# **4.1.0 Was ist ein Netzwerk?**

Ein Netzwerk ist eine Verbindung mehrerer Computer zum Zweck des Datenaustauschs, für verteilte Anwendungen oder auch für die Kommunikation zwischen ihren Benutzern.

* Die Verkabelung oder allgemein die Hardwaregrundlage reicht von der Verwendung gewöhnlicher Telefonleitungen mit besonderen Verbindungsgeräten, den Modems, über speziell für die Anwendung in lokalen Netzwerken entwickelte Netzwerkkarten und Netzwerkkabel bis hin zu Hochgeschwindigkeitsnetzen, etwa über Glasfaserkabel. Auch die diversen Möglichkeiten der drahtlosen Übertragung werden immer wichtiger.
* Kommunikationsstrukturen, definiert durch sogenannte *Netzwerkprotokolle*, gibt es unzählige. Viele sind von einem bestimmten Hersteller, einer Plattform oder einem Betriebssystem abhängig, andere – wie die Internet-Protokollfamilie TCP/IP – sind offen, unabhängig und weit verbreitet.
* Was die Anwendungsgebiete angeht, reichen diese vom einfachen Dateiaustausch in Arbeitsgruppen über die gemeinsame Nutzung teurer Hard- und Software bis hin zu hochkomplexen, spezialisierten und verteilten Anwendungen.

# Paketvermittelte Datenübertragung

Ein wesentliches Merkmal der meisten Netzwerkformen ist die Übertragung von Daten mithilfe sogenannter *Datenpakete*.

Um die *Paketvermittlung* (Packet Switching) zu verstehen, sollten Sie zunächst ihr Gegenteil, die *Schaltkreisvermittlung* (Circuit Switching) der herkömmlichen Telefonleitungen, betrachten. (Hinweis: Inzwischen gilt dies nicht mehr zwingend; durch die Einführung neuer Technik laufen auch immer mehr Telefonverbindungen hinter den Kulissen paketvermittelt ab – per *Voice over IP* (VoIP) sogar bis zum Endkunden. Durch geeignete Kommunikationsprotokolle wird aber dafür gesorgt, dass die Nutzer dies nicht bemerken.) Durch das Wählen einer bestimmten Rufnummer (oder früher durch die Handvermittlung) werden bestimmte Schalter geschlossen, die für die gesamte Dauer des Telefongesprächs eine feste Punkt-zu-Punkt-Verbindung zwischen beiden Stellen herstellen. Über diese dauerhafte Leitung können Sprache oder Daten in Echtzeit und in der korrekten Reihenfolge ohne Unterbrechung übertragen werden. Nachdem die Übertragung beendet ist, wird die Verbindung wieder abgebaut, und die betroffenen Leitungen stehen für andere Verbindungen zur Verfügung.

Ganz anders sieht es bei der Paketvermittlung aus: Zu keinem Zeitpunkt der Datenübertragung wird eine direkte Verbindung zwischen den beiden beteiligten Stellen hergestellt. Stattdessen sind beide nur indirekt über ein loses Netz von Vermittlungsstellen, *Router* genannt, miteinander verbunden. Damit auf diesem Weg Daten übertragen werden können, wird folgender Mechanismus verwendet:

* Die Daten werden in kleinere Einheiten unterteilt, die *Datenpakete*.
* Jedes einzelne Datenpaket wird mit der Absender- und der Empfängeradresse versehen.
* Der Absender übergibt jedes Datenpaket an den nächstgelegenen Router.
* Jeder beteiligte Router versucht, das Paket anhand der Empfängerangabe an den günstigsten Router weiterzuleiten, damit es letztendlich an seinem Ziel ankommt.
* Der Empfänger nimmt die Datenpakete entgegen und interpretiert sie je nach Daten- und Übertragungsart auf irgendeine zwischen den beiden Stellen vereinbarte Art und Weise.

Zur reinen Paketvermittlung gehört zunächst einmal kein Mechanismus, der die vollständige Auslieferung aller Datenpakete garantiert. Es wird standardmäßig weder der Erfolg noch das Ausbleiben einer Paketlieferung gemeldet. Im Übrigen wird auch keine verbindliche Reihenfolge festgelegt. Da jedes einzelne Paket einen beliebigen Weg durch das Netzwerk nehmen kann, kommt mitunter ein später abgesendetes Paket noch vor einem früher versandten beim Empfänger an.

Um die potenziell unsichere Datenübertragung per Paketvermittlung für bestimmte Anwendungen zuverlässiger zu machen, wird zusätzlich eine Erfolgskontrolle implementiert. Außerdem werden die Pakete oft durchnummeriert, um die korrekte Reihenfolge wiederherzustellen. Allerdings haben solche Maßnahmen nichts mit der eigentlichen Paketvermittlung zu tun und müssen in diesem Zusammenhang nicht beachtet werden. In der Regel sind die Softwarekomponenten, die sich um die Übertragung der Datenpakete kümmern, gar nicht in der Lage, diese zusätzlichen Kontrollinformationen selbst auszuwerten.

**Lokale Netze**

Einen vollkommen anderen Anstoß zur Entwicklung von Netzwerken gab das Aufkommen des sogenannten *Outsourcings* in der Computertechnik, also der Verlagerung der Rechenleistung von einem Zentralcomputer auf den einzelnen Schreibtisch.

So entstanden viele verschiedene Arten der Netzwerkhardware. Neben dem bereits genannten Ethernet mit seinen vielfältigen Varianten gab es beispielsweise auch Token Ring von IBM, ARCnet oder auch einfache serielle Direktverbindungen zwischen Computern über die sogenannten *Nullmodemkabel*.

Was die Software angeht, wurden die eigentlich nicht dafür geeigneten PC-Betriebssysteme um Netzwerkfähigkeiten erweitert. Hinzu kamen spezielle Betriebssysteme für Server, also solche Rechner, die anderen im Netzwerk verschiedene Ressourcen zur Verfügung stellen. Bekannt sind hier etwa Novell NetWare, IBM OS/2 oder später auch Windows NT Server.

# OSI-Referenzmodell

Um die Ebenen, die ein Netzwerk ausmachen, ganz genau auseinanderhalten zu können, bedient man sich sogenannter *Schichtenmodelle* (Layer Models). Das bekannteste und verbreitetste von ihnen ist das OSI-Referenzmodell der internationalen Standardisierungsorganisation ISO. OSI steht für »Open Systems Interconnect«, also etwa »Verbindung zwischen offenen Systemen«

1. Bit-Übertragungsschicht (Physical Layer)
2. Sicherungsschicht (Data Link Layer)
3. Vermittlungsschicht (Network Layer)
4. Transportschicht (Transport Layer)
5. Kommunikationssteuerungsschicht (Session Layer)
6. Darstellungsschicht (Presentation Layer)
7. Anwendungsschicht (Application Layer)

Die Bezeichnung *OSI-Referenzmodell* deutet bereits darauf hin, dass es sich nicht um einen Standard handelt, der konkrete Netzwerkprotokolle definiert. Das OSI-Modell definiert nur die Funktionen der einzelnen Schichten und ist somit ein Schema zur Definition solcher Standards

Die *Bit-Übertragungsschicht* oder auch *physikalische Schicht* beschreibt nur, wie die reine Übertragung der Daten elektrisch beziehungsweise allgemein physikalisch vonstattengeht Die *Sicherungsschicht* beschreibt alle Vorkehrungen, die dafür sorgen, dass aus den einzelnen zu übertragenden Bits, also dem reinen physikalischen Stromfluss, ein verlässlicher Datenfluss wird. Dazu gehören die beiden Teilaufgaben Media Access Control (MAC) – die Regelung des Datenverkehrs, wenn mehrere Geräte den gleichen Kanal verwenden – sowie Logical Link Control (LLC), wobei es um die Herstellung und Aufrechterhaltung von Verbindungen zwischen den Geräten geht

Die *Netzwerkschicht* oder *Vermittlungsschicht* definiert diejenigen Komponenten und Protokolle des Netzwerks, die an der indirekten Verbindung von Computern beteiligt sind. Hier ist sogenanntes *Routing* erforderlich, das Weiterleiten von Daten in andere logische oder auch physikalisch inkompatible Netzwerke. Auch auf der Netzwerkschicht werden die Daten in Pakete unterteilt, deren Namen sich je nach konkretem Protokoll unterscheiden.

Die Protokolle der *Transportschicht* lassen sich in verbindungsorientierte Protokolle wie TCP und verbindungslose Protokolle wie etwa UDP unterteilen. Auf dieser Schicht werden vielfältige Aufgaben erledigt. Ein wichtiger Aspekt sind Multiplex-Mechanismen, die die Anbindung der Datenpakete an konkrete Prozesse auf den kommunizierenden Rechnern ermöglichen, bei TCP und UDP beispielsweise über Portnummern, bei SPX über Connection-IDs.

Die *Kommunikationssteuerungsschicht* oder *Sitzungsschicht* stellt die Kommunikation zwischen kooperierenden Anwendungen oder Prozessen auf verschiedenen Rechnern sicher.

Die *Darstellungs-* oder *Präsentationsschicht* dient der Konvertierung und Übertragung von Datenformaten, Zeichensätzen, grafischen Anweisungen und Dateidiensten.

Die *Anwendungsschicht* schließlich definiert die unmittelbare Kommunikation zwischen den Benutzeroberflächen der Anwendungsprogramme, kümmert sich also um die Verwendung derjenigen Dienste über das Netzwerk, die Benutzer unmittelbar zu Gesicht bekommen.

### Das Schichtenmodell der Internetprotokolle

Im Bereich der TCP/IP-Netzwerkprotokolle, die unter dem Betriebssystem Unix und im Internet den Standard darstellen, wird zum Beispiel häufig ein Modell aus nur vier Schichten verwendet.

