CE

Grundlagen der Netzwerktechnik

Inhaltsverzeichnis

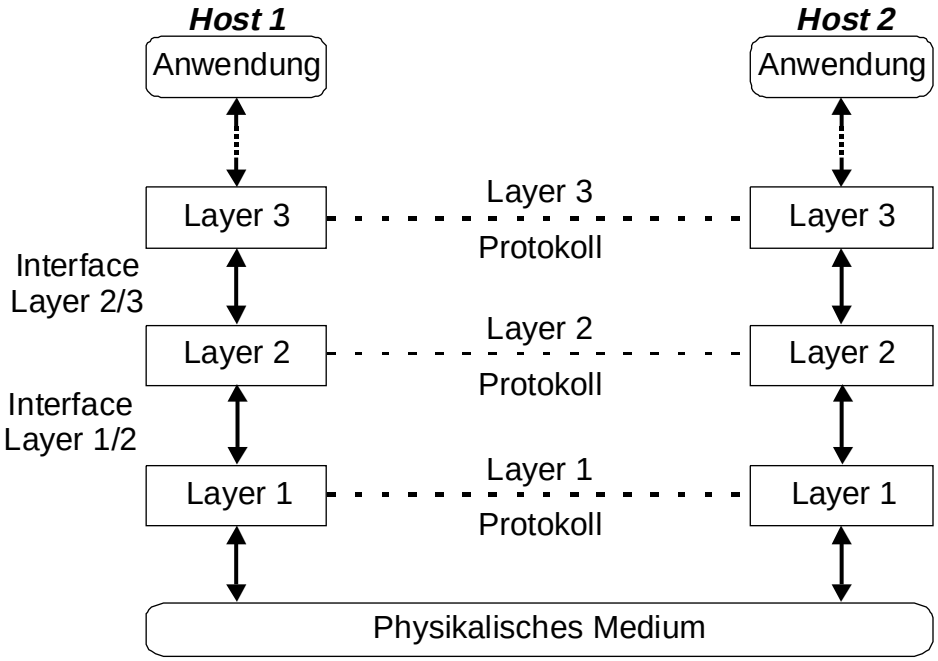
Index ... 42

EINFÜHRUNG

Schichtenmodelle

Auf Grund der vielfältigen Möglichkeiten, mit denen einzelne Rechner vernetzt werden können, wurden verschiedene Schichtenmodelle entwickelt. Zielsetzung hierbei ist es, die Anpassung an neue Rahmenbedingungen so einfach wie möglich zu gestalten.

Einerseits können sich die Netzwerke selbst ändern, andererseits muß es auch möglich sein, ein Netzwerk an eine vorhandene Infrastruktur anzubinden. Aufgrund dieser umfangreichen Forderung muß die Verständigung über ein Netzwerk durch mehrere verschiedene „Schichten“ laufen, die jede für sich spezielle Aufgaben warnimmt. Dadurch entstehen für die verschiedenen Schichten Nutzungsprofile, die den Aufgabenbereich klar abgrenzen. Die entstehende hierarchische Struktur läßt sich folgendermaßen skizzieren:



Jede Schicht („Layer“) stellt jeweils *nur den benachbarten* Schichten Dienste zur Verfügung. Dabei Nutzt z.B. Schicht 2 die Dienste („Services“) von Layer 1 um seinerseits wieder Layer 3 einheitliche Dienste zu bieten. Die Architektur einer Schicht ist einheitlich, so daß Schicht 2 nichts davon merkt, wie Schicht 1 die Daten letztendlich überträgt.

OSI - Open Systems Interconnected

Der OSI - Layer wurde von der Industrial Standards Organisation (ISO) definiert, um einen Anhaltspunkt für die Entwicklung von Netzwerkapplikationen zu bieten. Es besteht aus sieben Schichten, die wie folgt definiert sind:

7	Application	Anwendung
6	Presentation	Darstellung
5	Session	Kummunikation
4	Transport	Transport
3	Network	Vermittlung
2	Data Link	Sicherung
1	Physical	Bitübertragung

Layer 1: Physical Link Layer

Der Physical Link Layer übernimmt die reine Binärdatenübertragung zwischen zwei benachbarten Knoten. Er stellt lediglich eine ungesicherte Verbindung zur Verfügung. Dies bedeutet, daß die Sicherung der Übertragung von höheren Schichten übernommen werden.

Layer 2: Data Link Layer

Der Data Link Layer nutzt die Dienste, die vom Physical Link Layer bereitgestellt werden, um eine gesicherte Verbindung zwischen zwei benachbarten Knoten herzustellen. Auch der Data Link Layer ist eine reine Übertragungsschicht, die nichts mit der eigentlichen Anwendung zu tun haben.

Layer 3: Network Layer

Der Network Layer nutzt die gesicherten Verbindungen des Data Link Layers um die eigentliche Kommunikation zwischen zwei Netzteilnehmern in einem LAN oder WAN zur Verfügung zu stellen. Er sorgt primär für die anfallenden Arbeiten im Zusammenhang mit Routing.

Layer 4: Transport Layer

Der Transport Layer teilt den aufkommenden Datenstrom in einzelne Pakete auf, die dann nach erfolgreichem Empfang wieder von ihm in der richtigen Reihenfolge zusammengesetzt werden.

Layer 5: Session Layer

Der Session Layer steuert die Kommunikation selbst. Er nimmt Datenströme in Empfang und reicht sie entweder an den Presentation Layer oder den Transport Layer weiter, er öffnet und schließt die Verbindungen usw.

Layer 6: Presentation Layer

Der Presantation Layer konvertiert eingehende Anwenderdaten in ein Format, das vom Session Layer verwendet werden kann.

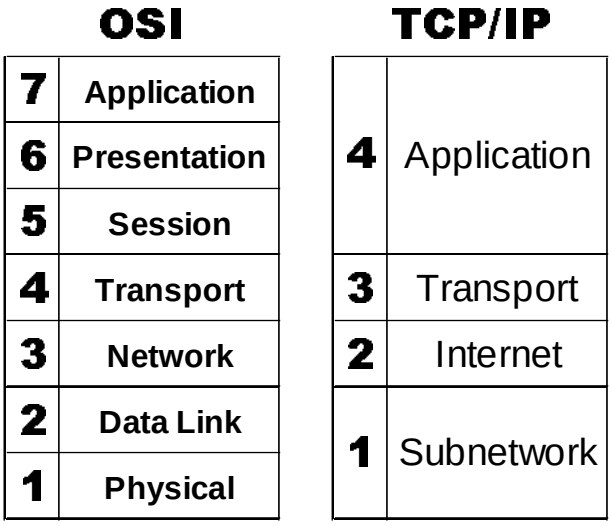
Layer 7: Application Layer

Der Application Layer repräsentiert die eigentliche Kommunikationsschnittstelle der Anwendung. Er verwaltet die Interaktion zwischen der eigentlichen Anwendung und den darunterliegenden Netzwerk-Layern.

Auf Grund der Komplexität des OSI Modells hat es sich nie durchgesetzt, da es schlichtweg von der Industrie nicht finanzierbar war. Lediglich die Grundlegende Struktur des ISO Modelles wird heute noch verwendet, und zwar im TCP / IP Referenzmodell.

Dieser Standard wurde 1983 also offizieller Standard des Internets eingeführt und ist dadurch heute in vielen Netzwerken ebenfalls zum Standard avanciert. Es wurde vor allem durch die verschiedenen UNIX Systeme verbreitet.

TCP / IP Referenzmodell



Layer 1: Subnetwork Layer

Der Subnetwork Layer ist im TCP/IP Modell nicht definiert. Es steht hier theoretisch jeder beliebige Netzwerkdienst zur Verfügung. Dies reicht von herkömmlichen LAN - Verbindungen über Punkt-zu-Punkt Verbindungen bis hin zu Richtfunkstrecken.

Layer 2: IP - Internet Protocol

Der IP Layer stellt eine ungesicherte Verbindung zwischen zwei Hosts zur Verfügung. Er prüft weder, ob die Daten empfangen wurden, noch ob die empfangenen Daten selbst korrekt sind.

Layer 3: TCP / UDP

Mit Hilfe des Transmission Control Protocols (TCP) wird auf der Basis des IPs eine gesicherte Verbindung zwischen zwei Hosts hergestellt. Es ist derzeit das weitverbreiteste Netzwerkprotokoll

Das User Datagram Protocol (UDP) ist ein ungesichertes, verbindungsloses und paketorientiertes Protokoll. Es wird bei Anwendungen genutzt, bei denen es nicht notwendig ist, das der komplette Datenstrom sein Ziel erreicht. Ein Beispiel hierfür ist die Life-Übertragung von Musik oder Video aus dem Internet.

Layer 4: Application Layer

Der Application Layer ist, wie der Subnetwork Layer nicht standarisiert. Im Regelfalle sind die betreffenden Programme direkt auf TCP / IP abgestimmt.

Im folgenden werden die Teile dieser beiden Referenzmodelle beschrieben, die heute relevant sind.

BESCHREIBUNG DER SCHICHTEN

Physical Link Layer

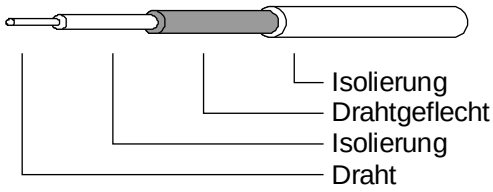
Der Physical Layer ist der erste Layer im Schichtenmodell. Er ist für die Übertragung der Daten über das eigentliche Medium verantwortlich. Dabei ist natürlich das Übertragungsmedium selbst ein integraler Bestandteil. Daher werden zuerst die verschiedenen gebräuchlichen Medien betrachtet.

Übertragungsmedien

Symmetrische Kabel (Twisted Pair)

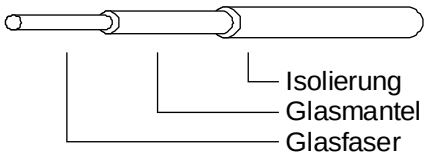
Das Twisted Pair besteht aus zwei verdrillten isolierten Kupferkabeln, die wiederum gebündelt werden. Dieses Anschlußsystem wird ebenfalls für Telefone verwendet.

Koaxialkabel



Koaxialkabel wird in vielen Gebieten verwendet. Es findet sowohl als Anschlußkabel für Fernseher als auch für kostengünstige LANs Anwendung.

Glasfaserkabel



Im Glasfaserkabel wird die innerste Glasfaser zur Datenübertragung genutzt. Dabei dienen als Sender entweder LEDs oder Halbleiterlaser und als Empfänger Fotodioden. Glasfaserkabel bieten die höchste möglich Bandbreite im Verhältnis zur möglichen Distanz zwischen zwei Hosts, sind aber die teuerste der hier vorgestellten Übertragungsmedien.

Elektromagnetische Wellen

Die Datenübertragen per elektromagnetischer Welle wird heute immer mehr genutzt. Auf Grund der begrenzten Bandbreite und der möglichen Störungen durch Wetter, andere Sender und allgemeine Einflüsse auf Grund anderer Technischer Geräte wird es hier jedoch nicht in der Tiefe behandelt. Der Hauptvorteil einer Funkstrecke liegt in der einfachen Installation und der Verfügbarkeit mobiler Stationen.

Im folgenden werden die Vor- und Nachteile von symmetrischem Kabel, Koaxialkabel und Glasfaserkabel gegenübergestellt. Hierbei kristallisiert sich bereits ein entsprechendes Anwendungsprofil heraus.

	Symmetrisches Kabel	***Coaxialkabel***	***Glasfaserkabel***
Übertragungsart	analog, digital		digital
maximale Datenrate	einige Mbps bei einigen km	2 Gbps bei 1 km	mehrere Gbps
Grund des Limits	Physikalische Eigenschaften des Kabels		Konvertierung zwischen elektrischen und optischen Medium
Verstärkerabst.	5 km		30 km
Platzverbrauch im Verhältnis zur Bandbreite	groß	mittel	klein
Interface - Kosten	kleiner		größer
Verwendungs-zweck	vom Telefon zur Ortsvermittlungsstelle, LAN	Fernsehen, LAN	Teile eines LANs, bei dem hohe Bandbreiten und / oder größeren Entfernungen benötigt werden.

Übertragungsarten

Es gibt zwei mögliche Übertragungsarten: Analog und digital. Jedes technische System setzt eine der beiden Übertragungsarten ein. Da aber, schon aus technischen Gründen, es nicht möglich ist, nur eines dieser beiden Systeme zu nutzen, sind zwangsläufig Umwandlungsmechanismen erforderlich. Grundsätzlich ist eine digitale Übertragung die unempfindlichere, da dort keine Umwandlung erforderlich ist. Im Gegensatz hierzu ist die analoge Technik einfacher aufgebaut und „natürlich“.

Analoge Signale

Ein Analoges Signal besteht grundsätzlich aus graduellen Zustandsänderungen, wie zum Beispiel Schwankungen des Luftdrucks (Schall). Vorteil des Analogen Signales ist, daß man es direkt weiterverwenden kann, wenn ein analoges Ausgangssignal benötigt wird. Ein empfangenes Radiosignal muß beispielsweise nur noch verstärkt werden, um es auf einem Lautsprecher auszugeben.

Digitale Signale

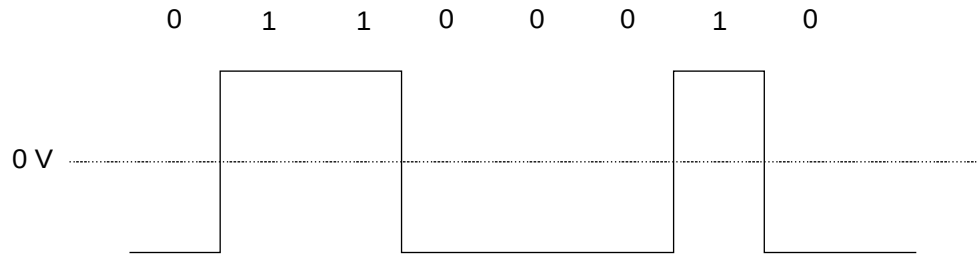
Ein digitales Signal besitze eine klar definierte Menge an Zuständen. In der Regel basieren alle diese Zustände auf einem Binären System. Übertragen werden lediglich diese Zustände. Hauptvorteil der Digitalentechnik liegt in der Tatsache, daß keine Übertragungsverluste auftreten. Das Signal exakt so empfangen, wie es gesendet wurde, da keine Umwandlungen nötig sind.

Der Trend geht derzeit im weitgehendsten Ersatz analoger Übertragungsmedien durch eine Digitale Übertragung. Die Umwandlung in ein Analoges Signal soll nur noch zur Endnutzung stattfinden, beispielsweise bei der Übertragung zum Lautsprecher. Grundgedanke ist es, so wenig Umwandlungsverluste wie möglich zu erzeugen.

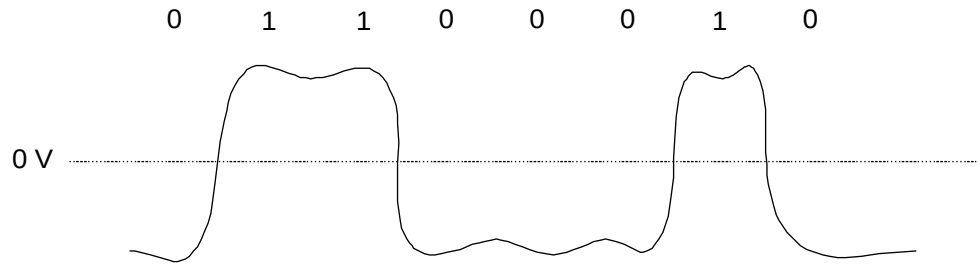
In einer Vernetzung, in der sowohl analoge als auch digitale Übertragungsarten genutztwerden, taucht zwangsläufig die Frage auf, wie beide Signalarten untereinander umgewandelt werden können.

Umwandlung eines digitalen zu einem analogen Signal

Ein einfaches Verfahren, das bei kurzen Wegen sehr oft verwendet wird, ist die **Gleichstromübertragung**. Sie ersetzt die beiden digitalen Signale 0 und 1 durch definierte Spannungszustände. Idealerweise ergibt sich dadurch ein rechteckiges Signal:



Durch externe Störungen wie Magnetfelder o.ä. und technischen Grenzen ergibt sich jedoch in der Praxis eher ein angenähertes Rechtecksignal:



Laut Shannon gibt es eine theoretische Grenze für die maximale Bandbreite einer solchen Übertragung:

$$U = H \log_2(1+\tfrac{S}{R})$$

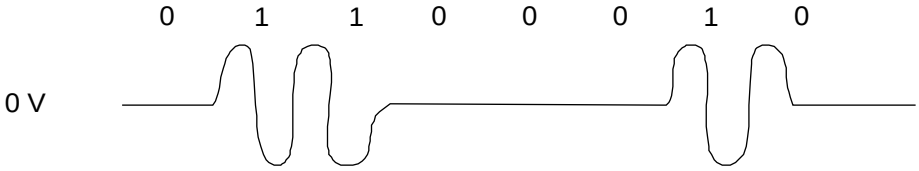
U = Maximale Bandbreite in Bit pro Sekunde (= Baud, bps)

H = Bandbreite in Hz

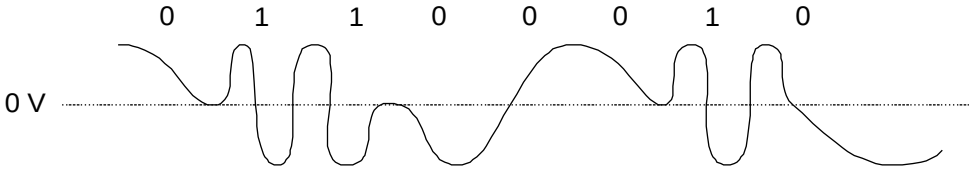
S/R = Signal - Rauschverhältnis

Es gibt drei weitere Grundtechniken für eine Signalumwandlung, die von modernen Geräten wie Modems sehr oft kombiniert werden.

