Zusammenfassung für die mündliche Diplomprüfung im Fach Telematik

Naja von Schmude

19. Oktober 2009

Inhaltsverzeichnis

Dies ist meine Zusammenfassung des Lernstoffs für die mündliche Diplomprüfung im Bereich technischer Informatik an der Freien Universität Berlin. Sie umfasst den Stoff der Vorlesung Telematik mit den untersten drei Schichten des ISO/OSI-Referenzmodells und Medienzugriffsverfahren der Mobilkommunikationsvorlesung. Als Grundlage dient das Telematikskript von der Vorlesung im Wintersemester 2008/2009 und das Skript von Mobile Communication vom Sommersemester 2009 sowie das Buch Computernetzwerke von Andrew S. Tanenbaum und Mobilkommunikation von Jochen Schiller.

1 Bitübertragungsschicht

1.1 Theoretische Grundlagen

- Periodische Funktionen können als (unendliche) Summe von Sinus- und Kosinusfunktionen gebildet werden (*Fourier-Analyse*).
- Den Frequenzbereich, der ohne größere Dämpfung übertragen wird, bezeichnet man als *Bandbreite*. Das ist eine physikalische Eigenschaft des Übertragungsmediums und hängt i.A. von Aufbau, Dicke und Länge ab. Es gilt, dass die Einschränkung der Bandbreite (auch bei perfekten Kanälen) die Datenrate begrenzt.
- Die *Symbolrate* ist die Anzahl der Abtastwerte pro Sekunde (in Baud), also die Anzahl der Signalisierungen, die auf das Medium gegeben werden. Die *Datenrate* ist die über den Kanal gesendete Menge an Information. Sie entspricht der Symbolrate mal der Anzahl von Bit pro Zeichen.
- Maximale Datenübertragungsraten:
  Nyquist-Theorem für rauschfreie Kanäle:

$$\max Datenrate = 2 * Bandbreite * \log_2 Signallevel \frac{Bit}{s}$$

  Shannon-Theorem für Kanäle mit Rauschen:

$$\max Datenrate = Bandbreite * \log_2 \left(1 + \frac{S}{N}\right) \frac{Bit}{s}$$

  Meistens wird Signal-Rausch-Verhältnis in dB angegeben: $10 \log_{10} \frac{S}{N}$
- Bei der *Basebandübertragung* wird direkt ein digitales Signal so wie es ist übertragen (Ethernet). Bei der *Broadbandübertragung* wird hingegen die Information als analoges Signal übertragen. Dabei wird es auf ein Trägersignal moduliert (Funk, Glasfaser).

- Per *Pulse-Code-Modulation* (PCM) werden analoge Signale in digitale überführt (und andersrum). Das Verfahren beruht auf dem Samplingtheorem von Nyquist, nach dem die doppelte Frequenz der im Signalverlauf vorkommenden höchsten Frequenzkomponente zum *Abtasten* ausreicht um alle Informationen des Originalsignals zu erhalten. Danach folgt die *Quantisierung*, bei der das analoge Signal in diskrete Werte gewandelt wird. Der *Quantisierungsfehler* ist die Differenz zischen analogem Signalwert und digitalem Signalwert. Bei der *Kodierung* wird die Quantisierung einem Binärcode zugeordnet. Es gibt auch noch weitere Verfahren: Differentielle PCM oder die Deltamodulation.

1.2 Leitungskodierung

- Bei *RZ* (Return to Zero) kehrt das Signal zwischen jedem Puls zurück zur Null, so dass das Signal selbst den Takt angibt aber doppelte Bandbreite benötigt.
- Bei *NRZ* (Non Return to Zero) wird das Signal direkt in Spannung übersetzt, so dass z.B. hohe Spannung für eine 1 steht und niedrige Spannung für eine 0. Kann dazu führen, dass man die Synchronisation verliert, wenn lange Sequenzen an gleichen Bits kommen.
- Beim *differential NRZ* wird die 1 als Wechsel des Levels kodiert und eine 0 als gleichbleibendes Level.
- Der *Manchester Code* benutz einen Spannungswechsel in der Mitte von jedem Bit. Eine 0 wird als Änderung von Positiv zu Negativ kodiert und eine 1 als Negativ-zu-Positiv-Flanke. (In Ethernet sind die Flanken vertauscht).
- Beim *Differential Manchester Code* wird immer ein Levelwechsel gemacht und entsprechend der Kodierung noch ein zweiter: Eine 1 wird als wegbleibender Levelwechsel kodiert und eine 0 als ein zusätzlicher Levelwechsel.
- Bei *4B/5B* werden vier Bit in fünf Bit kodiert, so dass es eine höhere Effizient hat (Manchester z.B. nur 50%). Jeder der 16 möglichen 4-Bit-Kombinationen ordnet man eine 5-Bit-Kombination zu. Übrigbleibende 5-Bit-Wörter können dann für Steuerzwecke o.ä. verwendet werden.

1.3 Übertragungsmedien

Kabelgebundene Medien

Twisted Pair Zwei ineinander verdrillte Kupferdrähte. Die Verdrillung bewirkt, dass die Wellen durch die Verwindungen gebrochen werden und das Kabel weniger Störungen ausstrahlt.
Es gibt verschiedene Typen: Cat3, Cat5 (mit mehr Windungen), Cat6 und Cat7 (wo jedes Kabel zusätzlich noch mit Silberfolie geschützt ist).
Die Bitfehlerrate liegt bei ungefähr 10^{-5}.

Koaxialkabel Ein starrer Kupferdraht, der mit eine Isolierschicht ummantelt ist. Der Isolator ist wiederum von einem zylindrischen Leiter umschlossen (geflochtenes Netz), der Interferenzen verhindert.
Koaxialkabel unterstützen höhere Datenraten über längere Distanzen als Twisted-Pair (1-2Gbit/s über 1km) und haben niedrigere Bitfehlerraten von 10^{-9}.

Glasfaserkabel Es besteht aus drei Komponenten: Lichtquelle, die für eine 1 ein Impuls sendet und für eine 0 kein Impuls sendet, das Übertragungsmedium, welches aus einer hauchdünnen Glasfaser besteht, und einem Detektor, der ein Lichtsignal wieder in ein elektrisches Signal umwandelt.
Die Lichtquelle kann eine LED (billiger und zuverlässiger, dafür breiteres Spektrum an Licht und somit geringere Datenraten) oder Laser (teurer aber dafür geringeres Spektrum und somit höhere Datenraten). Die Glasfaser ist von einem Glasmantel umgeben, der einen höhere Brechungsindex hat als die Faser selbst, so dass das Licht vollständig reflektiert wird und in der Faser gefangen bleibt.
Probleme sind die Streuung von Strahlen, die die Reichweite begrenzt und der Modus, der dafür sorgt, dass jeder Strahl einen anderen Weg nimmt und somit sich gegenseitig beeinflussen. Dagegen gibt es *Einzelmodefasern*, die so dünn sind (8-10 μm), dass sich das Licht nur in einer Richtung ausbreiten kann (sind aber auch teurer) und somit höhere Bandbreiten und Übertragungsdistanzen möglich sind. *Multimodefasern* haben eine Dicke von 50 μm und lassen zu, dass Lichtstrahlen verschiedene Wege nehmen.
Man benutzt Wellenlängen von 0,85 μm, 1,3 μm und 1,55 μm, da hier die Dämpfung am geringsten ist. Die Datenrate geht theoretisch bis ca. 50000 Gbit/s bei einer Bitfehlerrate von 10^{-12}.

Im Vergleich von Kupferdraht zu Glasfaser schneidet zumeist Glasfaser besser ab. Sie unterstützt wesentlich höhere Bandbreiten. Außerdem ist die Dämpfung geringer (nur alle 50km wird ein Repeater benötigt, bei Kupfer ca. alle 5km). Außerdem ist Glasfaser durch Stromstöße, elektromagnetischen Störungen und Stromausfällen nicht beeinflussbar. Zudem sind sie dünn und wiegen weniger als Kupferkabel. Auch sind sie abhörsicherer, da man sie nicht anzapfen kann. Nachteil von Glasfaser ist, dass man sie leicht durch starkes Biegen zerbrechen kann und für bidirektionale Kommunikation zwei Leitungen benötigt.

Drahtlose Übertragungen

Funk

Satelliten

1.4 Modulationsverfahren und Modems

- Man benutzt das normale Telefonnetz für die Datenübertragung. Dazu muss man digitale Daten über ein analoges Medium schicken. Um die digitalen Daten vom Computer nun in analoge zu wandeln, benötigt man ein *Modem* (Modulator-Demodulator).

- Rechteckswellen, die durch digitale Signale entstehen, haben ein großes Frequenzspektrum und sind somit besonders der Dämpfung und Laufzeitverzerrung ausgesetzt. Daher benutzt man AC-Signalübertragung, die eine sinusförmige Trägerfrequenz (Sinusträger) gebraucht.

- Es wird ein Sinusträger in seiner Amplitude, Frequenz oder Phase moduliert, um Informationen zu übertragen.

  Amplitudenmodulation (Amplitude Shift Keying) Es werden verschiedene Amplituden verwendet, um 0 und 1 zu kodieren. Die Variante ist leicht zu realisieren, ist aber gegenüber Störungen anfällig.

  Frequenzmodulation (Frequency Shift Keying) Es werden hier verschiedene Frequenzen verwendet, um 0 und 1 darzustellen. Hier benötigt man eine hohe Bandbreite, da man Frequenzen verschwendet

  Phasenmodulation Beim Phase Shift Keying wird je nach Information die Phase gewechselt. Dies ist das aufwendigste Verfahren, dafür aber auch am robustesten gegenüber äußeren Einflüssen.

  QPSK (Quadrature Phase Shift Keying) Es werden vier verschiedene Phasen verwendet, so dass man vier Zustände hat, dadurch kodiert man 2 Bit auf einmal (dadurch doppelte Datenrate).

  Man kann natürlich auch mehrere Modulationsverfahren kombinieren, um so noch höhere Datenraten zu erzielen. Bei QAM-16 verwendet man z.B. vier verschiedene Amplituden und vier Phasen, so dass man 4 Bit pro Symbol überträgt. Es gibt noch viele weitere Standards (QAM-64, V.32 für 9,6kBit/s und V.32 für 14.4kbit/s).

1.5 Multiplexing

Um nicht tausende Leitungen zu legen, will man mehrere Verbindungen über dieselbe Leitung laufen lassen und so eine Ressource mehrfach verwenden. Man unterscheidet mehrere Multiplexingvarianten.

Frequenzmultiplexing Hier wird das Frequenzspektrum in Frequenzbänder aufgetailt, wobei jeder Benutzer nur im Besitz von einem Band ist. Dabei werden die Kanäle jedes Benutzers auf die entsprechende Frequenz gebracht und zusammengeführt. Zwischen den Bändern ist ein gewisser sich überlappender Bereich (da durchs Filtern keine scharfen Grenzen entstehen), so dass es zu gegenseitiger Beeinflussung in Form von Rauschen kommen kann.
Ein Standard ist 12 4-kHz-Kanäle in das Band von 60-108 kHz zu multiplexen. Eine solche Einheit ist eine *Gruppe*. Fünf Gruppen bilden eine *Supergruppe* und fünf Supergruppen eine *Mastergruppe*.

Wellenlängenmultiplexing Eine Art Frequenzmultiplexing. Es werden vier Glasfasern in einem optischen Combiner zusammengefasst, wobei hier jede Glasfaser eine andere Wellenlänge verwendet. Über eine Glasfaser wird werden die vier Signale

dann übertragen und am Ende wieder auf vier verschiedene Glasfasern aufgeteilt. Dazu werden sie passiv durch ein Beugungsgitter wieder auseinander gefiltert (dadurch völlig zuverlässig).

Zeitmultiplexing Hierbei bekommen die Benutzer hintereinander nach dem Round-Robin-Verfahren einen Timeslot zugewiesen, in dem sie die volle Bandbreite zu Verfügung haben. Es kann nur für digitale Daten verwendet werden. Ein Codec (Coder-Decoder) wandelt analoge Signale in digitale um (und umgekehrt). Beim T1-Träger werden 24 Sprächkanläle gemultiplext. Dabei werden die Analogsignale reihum abgetastet. Jeder der 24 Kanäle darf reihum 8 Bit in den Ausgangsstrom einfügen.

1.6 Vermittlung

Leitungsvermittlung (Circuit Switching) Die Vermittlungseinrichtungen suchen einen durchgehenden physikalischen Pfad von einem Teilnehmer zum anderen. Es ist notwendig eine Ende-zu-Ende-Verbindung einzurichten bevor Daten gesendet werden. Der dedizierte Weg bleibt dann solange erhalten bis die Verbindung beendet wird. Diese Art ist für die Benutzer völlig transparent, die Rahmenbedingungen wie Übertragungsrate, Format usw. können selber gewählt werden.

Nachrichtenvermittlung (Message Switching) Es wird kein Pfad im Voraus berechnet, stattdessen wird jede Nachricht zur ersten Vermittlungsstelle geleitet und von dort Hop für Hop weiter. Hier ist der Größe der Nachricht keine Grenzen gesetzt, so dass die Stationen in der Lage sein müssen auch große Blöcke zwischen zuspeichern.

Paketvermittlung (Packet Switching) Hier wird die Blockgröße nach oben begrenzt. Wie bei der Nachrichtenvermittlung wird kein Pfad im Voraus reserviert, sonder jedes Paket kann seinen eigenen Pfad nehmen, so dass unter Umständen die Reihenfolge der Pakete auch vertauscht werden kann. So ist diese Art auch fehlertoleranter und effizienter, da keine Bandbreite verschwendet wird. Es verwendet eine Store-and-forward-Übertragung, d.h. ein Paket wird im Hauptspeicher eines Routers gespeichert und dann an den nächsten Router weitergesendet, das bringt eine gewisse Verzögerung mit sich. Hier bestimmt der Träger die Parameter der Übertragung wie Übertragungsrate usw.