1. Netzzugangsschicht (Network Access Layer oder Link Layer)
2. Internetschicht (Internet Layer)
3. Host-zu-Host-Transportschicht (Host-to-Host Transport Layer oder einfach Transport Layer)
4. Anwendungsschicht (Application Layer)

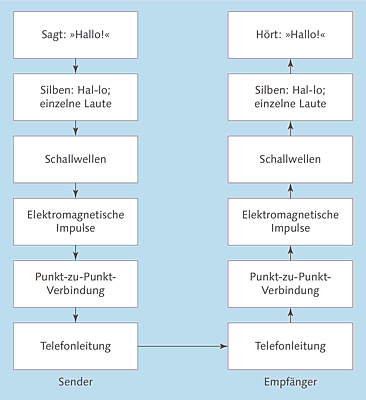
|  |  |
| --- | --- |
| **OSI-Modell** | **DDN-Modell** |
| 7. Anwendungsschicht | 4. Anwendungsschicht |
| 6. Darstellungsschicht |
| 5. Sitzungsschicht |
| 4. Transportschicht | 3. Host-zu-Host-Transportschicht |
| 3. Vermittlungsschicht | 2. Internetschicht |
| 2. Sicherungsschicht | 1. Netzzugangsschicht |
| 1. Bit-Übertragungsschicht |

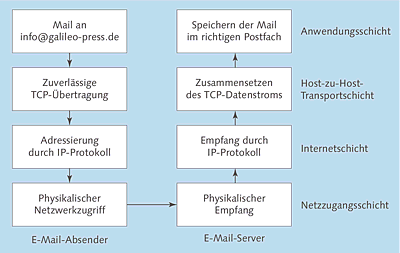
Die Netzzugangsschicht (*Network Access Layer* oder *Link Layer*) beschreibt, wie die physikalische Datenübertragung erfolgt. Die Aufgaben, die auf dieser Schicht zu erledigen sind, werden durch viele recht unterschiedliche Protokolle erbracht, einfach deshalb, weil es kaum eine Sorte von Netzwerkhardware gibt, auf der die Internetprotokolle noch nicht implementiert worden wären.

Die Internetschicht (*Internet Layer*), die im Wesentlichen der Vermittlungsschicht des OSI-Modells ähnelt, kümmert sich um die logische Adressierung der Rechner im Netz, durch die die grundsätzliche Identifizierbarkeit des jeweiligen Rechners sichergestellt wird. Eine weitere wichtige Aufgabe auf dieser Ebene ist das Routing, also die Weiterleitung von Daten über verschiedene physikalisch und/oder logisch getrennte Netze hinweg. Grundlage dieser Tätigkeiten ist das IP-Protokoll (Internet Protocol). Außerdem versieht es jedes Datenpaket mit einem Header, also einer Zusatzinformation, die insbesondere die Quelladresse des sendenden Rechners und die Zieladresse des empfangenden Hosts enthält. Ein Datenpaket dieser Ebene wird als *Datagramm* bezeichnet.

Die Host-zu-Host-Transportschicht (*Host-to-Host Transport Layer*) kümmert sich um den zuverlässigen Datenaustausch zwischen den kommunizierenden Rechnern. Im Wesentlichen sind hier zwei verschiedene Protokolle verantwortlich (neben anderen, selten verwendeten). Diese beiden Protokolle werden jedoch niemals gleichzeitig, sondern immer alternativ verwendet. Das einfachere und weniger robuste *UDP* (User Datagram Protocol) stellt einen schlichten und wenig datenintensiven Mechanismus zur Verfügung, der die direkte Nutzung der IP-Datagramme für die Host-zu-Host-Kommunikation erlaubt. Hierbei wird keine virtuelle Verbindung zwischen den beiden Rechnern hergestellt; es findet also keine Kontrolle über einen kontinuierlichen Datenstrom statt. Das erheblich komplexere *TCP* (Transmission Control Protocol) hat zwar einen deutlich größeren Overhead (Daten-Mehraufwand durch Verwaltungsinformationen) als UDP, stellt aber dafür einen zuverlässigen Transportdienst dar: Es wird eine virtuelle Verbindung zwischen den beiden Hosts hergestellt. Sie besteht darin, dass die Datenpakete durchnummeriert werden und eine Übertragungskontrolle und eventuelle Neu-Übertragung jedes einzelnen Pakets stattfinden. Ob eine Anwendung nun UDP oder TCP verwendet, ist ihre eigene Entscheidung. Allgemein benutzen Dienste, die kontinuierlich größere Datenmengen transportieren müssen, eher TCP, während etwa Verwaltungs- und Konfigurationsdienste zu UDP tendieren. Der Begriff *Host* (Gastgeber) bezeichnet übrigens jeden Computer, der an ein Netzwerk angeschlossen ist und mit anderen Geräten kommuniziert. Es ist keine Bezeichnung für einen expliziten Dienstleistungsrechner, dieser (oder vielmehr die darauf ausgeführte Software) wird *Server* genannt.

Die Anwendungsschicht (*Application Layer*) schließlich definiert die Kommunikation zwischen den Anwendungsprogrammen auf den einzelnen Rechnern; hier arbeiten Protokolle wie HTTP für Webserver, FTP zur Dateiübertragung oder SMTP für den E-Mail-Versand.

[](https://openbook.rheinwerk-verlag.de/it_handbuch/bilder/04_001.png)

[](https://openbook.rheinwerk-verlag.de/it_handbuch/bilder/04_002.png)

### Die Reichweite des Netzwerks[Zur nächsten Überschrift](https://openbook.rheinwerk-verlag.de/it_handbuch/04_003.html#dodtp8236431e-3595-4ddf-9f22-c21aa6991edb)[Zur vorigen Überschrift](https://openbook.rheinwerk-verlag.de/it_handbuch/04_003.html#dodtp027a6146-15d2-4a51-864b-5791d52437a1)

Bei der Unterteilung der Netzwerke entsprechend der Reichweite – also nach der geografischen Größenordnung, die das Netzwerk überbrückt – werden insgesamt vier Stufen unterschieden:

* Das *Local Area Network* (LAN) – lokales Netzwerk – beschreibt ein Netzwerk, das an ein einzelnes zusammenhängendes Areal gebunden ist, also etwa einen Raum, ein Gebäude oder maximal ein zusammenhängendes (Firmen-)Gelände. LANs sind heutzutage weit verbreitet in Wirtschaftsunternehmen, Schulen und Universitäten oder anderen Organisationen und Instituten.
* Das *Metropolitan Area Network* (MAN) – Stadtgebiets-Netzwerk – bezeichnet ein Netz, das eine Stadt, Gemeinde oder auch eine Region umfasst. Ein Beispiel wären die verschiedenen eigenen Netze von NetCologne in Köln. Die Ausdehnung für ein MAN liegt bei 100 km und mehr.
* Das *Wide Area Network* (WAN) – Fernnetzwerk – ist ein Netz, das mehrere Städte, eine ganze Region oder sogar ein ganzes Land umfasst. In Deutschland gibt es beispielsweise das Deutsche Forschungsnetz (DFN).
* Das *Global Area Network* (GAN) – weltweites Netzwerk – ist über mehrere Länder, einen ganzen Kontinent oder sogar die ganze Welt verbreitet. Das bei Weitem größte GAN ist heutzutage natürlich das Internet – im engeren Sinne ist ein GAN allerdings ein homogenes Netzwerk, während das Internet aus zahllosen Einzelnetzen mit unterschiedlichen Architekturen zusammengesetzt ist.

Es sei noch angemerkt, dass die drei Netzwerkarten, die größere Entfernungen überbrücken – also MAN, WAN und GAN – oftmals einfach unter dem Sammelnamen WAN zusammengefasst werden. Dies umso mehr, als alle drei Typen von Fernnetzen im Wesentlichen die gleiche Art von Technologie verwenden – oder genauer gesagt: Alle Arten von Technologien für Fernnetze werden von allen drei Netzarten benutzt.

Es gibt Fernnetze, die Wählleitungen, also einfache Telefonverbindungen, verwenden, sowohl das klassische Analog- als auch das digitale ISDN-Netz. Immer beliebter werden auch die neuen diversen DSL-Dienste, bei denen durch die Verwendung besonders hochfrequenter Signale über die normalen Kupferdrähte der Telefonleitungen wesentlich höhere Datenübertragungsraten erzielt werden. Zu einer besonderen Form der Wählleitung zählen Verbindungen über die digitalen GSM-Mobilfunknetze und deren Nachfolger GPRS und UMTS. Daneben existieren unterschiedliche Arten von Standleitungen, die für besonders häufig beanspruchte oder besonders zuverlässige Leitungen verwendet werden. Hierbei gibt es unter anderem spezielle DSL-Standleitungen oder Glasfasernetze. Auch drahtlose Übertragung, etwa über Funk- oder Satellitenverbindungen, spielt eine immer größere Rolle.

Zu beachten ist allerdings, dass DSL, Wireless LAN und Mobilfunknetze im Wesentlichen die Technologien für den Zugang einzelner Hosts zu einem MAN oder WAN darstellen. Im Backbone-Bereich, also in der eigentlichen Netzwerkinfrastruktur, kommen vor allem Zeitmultiplexing-Verfahren über Glasfasernetze zum Einsatz. Sprache, Video und sonstige Daten werden dabei über Gigabit-Ethernet, SDH/SONET, ATM oder manchmal auch Frame-Relay übertragen. Eine zunehmende Bedeutung erlangten in den letzten Jahren auch DWDM-Verfahren (Dense Wavelength Division Multiplexing). Dabei werden über ein und denselben Lichtwellenleiter mehrere Signale mit unterschiedlicher Wellenlänge gleichzeitig versandt, was für extrem hohe Datenraten sorgt.