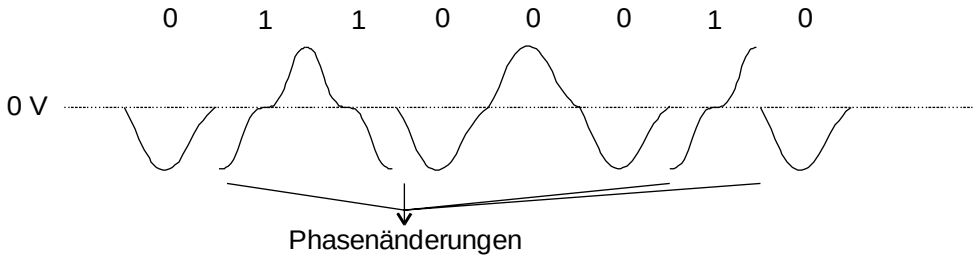
Amplitudenmodulation:



Frequenzmodulation:



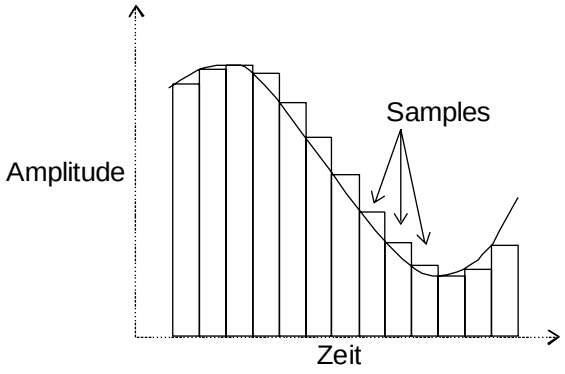
Phasenmodulation:



Umwandlung eines analogen in ein digitales Signal:

Ein analoges Signal kann relativ einfach digital verarbeitet werden: In regelmäßigen Abständen wird die Pegelhöhe gemessen und in ein digitales Signal umgewandelt. Fachbegriff hierfür ist „Sampling“. Hier sind zwei Größen von Bedeutung: Die **Sampling - Frequenz** legt fest, wie oft ein Wert eingelesen wird, während die **Sampling - Breite** die Genauigkeit dieser Werte festlegt. Folgende Tabelle stellt einige typische Übertragungswerte dar:

Gerät	**Telefon**	**Radio**	**CD**
Frequenz	8.000 Hz	22.500 Hz	44.000 Hz
Breite	8 Bit	8 Bit	16 Bit



Zusammenfassung

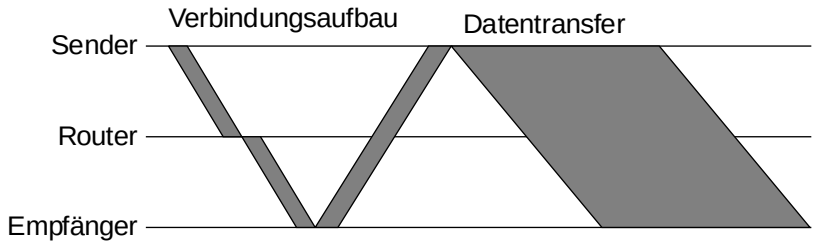
Auf Grund technischer Einschränkungen ist die Analoge Signalverarbeitung immer mit Verfälschungen verbunden. Da dies zu einem sehr komplexen Fehlerverhalten führt, wird die digitale Verarbeitung immer mehr angewendet. Digitale Verarbeitung und Signalverstärkung arbeiten fehlerfrei, Übertragungsfehler sind meist die Ausnahme, woraus ein sehr einfaches fehlerverhalten resultiert.

Verbindungsarten

Leitungsvermittelt

Unser Telefonnetz arbeitet leitungsvermittelt. Dies bedeutet, das Verbindungen nach Bedarf dynamisch aufgebaut werden. Dies führt zu einer schlechten Bandbreitenausnutzung und zu schweren Problemen, wenn eine Verbindungsstrecke ausfällt. Ausweichrouten sind nur in begrenztem Rahmen aufbaubar. Sehr oft sind die direkt betroffenen Teilnehmer abgeschnitten.

Der Datenfluß in einem leitungsvermitteltem Netz läßt sich folgendermaßen darstellen:

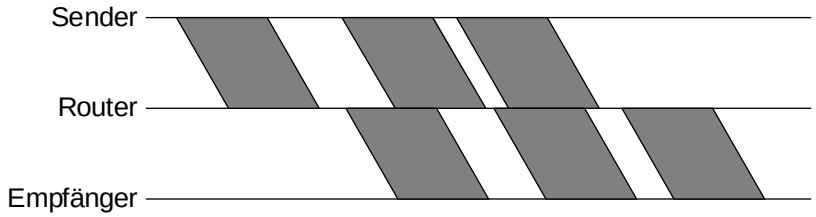


Ein solches leitungsvermitteltes System führt zur Bildung einer Hierarchie. Ortsnetze haben zentrale Vermittlungsstellen, die wiederum zu regionalen Subnetzen zusammengefaßt werden u. s. w. Dies ist besonders dann sinnvoll, wenn Nahverbindungen wesentlich häufiger sind, als Fernverbindungen.

Paketvermittelt

Paketvermittelte Netze bieten mehrere Vorteile gegenüber leitungsvermittelten Netzen. Wichtigster Vorteil ist die wesentlich bessere Ausnutzung der Bandbreite. Dies resultiert aus der Tatsache, daß in einem paketvermitteltem Netz die Daten keinen festen Weg mehr vorgeschrieben haben. Dadurch können bestimmte Teile eines Netzes umgangen werden, falls beispielsweise in benachbarten Netzen wesentlich weniger Auslastung herrscht. Ein weiterer wichtiger Aspekt ist das Verhalten beim Ausfall eines Teilstückes: Die Pakete werden dann automatisch auf einem anderen Weg zum Ziel geleitet, genau so, als wäre die betroffene Region überlastet.

Entsprechend gestaltet sich der Datenfluß:



Ein einem paketvermitteltem Netz finden sich somit keine festen Routen. Ist ein Teil der Bandbreite verfügbar, so kann er auf der gleichen Teilstrecke gleichzeitig von mehreren Verbindungen genutzt werden.

Leitungsvermittlung im Vergleich zur Paketvermittlung

	Leitungsvermittelt	Paketvermittelt
Zur Verbindung zugeordnetes "Kabel"	ja	nein
Feste Route	ja	nein
Welche Bandbreiten vorhanden?	fest	variabel
Ausnutzung der insgesamt vorhandenen Bandbreite	schlechter	besser
Verbindungsaufbau	notwendig	optional
Verhalten bei Ausfall von Netzelementen	Abbruch der Verbindung	Pakete auf anderem Weg
Wann können Warteschlangen entstehen?	Verbindungsaufbau	bei jedem Paket
Kostenberechnung	pro Zeiteinheit	pro Paket

Frame Relay

Frame Relay wurde von der Telekom eingeführt. Es soll eine einfache Verbindung zwischen mehreren privaten Netzen ermöglichen. Im Regelfalle liegt die Bandbreite bei 1,5 Mbps. Es wird eine maximale mittlere Last vereinbart, die unter der maximalen Bandbreite liegt. Dadurch ist Frame Relay günstiger als eine Mietleitung. Frame Relay verschickt Daten

Data Link Layer

Die Aufgabe des Data Link Layers liegt in der Übertragung von Daten von einer Maschine zur nächsten.

Methoden zur Rahmenerkennung

Daten werden innerhalb von Rahmen (engl. „Frame“) übertragen. Folglich muß der Empfänger eine Möglichkeit besitzen, einen Rahmen zu erkennen. Es gibt drei Methoden, die hier genutzt werden können:

Längenangabe des Rahmens: Zu Beginn des Rahmens kann eine Längenangabe eingefügt werden. Dies wird nicht genutzt, da im Falle eines Übertragungsfehlers im Längenfeld der Empfänger nicht mehr synchronisiert werden kann.

Spezielle Codes für Rahmenstart und -ende: Es werden Codes jeweils für den Rahmenbeginn und das Rahmenende vereinbart. Treten diese innerhalb der Daten selbst auf, müssen sie vom Sender verändert werden.

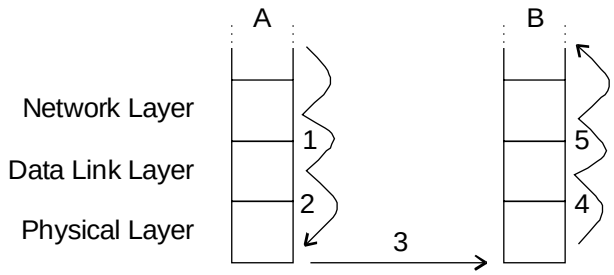
Protokollverletzung im Physical Layer: Rahmenstart und -ende können durch beabsichtigte Verletzung des Protokolls im Physical Layer festgelegt werden.

Protokollbeispiele

Alle hier aufgeführten Beispiele sind verbindungsorientiert. Sie basieren auf einer im Physical Link Layer bestehenden Duplex - Verbindung. Einige der hier vorgestellten Konzepte müssen nicht unbedingt innerhalb des Data Link Layers implementiert werden. Im Falle von TCP/IP sind die hier gezeigten Funktionen im Transport Layer implementiert.

Die Beispiele sind aus einem Buch von Tannenbaum. Sie sind vom Autor in Form eines Simulators in C unter http://www.cs.vu.nl/~ast/ verfügbar.

Einfaches Simplex - Protokoll



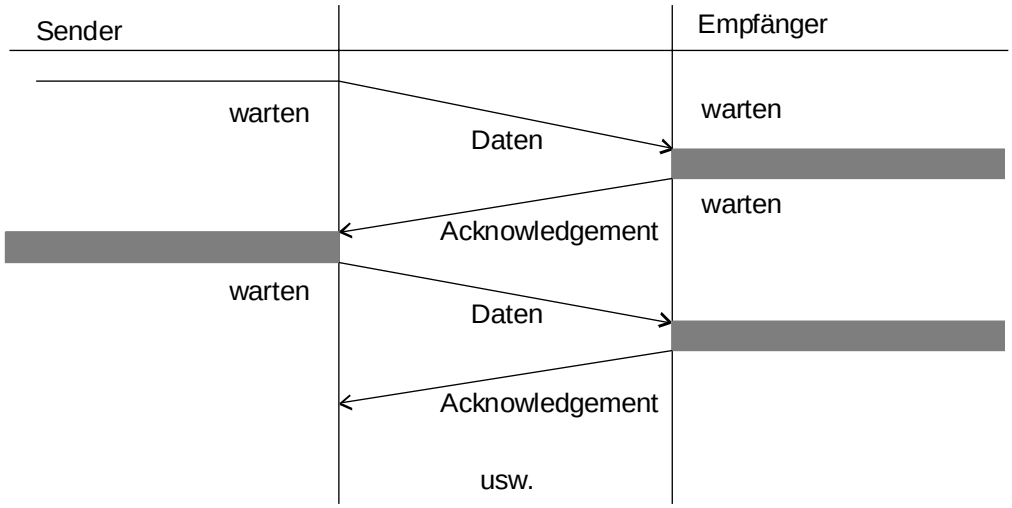
1. Die Daten werden dem Data Link Layer von A übergeben.
2. Dieser leitete sie in Rahmen verteilt an den Physical Layer weiter.
3. Der Physical Layer schickt die Daten durch die bestehende Verbindung an B.
4. Der Physical Layer von B übergibt die Daten dem Data Link Layer.
5. Dieser fügt sie wieder zu einem Datenstrom zusammen und reicht sie an den Network Layer weiter.

Diese einfache Protokoll hat zwei schwere Nachteile:

Einerseits muß der Network Layer des Empfängers ständig verfügbar sein, andererseits könne Übertragungsfehler nicht korrigiert werden.

Stop-And-Wait Simplex Protokoll

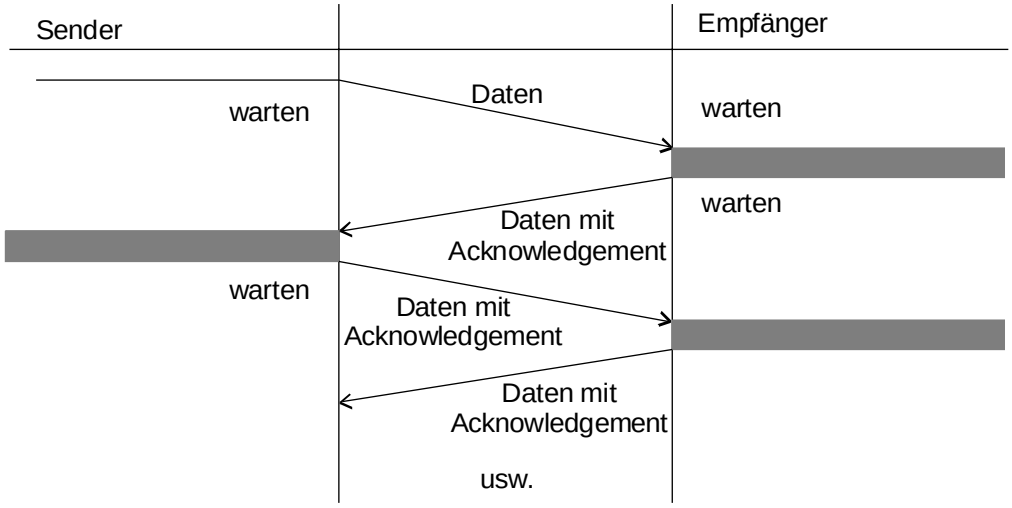
Das Stop-And-Wait Protokoll sieht vor, jeden Rahmen vom Empfänger bestätigen zu lassen:



Herrscht keine Netzaktivität, warten beide Teilnehmer. Es ist ein stabiler Endzustand. Sollten in diesem Fall Daten verloren gehen, wird dies weder von Sender noch Empfänger registriert, solange der Sender nicht mit Hilfe von Timeouts auf ankommende Acknowledgements reagiert. Probleme können hier von doppelt gesendeten Rahmen durch zu kurze Timeouts, doppeltem Empfang u.ä. entstehen. Dies wird für gewöhnlich durch Nummernvergabe gelöst, mit dem die Teilnehmer die Rahmen unterscheiden können. Diese ID wird sowohl im Datenpaket, als auch im Acknowledgement angegeben.

Einfaches Duplex - Protokoll

Duplexprotokolle sehen die Datenübertragung in beide Richtungen gleichzeitig vor. Es ähnelt stark dem Simplex Protokoll mit der Ausnahme, das die Acknowledgements jetzt zu einem Datenrahmen hinzugefügt wird:



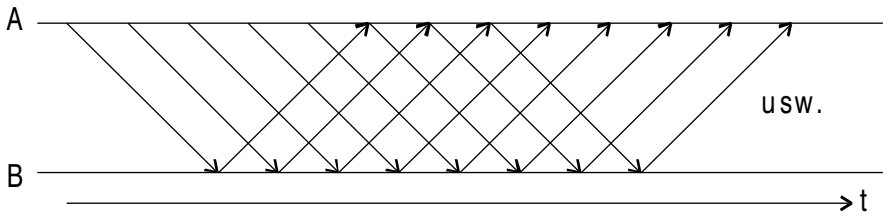
Aufgrund Der Wartezeiten durch die Übertragung entstehen riesige Lücken in der Bandbreitenauslastung. Ein Beispiel soll dies verdeutlichen:

Eine Satellitenverbindung mit einem Round Trip von 500 ms hat eine Bandbreite von 500 kbps. Verschickt werden Rahmen mit 1000 Bit, Verarbeitungszeiten sollen vernachlässigbar sein. Die Übertragungsrate von A nach B erreicht jetzt effektiv ca. 2 kbps. Dies resultiert hieraus, das zwar nur 2 ms gesendet wird, jedoch insgesamt 500 ms auf ein

Acknowledgement gewartet wird. Dadurch entsteht für 1000 Bit eine Übertragungszeit von 504 ms. Die Lösung stellt hier Pipelining dar.

Sliding Window Protokoll

Das Feld für die laufende Paketnummer erhält 3 Bit Breite. Dadurch können maximal sieben Datenrahmen gesendet werden, bevor auf das erste Acknowledgement gewartet werden muß. Jede Rahmen erhält einen Timer. Läuft dieser ab, werden ab dieser Stelle alle Rahmen neu gesendet. Es werden keine Fehler zurückgemeldet. Der Sender erkennt dies durch ein Timeout. Das Acknowledgement wird immer mit Daten gesendet, wobei es immer den zuletzt korrekt empfangenen Rahmen bestätigt. Dadurch können mehrere Rahmen mit einem Acknowledgement bestätigt werden.
Das Sliding-Window Protokoll löst die im letzten Abschnitt angesprochenen Probleme, realisiert also Pipelining:



Verbindungen über ein Modem: Point-to-Point Protocol (PPP)

PPP besitzt zwei Hauptanwendungsgebiete: Die Verbindung einzelner LANs und die Verbindung des Endbenutzers mit dem Internetprovider via Telefonleitungen. Neben PPP gibt es ein weiteres Protokoll für Punkt-zu-Punkt Verbindungen, SLIP (Serial Line Internet Protocol), das sich jedoch nicht durchgesetzt hat. Wir behandeln deshalb im weiteren nur PPP.