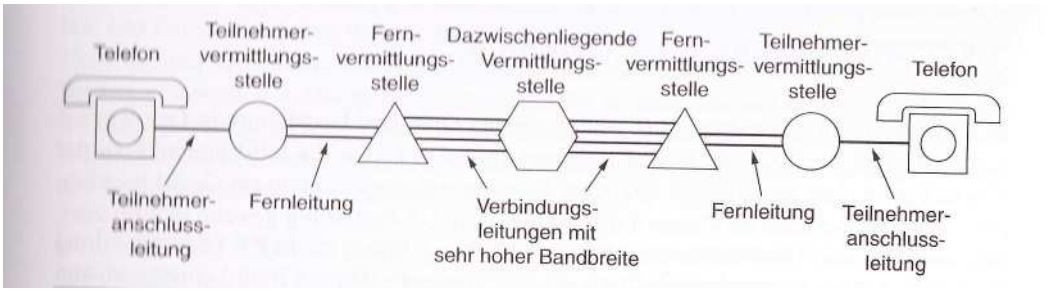
1.7 Telefonsystem / ISDN / DSL

- Drei wichtige Komponenten im Telefonnetz:

  Teilnehmeranschlussleitungen „Letzte Meile“ meist Twisted-Pair-Kabel direkt zum Endnutzer mit analogem Signal.

  Verbindungsleitungen sind digitale Glasfaserkabel, die die Vermittlungsstellen verbinden

Vermittlungsstellen leiten Anrufe von einer Verbindungsleitung zu nächsten weiter (es gibt eine ganze Hierarchie davon)



- ...

2 Sicherungsschicht - Logical Link Control

Die Sicherungsschicht soll der Vermittlungsschicht *Dienste* zu Verfügung stellen, und zwar solche, die dazu dienen Daten von der Vermittlungsschicht von einer Quelle zur Vermittlungsschicht des Ziels zu übertragen. Man unterscheidet drei Arten von Diensten:

Unbestätigter verbindungsloser Dienst Es wird keine logische Verbindung zwischen den Stationen aufgebaut und die Frames werden unabhängig zum Ziel gesendet und dort der Empfang nicht dem Sender bestätigt. Dies ist angemessen, wenn die Fehlerquote niedrig ist oder z.B. bei Echtzeitanwendungen, wo die Fehlerkorrektur und Wiederanforderung von Frames zu lange dauern würde.

Bestätigter verbindungsloser Dienst Es wird immer noch keine Verbindung zwischen Sender und Empfänger aufgebaut, aber der Empfang jedes Frames wird bestätigt, so dass verloren gegangene oder beschädigte Frames erneut übertragen werden können. Benutzt man in unzuverlässigen Kanälen wie man sie z.B. in drahtlose Systemen antrifft.

Bestätigter verbindungsorientierter Dienst Eine logische Verbindung wird aufgebaut, über die dann durchnummerierte Frames gesendet werden. Diese Verbindungsart garantiert, dass jeder Frame empfangen wird und zwar auch nur genau einmal und dass die Reihenfolge der Frames beibehalten wird.

2.1 Frames

Der Bitstrom wird in diskrete Einheiten unterteilt, die so genannten *Frames*. Für die Abgrenzung von mehreren Frames verwendet man verschiedene Methoden:

Zeichenzahl Es wird ein Feld im Header verwendet, in dem die Zeichenzahl (also die Länge des Frames) angegeben wird. Wird das Zeichenzahlfeld durch einen Übertragungsfehler beschädigt oder verändert, werden dadurch die Frames falsch begrenzt und nichts stimmt mehr. Die Synchronisation ist quasi im Arsch.

Flagbyte mit Byte Stuffing Es wird zur Synchronisation ein spezielles Flagbyte am Anfang und am Ende jedes Frames verwendet. Falls das Bitmuster des Flags in den Nutzdaten vorkommt, muss es „gestufft“ werden. Dazu wird vor jedem Auftauchen des Flagbytes in den Daten ein spezielles Escapebyte eingefügt, welches beim Empfänger automatisch entfernt wird. Sollte das Escapezeichen selbst vorkommen, wird auch dieses gestufft. Der Nachteil hierbei ist, dass man hier an 8 Bit-Zeichensätze gebunden ist.

Flags mit Bit Stuffing Auch hier wird ein spezielles Flagbyte (01111110) zur Markierung von Anfang und Ende jedes Frames verwendet. Nun funktioniert das aber so, dass man immer nach fünf aufeinanderfolgenden 1en eine 0 einfügt. Dies nennt man Bit Stuffing. Beim Empfänger wird wieder nach jeweils fünf 1en eine folgende 0 entfernt. Im Gegensatz zum Byte Stuffing können hier die Rahmen Größen haben, die nicht in Byte anzugeben sind.

Kodierungsverletzungen auf Bitübertragungsschicht Findet nur Verwendung, wenn ein Datenbit mit zwei physikalischen Bits kodiert wird. Bei Verwendung von Manchester kann man z.B. die nicht erlaubten Übergänge Low-Low und High-High zur Rahmenbegrenzung verwenden.

Verschiedene Verfahren können natürlich auch kombiniert werden.

2.2 Flusskontrolle

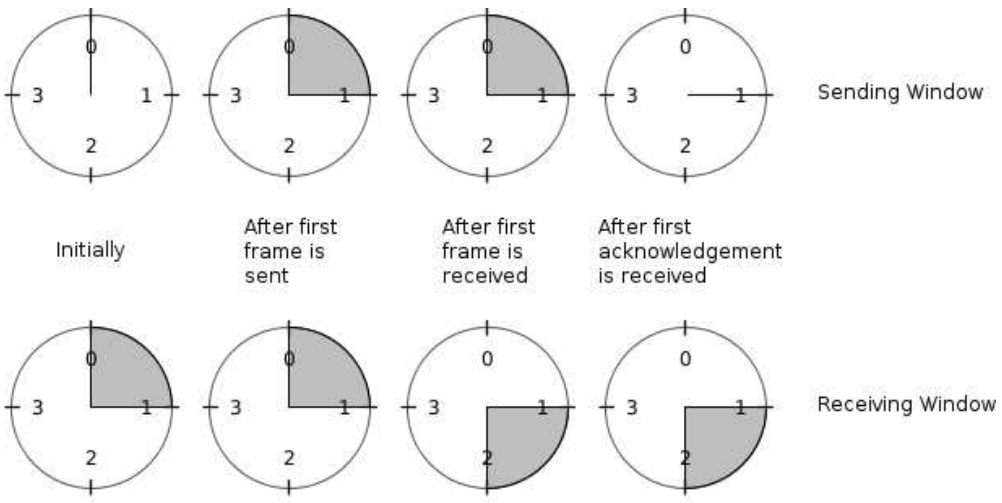
- Flusskontrolle soll sich darum kümmern, dass ein langsamer Empfänger nicht durch einen schnelleren Sender überschwemmt wird, so dass der Empfänger mit der Verarbeitung der Frames nicht hinterher kommt.

- Bei der *Feedback-based Flusskontrolle* sendet der Empfänger Informationen an den Sender zurück, um ihm die Erlaubnis zum weiteren Senden zu geben bzw. über den aktuellen Zustand zu informieren.

- Um Flusskontrollprotokolle zu entwickeln, kann man einen einfachen Ansatz wählen und von unrealistische Annahmen (Fehlerfreies Medium, Sender und Empfänger haben unbegrenzten Puffer usw.) ausgehen und darauf basierend nach und nach Verschärfungen einführen und so das Protokoll nach und nach entwickeln. Im Skript sind das die *Simplex-Protokolle*.

- Mit dem *Sliding-Window Protokoll* will man eine vollduplex Kommunikation ermöglichen. Dabei nimmt man denselben Kanal für Datenverkehr in beide Richtungen, so dass ACKs und Daten vermischt sind. Um die Bandbreite besser zu nutzen, schickt man ACKs mit Datenpakete (als Feld im Header) mit; dies nennt

man *Piggybacking*. Natürlich geht das nur, wenn in einem kurzen Zeitintervall ein Paket auch verschickt werden soll, falls gerade nichts anfällt, wird das ACK halt alleine übermittelt.
Bei Sliding-Window sind alle Frames durchnummeriert (meist von 0 bis $2^n - 1$). Das Hauptmerkmal ist, dass Sender und Empfänger jeweils ein *Fenster* verwalten, dass angibt wie viele Frames sie noch Senden bzw. Empfangen können.
Das *Sendefenster* repräsentiert die Frames, die versendet aber noch nicht bestätigt wurden (oder noch nicht gesendet). Kommt aus der Vermittlungsschicht ein neues Paket, wird die nächsthöhere Sequenznummer vergeben, die Obergrenze des Fenster um eins verschoben. Kommt eine Bestätigung rein, wird die Untergrenze des Fensters verschoben. Alle Frames, die sich im Sendefenster befinden, müssen gepuffert werden, da sie bei einem Fehler erneut übertragen werden müssen.
Das *Empfangsfenster* repräsentiert die Anzahl an Frames, die der Empfänger aufnehmen kann. Alles, was nicht in das Fenster passt, wird verworfen. Auch hier wird beim Empfang eines Paktes das Fenster weitergeschoben, wenn die Framenummer der unteren Fensterbegrenzung entspricht. Das Fenster behält aber immer seine ursprüngliche Größe bei.



A sliding window with a 2-bit sequence, of size 1

Es gibt verschiedene Variationen von Sliding-Window:

1-Bit-Sliding-Window Bei der Fenstergröße von 1 geht das Protokoll nach *Stop-and-Wait* vor, d.h. der Sender sendet ein Frame und wartet so lange, bis er die Bestätigung vom Empfänger bekommt. Erst dann kann er den nächsten Frame übermitteln. Wenn beide Kommunikationspartner gleichzeitig anfangen zu senden, kann es vorkommen, dass obwohl keine Übertragungsfehler

auftreten, viele Frames doppelt übermittelt werden.

Go-Back-N Die Übertragungszeit hat erhebliche Auswirkungen auf die effiziente Nutzung der Bandbreite. Daher wählt man die Fenstergröße w des Sender so, dass während der Dauer der Hin- und Rücksendung fortlaufend Frames gesendet werden können, ohne dass das Fenster voll wird (*Pipelining*). Im Falle eines kaputten oder verloren gegangenen Frames verwirft der Empfänger alle nachfolgenden Frames und schickt für diese keine Bestätigung. Wenn beim Sender der Timer abläuft oder das Sendefenster voll ist, überträgt dieser alle unbestätigten Frames erneut und fängt mit dem verloren gegangenen bzw. zerstörten an, da für dieses noch keine ACK angekommen ist.

Selective Repeat Hier werden im Falle eines Fehlers nicht alle danach ankommenden Frames verworfen, sondern zwischengespeichert. Tritt beim Sender ein Timeout auf oder ist das Sendefenster voll, beginnt er wie bei Go-Back-N sukzessive die Frames neu zu übertragen (beginnend mit dem kaputt gegangenen, denn für den Frame fehlt die Bestätigung). Tritt beim Empfänger nun der Frame richtig ein, sendet er eine Bestätigung für alle gepufferten Frames, in dem er ein ACK für den letzten korrekten gepufferten Frame sendet. Daraufhin kann der Sender die Wiederholung lassen und mit neuen Frames fortsetzen.

Selective Reject Es funktioniert wie Selective Repeat, bloß dass negative Bestätigungen (NACK) eingeführt werden, mit denen der Empfänger dem Sender mitteilen kann, welcher Frame verloren gegangen ist. So kann der Sender lediglich den verlorenen Frame erneut übertragen und Bandbreite wird eingespart.

- Protokolle kann man *verifizieren*. Dazu verwendet man verschiedenste Methoden der Graphentheorie. Z.B. kann man *endliche Automaten* benutzen oder *Petrinetze*.
  Bei Ersterem sind die Zustände eine Kombination aus allen Zuständen der beiden Protokollautomaten und des Kanals. Von den Zustand gibt es Übergänge zu anderen Zuständen, die durch bestimmte Ereignisse ausgelöst werden (z.B. Frame Ankunft).
  Bei Letzterem gibt es die Stellen, die die Zustände darstellen. Das System befindet sich in einem Zustand, wenn die Stelle durch eine Marke gekennzeichnet ist. Eine Transition hat beliebig viele Eingangsverbindungen und beliebig viele Ausgangsverbindungen, die zu anderen Stellen führen. Eine Transition ist aktiv, wenn an jeder Eingangsstelle mind. eine Marke vorhanden ist. Wenn sie Transition ausgelöst wird, dann wandern von jeder Eingangsstelle eine Marke zu einer Ausgangsstelle. Bei mehreren aktiven Transitionen kann eine beliebige auslösen.
  Mit diesen beiden Modellen können nun Schwachstellen in der Protokollspezifikation gefunden werden wie Deadlocks, nicht erreichbare Zustände oder unvollständige Spezifikationen.

2.3 Fehlererkennung und Korrektur

- Fehler treten oft nicht vereinzelt, sonder massenweise in *Bursts* auf. Der Vorteil davon ist, dass oft nur eine geringe Anzahl von Frames zerstört wird (in einem treten dann haufenweise Bitfehler auf), so dass nur wenig erneut übertragen werden muss. Andererseits können sie nicht so leicht erkannt und korrigiert werden.

- Es gibt zwei Ansätze:

  Forward-Error-Correction Man fügt so viel redundante Informationen mit ein, dass Fehler erkannt und auch behoben werden können. Dies benutzt man bei fehleranfälligen Medien wie Funkverbindungen.

  Fehlererkennung (Automatic Repeat Request) Man fügt nur eine geringe Menge an redundanter Information ein, um Fehler nur zu erkennen. Falls Fehler auftreten muss der Frame erneut gesendet werden. Dies benutzt man bei zuverlässigeren Medien wie Glasfaser.

- Als *Hamming-Abstand* bezeichnet man die Anzahl der unterschiedlichen Stellen von zwei Codewörtern. Der Hamming-Abstand eines Codes ist die minimale Distanz von zwei Codewörtern des Codes.