### Die Netzwerktopologie[Zur nächsten Überschrift](https://openbook.rheinwerk-verlag.de/it_handbuch/04_003.html#dodtp0d7f13cf-03a3-4507-adb6-214792784b21)[Zur vorigen Überschrift](https://openbook.rheinwerk-verlag.de/it_handbuch/04_003.html#dodtpc9c9e325-bc30-4f10-aef4-4c459175b33d)

Die *Topologie* eines Netzwerks beschreibt, in welcher physikalischen Grundform die einzelnen Geräte organisiert sind. Manche Arten von Netzwerkhardware setzen eine bestimmte Topologie voraus, andere überlassen dem Einrichtenden die Entscheidung zwischen mehreren Möglichkeiten. Topologie ist normalerweise eine Eigenschaft lokaler Netzwerke oder gar einzelner Netzsegmente. Die meisten Fernnetze verbinden ohnehin nicht einzelne Rechner, sondern ganze Netzwerke an unterschiedlichen Orten miteinander.

Es werden im Wesentlichen folgende Grundformen unterschieden:

* Die *Bus-Topologie* beschreibt ein Netzwerk, bei dem die einzelnen Knoten (Anschlüsse) hintereinander an einem einzelnen Kabelstrang angeschlossen sind, dessen Enden nicht miteinander verbunden werden dürfen (sonst würde es sich um eine Ring-Topologie handeln!). Häufig werden die beiden Enden des Kabelstrangs durch Abschlusswiderstände (Terminatoren) abgeschlossen. Ein Beispiel für echte busförmige Netzwerke ist Ethernet über Koaxialkabel.
* Die *Stern-Topologie* ist die Form eines Netzes, bei dem alle Knoten mit jeweils eigenem Kabel an einem zentralen Gerät miteinander verbunden werden. Dieses zentrale Bindeglied heißt, je nach seiner genauen Funktionsweise, *Hub* oder *Switch*. Die Stern-Topologie wird zum Beispiel von Ethernet über Twisted-Pair-Kabel verwendet.
* Die *Ring-Topologie* ähnelt der Bus-Topologie insofern, als auch hier alle Knoten an einem zentralen Strang aufgereiht sind. Dieser zentrale Kabelstrang bildet jedoch einen geschlossenen Ring. Daraus ergibt sich automatisch eine Datenstromrichtung, in die die Datenpakete grundsätzlich weitergereicht werden. Bekanntestes Beispiel der ringförmigen Vernetzung ist Token Ring.
* Die *Baum-Topologie* schließlich ist eher ein Standard für den Zusammenschluss verschiedener Netzsegmente. Von einem zentralen Kabelstrang, gewissermaßen dem »Stamm« des Baums, gehen nach beliebigen Richtungen einzelne Verästelungen ab, an denen entweder eine einzelne Station oder ein ganzes Netz hängt.

Wichtig ist zu guter Letzt, dass ein Unterschied zwischen einer physikalischen und einer logischen Topologie bestehen kann, denn die äußere Form der Verkabelung (physikalische Topologie) kann einfach aus praktischen Erwägungen heraus gewählt worden sein, obwohl von der Funktion her eine völlig andere Struktur herrscht, nämlich die logische Topologie.

Ein gutes Beispiel für eine unterschiedliche physikalische und logische Struktur sind neuere Token-Ring-Varianten: Die eigentliche Vernetzung erfolgt sternförmig, logisch gesehen handelt es sich jedoch um einen Ring. Auch Ethernet über Twisted-Pair-Kabel verwendet physikalisch gesehen die Stern-Topologie, die logische Funktionsweise hängt von der Art des zentralen Verteilers ab: Ein Hub erzeugt letztlich die Funktion eines busförmigen Netzes, da es einen durchgehenden Strang enthält, an dem alle Stationen angeschlossen sind. Ein Switch dagegen stellt jeweils eine gesonderte Verbindung zwischen zwei Stationen her, die miteinander Daten austauschen; mithin handelt es sich hier auch logisch um die echte Sternform.

### Der Zentralisierungsgrad des Netzwerks[Zur nächsten Überschrift](https://openbook.rheinwerk-verlag.de/it_handbuch/04_003.html)[Zur vorigen Überschrift](https://openbook.rheinwerk-verlag.de/it_handbuch/04_003.html#dodtp8236431e-3595-4ddf-9f22-c21aa6991edb)

Ein weiteres wichtiges Kriterium bei der Einteilung von Netzwerken in unterschiedliche Gruppen ist die Frage nach der Arbeitsaufteilung in ihnen. Kleine Arbeitsgruppen, die jeweils mit ihren Arbeitsplatzrechnern untereinander Dateien austauschen möchten, haben hier sicherlich andere Bedürfnisse als riesige Organisationen, in denen Tausende von Anwendern auf bestimmte Datenbestände zugreifen müssen. Deshalb werden die sogenannten *Client-Server-Netzwerke*, in denen zentrale Dienstleistungsrechner, die Server, arbeiten, von den Peer-to-Peer-Netzwerken unterschieden, in denen die einzelnen Computer gleichberechtigt Ressourcen freigeben und verwenden können.

* Das *Client-Server-Netzwerk* unterscheidet generell zwei Arten von beteiligten Rechnern: Der Server (Dienstleister) ist ein Computer, der den Arbeitsstationen der einzelnen Anwender an zentraler Stelle Ressourcen und Funktionen zur Verfügung stellt; der Client (Kunde) nimmt diese Dienstleistungen in Anspruch. Die Dienste, die von Servern angeboten werden, sind sehr vielfältig: Sie reichen vom einfachen Dateiserver, der Dateien im Netzwerk verteilt oder Festplattenplatz für andere freigibt, über Druckserver, Mail- und andere Kommunikationsserver bis hin zu ganz speziellen Diensten wie Datenbank- oder Anwendungsserver.
* Das *Peer-to-Peer-Netzwerk* besteht aus prinzipiell gleichberechtigten Arbeitsplatzrechnern (*peer* heißt etwa »Kollege«). Jeder Anwender ist in der Lage, Ressourcen seines eigenen Rechners an andere im Netzwerk freizugeben. Das heißt, dass alle Rechner im Netz bis zu einem gewissen Grad Serverdienste wahrnehmen.

In der Praxis sind allerdings Mischformen häufiger anzutreffen als reine Client-Server- oder absolute Peer-to-Peer-Netze. Beispielsweise könnte man sich in einem Unternehmen die folgende Situation vorstellen: Aufgaben wie die direkte Kommunikation (E-Mail), der Zugang zum Internet (über einen Proxyserver oder einfach einen Router) oder Lösungen zum Backup (Datensicherung) werden durch zentrale Server zur Verfügung gestellt; der Zugang zu Dateien innerhalb der Abteilungen oder auf Drucker der Kollegen innerhalb eines Büros wird dagegen im Peer-to-Peer-Verfahren, unter Umgehung von Servern, geregelt.

Wichtig ist außerdem, zu verstehen, dass die Begriffe *Client* und *Server* im engeren Sinne nicht unbedingt spezifische Rechner, sondern besondere Softwarekomponenten bezeichnen.

Ein *Server* ist einfach ein Programm, das meist automatisch gestartet wird und im Hintergrund darauf »lauert«, irgendeine Dienstleistung zur Verfügung zu stellen. Allgemeiner werden solche Programme zum Beispiel im Unix-Umfeld als *Daemon* bezeichnet, unter Windows NT und seinen Nachfolgern (Windows 2000, XP, Vista, Windows 7 und Windows 8) heißen sie Dienst (*Service*). Grundsätzlich kann ein solcher Serverdienst auf jedem beliebigen Rechner laufen – vorausgesetzt natürlich, er ist für die Hardwareplattform und das Betriebssystem dieses Rechners bestimmt. Der Grund für den Einsatz besonders leistungsfähiger Hardware (eben der Serverhardware) und spezialisierter Betriebssysteme liegt einfach in ihrer höheren Belastbarkeit, wenn viele Benutzer gleichzeitig diese Dienste benötigen.

Ein *Client* ist zunächst eine Software, die in der Lage ist, mit der Serversoftware zu kommunizieren; üblicherweise stellt sie dem Benutzer auch eine Schnittstelle zur Verfügung, um diese Kommunikation in Anspruch zu nehmen. So ist beispielsweise ein Webbrowser ein Client für das HTTP-Anwendungsprotokoll, er kommuniziert also mit HTTP-Servern. Interessanterweise können Webserver und Browser auch beide auf demselben Rechner laufen. Dies ist nützlich, um Webanwendungen zunächst lokal auszuprobieren.

# Arten von Servern

Im Folgenden sollen einige Serverarten genauer vorgestellt werden. Sie sollten auf jeden Fall das zuvor Gesagte im Hinterkopf behalten: Es spielt überhaupt keine Rolle für die allgemeine Funktion, ob ein Serverdienst, also die Software, die diesen Dienst zur Verfügung stellt,

* mit anderen Diensten zusammen auf dem gleichen Rechner läuft,
* allein auf einem separaten Serverrechner ausgeführt wird oder
* sogar auf mehrere Server verteilt ist, weil ansonsten die Belastung zu groß  
  wäre.

Letzteres ist insbesondere im Bereich öffentlicher WWW-Server sehr häufig zu finden, da populäre Sites wie etwa Suchmaschinen oder große Webshops sehr viel Datenverkehr zu verkraften haben. Hier werden sogenannte *Load-Balancing-Systeme* eingesetzt, die die hereinstürmenden Anfragen automatisch möglichst gerecht auf mehrere physikalische Server verteilen.