PPP besitzt drei Grundlegende Funktionen:

- **Einteilung des Datenstroms in Rahmen**, eine Fehlerentdeckung ist implementiert.
- **LCP** (Link Control Protocol) zum konfigurieren, Starten und Beenden einer Verbindung
- **NCP** (Network Control Protocol) zum Konfigurieren des Network - Layers, zum Beispiel für DHCP

Typischer Verbindungsaufbau eines Endbenutzers:

1. Es wird vom PC des Endnutzers zum Inernet Service Provider (ISP) eine Verbindung hergestellt.
2. Die Verbindung im Physical Layer wird etabliert.
3. Die eigentliche PPP - Verbindung wird mit Hilfe von LCP Paketen, die sich bereits in PPP - Rahmen befinden, aufgebaut.
4. In der Regel wird mit Hilfe von NCP Paketen dynamisch eine IP - Adresse zugeordnet.
5. Die Verbindung seht, so daß mit Hilfe von IP Paketen die Verbindung endgültig aufgebaut werden kann. Dies geschieht im Regelfalle mit Hilfe von DHCP.
6. Der Verbindungsabbau wird mit Hilfe von NCP Paketen erreicht, die dynamisch zugewiesene IP Adresse wird wieder freigegeben.

Bytes	Wert	Beschreibung
1	01111110	Flag
1	11111111	Adresse
1	00000011	Control
1 oder 2		Protokoll
variabel		Nutzdaten
2 oder 4		Cheksumme
1	01111110	Flag

Das PPP - Rahmenformat:

Flag: Das Flag dient zur Erkennung des Rahmens, wird dieser Wert innerhalb der Nutzdaten entdeckt, werden diese angeglichen (->Character Stuffing)

Adresse: Dieses Feld wird verwandten Protokollen benötigt, ist hier immer 11111111.

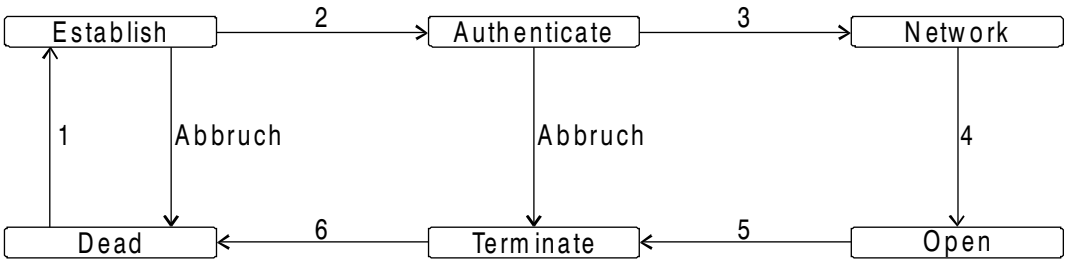
Control: Der Standardwert 00000011 besagt, das die Rahmen weder numeriert noch bestätigt werden. Bei schlechten Verbindungen kann dies bei Bedarf aktiviert werden.

Protokoll: Legt das für die Nutzdaten verwendete Protokoll fest. Mögliche Werte sind beispielsweise LCP, NCP, IP, IPX, Appletalk usw. Dies wird in der Regel mit Hilfe von LCP und NCP während des Verbindungsaufbaus vereinbart.

Nutzdaten: Die Länge dieses Feldes ist variabel, die Maximale Länge wird aber für gewöhnlich während des Verbindungsaufbaus vereinbart.

Checksumme: Diese wird über die Nutzdaten gebildet.

Statusdiagramm einer PPP - Verbindung



1. Der Dienst wird von Benutzer angefordert oder ein Signal wurde auf dem Physical Layer entdeckt (->Anruf)
2. Mit LCP werden die verwendeten Optionen vereinbart.
3. Wenn nötig, wird eine Nutzeridentifikation mittels Name und Passwort durchgeführt.
4. Mit Hilfe von NCP wird der Network Layer konfiguriert.
5. Eine Seite beendet die Verbindung
6. Verbindung im Physical Layer wird abgebaut.

LCP Pakettypen

Folgende LCP Paktetypen sind definiert. Die Richtung gibt hierbei immer den Paketweg zwischen dem initiierenden Rechner (*I*) und dem antwortenden Rechner (*A*) an.

Name	Richtung	Beschreibung
Configure-Request	I->A	Vorgeschlagenen Optionen werden gesendet
Configure-Ack	I<-A	Alle Optionen wurden akzeptiert
Configure-Nack	I<-A	Einige Optionen werden nicht akzeptiert
Configure-Reject	I->A	Einige Optionen sind nicht verhandelbar
Terminate-Request	I->A	Verbindungsabbau
Terminate-Ack	I<-A	Verbindungsabbau bestätigen
Code-Reject	I<-A	Fehler oder verschiedene LCP Versionen
Protocol-Reject	I<-A	Fehler oder verschiedene LCP Versionen
Echo-Request	I->A	Leitungstest
Echo-Reply	I<-A	Leitungstest
Discard-Request	I->A	Leitungstest

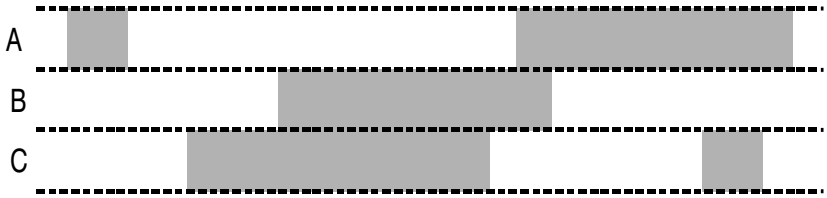
Medium Access Sublayer (MAC)

Im MAC Sublayer sind Protokolle zur Datenübertragung über ein Broadcastnetzwerk definiert. Es handelt sich hier für gewöhnlich um ein LAN. Innerhalb der OSI Protokollarchitektur ist der MAC Sublayer ein Bestandteil des Data Link Layers.

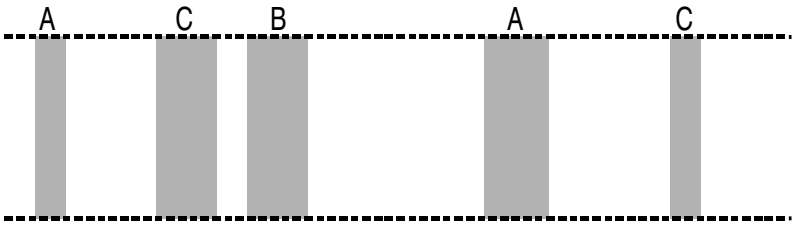
Broadcast - Netzwerke

In einem Broadcast - Netzwerk ist immer nur ein Teilnehmer sendeberechtigt. Jeder angeschlossene Teilnehmer kann die über das Medium übertragenen Daten lesen und, falls sie für ihn bestimmt sind, verarbeiten.

Frühere Lösungen beinhalteten die Aufteilung der verfügbaren Bandbreite in „Subkanäle“, die den einzelnen Teilnehmern zugeordnet wurden:



In Computernetzen hat sich diese Lösung nicht bewährt, da sie die vorhandene Bandbreite schlecht ausnutzt. Hauptgründe hierfür ist einerseits die wechselnde Anzahl an Sendern und andererseits die große Varianz an benötigter Bandbreite, die Computeranwendungen mit sich bringen. Diese Überlegungen führten zur Entwicklung der Broadcast-Netzwerke, die jeweils den kompletten Kanal einem Sender zur Verfügung stellen:

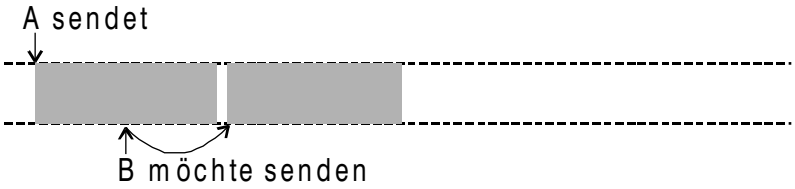


Ethernet

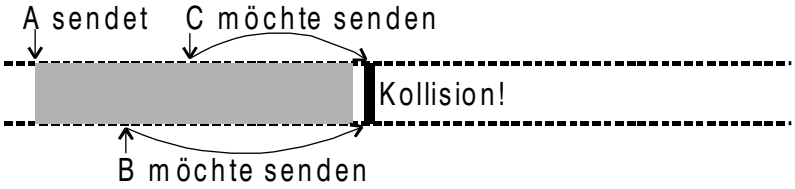
Ethernet wurde in den 70er Jahren von Xerox ins Leben gerufen. Es besaß damals eine Bandbreite von 2,94 Mbps. Einige Zeit später wurde dieser Standard von Xerox, Intel und DEC aufgegriffen und zu einem Standard für ein 10 Mbps Ethernet erweitert. Dieser Standard bildete die Grundlage für die IEEE 802.3 Norm (Institute of Electrical and Electronics Engineers).

Grundprinzip CSMA/CD

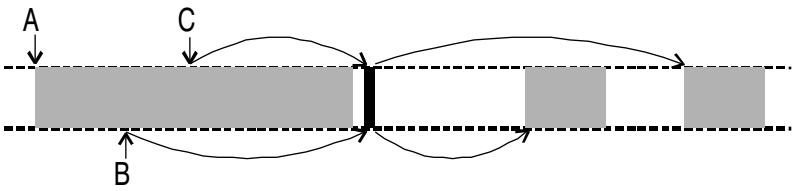
Das Grundprinzip für ein solches Netzwerk ist CSMA/CD (Carrier Sense Multiple Access with Collision Detection). Ethernet ist als Bussystem konzipiert, in dem jede Station an das selbe Medium angeschlossen ist. Dadurch kann jede Station den Kanal abhören und beginnt nur dann mit einem Sendevorgang, wenn der Kanal frei ist:



Dieses System bringt ein Problem mit sich: Warten zwei Stationen gleichzeitig auf einen freien Kanal, kommt es zwangsläufig zur Kollision, da zwei Teilnehmer gleichzeitig den Sendeversuch starten. Dieses Problem wird dadurch gelöst, das der Sender auch während des Sendevorgangs den Kanal abhört. Dadurch kann an einem verfälschten Signal erkannt werden, daß ein zweiter Sender Aktiv ist. Tritt dies ein, wird der Sendevorgang bei beiden Teilnehmern sofort abgebrochen:



Nach einer Kollision wartet der Sender eine von einem Zufallsgenerator bestimmte Zeit und beginnt dann erneut mit dem Sendevorgang. Dies stellt sicher, daß die Wahrscheinlichkeit einer erneuten Kollision zwischen den beteiligten Teilnehmern gering ist:



Das Protokoll paßt die durchschnittliche Wartezeit dynamisch an. Sie wird bei häufigen Kollisionen erhöht, sinkt die Kollisionshäufigkeit wieder, werden auch die Wartezeiten verkleinert.

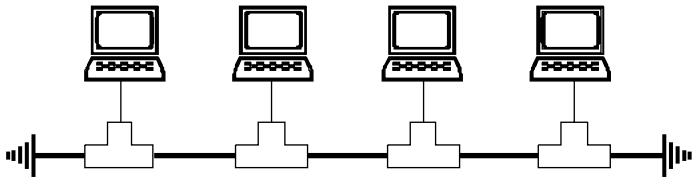
Verkablungsarten

Innerhalb der Norm für 10 Mbps Ethernet sind folgende Verkablungsarten definiert. Die Segmentgröße gibt hierbei die maximale Länge eines Netzwerksegmentes ohne die Verwendung von Bridges oder Repeatern an:

Name	Medium	Maximale Segmentgröße	Knoten pro Segment
10Base5, Thik Ethernet	Dickes Koaxialkabel	500 m	100
10Base2, Thin Ethernet	Dünnes Koaxialkabel, RG58	200 m	30
10Base-T, Twisted Pair	Symmetrisches Kabel, RJ45	100 m	1.024
10Base-F, Fibrechannel	Glasfaserkabel	2.000 m	1.024

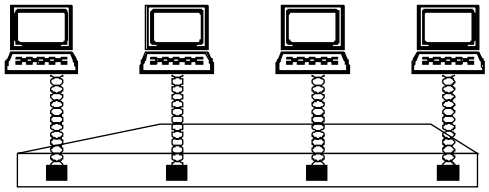
Koaxialkabel

10Base5 ist der ältester dieser Standards, ist heute aber noch bei Backbones von Bedeutung. Da 10Base5 Verkabelungen relativ teuer sind, wurde als kostengünstigere Alternative 10Base2 eingeführt, das heute weit verbreitet ist. Die Verbindung zum Rechner wird über ein T-Stück hergestellt. Für beide Systeme gibt es Geräte, die anhand von Signaltests fehlerhafte Kabel identifizieren können. Es entsteht eine busförmige Topologie, die an den Enden mit Endwiederständen bestückt wird:



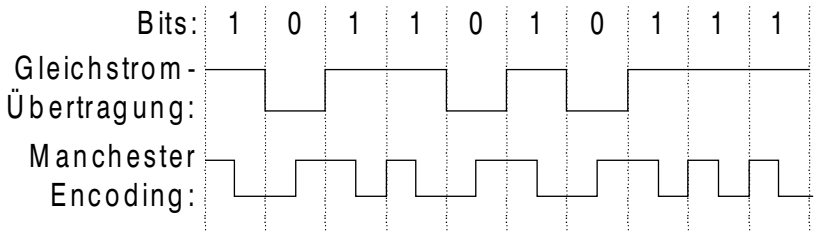
Twisted Pair, Glasfaserkabel

10Base-T besitzt kein Buskabel im eigentlichen Sinne. Dieses wird durch einen Hub ersetzt, wodurch sich eine sternförmige Topologie ergibt. Eine solche Verkabelung kann bautechnische Vorteile mit sich bringen, da kein Bus mehr „rundherum“ gelegt werden muß. Mehrere Hubs können untereinander verbunden werden. Oft werden Glasfaserkabel für Verbindungen zwischen entfernten Hubs oder Gebäuden verwendet:



Datenübertragung

Als Codierung für die eigentliche Datenübertragung wird Manchester Encoding verwendet. Dabei ist High = +0,85 V, Low = -0,85 V:



Begrenzungen

In einem Ethernet LAN darf die maximale Kabellänge zwischen zwei Stationen 2,5 km nicht überschreiten. Es darf nur einen Weg geben und auf diesem dürfen maximal vier Repeater (=Signalverstärker) sein. Damit bei maximaler Entfernung und maximaler Repeaterzahl die Kollisionsabfrage während des Sendens noch korrekt arbeitet, haben Ethernet-Rahmen eine Mindestlänge von 72 Byte.

Der Ethernet - Rahmen

Länge in Byte	Bedeutung
7	Präambel
1	Start des Rahmens
6	Empfängeradresse
6	Senderadresse
2	Pakettyp
46-150	Daten
4	CRC - Checksumme des Rahmens

Präambel: Bitfolge 1010... zur Synchronisation des Empfängers

Start des Rahmens: 10101011, markiert den Beginn des eigentlichen Rahmens

Sender- und Empfängeradresse: 48 Bit breite ID - Nummer, die weltweit für jede Ethernet Adapterkarte eine eindeutige Zuordnung erlaubt. Das IEEE vergibt hier Nummernkontigente an die einzelnen Hersteller.

Pakettyp: Gibt das Network Layer Protokoll an, beispielsweise 0x0800 für IP

Höhere Bandbreiten im Ethernet

Switched Hubs: Ein Switched Hub teilt das LAN in mehrere Kollisionsdomänen ein. Da pro Kollisionsdomäne eine Einheit senden, kann, läßt sich durch die Unterteilung innerhalb eines Switches höhere Bandbreiten erreichen. Die Kommunikation unter verschiedenen Kollisionsdomänen geschieht über ein hubinternes Protokoll. Ein Switched Hub besitzt eine gesamte Bandbreite, mit der die einzelnen Netzsegmente kommunizieren können. Im allgemeinem ist ein Switched Hub nur sinnvoll, wen ein größerer Anteil der Kommunikation lokal innerhalb des Hubs ist.

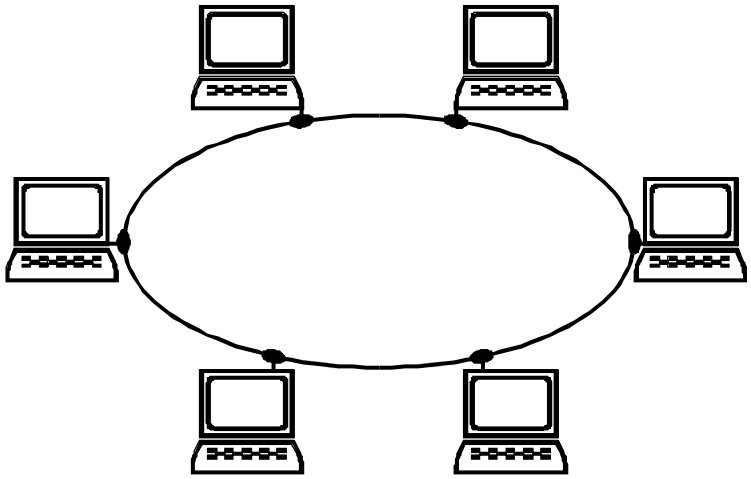
Fast Ethernet: Fast Ethernet erweitert ein herkömmliches 10 MBit Ethernet auf 100 MBit. Die Abwärtskompatibilität soll dabei weitesgehend erhalten bleiben, so daß vorhandene Netzsegmente problemlos aufgerüstet werden können. Dies zeigt sich beispielsweise im identischen Ethernet Rahmenformat.

