# Telematik on a Spickzettel

Daniel Hahn

17. August 2001

**Zusammenfassung**

Dies ist eine Zusammenfassung der verschiedenen Telematik-Inhalte die ich als Vorbereitung auf meine Prüfung bei Prof. Zitterbart im SS 01 zusammengeschrieben habe. Die Erklärungen in diesem Dokument müssen nicht unbedingt richtig sein, ich habe das ganze zusammengeschrieben um es selber besser zu verstehen. Du wurdest gewarnt. Außerdem sind noch viele Teile unvollständig, z.B. gibt es zu ISDN viel mehr zu sagen als hier steht - und ich werde das Dokument auch nicht fortschreiben

**Nochmal: Dieses Werk wurde zu Übungszwecken angefertigt. Es ist teilweise fehlerhaft und unvollständig. Es ist NICHT geeignet als alleiniges Lehrmaterial für eine Prüfung. Du wurdest gewarnt.**

## Die sieben Schichten

Die sieben Schichten stammen aus dem ISO/OSI Basis-Referenzmodell. Eigentlich sind es nur sieben, weil IBM damals sieben hatte, und das ganze Modell ist leicht konfus. Das Modell geht davon aus, daß jede Schicht ihre Dienste an Dienstzugangspunkten (SAP - Service Access Point) der nächsthöheren Schicht zur Verfügung stellt.

## 1 Bitübertrangsschicht

Auch: Physical Layer. Diese Schicht befasst sich mit dem Transpor einzelner Bits zwischen zwei Geräten.

### Theoretische Grundlagen

#### Fourier-Analyse

M. Fourier entdeckte, daß jede jede periodische Funktion $g(t)$ sich durch die Summe (unendlich) vieler Sinus- und Cosinusfunktionen nachbilden lässt.

$$g(t) = \sum_{n=1}^{\infty} a_n \sin(2\pi n f t) + \sum_{n=1}^{\infty} b_n \cos(2\pi n f t)$$

Dabei ist $f = 1/T$ die Grundfrequenz und $a_n$ und $b_n$ die Amplituden der $n$-ten Harmonischen.

**Bedeutung:** Eine Funktion (also z.B. ein Datensignal) lässt sich als Kombination (möglicherweise unendlich) vieler Sinus- und Cosinusschwingungen verschiedener Frequenz auffassen. Ist der Frequenzraum beschränkt (durch Dämpfung oder per Filter), so können einige der Sinus- und Cossinusschwingungen nicht passieren, und das Ausganssignal wird verfälscht.

### Nyquist-Theorem

Wenn ein Signal eine Bandbreite von $H$ Hertz zur Verfügung hat, kann es durch $2H$ Samples pro Sekunde vollständig wieder hergestellt werden. Eine höhere Abtastrate bringt keinen Vorteil, da ja keine höheren Frequenzen vorhanden sind, die durch die höhere Abtastrate entdeckt werden könnten.

**Dumme Frage:** Warum $2H$ und nicht $H$? (Vielleicht weil nur so jede wichtige Änderung ‚erwischt' wird?

Dies begrenzt die maximale Datenrate auf einem (rauschfreien!) Kanal auf

$$2H \log_2 V \text{ Bit/Sekunde}$$

wenn pro Abtastung $V$ diskrete Signale unterschieden werden können. (Logischerweise kann die Kapazität hier immer noch beliebig groß sein, wenn man $V$ entsprechend wählt).

### Shannon-Theorem

Shannon hat gezeigt, daß in einem verrauschten Kanal maximal

$$Hlog_2(1 + S/N)$$

Bit pro Sekunde übertragen können (egal mit welcher Technik). Im Vergleich zu Nyquist: Es ist nicht möglich, beliebig viele diskrete Stufen in einem verrauschten Kanal zu unterscheiden.

**Rauschabstand:** $(S/N)$ ist im vorigen Beispiel der Rauschabstand, d.h. das Verhältnis zwischen Signal und Rauschen. $S$ ist hier die Signalstärke, und $N$ die Stärke des Rauschens – je höher dieser ist, desto besser für die Übertragung. Der Rauschabstand wird auch in dB (Dezibel angegeben), diese Einheit ist bezeichnet den Wert von $10 \log_1 0(S/N)$. (D.h. RA 10 = 10 db, RA 100 = 20 dB, RA 1000 = 30 dB)

## Übertragungstechniken

### Medien

- Massenspeichermedien
- Vedrilltes Kabelpaar/Twisted Pair (Vedrillung hilf gegen Störungen
- Koaxialkabel, Basisband
- Koaxialkabel, Breitband (mehrere Kanäle, mit Verstärkung)
- Lichtwellenleiter/Glasfaser (s. unten)
- Infrarot
- Funk (Radio/Mikrowelle)
- Satellit
- Laser

**Glasfasern** haben eine besonders hohe Übertragungsrate, sie können Multimode (mehrere Frequenzen in der Faser, billige LED-Technik) oder Single-Mode (nur eine Frequenz in der Faser, teuer, nur Laser, bessere Übertragungsqualität sein.

## Besonderheiten bei der (drahtlosen) Übertragung

### Frequenzband

Radioübertragung: LF 148, 5 bis 283,5 kHz, MF 520 kHz bis 1605,5 kHz, SW 5,9 MHz bis 26,1 MHz, UKW 87,5 bis 108 MHz.

Interessant: DAB 223-230, 1452-1472 MHz, dig. Fernsehen 470-862 MHz. Mobiltelefon: Analog: 450-465 MHz, GSM 890-960 MHz, 1710-1880 MHz u.a. (ca. 500 MHz - 2GHz). Schnurlose Telfone 1-2 GHz (DECT: 1880-1900 MHz). IEEE 802.11: 2.4 GHz

**Generell gilt:** Je höher die Frequenz, desto eher verhält sich die Sendung wie Licht, und desto besser für die Datenübertragung.

**Dumme Frage:** Warum sind höhere Frequenzen besser?

### Signalausbreitung

Eine „Funkzelle“ um einen Sender besteht aus dem Übertragungsbereich, in dem kommuniziert werden kann, dem Erkennungsbereich, in dem eine Sendung erkannt, aber nicht verstanden werden kann, und dem Interferenzbereich, in dem eine Sendung nur noch als Hintergrundrauschen ist. Die Ausdehnung dieser Bereich hängt von verschiedenen Parametern ab.

**Dämpfung:** Im Vakuum nimmt die Leistung des Signals im Quadrat der Entfernung ab. Im wirklichen Leben wird die Ausbreitung außerdem noch von Athmosphärischen- und Geländebedingungen beeinflußt. Je höher die Sendefrequenz, umso stärker werden die Wellen von Hindernissen absorbiert, bis hin zur völligen Abschattung.

**Reflexion, Streuung, Beugung:** Ist ein Hindernis im Vergleich wesentlich größer als die Wellenlänge, so kann es die Wellen reflektieren. An kleinen Objekten kommt es dagegen zur Streuung der Wellen. Zusätzlich kann an Kanten von Hindernissen eine Beugung auftreten.

### Mehrwegeausbreitung

Für ein Signal (durch Streuung und Reflexion) viele verschieden lange Ausbreitungspfade bis zum Empfänger, und die Signale dieser Pfade gehen zu verschiedenen Zeiten dort ein. Dadurch kann einerseits das gerade gesendet Symbol „verschmiert“ werden, und es kann zu Überlappungen mit den nachfolgenden Daten kommen (ISI, Intersymbolinterferenz)

Damit das System dies kompensieren kann, kann der Sender eine bekannte Trainingssequenz senden, und der Empfänger kann darauf seinen Entzerrer entsprechend programmieren.

### Regulierung der Frequenzen

Die Frequenzen werden meist durch nationale Behörden (USA: FCC, Europa: CEPT) vergeben. Internationale Koordinierung durch ITU (International Telecommunications Union) in Genf. (Unterteilung nach Region 1: Europa, Mittlerer Osten, Afrika, Region 2: Amerika, Grönland, Region 3: Ferner Osten, Australien, Neuseeland)

## Modulationsverfahren

### Basisbandmodulation

Das einfachste: Strom/kein Strom. Abgesehen von den Gleichstromanteilen ist es für die Funkübertragung auch nicht besonders geeignet.

#### Amplitudenmodulation

Die Amplitude der Schwingungen (z.B. Sinusschwingungen) wird variiert. Einfachster Fall: Sinus/kein Sinus. Für Glasfaserübertragung geht das (Licht/kein Licht), im Funkverkehr wird diese Methode stark von Störungen beeinflusst.

#### Frequenzmodulation

Die Frequenz der Schwingung wird variiert, um Daten zu übertragen (z.B. niedrige Frequenz/hohe Frequenz). Spezielle Modulatoren können Phasensprünge verhindern (Continuous Phase Modulation, CPM). Bei Phasensprüngen würden unerwünscht hohe Frequenzen auftreten.

#### Phasenmodulation

Die Phase der Sinusschwingung wird verschoben um Daten zu codieren. (z.B. Sinus normal, 180 Grad verschoben). Dieses Verfahren wird besonders bei der Modemkommunikation gern angewendet.

#### Kombinationen

Diese Verfahren können fast beliebig kombiniert werden, z.B. bei Modems gerne Phasen- plus Amplitudenmodulation.

#### Manchester-Codierung

Codierung für Kabelübertragung. Zwar soll auch basisbandmäßig Strom/kein Strom verwendet werden, allerdings sollen Gleichstromteile vermieden werden. Deshalb gibt es in jedem Bit auf jeden Fall einen Übergang: high-¿low bedeutet 1, low-¿high bedeutet 0 (normales Manchester. Differenzielle Manchester: Anfang des Intervalls ist wichtig: Ein Übergang bedeutet 0, kein Übergang bedeutet 1. Nachteil dieser Methode: Die Übertragungsrate wird halbiert (da doppelte Frequenz benötigt wird).

### Multiplexverfahren

#### Raummultiplex

Niemand merkt, was auf der anderen Seite des Zimmers passiert – oder in der nächsten Funkzelle.

#### Frequenzmultiplex

Eine Frequenz pro Übertragung. Funktioniert prima einfach, allerdings brauchen wir einen Schutzabstand zwischen den Frequenzen. Eignet sich hauptsächlich für dauerhaft benötigte Kanäle.

#### Zeitmultiplex

Jeder kommt abwechselnd dran. Die Bandbreite kann nach Bedarf neu verteilt werden, allerdings ist Synchronisation erforderlich.

#### Kombiverfahren, Channel Hopping

Alle Verfahren können natürlich auch kombiniert werden. Bei der Kanalspringerei wird hat jeder Kanal bestimmte Zeitschlitze auf verschiedenen Kanälen. Das gibt es in langsam (Frequenz wird alle paar Bit gewechselt) und schnell (Frequenz wird

pro Bit mehrfach gewechselt). Beispiel für schnellen Wechsel: Bluetooth, (GSM optional). Beispiel für langsamen Wechsel: GSM.

#### Codemultiplex

Etwas kompliziert: Alle senden gleichzeitig, und trotzdem klappt es: Jede Station hat eine sogenannte Chippingsequenz (typischerweise 64 oder 128 bit, oder 1000 bei der Army). Die Sequenzen der Stationen sollten orthogonal zueinander sein.

Bei der Übertragung hat ein Bit genau die Länge der Chippingsequenz und wird mit dieser per XOR verknüpft (bzw.: Für eine 0 wird die Chipfolge übertragen, für eine 1 ihr Komplement (oder auch umgekehrt). Dadurch wird die benötigte Bandbreite multipliziert, allerdings können alle Stationen im selben Frequenzbereich übertragen.

Der Empfänger muß mit dem Sender exakt synchronisiert sein, beim Empfang wird das Produkt aus dem empfangenen Signal und der Chipfolge gebildet. Signale anderer Sender werden (da sie orthogonal sind) „herausgekürzt“. Einzelne falsch empfangene Chips werden ausgeglichen, da das Produkt durch einzelne Fehler nur leicht verfälscht wird.

## PCM und SONET

PCM: Standardverfahren für die Telefonie. Es werden 8k Abtastungen pro Sekunde gemacht, laut Nyquist genug für einen 4kHz-Kanal. Jede Abtastung liefert 7 Bit, eine Abtastung erfolgt alle 125 $\mu$S. Zeitintervalle in TK-Einrichtungen sind daher meist Vielfache von 125 $\mu$S.

Die Telefongesellschaften benutzen als kleinste Einheit den **T1-Träger**. Es werden 24 Kanäle übertragen, und zwar in Einheiten von 193 Bit/s (1,544 MBit/s). Jeder Kanal hat pro Block 8 Bit, 7 für Daten, 1 Bit für Steuerung. 1 Bit dient der Rahmenbildung (abwechselnd 0/1). Die CCITT-Empfehlung schlägt vor, 8 Bit für Daten zz verwenden. Entweder wird ein zusätzliches Zeichengabebit am Rahmen angehängt, oder jeder 6 Sample enthalt ein Zeichengabebit (d.h. jeder 6 Sample enthält nur 7 Bit Daten). Die 24 Kanäle werden meist reihum abgetastet.

Man kann in PCM-Kanälen zusätzlich differenzielle oder prädiktive Codierung verwenden (codiere Abstand zum vorherigen und vorhergesagten Wert), um die Bitmenge zu verkleiner.

**SONET** (Syncronous Optical Network) ist ein Standard für optische Übertragungsnetze. SONET besteht aus einer kompletten Architektur: Multiplexer und Repeater. Die Daten werden über einen *Pfad* vom Start bis zum Ziel geleitet. Die Strecke dazwischen besteht aus mehreren *Leitungen* (zwischen verschiedenen Multiplexern, die aus *Abschnitten* (zwischen Multiplexern und Repeatern) bestehen. SONET wird von einem Mastertakt einer Genauigkeit von $10^9$ gesteuert.