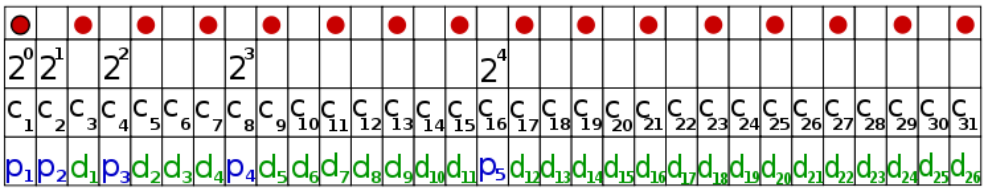
- Zum Entdecken von d Bitfehlern benötigt man einen Code mit Hamming-Abstand von $d + 1$, zur Fehlerkorrektur von d Fehlern sogar einen Abstand von $2d + 1$.

- Die verwendeten verfahren sind folgende:

  Paritätsbit Es wird ein Bit angehängt, welches angibt ob die Anzahl der 1en gerade (0) oder ungerade (1) ist. Es kann ein 1-Bit-Fehler erkannt werden.

  Kreuzpariät Die Bits werden in einer Matrix angeordnet, so dass über jede Zeile und Spalte ein Paritätsbit gebildet wird. Hier kann sogar ein 1-Bit-Fehler behoben werden.

  Hamming-Code Die Paritätsbits (p_i) liegen auf den Zweierpotenzen und bilden sich wieder aus der Anzahl der 1en Modulo 2. Und zwar werden die Datenbits (d_i) betrachtet, die auf den Indizes i des Codeworts stehen, bei denen die entsprechende Zweierpotenz in der Summenbildung von Zweierpotenzen des Index vorkommt. Z.B. prüft das Paritätsbit an der Stelle 2 die Datenbits an den Stellen $3, 6, 7, 10, 11, ...$ Für das Datenbit an Stelle $5 = 1 + 4$ sind die Paritätsbits 1 und 4 zuständig. Der Hamming-Code kann 1-Bit-Fehler korrigieren, in dem der Empfänger die Paritätsbits analysiert und die Indizes der falschen Paritätsbits aufsummiert. Das Ergebnis ist der Index des gekippten Bits.



Wenn man einen Trick macht und die Bits wieder in einer Matrix anordnet, kann man sogar Burstfehler erkennen und korrigieren. Und zwar ordnet man in jeder Zeile ein Codewort an und verschickt die Matrix aber spaltenweise. So ist pro Codewort nicht so viel gekippt (idealerweise nur ein Bit).

CRC (Cyclic Redundancy Check) Zum Erkennen von Burstfehlern eine häufig verwendete Methode. Man betrachtet die Bitketten als Polynome mit den Koeffizienten 0 und 1. Ein Frame mit k Bits wird als $x^{k-1}+x^k+\ldots+x^1+x^0$ interpretiert. Der Grad wäre hier $k-1$. Sender und Empfänger einigen sich auf ein *Generatorpolynom*, dessen höchstwertiges als auch niedrigstwertiges Bit 1 ist. Die Idee ist nun, dass man mit Hilfe des Generatorpolynoms eine Prüfsumme berechnet und an den Frame anhängt, so dass das aus dem Frame plus Prüfsumme entstandene Polynom durch das Generatorpolynom teilbar ist. Wenn jetzt der Empfänger den empfangenen Frame durch das Generatorpolynom teilt und ein Rest bleibt, ist sicher, dass ein Fehler aufgetreten ist. Die Prüfsumme berechnet sich nach folgendem Schema:

1. Hänge r (Grad vom Generatorpolynom) 0-Bits an die Bitfolge an.
2. Teile die Bitkette mit Modulo-2-Division durch das Generatorpolynom (Polynomdivision mit XOR)
3. Ziehe den Divisionsrest mit Modulo-2-Subtraktion (XOR) von der Bitkette mit 0en ab. Das Ergebnis ist der um die Prüfsumme ergänzte Frame, der übertragen wird.

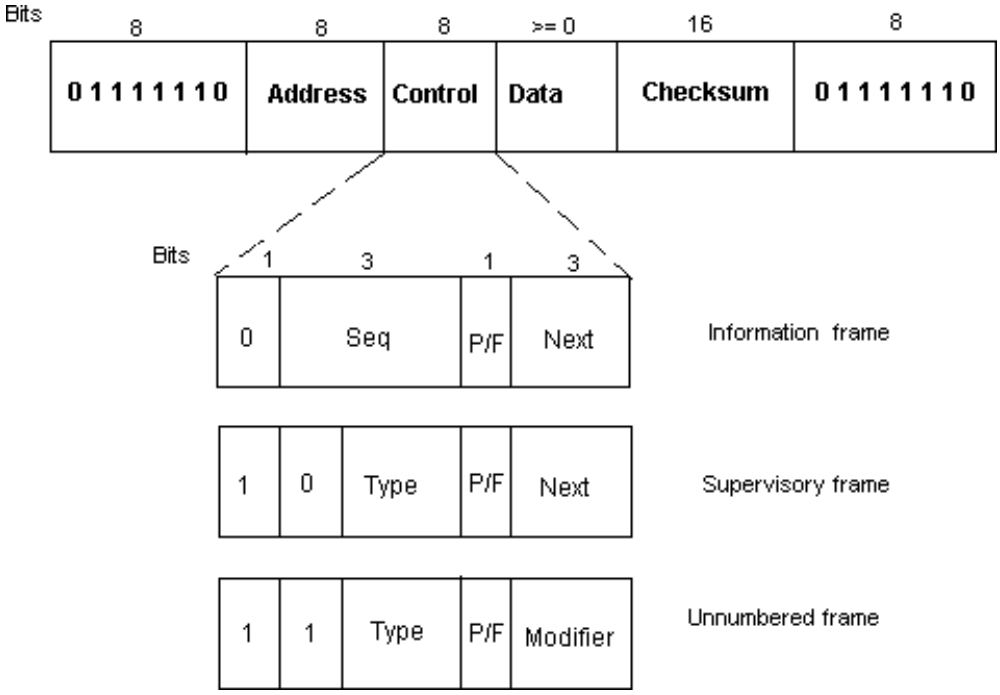
Tritt ein Fehler auf, entspricht das empfangene falsche Polynom statt $T(x)$ $T(x)+E(x)$. Jedes Bit in dem Fehlerpolynom $E(x)$ entspricht einem gekippten Bit. Falls k die Anzahl der 1 Bit in $E(x)$ ist, dann sind k 1-Bit-Fehler aufgetreten. Ist $E(x)$ ein Vielfaches vom Generatorpolynom, so wird der Fehler nicht erkannt. Es werden durch CRC alle 1-Bit-Fehler erkannt. Mit einem CRC-Code mit r Prüfbits erkennt man alle Burstfehler der Länge $\leq r$. In besonderen Fällen kann man mit CRC auch Fehler beheben.

2.4 HDLC, PPP und 802.2

HDLC und PPP sind beides häufig auf der Sicherungsschicht (in der Logical Link Control Teilschicht) verwendete Protokolle. Sie bieten Framing, Flusskontrolle und Fehlererkennung an.

HDLC (High-Level Data Link Control) Dies ist ein bitorientiertes Protokoll, welches Bit Stuffing einsetzt um die nötige Datentransparenz zu erreichen. HDLC ver-

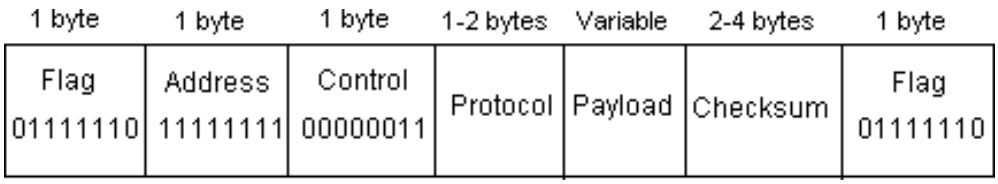
wendet ein Sliding-Window-Algorithmus mit 3-Bit-Sequenznummern, bei denen die Fenstergröße 7 ist. Die Framestruktur sieht wie folgt aus:



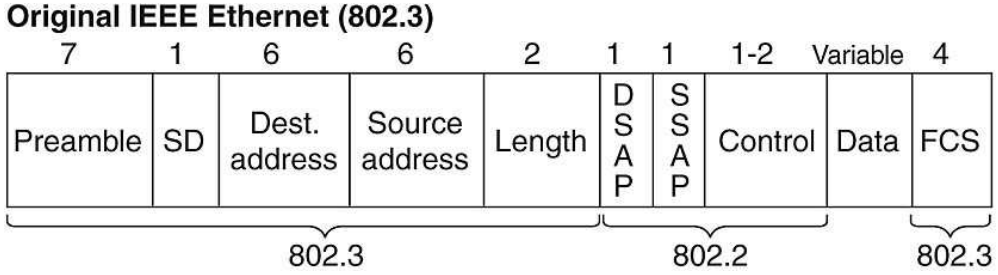
Es gibt verschiedene Framearten: Information, Supervisory und Unnumbered. Informationsframes transportieren Daten mit Flusskontrolle und Fehlerkorrekturmethoden und Piggybacking. Supervisoryframes stellen Flusskontrolle und Fehlerkorrekturmechanismen zu Verfügung, sollte Piggybacking nicht möglich sein. Für Steuerzwecke oder unzuverlässige, verbindungslose Dienste sind die Unnambered Frames gedacht.

PPP (Point-to-Point Protocol) Das Protokoll ist zur Kommunikation zwischen zwei Punkten gedacht (vorwiegend Host und Router bei Internetverbindungen). PPP unterstützt Fehlererkennung, mehrere Protokolle, die Vergabe von IP-Adressen zum Zeitpunkt des Verbindungsaufbaus. Es hat neben der Framebildung auch noch zwei wichtige Merkmale, nämlich ein Verbindungsprotokoll zum Aktivieren und Testen von Leitungen, Verhandeln von Optionen und Beenden von Verbindungen (LCP - Link Control Protocol) und eine Methode zum Aushandeln von Optionen auf der Vermittlungsschicht, die von dem auf der Vermittlungsschicht benutzen Protokoll unabhängig sind (NCP - Network Control Protocol). Nach dem Verbindungsaufbau werden zunächst LCP-Pakete versendet, die die PPP-Parameter festlegen, versendet. Danach folgen NCP-Pakete (hier können z.B. auch IP-Adressen angefordert werden). PPP ist zeichenorientiert und benutzt

Byte Stuffing, lehnt sich im Frameformat aber an HDLC an. Wird nichts anders verhandelt, so ist die Datenlänge auf 1500 Byte eingestellt. NCP und LCP sind auch PPP-Pakete, in deren Payload dann die Optionen stehen. Über das Protocol-Feld wird dies dann definiert (oder welche Vermittlungsschicht angesprochen werden soll).



802.2 802-Protokolle (Ethernet usw.) bieten nur Datenübertragung nach „Best Effort“ an, also unzuverlässige Datagramme. 802.2 bietet nun dafür Fehlererkennung und Flusskontrolle und abstrahiert durch dieses gemeinsame Protokoll für darüber liegenden Schichten die verschiedenen MAC-Protokolle. Es ist an HDLC angelehnt. Es bietet unzuverlässige Datagrammdienste, bestätigte Datagrammdienste und zuverlässige verbindungsorientierte Dienste an. Der Header hat drei Felder: Zielzugriffspunkt, Quellzugriffspunkt und ein Steuerfeld (enthält Folge und Bestätigungsnummern). Die Zugriffspunkte geben an, von welchem Prozess der Frame kommt und zu welchem er übergeben werden soll.



3 Sicherungsschicht - Medium Access Control

3.1 Medienzugriffsverfahren

- Bei Broadcastmedien muss festgelegt werden, wer den Kanal benutzen darf. Protokolle, die den Zugriff für mehrere Benutzer regeln, sind Medienzugriffsverfahren.
- Statische Methoden wären z.B. Frequency Division Multiple Access mit Hilfe Frequenzmultiplexing zu machen, in dem man n Benutzern n gleich große Teile

der Bandbreite zuteilt. Genauso kann man mittels Zeitmultiplexing Time Division Multiple Access realisieren, in dem für n Benutzer n Timeslots zuordnet. Beide Verfahren haben den Nachteil, dass sie nicht für variable Benutzerzahlen funktionieren, da man viel Bandbreite verschenkt.

- Da die statischen Verfahren nicht geeignet sind, versucht man die Ressourcen dynamisch zu verteilen. Dazu gibt es folgende Methoden:

  ALOHA Benutzer dürfen hier völlig unkoordiniert Daten senden. Sobald eine Station Daten hat, sendet sie Frames gleicher Länge. Dabei treten natürlich Kollisionen auf, sobald eine Kollision auftritt, wartet der Sende eine zufällige Zeitspanne und probiert es erneut. Die gefährliche Zeitspanne, in der Kollisionen auftreten können, beträgt 2 mal die Framelänge. ALOHA nutzt den Kanal bestenfalls zu 18%.

  Slotted ALOHA Die Zeit wird in Intervalle aufgeteilt, wobei in jeden Timeslot ein Frame passt. Nun können die Stationen nur noch zum Beginn eines Frames anfangen zu senden. Die gefährliche Zeitspanne ist hier dann nur noch so lang wie ein Frame. Die Kanalnutzung ist hier bei 37%.

  CSMA (Carrier Sense Multiple Access) Dies bedeutet, dass der Sender das Medium abhört bevor er sendet. Wird derzeit nichts übertragen ist das Medium frei und der Sender kann mit der Übertragung beginnen. Funktioniert nur, wenn der Übertragungsdelay kurz ist.

  1-persistent CSMA Ist der Kanal beim Abhören besetzt, wartet die Station, bis das Medium frei wird und fängt dann sofort an zu senden. Bei einer Kollision wird eine zufällige Zeitspanne und versucht es erneut. Sollten mehrere Stationen das Medium als besetzt sehen, dann fangen sie sofort beide nach Beendigung der Übertragung an zu senden, was zu einer Kollision führt.

