Specifikation, Version 4.0.* April 1996.

[48] VU, DUY LOI: *Verkehrs- und Überlaststeuerung in ATM-basierten LANs.* DATACOM 2/1996, S.150.

[49] ATM-FORUM NEWSLETTER: *The Available Bit Rate Service.* October 1995.

[50] R. JAIN: *ABR Service on ATM Networks: What is it?* Network World, June 24, 1995.

[51] *ATM Net Management: A Status report.* Data Communication, Sept. 1995.

[52] A. ALLES: *ATM Internetworking.* CISCO Systems, May 1995.

[53] L. KLEIN: *Angeschoben. IP über ATM mit Classical IP oder LAN-Emulation.* iX 10/1995, S.170.

[54] F.J. KAUFFELS: *Lokales ATM: Der Lange Weg.* DATACOM 11/1995, S. 74.

[55] DATACOM-BLICKPUNKT: *Netz-Kopplung: Schalten und Walten im Netz.* DATACOM 3/1996, S.40-98.

[56] P. BOROWKA: *Virtuelle LANs – Allheilmittel für flexible Netzwerke?* DATACOM 11/95, S.60.

[57] HEWLETT-PACKARD: *Netperf: A Network Performance Benchmark.* Revision 2.0. Information networks Division, February 15, 1995.
≪`http://www.cup.hp.com/netperf/NetperfPage.html`≫.

[58] F. BROCKNERS, C. CSEH, M. HORNEFFER: *IP über ATM Performance am Beispiel des RTB-NRW.* 1996.

[59] T. LUCKENBACH, R. RUPPELT, F. SCHULZ: *Performance Experiments within Local ATM Networks.* GMD-FOCUS, June 1994.

[60] L. KLEIN: *Ausgebremst. ATM-LAN in Einsatz.* iX 5/1996, S.128.

[61] R. NIEDERBERGER: *Schnelle Netze für das Metacomputing. Anspruch und Wirklichkeit im RTB-NRW.* Forschungszentrum Jülich (KFA) 1996.

[62] D.E.COMER: *Internetworking with TCP/IP, Volume 1, Principles, Protocols, and Architecture.* Prentice Hall, 1991.

[63] D.E.COMER, D.L.STEVENS: *Internetworking with TCP/IP, Volume 2, Design, Implementation, and Internals.* Prentice Hall, 1991.

[64] D. CONRADS: *Datenkommunikation. Verfahren, Netze, Dienste.* Vieweg Verlag, 1996.

[65] D. CONRADS: *KFAnet – Status und Perspektiven.* Interner Bericht des Forschungszentrums Jülich (KFA), KFA-ZAM-IB-9408, April 1994.

[66] D. CONRADS: *ATM – die Vermittlungs- und Multiplextechnik des Breitband-ISDN.* Interner Bericht des Forschungszentrums Jülich (KFA), KFA-ZAM-IB-9504, März 1995.

[67] M. PRYCKER: *Asynchronous Transfer Mode. Die Lösung für Breitband-ISDN.* Prentice-Hall 1994.

[68] A. BADACH, E. HOFFMANN, O. KNAUER: *High Speed Internetworking. Grundlagen und Konzepte für den Einsatz von FDDI und ATM.* Addison-Wesley 1994.

[69] O. KYAS: *ATM-Netzwerke. Aufbau – Funktion – Performance.* DATACOM-Verlag 1995.

[70] J. CLAUS, G. SIEGMUND: *ATM Handbuch. Grundlagen, Planung, Einsatz.* Hüthig-Verlag, 1995.

Danksagung

Die vorliegende Arbeit wurde als Diplomarbeit in Informatik am Lehrstuhl für Technische Informatik und Computerwissenschaften der Fakultät für Elektrotechnik der Rheinisch-Westfälischen Technischen Hochschule (RWTH) Aachen erstellt.

Dem Lehrstuhlinhaber, Herrn Prof. Dr. F. Hoßfeld, danke ich für die Möglichkeit, diese Arbeit am Zentralinstitut für Angewandte Mathematik (ZAM) des Forschungszentrum Jülich GmbH (KFA) anfertigen zu können.

Herrn Prof. Dr. O. Spaniol, dem Inhaber des Lehrstuhls für Informatik IV an der RWTH Aachen, bin ich für die Übernahme des Koreferates dankbar.

Mein besonderer Dank gilt Herrn Dr. D. Conrads für die Betreuung dieser Arbeit und die vielen konstruktiven Diskussionen. Seine wertvollen Anregungen und fachlichen Ratschläge haben mir sehr beim Verfassen der Arbeit geholfen.

Weiterhin danke ich allen Mitarbeitern des ZAM, die durch ihre Kooperationsbereitschaft auf verschiedene Weise zum Gelingen dieser Arbeit beigetragen haben. Herrn M.Sczimarowsky möchte ich für die gute Zusammenarbeit und die Unterstützung bei der Durchführung der Tests im lokalen ATM-Testbett danken. Bei Frau S. Werner bedanke ich mich für ihre Hilfsbereitschaft, insbesondere bei dem für die Messungen oftmals notwendigen Umkonfigurieren der Testgeräte.

Ferner danke ich den vielen Freunden und Bekannten, die mir bei der Diplomarbeit hilfreich zur Seite standen und viel Verständnis aufbrachten. Vor allem möchte ich Herrn V. Siedt danken, der mir immer bei meinen Problemen mit der deutschen Sprache behilflich war.

Der größte Dank gilt jedoch meinen Eltern, die mich während meiner gesamten Studienzeit tatkräftig unterstützten und die letztendlich diese Ausbildung ermöglichten.