Mögliche Verkablungsarten sind:

Name	*Kabelpaare*	*Kabeltyp*	*maximale Entfernung*
100Base-TX	2	Kategorie 5	100 m
100Base-FX	2	Glasfaser	2000 m

Token-Ring

Token-Ring Netzwerke wurde 1980 von IBM entwickelt. Im Gegensatz zum busförmigen Ethernet sind Token-Ring Netzwerke in Form eines unidirektionalen Rings aufgebaut.



Sendet keine Station, so kreist ein 3 Byte langes Token im Ring. Will eine Station senden, so wartet sie auf das Token, ändert ein Bit, so daß ein Rahmenanfang entsteht. Danach wird der Rahmen gesendet. Kommt ein Rahmen wieder bei der Station an, nimmt sie ihn vom Ring und sendet einen weiteren Rahmen oder wieder das Token. In letzterem Fall schaltet die Station rechtzeitig wieder in den Empfangsmodus, so daß das Token im Netz bleibt. Jede Station vergleicht die Zieladresse mit der eigenen Adresse. Ist sie gleich, wird der Rahmen kopiert und weiterverarbeitet. Am Ende des Rahmens wird ein Bit als Quittung gesetzt.

Eine Station innerhalb des Token-Ring Netzwerks fugniert als Monitorstation. Er sorgt beispielsweise für die notwendige Länge des Rings, ob das Token kreist oder ob fehlerhafte Rahmen auftreten. Ist ein Monitor nicht mehr aktiv, wird eine beliebige, andere Station zum Monitor. Übliche Geschwindigkeiten sind 4 oder 16 Mbps.

Zur Verkabelung werden symmetrische Kabel verwendet, die in einem Verteiler zusammengeführt werden. Dies führt dazu, daß ein Ethernet und ein Token-Ring LAN eine identische Verkabelungsstruktur besitzen.

Vergleich Ethernet, Token-Ring

Technisch sind Ethernet und Token-Ring gleichwertig. Da Ethernet einen weit höheren Marktanteil als Token-Ring besitzt, wird im Falle einer Neuinstallation eher Ethernet bevorzugt. Bei vorhandenen Installationen wird die vorhandene Infrastruktur bestimmend sein, da eine Umstellung zwischen den beiden Systemen sehr teuer und aufwendig wäre.

Bridges

Mit Hilfe von Bridges lassen sich mehrere Teile eines LANs verbinden. Es können verschiedene LAN Typen, die ansonsten inkompatibel währen, zusammengeschaltet werden. Im Gegensatz zu einem Repeater wird nicht einfach das Signal kopiert. Die Bridge entscheidet bei jedem Rahmen, ob er weitergeleitet werden soll. Die notwendigen Informationen stammen aus dem Data Link Layer. Folglich ist die einzige Unterscheidungsmöglichkeit für die Rahmenübertragung die Ethernet ID. Bridgepaare können entfernte LANs miteinander verbinden, in dem sie die auf dem Netz laufenden Daten in einen zweiten Rahmen kapseln und über ein WAN übertragen.

Transparent Bridges

Mit Transparent Bridges kann mit minimalem Aufwand ein komplexes LAN aufgebaut werden. Die Konfiguration erfolgt automatisch, was allerdings dazu führt, daß sich das System nicht optimal konfiguriert. Wird von einer Bridge ein Rahmen mit einer unbekannten Zieladresse empfangen, wird der Rahmen an alle LANs außer dem Ursprungs-LAN weitergeleitet. Ist die Zieladresse bekannt, wird der Rahmen nur in das entsprechende Ziel-LAN gesendet. Diese Tabelle wird durch Analyse der Senderadressen aufgebaut. Kommt einige Minuten lang kein Rahmen eines Senders bei der Bridge an, wird der Eintrag wieder aus der Tabelle entfernt. Da dieses System nur in einer baumartigen Netzstruktur zuverlässig arbeitet, konfigurieren sich die Bridges automatisch selbst, so daß ein Spanning Tree entsteht. Da dieser Baum sich mehr oder weniger zufällig aufbaut, ist er, wie erwähnt, nicht immer optimal.

Network Layer

Der Network Layer hat die Aufgabe, Pakete vom Ursprung zum Ziel zu bringen. Dazu könnte auf das Überspringen mehrerer dazwischengeschalteter Router gehören. Diese Aufgabe unterscheidet sich stark vom Data Link Layer, der lediglich Rahmen von einem Ende der Leitung zum anderen befördern muß.

Designaspekte des Network Layers

Hier sollen im folgenden kurz die Grundlagen der Vermittlungsschicht erläutert werden, mit denen sich Entwickler der Vermittlungsschicht befassen müssen.

Dienste für den Transport Layer

Der Network Layer stellt die Dienste für den Transport Layer zur Verfügung. Für diesen sollte es möglichst transparent sein, so daß der Transport Layer unabhängig von der verwendeten Verbindung arbeitet. Dies führt dazu, daß der Network Layer in der Regel die Schnittstelle zwischen Netzwerkbetreiber und dem Kunden, also die Grenze des Teilnetzes bildet. Aus diesem Grund muß der Network Layer besonders exakt definiert werden. Die Dienste des Network Layers werden mit folgenden Zielen ausgelegt:

1. Die Dienste sollen unabhängig von der Technologie des Teilnetzes sein.
2. Der Transport Layer muß vor der Anzahl, der Art und der Topologie der vorhandenen Teilnetze abgeschirmt werden.
3. Die dem Transport Layer zur Verfügung gestellten Netzadressen müssen ein einheitliches Numerierungsschema darstellen, auch zwischen LANs und WANs.

Die Entwickler des Network Layers bekommen durch diese Vorgaben sehr viel Freiraum bei den genauen Spezifikationen, so daß sich inzwischen zwei Standpunkte herauskristallisiert haben. Der wesentliche Streitpunkt besteht in der Frage, ob der Network Layer verbindungorientierte oder verbindungslose Dienste bereitstellen soll.

Die Internet - Gemeinde argumentiert damit, daß das Teilnetz lediglich Bits befördern und sonst nichts übernehmen soll. Aus seiner Sicht ist das Teilnetz inhärent unzuverlässig, gleichgültig, wie es entwickelt wird. Deshalb sollten die Hosts die Tatsache akzeptieren, daß es unzuverlässig ist, und Fehlerüberwachung sowie Flußsteuerung selbst übernehmen. Dieser Standpunkt führt schnell zu dem Schluß, daß der Dienst des Network Layers verbindungslos sein sollte, mit nicht viel mehr als den Dienstelementen send und recieve packet. Dadurch muß jedes Paket die volle Zieladresse enthalten, da es gänzlich unabhängig von seinen Vorgängern verschickt wird. Des weiteren ist die Rechnerleistung auf der Benutzerseite inzwischen derart preisgünstig geworden, daß kein Grund besteht, die Komplexität nicht in den Hosts abzustellen. Weiterhin ist der Aufbau eines Teilnetzes eine Investition für Jahrzehnte, weshalb dieses nicht mit Funktionen gefüllt werden sollte, die eventuell künftig nicht mehr relevant sind. Außerdem existieren Anwendungen, beispielsweise Internet - Telefonie, bei denen Schnelligkeit wichtiger ist, als Genauigkeit.

Die Netzbetreiber argumentieren damit, daß das Teilnetz einen zuverlässigen, verbindungsorientierten Dienst bieten soll. Sie verweisen auf eine hundertjährige praktische Erfahrung mit dem weltweiten Telefonsystem. Aus ihrer Sicht sollten Verbindungen folgende Merkmale aufweisen:

1. Vor dem Übertragen der Daten muß der Network Layer auf der Sendeseite eine Verbindung mit der gleichen Schicht auf der Empfangsseite aufbauen.
2. Beim Aufbau läßt sich über Parameter, Qualität und Kosten des Dienstes verhandeln.
3. Die Kommunikation ist bidirektional und Pakete werden in Folge zugestellt
4. Flußsteuerung wird automatisch bereitgestellt, um die Synchronisation zwischen beiden Teilnehmern zu gewährleisten.

Interne Organisation

Im Zusammenhang mit dem internen Aufbau des Teilnetzes spricht man hier von *Datengrammen* beziehungsweise von *virtuellen Verbindungen*.

Virtuelle Verbindungen werden im allgemeinen in Teilnetzen benutzt, deren primärer Dienst verbindungsorientiert ist.

Ziel ist, nicht für jedes Paket eine neue Route wählen zu müssen. Allen voran wird diese Organisation vom Telefonnetz genutzt.

Demgegenüber werden bei einem Datengrammteilnetz keine Routen im voraus festgelegt, auch wen der Dienst an sich verbindungsorientiert ist. Jedes Paket wird unabhängig von seinen Vorgängern befördert. Datengramteilnetze müssen eventuell mehr Arbeit übernehmen, sind im allgemeinen jedoch robuster als Teilnetze mit virtuellen Verbindungen.

Vergleich: Virtuelle Verbindungen und Datengramme

Diskussionspunkt	Verbindungslos	Verbindungsorientiert
Verbindungsaufbau	Nicht erforderlich	Erforderlich
Definition der Qualität	Mit jedem einzelnen Paket	Beim Verbindungsaufbau
Adressierung	Jedes Paket enthält die volle Quell- und Zieladresse	Jedes Paket enthält eine kurze Nummer der virtuellen Verbindung
Statusinformationen	Das Teilnetz muß keine Statusinformationen führen	Für jede virtuelle Verbindung ist ein Tabelleneintrag erforderlich
Routing	Jedes Paket wird unabhängig befördert.	Die Route wird beim Aufbau der virtuellen Verbindung gewählt; alle Pakete folgen dieser Route.
Wirkung von Router - Fehlern	Keine, außer daß Pakete verloren gehen	Alle virtuellen Verbindungen über den ausgefallenen Router werden beendet.
Überlastungsüberwachung	Schwierig	Einfach, wenn im voraus für jede virtuelle Verbindung ausreichend Puffer bereitgestellt wird.

Man sieht deutlich, daß die verschiedenen Verbindungstypen individuelle Vor- und Nachteile haben.

Eine virtuelle Verbindung benötigt eine gewisse Aufbauphase, wodurch sie für kurze Verbindungen nicht undbedingt geeignet ist. Demgegenüber ist die schnelle Behandlung der einzelnen Pakete im Zusammenhang mit der konstanten Bandbreite eine ideale Basis für Echtzeitanwendungen. Ist eine permanente, möglichst ausfallsichere, nicht zeitkritische Verbindung notwendig, sind Datengramme die bessere Lösung. Ausgefallene Router werden automatisch umgangen, so daß eine Verbindungslose Übertragung wesentlich unempfindlicher sind.

Routing-Algorithmen

Die wichtigste Aufgabe des Network-Layers besteht darin, für Pakete Routen von der Quell- zur Zielmaschine zu bestimmen. Die Routing-Algorithmen und die von ihnen verwendeten Datenstrukturen sind bei der Auslegung des Network-Layers eine der wichtigsten Aufgaben. Da ein Netzwerk möglichst ausfallsicher sein soll, werden dynamische Routingtabellen notwendig. Diese Tabellen sollten berechnet werden, da sie bei größeren Netzwerken sehr komplex sind, und bei kleineren Netzen die manuelle Pflege zu aufwendig wäre.

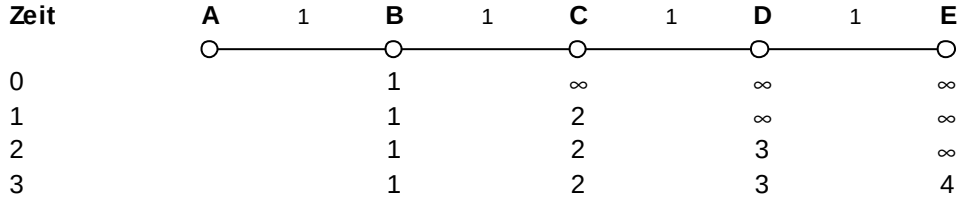
Unabhängig für verbindungsorientierte und verbindungslose Übertragungen sollte der Routing-Algorithmus genau, einfach, robust, stabil, fair und optimal arbeiten. Genauigkeit und Einfachheit bedürfen keiner weiteren Erklärung. Der Anspruch auf Robustheit mag zunächst nicht einleuchten. Wird ein großes Netz eingerichtet, geht man davon aus, daß es über Jahre hinweg ohne Systemfelder läuft. In dieser Zeit treten Hard- und Softwarefehler aller Art auf. Hosts, Router und Leitungen werden wiederholt ausgewechselt, und die Topologie ist einer ständigen Änderung unterworfen. Der Routing-Algorithmus muß mit Veränderungen dieser Art fertigwerden, ohne daß das Netz zusammenbricht. Weiterhin sollten solche Algorithmen stabil sein, es gibt manche Algorithmen, die nie einen stabilen Endzustand erreichen.

Moderne Rechnersysteme nutzen im allgemeinen dynamische Routing-Algorithmen. Am häufigsten werden *Distance-Vector-Routing* und *Link-State-Routing* verwendet.

Distance-Vector-Routing

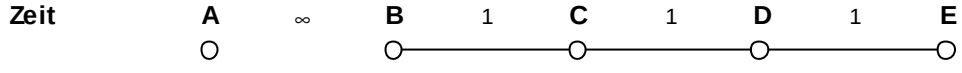
Distance-Vector-Routing ist ein Algorithmus, bei dem jeder Router eine Tabelle verwaltet, auf deren Grundlage er die am besten bekannte Entfernung zu jedem Ziel und die zu benutzende Leitung zu diesem Ziel ermittelt. Diese Tabellen

werden durch Austausch von Informationen mit den benachbarten Routern aktualisiert. Mit Hilfe dieser Tabelle wird ermittelt, ob einer der benachbarten Routern eine schnellere Strecke zu einem der Router in der bestehenden Tabelle kennt. Anhand dieser Informationen benötigt dieser Algorithmus einige Zeit bis er konvergiert und die richtige Lösung ist gefunden. Genau an dieser Stelle liegt das Problem des Distance-Vector-Routings. Es reagiert schnell auf „gute Nachrichten", braucht aber unter Umständen sehr lange, um „schlechte Nachrichten" zu verarbeiten. Dies soll ein einem Beispiel verdeutlicht werden:



Zeit	A	B	C	D	E
0		1	∞	∞	∞
1		1	2	∞	∞
2		1	2	3	∞
3		1	2	3	4

Innerhalb von drei Zyklen nach Beginn sind die optimalen Wegen nach A ermittelt. Bricht jetzt die Verbindung zwischen A und B zusammen, so beginnt eine Endlosschleife, da die Knoten B bis E der Meinung sind, sie könnten A alternativ über ihre Nachbarknoten erreichen.



Zeit	A	B	C	D	E
3		1	2	3	4
4		3	2	3	4
5		3	4	3	4
6		5	4	5	4
7		5	6	5	6

In diesem Beispiel stellt zum Zeitpunkt 4 B fest, daß er A nicht mehr direkt erreichen kann. Der Distance-Vector von C teilt ihm jedoch mit, daß C A über 2 Zeiteinheiten erreichen kann. Das diese Route ursprünglich über B führte, ist B nicht bekannt. Unendlich ist in diesem Algorithmus die Länge des längsten Pfades plus eins. Dadurch kann, besonders in großen Netzen, es einen recht Großen Zeitraum beanspruchen, bis ein solches Problem erkannt wird. Man nennt dies *Count-To-Infinity.*

Es ist natürlich möglich, das Distance-Vector-Routing zu verbessern, damit es auf obiges Verhalten besser reagiert. Es existieren aber noch mehrere andere verschiedene Konstellationen, die zum Count-To-Infinity führen. Deshalb wurde gerade für größere Netze ein alternativer Routing-Algorithmus entwickelt:

Link-State-Routing

Das Link-State-Routing entstand aus dem Distance-Vector-Routing. Anstelle des Distance-Vectors wird ein kompletter Baum des Netzes - aus der Sicht des betroffenen Routers - übermittelt. Mit diesem lassen sich alle Arten von Fehlern leicht erkennen. Es ist ein einfacher Algorithmus, der in fünf Teilen beschrieben werden kann:

1. Jeder Router muß seine Nachbarn entdecken und ihre Netzadressen feststellen.
2. Zu diesem Routern werden die Verzögerungen oder die Kosten ermittelt.
3. All diese Informationen werden in einem Paket zusammengefaßt.
4. Dieses Paket wird an alle anderen Router gesendet.
5. Aus diesen zusammengestellten Informationen berechnet jeder Router sich den kürzesten Pfad zu allen anderen Routern. Dies könnte beispielsweise mit Dijkstras Algorithmus geschehen.

IP - Internet Protocol

Innerhalb des Network Layers kann das Internet als Sammlung von Teilnetzen - sogenannten *autonomen Systemen* - betrachtet werden, die alle untereinander verbunden sind. Es gibt keine echte Struktur, sondern mehrere größere

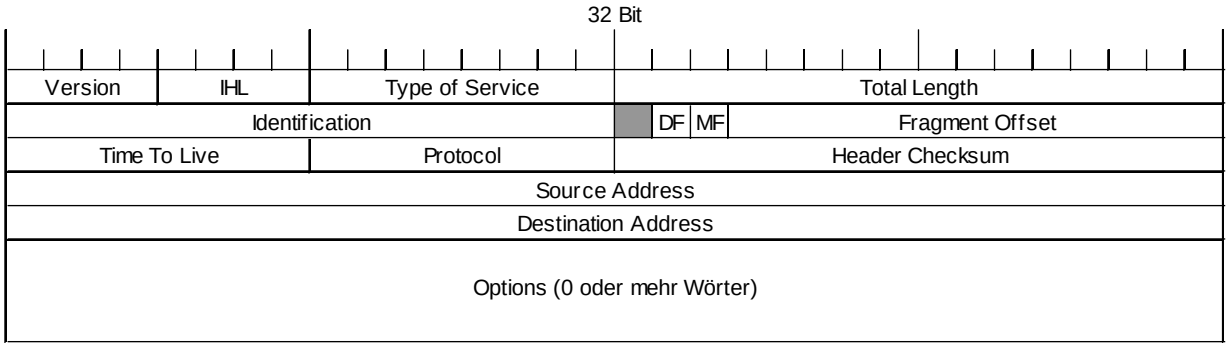
Backbones. Diese Backbones werden aus Leitungen mit hoher Bandbreite und schnellen Routern gebildet. An die Backbones werden regionale Netze und an diese die LANs von Universitäten, Unternehmen und Internet-Service-Providern angeschlossen.