Im Wesentlichen gibt es die folgenden wichtigen Arten von Serverdiensten:

* Fileserver
* Printserver
* Mailserver
* Webserver
* Verzeichnisdienst-Server
* Anwendungsserver und Serveranwendungen

In den folgenden Abschnitten wird jeder dieser Servertypen kurz vorgestellt; in späteren Kapiteln lernen Sie auch konkrete Beispiele für viele von ihnen kennen.

#### Fileserver

Der Fileserver (Dateiserver) stellt anderen Rechnern im Netzwerk freigegebene Verzeichnisse zur Verfügung. Auf diese Weise können sich die Anwender über einen zentralen Austauschpunkt gegenseitig Dateien zukommen lassen. Der Fileserver ist relativ stark an ein bestimmtes Betriebssystem oder eine Plattform gebunden. Erst allmählich setzen sich neuere Möglichkeiten durch, die in der Lage sind, auch unterschiedliche Rechner gleichzeitig zu bedienen. Denn die Besonderheit eines Fileservers ist, dass die Benutzer ihn völlig transparent genau so benutzen können wie die lokalen Dateisysteme ihres Arbeitsplatzrechners. In einem idealen (lokalen) Netzwerk sollte es dem normalen Anwender vollkommen egal sein, ob seine Dateien am Arbeitsplatz oder auf einem Fileserver zu finden sind.

Sehr wichtig ist im Zusammenhang mit Fileservern die Verwaltung von Zugriffsrechten, da nicht jede Datei für alle Benutzer gedacht ist.

Der Internetdienst FTP (File Transfer Protocol) ist übrigens kein vollwertiger Fileserver, sondern dient lediglich der einfachen Dateiübertragung. Die Informationen über die Dateien des entfernten Rechners sind nicht vollständig genug, um das Äquivalent eines Dateisystems abzubilden.

#### Printserver

Der Printserver (oder Druckserver) erlaubt mehreren Anwendern beziehungsweise Arbeitsstationen den gemeinsamen Zugriff auf einen Drucker. Die größte Herausforderung besteht darin, den einzelnen Arbeitsstationen automatisch den passenden Druckertreiber für ihr jeweiliges Betriebssystem zur Verfügung zu stellen, sodass diese den Drucker einfach verwenden können, ohne dass der Treiber zuvor noch einmal lokal installiert werden müsste.

Der Betrieb von Printservern ist besonders in Windows-Netzwerken weit verbreitet, da hier der Drucker gewöhnlich über ein USB-Kabel an einen einzelnen Rechner angeschlossen wird. Dieser Rechner wird dann so eingerichtet, dass er den Zugriff auf den Drucker auch den anderen Computern erlaubt.

Bei anderen Plattformen gibt es das Problem in dieser Form seltener. In klassischen Macintosh-Netzwerken ist es beispielsweise üblich – und viel bedienungsfreundlicher –, den Drucker unmittelbar per Ethernet ans Netzwerk anzuschließen, denn damit ist er automatisch für alle freigegeben.

In heterogenen Netzen war es bis vor wenigen Jahren verhältnismäßig schwierig, über Betriebssystemgrenzen hinweg gemeinsam auf einen Drucker zuzugreifen. Inzwischen ist jedoch beispielsweise das Drucksystem CUPS für alle Unix-Varianten verfügbar, das sogar Windows-Clients relativ problemlos bedienen kann.

#### Mailserver

Ein Server für elektronische Post (E-Mail) muss nicht immer bei einem Internetprovider installiert sein, sondern kann auch im lokalen Netz seinen Dienst verrichten. Denn erstens ist es in Unternehmen oder Organisationen oft von Vorteil, wenn die Mitarbeiter untereinander per E-Mail kommunizieren können, und zweitens ist es manchmal schon allein deshalb erforderlich, einen internen Mailserver zu betreiben, weil der Zugang zum Internet aus Sicherheitsgründen stark eingeschränkt ist und etwa die Kommunikation eines Arbeitsplatzrechners mit einem externen Mailserver gar nicht zulässt.

Obwohl im Lauf der Netzwerk-Entwicklungsgeschichte verschiedene Formen der elektronischen Post entstanden sind, gibt es heute eigentlich keine Alternative mehr zu Internet-E-Mail. Diese verwendet verschiedene Serverdienste zum Senden und Empfangen der E-Mail: Das SMTP-Protokoll (Simple Mail Transport Protocol) bestimmt, wie zu versendende E-Mails zu transportieren sind; POP3 (Post Office Protocol Version 3) oder das modernere, komfortablere IMAP (Internet Message Access Protocol) beschreiben ein Benutzerkonto (Postfach) für eingehende E-Mails sowie den Vorgang der »Abholung«.

Rein theoretisch kann Internet-E-Mail direkt zum einzelnen Host gesendet werden. Das ist aber insofern problematisch, als normale Arbeitsplatzrechner manchmal ausgeschaltet werden und private Einzelplatzrechner meist nur temporär über Wählleitungen mit dem Internet verbunden sind. Dies ist überhaupt der wichtigste Grund dafür, warum sich Posteingangsserver etabliert haben, auf denen die Mail für einen bestimmten Anwender im Prinzip vorgehalten wird, bis dieser sie abruft.

Da die gewöhnliche Form der E-Mail auf den Standard-Internetprotokollen aufsetzt, gibt es übrigens kein Problem, sie plattform- und betriebssystemübergreifend zu verwenden.

#### Webserver

Ein Webserver (die exakte Bezeichnung ist eigentlich *HTTP-Server*) liefert auf Anfrage Webseiten über ein Netzwerk aus. In der Regel ist dieses Netzwerk das Internet. In den lokalen Netzen von Unternehmen und Institutionen setzt sich diese Form der Informationsübermittlung aber auch immer mehr durch. Ein solches lokales Netz, das Technologien und Dienste der Internetprotokolle verwendet, wird *Intranet* genannt. Der Anwender verwendet ein Anzeigeprogramm für Webseiten, den sogenannten *Browser*, um Webseiten anzufordern, anzusehen und auch, um die enthaltenen Hyperlinks – also Verknüpfungen zu anderen Dokumenten auf dem gleichen oder einem anderen Server – per Mausklick zu folgen.

Webseiten sind prinzipiell Textdokumente, die in der Strukturierungssprache HTML geschrieben werden. Viele dieser Dokumente liegen statisch auf dem Server und werden einfach auf Anfrage ausgeliefert. Eine wachsende Anzahl solcher Dokumente wird aber auch aus Vorlagen und dynamischen Daten, etwa aus einer Datenbank, kombiniert und dann an den anfragenden Host geschickt. Diese Entwicklung ist für Websites mit umfangreichem, schnell wechselndem Inhalt, etwa Online-Tageszeitungen oder die Kataloge in E-Commerce-Sites, unvermeidlich.

Webserver sind im Übrigen schon von ihrer Grundidee her dafür gedacht, Clients unter vielen verschiedenen Betriebssystemen zu bedienen. Falls es Inkompatibilitäten geben sollte, liegt das höchstens daran, dass bei der Erstellung des HTML-Codes Steuerbefehle verwendet wurden, die nicht jeder Browser versteht.

#### Verzeichnisdienst-Server

Verzeichnisdienste (*Directory Services*) gewinnen in der IT seit längerer Zeit stark an Bedeutung. Ein Verzeichnis ist in diesem Zusammenhang kein Dateisystem, sondern ein datenbankähnlicher, standardisierter Katalog von Benutzern, Computern, Peripheriegeräten und Rechten in einem Netzwerk. Durch den Eintrag in das Verzeichnis können diese Informationen netzwerkweit abgerufen werden, sodass Verzeichnisdienste eine praktische Grundlage für zahlreiche Dienste legen, die in einer größeren Netzwerkumgebung die Arbeit der Administratoren und das Leben der Anwender erleichtern. Hier nur einige Beispiele:

* automatisierte Softwareverteilung und -installation
* mobile Benutzerprofile (Roaming User Profiles)
* zentralisierte Anmeldedienste (Single-Sign-on)
* rechner-, benutzer- und eigenschaftsbasierte Rechtekontrolle

#### Anwendungsserver und Serveranwendungen

Ein Anwendungsserver (*Application Server*) erlaubt den Benutzern die Verwendung von Anwendungsprogrammen, die sich eigentlich auf dem Server befinden, über das Netzwerk.

Bei der einfachsten Form des Anwendungsservers liegt der Datenbestand der Anwendung auf den Datenträgern des Servers, die Anwendung wird über das Netzwerk in den Arbeitsspeicher des Clients geladen und dort lokal ausgeführt. Der Unterschied zum Fileserver ist hier minimal: Es muss der Anwendung lediglich klar sein, dass eventuell notwendige Zusatzkomponenten oder Konfigurationsdaten nicht auf dem Rechner liegen, auf dem sie ausgeführt wird, sondern auf der Maschine, von der sie geladen wurde.

Bei vielen normalen Einzelplatz-Anwendungsprogrammen kann eine solche Einstellung vorgenommen werden. Diese Verwendung von Software hat vor allem zwei Vorteile: Erstens kann es weniger Arbeit bedeuten, ein Programm einmal auf dem Server statt auf mehreren Arbeitsplatzrechnern zu installieren, und zweitens können Kosten gespart werden – die meisten Softwarelizenzen gelten jeweils pro Rechner, auf dem das jeweilige Programm installiert ist. Wird eine Anwendung auf mehreren Rechnern genutzt, aber nicht gleichzeitig, kann die Software auf dem Server installiert werden; damit werden die Lizenzgebühren dann nur einmal fällig.