  Nonpersistent CSMA Hier fängt die Station zu senden an, wenn das Medium frei ist. Ansonsten wartet sie eine zufällige Zeit ab und probiert es erneut. Insbesondere muss die Station hier nicht ständig den Kanal abhören, um mitzubekommen, wann das Medium frei ist. Im Gegensatz zu 1-persistent CSMA hat man eine bessere Kanalauslastung aber dafür mehr Wartezeit.

  p-persistent CSMA Dieses Verfahren funktioniert mit Verfahren, die Timeslots benutzen. Es wird bei einem freien Kanal mit Wahrscheinlichkeit p die Übertragung begonnen und mit Wahrscheinlichkeit $1-p$ auf den nächsten Timeslot gewartet. Ist dieser auch frei, sendet man wieder mit Wahrscheinlichkeit p. Das geht so lange, bis der Frame übertragen wurde oder eine andere Station anfängt zu senden (wie Kollision behandelt), so dass dann eine zufällige Zeitspanne gewartet wird.

  CSMA/CD (Carrier Sense Multipe Access with Collision Detection) Diese Variante bricht zusätzlich noch die Übertragung ab, sobald eine Kollision erkannt wird, so wird Zeit und Bandbreite gespart. Kollisionen

werden daran erkannt, dass Leistung oder Impulsbreite des empfangenen Signals nicht mit dem vom Sender übertragenen übereinstimmt (dazu muss die Station während dem Senden weiter den Kanal abhören). Die minimale Zeit, bis zur Entdeckung einer Kollision ist die Zeit, die ein Signal brauch um von einer Station zu anderen zu gelangen. Nach dem Abbruch der Übertragung wird dann wieder eine zufällige Zeit gewartet und dann erneut probiert. CSMA/CD besteht so aus sich abwechselnden Konkurrenz- und Übertragungsperioden. Dabei wird die Konkurrenzperiode als Slotted ALOHA System mit Timeslots der Größe der RTT (Round Trip Time) modelliert. CSMA/CD mit einem Kanal ist ein Halbduplexsystem, da während des Sendens die Empfangslogik zum Entdecken von Kollisionen verwendet wird.

Kollisionsfreie Protokolle Bei kollisionsfreien Protokollen können, wie der Name schon sagt, keine Kollisionen mehr auftreten. Wir nehmen an, dass es n Stationen gibt, die von 0 bis $n-1$ durchnummeriert sind.

Bitmap-Protokoll Es gibt wieder eine Konkurrenzperiode, die aus genau n Timeslots besteht. Will eine Station i einen Frame senden, so setzt sie im $i$ten Timeslot der Konkurrenzperiode eine 1. Nach der Konkurrenzphase weiß jede Station bescheid, wer senden will und sie können der Reihe nach mit der Übertragung beginnen.

Binary Countdown Da das Bitmap-Protokoll für viele Stationen nicht gut skaliert bzw. viel Overhead durch die feste Anzahl an Slots zur Konkurrenzphase hat, überlegt man sich, dass die Stationen, die senden wollen, ihre Stationsadresse in Binärdarstellung beginnend mit dem höchstwertigen Bit senden. Die Bits an derselben Adressposition der verschiedenen Stationen werden mittels `OR` verknüpft. Sobald nun eine Station erkennt, dass ein höherwertiges Bit (also Station sendet 0 und 1 ist auf Leitung) da ist, bricht sie ab. So geht das dann nach und nach die Bits durch, bis es einen Sieger gibt. Diese Station startet dann die Übertragung.

Implizite Reservierung Ein Fenster besteht aus n Datenslots einer gewissen Länge, die zusammen größer als die RTT sind. Will jetzt eine Station senden, beobachtet sie ein Fenster und markiert sich die Slots mit 0, wenn sie frei sind und mit 1, falls sie belegt sind. Im folgenden Fenster wählt sie zufällig ein Slot aus, der mit 0 markiert wurde, und sendet auf ihm. Falls ein Konflikt auftritt, probiert man es später erneut, ansonsten ist implizit der Slot so lange für die Station reserviert, wie sie Daten sendet.

Adaptive Tree-Walk-Protokoll Man versucht bei *Protokollen mit eingeschränkter Konkurrenz* die Vorteile von Konkurrenzmethoden und kollisionsfreien Methoden zu kombinieren. Dazu teilt man alle Stationen in Gruppen ein, so dass um Slot 1 nur Mitglieder der Gruppe 1 konkurrieren usw. Im Tree-Walk-Protokoll sind die Stationen die Blätter eines Binärbaums und die un-

ter einem Knoten sich befindenden Stationen bilden eine Gruppe. Im ersten konkurrierenden Timeslot nach einer Frameübertragung dürfen alle Stationen versuchen den Kanal zu bekommen. Gelingt es nicht, dürfen nur die Stationen im linken Teilast um den nächsten Slot konkurrieren. Bekommt eine Station den Zuschlag, ist der darauf folgende Slot für die Stationen des rechten Teilbaums reserviert. Bei jeder Kollision wird dann eine Ebene tiefer in den Baum vorgedrungen.

Token Ein Token wird eingeführt, der zyklisch durchgereicht wird. Nur der Besitzer des Tokens darf senden, der Rest nicht.

Wavelength Division Multiple Access Hierbei werden jeder Station zwei Kanäle zugeordnet, ein schmaler Kanal zum Steuern und ein Kanal höherer Bandbreite, der zum Senden von Daten ist. Jede Station hat auch zwei Sender und Empfänger, und zwar einen Empfänger mit fester Wellenlänge zum Abhören des eigenen Steuerkanals (andere Stationen senden darauf Anfragen) und einen einstellbaren Empfänger, der einen Datensender auswählt, den er abhört. Dann einen einstellbaren Sender, um auf die Steuerkanäle anderer Stationen zu senden und einen Sender fester Wellenlänge zur Ausgabe der eigenen Datenströme. Jede Station prüft den eigenen Steuerkanal auf eingehende Anforderungen, dann muss man sich aber auf die Wellenlänge des Senders einstellen, um Daten zu erhalten.

- Eine andere Aufteilungsvariante wäre nach *synchonen* und *asynchronen* Verfahren bzw. *konkurrierend* oder *geregelt*.

3.2 LAN (Ethernet, Token Ring, Token Bus, FDDI)

Ethernet

- Das ursprüngliche Ethernet definiert vier verschiedene Kabeltypen, die alle mit 10 Mbit/s arbeiten. Es kommt jeweils die Basebandübertragung zum Einsatz.

  10Base5 Dickes Koaxialkabel, bei dem zum Verbinden Vampire Taps benutzt werden. Die maximale Segmentlänge beträgt 500m und 100 Geräte pro Segment sind möglich

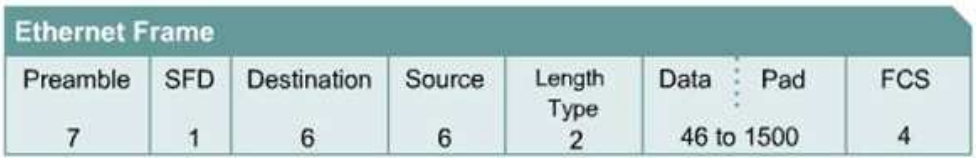
  10Base2 Dünneres Koaxialkabel, bei dem zum Verbinden ein BNC-Stecker verwendet wurde. Die Segmentlänge betrug 185m und man könnte 30 Knoten pro Segment bedienen.

  10Base-T Benutzt ein Twisted-Pair Kabel. Segmentlänge ist 100m aber 1024 Host pro Segment sind möglich. Stationen laufen bei einem Hub zusammen.

  10Base-F Benutzt Glasfaser. Segmentlänge ist 2000m mit 1024 Host pro Segment.

  Man kann das Netz vergrößern, in dem man Repeater zwischen zwei Segmente schaltet, der das Signal verstärkt. Die maximale Reichweite von Ethernet beträgt 2,5km.

- Ethernet verwendet als Leitungskodierung die Manchester-Kodierung mit einer hoch/niedrig-Flanke als 1 und einer niedrig/hoch-Flanke als 0.
- Ein Ethernetframe hat eine Mindestlänge (ohne Präambel) von 64 Byte. Sollte die Daten weniger sein, so wird mittels des *Padding-Felds* aufgefüllt. Die Maximallänge beträgt 1518 Byte (maximal 1500 Byte Daten).
  Mit der *Präambel*, die in jedem Byte die Bitfolge `10101010` enthält, können sich Sender und Empfänger synchronisieren. Das Feld *Start of Frame delimiter* soll die Kompatibilität mit 802.4 und 802.5 gewährleisten. Danach folgen jeweils 48 Bit Adressen von Ziel und Quelle (erstes Bit des Ziels entscheidet über Unicast / Multicast). Das Feld *Length/ Type* definiert entweder die Länge des Datenteils (bei Werten unter 1500) oder den Typ der Daten (Werte über 1500). Über den Typ kann der Empfänger wissen, an welches Vermittlungsschichtprotokoll es übergeben werden soll. Die *Checksum* ist eine Prüfsumme, die aus einem CRC-Code generiert wird. Es wird nur Fehlererkennung unterstützt.



- Ethernet setzt CSMA/CD als Medienzugriffsverfahren ein. Für CSMA/CD ist auch die Mindestlänge von 64 Byte nötig. Und zwar ist bei einer maximalen Entfernung von Sender und Empfänger von 2,5 km die RTT im schlechtesten Fall 50 μs. So lange muss der Sender mindestens ins Medium horchen, um eine Kollision zu erkennen. Da der Sender nur so lange ins Medium hört, wie er auch sendet, muss ein Paket mindestens 50 μs zum Senden brauchen. So hat der Empfänger die Übertragung noch nicht beendet, wenn das Jamming-Signal, das der Empfänger bei einer Kollision sendet, bei ihm eintrifft, und somit seine Übertragung abbricht. Ein Paket von 64 Byte benötigt bei Ethernet genau diese Zeit zum Senden.
- Die Wartezeit nach einer Kollision wird mittels des *binären exponentiellen Backoff-Algorithmus* bestimmt. Hierbei wird die Zeit nach der Kollision in Timeslots unterteilt. Nach der i-ten Kollision wartet die Station zwischen 0 und $2^i - 1$ Timeslots. Bei 10 Kollisionen wird das Intervall eingefroren und bei 16 wird die Übertragung abgebrochen. Mittels des BEB können kurze Verzögerungszeiten gewährleistet werden, falls nur wenige Stationen im Spiel sind, aber bei mehr Stationen auch das Kollisionsproblem aufgelöst werden, in dem hier größere Intervalle benutzt werden.
- Durch die Verwendung von Switches statt Hubs (*Switched Ethernet*) kann die Leistung verbessert werden, da die Kollisionsdomänen verkleinert werden (teilweise auf 1 Peer) und somit Kollisionen verringert bzw. gänzlich vermieden werden. Außerdem kann man teilweise den Vollduplexbetrieb erreichen.

Fast Ethernet

- Mit Fast Ethernet wollte man Ethernet einfach nur schneller machen und eine Übertragungsgeschwindigkeit von 100 Mbit/s bereitstellen. Dazu wurden folgende Kabeltypen definiert:

  100Base-T4 Vier Twisted-Pair Kabel der Kategorie 3 (1 up, 1 down, Rest beliebig zuschaltbar), es wird eine maximale Segmentlänge von 100 m unterstützt. Man benutzt 8B/6T als Leitungskodierung.

  100Base-TX Verwendet zwei Twisted-Pair Kabel der Kategorie 5 (1 up, 1 down) und die Segmentlänge ist auch hier auf 100 m begrenzt. Hier benutzt man 4B/5B zur Leitungskodierung. Dies ist ein Vollduplexsystem.

  100Base-FX Verwendet zwei Multimodefasern zum Vollduplexbetrieb auf einer maximalen Segmentgröße von 2000 m.

- Bei 100Base-T werden Hubs und Switchs unterstützt, 100Base-FX erlaubt nur Switchs. Der Vorteil bei Switchs ist, dass sie Geschwindigkeiten von 10 Mbit/s und 100 Mbit/s zulassen und Halbduplex bzw. Vollduplex erlauben.

Gigabit-Ethernet

- Gigabit-Ethernet unterstützt 1000 Mbit/s und ist wieder abwärtskompatibel. Es unterstützt einen Vollduplex- und einen Halbduplexmodus. Vollduplex ist Standard und wird für Verbindungen zwischen Switch und Rechner (oder anderen Switch) verwendet. Da keine Kollisionen auftreten können, wird CSMA/CD nicht verwendet. Halbduplex wird in Verbindung mit Hubs benutzt, da hier Kollisionen möglich sind, ist CSMA/CD unabdingbar.

- Die minimale Framelänge wurde auf 512 Byte erhöht (bei 64 Byte wäre der Radius auf 25 m beschränkt gewesen aufgrund von den CSMA/CD Anforderungen). Mittels *Carrier Extension* kann ein Frame einfach auf die entsprechende Größe aufgefüllt werden, ohne das die Software davon was merkt. Dazu ist ein *No Data* Feld hinter der Checksum angehangen. Mit *Frame Bursting* wird dem Sender erlaubt, miteinander verknüpfte Folgen von Frames in einer Übertragung zu bündeln (und wenn es immer noch zu kurz ist wieder aufzufüllen).

- Es sind folgende Kabelarten definiert:

  1000Base-SX Multimode-Faser (50 oder 63,5 μm Durchmesser) mit Segmentlänge von bis zu 550 m. Verwendet zur Kodierung 8B/10B.

  1000Base-LX Single (10 μm Durchmesser) oder Multimode-Faser mit maximaler Segmentgröße von 5000 m. Verwendet zur Kodierung 8B/10B.

  1000Base-CX 2 paar abgeschirmte Twisted-Pair Kabel mit Segmentlänge von 25 m

  1000Base-T Vier Kategorie 5 Twisted-Pair Kabel mit einer maximalen Segmentlänge von 100 m. Hier werden fünf Signalstufen zur Kodierung verwendet (vier zur Kodierung von 2 Bit und 1 für Steuerzwecke).