Der Klebstoff, der das alles zusammenhält, ist das Protokoll der Network Layers - IP (Internet Protocol). Im Gegensatz zu den meisten älteren Protokollen des Network Layers wurde es von Anfang an mit Blick auf Netzverbund ausgelegt. Seine Aufgabe ist die Bereitstellung einer Möglichkeit, nach bestem Bemühen, Datengramme von der Quelle zum Ziel zu befördern, ungeachtet dessen, ob sich diese Maschinen im gleichen Netz befinden. Wichtigstes Merkmal des IP ist, daß es verbindungslos ist, also weder eine Sicherung besitzt, noch dafür sorgt, daß die einzelnen Datengramme in der richtigen Reihenfolge beim Empfänger ankommen.

Aufbau eines IP Paketes

IP verwaltet alle Daten im Big-Endian Format, also „most-significant-byte-first“. Da Intel’s x86-Architektur ist hier eine Konvertierung erforderlich, da diese die Daten im Little-Endian, also genau umgekehrt, verwalten.

Ein IP - Header besitzt folgenden Aufbau:



Das *Version*-Feld gibt die Versionsnummer des Protokolls an. Dadurch wird die Migration verschiedener Protokolle möglich, da ein Protokollwechsel von mehreren Monaten bis hin zu mehreren Jahren dauern kann.

Da die Header-Länge nicht konstant ist, wird das Feld *IHL (Internet Header Length)* bereitgestellt. IHL gibt die Header-Länge in 32-Bit Wörtern an. Der Mindestwert ist 5, was zutrifft, wenn keine Optionen vorhanden Sind. Der Höchstwert dieses 4-Bit-Feldes ist 15, was den Header auf 60 Byte und damit das *Options*--Feld auf 40 Byte begrenzt.

Type Of Service wird in der Praxis von den heute eingesetzten Routern gänzlich ignoriert.

Total Length gibt die gesamte Länge des Datengramms an, inklusive des Headers. Die Höchstgrenze ist 65.535 Byte.

Identification wird benötigt, damit der Zielhost feststellen kann, zu welchem Datengramm ein neu angekommenes Fragment gehört. Alle Fragmente eines Datengramms enthalten den gleichen Wert.

Mit Hilfe des Flags *DF (Don’t Fragment)* Flags kann verhindert werden, daß das IP Paket in mehrere Fragmente zerlegt wird. Dies ist beispielsweise notwendig, wenn der Zielhost fragmentierte Datengramme nicht wieder zusammensetzen kann.

Hingegen zeigt *MF (More Fragments)* an, daß dieses Datengramm nur ein Teil eines größeren, fragmentierten Datengramms ist. Dieses Bit ist bei allen außer dem letzten Fragment gesetzt, damit erkannt werden kann, wann alle Fragmente eingetroffen sind.

Fragment Offset bezeichnet, an welcher Stelle im Datengramm ein Fragment gehört. Dieser Wert muß mit 8 multipliziert werden, damit die korrekte Position innerhalb eines Datengramms wieder ermittelt werden kann.

Das Feld *Time To Live* ist ein Zähler, mit dem die Lebensdauer von Paketen begrenzt werden kann. Erfaßt werden soll die Lebensdauer in Sekunden. Zulässig ist eine maximale Lebensdauer von 255 s. Der Zähler muß bei jeder Teilstrecke gesenkt werden, Bei einer längeren Wartezeit im Router mehrmals. In der Praxis werden nur die Teilstrecken gezählt. Erreicht dieser Zähler den Wert Null, wird das Paket verworfen und ein Warnpaket wird an den Quellhost gesendet. Durch dieses Merkmal werden Datengramme daran gehindert, ewig herumzuschwirren, was manchmal passiert, wenn fehlerhafte Routing-Tabellen vorliegen.

Wurde ein komplettes Datengramm zusammengestellt, muß der Network Layer wissen, wohin das Datengramm weiterzuleiten ist. Durch das Feld *Protocol* kann er erkennen, an welchen Transportprozeß es weiterzugeben ist. Die Numerierung der Protokolle ist durch RFC 1700 festgelegt.

Das Feld *Header Checksum* prüft nur den Header. Der Algorithmus addiert alle 16-Bit-Halbwörter bei ihrer Ankunft und

bildet das Komplement. Bei dieser Berechnung wird das Prüfsummenfeld als Null angenommen.

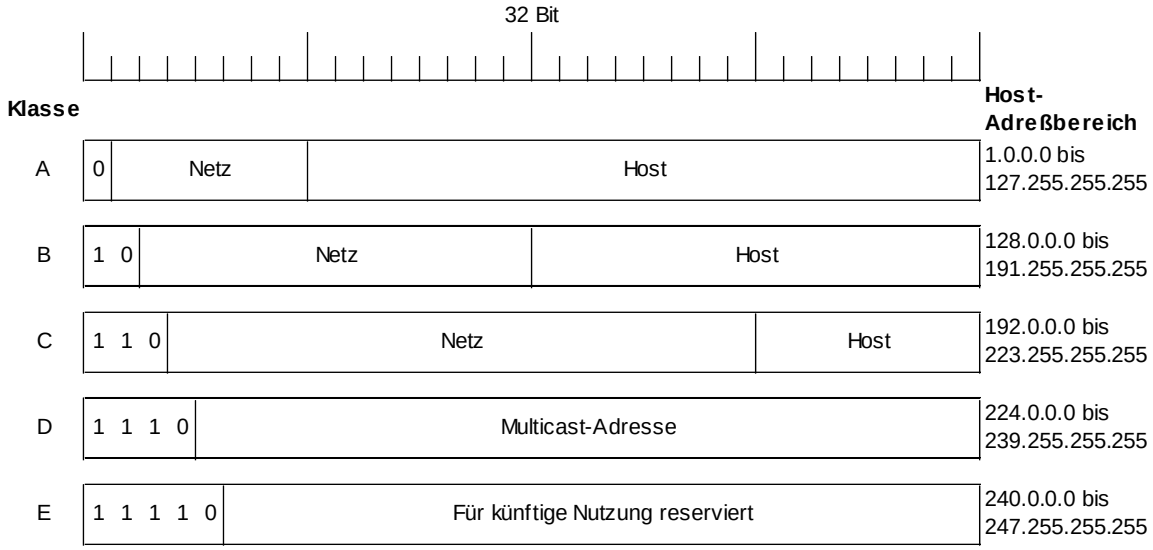
Source und Destination Address werden im nächsten Abschnitt besprochen.

Innerhalb der *Options-Felder* können für spezielle Aufgaben weitere Felder angefügt werden.

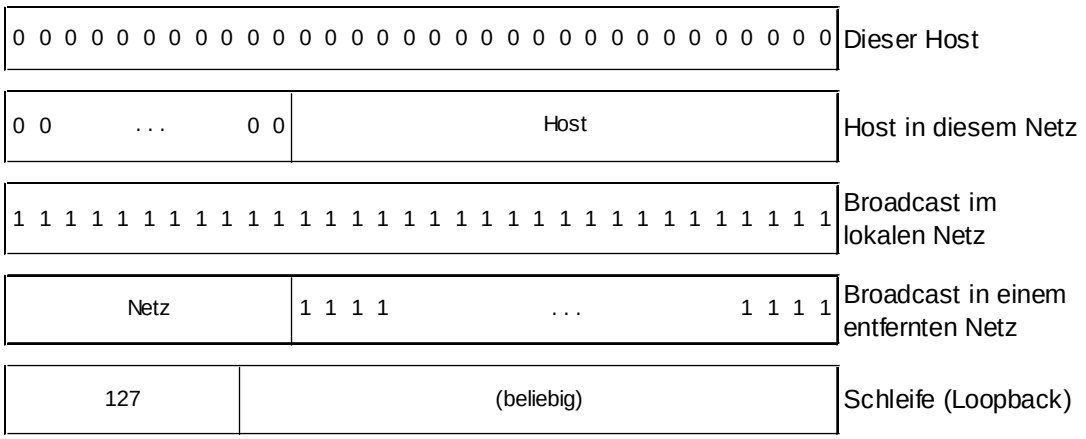
Adressierung im Internet

Jeder Host im Internet hat seine eigene IP-Adresse, die Netz- und Hostnummer kodiert. Die Kombination ist weltweit eindeutig. IP-Adressen sind 32 Bit lang und werden in den Felder Source und Destination Address von IP-Paketen benutzt. Maschinen , die an mehrere Netze angeschlossen sind, haben in jedem Netz eine ander IP-Adresse. Die 32 Bit einer IP-Adresse werden für gewöhnlich ihn vier Bytes dezimal dargestellt: Die (hexadezimale) Adresse C0290614 wird also im Format 192.41.6.20 geschrieben.

Die für die IP-Adresse benutzten Formate werden in folgender Übersicht dargestellt:



Die Adressen der Klassen-Formate A, B, C und D erlauben die Bildung von bis zu 126 Netzen mit 16 Millionen Hosts, 16.382 Netzen mit 64.000 Hosts, 2 Millionen Netze (beispielsweise LANs) mit 254 Hosts und Multicast, durch das ein Datengramm an mehrere Hostadressen gesendet werden kann. Dies wird derzeit nicht genutzt und steht zur künftigen Verwendung zur Verfügung. Derzeit sind zehntausende von Netzen im Internet verbunden, und die Zahl verdoppelt sich jedes Jahr. Die Netznummern werden vom *NIC (Network Information Center)* zugewiesen.



Die Werte 0 und -1 haben eine besondere Bedeutung. Der Wert 0 bezeichnet ein bestimmtes Netz oder einen bestimmten Host. Der Wert -1 wird als Broadcast-Adresse benutzt und richtet sich an alle Hosts im bezeichneten Netz.

Die IP - Adresse 0.0.0.0 wird von Hosts benutzt, wenn sie gestartet werden, danach aber nicht mehr. IP - Adressen mit

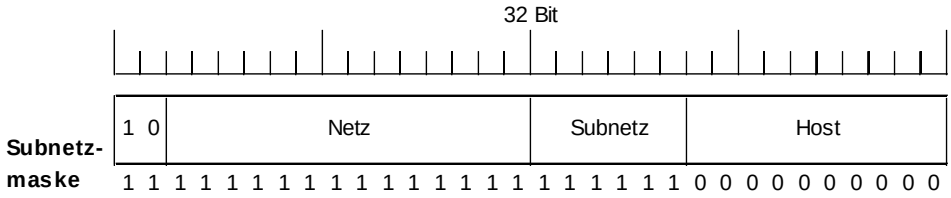
0 als Netznummer beziehen sich auf das aktuelle Netz. Mit der nur aus Einsen bestehenden Adresse ist Broadcasting im lokalen Netz möglich, das normalerweise ein LAN ist.

Teilnetze

Alle Hosts in einem Netz müssen die gleiche Netznummer haben. Dieses Merkmal der IP-Adressierung kann mit zunehmender Größe von Netzen Probleme verursachen, zum Beispiel bei einem Unternehmen, das mit einem LAN der Klasse C im Internet beginnt und im Laufe der Zeit mehr als 254 Maschinen anschließt, so daß eine zweite Adresse der C-Klasse erforderlich ist.

Je mehr die Zahl der einzelnen lokalen Netze zunimmt, desto schwieriger gestaltet sich deren Verwaltung. Jedesmal, wenn ein neues Netz installiert wird, muß der Systemverwalter mit dem NIC Kontakt aufnehmen, um eine neue Netznummer einzuholen. Um dieses Problem zu minimieren, kann ein Netz für interne Verwendungszwecke in mehrere Teilnetze (Subnetz, Subnet) aufgeteilt werden. Diese Aufteilung bietet sich besonders bei Netzen der Klasse A und B an, da dort die Zahl der Hosts sehr groß sein kann.

Es entstehen Subnetz-Masken, anhand denen Pakete geroutet werden. Dadurch verkleinern sich auch die Routing-Tabellen. Bei der Übermittlung eines IP - Paketes wird zuerst überprüft, ob sich das Ziel im gleichen Subnetz wie der Router befindet. Wenn ja, wird das Paket wie gewöhnlich an den Host weitergeleitet, wenn nicht wird es an einen Router, der Verbindung zu den anderen Netzen hat („Default Gateway“) weitergeleitet.



Es kann auf diese Weise leicht mit Hilfe von boolschen Operation die Netz- oder Hostadresse extrahiert werden.

ICMP - Internet Control Message Protocol

Der Betrieb des Internets wird eng von den Routern überwacht. Geschieht etwas Unerwartetes, wird das Ereignis mittels ICMP berichtet. ICMP, das auch für Testzwecke verwendet wird, hat di IP Protokollnummer 1. Folgende Ereignisse sind für ICMP definiert:

Nachrichtentyp	Beschreibung
Destination Unreachable	Paket kann nicht zugestellt werden
Time Exceeded	Das Feld Time To Live hat den Wert 0
Parameter Problem	Ungültiges Header-Feld
Source Quench	Chocke-Paket
Redirect	Bringt dem Router etwas über Geographie bei
Echo Request	Fragt eine Maschine, ob sie noch am Leben ist
Echo Reply	Maschine antwortet, daß sie noch am Leben ist
Timestamp Request	Wie Echo Request, aber mit Zeitstempel
Timestamp Reply	Wie Echo Reply, aber mit Zeitstempel

Die Nachricht *Destination Unreachable* wird benutzt, wenn das Teilnetz oder ein Router das Ziel nicht finden kann oder ein Paket mit dem DF-Bit nicht zugestellt werden kann.

ARP - Address Resolution Protocol

Jede Maschine im Internet hat zwar eine (oder mehrere) IP-Adressen, jedoch können diese Adressen nicht zum Versenden von Paketen benutzt werden, weil die Hardware auf der Sicherungsschicht Internet-Adressen nicht versteht. Inner-

halb dieser Schichten werden die 48 Bit breiten MAC-Adressen benötigt. An dieser Stelle kommt ARP zum Zuge, mit dessen Hilfe die IP-Adressen in MAC-Adressen aufgelöst werden können. Dazu wird eine Broadcast Nachricht mit der gesuchten IP-Adresse versandt. Ist der gesuchte Host im gleichen Subnetz, so antwortet er auf diese Nachricht. Nach diesem Vorgang kennen sich beide Hosts gegenseitig und die MAC-Adressen werden im „ARP-Cache“ für eine gewisse Zeit gehalten. Ist der gesuchte Rechner nicht im gleichen LAN, werden zwei Lösungen genutzt. Mit Hilfe der Default Gateway wird festgelegt, welcher Rechner Pakete für andere Netze in Empfang nimmt. Ist die Unterteilung in mehrere Subnetze mit Gateways nicht möglich, so muß ein Router so konfiguriert sein, daß er auf ARP Anfragen mit seiner eigenen MAC Adresse antwortet, da Router LAN-Broadcasts nicht weiterleiten. Diese Lösung wird Proxy ARP genannt.

DHCP - Dynamic Host Configuration Protocol

DHCP kann dazu genutzt werden, Rechnerkonfigurationen dynamisch festzulegen. Während der Netzwerk-Initialisation wird innerhalb des LANs mit Hilfe eines Broadcasts ein DHCP-Host gesucht, von dem daraufhin alle notwendigen TCP/IP Konfigurationsdaten wie IP-Adresse, DNS-Adresse oder Subnetz-Maske dem Rechner übermittelt. Dies wird von Internet-Provider dazu genutzt, mehr Kunden, als vorhandene IP-Adressen zu bedienen. Dazu wird einem Kunden während der Einwahl eine IP-Adresse zugeordnet, die für eine gewisse Dauer Gültigkeit besitzt.

Routing im Internet

Das Internet setzt sich aus einer großen Zahl autonomer Systeme zusammen. Jedes autonome System wird von einem anderen Unternehmen beschrieben und kann intern einen eigenen Routing-Algorithmus verwenden. Dennoch gibt es Standards, auch für internes Routing, durch das sich die Implementierung an den Grenzen zwischen autonomen Systemen vereinfacht. Zuerst wird das Routing innerhalb eines autonomen Systems behandelt. Ein solcher Routing-Algorithmus wird als *internes Gateway Protokoll (Interior Gateway Protocol)* bezeichnet. Wird ein Routing zwischen autonomen Systemen notwindig, wird ein *externes Gateway Protokoll (Exterior Gateway Protocol)* verwendet.

OSPF - Internes Gateway Protokoll (Interior Gateway Protocol)

Ursprünglich wurde als internes Gateway Protokoll ein Distance Vector Routing benutzt. Dieses - RIP (Routing Information Protocol) genannt - hatte seinen Ursprung im BSD UNIX. Da dieses Routing nicht die Probleme des Distance Vector Routing löste, wurde 1990 der Nachfolger von RIP, OSPF (*Open Shortest Path First)* eingeführt und zum neuen Standard erklärt. Die vollständige Beschreibung des OSPF findet sich in RFC 1247. Nachfolgend sollen die wichtigsten sieben Punkte von OSPF kurz erkläutert werden.