Bei komplexeren Formen von Anwendungsservern werden Teile des Programms – oder unter Umständen auch das ganze Programm – direkt auf dem Server ausgeführt. Die möglichen Gründe hierfür sind im Einzelfall genauso vielfältig wie die verschiedenen Formen der Verwirklichung. Beispielsweise ist es bei großen Datenbanken üblich, dass der Datenbestand als solcher auf einem Server liegt, ebenso die grundlegende Datenverwaltungssoftware. Auf den Clients existieren dann in der Regel sogenannte *Frontends*, also Softwarekomponenten, die den Benutzern eine Bedienoberfläche für die eigentliche Datenbank bereitstellen. (Den Gegenbegriff zum Frontend bildet das *Backend*, wobei es sich um einen nur für spezielle, angemeldete Benutzer zugänglichen Teil des Clients handelt, der der Verwaltung des Servers dient.)

Noch einen Schritt weiter gehen die sogenannten *verteilten Anwendungen* oder *Enterprise-Anwendungen*. Sie basieren in der Regel auf einem oder mehreren Datenbankservern für den Datenbestand, einem Anwendungsserver für die Geschäftsabläufe und diversen Client-Frontends (sowohl native Programme für bestimmte Betriebssysteme als auch Webanwendungen).

Eine andere Form der Serveranwendung existiert bei der Verwendung der sogenannten *Terminalserver*. Die einfachste Form, der Internetdienst Telnet, stellt dem Anwender eine Konsolenoberfläche zur Verfügung, über die sich von fern auf dem Server selbst mithilfe von Kommandoeingabe arbeiten lässt. Das heißt, die Ein- und Ausgabe von zeilenorientierten Kommandos und Anwendungsprogrammen erfolgt auf dem Client, die eigentliche Ausführung auf dem Server – der eigene Rechner wird somit zu einem Terminal für den entfernten Server.

Eine sehr merkwürdige Form der Serversoftware ist in diesem Zusammenhang der aus dem Unix-Bereich stammende X-Window-Server oder einfach X-Server (siehe [Kapitel 7](https://openbook.rheinwerk-verlag.de/it_handbuch/07_001.html#rxz1009500), »Linux«). Die Bezeichnung *Server* für diese Software erscheint zunächst sehr irreführend, handelt es sich doch einfach um die Grundlage der grafischen Benutzeroberfläche (GUI) unter Unix. Der X-Server stellt den Anwendungsprogrammen seine Dienste zur Verfügung, die darauf zugreifen, um Fenster und andere Komponenten der GUI darzustellen.

Die Tatsache, dass hier ein Dienst verfügbar gemacht wird, ist es übrigens, die den Begriff *Server* rechtfertigt. Dabei müssen Anwendung und X-Server auch nicht unbedingt auf dem gleichen Rechner laufen. Erstaunlicherweise läuft aber der X-Server auf dem Anwendungsclient! Denn da die Programmausführung auf dem entfernten Rechner stattfindet, aber die grafische Darstellung auf dem lokalen Rechner, muss hier dieser Dienst angeboten werden.

Terminalserver gibt es auch unter Windows Server 2008 und anderen Microsoft-Systemen; auch hier läuft die eigentliche Anwendung auf dem Server, der Client erlaubt deren Bedienung und Anzeige. Das Angebot solcher Anwendungsdienste über das Internet wird allmählich beliebter. Ein ASP (Application Service Provider) lässt Anwendungen wie beispielsweise Bürosoftware auf seinen Servern laufen; über eine spezielle Clientsoftware oder sogar über einen Webbrowser kann der Kunde darauf zugreifen und die angebotene Software von der ganzen Welt aus benutzen. Eine wesentlich einfachere Form solcher Serveranwendungen, die über das Web verwendet werden und die Sie wahrscheinlich gut kennen, ist der weitverbreitete webbasierte E-Mail-Dienst mit diversen Zusatzfunktionen, wie ihn GMX, Google Mail oder web.de anbieten.

Die Quintessenz der Verwendung von Anwendungsservern ist die von einigen Firmen (Sun Microsystems, Oracle) seit Jahren angestrebte Abschaffung der gewöhnlichen Personal Computer und deren Ersatz durch sogenannte *Thin Clients* – Rechner ohne Festplatte, die ihr Betriebssystem und die Anwendungsprogramme vollständig aus dem Netzwerk oder aus dem Internet beziehen. Allerdings konnte sich das Konzept bisher nicht recht durchsetzen. Das Hauptargument der entsprechenden Unternehmen, nämlich die geringeren Kosten, lässt sich angesichts des massiven Preisverfalls bei den »ausgewachsenen« PCs nicht aufrechterhalten.

Inzwischen sind es auf der Clientseite eher die Tablets, die den gewöhnlichen PCs den Rang ablaufen. Zudem bieten immer mehr Hardware- und Betriebssystemhersteller bereits ab Werk Cloud-Dienste zur Daten- und Anwendungsspeicherung.

## 4.4 Netzwerkkarten, Netzwerkkabel und Netzzugangsverfahren[Zur nächsten Überschrift](https://openbook.rheinwerk-verlag.de/it_handbuch/04_004.html#dodtpda21b702-f590-4d9e-abf8-73a14145d806)

Im Laufe der Entwicklungsgeschichte der Netzwerke, die in diesem Kapitel bereits skizziert wurde, haben sich viele verschiedene Formen der Netzwerkhardware entwickelt. Jede von ihnen hatte zum Zeitpunkt ihrer Entstehung ihre Berechtigung, und dennoch haben sich einige auf breiter Front durchgesetzt, während andere schnell wieder vom Markt verschwunden sind. Die verbreitetste Art der Netzwerkhardware ist heute Ethernet in seinen vielfältigen Varianten.

Analog zu den zuvor beschriebenen Schichtenmodellen – vor allem dem standardisierten OSI-Referenzmodell – gibt es auch Standards, die speziell die Netzwerkhardware und den Netzzugang betreffen, also die beiden untersten Ebenen des OSI-Modells. Die umfangreichste Sammlung ist IEEE 802 des Institute of Electrical and Electronical Engineers. Die Nummer 802 bezeichnet Jahr und Monat der ursprünglichen Festlegung, nämlich den Februar 1980. Innerhalb dieser Sammlung existiert eine Reihe verschiedener Unterstandards beziehungsweise Arbeitsgruppen. Zu den wichtigsten gehören 802.1 (allgemeine Netzwerkstandards), 802.3 (Netzzugangsverfahren CSMA/CD, besonders Ethernet) und 802.11 (drahtlose Netze). [Tabelle 4.2](https://openbook.rheinwerk-verlag.de/it_handbuch/04_004.html#rxz1004048) zeigt die vollständige Liste. Einige dieser Standards werden im Folgenden näher beschrieben.

# Tabelle IEEE-Gruppe

|  |  |
| --- | --- |
| **Tabelle 4.2**Die IEEE-802-Arbeitsgruppen im Überblick | |
| **IEEE-Gruppe** | **Bezeichnung** |
| 802.1 | Internetworking |
| 802.2 | Logical Link Control (LLC) |
| 802.3 | CSMA/CD, Ethernet |
| 802.3u | Fast Ethernet |
| 802.3z | Gigabit Ethernet über Glasfaser |
| 802.3ab | Gigabit Ethernet über Twisted Pair |
| 802.4 | Token-Bus-Zugriffsverfahren |
| 802.5 | Token-Ring-Zugriffsverfahren |
| 802.6 | Metropolitan Area Network (MAN) |
| 802.7 | Breitbandübertragungstechnologie |
| 802.8 | Glasfaserübertragungstechnologie |
| 802.9 | integrierte Sprach- und Datendienste |
| 802.10 | Netzwerksicherheit |
| 802.11 | drahtlose Netze |
| 802.12 | Demand-Priority-Verfahren |
| 802.14 | Breitband-Kabelfernsehen (CATV) |
| 802.15 | Wireless Personal Area Network (WPAN) |
| 802.16 | Broadband Wireless Access (BWA) |
| 802.17 | Resilient Packet Ring (RPR) |
| 802.18 | Radio Regulatory Technical Advisory Group (RRTAG) |
| 802.19 | Coexistence TAG |
| 802.20 | drahtlose Breitbandnetze |
| 802.21 | medienunabhängiges Handover |
| 802.22 | drahtlose Regionalnetze (WRAN) |
| 802.23 | Emergency Services Working Group |
| 802.24 | Smart Grid TAG |
| 802.25 | Omni-Range Area Network (Gruppe im Aufbau) |
| 802.30 | 100BaseX, 100BaseT, Fast Ethernet |

[Rheinwerk Computing - Zum Seitenanfang](https://openbook.rheinwerk-verlag.de/it_handbuch/04_004.html#top)

### 4.4.1 Die verschiedenen Ethernet-Standards[Zur nächsten Überschrift](https://openbook.rheinwerk-verlag.de/it_handbuch/04_004.html#dodtp90a11140-3741-47fc-ac30-1f295d1432ee)[Zur vorigen Überschrift](https://openbook.rheinwerk-verlag.de/it_handbuch/04_004.html#dodtp394801fa-91d7-4c41-b617-3f1acc499df9)

Ethernet ist heute der verbreitetste Standard für lokale Netze (LANs). Zehntausende von Hersteller weltweit unterstützen diese Art von Netzwerken mit ihrer Hard- und Software.

Jede Ethernet-Schnittstelle, also die Netzwerkkarte oder der fest eingebaute Anschluss, ist mit einer weltweit einmaligen Identifikationsnummer ausgestattet, der sogenannten *MAC-Adresse* (für Media Access Control, einer der beiden Bestandteile der OSI-Netzzugangsschicht). Es handelt sich um eine 48 Bit lange Zahl, die in sechs hexadezimalen Blöcken zwischen 0 und 255 (00 bis FF hex) geschrieben wird, zum Beispiel 00-A0-C9-E8-5F-64.