Ein SONET-Block besteht aus 810 byte und es werde 8000 pro Sekunde übertragen. Die 810 byte können in einer $9 \times 90$ Tabelle dargestellt werden. Die ersten 3 Spalten sind für Leitungs- und Abschnittsoverhead reserviert (3 Zeilen Leitung, 6 Zeilen Abschnitt). Ein SPE-Block (Synchronous Payload Envelope) ist 87 byte lang, das erste Byte enthält Pfadoverhead. Er kann an einer beliebigen Stelle im Rahmen beginnen, das erste Byte im Abschnittsoverhead enthält einen Zeiger auf den Anfang. Die Datenrate beträgt 51,84 Mbps brutto, und 50,112 Mbps netto.

## Repeater

Ein Repeater verbindet Netzwerke auf der Bitübertragungschicht. Er ist zwar aufwendiger als ein reiner Verstärker, indem er digitale Signale decodiert und reproduziert, weiß jedoch nichts von der Semantik der Daten.

# 2 Sicherungsschicht

Die Sicherungsschicht hat die Aufgabe, eine zuverlässige (gesicherte) Verbindung zwischen zwei benachbarten Kommunikationspartnern herzustellen.

Die Sicherungsschicht kann der Vermittlungsschicht verschiedene Dienstgüten anbieten: Verbindungslos (bestätigt oder unbestätigt) und verbindungsorientiert.

## Rahmenbildung

Die Vermittlungsschicht muß in jedem Fall die Daten irgendwie verpacken. Daher ist es notwendig, den kontinuierlichen Bitstrom, den Schicht 1 zur Verfügung stellt, aufzuteilen. Es gibt mehrer Möglichkeiten, Rahmen zu erzeugen:

Am simpelsten ist die Methode, die eingegangenen Bytes abzuzählen, möglicherweise wird am Anfang des Rahmens jeweils noch dessen Länge eingefügt. Diese Methode versagt jedoch bei Byteverlusten, oder wenn die Länge des Rahmens falsch übertragen wird. Besser ist es, Anfang und Ende eines Rahmens irgendwie zu markieren: Beim **Zeichenstopfen** wird Anfang und Ende des Rahmens durch eine bestimmte Zeichenkombination markiert. Kommt ein Markierungszeichen in den Daten vor, so wird es „escaped “ (z.B. verdoppelt). Die Escape-Codes werden dann beim Empfänger wieder entfernt. Beim **Bitstopfen** wird eine bestimmte Bitkombination als Zeichengrenze eingefügt. Kommt diese Kombination in den Daten vor, so wird ein zusätzliches Bit „eingestopft “, das beim Empfänger wieder entfernt wird.

Eine weitere Methode, Rahmen zu bilden ist die **Coderegeln der Bitübertragungsschicht zu verletzen**. z.B. würde bei der Manchester-Codierung in der Mitte eines Bit kein Übergang stattfinden.

## Fehlererkennung/-korrektur

Fehlererkennung und -korrektur beruht auf dem absichtlichen Einfügen von Redundanz in die Daten.

### Hamming-Abstand

Die Hamming-Distanz zweier Bitfolgen ist die Zahl der Bits, die geändert werden müssen um die eine Folge in die andere zu verwandeln. Dieser Abstand gibt auch an, wie viele Bits verändert werden müssen, damit die beiden Folgen nicht mehr unterscheidbar sind.

## Paritätsbit

Sollte klar sein.

### Hamming-Code

Damit n-bit-Fehler erkannt werden können müß die Hamming-Distanz mindestens $n+1$ betragen (damit der Abstand zum nächsten gültigen Codewort größer ist, als der Fehler. Entsprechend muß für n-bit Fehlererkennung der Abstand mindestens $2n+1$ betragen.

Der Hammingcode kann 1-bit-Fehler mit einer minimalen Anzahl von Prüfbits korrigieren. Dazu werden alle bits Positionen einer Zweierpotenz liegen Prüfbits. Die Prüfbits enthalten die Parität aller bits, die die entsprechende Zweierpotenz in ihrer Zerlegung enthalten (Bsp. 17 -¿ Prüfbits 16 + 1).

Bei einem Bitfehler werden die Werte aller Prüfbits, die falsch sind, addiert um die Position des falschen Bits zu erhalten.

### CRC Prüfsummen

Meist möchte man Fehler nicht korrigieren (zu hoher Overhead), sondern lediglich erkennen. Beim CRC werden die Daten als Polinom aufgefasst, das die Koeffizienten 0 und 1 hat. Außerdem wird ein Prüfpolinom $G(x)$ vereinbart, das beiden Parteien bekannt sein muß. Hat das Prüfpolynom den Grad $r$, so werden $r$ Prüfbits angehängt. Diese enthalten den Rest der Division des Datenpolynoms und $G(x)$. Das Ergebnis läßt sich ohne Rest durch $G(x)$ teilen, bleibt ein Rest bei der Überprüfung, liegt ein Fehler vor.

## Flusskontrolle

Die Flusskontrolle soll verhindern, daß der Empfänger von Daten überschwemmt wird, die er nicht annehmen kann. Außerdem sollen diese Protokolle den Verlust und die Duplizierung von Rahmen verhindern.

### Stop-and-Wait

Ganz einfach: Der Sender wartet vor jedem neuen Rahmen auf die Bestätigung des vorhergehenden. Es wird eine 1-Bit Sequenznummer verwendet, um den Verlust eines Rahmens (bzw. einer Bestätigung) zu erkennen. Für den Fall, das etwas hängt, gibt es Timeouts.

### Sliding Window (Schiebefenster)

Sender und Empfänger haben jeweils ein Fenster von Paketen, die ausstehen dürfen. Ein Sliding Window der Grösse 1 entspricht Stop-and-Wait. Alle Rahmen haben Sequenznummern, ein Fenster enspricht einem Sequenznummernbereich.

Hat der Sender ein Fenster von grösser 1, so kann er ohne Bestätigung Rahmen senden, bis er die Grenze des Fensters erreicht hat. (Alle Rahmen im Fenster müssen gepuffert werden). Erhält er eine Bestätigung für den ersten Rahmen, so wird das Fenster um eine Position weitergeschoben.

Hat der Empfänger ein Fenster grösser 1, so kann er Rahmen puffern, die ausser der Reihe ankommen. Ein Rahmen wird nur bestätigt, wenn alle vorherigen Rahmen korrekt angekommen sind. Sobald der unterste Rahmen bestätigt wurde, wird das Fenster um eine Position weitergeschoben. Rahmen, die außerhalb des Fensters ankommen, werden verworfen.

### Wiederholung von fehlerhaften Rahmen

Wird ein Rahmen vom Empfänger als fehlerhaft erkannt, oder geht er verloren, so muß er nochmals gesendet werden. Beim **Go-Back-N**-Verfahren werden, wenn der Timeout für einen Rahmen abläuft, dieser Rahmen und alle nachfolgenden erneut gesendet, d.h. alle Sendungen seit dem Fehler werden wiederholt. Dies entspricht einem Empfangsfenster von 1. Falls der Empfänger einen Puffer hat, kann stattdessen **Selective Repeat** verwendet werden. In diesem Fall wird nur der fehlerhafte Rahmen neu übertragen, alle anderen hat der Empfänger noch in seinem Puffer.

Eine Optimierung ist die NACK-Nachricht, mit der der Empfänger den Sender anweist, einen bestimmten Rahmen nochmals zu senden.

### Piggybacking

Bestätigungen können auch mit Datenpaketen in die Gegenrichtung „mitverschickt “ werden, allerdings muß man abwägen, wie lange die Bestätigung auf ein Datenpaket warten soll, bevor sie alleine losgeschickt wird.

## 2.1 Sicherungsschicht-Protokolle

### HDLC

HDLC (High Level Data Link Control) ist ein Protokoll der Sicherungsschicht, das in ziemlich vielen verschiedenen Variationen vorkommt. Andere Namen sind LAP und LAPD.

Diese Protokolle sind bitorientiert, ein Rahmen wird durch die Flagsequenz 01111110 an Anfang und Ende begrenzt. Die Rahmen enthalten ein ein Byte grosses Adressfeld und ein ebenso grosses Steuerungsfeld. Die Länge der Übertragenen Daten ist variabel. Das gesamte Paket wird durch eine 2-byte Prüfsumme geschützt (CRC-Variante, die fehlerhafte Flagbits erkennt).

HDLC kennt drei Rahmentypen: Informations-, Steuerungs- und unnummerierte Rahmen. Informationsrahmen enthalten eine Sequenznummer und ein ACK-Nummer (die angibt, welche Sequenznummer von der Gegenstelle als nächstes erwartet wird). Ein P/F-Bit (Poll/Final) war hauptsächlich für pollende Terminalapplikationen gedacht.

Die Steuerungrahmen können verschiedene Typen haben, z.B. Bestätigung (ACKNOWLEDGEMENT), Zurückweisung (REJECT, für go-back-n), selektive Zurückweisung (SELECTIVE REJECT, fordert selektive Wiederholung, u.).

Unnummerierte Rahmen enthalten Steuerinformationen, wie z.B. Verbindungsabbauwünsche (DISC, Disconnect). Außerdem gibt es einen UA (Unnumbered Acknowledgement) Rahmen, der den letzten ausstehenden Steuerrahmen bestätigt (Steuerrahmen haben keine Sequenznummer, es darf immer nur ein Steuerrahmen ausstehen).

### SLIP

SLIP ist ein recht primitives Protokoll zur Übertragung von IP-Paketen über serielle Leitungen (daher auch Serial Line IP). Die IP-Pakete werden einfach in Rahmen eingebaut, die durch ein Flagbyte gebildet werden. Das Flagbyte wird in den Daten durch eine seltsame Escapesequenz umgangen, so daß das erste Byte der Escapesequenz selbst wieder gestopft werden muß... Optimierungen in SLIP betreffen das Weglassen mehrfach vorhandener Header und das inkrementelle übertragen Headerfeldern.

SLIP ist kein Internetstandard und in teilweise inkompatiblen Versionen vorhanden. Es unterstützt nur IP, keine dynamischen Adressen, keine Authentifiezierung und keinerlei Fehlerkontrolle. SLIP wird im Augenblick an fast allen Stellen von PPP abgelöst.

### PPP

PPP ist ein recht leistungsfähiges Protokolle für Punkt-zu-Punkt-Verbidungen. Die Verbindung kann authentifieziert aufgebaut werden, es können fast beliebige Protokolle über PPP übertragen werden und die Partner können die Parameter der Verbindung dynamisch aushandeln.

PPP ist zeichenorientiert, das Rahmenformat ist allerdings ähnlich dem HDLC-Rahmenformat. Auch PPP verwendet das Flagbyte 01111110, es wird allerdings zeichengestopft, wenn es in den Daten vorkommt. Das Adressfeld (11111111) und das Steuerungsfeld (00000011) haben im allgemeinen feste Werte, es kann vereinbart werden, sie wegzulassen. Danach folgen zwei (optional ein) Byte, die das verwendete Protokoll angeben (NCP, LCP, AppleTalk, IP...). Die Daten haben eine variable Länge (bis zu einem vereibarten Maximum), die Prüfsumme hat zwei oder vier Byte.

**LCP**, das Link Control Protocol, dient hauptsächlich zum Auf- und Abbau von PPP-Verbindungen. Mit dem LCP können verbindunsspezifische Parameter (Authetifikation, Art der unterstützten Protokolle, maximale Datenlänge...) ausgehandelt werden.

**NCP**, das Network Control Protocol, ist spezifisch für jeden unterstützten Protokolltyp (z.B. IP oder AppleTalk). Es dient dazu, Parameter für das über PPP betriebene Protokoll auszuhandeln. Bei IP wären dies z.B. die IP-Adressen und die Adressen von Gateways.

# Sicherungsschicht: MAC Teilschicht

Die MAC (Medium Access Control) ist ein Teil der Sicherrungsschicht, und hat die Aufgabe, den Zugriff mehrerer Kommunikationspartner auf ein einzelnes Medium zu koordinieren.

## ALOHA

Einfach und geradeaus: Jeder sendet wann es ihm passt, wenn es eine Kollision gibt: Pech gehabt. Das ganze hat eine maximale Effizienz von 10 Prozent. Wenn man immerhin Zeitschlitze verwendet (Slotted Aloha) verdoppelt sich die Effizienz - allerdings ist dann eine Synchronisation aller Stationen notwendig.

## CSMA

Etwas schlauer ist es, vor dem Senden das Medium abzuhören, um zu sehen ob schon jemand sendet. Im einfachsten Fall (1-persistent CDMA) wird sofort gesendet, wenn das Medium frei ist (Wahrscheinlichkeit 1). Wenn eine Kollision auftritt, wartet jede Station eine zufällige Zeit, bevor sie es noch einmal versucht. Wenn mehrere Stationen auf das Medium warten, tritt auf jeden Fall eine Kollision auf. Um das zu verhindern, kann etweder die Wahrscheinlichkeit gesenkt werden, mit der die Station sendet (z.B. 0,5 das gesendet wird, und 0,5 das noch einen Schlitz abgewartet wird -¿ 0.5-persistent), oder die Station wartet eine zufällige Zeit, wenn das Medium belegt ist, und probiert es dann noch einmal (nonpersistent). Dieses Verfahren heißt Carrier Sense Multiple Access.

## CSMA/CD

Noch besser kann der Zugriff auf das Medium geregelt werden, wenn die angeschlossenen Stationen eine Kollision sofort erkennen, und dann das Medium wieder freigeben. In diesem Fall kann, wenn eine Kollision erkannt wird, gleich ein neuer Versuch zur Belegung gestartet werden.

Bei CSMA/CD besteht die Übertragung aus Perioden, in denen Daten übertragen werden und Konkurrenzperioden. Während einer Konkurrenzperiode versuchen verschiedene Stationen, das Medium zu belegen, und erst wenn keine Kollisionen mehr auftreten, beginnt die eigentliche Sendeperiode.