1. Der Algorithmus sollte veröffentlicht werden, also ein offener Standard werden. Eine herstellereigene Lösung eines bestimmten Unternehmens war nicht erwünscht.
2. Das neue Protokoll sollte eine Vielzahl von Entfernungsparametern unterstützen, darunter die physische Entfernung, Verzögerung, Koste usw.
3. Es sollte ein dynamischer Algorithmus sein, der sich schnell und automatisch an Änderungen der Topologie anpaßt.
4. OSPF sollte durch die Art des verlangten Dienstes bestimmt werden. Beispielsweise sollten Echtzeit-Sendungen auf eine Art, sonstige Sendungen auf eine andere Art behandelt werden können. IP besitzt zwar das Feld *Type Of Service*, aber es wird von keinem Protokoll genutzt.
5. Es sollte eine Möglichkeit für einen gewissen Lastenausgleich bieten, d.h. die Last auf mehrere Leitungen verteilen. Die meisten früheren Protokolle schickte alle Pakete über die beste Route. Die zweitbeste Route wurde überhaupt nicht benutzt. In vielen Fälle führt die Aufteilung der Last auf mehrere Leitungen zu einem besseren Ergebnis.
6. Die Unterstützung für hierarchische Systeme war gefordert. Im Jahr 1988 war das Internet schon derartig angewachsen, daß von keinem Router erwartet werden konnte, die gesamte Topologie zu kennen. Das neue Routing-Protokoll sollte so ausgelegt werden, daß das von vornherein von keinem Router erwartet werden muß.
7. Gewisse Sicherheitsmaßnahmen sollten geboten werden, um Leute mit seltsamen Humor daran zu hindern, Router durch Übersenden falscher Routing-Daten ganz aus dem Konzept zu bringen. Schließlich waren Vorkehrungen nötig, um Router zu handhaben, die an das Internet über einen Tunnel angeschlossen sind. Frühere Protokolle haben diese Aufgabe nicht gut bewältigt.

Die Komplexität und Leistungsfähigkeit von OSPF macht es besonders geeignet für größere Netzwerke. Drei Arten von Netzverbindungen werden von OSPF unterstützt:

1. Punkt-zu-Punkt-Leitungen zwischen genau zwei Routern
2. Mehrfachzugriffsnetze mit Broadcasting (das sind die meisten LANs)

3. Mehrfachzugriffsnetze ohne Broadcasting (das sind die meisten paketvermittelten LANs)

BGP - Externes Gateway Protokoll (Exterior Gateway Protocol)

Innerhalb eines autonomen Systems ist das empfohlene Routing-Protokoll im Internet OSPF, obwohl es bei weitem nicht das einzige ist. Zwischen autonomen Systemen wird das BGP (*Border Gateway Protocol)* verwendet. Zwischen autonomen Systemen wird ein anderes Protokoll benötigt, da interne und externe Gateway Protokolle nicht die gleichen Ziele verfolgen. Ein internes Gateway Protokoll muß lediglich Pakete so effizient wie möglich von der Quelle zum Ziel befördern. Es muß sich um keinerlei Regeln kümmern.

Demgegenüber haben Router mit externen Gateway Protokollen die bestehenden Regeln sehr wohl zu beachten. Das autonome System eines Unternehmens muß eventuell die Fähigkeit aufweisen, Pakete an jeden beliebigen Internet-Standort zu senden und von jedem Internet-Standort zu empfangen. Eventuell ist das Unternehmen nicht willens, Pakete auf der Durchfahrt von einem fremden zu einem anderen, fremden autonomen System zu befördern, obwohl es für die beiden fremden Systeme der kürzeste Weg wäre. Andererseits ist es vielleicht bereit, Transitverkehr für seine Nachbarn oder bestimmte, andere autonome Systeme zu übernehmen, falls sie für diesen Dienst zahlen. Folglich müssen externe Gateway Protokolle in der Lage sein, viele Arten von Routing-Regeln zu unterstützen.

Zu den typischen Regeln gehören politische, wirtschaftliche oder sicherheitstechnische Maßnahmen, beispielsweise:

1. Kein Transitverkehr durch bestimmte autonome Systeme
2. Vom Pentakom ausgehender Datenverkehr darf nie über Irak übertragen werden
3. Datenverkehr von British Columbia nach Ontario darf nicht über die USA fürhen
4. Albanien darf nur durchquert werden, wenn es keine Alternative zum Ziel gibt
5. Datenverkehr, der bei IBM startet und endet, darf nicht über Microsoft führen.

Solche Regeln und Maßnahmen werden innerhalb eines jeden BGP - Routers konfiguriert. Sie sind nicht Teil des Protokolls.

BGP ist grundsätzlich ein Distance-Vector Protokoll, unterscheidet sich aber von anderen solchen, z.B, RIP, beträchtlich. Es verwaltet nicht nur die Kosten an jedem Ziel, sondern führt auch über den benutzten Pfad genau Buch. Anstatt die Angaben periodisch mit den geschätzten Kosten jedes möglichen nächsten Ziels zu versorgen, teilt jeder BGP-Router seinen Nachbarn genau den Pfad mit, den er benutzt. Auf diese Weise können nutzlose Routen sofort verworfen werden, wenn sie beispielsweise über sich selbst führen.

Auch das Count-To-Infinity-Problem wird von BGP mit Leichtigkeit gelöst. Fällt ein Router aus, so kann das Protokoll dies durch vergleich der benutzten Routen erkennen. Da im Fehlerfalle alle falschen Routen über den besagten Router selbst führen, werden sie von BGP sofort verworfen, und es wird eine alternative Route gesucht.

CIDR - Classles InterDomain Routing

IP wird seit über einem Jahrzehnt umfangreich eingesetzt und hat sich in dieser Zeit bewährt. Dies führte dazu, daß IP daß Opfer seiner eigenen Beliebtheit geworden ist. Das Problem liegt, einfach ausgedrückt, darin, daß im Internet bald keine IP-Adressen mehr verfügbar sind. Im Grunde gibt es über 2 Milliarden Adressen. Durch das Vorgehen, den Adressraum in Klassen zu organisieren werden aber Millionen davon vergeudet. Ein besonderer Übeltäter ist hierbei die Klasse B. Für die meisten Unternehmen ist ein Netz der Klasse A mit 16 Millionen Adressen zu groß, dasjenige der Klasse C mit 256 Adressen aber zu klein. Folglich nutzen viele Unternehmen Netze der Klasse B mit seinen 65.536 Adressen.

In Wirklichkeit ist eine Adresse der Klasse B für die meisten Unternehmen aber viel zu groß. Untersuchungen haben gezeigt, daß mehr als die Hälfte aller Klasse B Netze weniger als 50 Hosts haben. Zweifellos hatten solche Unternehmen das eventuell irgendwann zu knappe 8-Bit-Hostfeld vor Augen. Rückblickend wäre es besser gewesen, man hätte die Klasse C mit einem 10-Bit-Feld für die Hostnummer ausgestattet, was 1.022 Hosts pro Netz ermöglicht hätte.

Allerdings kommt genau an dieser Stelle ein weiteres Problem ins Spiel: Die Routing-Tabellen währen explodiert. Aus Sicht der Router ist der IP-Adressraum eine zweistufige Hierarchie mit Netz- und Hostnummern. Router müssen nicht alle Hosts kennen, aber alles über das Netz wissen. Wäre heute eine halbe Million Netze der Klasse C in Gebrauch, müßte jeder Router im gesamten Internet eine Tabelle mit einer halben Million Einträgen führen, d.h. für jedes Netz einen, mit der zu benutzenden Leitung und den entsprechenden, weiteren Informationen.

Es existieren viele Lösungsansätze für diese Problem, nur schaffen die meisten dafür ein neues. Eine Lösung, die derzeit implementiert wird und dem Internet wahrscheinlich ein wenig mehr Atemfreiheit bringt, ist *CDIR (Classless InterDomain Routing)*. CDIR ist in RFC 1519 beschrieben und basiert auf dem Konzept, daß die verbleibenden Netze der Klasse C, von denen es fast zwei Millionen gibt, in Blöcken mit variabler Länge zugewiesen werden. Braucht ein

Standort beispielsweise 2.000 Adressen, erhält er einen Block von 2.048 Adressen, also acht aufeinanderfolgende Netze der Klasse C, und nicht eine volle B-Klasse.

Zusätzlich zur Verwendung von Blöcken aufeinanderfolgender Netze der Klasse C als Einheiten, wurden auch die Zuweisungsregeln für Adressen der Klasse C in RFC 1519 geändert. Die Welt wurde dabei in vier Zonen aufgeteilt. Jede Zonen erhält einen Teil des Adressraums der Klasse C. Die Zuweisung ist wie folgt:

Adressen 194.0.0.0 bis 195.255.255.255 für Europa

Adressen 198.0.0.0 bis 199.255.255.255 für Nordamerika

Adressen 200.0.0.0 bis 201.255.255.255 für Mittel- und Südamerika

Adressen 202.0.0.0 bis 203.255.255.255 für Asien und den pazifischen Raum

Auf diese Weise erhält jede Region etwa 32 Millionen Adressen zur Zuweisung. Weitere 320 Millionen Adressen der Klasse C von 204.0.0.0 bis 223.255.255.255 bleiben als Reserve zur künftigen Nutzung. Diese Zuweisung hat den Vorteil, daß jetzt jeder Router außerhalb Europas, der ein mit 194.x.y.z oder 195.x.y.z adressiertes Paket erhält, das Paket einfach an sein europäisches Standard-Gateway senden kann. Im Grunde wurden 32 Millionen Adressen zu einem Routing-Tabelleneintrag komprimiert.

Selbstverständlich sind nach wie vor ausführliche Routingtabellen erforderlich. Die naheliegenste Möglichkeit wären 131.072 Einträge für die oben genannten Netze, aber das wäre genau die Explosion der Routing-Tabellen, die ja vermieden werden soll. Statt dessen wird jeder Routing-Tabelleneintrag mit einer 32-Bit-Maske erweitert. Bei Ankunft eines Paketes wird zuerst seine Zieladresse herausgezogen. Dann wird die Routing-Tabelle (konzeptionell) Eintrag für Eintrag abgetastet, wobei die Zieladresse „maskiert“ und mit den Tabelleneinträgen verglichen wird, um einen Treffer zu ermitteln. Auf diese Weise wird erreicht, daß lediglich der primäre Router eines jeden Netzes bekannt sein muß, der sich danach um die Weitervermittlung des Paketes kümmern kann.

Dieses Konzept kann selbstverständlich für alle Netzklassen, nicht nur für die Klasse C, angewandt werden. Die ursprüngliche Klasseneinteilung der IP-Adressen geht somit verloren. Aus diesem Grund wird CIDR „klassenloses Routing“ genannt.

IP Version 6 (IPv6)

CIDR schafft zwar wieder etwas Luft für ein paar Jahre, jedoch ist sich jeder dessen bewußt, daß die Tage von IP in seiner heutigen Form (IPv4) gezählt sind. Mit dem ungeahnten Zustrom auf das Internet seit Mitte der Neunziger Jahre und dem entsprechenden, explosiven Wachstum wird es aller Wahrscheinlichkeit nach im nächsten Jahrtausend von einer viel größeren Gruppe von Menschen benutzt werden. Allein die bevorstehende Verschmelzung der Computer-, Kommunikations- und Unterhaltungsindustrie wird dazu führen, daß bald jeder Fernseher der Welt zum Internet-Knoten wird. Unter diesen Umständen ist es gänzlich augenscheinlich, daß sich das IP weiterentwickeln und flexibler werden muß.

Angesichts dieser Probleme am Horizont begann 1990 die Arbeit an einer neuen Version von IP. Angestrebt wird eine Version, der nie die Adressen ausgehen, die eine Vielzahl anderer Probleme löst und die auch flexibler und effizienter ist. Die wesentlichen Ziele des Projekts waren:

1. Unterstützung von Milliarden von Hosts, auch bei ineffizienter Zuweisung des Adreßraums
2. Reduzierung des Umfangs der Routing-Tabellen
3. Vereinfachung des Protokolls, damit Router Pakete schneller abwickeln können
4. Höhere Sicherheit (Authentifikation und Datenschutz) als das heutige IP
5. Mehr Gewicht auf Dienstarten, insbesondere für Echtzeitanwendungen
6. Unterstützung von Multicasting durch die Möglichkeit, den Umfang zu definieren
7. Möglichkeit für Hosts, ohne Adressänderung auf Reisen zu gehen
8. Möglichkeit für das Protokoll, sich künftig weiterzuentwickeln
9. Unterstützung der alten und neuen Protokolle in Koexistenz für Jahre

IPv6 erfüllt diese Ziele ziemlich gut. Es hat die guten Eigenschaften von IP, wurde von den schlechten befreit und umfaßt neue. Im allgemeinen ist IPv6 nicht mit IPv4 kompatibel, wohl aber mit den anderen Internet-Protokollen, darunter TCP, UDP, ICMP, IGMP, OSPF, BGP und DNS, eventuell nach geringen Modifikationen (die vorwiegend mit den längeren Adressen zu tun haben). Die wichtigsten Merkmale von IPv6 werden im folgenden beschrieben. Ausführliche Informationen sind in RFC 1883 bis RFC 1887 enthalten.

Als wichtigstes Merkmal hat IPv6 gegenüber IPv4 längere Adressen. Sie sind 16 Byte lang, was das Problem löst, für das IPv6 überhaupt entwickelt wurde. Das Thema Adressen wird gleich noch einmal aufgegriffen.

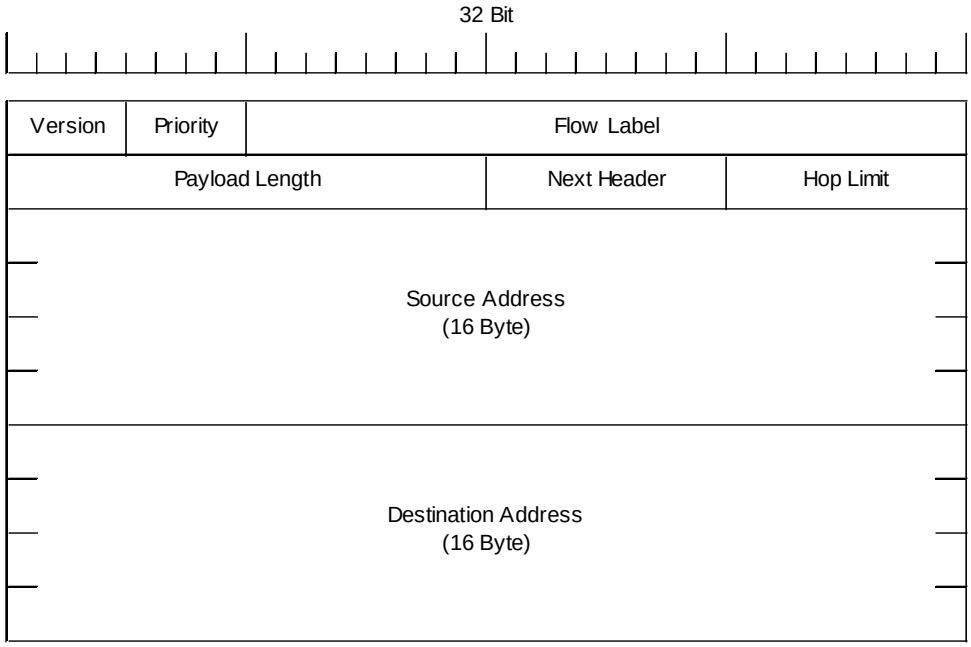
Die zweite Verbesserung von IPv6 ist die Vereinfachung des Headers. Er enthält nur 7 Felder (gegenüber 13 in IPv4). Diese Änderung ermöglicht Routern, Pakete schneller zu verarbeiten. Damit verbessert sich der Durchsatz.

Die dritte wichtige Verbesserung ist eine bessere Unterstützung von Optionen. Diese Änderung war für den neuen Header erforderlich, weil vormals zwingende Felder nun optional sind. Darüber hinaus unterscheidet sich auch die Art, wie Optionen dargestellt werden. Für Router ist es damit einfacher, Optionen, die sie nichts angehen, zu überspringen. Dieses Merkmal beschleunigt ebenfalls die Verarbeitungszeit von Paketen.

Ein vierter Bereich, in dem IPv6 deutliche Vorteile bringt, ist die Sicherheit. Authentifikation und Datenschutz werden ermöglicht.

Schließlich wurde auf die Dienstarten mehr Gewicht gelegt als in der Vergangenheit. IPv4 hat für diese Angelegenheit ein 8-Bit-Feld, was mit Blick auf Multimedia Verkehr nicht ausreicht.

Der IPv6 Hauptheader



Das Feld *Version* enthält jetzt immer den Wert 6, wodurch eine Unterscheidung der beiden Protokolle leicht möglich ist. Dies ist besonders im Hinblick auf die schätzungsweise zehn Jahre lange Umstellungszeit notwendig.

Mit Hilfe von *Priority* läßt sich zur Kontrolle der Flußsteuerung einsetzen. Werte von 0 bis 7 gelten für Übertragungen, die sich im Fall von Überlastungen verlangsamen. Die Werte 8 bis 15 gelten für Echtzeitverkehr, der in einer konstanten Rate übertragen werden muß.

Flow Label, derzeit noch in der Experimentierphase, soll aber benutzt werden, um es einer Quelle und einem Ziel zu ermöglichen, eine Pseudoverbindung mit bestimmten Merkmalen und Anforderungen aufzubauen. Flüsse sind im Grunde der Versuch, die Flexibilität eines Datengrammteilnetzes und die Garantien eines Teilnetzes mit virtuellen Verbindungen zu bieten.

Payload Length gibt an, wie viele Bytes dem 40 Byte langen Header folgen.