Die Datenpakete – auf der Netzzugangsschicht *Frames* genannt – werden mit den MAC-Adressen der sendenden und der empfangenden Station versehen und in der Regel an alle Stationen im Segment versandt. Jede Station überprüft daraufhin, ob die Daten für sie bestimmt sind. Im Übrigen kann man Ethernet-Schnittstellen auch in den »Promiscuous Mode« schalten, in dem sie ohne Unterschied alle Daten entgegennehmen. Auf diese Weise kann der gesamte Datenverkehr in einem Netzsegment überwacht werden.

Die MAC-Adresse wird normalerweise nicht über das jeweilige Teilnetz hinaus weiterverbreitet.[Anm.: Ausnahmen: Das im weiteren Verlauf des Kapitels beschriebene IPX/SPX-Protokoll verwendet die MAC-Adresse auch für die Adressierung auf der Netzwerkschicht, und die IP-Weiterentwicklung IPv6 benutzt die MAC-Adresse als Teil der 128 Bit langen IP-Adresse.] Nach außen ergäbe ihre Verwendung auch keinen Sinn, da das nächste Teilnetz auf einer Route womöglich noch nicht einmal zum Ethernet-Standard gehört.

# Das Netzzugangsverfahren CSMA/CD

Es ist wichtig, zu verstehen, dass mit dem Namen *Ethernet* gar keine einheitliche Netzwerkhardware bezeichnet wird. Vielmehr handelt es sich um einen Sammelnamen für diverse Netzwerkstandards, die ein bestimmtes Netzzugangsverfahren verwenden. Insofern sind alle Ethernet-Varianten auf der OSI-Schicht 2 identisch, unterscheiden sich aber auf der untersten Schicht.

Als der Vorläufer von Ethernet Ende der 60er-Jahre des vorigen Jahrhunderts an der Universität von Hawaii konzipiert wurde (anfangs unter dem geografisch passenden Namen ALOHANet), handelte es sich zunächst um Datenfunk. Diesem Umstand ist übrigens auch der endgültige Name zu verdanken: *ether*, zu Deutsch Äther, ist das gedachte Medium, durch das sich Funkwellen fortpflanzen. Erst in den 70er-Jahren wurde dasselbe Netzzugangsverfahren auch für die Datenübertragung per Kabel eingesetzt, und zwar zunächst über Koaxialkabel.

Das gemeinsame Netzzugangsverfahren aller Ethernet-Formen trägt den Namen CSMA/CD: Carrier Sense Multiple Access with Collision Detection. Schematisch gesehen funktioniert dieses Verfahren wie folgt:

1. Ein Gerät, das Daten senden möchte, lauscht den Netzabschnitt ab, um festzustellen, ob dieser gerade frei ist, das heißt, ob gerade kein anderes Gerät sendet (*Carrier Sense*).
2. Wurde in Schritt 1 festgestellt, dass der Netzabschnitt frei ist, beginnt die Station mit dem Senden der Daten. Möglicherweise hat auch eine andere Station festgestellt, dass das Netz frei ist, und beginnt gleichzeitig ebenfalls mit dem Senden (*Multiple Access*).
3. Falls auf die beschriebene Art und Weise zwei Stationen gleichzeitig mit dem Senden begonnen haben, findet eine sogenannte *Datenkollision* statt, die von den beteiligten Stationen entdeckt wird (*Collision Detection*). Eine Station, die eine Kollision bemerkt, stellt das Senden von Nutzdaten ein und versendet stattdessen eine Warnmeldung (Jam Signal).
4. Eine Station, die wegen einer Datenkollision das Senden abgebrochen hat, beginnt nach einer zufällig gewählten Zeitspanne von wenigen Millisekunden erneut mit dem Senden. Genau diese Zufälligkeit der Zeitspanne, die nach einem komplizierten Verfahren berechnet wird, ist enorm wichtig, damit die beiden Stationen beim nächsten Versuch nicht wieder genau gleichzeitig mit dem Senden beginnen.

Das große Problem von Ethernet besteht darin, dass das CSMA/CD-Verfahren umso ineffektiver wird, je frequentierter der jeweilige Netzabschnitt ist: Ab einem gewissen Grenzwert überschreitet die Anzahl der Datenkollisionen die Menge der Nutzdaten. Heutzutage umgeht man dieses Problem in der Regel durch die Verwendung sogenannter *Switches*, die für zwei miteinander kommunizierende Stationen jeweils eine exklusive Punkt-zu-Punkt-Verbindung einrichten. Wo diese Möglichkeit aufgrund veralteter, inkompatibler Hardware nicht zur Verfügung steht, muss ein Netz mit viel Datenverkehr stattdessen segmentiert, das heißt in kleinere Abschnitte unterteilt werden.

#### Ethernet-Hardware

Die Bezeichnungen der verschiedenen Arten der Hardware, die für Ethernet-Netzwerke verwendet werden, setzen sich aus der Übertragungsgeschwindigkeit des jeweiligen Netzes in MBit/s und einer spezifischen Bezeichnung für den Kabeltyp oder die maximal zulässige Kabellänge zusammen.

Wie bereits erwähnt, waren Koaxialkabel die ersten für Ethernet verwendeten Kabel.[Anm.: Die Verwendung von Koaxialkabeln für Ethernet ist weitgehend historisch – aus Gründen des Leseflusses habe ich mich aber entschlossen, die Beschreibungen im Präsens zu belassen. Interessant ist die Entwicklung allemal; sie erklärt, warum bei Ethernet viele Dinge so und nicht anders gelöst wurden.] Der Aufbau dieser Kabel ist folgender: Im Zentrum befindet sich ein leitender Draht, der von einer Isolationsschicht umgeben ist, darüber befindet sich ein weiterer Ring aus leitendem Metall und außen natürlich wiederum eine Isolationsschicht. Das bekannteste Alltagsbeispiel für ein Koaxialkabel ist ein handelsübliches Fernsehantennenkabel.

Es gibt zwei Arten von Koaxialkabeln, die für Ethernet eingesetzt werden:

* 10Base2: dünnes schwarzes Koaxialkabel  
  Die 10 steht für die maximale Datenübertragungsgeschwindigkeit des Netzes, in diesem Fall 10 MBit/s. Die nähere Spezifikation, die durch die 2 angegeben wird, betrifft die maximal zulässige Gesamtlänge eines 10Base2-Netzsegments von etwa 200 Metern (eigentlich 200 Yard, was ca. 185 Metern entspricht). In einem Segment dürfen sich maximal 30 Stationen befinden. Um eine größere Entfernung zu überbrücken oder mehr Stationen zu betreiben, muss eine Signalverstärkung durch sogenannte *Repeater* durchgeführt werden.

Alternative Bezeichnungen für diese Ethernet-Form sind *Thinnet Coaxial* oder *Cheapernet*, weil es sich früher um die billigste Art der Vernetzung handelte.

An der Netzwerkkarte wird an eine BNC-Buchse ein T-Adapter angeschlossen. An dessen beiden Seiten werden wiederum über BNC-Stecker die Koaxialkabel angeschlossen, die zu den T-Stücken der Netzwerkkarten der benachbarten Rechner führen. Der Mindestabstand zwischen zwei T-Stücken, das heißt die minimale Länge eines einzelnen Kabels, beträgt 50 cm. Das Netzwerk ist in einer Bus-Topologie organisiert; die T-Stücke des ersten und des letzten Rechners im Netzwerk werden auf je einer Seite mit einem Abschlusswiderstand oder Terminator versehen.

* 10Base5: dickes gelbes Koaxialkabel  
  Der Vorteil dieser auch *Thicknet Coaxial* genannten Variante besteht in der größeren zulässigen Länge des Netzsegments, nämlich – wie die Zahl 5 vermuten lässt – 500 Yard (knapp 460 m). Andererseits ist dieses erheblich dickere Kabel weniger flexibel als das dünnere 10Base2. Beispielsweise ist es schwieriger, solche Kabel durch verwinkelte Kabelkanäle zu ziehen.

Auf dem Kabel sitzen bei dieser Ethernet-Form sogenannte *Transceiver*, die über 15-polige Buchsen an die Netzwerkkarten angeschlossen werden. Zwischen zwei Transceivern muss ein Mindestabstand von 2,5 Metern eingehalten werden; das Kabel enthält ab Werk Markierungen in diesem Abstand. Die Transceiver werden an diesen Stellen einfach in das Kabel hineingebohrt (deshalb werden sie als *Vampirabzweige* bezeichnet). In einem Segment dürfen sich maximal 100 davon befinden. Auch dieses Netz ist busförmig, und beide Enden müssen durch Abschlusswiderstände terminiert werden.

Heutzutage wird Ethernet viel häufiger über Twisted-Pair-Kabel betrieben. Bei dieser Kabelsorte handelt es sich um einen verdrillten Kupfer-Zweidrahtleiter: Je zwei isolierte Kupferdrähte werden umeinandergewickelt. Dies verhindert die gegenseitige Beeinträchtigung der Signalqualität, die bei parallel zueinander verlaufenden Kabeln durch die elektromagnetischen Felder aufträte. In einem Twisted-Pair-Kabel verlaufen üblicherweise vier, manchmal auch acht solcher Doppeladern nebeneinander. Sie enden auf beiden Seiten in einem RJ-45-Stecker, der auch für ISDN-Anschlüsse verwendet wird. Bekannt sind solche Kabel vor allem durch ihre Verwendung als Telefonleitungen.