- Gigabit-Ethernet unterstützt eine Art von Flusskontrolle, so dass der Empfänger einen Steuerrahmen an den Sender schicken kann, um ihm mitzuteilen, das er eine Sendepause einlegen soll.

Token Bus

- Über ein Bus wird ein logischer Ring definiert. In dem Ring kreist ein Token, der der besitzenden Station das exklusive Senderecht verleiht.
- Es werden zwei Nachrichtentypen unterschieden: *Token* und *Daten*. Der Token enthält die ID des Senders und die ID desjenigen, der den Token bekommt.
- Token Bus unterstützt vier *Prioritäten*.
- Wenn eine Station den Ring verlässt, muss sie ihren Vorgänger im Ring über den neuen Nachfolger informieren. Um neue Stationen in den Ring aufzunehmen, öffnet sich von Zeit zu Zeit ein Fenster zwischen zwei Nachbarn, so dass Stationen mit entsprechenden IDs dem Ring beitreten können. Bei einem Konflikt wird ein Binary Countdown angewandt, um die Station zu ermitteln, die beitreten darf.

Token Ring

- Die Stationen sind punktweise zu einem Ring verbunden. Es gibt wieder einen Token (oder mehrere), der dem Besitzer das Senden von Nachrichten erlaubt. Charakteristiken sind garantierter Zugriff, keine Kollisionen und Fairness. Unter geringer Last ist es aber teilweise ineffizient, da eine Station auf den Token warten muss.
- Eine sendebereite Station muss auf den Token warten. Hat sie ihn bekommen, wandelt sie ihn in einen occupied-Token um und sendet den Frame. Die Station wartet so lange, bis der Frame wieder ankommt und löscht ihn dann vom Ring (der Empfänger kopiert natürlich auch vorher den Frame). Gleichzeitig wird dann von der Station ein neuer freier Token erzeugt.
- Bei mehreren Token läuft die Prozedur identisch ab, bis darauf, dass die sendende Station nicht wartet, bis sein Frame einmal rum ist und dann einen neuen Token erzeugt, sondern sofort nach dem Sendevorgang.
- Mit einem *Wire Center* kann die Robustheit erhöht werden. Jede Station ist mit dem Wire Center verbunden. Das Center enthält Bypass-Switches, die intern eine Verbindung herstellen, sobald eine Station nicht antwortet.
- Zur Ringverwaltung wird eine Monitor-Station eingeführt. Sie hat die Aufgabe nach einem Tokenverlust einen neuen Token zu generieren, bei Zusammenbrüchen des Rings zu reagieren, Framefragmente zu entfernen und veraltete noch zirkulierende Frames zu löschen.
- Token Ring benutzt die differentielle Manchesterkodierung. Im Datenteil kann wie bei Ethernet ein 802.2 LLC Paket stecken.

FDDI (Fiber Distributed Data Interface)

- FDDI ist ein hoch performanter Token Ring auf Glasfaserkabeln. Er unterstützt Datenraten bis 100 Mbit/s und kann eine Reichweite bis 200 km haben und 1000 Stationen unterstützen (maximal 2 km Entfernung zwischen zwei). Es kommen mehrere Tokens zum Einsatz.
- Die Struktur besteht aus zwei entgegengesetzt gerichteten Glasfaserringen. Dabei ist ein Ring der Primärring, der Sekundärring ist dazu gedacht, bei einem Verbindungsabbruch auf diesen umzuswitchen. Wenn beide Ringe an einer Stelle brechen, können beide kombiniert werden zu einem Ring doppelte Länge.
- Als Kodierung wird 4B/5B verwendet. Zur Synchronisation der Stationen wird eine lange Präambel verwendet

3.3 MAN (DQDB)

- Ein *MAN* (Metropolitan Area Network) überbrückt größere Distanzen als ein LAN. Weiterhin benutzt es einen Clock Pulse.
- *DQDB* (Distributed Queue Dual Bus) ist ein MAN, dass als Struktur zwei unidirektionale Busse verwendet, an die alle Computer angeschlossen werden. Dabei ist ein Bus jeweils für die Kommunikation in eine Richtung zuständig. An jedem Bus ist ein Kopfende, dass die Übertragungen kontrolliert; dazu werden Slots der Größe von 53 Byte alle 125 μs erzeugt. Das Netz kann sich über 100 km erstrecken und unterstützt Datenraten bis 150 Mbit/s (Glasfaser) oder 44 Mbit/s (Koaxialkabel).
- Bevor eine Station bei DQDB senden kann, muss sie einen Slot bekommen. Dazu wird eine Reservierung entgegen der Senderichtung geschickt. Es wird dann verteilt eine Warteschlange aufgebaut, wobei sich jede Station merkt, wie viele Stationen sich als sendebereit gemeldet haben und wie viele Pakete durchgelassen werden müssen, bevor man selber an der Reihe ist.

3.4 WAN (ATM, SDH, Frame Relay)

*WAN*s (Wide Area Network) sind noch größer in der Ausdehnung als MANs, bis mehrere 1000 km können abgedeckt werden. Es ist ein Netz aus Routern, die über Punkt-zu-Punkt-Verbindungen verbunden sind. WANs können paketvermittelt oder leitungsvermittelt sein.

Frame Relay

Frame Relay verfolgt einen paketvermittelten Ansatz bei dem die Frames eine variable Länge haben können. Es wird ein statistisches Multiplexing verwendet um die verschiedenen Datenströme zu mischen. Es werden Datenraten zwischen 56 kbit/s und 45 Mbit/s unterstützt. Frame Relay bietet verbindungsorientierte Dienste auf der

LLC-Schicht an, besitzt allerdings keine Flusskontrolle (nur Mechanismen um höhere Schichten über Bottlenecks o.ä. zu informieren).

ATM (Asynchronous Transfer Mode)

- ATM kombiniert die Vorteile von leitungsvermittelten und paketvermittelten Technologien und stellt Garantien bzgl. Kapazität und Delay sowie flexible und effiziente Übertragungen zu Verfügung. ATM ist verbindungsorientiert und baut somit virtuelle Verbindungen auf. Es hat wie Frame Relay auch keine Flusskontrolle und auch keine Fehlerbehandlung. Es werden Datenraten von 155 Mbit/s und 622 Mbit/s unterstützt.
- ATM benutzt Zellen fester Größe (53 Byte), die dann ähnlich wie bei TDM gemultiplext werden. Dadurch erhält man ein asynchrones Zeitmultiplexing mehrere virtueller Verbindungen, die einen kontinuierlichen Zellstrom ergeben. Unbenutzte Zellen werden leer gesendet.
- Ein ATM Netzwerk besteht aus ATM Switchs und ATM Endpunkten.
- Für Verbindungen werden keine Adressen benötigt sonder lediglich ein *Virtual Connection Identifier*. Beim Verbindungsaufbau erfragt der Sender bei einem ATM Switch einen Verbindungsaufbau. Der Switch baut eine virtuelle Verbindung zum nächsten Switch auf der Route auf (vergibt einen Connection Identifier) und leitet den Request weiter. Kommt der Request beim Ziel an, so sendet dieser den nun hergestellten Pfad zurück und bestätigt die Verbindung. Dann können Daten übermittelt werden, die mit der Connection Identifier adressiert werden können.
- ATM definiert einen eigenen Protokollstapel und ist somit nicht von haus aus mit ISO/OSI kompatible. Um es attraktiver zu machen, gibt es aber Bindungen für IP über ATM.
- ATM definiert mehrere Service Klassen und kann so QoS (Quality of Service) Garantien geben.

SDH (Synchronous Digital Hierarchy)

- SDH oder auch SONET ist eine hierarchische Multiplex-Struktur, bei der Leitungen geringerer Datenrate auf Leitungen höherer Datenrate (bis zu 40 Gbit/s) hierarchisch gemultiplext werden (Byte für Byte). Die Bitraten sind weltweit standardisiert, die auf den Hierarchieebenen zum Einsatz kommen.
- SDH ist zentral getaktet (daher auch der Name synchron). Daher müssen alle Elemente synchronisiert sein, die Synchronisierung erfolgt über den gleichen Pfad wie die Datenübertragung.

- Als *STM* (Synchronous Transport Module) werden die verwendeten Pakete bezeichnet. Der Payload wird in einen Container gepackt (je nach Größe C-1 bis C-4). Wenn mehrere Container in einem STM stecken, werden diese zu einer TUG (Tributary Unit Group) gemultiplext. Vier STM-1 können zu einem STM-4 Paket gemultiplext werden usw. (jeweils dann mit höheren Datenraten).

3.5 Netzwerkkomponenten und -strukturen

- Es gibt viele Netzwerkstrukturkomponenten, die die unterschiedlichsten Aufgaben erledigen:

  Repeater Sie arbeiten auf der Bitübertragungsschicht und dienen dazu ein eingehendes analoges Signal einfach zu verstärken und wieder auszugeben. Damit können sie die Reichweite erhöhen.

  Hub Ein Hub verfügt über mehrere Eingangsleitungen, die elektronisch verbunden werden. Ein eingehendes Signal wird auf allen übrigen Leitungen versendet. Wenn von mehreren Leitungen gleichzeitig Signale eintreffen, kollidieren sie (Hub bildet eine Kollisionsdomäne). Ein Hub kann nur Leitungen gleicher Geschwindigkeit unterstützen.

  Bridge/ Switch Bridge und Switch wird hier als Synonym verwendet. Sie arbeiten auf der Sicherungsschicht. Eine Bridge verbindet zwei oder mehrere LANs (auch unterschiedlichen Typs) transparent. Dazu wird in der Bridge der MAC-Header des eingehenden Frames entfernt und der MAC-Header des Zielnetzes angefügt. Man muss aufpassen, da die LANs nicht unbedingt dieselbe Geschwindigkeit unterstützen oder das gleiche Frameformat haben. Auch kann z.B. nicht unbedingt immer die Dienstgüte beibehalten werden oder die Sicherheitsaspekte.
  Eine Bridge muss wissen, wie sie ein Frame weiterleiten soll (vor allem, an welches Zielnetz). Dazu verwaltet die Bridge ein Art Routing-Tabelle. Weiß die Bridge nicht, wo der Empfänger sitzt, so flutet sie den Frame in alle anderen Netze. Lernen tut die Bridge darüber, dass sie sich die Quelladresse eines eingehenden Frames anschaut und so weiß, über welchen Port der Sender zu erreichen ist. Das Tupel (Adresse, Port, Alter) wird gespeichert. Ist ein Tupel zu alt, wird es gelöscht. Das Routing verläuft dann wie folgt: Sind Sender und Empfänger im selben LAN, so leitet die Bridge das Paket nicht weiter. Ansonsten wird in der Tabelle nach einem entsprechenden Eintrag gesucht und entweder direkt das richtige LAN adressiert oder Flooding benutzt.
  Verbinden mehrere Bridges dieselben LANs, so kann es hier zu Loops kommen. Damit Frames nicht anfangen zu kreisen, wird ein *Spanning Tree* der Bridges aufgebaut. Die Wurzel des Baums bildet die Bridge mit der kleinsten ID. Am Anfang denken alle Bridges, sie wären die Root-Bridge und jeder sendet ein Frame (RootID, Kosten, Bridge ID, Port ID) aus. Kommt der Frame bei einer Bridge an, die eine kleinere ID hat, so ändert sie die Einträge für Root ID und Kosten ab. Root ID wird ihre eigene Adresse und

zu den vorhandenen Kosten werden die eigenen Kosten addiert. Sind alle upgedateten Frames bei allen eingetroffen, so kennt jeder die Root Bridge inkl. die Kosten dorthin und den zu verwendenden Pfad.
Bei Switches bildet jeder Port eine Kollisionsdomäne, so dass es zu keinem Frameverlust durch Kollisionen kommen kann.

Router Bridges unterstützen nur wenige tausend Stationen und Bridges können nicht mit den Hosts kommunizieren, zudem senden sie Broadcasts in alle angrenzenden LANs weiter (Broadcast Storm). Um diese Probleme zu überwinden, benötigt man Router, die auf der Vermittlungsschicht arbeiten. Sie routen Pakete anhand von den IP Adressen auf dem besten Weg zum Ziel, lassen keine Broadcast durch (Multicast bedingt) und verbessern die Performance indem Router und Host über Nachrichten kommunizieren.

Gateway Gateways kann man auf den oberen Schichten ansiedeln. Auf der Transportschicht können sie verschiedene Transportprotokolle auf den verschiedenen Host unterstützen und konvertieren (z.B. TCP/IP an ATM). Auf der Anwendungsschicht können Gateways das Format und die Inhalte von Daten in das Format anderer Anwendungen übertragen (z.B. SMS in Email).

3.6 VLAN

- Aus verschiedenen Gründen (Sicherheit, Lastverteilung, Broadcasting) ist es sinnvoll die physikalische LAN Struktur von der logische Struktur zu trennen. Dazu verwendet man *VLAN* (Virtual LAN).

- VLAN braucht Switchs, die VLAN erkennen. Dazu benötigen sie spezielle Tabellen, in denen vermerkt ist, über welchen Port welche VLANs erreicht werden. Es gibt verschiedene Methoden, wie ein Switch wissen kann, wo welche VLANs sind:

  1. Jedem Port wird eine VLAN-ID zugeordnet. Dies funktioniert nur, wenn alle an dem Port angeschlossenen Rechner im selben VLAN sind.
  2. Jeder MAC-Adresse wird einer VLAN-ID zugeordnet. In einer Tabelle steht dann für jeden angeschlossenen Rechner der VLAN-Tag und die MAC-Adresse.
  3. Jedes Vermittlungsschichtprotokoll bzw. IP Adresse wird einem VLAN zugeordnet. Dies ist praktisch, wenn viele mobile Geräte verwendet werden, die an unterschiedlichen Dockingstations benutzt werden können, an denen jeweils eine unterschiedliche MAC-Adresse herrscht. Dies verletzt aber die Unabhängigkeit der Schichten!