Die Konkurrenzperiode kann man sich in Zeitschlitze unterteilt vorstellen, die der Zeit entsprechen, die zum Erkennen einer Kollision benötigt werden.

### Kollisionserkennung

Wenn ein Signal eine Zeit von $\tau$ benötigt, um ein Kabel der maximalen Länge zu durchqueren, so kann eine Kollision in einer Zeit von $2\tau$ sicher erkannt werden. Ein Senderahmen muß mindestens so lang sein, daß eine Kollision einwandfrei erkannt wird, d.h. er muß aus einer Übertragung der Länge $2\tau$ bestehen.

Ein konkurrierender Zugriff wird daher nach ca. $2\tau$ abgebrochen. (Dies ist der Zeitschlitz für die Konkurrenzperiode).

#### Exponential Binary Backoff

Dies ist eine Methode, mit der der Zugriff auf ein Medium für viele Stationen organisiert werden kann. Sobald das Medium nach einer Sendung wieder freigegeben wird, wartet eine Station zufällig 0 oder 1 Zeitschlitze bevor sie zu senden versucht. Kommt es zu einer Kollision, wird die Anzahl der Wartschlitze verdoppelt (d.h. es wird jetzt 0-3 Schlitze gewartet). Diese Verdoppelung tritt nach jeder Kollision ein, solange bis eine Station kollisionsfrei sendet, oder ein Maximalwert erreicht wird.

### RTS/CTS, versteckte und ausgelieferte Geräte

Kollisionserkennung läßt sich bei Funknetzen nicht durchführen, da der Sender prüft, ob das Medium belegt ist, Kollisionen aber beim Empfänger auftreten. Bei der Funkübertragung können allerdings nicht alle Stationen alle anderen Stationen empfangen.

#### Versteckte Endgeräte

Die Station A sendet an die Station B. Station C liegt in Reichweite von B, nicht jedoch von A. Wenn nun C an B senden möchte prüft sie, ob das Medium belegt ist. Da sie die Signal von A nicht empfangen kann, beginnt sie zu senden. Bei B treten nun Kollisionen auf.

#### Ausgelieferte Endgeräte

In der selben Situation möchte B an C senden, während A mit einer anderen Station kommuniziert, die außerhalb der Reichweite von B und C liegt. B findet das Medium belegt, und sendet nicht an C, obwohl die Sendung von A nicht stören würde.

#### Request to send/Clear to send

Diese Probleme lassen sich mit einem einfachen Mechanismus lösen: Bevor ein Rahmen gesendet wird, schickt die sendenden Station A ein RTS (Request To Send) Signal. Die Empfangsstation B bestätigt dieses Signal mit CTS (Clear To Send) und der Kennung der sendenden Station. Alle Stationen in Reichweite des Empfängers wissen nun, das er ein Signal von A empfängt und werden erst nach dem Ende dieser Übertragung wieder eine Verbindung versuchen. Wenn der RTS-Rahmen kollidiert wird nur eine relativ kurze Übertragung (der RTS-Rahmen verloren). Falls der RTS-Rahmen verlorgen geht (d.h. es wird kein CTS empfangen) wird ein Backoff-Verfahren angewendet.

Dieses Verfahren erforder einen erhöhten Overhead, so daß es z.B. bei 802.11 optional ist.

### Kollisionsfreie Verfahren u.a.

Es ist auch möglich, den Zugriff auf das Medium so aufzuteilen, daß keine Kollisionen auftreten. Diese Verfahren sind allerdings nicht so verbreitet wie die Kollisionbehafteten.

### Bitmusterprotokoll

Jede Station hat in der „Konkurrenzphase “ ein reserviertes Bit, welches gesetzt wird, wenn die Station übertragen möchte. Nach dieser Phase wissen alle Stationen Bescheid, wer senden möchte und jede Station darf der Reihe nach einen Rahmen senden.

### Binärer Countdown

Alle Stationen senden gleichzeitig ihre Adressen, dabei wird eine 0 von 1-Bits überschrieben. Alle Stationen, die bemerken daß ihr Adressbit überschrieben wurde, scheiden sofort aus (d.h. nur noch Stationen mit einer 1 an der aktuellen Stelle machen weiter). Am Ende gewinnt die Station mit der höchsten ID, die ihren Rahmen senden darf.

### Adaptive Tree Walk

Noch ein etwas esoterisches Protokoll: Es beruht darauf, das die Stationen in einem binären Baum angeordnet sind. Wenn eine Kollision auftritt, scheidet eine Seite des Baumes aus, und das Protokoll wird mit dem rechten oder linken Unterbaum fortgesetzt.

## Bridging

Eine Bridge ist ein Gerät, daß zwischen zwei (oder mehr) Netzwerken auf Sicherungsschicht-Ebene Vermittelt. Eine Bridge kann evtl. auch Rahmen von Format des einen Netzwerk in ein anderes umwandeln (z.b 802.3 nach 802.5).

### Transparente Bridges

Eine transparente Bridge kann in ein Netzwerk eingefügt werden, ohne daß die Stationen davon etwas mitbekommen. Die Bridge arbeitet ohne Eingriff und routet die Rahmen mit dem Backward-Learning-Verfahren (s. Routing/Vermittlungsschicht).

### Spanning Tree

Falls mehrere Bridges an ein LAN angeschlossen sind, verwenden sie den Spanning-Tree-Algorithmus um zu verhindern, daß Schleifen entstehen. (Dafür tauschen die Bridges untereinander Informationen aus). Die Bridge mit der niedrigsten ID wird die Wurzel des Baumes.

Jede Bridge kennt die Kosten eines ihrer Lan-Anschlüsse. Kennt eine Bridge bereits einen günstigen Weg in Richtung Wurzel, sendet sie dessen Kosten auf alle angeschlossenen LANs. Ein Bridge, die diese Information empfängt, wählt jetzt den Nachbarn aus, über den der optimale Pfad zur Wurzel aus. (d.h. eigene Kosten bis zum Nachbarn + dessen Kosten bis zur Wurzel sind minimal). Rahmen werden von Bridges jetzt nur noch entlang des Baumes weitergeleitet, ansonsten ist alles wie vorher.

Ein LAN wird seinen Verkehr immer über diejenige Bridge versenden, die die beste Anbindung zur Wurzel hat (designated Bridge).

### Source Routing Bridges

Diese Art von Bridges verlangt, daß die sendende Station den kompletten Pfad bis zum Ziel einträgt. Dies macht die Arbeit der eigentlichen Bridge zwar einfacher, ist aber nicht transparent. Außerdem müssen alle Stationen die Netztopologie kennen,

was mit einem hohen Aufwand (z.B. quadratisch viele Nachrichten) verbunden ist. Diese Technik wurde bei Bridges für 802.5 verwendet.

## Spezielle MAC-Systeme

### 802.3 (Ethernet)

Ethernet hat den Vorteil simpel (und billig) zu sein, eine Eigenschaft die sich durchsetzt. Ethernet arbeitet mit einem CSMA/CD-Verfahren mit exponentiellem Backoff. Es gibt verschiedene Verkabelungsvarianten (Thick, Thin, Twisted Pair, Glasfaser) und inzwischen auch verschiedene Geschwindigkeiten. In jedem Fall kann jedoch jede angeschlossene Station alle anderen hören, und die maximale Länge eines Segmentes beträgt 2500 Meter (wichtig für die Kollisionserkennung).

Ein Ethernet-Rahmen besteht aus einer 7-byte-Präambel (01010...), die zur Synchronisation dient. Dann folgt ein Startcode (10101011) als Anfangsmarkierung. Die Quell- und Zieladresse haben jeweils 6 Byte (möglich wären auch 2), dann folgen 2 Byte die die Länge der Daten angeben (max. 1500 Byte). Falls der Rahmen weniger als 64 Byte lang wäre, wird ein Padding-Feld eingefügt um ihn auf diese Mindestgröße zu verlängern (die Mindestgröße ist erforderlich, um bei einer Länge von 2500m noch Kollisionen zu erkennen. Der Rahmen wird durch eine 4-byte-Prüfsumme (CRC) geschützt. Ethernet-Adressen können lokal oder global sein, die Adresse 111.... dient als Broadcastadresse.

Der binäre Backoff verdoppelt die Größe des Konkurrenzfensters bis zu einem Maximalwert von 1023 Schlitzen, dann (nach der 10 Kollision) wird die Größe eingefroren. Nach 16 Kollisionen gibt der Algorithmus auf.

Aufgrund seiner Einfachheit hat Ethernet auch einige Nachteile: Die Zustellung eines Rahmens in einer gewissen Zeit kann nicht garantiert werden und es werden keine Prioritäten unterstützt. Ethernet eignet sich also nicht für Echtzeitanwendungen. Außerdem ist die mögliche Auslastung unter hoher Last nicht gerade optimal.

**Fast Ethernet** ist im Prinzip das gute alte Ethernet, nur daß der Takt der Manchester-Codierung verzehnfacht wurde und so statt 10 Mbit/s jetzt 100 Mbit/s übertragen werden können. Fast Ethernet unterstützt nur noch die Verkabelung über Hubs, mit Kabel der Kategorien 3 und 5 oder Glasfaser.(Kat 3 arbeitet mit vier verdrillten Kabelpaaren (zum Hub, vom Hub, und zwei umschaltbare).

**Switches** können Ethernet nochmals verbessern, indem der Switch Pakete an eine bestimmte Station nur an den Ausgang dieser Station kopiert. Andere Stationen werden also von dieser Kommunikation nicht belästigt.

### 802.4 (Token Bus)

Token Bus war der Versuch die Probleme des Ethernet zu lösen und ein System zu schaffen, daß sich besonders für Fertigungsstrassen eignet. Token Ring ist eine unnötig komplexe Kopfgeburt, die wohlverdient untergegangen ist.

Das Grundprinzip ist noch relativ elegant: Obwohl die Stationen physikalisch an einem Bus hängen, bilden sie einen logischen Ring. Eine Station darf nur senden, wenn sie das Token besitzt und sendet es, nachdem sie fertig ist an die logisch nächste Station weiter. Dadurch kommt jede Station garantiert regelmäßig an die Reihe. Allerdings besteht jede Station aus sechs Unterstationen, die jeweils eine der sechs Prioritätsklassen bedienen. Der höchsten Priorität wird ein gewisser Anteil der Bandbreite garantiert.

Die Rahmen von Token-Ring unterscheiden sich etwas von den Ethernet-Rahmen. Die Präambel ist nur ein Byte lang, und jeder Rahmen hat ein Start- und Endesignal das analog (!) codiert ist, deshalb muß keine Rahmenlänge übertragen werden.

Die Probleme von Token Ring fangen richtig an, wenn zusätzliche Stationen eingefügt werden müssen. Jede Station kann bietet, wenn sie das Toke hat, anderen Stationen (und zwar nur solchen, die nach ihr in den Ring einfügt werden können) an, ihr Nachfolger zu werden. Wenn sich mehrere Stationen melden, treten Kollisionen auf, und die ursprüngliche Station versucht dies zu beheben. Die genaue Zeit, um in den Ring einzutreten, läßt sich nicht vorhersagen.

Werden beide Nachfolgestationen einer Station abgeschaltet (d.h. sind plötzlich nicht erreichbar, muß der Ring neu initialisiert werden.

Token Ring untestützt bis zum 10 MBit/s.

### 802.5 (Token Ring)

Der Grundgedanke von Token Ring ist derselbe wie beim Token Bus: Nur die Station, die gerade das Token besitzt, darf senden. Der auffälligste Unterschied besteht darin, daß ein Token Ring auch physikalisch als Ring organisiert ist, d.h. eine Station kommuniziert nur mit ihren unmittelbaren Nachbarn direkt. Jede Station hat eine Ein- und eine Ausgangsschnittstelle. Normalerweise sind die Stationen miteinander über ein *Wire Center* verbunden, das eine ausgefallenen oder unterbrochene Station überbrücken kann.

Eine Station kopiert im Normalfall (Lesebetrieb) alle Daten von ihrer Eingangschnittstelle mit einer 1-Bit Verzögerung auf ihre Ausgangschnittstelle. Falls die Station das Schreibrecht hat, werden Ein- und Ausgang entkoppelt, die Station sendet auf ihrem Ausgang Daten und leitet die Daten, die sie auf dem Eingang erhält, nicht weiter.

Im Ruhezustand kreist auf dem Ring ein 3 Byte langes Token, das von den Stationen weitergegeben wird. Der Ring muß mindestens so lang sein, daß das komplette Token auf ihm Platz findet, ist dies nicht der Fall, wird eine zusätzliche Verzögerung eingefügt. Sobald eine Station senden will, wartet sie auf das Token, ändert ein Bit um es zu einer normalen Startfolge zu machen und sendet dann den Rest des Rahmens. Die Station darf jetzt bis zu einer maximalen Tokenhaltezeit Rahmen senden, sobald sie fertig ist, erzeugt sie ein neues Token und geht zurück in den Lesebetrieb.

Ein Token-Ring-Rahmen besteht aus eine Start- und einem Endebegrenzer von einem Byte die ungültige Manchester-Folgen enthalten um sie von Daten zu unterscheiden. Nach dem Endebegrenzer folgt noch ein Rahmenstatusbyte, in dem die Empfangstation Bestätigungen bzw. den Übertragungsstatus eintragen kann. Der Kopf des Rahmens besteht außer aus dem Startbegrenzer aus einem Zugriffsteuerungsbyte, daß Bits für Reservierung, Prioritäten und Überwachungszwecke enthält. Das Rahmensteurungsbyte codiert verschiedene Arten von Überwachungsrahmen. Der Rest des Token-Ring Rahmens besteht aus Start- und Zieladresse, der Checksumme und den Daten, die hier beliebig lang sein dürfen.