Man unterscheidet zwei verschiedene Grundarten von Twisted-Pair-Kabeln: UTP oder Unshielded Twisted Pair ist ein nicht abgeschirmter Zweidrahtleiter, STP (Shielded Twisted Pair) ein abgeschirmter, der eine höhere Signalqualität aufweist, sodass er zum Beispiel größere Entfernungen überbrücken kann.

Außerdem werden Twisted-Pair-Kabel in verschiedene Kategorien unterteilt, die unterschiedliche maximale Datenübertragungsraten zulassen. Diese sind in [Tabelle 4.3](https://openbook.rheinwerk-verlag.de/it_handbuch/04_004.html#rxz1004167) aufgelistet.

|  |  |  |
| --- | --- | --- |
| **Tabelle 4.3**Die verschiedenen Kategorien von Twisted-Pair-Kabeln | | |
| **Kategorie** | **Übertragungsrate** | **Verwendungszweck** |
| 1 | nicht festgelegt | Telefonie |
| 2 | 4 MBit/s | ISDN |
| 3 | 10 MBit/s | Ethernet; Token Ring |
| 4 | 16 MBit/s | verschiedene |
| 5 | 100 MBit/s | Fast Ethernet; allgemeiner Standard |
| 6 | 200 MBit/s | verschiedene |
| 7 | 600 MBit/s | verschiedene |

Alle über Twisted Pair verkabelten Arten von Ethernet weisen eine sternförmige Topologie auf, zumindest im physischen Sinn: Alle Stationen werden jeweils über ein eigenständiges Kabel an einen zentralen Verteiler angeschlossen. Der Vorteil dieser Form der Vernetzung besteht grundsätzlich darin, dass der Ausfall einer einzelnen Verbindung zwischen einem Rechner und dem Verteiler nicht zur Unterbrechung des gesamten Netzes führt, wie es beim busförmigen Koaxialkabel-Ethernet der Fall ist.

Der zentrale Verteiler wird in seiner einfacheren Form *Hub* genannt, die etwas teurere, aber leistungsfähigere Bauweise heißt *Switching Hub* oder kurz *Switch*. Die innere Struktur des Hubs ist letztlich busförmig, sodass es genau wie bei der Vernetzung über Koaxialkabel zu Datenkollisionen kommen kann. Ein Switch stellt dagegen für zwei Stationen, die miteinander kommunizieren möchten, eine exklusive Punkt-zu-Punkt-Verbindung bereit. Dies geschieht dadurch, dass ein Switch die MAC-Adressen aller Schnittstellen zwischenspeichert, an die er bereits Daten ausgeliefert hat, und auf diese Weise die restlichen Stationen nicht mehr mit Daten behelligen muss, die gar nicht für sie bestimmt sind. Da die Preise für Netzwerkzubehör in den letzten Jahren stark gesunken sind, gibt es eigentlich keinen Grund mehr, etwas anderes als einen Switch einzusetzen.

Bei einem Hub teilen sich alle Stationen die gesamte Übertragungsgeschwindigkeit, beim Switch steht sie dagegen jeder einzelnen Verbindung zur Verfügung.

Im Übrigen gibt es besondere Hubs, die als *Bridges* bezeichnet werden. Sie verbinden Ethernet-Netzwerke verschiedenen Typs miteinander, beispielsweise besitzen sie eine Reihe von RJ-45-Ports für Twisted-Pair-Kabel und zusätzlich einen Anschluss für 10Base2-BNC-Kabel; oder sie unterstützen einfach verschiedene maximale Übertragungsgeschwindigkeiten.

Hubs oder Switches weisen in der Regel 5 bis 24 Anschlüsse (Ports) auf, an die jeweils ein Gerät angeschlossen werden kann. Um Netzwerke mit mehr Geräten zu betreiben, sind diese Geräte kaskadierbar: Die meisten Hubs oder Switches besitzen einen speziellen Port, den sogenannten *Uplink-Port*, der über ein Kabel mit einem normalen Port eines weiteren Verteilers verbunden werden kann. Bei vielen Hubs/Switches kann ein einzelner Port über einen Schalter zwischen »Normal« und »Uplink« umgeschaltet werden.

Die einzige Ausnahme von der allgemeinen Regel, dass ein Hub oder Switch benötigt wird, bildet der Sonderfall, in dem nur zwei Rechner miteinander vernetzt werden sollen: Die beiden Stationen können unmittelbar über ein sogenanntes *Crosslink-Kabel* verbunden werden. Dieses spezielle Kabel besitzt überkreuzte Anschlusspaare anstelle der geradlinig verlaufenden bei normalen Twisted-Pair-Kabeln.

Historisch betrachtet existieren zwei Arten von Ethernet über Twisted Pair, die unterschiedliche Übertragungsgeschwindigkeiten unterstützen:

* 10BaseT: Die Datenübertragungsrate beträgt 10 MBit/s.
* 100BaseT (auch *Fast Ethernet* genannt): Daten werden mit bis zu 100 MBit/s übertragen; hierzu sind mindestens UTP-Kabel der Kategorie 5 erforderlich. Genauer gesagt gibt es zwei Unterarten: 100BaseTX ist voll kompatibel mit 10BaseT, sodass das Netz schrittweise umgerüstet werden kann. 100BaseT4 verwendet dagegen alle vier Kupferdrahtpaare eines Twisted-Pair-Kabels und ist mit den anderen Standards inkompatibel; in der Praxis spielt es keine Rolle mehr.

Die meisten Netzwerkkarten, Hubs und Switches, die heute verkauft werden, unterstützen beide Übertragungsraten. Der zu verwendende Wert kann bei vielen Netzwerkkarten per Software eingestellt werden, häufiger wird er automatisch gewählt. Natürlich sollten Sie prinzipiell darauf achten, keine reine 10-MBit-Hardware mehr zu kaufen. Aber möglicherweise hat 100-MBit-Hardware der ersten Generation, die nicht auf 10 MBit/s heruntergeschaltet werden kann, sogar noch schlimmere Einschränkungen zur Folge. Zwar ist es bei normalen Standard-PCs ein Leichtes, die Netzwerkkarte gegen ein neueres Modell auszutauschen, um die Kompatibilität zu einer aktualisierten Netzwerkumgebung aufrechtzuerhalten, aber bei anderen Geräten wie beispielsweise Netzwerkdruckern oder kompakten Router-Boxen ist das eventuell nicht möglich. Solche Geräte sind mit einem reinen 100er-Netz eventuell nicht mehr kompatibel.

Noch neuere Formen von Ethernet erreichen Übertragungsraten von 1.000 MBit/s (Gigabit-Ethernet), entweder über Lichtwellenleiter (1000BaseFL für »Fiber Logic«) oder über mehradrige Twisted-Pair-Kabel (1000BaseTX). Bereits entwickelt, aber noch nicht weit verbreitet, sind Ethernet-Varianten mit 10 oder gar 100 GBit/s – anfangs nur über verschiedene Arten von Lichtwellenleitern, aber inzwischen ebenfalls über Twisted Pair.

### 4.4.3 Drahtlose Netze[Zur nächsten Überschrift](https://openbook.rheinwerk-verlag.de/it_handbuch/04_004.html#dodtpb750477c-7a85-4d6e-9f77-c50c6057025d)[Zur vorigen Überschrift](https://openbook.rheinwerk-verlag.de/it_handbuch/04_004.html#dodtp90a11140-3741-47fc-ac30-1f295d1432ee)

Schon seit sehr langer Zeit werden über drahtlose Technologien wie Funk, Mikrowellen, Satellit oder Infrarot nicht nur Sprache, Radio- und Fernsehsignale, sondern auch Daten übertragen. Die digitale (!) Datenübertragung per Funk war sogar die erste Anwendung der drahtlosen Nachrichtentechnik überhaupt: Der Funkpionier Guglielmo Marconi erfand die drahtlose Telegrafie mithilfe des binären Morse-Alphabets[Anm.: Ganz und gar binär ist das Morsealphabet übrigens nicht: Neben »Lang« und »Kurz« muss die Pause zwischen zwei Zeichen als drittes mögliches Signal betrachtet werden, da die einzelnen Zeichen (übrigens gemäß ihrer Häufigkeit in englischen Texten) aus unterschiedlich vielen Einzelsignalen bestehen.] lange vor dem Sprechfunk.

Im Bereich der Netzwerke gibt es immer mehr Anwendungsfälle, bei denen sich der Einsatz drahtloser Techniken anbietet. Die folgenden Beispiele können als Anhaltspunkte dienen:

* In Privathaushalten wird WLAN inzwischen häufiger eingesetzt als kabelbasierte Netze. Da viele Menschen Laptops und/oder WLAN-fähige Mobiltelefone haben, ist dies auch viel praktischer. Für den Internetzugang kommen entsprechend oft WLAN-DSL-Router zum Einsatz, die eine Verbindung zwischen dem Internet und den Endgeräten vermitteln.
* In einem Unternehmen werden viele Außendienstmitarbeiter beschäftigt. Sie sind mit Notebooks ausgestattet und kommen nur gelegentlich in die Firmenzentrale.
* Eine Firma zieht in ein denkmalgeschütztes Haus ein, an dessen Bausubstanz nichts geändert werden darf – an das Verlegen von Kabelkanälen oder gar das Aufstemmen von Wänden für die Vernetzung ist nicht zu denken.
* Zwischen zwei Gebäuden eines Unternehmens verläuft eine öffentliche Straße; für die Überbrückung durch ein Kabel müsste ein langfristiges Genehmigungsverfahren mit ungewissem Ausgang eingeleitet werden.
* Auf LAN-Partys (Treffen von Netzwerkspielern), Messen, Kongressen oder ähnlichen Veranstaltungen müssen Unmengen von Computern für kurze Zeit vernetzt werden.