- In IEEE 802.1Q ist ein Ethernetframe mit zusätzlichem VLAN-Tag-Feld beschrieben. Damit man nicht alle alten Hardwarekomponenten, die diesen Standard noch nicht unterstützen, austauschen muss, führt man ein, dass das erste VLAN-fähige Gerät diesen Tag einfügt und das letzte noch VLAN-fähige es

wieder entfernt. Das erste Gerät muss das VLAN dann über die MAC Adresse oder eine andere oben genannte Methode bestimmen. Switches können die VLAN-Tabellen übrigens auch dynamisch Konfigurieren, in dem sie wieder bei eingehenden Frames prüfen, an welches VLAN ein Frame gesendet wird. Der Absender ist höchstwahrscheinlich im selben VLAN.

4 Vermittlungsschicht

Die Vermittlungsschicht hat die Aufgabe, Pakete von der Quelle zum Ziel zu befördern, was das Durchqueren vieler Teilstrecken umfasst (mittels Routing). Sie ist somit die erste Schicht, die sich mit der Übertragung von Endpunkt zu Endpunkt befasst.
Die Vermittlungsschicht stellt der Transportschicht Dienste in Form von verbindungslosen oder verbindungsorientierten Diensten zu Verfügung. Bei dem verbindungslosen Dienst wird jedes Paket einzeln in das Netz eingespeist und wird unabhängig von den anderen (auf eventuell unterschiedlichen Routen) zum Ziel befördert. Im Gegensatz dazu muss bei einem verbindungsorientierten Dienst zwischen Quelle und Ziel eine virtuelle Verbindung aufgebaut werden, so dass für alle Pakete, die über diese Verbindung gehen, nur einmal eine Route festgelegt wird, nämlich beim Verbindungsaufbau. Die Route wird dann in Form von Tabellen in den Routern verwaltet (über Verbindungskennungen). Somit ist die Reihenfolge der Pakete garantiert und es ist möglich im begrenzten Umfang QoS-Anforderungen zu stellen, da im Voraus z.B. Bandbreite reserviert werden kann. Allerdings sind virtuelle Verbindungen dahin gehend empfindlich, dass bei einem Routerausfall alle Verbindungen, die über diesen Router liefen, gekappt werden.
Auf der Schicht wird auch *Überlastungsüberwachung* (Congestion Control)betrieben. Man spricht von *Überlastung*, wenn in einem Verbindungsnetz zu viele Pakte vorhanden sind, so dass die Leistung erheblich abfällt. Im Gegensatz zur Flusskontrolle bezieht sich die Überlastungsüberwachung auf das gesamte Teilnetz, in dem sichergestellt werden muss, dass der anfallende Verkehr zu bewältigen ist. Flusskontrolle bezieht sich nur auf Punkt-zu-Punkt-Verkehr. Es können *Choke-Pakete* eingesetzt werden, die von einem Router zurück zur Quelle gesendet werden, und dort mitteilen, dass weniger gesendet werden soll (Reduktion um X Prozent). Eine Abwandlung ist, dass alle auf dem Weg vom Router zum Ziel liegenden Router ebenfalls beim Weiterreichen des Choke-Pakets ihren Datenfluss reduzieren.

4.1 Routing

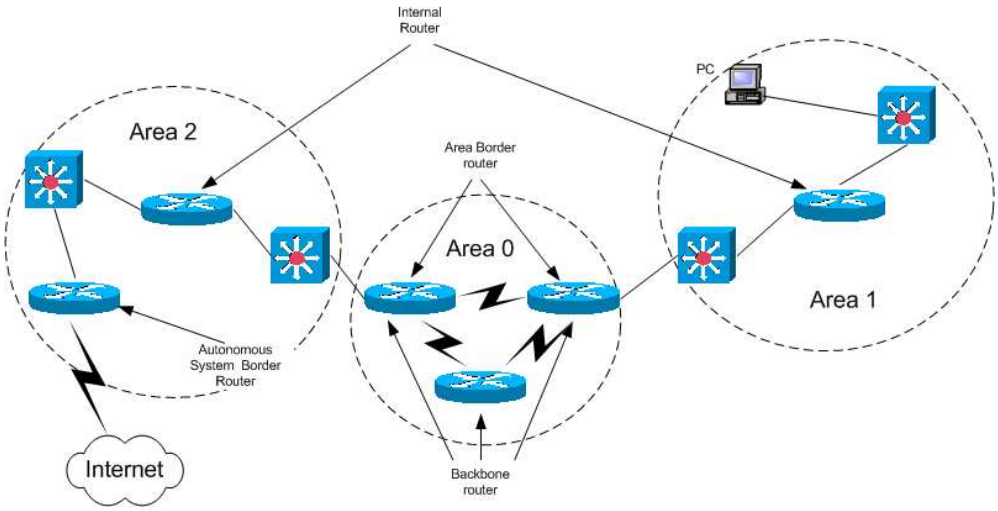
- Ein Routing-Algorithmus sollte einfach, robust, stabil, fair und optimal sein (es gilt zu beachten, dass Fairness und Optimalität oft widersprüchlich sind). Zudem soll eine Änderung der Topologie oder des Datenverkehrs kein Problem darstellen. Ein gute Heuristik ist, die Anzahl der Teilstrecken, die ein Paket durchlaufen muss, zu minimieren.

- Man unterscheidet zwei Hauptgruppen von Routing-Algorithmen:

Nichtadaptive Algorithmen Hier wird vorab festgelegt, welchen Pfad ein Paket nehmen soll. Der Algorithmus greift also nicht auf Messungen des aktuellen Netzstatus zurück, um seine Routingentscheidungen anzupassen. Dies nennt man auch *statisches Routing*. Ein Beispiel wäre *Source Routing*.

Adaptive Algorithmen Die Routingentscheidungen basieren hier auf Änderungen, die in der Topologie aufgetreten sind, oder aber im Verkehr. Es gibt verschiedene Ansätze, woher der Router seine Informationen bekommt (nur lokal, von andern Routern ...), zu welchem Zeitpunkt sich die Pfade ändern oder welche Metriken verwendet werden. Dies nennt man auch *dynamisches Routing*.

- Das Internet ist hierarchisch angeordnet und man unterscheidet deshalb *Autonome Systeme* und *Bereiche*. Autonome Systeme sind in mehrere Bereich eingeteilt. Jedes autonome System hat einen Backbone-Bereich, mit dem alle Bereiche verbunden sind. Jeder Router, der an mehrere Bereiche angeschlossen ist, gehört zu dem Backbone-Bereich (Bereich 0) und wird als Grenz-Router bezeichnet. Dementsprechend sind Backbone-Router, solche Router, die sich im Backbone befinden (nicht unbedingt auch einem Bereich zugeordnet), interne Router befinden sich vollständig in einem Bereich und AS-Grenz-Router vermitteln zwischen mehreren autonomen Systemen.



- Innerhalb eines autonomen Systems wird ein *Interior Gateway Protocol* eingesetzt, zwischen mehreren autonomen Systemen verwendet man *Exterior Gateway Protocols*. Im Gegensatz zu Interior Gateway Protocols, die auf Effizient angelegt sind, müssen die Exterior Gateway Protocols oft Policies (politische, ökonomische usw.) befolgen.
- Folgende Algorithmen gibt es:

Flooding Flooding ist ein statischer Algorithmus bei dem jedes eingehende Paket auf allen Ausgangsleitungen des Routers verteilt wird, außer auf der, von der es einging. Zur Eindämmung der Paketflut führt man einen Hopcounter ein, der bei jedem Hop um eins reduziert wird und bei 0 dann das Paket verwerfen lässt. Man kann Flooding noch dahin gehend verbessern, dass man die Ausgangsleitungen irgendwie selektiert, so dass das Paket näherungsweise in die richtige Richtung geroutet wird. Der Vorteil von Flooding ist die extreme Robustheit.

Hot Potato Es wird immer der Ausgang gewählt um ein Paket weiterzuleiten, auf dem die geringste Last ist. Problem ist das zirkulieren von Nachrichten, was man aber dadurch beheben kann, dass man eine Liste der schon besuchten Router mitreicht.

Probabilistic Routing Aufgrund von Leistungsmessungen von bereits gesendeten Paketen wird die beste Route berechnet. Dazu wird jedem Port ein Proportionalitätsfaktor zugeordnet.

Shortest Path Routing Das Netz wird als Graph dargestellt und es wird nun zwischen zwei Knoten der kürzeste Pfad gesucht. Dazu kann man den Algorithmus von Dijkstra verwenden:

1. Markiere den Arbeitsknoten als permanent.
2. Die Nachbarn des als permanent markierten Arbeitsknotens werden betrachtet und die Kanten mit der Entfernung zur Quelle basierend aus dem schon gesammelten Wissen und der Kantengewichte beschriftet.
3. Aus allen beschrifteten Knoten wähle den aus mit der geringsten Entfernung und setze ihn als Arbeitsknoten.

Shortest Path Routing kann man statisch (feste Entfernungen) oder dynamisch (Entfernungen können sich ändern, Updates von Routing-Tabellen sind nötig) machen.

Distance Vector Routing Dies ist einer der häufigsten verwendeten dynamischen Algorithmen. Hier verwaltet jeder Router eine Tabelle, die für jedes Ziel die bestmöglich bekannte Entfernung und die zu verwendende Leitung erhält.
Die Tabellen werden durch Nachrichtenaustausch mit den benachbarten Routern aktuell gehalten. Es wird angenommen, dass jeder Router die Entfernungen zu seinen Nachbarn kennt. In bestimmten Zeitintervallen sendet nun jeder Router an alle seine Nachbarn eine Liste mit den geschätzten Kosten zu allen möglichen Zielen. Wenn ein Router so eine Liste von seinen Nachbarn empfängt, kann er aus den darin enthaltenen Kosten und den bekannten Kosten zu dem entsprechenden Nachbarn sich daraus eine neue Tabelle zusammenbasteln. Es werden globale Informationen hierbei also nur unter Nachbarn ausgetauscht.
Das Hauptproblem beim Distance Vector Routing ist das, dass es zu träge auf schlechte Nachrichten reagiert. Das resultiert darin, dass bei einem

Linkbruch zwischen zwei Stationen (und es noch weitere Wege zwischen beiden gibt), diese sich langsam hochschaukeln, bis sie dann wieder zu einem stabilen Stand kommen. Im Extremfall kommt es zum *Count to Infinity* Problem, bei dem die Stationen ins unendliche zählen, da sie bei einem abgeschnittenen Host von ihren Nachbarn erfahren, dass diese sie noch erreichen können (und man nicht weiß, dass deren Route über einen selber läuft und dies auch nicht feststellen kann).
Ein mögliches Lösungsverfahren ist *Split-Horizon* nach dem die Routinginformationen über ein bestimmtes Netzwerk an alle Leitungen weitergegeben werden bis auf die Leitung, von wo man die Routinginformation bekommen hat.
Das Interior Gateway Protocol *Routing Information Protocol* (RIP) basiert auf diesem Verfahren. Es verschickt alle 30 Sekunden als UDP-Pakete die Nachrichten und benutzt als Metrik die Anzahl an Hops (16 entspricht Unendlich). In einer Nachricht können nur 25 Tabelleneinträge übermittelt werden.

Link State Routing Hier tauschen Router lokale Informationen (also über ihre Nachbarn) mit dem ganzen Netzwerk aus. Dazu muss jeder Router zunächst seine *Nachbarn ermitteln*. Dazu versendet er spezielle `HELLO`-Pakete auf jeder Punkt-zu-Punkt-Leitung. Router sind angewiesen auf solch eine Nachricht mit ihrer Identität zu antworten.
Genau wie beim DVR muss ein Router wissen, wie die *Übertragungszeit zu den Nachbarn* ist. Um diese gut ermitteln zu können, werden `ECHO`-Pakete verwendet, auf die sofort geantwortet werden muss. Durch Stoppen kann so der Sender die RTT messen (man geht implizit davon aus, dass die Übertragungszeiten symmetrisch sind). Wenn man den Timer zu dem Zeitpunkt startet, zu dem das `ECHO`-Paket in die Warteschlange gestellt wird, dann kann man die Last mitberücksichtigen (kann allerdings zu zyklischer Lastenverteilung führen). Wenn der Timer startet, sobald das Paket versendet wird, ist tatsächlich nur die Übertragungszeit berücksichtigt.
Hat der Router diese Informationen von seinen Nachbarn beisammen, kann er sie in *Link-State-Pakete packen*, die die Identität des Absenders (alle Router sind eindeutig benannt), eine Sequenznummer und das Alter sowie eine Liste der Nachbarn mit den gemessenen Zeiten enthält. Man kann diese Pakete entweder in regelmäßigen Abständen erstellen oder nur falls sich die Nachbarschaft ändert.
Die *Verteilung* der Pakete erfolgt durch Flooding. Die Router merken sich von eingehenden Link-State-Paketen die Quelladresse, Sequenznummer und natürlich die Nachbarschaftsinformationen. Bei jedem neuen Paket, das gesendet wird, wird die Sequenznummer erhöht, so dass ein Router schauen kann, ob es ein veraltetes Paket ist o.ä.. Ist die Sequenznummer höher als die aktuell gespeicherte, so ersetzt er seine gespeicherten Informationen und reicht das Paket an alle Ausgangsleitungen (bis auf der, wo es eingegangen ist) weiter. Ist sie dieselbe, so ist es ein Duplikat und das Paket wird ver-

worfen, genauso wenn die Sequenznummer kleiner ist und somit das Paket veraltet. Die Sequenznummern sind 32 Bit lang, so dass man kein Problem mit sich wiederholenden Nummern hat. Zusätzlich wird das Feld Alter des Pakets im Router jede Sekunde um eins reduziert, erreicht es 0, so wird das Paket verworfen (und beim Flooding-Prozess nicht weitergeleitet). So verhindert man bei Bitfehlern in der Sequenznummer, dass viele Pakete verworfen werden, da gedacht wird, sie wäre veraltet oder auch, dass bei einem Ausfall eines Routers es Schwierigkeiten gibt, wenn er wieder bei Sequenznummer 0 anfängt zu zählen. Die Link-State-Pakete werden dem Sender bestätigt.
Hat ein Router eine vollständige Menge von Link-State-Paketen erhalten, so kann er einen *Graphen des Teilnetzes aufbauen* und darauf lokal den Dijkstra-Algorithmus laufen lassen und die Ergebnisse in die Routing-Tabelle eintragen.