Das Feld *Next Header* läßt die Katze aus dem Sack. Der Grund, warum der Header vereinfacht werden konnte, liegt daran, daß es zusätzliche (wahlweise) Erweiterungs-Header gibt. Dieses Feld gibt an, welcher der (derzeit) sechs Erweiterungs-Header diesem folgt, falls überhaupt. Ist das der letzte IP - Header, gibt das Feld *Next Header* an, welchen Handler des Transportprotokolls (z.B. TCP, UDP) das Paket passiert.

Hop Limit wird benutzt, um Pakete am ewigen Leben zu hindern. Das entspricht dem Feld Time To Live aus IPv4, wurde jedoch umbenannt, um die tatsächliche Nutzung wiederzuspiegeln.

Source und *Destination Address* geben Quelle und Ziel in den 16 Byte langen IPv6 Adressen an.

Adressraum

IPv6 unterscheidet viele verschiedene Adresspräfixe, von denen hier nur vier Gruppen kurz angesprochen werden sollen:

Ein für IPv4 reservierter Adressraum, der für die Übergangszeit und zur Kompatibilität notwendig ist.

Adressräume für Internet Sevice-Provider, wodurch die einzelnen ISP Netze einfach erkannt und voneinander unterschieden werden können.

Das geographische Modell entspricht dem im heutigen Internet, bei dem Provider keine große Rolle spielen. Auf diese Weise kann IPv6 beide Adressarten handhaben.

Eine Multicast Adressgruppe zum ansprechen einer kompletten Host Gruppe auf einmal.

Die neuen, 16 Byte breiten, Adressen sind eine radikale Neuerung: Es gibt 2^{128}davon, was ungefähr 3×10^{38} entspricht. Wäre die ganze Erde, einschließlich der Meere, mit Computern bedeckt, würde IPv6 7×10^{23} IP-Adressen pro Quadratmeter bieten. Chemiestudenten werden feststellen, daß diese Zahl größer ist, als die von Avogadro. Obwohl es nicht beabsichtigt war, jedem Molekül auf der Erdoberfläche eine eigene IP-Adresse zu geben, sind wir nicht mehr weit davon entfernt. In der Praxis wird allerdings auch dieser Adressraum nicht effizient genutzt. In einem extrem pessimistischen Szenario könnten immer noch weit über 1.000 Adressen pro Quadratmeter der Erdoberfläche (Land und Wasser) genutzt werden. Übrigens wurden bisher erst 28% des Adreßraums zugeordnet. Die übrigen 72% sind für künftige Verwendungszwecke, die wir uns heute noch gar nicht vorstellen können, verfügbar.

Erweiterungsheader

In manchen Situationen sind einige der fehlenden Felder trotzdem notwendig, so daß IPv6 das Konzept eines (optionalen) *Erweiterungsheader* beinhaltet. Diese header können benutzt werden, um zusätzliche Informationen bereitzustellen, sie werden aber auf effiziente Weise kodiert. Derzeit sind sechs Erweiterungsheader definiert. Alle sind optional, werden aber mehrere Benutzt, müssen sie direkt nach dem festen Header vorzugsweise in der aufgeführten Reihenfolge erscheinen.

Erweiterungs-Header	Beschreibung
Optionen für Teilstrecken (Hop-by-Hop)	Verschiedene Informationen für Router, beispielsweise für Datengramme, die über 64 kByte hinausgehen
Routing	Definition einer vollen oder teilweisen Route
Fragmentierung	Verwaltung von Datenfragmenten
Authentifikation	Echtheitsüberprüfung des Senders
Verschlüsselte Sicherheitsdaten	Informationen über den verschlüsselten Inhalt
Optionen für Ziele	Zusätzliche Informationen für das Ziel

Transport Layer

Der Transport Layer ist nicht einfach eine weitere Schicht. Sie ist der Kern der gesamten Protokollhierarchie. Diese Schicht hat die Aufgabe, den Transport der Daten vom Quell- zum Zielrechner unabhängig von den physischen Netzen zuverlässig und kostengünstig zu übernehmen. Ohne den Transport Layer wäre das gesamte Konzept der Schichtenprotokolle wenig sinnvoll.

Der Transport Layer im Internet

Das Internet hat innerhalb des Transport Layers zwei Protokollarten, eine verbindungsorientierte und eine verbindungslose. In den folgenden Abschnitten werden beide Protokollarten beschrieben. Das verbindungsorientierte Protokoll ist TCP, das verbindungslose ist UDP. Da UDP im Grunde ein IP mit einem kürzeren Header ist, konzentrieren wir uns auf TCP.

TCP - Transmission Control Protocol

TCP wurde spezifisch zur Bereitstellung eines zuverlässigen Bytestroms von Ende zu Ende in einem unzuverlässigen Netzverbund entwickelt. Ein Netzverbund unterscheidet sich von einem Einzelnetz dahingehend, daß verschiedene Teile eventuell total unterschiedliche Topologien, Bandbreiten, Verzögerungen, Paketgrößen und andere Parameter haben. TCP wurde entwickelt, um die verschiedenen Merkmale von Verbundnetzen dynamisch anzupassen und das gesamte Gebilde robuster zu machen. TCP wurde formell in RFC 793 definiert.

Das TCP - Dienstmodell

Wie bereits erwähnt, soll TCP eine gesicherte Verbindung zur Verfügung stellen. Dies ist der wesentliche Teil von TCP.

Weiterhin soll der zu übertragende Bytestrom in Pakete unterteilt werden. Die Länge dieser Pakete hängt von den Eigenschaften des Netzwerks ab.

Da die IP-Schicht keinerlei Gewähr gibt, daß Datengramme richtig zugestellt werden, obliegt es TCP, ein Timeout zu veranlassen und die Daten bei Bedarf erneut zu übertragen. Datengramme können aber andererseits auch in einer fehlerhaften Reihenfolge ankommen, weshalb sie von TCP wieder in die Richtige Reihenfolge gebracht werden müssen.

Um einen TCP-Dienst nutzen zu können, müssen der Sender und der Empfänger einen Endpunkt namens *Socket* erstellen. Jeder Socket hat eine Socketnummer, die aus der IP-Adresse des Hosts und einer 16-Bit-Nummer für den lokalen *Port* des Hosts bereitsteht. Für den Zugriff auf einen TCP-Dienst muß eine Verbindung explizit zwischen dem Socket der sendenden und dem Socket der empfangenden Maschine aufgebaut werden.

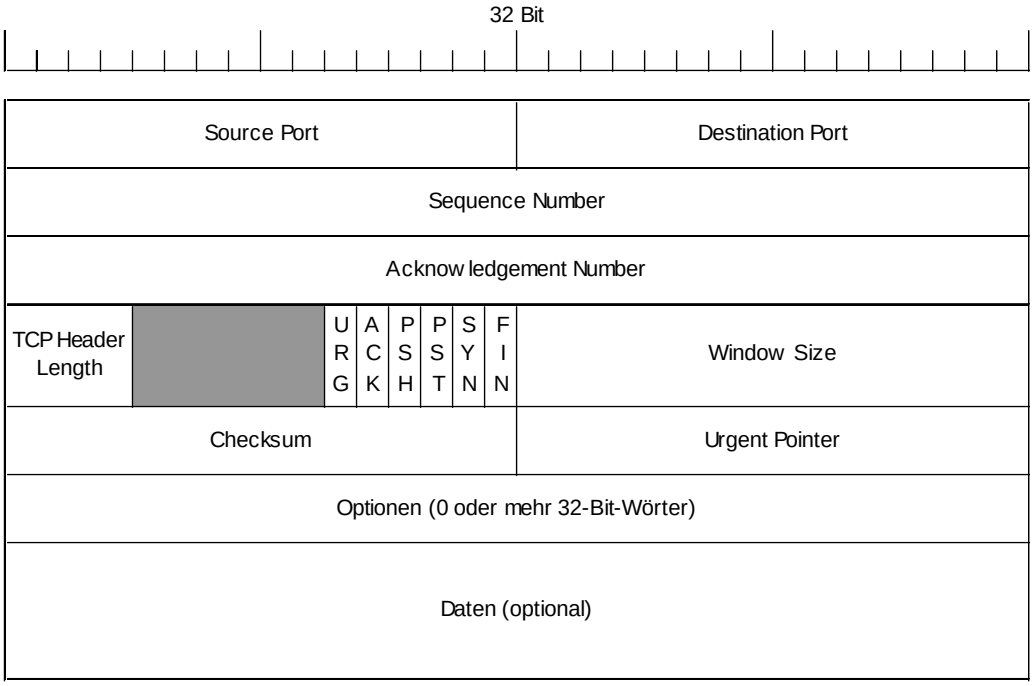
Portnummern unter 256 nennt man „well-known Ports“. Sie sind für Standarddienste reserviert. Jeder Prozeß, der z.B. mit einem Host eine Verbindung aufbaut, um über FTP eine Datei zu übertragen, kann sich an Port 21 des Zielhosts anklinken, um dessen FTP-Deamon anzusprechen. Eine Liste dieser Ports ist in RFC 1700 enthalten.

TCP-Verbindungen sind immer Vollduplex und Punkt-zu-Punkt.

Die sendende und die empfangende TCP-Einheit tauschen Daten in Form von Segmenten aus. Ein *Segment* besteht aus einem festen 20-Byte-Header (sowie einem optionalen Teil), gefolgt von Datenbytes. Die TCP-Software entscheidet, wie groß die Segmente sein sollen. Sie kann Daten von mehreren Schreiboperationen zu einem Segment zusammenfassen oder einen Schreibvorgang auf mehrere Segmente aufteilen. Die Maximale Segmentgröße hat zwei mögliche Grenzen. Einerseits muß jedes TCP-Segment in den maximal 65.535 Byte großen Nutzdatenbereich von IP hineinpassen, andererseits hat jedes Netz eine *maximale Transfereinheit (MTU - Maximum Transfer Unit)*. Diese gibt an, wie groß ein Segment sein darf, damit es durch das betroffene Netz geleitet werden kann. In der Praxis ist die MTU im allgemeinen einige tausend Byte groß und gibt die obere Grenze der Segmentgröße vor.

TCP-Einheiten benutzen ein Schiebefensterprotokoll. Überträgt ein Sender ein Segment, startet er gleichzeitig einen Timer. Kommt das Segment am Ziel an, sendet die empfangende TCP-Einheit ein Segment (falls vorhanden, mit Daten) mit eine Bestätigungsnummer. Läuft der Timer ab, wird das Segment erneut gesendet. Der Haken dieses Protokolls liegt in der Fragmentierung: Mehrere Segmente eines Datenstroms können in beliebiger Reihenfolge ankommen. TCP muß mit Problemen die hierraus resultieren fertig werden und sie effizient lösen können. Dadurch ist ein beträchtlicher Aufwand für die Optimierung der Leistung von TCP-Strömen erforderlich.

Der TCP-Segment-Header



Die Felder *Source Port (Quellport)* und *Destination Port (Zielport)* identifizieren die lokalen Endpunkte der Verbindung. Jeder Host kann selbst festlegen, wie er seine eigenen Ports ab 256 zuweist.

Die Felder *Sequence Number (Folgenummer)* und *Acknowledgement Number (Bestägigungsnummer)* führen die üblichen Funktionen durch. Das zweite Feld gibt das als nächstes zu erwartende Byte an, nicht das letzte korrekt empfange Byte. Beide Felder sind 32 Bit lang, weil in einem TCP-Strom jedes Datenbyte numeriert ist.

TCP Header Length gibt an, wie viele 32-Bit-Wörte im TCP Header enthalten sind. Diese Information ist erforderlich, weil das *Options*-Feld eine variable Länge hat.

Die nächsten 6 Bits werden nicht benutzt. Die Tatsache, daß es über ein Jahrzehnt überlebt hat, belegt, wie gut TCP ausgelegt wurde.

Das *Urgent-Flag (URG)* wird gesetzt, wenn der *Urgend Pointer (Dringendzeiger)* benutzt wird. *Urgent Pointer* verweist auf einen Byteoffset von der aktuellen Folgenummer, an der dringende Daten vorgefunden werden. Diese Funktion ersetzt Interrupt-Nachrichten.

Das *Acknowledgement Flag (ACK)* wird auf 1 gesetzt, wenn *Acknowledgement Number* einen gültigen Wert enthält.

Das *Push Flag (PSH)* fordert den Empfänger auf, die Daten der Anwendung bei Ankunft bereitzustellen und sie nicht erst zwischenzuspeichern, bis ein voller Puffer eingegangen ist (was aus Gründen der Effizienz wahrscheinlich der Fall währe).

Mit dem *Reset Flag (RST)* läßt sich eine Verbindung zurücksetzen.

Mit Hilfe des *SYN - Flags* werden Verbindungen aufgebaut.

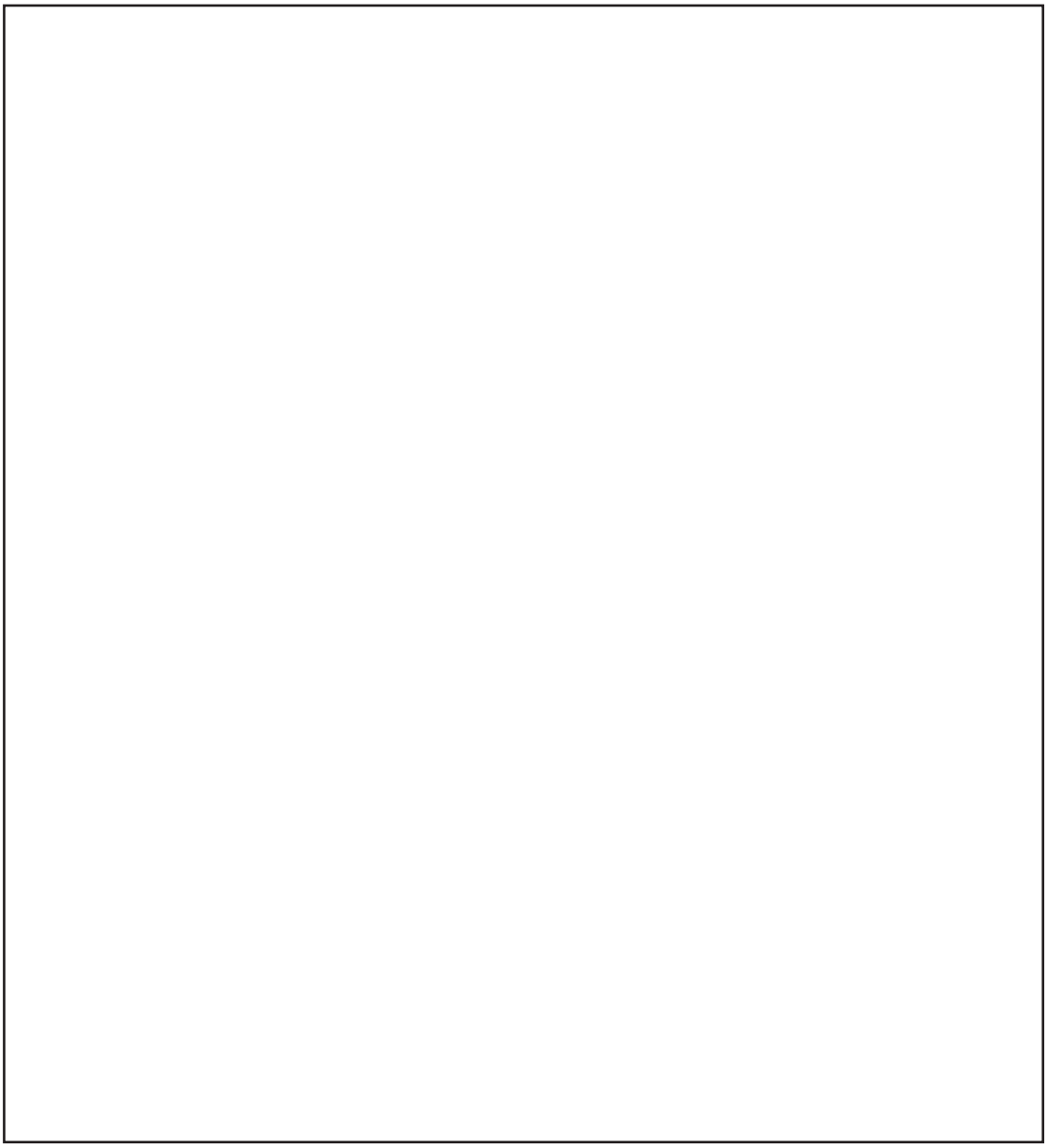
Das *Finish Flag (FIN)* baut eine Verbindung ab, wobei sie in jeder Richtung unabhängig voneinander abgebaut wird.

Die Flußrechnung erfolgt in TCP anhand eines Schiebefensters mit variabler Größe. *Window Size (Fenstergröße)* bezeichnet, wie viele Bytes ab dem bestätigten Byte gesendet werden können. Ein *Window Size* von 0 ist zulässig und besagt, daß die Bytes bis einschließlich zur *Acknowledgement Number* - 1 empfangen wurden, daß der Empfänger aber momentan unbedingt eine Verschnaufpause braucht, und vorläufig keine Daten mehr will. Die Erlaubnis zum Senden weiterer Daten kann später wieder gewährt werden, indem ein Segment mit der gleichen Bestätigungsnummer und einer *Window Size* größer Null gesendet wird.

Die *Checksum* wird aus Daten, dem TCP Header sowie eineigen Feldern aus dem IP Header gebildet. Errechnet wird sie wie im IP Header als Einerkomplement der Summe der 16-Bit-Worte.