Token Ring unterstützt drei **Prioritäten**, eine Station darf nur senden wenn sie ein Token gleich oder kleiner der anstehenden Priorität erhält. Eine Station kann ein Reservierungsbit setzen, um ein Token einer gewissen Priorität anzufordern, eine Reservierung mit hoher Priorität hat dabei Vorrang. Es gibt einige Mechanismen, die erreichen sollen, daß die Priorität nach der Übertragung auch wieder heruntergesetzt wird, allerdings kann es vorkommen daß eine Station mit niedriger Priorität verhungert.

Eine **Monitorstation** übernimmt in Token Ring die Administrationsaufgaben, fügt Verzögerungen ein falls notwendig, entfernt kreisende Rahmen ohne Heimat und sonstigen Müll vom Ring und ähnliches. Jede Station kann potentiell der Monitor des Rings sein, beim Ausfall des Monitors übernimmt eine andere Station diesen Job. Es kann natürlich vorkommen, daß der Monitor eine Fehlfunktion hat und

damit das Netz zum Erliegen bringt (Anhänger von Token Bus glauben daher, ihr System wäre zuverlässiger...).

Token Ring unterstützt bis zum 8(16) MBit/s.

### 802.11 (Wireless LAN)

802.11 ist ein neues Miglied in der 802.x-Familie zum Aufbau von drahtlosen Netzwerken. Naturgemäß ist es etwas komplizierter als die drahtgebundenen Protokolle: 802.11 unterstützt verschiedene Übertragungsarten, verschiedene MAC-Zugriffsverfahren, Stromsparmodi und Roaming.

Die **Übertragungsarten**, die unsterstützt werden sind eine Funkübertragung mit Frequenzsprungverfahren, eine Funkübertragung mit DSSS (Direct Sequence Spread Spectrum) und Infrarotübertragung. Es wird eine Übertragungsgeschwindigkeit von 1 MBit/s unterstützt, optional 2 MBit/s, moderne Varianten unterstützen auch 11 MBit/s. Die **Infrarotübertragung** hat dabei nur eine Reichweite von ca. 10 Metern, und setzt keine direkte Sichtverbindung voraus. Die Funkbasierten Verfahren arbeiten im lizenfreien Bereich um 2,4 GHz und haben eine Reichweite im Bereich um 50m (in Gebäuden). Die Übertragungstechniken haben jeweils eigene Paketformate für die physikalische Schicht. Alle drei Verfahren bieten ein CCA (Clear Channel Assement) Signal an, das angibt ob das Medium frei ist. Bei beiden Funkverfahren werden die Daten mit einem Polynom zerwürfelt (scrambling), um Gleichstromanteile zu vermeiden und das Signal zu glätten.

Ein Verfahren zur Funkübertragung ist ein **Frequenzsprungverfahren**. Die Daten werden mit einer Gausschen Frequenzmodulation codiert, es werden 2 (1 MBit/s) bzw. 4 (2 MBit/s) Frequenzen benutzt. Die Station wechselt die Kanäle mit einer pseudozufälligen Sprungsequenz, so daß mehrere Netzwerke im gleichen Raum betrieben werden können. Es stehen 79 Kanäle (Japan: 23) zur Verfügung. Das Paket dieser Schicht besteht aus einer Präambel (80 bit Synchronisation und 2 Startbyte), einer Längenangabe für die Folgenden Nutzdaten (12 Bit), einem 4-bit Steuerungscode der die Datenrate der Nutzdaten angibt (der Header wird immer bei 1 MBit/s übertragen) und einer 4 byte langen Checksumme. Danach folgen die Nutzdaten.

Das am weitesten verbreitete Verfahren ist allerdings **DSSS**. Hier werden die Daten per Phasenmodulation übertragen (Modulation je nach Übertragungsrate), und das Signal wird mit einem 11 Bit langen Barker-Code gespreitzt. Dieses Verfahren ist stabiler, erforder allerdings auch leistungsfähigere Sender und Empfänger. Eine neue Variante des DSSS-Verfahren kann bereits Datenraten von 11 MBit/s übertragen. Der Datenrahmen der DSSS-Schicht unterscheidet sich nur unwesentlich von dem beim Frequenzsprungverfahren. Es gibt wieder eine Präambel (diesmal mit 128 Bit Synchronisationsmuster) und Felder für die Länge der Daten, die Übertragungsgeschwindigkeit und eine Prüfsumme. Zusätzlich ist ein Feld für spätere Verwendung hinzugekommen, daß für weitere Dienste verwendet werden soll. Die Felder haben teilweise etwas andere Längen als beim Frequenzsprungverfahren.

Die **MAC-Schicht** in 802.11 unterstützt wiederum verschiedene Verfahren: Ein CSMA/CA-Verfahren für ad-hoc Netzwerke, ein optionales Verfahren mit RTS/CTS und ein kollisionsfreies Polling-Verfahren mit einer Masterstation. Die Verfahren verwenden drei verschiedene Wartezeiten, eine sehr kurze (SIFS) zum Kollisionsfreien Versenden von Bestätigungen u.ä., eine mittler für den privilegierten Zugriffe (PIFS) der Masterstation und eine lange Wartezeit (DIFS) für konkurrierenden Zugriff.

Beim **CSMA/CA-Verfahren** prüft eine Station zuerst (mit dem CCA-Signal), ob das Medium belegt ist. Ist es für mindestens die Zeit von DIFS frei, kann sie sofort zu senden beginnen. Ist das Medium belegt, müssen alle Stationen die senden wollen zunächst eine Zeit von DIFS abwarten. Ist das Medium dann immer noch frei, treten sie in eine Konkurrenzperiode ein: Jede Station wählt eine zufälli-

ge Zahl, und wartet nach DIFS noch entsprechend viele Zeitschlitze ab. Beginnt vorher eine andere Station zu senden, bricht sie ihren Versuch ab. Beginnen zwei Station im selben Zeitschlitz, kommt es zu einer Kollision. Stationen, die ihren Versuch abgebrochen haben, müssen nur noch die „Restzeit“ abwarten, bis sie es wieder versuchen können. Falls Kollisionen auftreten, wird wieder ein exponentielles Backoff-Verfahren verwendet: Die Anzahl der Zeitschlitze ist minimal 7, nach jeder Kollision wird sie bis zu einem Maximalwert von 256 verdoppelt. Ein Paket wird vom Empfänger sofort bestätigt, die Wartezeit bis zum Senden der Bestätigung beträgt nur SIFS, so daß die Bestätigung in jedem Fall Kollisionsfrei versendete wird.

CSMA/CD kann mit einem **RTS/CTS**-Mechanismus kombiniert werden. Jede Station muß diesen Mechanismus unterstützen, aber die Anwendung ist optional: RTS/CTS erzeugt eine Menge zusätzlicher Nachrichten, und kann oft weggelassen werden. Wenn eine Station den Mechanismus anwenden will, sendet sie (im normalen Konkurrenzverfahren einen RTS-Rahmen, der die Adresse des Empfängers und die Länge der Datenübertragung enthält. Alle Stationen, die diesen Rahmen hören passen ihren NAV (Net Allocation Vector) an, und werden das Medium für die vorgegebene Zeit nicht belegen. Der Empfänger sendet dann (nach SIFS) ein CTS-Signal, alle Stationen die dieses hören passen ihren NAV ebenfalls (neu) an. Danach erfolg die Sendung, die von keiner anderen Station unterbrochen wird. 802.11 unterstützt auch ein **Fragmentierungsverfahren** (damit nicht zu große Blöcke übertragen werden. In diesem Fall enthält die Bestätigung eines Fragmentes gleichzeitig die Reservierung des Netzes für das nächste Paket.

Das **Polling-Verfahren** teilt die Zeit in sogenannte Superrahmen ein, die aus einer Polling-Phase und einer Konkurrenzphase bestehen (diese funktioniert mit CSMA/CD). Da die Konkurrenzphase das Polling verzögern kann, kann sie auch ganz weggelassen werden. Beim Polling fragt die Masterstation (nach PIFS, also bevor jemand anderes drankommt) die einzelnen Stationen ab, und diese können auf die Abfrage mit ihren Daten (nach SIFS) antworten. Darauf fragt die Masterstation (nach SIFS) die nächste Station ab. Die Masterstation zeigt den Beginn der Konkurrenz mit einem bestimmten Signal (Contention Free End) an.

Der **MAC-Rahmen** von 802.11 enthält Kontrollinformationen, eine Sequenznummer, Daten über die Dauer der Übertragung (für die RTS/CTS-Reservierung), Daten (bis 2312 bit) und 4 Adressen. Diese dienen zur Unterstützung des Roaming. Wenn die Stationen untereinander kommunizieren, enthalten zwei der Adressfelder die benutzten Adressen, das dritte die Adresse des Netzwerkes (BSSID Basic Service Set Identifier). Werden Daten über einen Zugangspunkt versandt, ist sowohl die physikalische Adresse des Zugangspunktes als auch die logische Adresse des Partners enthalten.

Wichtig für die Stromsparmodi ist eine genaue **Zeitsynchronisation**. Zu diesem Zweck sendet die Masterstation (falls vorhanden) in regelmässigen Abständen ein Timestamp-Paket (Beacon) aus. Falls es sich um ein Ad-Hoc-Netzwerk handelt, wird jede Station zum vorgesehenen Zeitpunkt versuchen ein Beacon zu senden, auf das sich die anderen synchronisieren. Die Pakete werden immer möglichst nah am vorgesehenen Zeitpunkt gesendet. Verschiebt sich ein Timestamp-Paket, verschieben sich die nachfolgenden nicht.

Um **Strom zu sparen** kann sich eine 802.11-Station vorübergehend abschalten. Andere Stationen müssen dann die Daten speichern, bis sie wieder erwacht. Gibt es eine Masterstation, übernimmt diese das Zwischenspeichern der Daten. Die anderen wachen regelmäßig auf, um das Beacon zu hören. Zusammen mit dem Beacon wird auch eine Traffic Idenfication Map gesendet, die angibt für welche Stationen Daten vorliegen. Diese bleiben dann wach. In einem Ad-Hoc-Netzwerk überträgt jede Station die senden will eine (A)TIM, die entsprechende Station bleibt dann wach. Dies skaliert schlecht, da bei einem vollen Netz sehr viele Tabellen übertragen werden.

#### 802.6 DQDB

Distributed Queue Dual Bus: Ein Protokoll für MANs (ungefähr die Grösse einer Stadt). Die Besonderheit ist, daß es zwei unidirektionale Leitungen gibt, auf denen die Pakete weitergereicht werden. Die Station sind als dezentrale FIFO-Warteschlange reserviert: Eine Station kann die anderen zwingen, ihr einen Rahmenschlitz zur reservieren indem sie in der Gegenrichtung ein Reservierungsbit setzt.

#### 802.2 LLC für 802.x

802.x sieht keine Bestätigungen oder zuverlässige Verbindungen vor. 802.2 definiert daher ein einheitliches LLC (Logical Link Control) Protokoll, das einen einheitlichen LLC-Rahmen für alle 802.x-Netzwerke vorsieht. Dies ist eine Teilschicht oberhalb der MAC-Schicht. LLC bietet bestätigten und unbestätigten Datagrammdienst und einen verbindungsorientierten Dienst.

#### FDDI

Fiber Distributed Data Interface - ist im Prinzip ein Token Ring auf Glasfaserbasis mit einer Geschwindigkeit von 100 MBit/s. Es benutzt allerdings eine Prioritätsregelung ähnlich von Token Bus: Solange das Token voraus ist, dürfen alle senden, ansonsten nur die hohen Prioritäten. FDDI braucht eine lange Präambel zur Synchronisation, da es nicht Manchester-codiert.

#### HIPPI

High Performance Parallel Interface - Eine Supercomputer-Verbindungschnittstelle mit 800 MBit/s. Das ganze funktionierte schon in den 80er Jahren mit einem Kabel mit 50 verdrillten Kabelpaaren, davon eines in jede Richtung. Will man die doppelte Bandbreite, nimmt man vier Kabel. Das ganze funktioniert immerhin auf 25m und wurde so eine Art Industriestandard bei Supercomputern.

#### Fiber Channel

Der designierte Nachfolger von HIPPI auf Glasfaserbasis. Kommt komplett mit einem Kann-Alles-Gewinnt-Immer-Protokoll und bis zu 800 Mbit/s.

# 3 Vermittlungsschicht

Die Vermittlungsschicht (Network Layer) hat die Aufgabe, Pakete zwischen zwei Kommunikationspartnern zu übertragen, und zwar auch über zwischengeschaltete Router o.ä. Dies ist die erste Ende-zu-Ende Schicht. Auch wenn IP (die Internet-Vermittlungsschicht mit Paketen arbeitet, kann eine Vermittlungsschicht (wie bei ATM) auch zuverlässige und/oder verbindungsorientierte Dienste anbiete.

## Routing

Dies ist eine der Hauptaufgaben der Vermittlungsschicht: Eine Route vom Sender bis zum Empfänger zu finden. Allgemein kann man das Netz als Graphen auffassen, bei dem die Knoten (=Router) über die Kanten Informationen austauschen. Jede Kante hat ein Gewicht, das die Entfernung zwischen den Routern sein kann oder die Antwortzeit oder sonst etwas sinnvolles. Kennt man den gesamten Graphen, ist es möglich eine optimale Route von einem Knoten zu berechnen (z.B. mit Dykstra's Algorithmus). Allerdings weiß nicht immer jeder Router über alles bescheid, und die

Toplogie des Netzes ändert sich oft. (Es ist auch möglich, statt einer einzelnen Route die Leistung des ganzen Netzes zu optimieren, wenn man den üblichen Verkehrsfluß kenn: **Flußbasiertes Routing**)

Also muß man ein Verfahren finden, mit dem jeder Router erfährt, was zu tun ist wenn ein Paket empfangen wird.