Hierarchisches Routing Da bei zunehmender Größe des Netzes die Router immer größere Tabellen verwalten müssen und der ganze Informationsaustausch immer mehr Bandbreite benötigt, teilt man beim hierarchischen Routing die Router in *Regionen* ein. Jeder Router weiß dann über die Struktur seiner Region bescheid, alles darüber hinaus weiß er aber nichts. Natürlich muss jeder Router innerhalb der Region wissen, wie er zu anderen Regionen kommt (ein Router ist dann z.B. für den Verkehr in eine andere Region zuständig). Dadurch verkürzen sich die Routing-Tabellen enorm, allerdings steigen die Pfadlängen an; dies kann aber vernachlässigt werden. Bei n Routern liegt die optimale Anzahl an Ebenen bei $\ln n$ und pro Router sind insgesamt $e \ln n$ Einträge nötig.

OSPF (Open Shortest Path First) OSPF ist ein Interior Gateway Protocol, welches RIP absetzt. Es basiert auf dem Link State Routing und unterstützt eine Vielzahl von Entfernungsmetriken (physikalische Entfernung, Übertragungsverzögerung usw.), bietet Routing auf der Grundlage von Diensttypen und, Lastenausgleich sowie gewisse Sicherheitsmaßnahmen. In OSPF werden die verschiedenen oben genannten Routertypen unterschieden (also Hierarchien). Ansonsten funktioniert es ähnlich wie Link State. Es werden `HELLO`-Pakete gesendet um seine Nachbarschaft zu erkunden und die Informationen eines Senders werden regelmäßig oder bei Änderung an einer Leitung mit `LINK STATE UPDATE`-Paketen gesendet, die den Status des Senders und die Kosten in seiner Datenbank angeben (sind durchnummeriert und werden bestätigt). Mit `DATABASE DESCRIPTION` kann der Sender die Sequenznummern seiner eigenen Datenbankeinträge bekannt geben und ein Empfänger kann dann vergleichen, wer die aktuelleren Werte hat. Diese Pakete werden gesendet, wenn eine Leitung zugeschaltet wird. Über `LINK STATE REQUEST` können auch Verbindungszustandsinformationen angefordert werden. OSPF verwendet IP Pakete für seine Nachrichten.

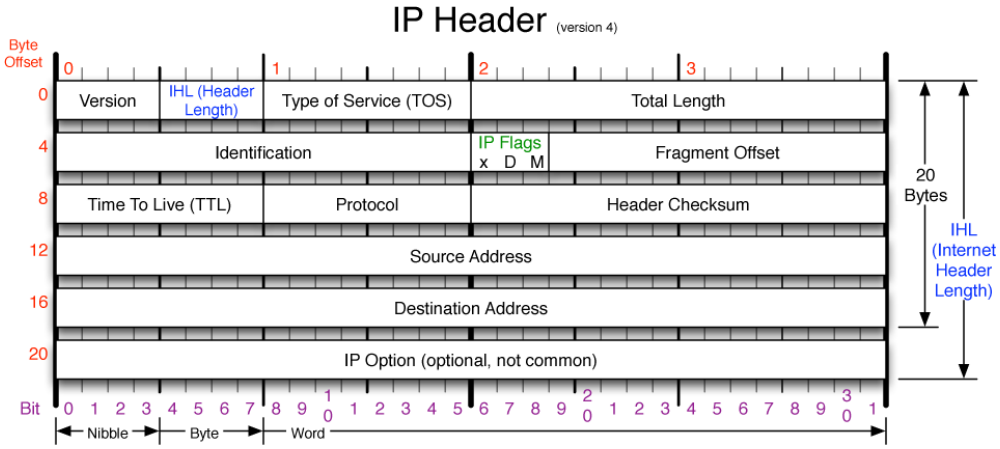
BGP (Border Gateway Protocol) BGP ist ein Exterior Gateway Protocol und

vermittelt zwischen mehreren autonomen Systemen. Es ist im Grunde ein Distance Vector Routing, dass aber anstelle der Kosten eines Pfades die gesamte Pfadbeschreibung austauscht (so auch kein Count to Infinity Problem!). Aus Sicht eines BGP-Routers besteht das ganze Netz nur aus anderen autonomen Systemen und den verbindenden Leitungen. Zum Austausch der Daten zwischen den Routern werden TCP Verbindungen aufgebaut.

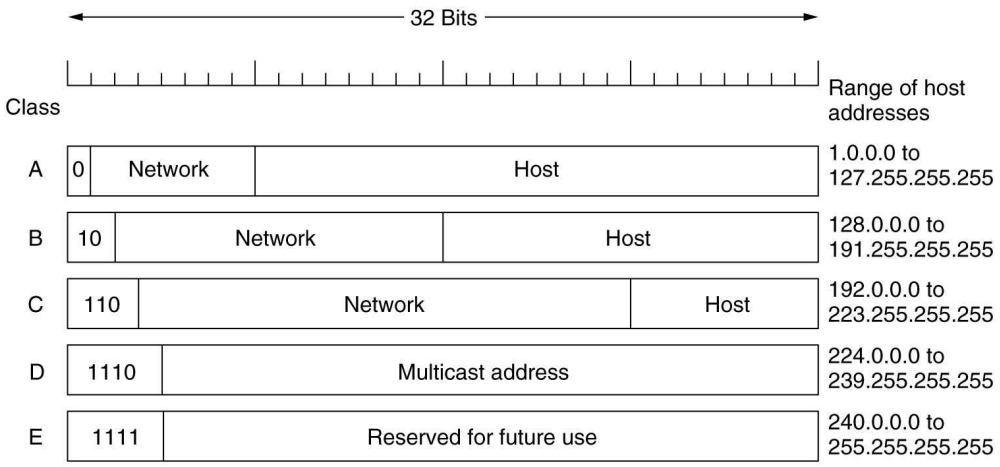
4.2 IP

- Das *Internet Protocol* (IP) bietet einen verbindungslosen, unzuverlässigen Übertragungsdienst von Datagrammen bzw. Paketen an. Es arbeitet also nach „Best Effort“. Durch IP ist eine transparente Ende-zu-Ende Kommunikation zwischen Hosts möglich. Mittels Adressierung über die IP-Adresse können die Host angesprochen werden. Da man den Adressraum in Subnetze einteilen kann, kann man auch logisch Gruppieren. Die IP-Adressen sind auch die Grundlage für das Routing.

- Ein IP-Paket kann maximal 64 kByte groß sein (im Einsatz meistens allerdings nur 1500 Byte, damit das noch in einen Frame passt). Ein Paket besteht aus 20 Byte Header und einem optionalen Teil. Dann folgen die Daten.
  Version gibt die IP Version an (4 oder 6). *IHL* gibt die Länge des Headers in 32 Bit Wörtern angegeben (also 5 ohne Optionen, maximal 15, was einem optionalen Header der Größe von 40 Byte entspricht). Bei *Type of Service* können verschiedene Dienstklassen unterschieden werden. Man kann in gewissem Maße Zuverlässigkeit und Geschwindigkeit einstellen. *Total Length* gibt die Gesamtlänge des Pakets an, inkl. Header. Über *Identification* wird dem Zielhost mitgeteilt, zu welchem Datagramm ein Fragment gehört, denn jedes Fragment eines Datagramms enthält dort den gleichen Wert. Mit dem Flag *DF* wird den Routern angewiesen, dieses Datagramm nicht zu fragmentieren (alle Komponenten müssen in der Lage sein ein Paket der Größe von 567 Byte zu verarbeiten). Über *MF* wird angegeben, dass weitere Fragmente folgen. Dieses Bit ist bei allen Fragmenten außer dem letzten gesetzt. Mit dem *Frame Offset* wird angegeben, an welche Stelle im Datagramm ein Fragment gehört. *TTL* ist ein Zähler, mit dem die Lebensdauer eines Pakets begrenzt wird. Bei jedem Hop wird er dekrementiert. Erreicht es 0, wird das Paket verworfen. Durch das Feld *Protocol* wird angegeben, an welchen Prozess das Paket übergeben werden soll. Die *Header Checksum* prüft nur den Header nach Fehlern, die durch Router entstanden sind. Nach jedem erniedrigen der TTL muss sie neu berechnet werden. Dann folgen die beiden Felder für die 32-Bit Adressen von Quelle und Ziel. Es gibt verschiedene Optionen, die Sicherheit, exaktes oder freies Routing, Wegaufzeichnung usw. unterstützen.



- Jeder Host und Router im Internet hat eine *IP-Adresse*, die seine Netz- und Hostnummer kodiert. Diese Kombination ist eindeutig.
- Man hat am Anfang alle IP Adressen einer von fünf Klassen (A, B, C, D, E) zugewiesen. Dies nennt man *Classful Addressing*. Dabei hat man jeweils einen bestimmten Präfix der 32 Bit für die Adressierung des Netzes verwendet und die übrigen Bit für die des Hosts im Netz. Man hat somit eine gewisse Hierarchie. Das Schema ist wie folgt definiert:



Es gibt noch ein paar reservierte Adressen: Überall 0en steht für den Host, wenn der Netzteil nur 0en enthält, dann wird ein Host in dem Netzwerk angesprochen. Überall 1en steht für den lokalen Broadcast und eine Netzadresse und 1en im Hostteil steht für einen Broadcast in dem angegebenen Netz. Wenn vorne die 127 steht, ist ein Loopback gemeint, d.h. Pakete werden nicht auf die Leitung ausgegeben sondern direkt als Eingabe behandelt.

- Zur besseren Verwaltung hat man eingeführt, dass man ein Netz für interne Verwendungszecke in mehrere Teile, die so genannten *Subnets*, aufteilen kann, die nach außen aber wie ein Netz dargestellt werden. Normalerweise hat jedes Subnet seinen eigenen Router, die alle an einen Hauptrouter angeschlossen sind. Um jetzt geschickter zu adressieren, nimmt man einfach ein paar Bit des Hostteils und funktioniert diese als Identifikator des Subnet um. Eine *Subnetmask* gibt durch die Markierung mit 1en das Präfix der Adresse an, welches als Netzteil (inkl. Subnet) gehandhabt werden soll. Router müssen diese Subnetmask kennen. Router verwalten demnach dann Tabellen mit (Dieses Netz, Teilnetz, 0) Adressen (für andere Teilnetze) und (dieses Netz, dieses Teilnetz, Host) Adressen für Host im eigenen Teilnetz. Durch Einführung der Teilnetze wird aus der zweistufigen Hierarchie eine dreistufige.

- Da IP-Adressen knapp werden und man durch das Klassenkonzept ziemlich restriktiv und verschwenderisch ist, hat man das *CIDR* (Classless Interdomain Routing) eingeführt. Hierbei will man die verbleibenden IP-Adressen in Blöcken variabler Länge unabhängig der Klassen vergeben. Wie beim Subnet dient hier eine Maske zur Kennzeichnung des Netzbereichs. Dies macht allerdings die Weiterleitung komplizierter. Die Routing-Tabelle für alle Netze wird um eine 32-Bit-Maske erweitert, so dass die Einträge die Form (IP-Adresse, Teilnetzmaske, Ausgangsleitung) haben. Kommt ein Paket an, wird die Adresse extrahiert und die Tabelle Eintrag für Eintrag abgetastet. Dabei wird die Zieladresse mit der Teilnetzmaske verundet und geprüft, ob die resultierende Adresse zu dem aktuellen Eintrag passt. Passen mehrere Einträge, so wird derjenige mit der längsten Maske verwendet und an dessen Leitung das Paket geschickt.

- Eine weitere Methode, um der Adressknappheit entgegenzuwirken ist *NAT* (Network Address Translation). Hierbei bekommen Hosts innerhalb des Intranets eindeutige Adressen aus dafür reservierten privaten Adressbereichen vergeben. Soll eine Verbindung nach außen gehen, findet eine Adressübersetzung in eine globale Adresse statt. Wichtig ist nur, dass keine Pakete, die an eine private Adresse gerichtet sind, nach außen gelangen dürfen. Es gibt verschiedene Verfahren:

  Statisches NAT Jede private Adresse wird hierbei einer bestimmten externen Adresse zugewiesen. Dabei braucht man allerdings genauso viele externe IP Adressen wie private und gewinnt nichts.

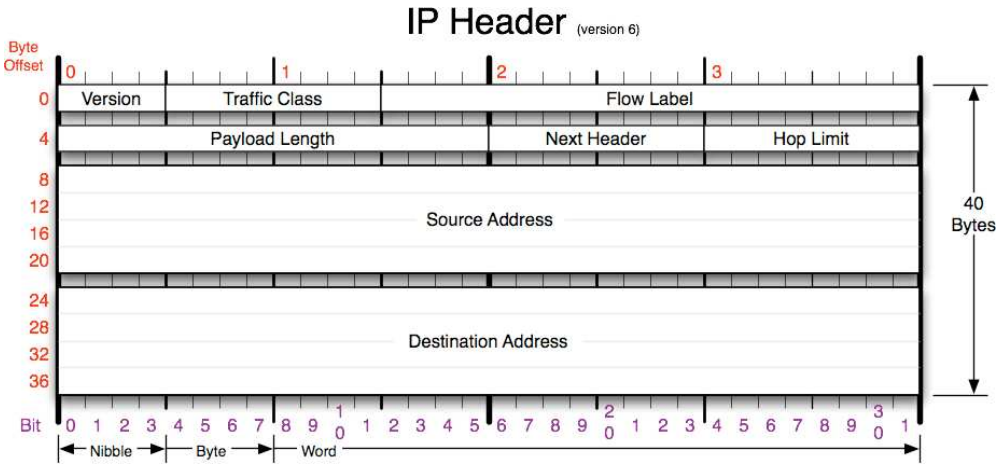
  Dynamisches NAT Die NAT Box verwaltet eine gewisse Anzahl externer Adressen (die Anzahl ist kleiner als die der privaten) und weist diesen dynamisch je nach Anfrage privaten Adressen zu. Hier spart man externe Adressen, kann aber Probleme bekommen, wenn viele interne Hosts nach außen wollen. Eventuell kann dann keine Adresse zugewiesen werden.