Protokollszenarien

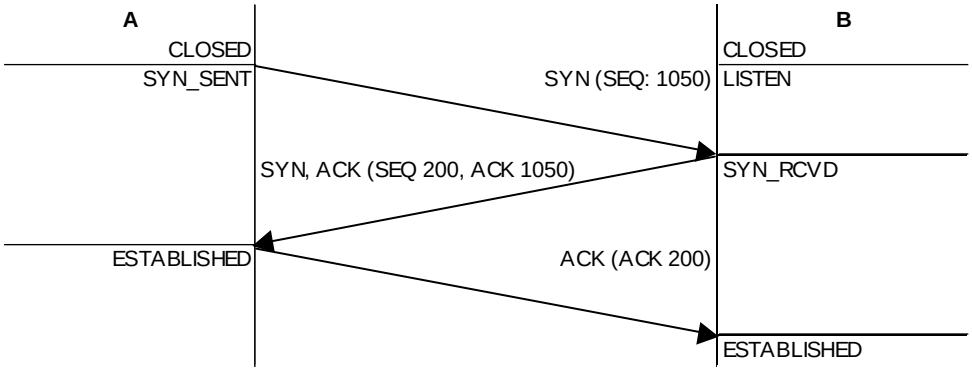
Das Socket-Interface besteht aus neun wesentlichen Aufrufen:
socket: Einrichten eines Kommunikationsendpunktes (socket)
bind: Verbinden der eigenen Adresse und Portnummer mit einem socket
listen: Es wird darauf gewartet, daß ein Client den Verbindungsaufbau initiiert
accept: Akzeptieren einer Verbindung
connect: Aktiver Aufbau einer Verbindung
send: Senden von Daten
receive: Empfangen von Daten
shutdown: Kompletter Verbindungsabbau
close: Schließen der Verbindung
Stellt man diese Vorgänge in einem Zustandsdiagramm dar, erhält man:



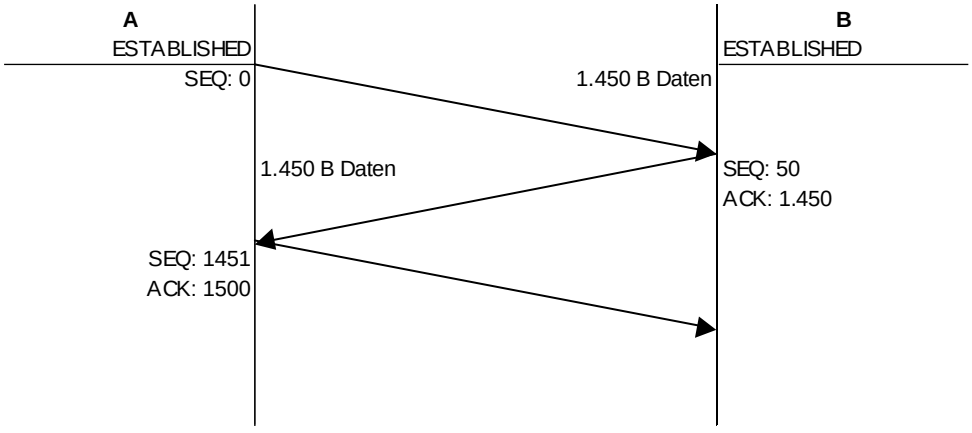
Folgende Zustände sind während einer TCP-Verbindung relevant:

Zustand	Beschreibung
CLOSED	Keine Verbindung aktiv oder anstehend
LISTEN	Der Server wartet auf eine ankommende Verbindung
SYN_RCVD	Ankunft einer Verbindungsanfrage und Warten auf Bestätigung
SYN_SENT	Die Anwendung hat begonnen, eine Verbindung zu öffnen
ESTABLISHED	Zustand der normalen Datenübertragung
FIN_WAIT_1	Die Anwendung möchte die Übertragung beenden
FIN_WAIT_2	Die andere Seite ist einverstanden, die Verbindung abzubauen
TIMED_WAIT	Warten, bis keine Pakete mehr kommen
CLOSING	Beide Seiten haben versucht, gleichzeitig zu beenden
CLOSE_WAIT	Die Gegenseite hat den Abbau eingeleitet
LAST_ACK	Warten, bis keine Pakete mehr kommen

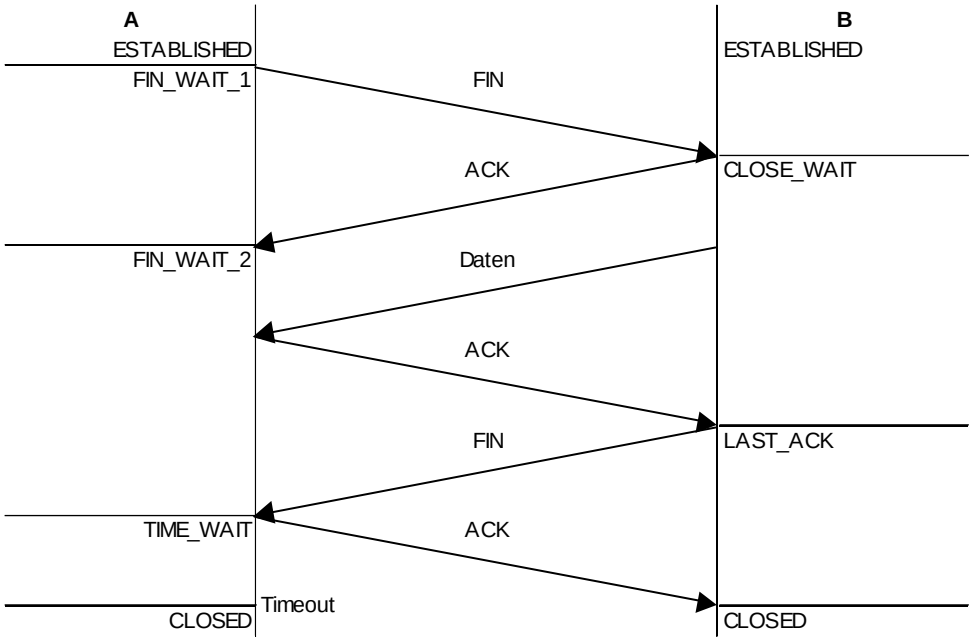
Verbindungsaufbau



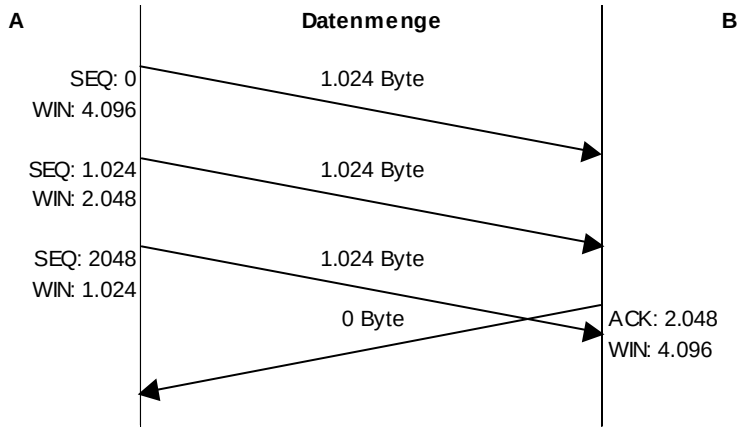
Datenaustausch



Verbindungsabbau



Window-Size



TCP-Timer-Management

TCP benutzt mehrere Timer (zumindest konzeptionell). Der wichtigste ist der *Retransmission Timer*. Wird ein Segment gesendet, startet ein Retransmission-Timer. Wird das Segment bestätigt, bevor der Timer abläuft, wird der Timer gestoppt. Läuft andererseits der Timer vor Ankunft der Bestätigung ab, wird das Segment erneut übertragen (und der Timer startet von vorn). Hier lautet die wichtigste Frage, wie lang das Timer-Intervall sein sollte.

Dieses Problem ist auf der Internet-Transportschicht viel schwieriger als bei den allgemeinen Protokollen der Sicherungsschicht, die in Kapitel 3 beschrieben wurden. Bei diesen Protokollen ist die erwartete Verzögerung gut vorhersehbar. TCP muß sich mit einer anderen Umgebung befassen. Ist der Timeout zu kurz, erfolgen unnötige Neuübertragungen. Ist er zu lang, entstehen Leistungseinbußen durch lange Verzögerungen für Neuübertragungen. Die mögliche Lösung liegt in einem sehr dynamischen Algorithmus, der das Timeout-Intervall auf der Grundlage laufender Messungen der Netzleistung ständig berichtigt. Der mit TCP allgemein angewandte Algorithmus ist auf Jacobson (1988) zurückzuführen und funktioniert wie folgt:

Für jede Verbindung verwaltet TCP die Variable *RTT (Round-Trip-Time)*. Das ist die momentan beste Schätzung der Rundreisezeit zum fraglichen Ziel. Wird ein Segment gesendet, startet ein Timer. Er überwacht, wie lange die Bestätigung braucht, und löst bei Bedarf eine Neuübertragung aus. Kommt die Bestätigung zurück, bevor der Timer abläuft, mißt TCP, wie lange die Bestätigung dauerte, z.B. *M*. Dann wird *RTT* nach der Formel

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

aktualisiert. Dabei ist α ein Glättungsfaktor, der bestimmt, wieviel Gewicht dem alten Wert beigemessen wird. Ein typischer Wert ist $\alpha = 7/8$.

Auch bei einem guten *RTT*-Wert ist die Auswahl eines geeigneten Timeouts für Neuübertragungen keine leichte Aufgabe. Zur Anpassung an die Varianz der Übermittlungszeiten wird die geglättete Variable

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

als Abweichung gegeben, wobei α nicht unbedingt der gleiche Wert sein muß, der zum Glätten von *RTT* benutzt wurde. *D* ist zwar nicht genau das gleiche wie die Standardabweichung, aber gut genug. Die meisten TCP-Implementierungen benutzten heute diesen Algorithmus und setzen das Timeout-Intervall auf

$$\text{Timeout} = RTT + 4 \times D$$

Die Auswahl von Faktor 4 ist mehr oder weniger arbitär, hat aber zwei Vorteile. Erstens sind Multiplikationen um 4 ohne Verschiebungen möglich. Zweitens werden unnötige Timeouts und Neuübertragungen vermieden, weil weniger als 1 % aller Pakete um mehr als vier Standardabweichungen zu spät kommen.

Ein Problem mit der dynamischen Schätzung von *RTT* entsteht dann, wenn zu entscheiden ist, was mit einem abgelaufenen und erneut gesendeten Segment geschehen soll. Wenn die Bestätigung ankommt, ist nicht klar, ob sie sich auf die erste oder zweite Übertragung bezieht. Eine Fehleinschätzung kann den *RTT*-Wert ernsthaft beeinträchtigen. *RTT* sollte nicht auf der Basis eins Segments aktualisiert werden, das erneut übertragen wird. Statt dessen wird das Timeout bei jedem Ausfall solange verdoppelt, bis die Segmente beim ersten Mal durchkommen - man nennt dies den *Kern-Algorithmus*. Er wird in den meisten TCP-Implementationen angewandt.

Ein weiterer TCP-Timer ist der *Persistence-Timer*. Er ist ausgelegt, um folgendes Problem zu erkennen: Der Empfänger sendet eine Bestätigung mit einer Fenstergröße von 0 und weist damit den Sender zum Warten an. Später aktualisiert der Empfänger das Fenster, aber das Paket mit der Aktualisierung ist verloren. Nun warten beide, der Sender und der Empfänger, bis einer etwas unternimmt. Hier tritt der Persistence-Timer in Aktion, da es sonst zu einem Protokoll-Deadlock führen würde. Läuft der Persistence-Timer ab, überträgt der Sender ein Paket zum Empfänger. Die Antwort auf dieses Paket ergibt die Fenstergröße. Ist sie immer noch Null, wird der Persistence-Timer wieder gesetzt und der Zyklus beginnt von neuem. Ist er nicht Null, so können Daten übertragen werden.

Der *Keep-Alive Timer* wird jedes mal neu gestartet, wenn an Paket abgeschickt wurde. Ist die Verbindung für die Dauer eines Keep-Alive Zyklus inaktiv, wird ein leeres Segment verschickt, um zu prüfen, ob die Gegenstelle noch arbeitet. Wird für die Dauer des *Idle-Timers* kein Acknowledgement empfangen, wird die Verbindung abgebrochen.

Quiet-Time wird während des Schließens im Zustand TIME_WAIT verwendet. Es läuft über das Doppelte der maximalen Paketlebensdauer, um sicherzustellen, daß alle Pakete übertragen wurden, bevor eine Verbindung geschlossen wird.

Verbesserung der Effizienz

Da jedes Segment unabhängig von der Größe einen gewissen Overhead verursacht, gibt es einige Vorkehrungen bei TCP, mit denen die Anzahl der verschickten Segmente verringert wird.

Acknowledgement Delay

Verarbeitet eine Anwendungen die eingehenden Daten sofort, können unter verschiedenen Bedingungen in kurzer Zeit mehrere Rahmen zur Bestätigung und zur Rahmenfenster-Anpassung versendet werden. Dies tritt im Speziellen bei Remote-Shells wie Telne auf, bei denen nur wenige Daten verarbeitet werden. Um diese unnötige Netzlast zu vermeiden, wird das Acknowledgement um einige Zehntelsekunden verzögert, damit der betroffene Host die Daten verarbeiten und die Antwort bereits mit dem Acknowledgement zurücksenden kann.

Silly-Window-Syndrome

Hat beispielsweise der Sender das Empfangsfenster ausgeschöpft und entnimmt die Anwendung des Empfängers jeweils einzelne Bytes, so sollte der Empfänger nicht bei jeder Entnahme ein Segment dem Sender schicken und ihm so die neue Empfangsfenstergröße mitzuteilen. Dies würde dazu führen, daß immer einzelne Bytes übertragen wurden. Stattdessen wird die Empfangsfenstergröße erst ab einem gewissen Schwellenwert aktualisiert.

Nagle's Algorithmus

Eine Ähnliche Situation währe auf der Senderseite denkbar, wenn der Sender sehr viele, kleine Datenmengen verschickt. Hier sollte im Idealfall solange gewartet werden, bis eine gewisse Datenmenge zur Übertragung bereitsteht, die dann innerhalb eines Segments übertragen wird.

Slow Start with Congestion Avoidance

Werden Segmente über ein paketvermitteltes Netz befördert, so kann es bei Überlastung zum Aufbau von Warteschlangen sowohl bei dem Ziel-Host, als auch innerhalb des dazwischenliegenden Netzwerkes kommen. Bei TCP wird auf beide Probleme verschieden reagiert. Um Engpässe beim Zielhost zu erkennen, steht die Übertragung der Fenstergröße zur Verfügung. Um Probleme beim dazwischen liegenden Netz zu erkennen, wird die *Congestion Window Size* (Überlastungsfenster) definiert. Das Tatsächliche Limit der Segmentgröße ist das Limit aus beiden Werten.

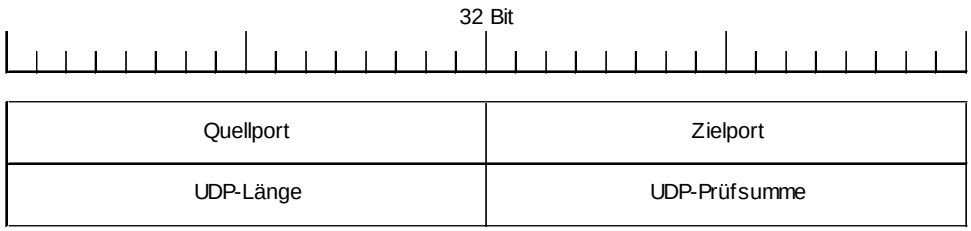
Wird eine Verbindung aufgebaut, so ist die Congestion Window Size auf die Größe eines Segmentes gesetzt. Kommen die Acknowledgements der versendeten Segmente ohne Timeout an, wird das Congestion Window verdoppelt. Dies geschieht solange, bis ein Schwellenwert (Treshold) erreicht ist. Dieser liegt zuerst bei 64 kB. Oberhalb dieser Grenze wird das Congestion Window pro erfolgreichem Sende- und Empfangszyklus jeweils um eine Segmentgröße erweitert. Tritt ein Timeout ein, so wird der Schwellenwert auf die Hälfte des aktuellen Überlastungsfensters verringert und das Fenster selbst auf die Größe eines Segmentes zurückgesetzt.

Basis dieses Algorithmus ist, daß Retransmission Timeouts aufgrund einer Netzüberlastung auftreten. Dies gilt für heutige Festverbindungen, die eine hohe Fehlersicherheit haben. Liegen fehleranfällige Leitungen, wie zum Beispiel Funkstrecken, vor, würde dieser Algorithmus die effektive Bandbreite stark verringern, weshalb er für solche Netzwerke deaktiviert werden sollte.

UDP - User Datagram Protocol

Die Internet-Protokollfolge unterstützt auch ein verbindungsloses Transportprotokoll, das *UDP*. UDP bietet Anwendungen die Möglichkeit, gekapselte rohe IP-Datengramme zu übertragen, ohne eine Verbindung aufzubauen. In vielen Client/Server-Anwendungen, die auf der Grundlage einer Anfrage und einer Antwort laufen, wird UDP benutzt. UDP ist in RFC 768 definiert.

UDP - Header



Ein UDP-Segment besteht aus einem 8-Byte-Header, gefolgt von den Daten. Das Feld *UDP Length* beinhaltet den 8-Byte-Header und die Daten. Die *UDP-Prüfsumme* wird analog zu TCP gebildet.

UDP bietet primär den Vorteil, das bei einfachen Request-Reply Paaren nicht jeweils Verbindungen aufgebaut werden müssen. Portnummern werden für TCP und UDP separat vergeben.

Index

Index