#### Flooding

Dieser Mechanismus hat den Vorteil, daß er immer funktioniert, ist aber nicht sehr effizient: Wenn ein Router ein Paket erhält, wird es auf allen Leitungen wieder ausgegeben, außer der, auf der es gekommen ist. Die Pakete müssen mindestens eine vorgegebene Lebensdauer haben (z.B. maximalen Hop-Count), damit keine endlos kreisenden Pakete entstehen.

#### Backward Learning

Flooding leidet darunter, daß es einen Haufen unnötiger Pakete erzeugt. Beim Backward Learning (z.B. bei Bridges eingesetzt), verwendet der Router Flooding, falls er den Weg zu einem Ziel nicht kennt. Außerdem wird die Ursprungsadresse des Paketes untersucht: Die Leitung, auf der es ankam ist offensichtlich der richtige Weg zum entsprechenden Subnetz. Auf diese Weise baut sich der Router (oder die Bridge) eine Tabelle auf, die Einträge haben nur eine gewisse Lebensdauer damit auch auf Änderungen der Topologie reagiert werden kann.

Backward Learning funktioniert nur, wenn es keine Schleifen in der Topologie gibt.

#### Hot Potato

Ein besonders lustiges Verfahren: Wir versuchen das Paket möglichst schnell wieder los zu werden und stellen es in den Ausgang mit der kürzesten Warteschlange.

#### Statisches Routing

Die Routingtabellen werden an den Routern von Hand eingestellt. Funktioniert prima, solange die Hierarchie sich nicht ändert.

#### Zentralisiertes- und Delta Routing

Man kann eine zentrales Routing Control Center einrichten, daß alle Routingentscheidungen trifft. Es muß einen Gesamtüberblick über das Netz haben (etwa indem es Statusmeldungen von allen Stationen einsammelt), berechnet für jeden Router die Routingtabelle und schickt sie ihm. Das Routing Control Center darf nie ausfallen und muß ausreichend leistungsfähig sein.

Beim Delta Routing bestimmt das Kontrollzentrum äquivalente Routen in der Routingtabelle (d.h. Routen, die sich nur um maximal ein Delta unterscheiden). Der Router darf dann selbst entscheiden, welchen der äquivalenten Ausgänge er benutzen will.

#### Distance Vektor Routing

Ein dynamisches Verfahren, bei dem jeder Router nur über seine Nachbarschaft bescheid weiß. Jeder Router weiß, zu welchen Kosten er einen anderen Router (am Anfang seine Nachbarn) erreichen kann. Diese Information teil er seinen Nachbarn mit. Merkt jetzt z.B. Router A, daß sein Nachbar B den Router C mit 1 Hops

erreichen kann, weiß A daß er C mit 2 Hops erreichen kann (indem er das Paket an B sendet). Diese Information teilt er wieder seinen Nachbarn mit.

Fällt allerdings C aus, so kann B ihn nicht mehr direkt erreichen. B weiß aber, das A behauptet C in 2 Hops erreichen zu können. Also glaubt B, es könne C über A in 3 Hops erreichen. Wenn A dies hört, glaubt es C über B in 4 Hops erreichen zu können, und so weiter. (**Count-To-Infinity-Problem**). Es gibt einige Optimierungen, die dieses Problem mildern sollen (z.B. Split Horizon: Der Router schickt die Entfernungsinformationen nicht auf die Leitung, mit der er den entsprechenden Partner erreicht), diese Hacks funktionieren aber alle nicht zuverlässig. Das Border Gateway Protokoll vermeidet jedoch das Problem, indem komplette Pfade übertragen werden.

#### Destination Sequence Distance Vektor Routing

Eine Variante des Distance Vektor Routing, die für mobile ad-hoc-Netze gedacht ist. Das Problem dort ist, das die Topologie sich schnell ändert, und sich das Routing schnell anpassen muß. Insbesondere count-to-infinity ist hier absolut tödlich.

Diese Variante verwendet zusätzlich eine Sequenznummer für jede gesendete Routinginformation (um Schleifenbildung zu vermeiden), und merkt sich deren Alter. Änderungen werden erst nach einer kurzen Bedenkzeit verwendet, um kurzfristige Schwankungen auszugleichen.

#### Dynamic Source Routing

Ein weiteres Verfahren für mobile ad-hoc-Netze. Das eigentliche Routing entspricht dem bei den Bridges erklärten Source Routing: Jeder Sender schickt den kompletten Pfad bis zum Ziel in seiner Nachricht. Allerdings wird hier nicht die komplette Netztopologie ausgetauscht, sondern ein Knoten versucht nur einen Pfad zu finden, wenn er etwas zu senden hat.

Kennt der Sender den benötigten Pfad noch nicht, so wird die Nachricht geflutet, und jeder Router hängt seine Adresse an das Paket an. Der Empfänger erhält dann Pakete mit kompletten Pfadangaben, kann den günstigsten aussuchen und mit der Bestätigung an der Sender zurückschicken.

Dieses Verfahren hat den Vorteil, nicht ständig Routinginformationen auszutauschen. Es wird aber problematisch in hochdynamischen Umgebungen in denen sich die Pfade immer ändern, und wenn die Verbindungen zwischen zwei Knoten nicht symmetrisch sind.

#### Link State Routing

Hier sammelt jeder Router Informationen über die Verbindungen zu seinen Nachbarn und stellt diese in einem Link-State-Paket zusammen. Diese Pakete werden dann (z.B. per Flooding) an alle anderen Router verteilt. Die Link-State-Pakete haben (zur Vermeidung von Zweideutigkeiten) eine Sequenznummer (um die richtige Reihenfolge festzustellen) und eine maximale Lebensdauer (die heruntergezählt wird, damit z.B. nicht ewig Pakete mit niedriger Sequenznummer nach einem Neustart verworfen werden).

Hat ein Router die Link-State-Pakete von allen anderen Routern erhalten, kann (z.B. mit Dijkstras Algorithmus) die optimalen Routen berechnen.

#### Hierachisches Routing

Um die Routingentscheidungen nich endlos komplex werden zu lassen, kann man Netze als Hierachien auffassen. Innerhalb eines Teilnetzes wird ein beliebiges Routingverfahren verwendet, alle Pakete die das Teilnetz verlassen gehen an einen be-

sonders designierten Router. Dieser ist Teil der nächsthöheren Hierarchiestufe, in der das Routing ohne Betrachtung der tieferen Stufen erfolgt.

#### Broadcast/Multicast

Sollen mehrere Empfänger angesprochen werden, kann man natürlich einfach das Netz fluten. Eine Alternative ist, daß jeder Router das Broadcast/Multicast-Paket auf allen Leitungen verschickt, die eine korrekte Route zum einem der Ziele darstellen (Multidestination-Routing). Eine weitere Methode ist, einen *Spanning Tree* über alle Knoten aufzubauen, und die Pakete an dessen Kanten weiterzuleiten - dazu muß aber jeder Knoten die komplette Netztopologie kennen. Beim Multicast existiert zusätzlich ein Spanning Tree für jedes Mitlied der Gruppe (jeder ist beim Senden die Quelle des Spanning Tree) was zu Speicherproblemen führen kann. (Alternativ kann auch nur ein Baum existieren, und die Daten werden zuerst an die Wurzel gesendet. (Beim Multicast existiert zusätzlich das Problem, den Baum auf alle Mitglieder der Gruppe zu beschneiden.)

Ein Verfahren, um auch ohne komplette Kenntnis der Topologie einen Baum zu erhalten ist das **Reverse Path Forwarding**. Hier nimmt ein Router an, daß ein Multicast-Paket das auf der üblichen Route von seinem Sender ankommt, das „richtige " ist. Alle Pakete, die nicht über die Leitung kommen, über die normalerweise Pakete an den Sender gehen, werden als Duplikate verworfen, das „richtige „ Paket wird weitergeflutet. Mit diesem Ansatz wird das Verhalten eines spannenden Baumes nachgebildet. Beim Multicast-Routing kann ein Router, der keine Hosts der entsprechenden Gruppe besitzt verlangen, daß er keine Pakete mehr erhält (PRUNE). Ein Router, der von allen seinen Nachbarn PRUNE-Meldungen bekommt, kann sich ebenfalls ausklinken (dieser Mechanismus entspricht dem Beschneiden des Baumes).

### Überlastungsüberwachung

Die Steuerung von Überlastungen unterscheidet sich etwas von der Flußkontrolle: Hier geht es darum, daß das Netz als solches die Last nicht mehr verkraften kann (z.B. ein Router ist überlastet) - es geht also nicht um die beiden Endsystem, sondern um das, was dazwischenliegt. Auf der Vermittlungsschicht geht es hauptsächlich darum, daß Router die Datenmenge, die sie erhalten, nicht mehr richtig verarbeiten können.

Bei der Umschiffung von Überlastungen gibt es zwei Möglichkeiten: Entweder das System so zu konstruieren, daß keine Überlastungen entstehen (offene Schleife) oder zu versuchen, bei Überlastungen Gegenmaßnahmen zu treffen (geschlossene Schleife). Manche Methoden funktionieren besser (oder nur) mit virtuellen Verbindungen, manche eignen sich auch Datagrammdienste.

#### Traffic Shaping

Hier sollen die Station gezwungen werden, ihre Daten in einer einergermaßen vorhersehbaren Rate zu senden, was die Vermeidung von Überlastungen ungemein vereinfacht. ATM wendet z.B. Traffic-Shaping an um eine maximale Auslastung der virtuellen Verbindungen zu garantieren. Traffic Shaping geht davon aus, daß die Transportschicht einfach Daten sendet, eventuell mit plötzlichen Spitzen, oder viel zu viele.

Um das wieder in den Griff zu bekommen, die Datenrate zu beschränken und etwas gleichmäßiger zu machen, kann man den **Leaky-Bucket**-Algorithmus verwenden. Er modelliert einen Eimer, in den (schwallweise?) Wasser geschüttet wird,

das durch ein Loch im Boden wieder heraustropft. Läuft der Eimer über, geht Wasser verloren. Praktisch läß also der Router immer nur eine gewisse Anzahl Pakete pro Zeittakt passieren und sorgt damit für eine gleichmäßige Datenrate. Kommen mehr Pakete an, werden sie gepuffert, läuft der Puffer über, werden sie verworfen.

Will man kurzzeitige Spitzen zulassen, kann man den **Token-Bucket** Algorithmus verwenden. Hier nimmt der Eimer Tokens auf, die in einer gleichmäßigen Rate erzeugt werden. Jedes Paket, das verschickt wird, verbraucht ein Token. Wenn eine Sendung beginnt, kann sie solange mit voller Geschwindigkeit senden, bis die Token im Eimer verbraucht sind. Danach kann nur noch mit der Geschwindigkeit gesendet werden, wie neue Token erzeugt werden. Beide Mechanismen lassen sich statt mit Paketen auch byteweise durchführen.

Damit klar ist wie viele Resourcen gebraucht werden, können die Partner (die Stationen und das Netz) eine Flußspezifikation vereinbaren. Hier werden die maximalen Datenraten, zulässige Verzögerungen, die Größe der Buckets und ähnliches festgelegt.

#### Choke-Pakete

Wenn ein Netz überlastet ist, hilft es irgenwann nur noch, wenn die Sender ihre Last zurückfahren. Eine Möglichkeit, die Sender von der Überlast zu informieren, sind sogenannte *Choke-Pakete*. Dies sind Warnungen die (eventuell huckepack) von der überlasteten Stelle an den Sender geschickt werden, damit dieser seine Sendung reduziert. Erhält der Sender weitere Choke-Pakete, drosselt er die Leistung weiter, kommen keine mehr legt er langsam wieder zu. Choke-Pakete habe allerdings das Problem bei einer Überlassituation noch weitere Pakete zu erzeugen, außerdem kann die Reaktionszeit auf langen Leitungen recht lang sein. Letzteres Problem kann behoben werden, wenn jeder Router, der ein Choke-Paket erhält die Leistung sofort drosselt und die restlichen Pakete puffert (**Hop-by-Hop-Choke**).

#### Weighted Fair Queuing

Bei den Choke-Funktionen (nicht nur da) kann es passieren, daß sich ein Sender nicht an die Vereinbarungen hält und dadurch unfairerweise bevorzugt wird (weil er als einziger seine Last nicht drosselt). Um ihn daran zu hindern kann jeder Router für jeden Eingang eine eigene Warteschlange haben, diese werden dann reihum bedient. Ein „unfairer“ Router bekommt so nicht mehr Bandbreite als einer, der sich korrekt verhält. Soll eine Leitung bevorzugt werden, so darf sie mehr als ein Paket pro Zeiteinheit versenden. (Das ganze funktioniert wieder auch byteweise statt paketweise).

#### Load Shedding

Gibt es keine andere Möglichkeit, muß der Router irgendwann anfangen, Daten zu verwerfen. Damit sollte er auch noch einigermaßen frühzeitig anfangen, bevor die Situation völlig verfahren ist. Die Frage ist jetzt nur noch: Welche Daten können weg? Sind die Daten z.B. ein Multimedia-Strom, können eher alte Pakete verworfen werden (Milch). In anderen Fällen sind neue Pakete weniger wichtig (Wein). In anderen Fällen könnte der Router sinnvollere Entscheidungen treffen, wenn er den Inhalt der Daten kennt. (Wird z.B. ein Teil eines IP-Paketes nicht übertragen, braucht man den Rest auch nicht mehr).

#### Jitter-Kontrolle

Manchmal benötigt man möglichst gleichbleibende Verzögerungszeiten. In diesem Fall könnten Route Pakete, die zu schnell sind verzögern, und zu langsame Pakete

bevorzugt zustellen.