  NAPT Beim Network Address Port Translation werden mehrere lokale Adressen auf dieselbe externe Adresse gemappt. Damit man trotzdem noch bei Antworten das Paket richtig zustellen kann, benötigt man mehr Informationen. Dazu schaut man sich einfach in den Nutzdaten des Pakets die TCP (oder UDP) Portnummer an und kann dadurch die Verbindungen richtig

zuordnen. Ein Problem hierbei ist, dass man gegen das Grundprinzip der Schichtenarchitektur verstößt, in dem man Annahmen darüber trifft, dass auf der Transportschicht TCP oder UDP verwendet wird. Hier sieht man die Schichten nicht als unabhängig an. Verwendet man z.B. ein anders Protokoll funktioniert NAPT nicht.

Generell weicht NAT auch die Bedeutung von IP Adressen auf, da diese eigentlich global eindeutig sein sollen (was durch die millionenfach verwendeten privaten Adressen nicht gegeben ist).

- Mit *IPv6* will man die Mängel an *IPv4* ausbügeln und Milliarden von Host durch längere IP-Adressen (128 Bit) unterstützen, Routing-Tabellen kleiner machen, das Protokoll vereinfachen, mehr Sicherheit (Authentifizierung und Datenschutz) einführen und Dienstgüten besser unterstützen usw. Der Header eines IPv6-Pakets sieht wie folgt aus:



Version bezeichnet immer noch die IP Version, *Traffic Class* dient zur Unterscheidung von Paketen mit unterschiedlichen Echtzeitanforderungen. Das *Flowlabel* soll es Quelle und Ziel ermöglichen eine Pseudoverbindung mit bestimmten Eigenschaften herzustellen. *Payload Length* gibt die Länge des Payloads in Byte an. Im Feld *Next Header* ist versteckt, wieso der IPv6 Header aufgeräumter ist: Hier rüber können nämlich optionale Header angegeben werden, die dem Header folgen. Folgt kein weiterer Header, gibt dieses Feld das Protokoll an, welches auf der Transportschicht angesprochen werden soll. *Hop Limit* zählt wieder die Hops und entspricht der TTL. Dann folgen die 16 Byte Adressen von Quelle und Ziel. In IPv6 wird Fragmentierung anders gehandhabt. Es kann nur die Quelle fragmentieren. Ist für einen Router das Paket zu groß, kann er nur einen Fehler zurück melden. Als optionale Header gibt es sowas wie Verfügungstellung von Informationen für die Router, zusätzliche Informationen fürs Ziel, Fragmentierung, Authentifizierung, Verschlüsselung.

- IPv4 und IPv6 können gut nebeneinander existieren. Um von einem IPv6-Netz mit einem IPv4-Netz zu sprechen gibt es zwei Methoden: Entweder man versucht die Header ineinander zu konvertieren (geht nicht immer gut) oder man tunnelt ein IPv6-Paket durch den IPv4-Bereich durch, in dem man das gesamte Paket als Payload in ein IPv4-Paket packt.

4.3 Hilfsprotokolle

ARP (Address Resolution Protocol) ARP übersetzt IP Adressen des eigenen LANs in 48 Bit MAC-Adressen, um ein IP-Paket richtig in der Sicherungsschicht zu adressieren. Kennt ein Host nicht die MAC-Adresse des Ziels, so wird ein Broadcast im Ethernet ausgegeben mit der Frage, wer die entsprechende IP hat. Das Ziel antwortet dann mit seiner MAC-Adresse. Die MAC-Adresse wird dann eine bestimmte Zeit zwischengespeichert. Damit das Ziel für die Antwort nicht auch eine ARP Anfrage stellen muss, sendet der Host seine MAC-Adresse und IP Adresse mit (auch die anderen Hosts können sich dieses Paar dann merken). Eine weiter Optimierung wäre, wenn jeder Rechner seine Identität beim Starten broadcastet. In der Regel wird hierbei nach seiner eigenen IP gefragt (es kommt keine Antwort, aber alle Host merken sich das Adresspaar). Bei Adressierungen, die über das lokale Netzwerk hinausgehen, muss als MAC-Adresse zunächst der Router angegeben werden, der den Frame dann umadressiert auf die nächste MAC-Adresse zum Ziel.

RARP (Reverse Address Resolution Protocol) Mit RARP soll zu einer MAC-Adresse die gehörige IP-Adresse angefordert werden (wichtig für IP Zuweisung beim Booten einer Workstation). Dabei wird beim Starten eine Anfrage gesendet, die die eigene MAC-Adresse enthält. Ein RARP-Server beantwortet diese Anfrage mit der zugehörigen IP-Adresse. RARP funktioniert auch nur lokal, so dass in jedem Netz ein RARP-Server nötig ist (Ethernet Broadcasts werden nicht von Routern weitergegeben).

DHCP (Dynamic Host Configuration Protocol) Statt einer RARP Anfrage wird hier ein `DHCP DISCOVER` Paket gesendet. Ist der DHCP-Server nicht im selben Netz, so leitet ein DHCP-Relay-Agent es an den Server weiter. Neben der Zuweisung von IP Adressen kann mit DHCP auch die Subnetmask, Domain Namen usw. übertragen werden und somit der Host vollständig konfiguriert.

ICMP (Internet Control Message Protocol) Mit ICMP können unerwartete Ereignisse berichtet werden. Dabei ist jedes ICMP-Paket in einem IP-Paket gekapselt. Folgende Nachrichtentypen gibt es:

DESTINATION UNREACHABLE Ein Router kann das Ziel nicht finden oder aus anderen Gründen (Don't Fragment gesetzt) das Paket nicht zustellen.

TIME EXCEEDED Der Zähler hat null erreicht und das Paket wird verworfen.

PARAMETER PROBLEM Ein unzulässiger Wert in einem Header-Feld

SOURCE QUENCH Wurde verwendet, um einen Host zu drosseln, so dass er seine Aktivitäten etwas herunterschraubt. Heute wird Lastüberwachung meist der Transportschicht überlassen.

REDIRECT Ein Paket wurde falsch weitergeleitet.

ECHO REQUEST und `ECHO REPLY` werden benutzt um zu testen, ob ein Host erreichbar ist (wird für Ping benutzt).

IGMP (Internet Group Management Protocol) Dieses Protokoll verwaltet die Multicast Gruppen. Dazu sendet jeder Multicast-Router etwa jede Minute einen Hardwaremulticast an die Hosts in seinem LAN und fordert sie auf, die Gruppen zu melden, an denen ihre Prozesse momentan teilnehmen. Die Hosts antworten mit den entsprechenden Klasse D Adressen, an denen sie interessiert sind. Das Protokoll ähnelt ICMP. Das Routing erfolgt dann über Spanning Trees. Dazu tauscht jeder Multicast-Router über Distance Vector Routing Informationen mit anderen Multicast-Routern aus, so dass jeder pro Gruppe einen Spanning Tree verwaltet, der alle Mitglieder enthält.

5 Zusatzthema: MAC in Mobilkommunikation

- Normales CSMA/CD funktioniert bei mobilen Hosts nicht auf Grund der *versteckten* und *ausgelieferten Endgeräte*. Folgendes Szenario: Drei Stationen A, B, C, wobei B A und C empfangen kann aber die anderen beiden sich gegenseitig nicht. Man bezeichnet A als versteckt für C, wenn A an B etwas sendet und C nun auch etwas senden will (er Carrier Sensing und empfängt nichts), und beginnt ebenfalls mit seiner Übertragung zu B. Dort kollidieren die Pakete. Wenn B etwas an A sendet und C an jemand völlig anderen senden will, dann hört er, dass das Medium besetzt ist und startet daher (unnötigerweise) nicht seine Übertragung. C ist also B ausgeliefert.

- Des weiteren gibt es noch die Problematik *ferner* und *naher Endgeräte*. Da die Signalstärke proportional zur Entfernung zum Quadrat abnimmt, sind bei zwei unterschiedlich von dem Empfänger C entfernten Stationen A und B unterschiedlich „laut“. Wenn B näher dran ist, dann übertönt er A, so dass C A nicht empfangen kann. Es ist also eine präzise Power-Control notwendig, so dass möglichst die Signale beim Empfänger gleich stark ankommen (wichtig für CDMA).

- Neben den schon bekannten statischen Medienzugriffsverfahren kommen noch zwei neue Möglichkeiten hinzu: *CDMA* (Code Division Multiple Access) und *SDMA* (Space Division Multiple Access). Bei CDMA können Stationen zur selben Zeit auf dem selben Frequenzband anhand unterschiedlicher Codes unterschieden werden. Empfänger, die den Code kennen, können die Nachrichten dann entschlüsseln. Der Vorteil von CDMA ist, dass keine Planung (außer der Codezuteilung) nötig ist und Interferenzen nicht kodiert werden. Ein Nachteil ist, dass diese Methode hohe Komplexität beim Empfänger fordert und außerdem müssen

die Signalstärken stark kontrolliert werden ,so dass beim Empfänger alle Signale gleich stark ankommen. Stichworte für das benutzen eines Codes sind noch Bandspreizverfahren wie DSSS (Direct Seuense Spread Spectrum). Bei SDMA teilt man den Raum in Zellen ein, in denen man dann z.B. wieder FDMA o.ä. einsetzten kann.

- Mit *MACA* (Multiple Access with Collision Avoidance) kann das Problem der versteckten und ausgelieferten Endgeräte gelöst werden. Hierbei fragt der Sender über ein `RTS` (Request to Send) Frame den Empfänger, ob gesendet werden kann. Ist der Empfänger bereit, so antwortet er mit einem `CTS` (Clear to Send), woraufhin der Sender mit der Übertragung der Daten beginnt. Die Pakete enthalten jeweils die Absender sowie Empfängeradresse und die zu sendende Paketgröße. Versteckte Endgeräte spielen hier keine Rolle, da in dem Beispiel oben zwar C nicht das `RTS`-Frame enthält, aber dafür die `CTS`-Nachricht von B und daraufhin am Senden gehindert wird. Ebenso ist C auch nicht mehr B ausgeliefert, da er zwar von B die `RTS` Anfrage hört, aber auf Grund der Tatsache, dass er von A kein `CTS` hört, darauf schließt, dass A außerhalb seiner Reichweite liegt und auch senden kann.

- Wenn eine Station von allen anderen empfangen werden kann (Basestation), so kann diese Station *Polling* einsetzten. Die Basestation fragt nach bestimmten Schemen (z.B. Round Robin) die anderen Stationen an, ob sie etwas zu senden haben und teilt ihnen in diesem Fall das Senderecht zu. (Wird bei Bluetooth und optional bei 802.11 eingesetzt).

- Wenn man CDMA und Aloha kombiniert, erhält man SAMA (Spread Aloha Multiple Access). Dabei spreizt man das Spektrum mit nur einem Code.

- In 802.11 sind folgende Medienzugriffsverfahren definiert (die auch als DFWMAC - Distributed Foundation Wireless Medium Access Control bezeichnet werden). Für die folgenden Verfahren werden drei Prioritäten definiert: *DIFS* für die längste Wartezeit und somit niedrigste Priorität. DIFS werden für den asynchronen Datendienst verwendet. Es folgen *PIFS*, die für zeitbeschränkte Dienste genommen werden. Die kürzeste Wartezeit haben *SIFS*, die für Steuernachrichten (z.B. Bestätigungen) verwenden sie.

  CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) Eine sendebereite Station hört das Medium ab. Wenn ist für die Dauer eines DIFS frei ist, kann die Station anfangen zu senden. Wenn es belegt ist, so muss die Station so lange warten, bis ein freies DIFS Platz hat. Zusätzlich muss dann noch eine zufällige Backoffzeit innerhalb des Wettbewerbfensters zur Kollisionsverhinderung gewartet werden. Diese Backoffzeit ist ein vielfaches der Timeslot-Dauer. Währenddessen wird weiterhin das Medium abgehört. Sendet jetzt jemand anders, so hat die Station es nicht geschafft und muss warten, bis das Medium wieder für DIFS frei ist (sie stoppt aber den Backofftimer an der Stelle, wo die andere Station den Zugriff gemacht hat und

geht mit diesem Zähler dann in die neue Phase). Ansonsten kann nach Ablauf der Backoffzeit gesendet werden.
Dieses Verfahren muss von allen Implementierungen von 802.11 unterstützt werden.

DFWMAC mit RTS/CTS Eine Station kann ein `RTS`-Paket schicken, nachdem er für DIFS gewartet hat und das Medium frei ist. Der Empfänger kann dann mit `CTS` bestätigten und braucht dafür nur SIFS zu warten. Nach weiteren SIFS kann dann die Station anfangen zu senden und wieder nach SIFS warten bestätigt werden vom Empfänger. Im `RTS`- und `CTS`-Paket ist neben den Adressen auch die voraussichtliche Dauer (inkl. Bestätigung) gespeichert, die sich alle übrigen Stationen, die das `RTS` empfangen, speichern müssen. Der Wert stellt dar, wann das Medium frühestens wieder frei wird. Man bezeichent den Vorgang auch als *virtuelle Reservierung*, da durch `RTS/CTS` das Medium exklusiv für den Sender reserviert wird.
Diese Variante ist in der Anwendung optional, jedoch müssen alle Stationen es korrekt beherrschen.

DFWMAC-PCF Hier wird eine Station benötigt, die den Medienzugriff steuert und nacheinander alle anderen Stationen pollt. Nach einer Wettbewerbsphase (die eine der vorherigen Methoden nutzt) muss der Koordinator für PIFS warten, bevor er den Medienzugriff machen kann. Nun können Daten an die erste Station gesendet werden, nach SIFS kann diese dann antworten, wiederum nach SIFS kann die zweite Station angesprochen werden usw. Dieses Verfahren kann gewisse Garantien bzgl. Zugriffsverzögerung und Bandbreite geben. Es wird aber nur sehr selten verwendet.