#### Resource Reservation Protocol

Dies ist ein Protokoll zur Reservierung von Resourcen beim Multicast. Es wird (irgendwie) ein spannender Baum aufgebaut. Entlang dieses Baumes können jetzt Resourcen für Multicast-Verbindungen reserviert werden, dabei wird jede Teilstreke logischerweise maximal für eine Übertragung reserviert.

### Tunneling

Manchmal muss ein Paket über eine Strecker übertragen werden, die das entsprechende Protokoll nicht unterstützt. In diesem Fall können die Pakete in Pakete des fremden Protokolls „eingepackt“ werden, und am Ende der Strecke (bzw. des Teilnetzes) werden sie wieder ausgepackt. Am Anfang und Ende der Tunnelingstrecke werden spezielle Router benötigt, die das Ein- und Auspacken erledigen.

### Fragmentierung

Manchmal unterstützt eine Teilstrecke nur eine maximale Paketgröße, die kleiner ist, als das Paket. In diesem Fall muß das Paket von einem Router in kleinere Teile zerlegt (fragmentiert) werden. Das Zusammensetzen der Fragmente erledigt entweder ein anderer Router (transparent) oder das Endsystem selber. Ein Fragment muß irgendwie die Information enthalten, zu welchem Paket es gehört, und an welche Stelle des Paketes. Das kann mit laufenden Nummern, oder einem Byteoffset zum Paketanfang geschehen. Problematisch wird es, wenn Pakete mehrfach fragmentiert werden, oder wenn ein Fragment verloren geht und wiederholt werden soll.

### IP - Internet Protocol

IP ist das Vermittlungsschicht-Protokoll im Internet. Es bietet einen verbindungslosen, unzuverlässigen, paketorientierten Dienst an. Die Steuerprotokolle ICMP und (R)ARP unterstützen IP bei seiner Aufgabe.

#### IP-Paketkopf

Der IP Paketkopf enthält die Version des verwendeten IP-Protokolls (4), die Länge des Paketkopfes, eine (nicht verwendete) Servicetypebeschreibung, die Gesamtlänge des Paketes, eine Eindeutige ID für die Fragmentierung, einen Fragmentoffsett für die Fragmentierung, die Bits DF (don't fragment) und MF (more fragments), die Restlebensdauer des Paketes, einen Code für das verwendete Protokoll der Transportschicht, eine Prüfsummen für den Paketkopf und jeweils eine 32 bit lange Ziel- und Quelleadresse die weltweit eindeutig ist. Außerdem stehen 40 Byte für optionale Header zur Verfügung. Es sind Header-Optionen wie Source Routing oder Route Logging oder Security definiert, diese werden jedoch von den existierenden Routern nicht unterstützt.

#### IP-Adressierungsschema

Der IP-Adressraum ist in mehrere Nezklassen eingeteilt. Ein IP-Adresse besteht aus einem Klassencode, der Netzadresse und der Hostadresse. Die Hostadresse kann eventuell noch in Subnetz- und Hostadresse aufgeteilt werden. Die ursprüngliche Form sah 5 Klassen vor: A (Code: 0), 16 Netze mit 16 Millionen Hosts, B (01), 16k Netze mit 64k Hosts, C (011), 2 Millionen Netze mit 256 Hosts, D (0111) fr Multicast-Adressen und E (01111) für spätere Verwendung. Es sind also zwei bzw.

drei Schichten für das Routing vorgesehen: Ein Paket wird zuerst zu seinem Netz geleitet und dort (vielleicht über Subnetze) intern weiterverteilt.

### CIDR

Leider waren nach dem ursprünglichen Schema die Adressen irgendwann viel zu knapp. Viele Firmen reservierten ein Class B Netz, obwohl sie gar nicht so viele Adressen brauchten, denn die 256 Host der Klasse C waren zu wenig. Die Einrichtung von Millionen Teilnetzen (z.B. mehr Klasse C) würde außerdem zu einer Explosion der Routingtabellen führen. Um das Problem zu entschärfen, wurde die Vergabe von Class-C-Subnetzen geändert: Statt als Subnetze werden die Adressen jetzt in variabel großen Blöcken vergeben. Um das Routing zu vereinfachen, wird der Adressraum in Zonen unterteilt (Nord- und Südamerika, Europa und Asien) die dann für das Routing als „Subnetze “ gelten. Damit nicht jedes C-Netz in einem Block einzeln in die Routingtabelle muß, wird zusätzlich zu einem Block eine Netzmaske vergeben. Die Netzmasken werden dann mit den eingehenden Paketen AND-verknüpft um den richtigen Eintrag zu finden (d.h. es wird geprüft, ob der Anfang der Adresse mit dem Anfang des entsprechenden Routing-Eintrages übereinstimmt).

### IP-Routing

Das Internet besteht aus einer großen Zahl von *autonomen Systemen* (also z.B. Firmennezen). Der Betreiber eines solchen Systems kann das Routingverfahren dort selbst bestimmen (internes Routing). Das Routing zwischen den einzelnen autonomen Systemen wird als externes Routing bezeichnet.

### Open Shortest Path First

OSPF ist das empfohlene Routingprotokoll für internes Routing, ein Link-State-Protokoll. (Früher wurde das Distance-Vektor-Protokoll RIP verwendet, aber es hatte recht viele Schwächen).

OSPF sieht ein System als Ansammlung von *Bereichen* an. Mehrere Bereiche überlappen sich nicht, es müssen aber nicht alle Hosts zu einem Bereich gehören. Alle Router sind auf jeden Fall an den Backbone-Bereich angeschlossen. Ein Router in einem Bereich berechnet immer nur die Routen für diesen Bereich, ein Paket für einen anderen Bereich wird an das Backbone gesandt, dort weiterverteilt und schließlich von den Routern im Zielbereich ausgelierfert. Ein Router muß also nicht immer die komplette Topologie des Netzes kennen.

OSPF verwendet ein einfaches Protokoll zum Datenaustausch zwischen den Routern das auf IP aufsetzt. Die Nachrichten werden bestätigt. Ein Router kann HELLO-Pakete aussenden, um seine Nachbarn zu finden (per Multicast). Ein LINK STATE-Paket wird von einem Router regelmäßig oder bei Bedarf verschickt und zwar nicht an alle Nachbarn, sondern nur an sogenannte *angrenzende* Router. Es gibt einen designierten Router, der an alle anderen angrenzt, ein Backup für ihn steht jederzeit bereit. Router können außerdem den Stand seiner Linkdatenbank aussenden, oder von einem anderen Link-State-Informationen einfordern.

Anhand ihrer Informationen berechnen die Router dann die jeweils kürzesten Pfade zu den einzelnen Zielen.

### Border Gateway Protocol

Dies ist das Protokoll für das externe Routing im Internet. Beim externen Routing müssen gewisse Regeln beachtet werden, z.B. wollen eventuell gewisse Netze bestimmten Verkehr nicht übertragen. Das BGP ist ein deutlich verbessertes Distance-

Vektor-Verfahren, daß mit solchen Regeln umgehen kann. Ein Router speichert nie nur die Entfernung und den nächsten Router zu einem Ziel, sondern den kompletten Pfad. Er erhält von den Nachbarn verschiedene Pfade zu den Zielen, und kann davon den besten aussuchen; Pfade die durch ihn selber führen (vermeidet Count-To-Infinity) und Pfade die nicht zu den Regeln passen, können dabei verworfen werden. Hat der Router einen Pfad ausgewählt, so überträgt er ihn an alle anderen als „seinen “ Pfad.

### ICMP

Das Internet Control Message Protocol dient zu Wartungszwecken im Internet. Hier liegt z.B. der ECHO-Mechanismus (Prüfen von Verbindungen) und CHOKE-Pakete. Jede ICMP-Nachricht wird in ein IP-Paket verpackt.

### ARP/RARP

Das Adress Resolution Protocol dient der Zuordnung von IP-Adressen zu Schicht-2-Hardwareadressen. Es wird ein Broadcast gesendet, mit der Aufforderung daß sich der Besitzer der IP-Adresse x bitte melden möge. Alle, die diesen Broadcast hören, können ihren ARP-Cache aktualisieren, und wissen jetzt unter welcher Hardwareadresse sie diese IP erreichen.

Beim RARP kann eine Station eine Anfrage rundsenden, welche IP-Adresse zu ihrer Hardwareadresse gehört. Diese Anfrage wird von einem designierten RARP-Server beantwortet, der die Zuordnungen gespeichert hat.

### Multicast-Routing

IP bietet Multicast-Dienst für Gruppen an. Multicast-Adressen liegen im Bereich 224.x.y.z, manche dieser Adressen haben festgelegte Bedeutungen (z.B. 224.0.0.1 für das lokale Netzwer). Für das Multicasting werden spezielle Multicast-Router benötigt (MBone), die regelmäßig die angeschlossenen Hosts nach ihrer Gruppenzugehörigkeit fragen. Die Multicast-Router tauschen die Daten dann entlang eines spannenden Baumes aus.

### IPv6

Die nächste Generation des IP-Protokolls soll vor allen Dingen den Mangel an Adressen beheben, außerdem wurde der Header vereinfacht, um die Verarbeitung der Pakete effizienter gestalten zu können.

Der **IPv6-Header** besteht aus dem üblichen Feld mit der Versionsnummer (6), einem Flow-Control-Feld das zusammen mit speziellen Routern Datenflüsse (d.h. Pseudo-Verbindungen) ermöglichen soll, einer Angabe über die Länge der Nutzlast (maximal 64k), einem Feld mit dem Typ des nächsten (optinalen) Headers (falls kein Header mehr folgt, steht hier der Code des Schicht-4-Protokolls), und ein Feld mit der restlichen Lebensdauer in Hops. Weggefallen sind die Checksumme (mußte bei IPv4 in jedem Router neu berechnet werden) und die Mechanismen zur Fragmentierung: Wenn in IPv6 ein Router ein Paket nicht annehmen kann, lehnt er es ab und informiert den Sender über seine MTU (maximum transfer unit). Der Sender ist dafür verantwortlich, daß die maximal zulässige Paketgröße nicht überschritten wird. Die Quell- und Ziel-**Adressen** schließlich sind 128 bit lang und wieder hierarchisch gegliedert (nach Regionen, dann nach Providern, etc.). Die Adressen des bisherigen IPv4 sind auch im IPv6-Adressraum enthalten.

IPv6 unterstütztl eine Reihe von **optionalen Headern**, z.B. für das Versenden von Jumbogrammen (Paketen ¿ 64k), Routingfunktionen, Sicherheit, Verschlüsselung und ähnliches.

# 4 Transportschicht

Die Transportschicht bietet zwei entfernten Kommunikationspartnen Dienste für die Ende-zu-Ende-Kommunikation an, z.B. zuverlässige virtuelle Verbindungen. Dies geschieht unabhängig von den Fähigkeiten der Vermittlungsschicht.

In gewisser Weise hat die Transportschicht ähnliche Aufgaben wie die Sicherungsschicht: Sie muß die Unzulänglichkeiten tieferer Schichten ausgleichen, um den Anwendungen einen zuverlässigen Dienst zu bieten. Deshalb kommen auch Mechanismen aus der Sicherungsschicht wie die Flusskontrolle hier wieder zum Einsatz. Andererseits unterscheidet sich die Situation hier grundlegend von einer Direktverbindung: Pakete können dupliziert und verzögert werden, und nicht nur der Kommunikationspartner sondern auch das Netz selbst kann überlastet sein. Außerdem hat eine Transportschicht-Instanz möglicherweise viele Verbindungen mit variabler Paketgröße gleichzeitig zu versorgen.

## Flußkontrolle

Grundsätzlich kommt hier, wie in der Sicherungsschicht, ein Schiebefenster zum Einsatz. Man kann hier aber nicht davon ausgehen, daß unbestätigte Pakete auch wirklich verschwunden sind, sie könnte ja auch nur verzögert sein und zu einem ungünstigen Zeitpunkt wieder auftauchen. Duplizierte Pakete (ein Duplikat tauch zum falschen Zeitpunkt noch mal auf) verschlimmern das Problem noch. Um dieses Problem zu lösen, benötigt man einen Mechanismus der verzögerte Pakete (und ihre Bestätigungen) irgendwann verwirft – es gibt also eine Zeitspanne $T$, nach der ein Paket und alle Bestätigungen auf jeden Fall verschwunden sind (Das läßt sich mit Timeouts erreichen). Andererseits ist es notwendig, daß ungültige Pakete erkannt und verworfen werden.

Eine Methode wäre die Verwendung von eindeutigen Verbindungs-IDs damit die Partner herausfinden können, ob ein Paket zu einer gültigen Verbindung gehört. Allerdings müssten in diesem Fall die IDs aller alten Verbindungen gespeichert werden (und sie wären verloren, wenn der Host abstürzt...). Eine andere Möglichkeit ist die Verwendung von Zeitangaben: Dazu müssen die beiden Rechner synchronisierte Uhren haben (die auch nach einem Absturz weiterlaufen). Als erste Sequenznummer einer Verbindung wird dann die aktuelle Uhrzeit verwendet (oder ein Teil davon), durch diese Methode sind die Sequenznummern eindeutig festgelegt, und veraltete Aufbauwünsche würden abgelehnt. Der Folgenraum sollte so groß sein, daß wandernde Pakete auf jeden Fall verschwunden sind, bevor sich die Folgennummern wiederholen (ein Problem bei Gigabitnetzen!).

Ein andere Problem kann noch auftauchen, wenn ein Host abgestürzt war, wieder aufsetzt und vor dem Absturz Folgennummern verwendet wurden, die nun gültig sind (d.h. Folgennummern, die um bis zu $T$ größer sind, als die aktuelle Uhrzeit) – sollten diese Pakete noch auftauchen, werden sie als gültig akzeptiert. Deshalb müssen Pakete, die für eine neu aufgesetzte Verbindung noch gültig wären, verboten werden. Das betrifft alle Sequenznummern in einer verbotenen Zone der Breie $T$ oberhalb der aktuellen Zeit (Pakete mit noch größerer Uhrzeit wären verworfen, bis die Uhrzeit diesen Stand erreicht hat). Das bedeutet allerdings auch, daß höchstens ein Paket pro Zeittakt gesendet werden darf (die Uhr müß also sehr schnell laufen), und daß die verbotene Zone auch erreicht wird, wenn sich der Sender sich ihr durch langsames Senden „vom anderen Ende “ aus nähert (die Folgenummern wiederholen sich ja).

## Pufferverwaltung

Die Transportschicht muß eventuell einen Haufen Verbindungen verwalten und unterstützt möglicherweise Transportschicht-Pakete (TPDUs) mit unterschiedlicher Größe. Es ist also oft nicht möglich und sinnvoll, einen Puffer mit festgelegten Größen einzurichten. Stattdessen sollte die Transportschicht den Puffer dynamisch verwalten und natürlich sinnvoll organisieren (verkettete Puffer, Kreispuffer, wasauchimmer...). Damit das funktioniert, wird dann ein **Schiebefenster mit dynamischer Größe** verwendet: Der Empfänger teilt in der Bestätigung jeweils mit, wie viele Pakete noch gesendet werden dürfen. Wartet der Sender, und es werden wieder Puffer frei, wird das dem Sender auch mitgeteilt. Diese Mitteilung sollte nach einer Zeit wiederholt werden, falls sie verlorenging und der Sender immer noch wartet.

## Überlastungssteuerung

Erhält Schicht 3 von Schicht 4 konstant mehr Daten als sie übertragen kann wird sie irgendwann beginnen, Pakete zu verwerfen. In diesem Fall kann nur eine Drosselung der Transportschicht-Senderate wirklich eine Abhilfe schaffen. Deshalb muß der Sender sein Sendefenster nicht nur an den Empfängern anpassen, sondern auch an die Gegebenheiten im Netz. Welche Mechanismen dort verwendet werden, wird am Beispiel von TCP erläutert.

## Verbindungsauf- und Abbau

Damit beim Verbindungsaufbau nichts schiefgeht, wird ein Drei-Wege-Handshake verwendet: Verbindungsanfrage, Bestätigung, Rückbestätigung (mit erstem Datenpaket). Gäbe es die Rückbestätigung nicht, könnte eventuell der zweite Partner zu senden beginnen obwohl die Bestätigung verlorgen gegangen ist. Auf der Transportschicht werden beim Verbindungsaufbau auch die intialen Sequenznummern ausgehandelt.

Beim Verbindungsabbau kann wieder ein Dreiwege-Protokoll verwendet werden, obwohl hier nicht alle Fehlerfälle auszuschließen sind (byzantinisches Problem).

## TCP

TCP ist das Protokoll der Transportschicht im Internet. Es stellt zuverlässige Punkt-zu-Punkt-Verbindungen zwischen zwei Applikationen zur Verfügung. Diese Verbindungen sind immer Vollduplex-Verbindungen, und übertragen Byteströme. TCP überträgt die Daten in sogenannten Segmenten, wobei die Aufteilung des Bytestromes in Segmente von TCP völlig willkürlich vorgenommen werden kann.

### Berkeley Sockets

Ein Socket ist ein Zugangspunkt zur Transportschicht. Mann kann ein Socket erzeugen, ihm eine Adresse zuweisen, Verbindungen auf ihm entgenenehmen oder eine ausgehende Verbindung aufbauen. Selbstverständlich können durch das Socket dann auch Daten gesendet und empfangen werden.

### TCP-Adressierungsschema

Eine TCP-Verbindung ist eindeutig gekennzeichnet durch die Adressen der beiden Endpunkte. Eine TCP-Adresse besteht aus der IP-Adresse und dem **Port**. Die Ports unter 256 sind wohlbekannt (**well known**) und Standarddiensten zugeordnet (z.B. Mail, Telnet, etc.) Höhere Portnummern (über 1024) können von Applikationen frei verwendet werden, z.B. zum Aufbau von Verbindungen.

### TCP-Semgmentheader

Der TCP-Segmentheader enthält zunächst einmal die Ziel- und Quellportnummer der Verbindung. Außerdem ist eine 32-Bit Sequenznummer und ein Bestätigungsnummer enthalten (die Bestätigungsnummer entspricht der nächsten erwarteten Sequenznummer). Ein weiteres Feld beschreibt das Empfangsfenster des Senders (d.h. wie viele Byte ab dem bestätigten noch gesendet werden dürfen). Ein Feld gibt die Länge des TCP-Headers an (es sind optionale Felder möglich) und eine Prüfsumme sichert den TCP-Header, die Daten und den sog. Pseudo-Header der die IP-Adressen von Sender und Empfänger enthält. Das ACK-Bit gibt an, daß die Bestätigung in diesem Segmentheader gültig ist. Das SYN-Bit dient dem Verbindungsaufbau. Das PSH-(Push-)Bit gibt an daß Push-Daten gesendet werden (die sofort an die Anwendung ausgeliefert werden sollen). Das URG-(Urgent-)Bit zeigt dringende Daten an, die sofort ausgeliefert werden (eine Art Interrupt-Signalisierung) in diesem Fall verweist ein Urgent-Zeiger im Header auf den Beginn der dringenden Daten. Das RST-(Reset-)Bit dient dem Zurücksetzen der Verbindung nach einer Störung, das FIN-Bit zum Verbindungsabbau.

Nach dem eigentlichen Segmentheader können noch weitere Optionen folgen. Tatsächlich verwendete Optionen sind **Window Size**, das eine andere Skalierung der Fenstergröße vorgibt um ein Empfangsfenster von ¿ 64k zu ermöglichen und die implementierung der selektiven Wiederholung.

### Verbindungsauf- und abbau

Eine TCP-Verbindung wird initiert, indem ein Prozeß (A) eine Verbindungsanfrage an einen warten Prozeß (B) (der ein Socket überwacht) stellt. Dazu sendet Prozeß A ein SYN-Segment mit seiner initialen Sequenznummer. B antwortet mit einem SYN-Paket, daß die Anfrage bestätigt und Bs initiale Segmentnummer enthält (SYN ACK). Daraufhin bestätigt A wiederum das eingegangene Paket (ACK), die Verbindung ist aufgebaut und beide Partner können Daten senden.

Die beiden Richtungen einer TCP-Verbindung werden getrennt abgebaut: Jede Seite sendet ein FIN-Segment, wenn sie ihre Übertragung beendet hat, und wartet auf die bestätigung. Wenn eine Station ein FIN-Segment erhält, kann sie mit der Bestätigung auch gleich den Abbauwunsch der anderen Richtung übermitteln (FIN ACK).

### Nagles Algorithmus und Silly Window Syndrome

Das Lesen und Schreiben von einzelnen Bites kann in TCP zu Problemen führen. Senden Stationen immer nur einzelne Bytes, so wird für jedes Byte ein eigenes, 21 Bit großes Segment erzeugt und unnötig Bandbreite verschwendet. Nagles Algorithmus sieht daher vor daß, wenn ein einzelnes Byte übertragen wird, die Übertragung angehalten wird bis die Bestätigung eintrifft (in der Hoffnung, daß sich noch ein paar Bytes mehr ansammeln.

Umgekehrt kann es vorkommen, daß ein Prozeß immer nur einzelne Bytes aus dem Empfangpuffer liest. In diesem Fall würde der Empfänger jedes Mal ein neues Empfangsfenster von einem Byte signalisieren, und der Empfänger jeweils ein einzelnes Byte übertragen. Dies ist das Silly-Window-Syndrom und erzeugt ebenfalls sinnlosen Overhead. Zur Abhilfe muß der Empfänger solange warten, bis er wieder ein Fenster mit einer sinnvollen Größe signalisieren kann.

### Überlastungsüberwachung

TCP verwaltet für jedes Segment einen Timeout, nachdem es erneut übertragen wird. Es wird davon ausgegangen, daß eine ausbleibende Bestätigung auf eine Über-

lastung im Netz zurückgeht (eine Annahme, die für Kabel- bzw. insbesonder für Glasfaserübertragung gemacht werden kann, bei Funkübertragung gibt es hier Probleme). TCP verwaltet zusätzlich zum Empfangsfenster ein sogenanntes Überlastungsfenster (Congestion Window), das ebenfalls angibt wie viele Bytes der Sender übertragen kann.

Das Überlastungsfenster wird in TCP durch den **Slow Start** Algorithmus angepasst: Das Fenster hat zuerst einen Minimalwert (z.B. eine maximale Segmentgröße) und wenn alle Daten im Segment erfolgreich übertragen wurden, wird das Fenster verdoppelt. Dies geht solange, bis entweder Daten verlorengehen, oder die Größe des Empfangsfenster erreicht wurde. Zusätzlich wird ein Schwellenwert geführt, der bei Paketverlusten auf die Größe des halben Überlastungsfensters gesetzt wird. Bis der Schwellenwert erreicht wird, verdoppelt der Algorithmus das Überlastungsfenster, darüber wird linear weitervergrößert.

#### Timermanagement

Ein Problem im TCP ist die Verwaltung des Timeouts für verlorene Pakete – einfach weil nicht genau bekannt ist wann die Bestätigung eigentlich ankommt. TCP behilft sich, indem es versucht, zuerst einmal die durchschnittliche Rundreisezeit bis zum Ziel zu bestimmen:

$$R = \alpha R + (1 - \alpha)M$$

Dabei ist $R$ die geschätzte Rundreisezeit und $M$ die gemessene Zeit von Senden bis zur Bestätigung des letzten Paketes. $\alpha$ ist ein Glättungsfaktor. Zusätzlich wird versucht, annähernd die Standardabweichung der Bestätigungsankunft zu bestimmen:

$$D = \alpha D + (1 - \alpha)|RTT - M|$$

Der Glättungsfaktor $\alpha$ braucht nicht unbedingt der selbe zu sein wie oben. $D$ ist zwar nicht die Standardabweichung aber eine hinreichend sinnvolle Näherung. Der Timeout $T$ berechnet sich dann nach

$$T = R + 4D$$

also ein Paket wird nach der durchschnittlichen Rundreisezeit plus der vierfachen Standardabweichung als verloren angesehen. Das ist relativ konservativ: Etwas zu lange warten ist besser als ständig neu zu übertragen und Slow Start anzuwenden.

### TCP in mobilen Umgebungen

Beim mobilen Einsatz macht sich vor allem Slow Start unangenehm bemerkbar: Funkverbindungen verlieren recht häufig Pakete, und eine sinnvolle Strategie wäre eigentlich, diese sofort neu zu übertragen. Andererseits würde ohne Slow Start das Internet zusammenbrechen, der Mechanismus sollte also weiterhin in IP implementiert bleiben.

#### Fast Retransmit/Fast Recovery

Diese TCP-Option geht davon aus, daß keine Überlastung vorliegt solange noch *irgendwelche* Bestätigungen von Empfänger eingehen (selbst wenn es nicht die für das aktuelle Paket sind). In diesem Fall versucht TCP die nicht bestätigten Segmente sofort noch einmal zu übertragen (Fast Retransmit). Außerdem liegt kein Grund vor, Slow Start anzuwenden und die Datenrate wird beibehalten (Fast Recovery). Slow Start wird nur angewendet, wenn die Bestätigungen ganz ausbleiben.

Eine weitere Möglichkeit ist es, daß der mobile Knoten diesen Mechanismus erzwingt, indem er von sich aus Bestätigungsduplikate versendet.

### Indirect TCP

Bei dieser Variante dient der Zugangspunkt (bzw. Foreign Agent) des mobilen Host gleichzeitig als TCP-Proxy. Er bestätigt alle Segmente des Kommunikationspartners, puffert sie und liefert sie dann selbst über eine TCP-Verbindung an den Knoten aus. Am Knoten selbst und am Kommunikationspartner muß nichts geändert werden, und das Verfahren ist recht effizient. Alllerdings geht die Ende-Zu-Ende-Semantik von TCP verloren, kann der Knoten nicht erreicht werden, werden die Segmente nicht ausgeliefert *obwohl sie schon bestätigt sind*. Außerdem müssen beim Wechsel des Zugangspunktes alle gepufferten Daten zum neuen Zugangspunkt übertragen werden.

### Snooping TCP

Hier trennt der Zugangspunkt die Ende-zu-Ende-Verbindung nicht, puffert allerdings alle Pakete und versucht sie schnell zu wiederholen, Pakete die im Datenstrom des Mobilknotens fehlen werden direkt nachgefordert und ähnliches. Dies hat den Vorteil, daß die Verbindung wirklich bis zur Mobilstation geht, allerdings wird das Netz nicht so gut gegen die Fehler der Funkstrecke abgeschirmt.

### Mobile TCP

Bei Mobile TCP trennt der Zugangspunkt die TCP-Verbindung zwar auch, puffert aber keine Daten und sendet auch keine Bestätigungen. (Der Zugangspunkt geht davon aus, daß die Qualität der Funkstreck o.k. ist, also sollen sich die Partner selbst um Wiederholungen kümmern). Stattdessen untersucht der Zugangspunkt, ob noch Bestätigungen vom Mobilen Knoten kommen. Ist das nicht der Fall geht er davon aus, daß der mobile Host nicht erreichbar ist und setzt das Empfangsfenster auf null. Daher wird der Kommunikationspatner nicht versuchen, in dieser Zeit Daten zu senden.

# 5 Sitzungsschicht

Diese Schicht sollte es Benutzern ermöglichen, Sitzungen zwischen zwei Maschinen aufzubauen. Dies geschieht allerdings in der Praxis auf Anwendungsebene, und nicht im Protokollstapel.

# 6 Darstellungsschicht

Diese Schicht kennt die Repräsentation von Daten und repräsentiert sie auf standartiesierte Weise. Z.b. könnten Datenformate entsprechend konvertiert werden. Diese Schicht wird praktisch nie eingesetzt, Konvertierungen finden auf Anwendungsebene statt.

# 7 Verarbeitungsschicht

Diese Schicht enthält die Anwendungsspezifischen Protokolle.

# 8 Sonstige Systeme

## ISDN

ISDN stellt verschiedene Digitale Dienste für Telefonie und Datenübertragung zur Verfügung. ISDN definiert eine Reihe von Übertragungskanälen. Benutzt werden allgemein der B-Kanal, ein 64 kbps Daten-/Sprachkanal (Ende-zu-Ende) der eine „Datenpipeline“ auf Schicht 1 darstellt (für Sprache wird er als PCM-Kanal benutzt), und der D-Kanal für Zeichengabe, der 16 kbps (paketweise) gesichert über ein HDLC/LAPD-ähnliches Protokoll überträgt (Schicht 3).

ISDN definiert eine Reihe von Standardschnittstellen: V, für die Übertragung zwischen Vermitllungstelle und Vermittlungsanschluß (?), U zwischen Vermittlungsstelle und Netzwerkterminator (NT, entspr. NTBA), und T nach dem Basisanschluß. Entweder werden an die T-Schnittstelle ($S_0$-Bus) bis zu 8 Geräte gleich angeschlossen, oder es gibt eine Nebenstellenanlage, die dann eine S-Schnittstelle (auch $S_0$-Bus) zur Verfügung stellt. Über einen Terminaladapter (TA, entspr. a/d-Wandler) können nicht ISDN-fähige Geräte an die R-Schnittstelle angeschlossen werden.

Die D-Kanal-Rahmen im ISDN enthalten eine SAPI (Service Access Point Identifier, gibt den Dienst an (so ähnlich wie eine Portnummer), und eine TEI (Terminal Endpoint Identifier), die (0-126) das ISDN-Gerät anspricht (127 -¿ Rundruf). Die TEI kann fest eingestellt sein, oder wird von der Vermittlungsstelle vergeben.

Beim ISDN werden die Kanäle von der Vermittlungsstelle bis zum NT (U-Schnittstelle) über eine gewöhnliche Telefonleitung gemultiplext. Es werden jeweils 48 bit in 250 $\mu$s übertragen, d.h. 192 kbit/s. Jeder dieser Rahmen enthält 12 bit an Protokollinformationen, so daß 144 kbit/s für zwei B-Kanäle und den D-Kanal bleibt.

## GSM

## Bluetooth

## Frame Relay

## X25

## ATM

ATM wird absichtlich nicht innerhalb der üblichen sieben Schichten behandelt, da ATM ein eigenes Referenzmodell hat. Zwar wird offensichtlich heiß diskutiert, welche ATM-Schicht jetzt welcher OSI-Schicht entspricht, doch das ist wohl eher eine Glaubensfrage. Wir nehmen das ATM Modell als gegeben hin.

Eigentlich ist das ATM-Referenzmodell kein reines Schichtenmodell, sondern wird oft als Würfel dargestellt. Zusätzlich zu den Schichten Besitzt ATM mehrere *Ebenen*, und zwar eine Ebene für das Ebenenmanagement, und eine für das Schichtenmanagement (die natürlich für jede Schicht vorhanden ist). Die eigentlichen Schichten sind nochmals in eine Netz- und eine Benutzerschicht geteilt.

Die eigentlichen **ATM-Schichten** sind die physikalische Schicht, die ATM-Schicht und die ATM-Anpassungsschicht.

Die ATM-Schicht ist der eigentliche Kern der Hierarchie, sie stellt einen reihenfolgentreuen aber unzuverlässigen Dienst zur Verfügung, der 53 Byte lange **ATM-Zellen** über virtuelle Verbindungen transportiert.

### Physikalische Schicht

Die physikalische Schicht in ATM ist zweigeteilt. Der untere Teil, die eigentliche physikalische Schicht, ist für die eigentliche Übertragung der Daten über das Medium zuständig.

Der obere Teil der Schicht, der Transmission Convergence (TC) Layer, übernimmt Aufgaben, die im OSI-Modell in Schicht 2 liegen: Zellenbildung, Berechnung der Zellchecksumme und ähnliches mehr. Beim einem synchronen Medium werden zusätzlich leeren (träge) Zellen in den Strom eingefügt wenn keine Daten vorliegen. Bei einem asychronen Medium ist das Verfahren zum Finden der Zellgrenzen recht witzig: Die Daten werden Bit für Bit durch ein Schieberegister geschoben, bis ATM etwas findet das wie ein Zellkopf mit einer gültigen Prüfsumme aussieht.

### Das ATM-Zellformat

Das Format der ATM-Zellen ist sowohl der ATM- als auch der physikalischen Schicht bekannt, was natürlich die Methoden des Schichtenentwurfes verletzt (allerdings scheint das bei praktischen Systemen häufiger vorzukommen... siehe TCP).

Die ATM-Zellen kommen in zwei Versionen: User-zu-Netz (User Network Interface UNI) und Netz-zu-Netz (Network Network Interface NNI). Der einzige Unterschied zwischen den beiden ist, daß die UNI-Zellen am Anfang ein zusätzliches Feld haben (dafür eine kürzer Path ID). Dieses Feld war für irgendwelche Steurzwecke vorgesehen, und wird nicht benutzt. Jede Zelle hat zunächst einmal eine Virtual Path ID (VPI) und eine Virtual Channel ID (VCI), die für Routing und Adressierung genutzt werden. Außerdem ist ein Feld für mit Informationen über die enthaltenen Daten vorhanden (Payload Type, PTI), und ein Bit, daß ob die Daten bei Staus als erste verworfen werden sollen (Cell Loss Priority, CLP). Der 40 Bit große Header wird durch eine 8 Bit Prüfsumme geschützt. Danach folgen 48 Byte ungeschützte Daten.

### Die ATM-Schicht

Die ATM-Schicht entspricht in vielen Belangen der Vermittlungsschicht im OSI-Modell. Insbesondere wird hier das Routing und die Stauüberwachung realisiert. Die ATM-Schicht bietet den Endstellen einen reihenfolgentreuen, nicht zuverlässigen Dienst an (d.h. Zellen dürfen verloren gehen). Über die Dienstgüte darf von den Endstellen und vom Netz verhandelt werden.

Beim **Verbindungsaufbau** kann zunächst einmal zwischen verschiedenen **Dienstklassen** gewählt werden:

- **Constant Bitrate, CBR:** Es wird eine feste Bandbreite für diese Verbindung reserviert. Diese Brandbeite steht immer zur Verfügung.
- **Variable Bitrate, VBR:** Dieser Dienst ist für Anwendungen mit variablen Bitraten ausgelegt, er ist in einer Version mit Echtzeitanforderungen (VBR-RT) ohne Jitter und in einer Version ohne Echtzeitanforderungen (VBR-NRT) zu haben. Auch hier werden die Resourcen im Vorraus reserviert
- **Available Bitrate, ABR:** Hier werden eine minimale und eine maximale Bitrate ausgehandelt. Die Resourcen für die minimale Bitrate werden reserviert, alles was darüber hinausgeht wird nach bestem Bemühen zur Verfügung gestellt. Dies ist die einzige Klasse, die Informationen über Überlastungen erhält.
- **Unspecified Bitrate, UBR:** Grob gesagt: Die Station darf Daten senden wie sie will und wenn Resourcen vorhanden sind werden die Daten sogar übertragen. Eine Benarichtigung über Überlastung findet nicht statt.

ATM benutzt in jedem Fall feste Verbindungen, die in alle Router entlang der Strecke eingetragen werden – Zellen der gleichen Verbindung laufen also auch immer über die gleiche Route. Welche Route eine Zelle einschlagen soll, entscheidet der Router anhand der Path ID, die Channel ID dient dazu, verschiedene Verbindungen

zwischen den gleichen Endsystemen zu unterscheiden (allerdings ist es theorestisch auch möglich, daß die Router auch die Channel ID heranziehen). Um einen solchen Pfad aufzubauen, wird zuerst eine Verbindungsanfrage (SETUP) über Pfad 0, Kanal 5 an die jeweils nächste Gegenstelle gesendet. Die Gegenstelle bestätigt die Verbindungsanfrage (CALL PROCEEDING), reserviert die benötigten Resourcen, trägt die Verbindung in seine Routingtabelle ein und kontaktiert den nächsten Router auf dem Pfad. Sind nicht genug Resourcen vorhanden, wird der Verbindungswunsch abgelehnt und es kann versucht werden einen anderen Pfad zu verwenden. Dies geht solange weiter, bis der gewünschte Teilnehmer erreicht ist. Dieser sendet dann eine CONNECT-Nachricht auf dem selben Weg zurückt (auch diese wird wieder auf jeder Teilstreck bestätigt. Danach steht die Verbindung, und ein Router muß ein eingehendes Paket nur noch noch der Pfad-ID überprüfen und entsprechend weiterleiten. Um beim Verbindungsaufbau den richtigen Pfad in Richtung Ziel zu finden, können die Router einen beliebigen Routingalgorithmus verwenden. Verbindungen können außerdem vom Netzbetreiber permanent eingerichtet werden. Um eine Verbindung abzubauen, sendet der Teilnehmer eine RELEASE-Nachricht, die von der Gegenstelle bestätigt wird. Diese Nachricht pflanzt sich dann wie gehabt durch das Netz fort, bis die Verbindung komplett abgebaut ist.

Bei jedem Verbindungswunsch wird die **Dienstqualität** mit ausgehandelt, also z.B. die minimale, maximale und durchschnittliche Zellrate, zulässige Verzögerung und Jitter, maximale Verlustrate und Toleranzen.

Eine **ATM-Adresse** kann verschiedene Formate haben, zulässig sind z.B. OSI-Adresse und ISDN-Telefonnummern.

### Traffic-Shaping

Damit eine Station die ihr zugewiesene Bandbreite nicht übersteigt wird eine Art Bucket-Mechanismus verwendet: ATM geht davon aus, daß höchsten alle $T$ Zeiteinheiten eine Zelle gesendet wird. Eine gewisse Toleranz ist zulässig, doch die nächste Zelle ist auf jeden Fall erst zum Zeitpunkt $2T$ zulässig, damit die Toleranz nicht ausgenutzt wird.

### Überlastungsüberwachung der ATM-Schicht

Eine Überlastungsüberwachung für das Netz wird nur in der ABR-Dienstklasse geboten. Dazu wird alle $k$ Zellen eine Resourcenmanagement- (RC-) Zelle gesendet, die nach ihrer Ankunft umkehrt und in die Gegenrichtung zurückgesendet wird. Sowohl auf dem Hin- als auch auf dem Rckweg können hier die Vermittler die maximal zulässige Bitrate eintragen (Vermittler können auch selbst RC-Zellen losschicken). Kommt die RC-Zelle nicht zurück, wird dies als Überlastung interpretiert.

### Die ATM-Anpassungsschicht (AAL)

Die AAL-Schicht bietet etwas praktischere Dienste an, insbesondere zum Verschicken größerer Datenpakete. Es gibt verschiedene Dienste, zuverlässige und unzuverlässige, aber keinen einfachen verbindungsorientierten Dienst wie TCP.

Eigentlich besteht die AAL- Schicht aus drei Unterschichten: Einer AAL-Konvergenzschicht, die in einen allgemeinen und einen anwendungsspezifischen Teil geteilt ist, und der SAR-(Segmentation and Reassembly-)Teilschicht.

**AAL 1** bietet einen unzuverlässigen Dienst für z.B. Videoübertragungen, der lediglich fehlende und falsch eingefügte Zellen erkennt. AAL 1 benutzt keinen Header auf der Konvergenzschicht und fügt zu jeder Zelle eine 3-bit Folgenummer hinzu die durch eine 3-bit Checksumme und ein Paritätsbit geschützt wird. Optional ist noch ein Zeiger auf die Daten vorhanden, falls diese nicht an einer Zellgrenze ausgerichtet sind.

**AAL 2** sollte einen ähnlichen Dienst wie AAL 1 anbieten, allerdings besser auf variable Bitraten ausgerichtet. Der Standard wurde so eingerichtet, daß er nicht zu benutzen ist (Feldlängen fehlen).

**AAL 3/4** bietet die zuverlässige und unzuverlässige Übertragung von Datenströmen oder Nachrichten bis 64k an. Außerdem kann AAL 3/4 mehrere Sitzungen über einen einzelnen virtuellen Kanal multiplexen. Diese AAL-Schicht verwendet einen Header sowohl auf der Konvergenzschicht, als auch für jede einzelne Zelle. Der Konvergenzschicht-Header enthält ein Statusfeld (Nachrichtentyp u.ä.), ein Start- und Endeflag und eine Längenangabe für die enthaltenen Daten. Eine Längenangabe wird auch noch noch dem Endeflag angehängt. Jede Zelle enthält dann noch einmal Typinformationen, einen Folgenummer, eine ID für das Multiplexing und eine 10-Bit Checksumme (plus ein LI-Feld).

**AAL 5** wurde entwickelt, weil einigen Leuten der hohe Overhead bei AAL 3/4 nicht gefiel. AAL 5 bietet ähnliche Dienste an, hat aber lediglich einen Header auf der Konvergenz-Schicht. Ein Paket besteht hier aus bis zu 64k Daten, einem Feld für Steuerinformationen der höheren Schicht, einem Feld für spätere Verwendung, einer Längenangabe und einer 4-Byte Prüfsumme. In jeder Zelle werden alle 48 Byte für Daten benutzt, Paketgrenzen werden über ein Bit in den Steuerinformationen der ATM-Zelle erkannt. (Nicht ganz elegant, aber effizient.

## Copyright