Für den Betrieb drahtloser Netzwerke kommen die verschiedensten Übertragungsmethoden zum Einsatz. Sie lassen sich nach folgenden Kriterien unterscheiden oder für den praktischen Einsatz auswählen:

* Welche maximale Entfernung zwischen zwei Stationen muss überbrückt werden?
* Besteht zwischen den einzelnen Standorten Sichtkontakt, oder befinden sich Wände oder andere Hindernisse zwischen ihnen?
* Soll eine freie Funkfrequenz genutzt werden, oder kann es auch eine lizenzpflichtige sein (Letzteres kann teuer werden)?
* Sind die vernetzten Geräte selbst stationär oder mobil?

Diese diversen Auswahlkriterien zeigen bereits, dass es so etwas wie »das« drahtlose Netz nicht gibt. Für jeden Anwendungszweck bieten sich verschiedene Lösungen an, die sorgfältig geprüft werden müssen.

Genau wie bei der verkabelten Konkurrenz lassen sich auch hier verschiedene Kategorien von Reichweiten unterscheiden. Das WLAN (Wireless LAN, auch *WiFi* genannt) nach IEEE 802.11 ist ein drahtloses Netz für den Nahbereich, also für die Vernetzung innerhalb einer einzelnen Institution. Das WWAN (Wireless Wide Area Network) dagegen ist ein drahtloses Fernnetzwerk. Dazu zählen unter anderem Satellitenverbindungen.

In diesem Abschnitt wird nur das 802.11-kompatible WLAN beschrieben, da es sich seit seiner Einführung 1997 sehr schnell verbreitet hat und heute von allen Wireless-Technologien am häufigsten eingesetzt wird. 802.11 besteht aus mehreren Unterstandards, die sich in den Punkten Frequenzspektrum, Übertragungsrate und Funktechnologie unterscheiden. Sie alle werden jedoch über Funk betrieben; eine ursprünglich ebenfalls spezifizierte Infrarotvariante hat sich nicht durchgesetzt. Infrarot wird größtenteils für den drahtlosen Anschluss von Peripheriegeräten wie Mäusen oder Tastaturen verwendet. [Tabelle 4.4](https://openbook.rheinwerk-verlag.de/it_handbuch/04_004.html#rxz1004312) zeigt eine Übersicht über die wichtigsten gebräuchlichen 802.11-Varianten.

|  |  |  |  |
| --- | --- | --- | --- |
| **Tabelle 4.4**Verschiedene Varianten von IEEE 802.11 | | | |
| **Standard** | **Frequenzbereich** | **Übertragungsrate** | **Funktechnik** |
| 802.11 | 2,4 GHz | 1 oder 2 MBit/s | FHSS/DSSS |
| 802.11a | 5 GHz | bis zu 54 MBit/s | OFDM |
| 802.11b | 2,4 GHz | 5,5/11/22 MBit/s | HR/DSSS |
| 802.11g | 2,4 GHz | bis zu 54 MBit/s | OFDM |
| 802.11n6 | 2,4 und 5 GHz | bis zu 600 MBit/s | MIMO |

Die Trägerfrequenz von 2,4 GHz wird vor allem deshalb am häufigsten verwendet, weil sie nicht lizenzpflichtig ist. Es handelt sich nämlich um diejenige Frequenz, mit der Mikrowellenherde arbeiten, da diese Wellenlänge Wassermoleküle am effektivsten erhitzt.

Die diversen Funkverfahren arbeiten alle mit verschiedenen Varianten der Frequency-Hopping-Methode, die auch im Mobilfunk eingesetzt wird: Nach einem bestimmten Schema werden die Funkwellen über mehrere Frequenzen übertragen, die mehrmals in der Sekunde wechseln. Dies ist erheblich weniger störanfällig als die Verwendung einer einzelnen Frequenz. Die grundlegende Technik wurde Mitte der 30er-Jahre von der österreichischen Schauspielerin Hedy Lamarr erfunden. Ihr damaliger Ehemann war Rüstungsfabrikant, und diese Funktechnik sollte helfen, Torpedos der Alliierten fernzusteuern, ohne dass die Signale abgefangen und verfälscht werden konnten. Im Einzelnen werden folgende Verfahren unterschieden:

* FHSS (Frequency Hopping Spread Spectrum): Die Frequenzen wechseln nach einem zufälligen Muster.
* DSSS (Direct Sequence Spread Spectrum): Es werden erheblich mehr Einzelfrequenzen verwendet; die Verteilung erfolgt nach einem komplexen mathematischen Verfahren.
* HR/DSSS (High Rate Direct Sequence Spread Spectrum): Entspricht DSSS mit speziellen Erweiterungen, die eine höhere Übertragungsrate ermöglichen.
* OFDM (Orthogonal Frequency Division Multiplexing): Jeder Kanal wird in mehrere Teilkanäle unterteilt, die Signale werden über alle Teilkanäle parallel übertragen. Aus diesem Grund ist OFDM das Übertragungsverfahren mit der höchsten Datenrate, andererseits aber auch das aufwendigste, sodass die entsprechende Hardware noch vor wenigen Jahren vergleichsweise teuer war.
* MIMO (Multiple Input/Multiple Output): Im Wesentlichen eine nochmals verbesserte OFDM-Variante, die wiederum erheblich höhere Übertragungsraten ermöglicht. Die Datenübertragung kann gleichzeitig über mehrere Frequenzbänder erfolgen.

Der größte Teil der Wireless-LAN-Hardware, der momentan verkauft wird, basiert auf den Standards 802.11b und 802.11g (die meisten Geräte unterstützen wahlweise beide). Die Preise für Hardware dieser Variante sind in den letzten Jahren stark gefallen. Ein WLAN-Adapter ist inzwischen ab etwa 20 € erhältlich, sowohl als PCI-Karte als auch als PCMCIA- oder USB-Adapter. Außerdem sind Notebooks (und meist auch Desktop-PCs) ab Werk standardmäßig mit einer WLAN-Schnittstelle ausgestattet. Vorreiter dürften das PowerBook und das iBook von Apple gewesen sein; Apple fördert diese Technologie unter dem Namen AirPort seit vielen Jahren.

Als Netzzugangsverfahren in 802.11-Netzen kommt CSMA/CA zum Einsatz (Carrier Sense Multiple Access with Collision Avoidance) – wie der Name vermuten lässt, werden Datenkollisionen von vornherein vermieden. Anders als bei CSMA/CD sendet eine Station, die ein freies Übertragungsmedium (in diesem Fall den entsprechenden Funkkanal) vorfindet, nicht einfach ihre Daten, sondern eine Sendeanforderung (RTS). Daraufhin warten andere sendebereite Stationen; und die erste Station, die das RTS gesendet hat, sendet ihre Daten, nachdem ihr die Empfängerstation ihre Empfangsbereitschaft (CTS) signalisiert hat. Abgeschlossen wird die Datenübertragung durch ein ACK-Signal, daraufhin kann die nächste Station ihren Sendewunsch bekanntgeben.

Das einfachste denkbare 802.11-WLAN besteht nur aus mehreren Rechnern mit entsprechender Schnittstelle, die auf direktem Weg miteinander kommunizieren. Ein solcher Aufbau wird als *Basic Service Set* (BSS) bezeichnet. Die Entfernung zwischen zwei beliebigen Stationen darf die maximale Reichweite des Funksignals nicht überschreiten, da jede Station die Signale nur senden und empfangen, aber nicht verstärken und weiterleiten kann. Da ein solches Netzwerk nicht mit anderen Netzen kommunizieren kann, wird es als *unabhängiges BSS* (Independent BSS oder kurz IBSS) bezeichnet. Derartige Netzwerke sind sinnvoll für die sogenannte *Ad-hoc-Vernetzung* temporärer Zusammenkünfte wie Messen oder LAN-Partys.

Ein wenig komplexer wird der Aufbau eines BSS, wenn ein Access Point hinzugefügt wird. Im Grunde funktioniert ein Access Point wie ein Ethernet-Hub, denn sobald er vorhanden ist, kommunizieren die Stationen nicht mehr direkt miteinander, sondern senden die Frames an den Access Point, der sie an den gewünschten Empfänger weitergibt. Die Identifikation der einzelnen Stationen erfolgt wie bei Ethernet anhand einer 48 Bit langen MAC-Adresse. Ein BSS mit einem Access Point wird als *Infrastruktur-BSS* bezeichnet. Für die Reichweite des Netzes ist nur noch die Entfernung zwischen einer Station und dem Access Point ausschlaggebend.

Die wichtigste Aufgabe eines Access Points besteht in seiner Funktion als Bridge. Er verbindet das WLAN mit einem Backbone-Netzwerk – meistens Twisted-Pair-Ethernet. Auf diese Weise kann das WLAN mit stationären Teilen des Netzes verbunden werden oder Zugang zu Servern und Routern erhalten, ohne dass diese selbst mit WLAN-Schnittstellen ausgestattet werden müssten.

Im Übrigen bildet ein Verbund aus miteinander vernetzten Access Points (entweder ebenfalls über Funk oder über Ethernet) ein sogenanntes *Extended Service Set* (ESS). Eine Station kann sich innerhalb eines ESS frei bewegen, weil die Access Points einander darüber auf dem Laufenden halten, welche Stationen sich gerade in ihrem Bereich befinden. Eine Station kann immer nur genau mit einem Access Point verbunden sein; sobald das Signal eines anderen Access Points stärker wird als das des bisherigen, meldet die Station sich bei ihrem alten Access Point ab und bei dem neuen an. Auf diese Weise werden Frames immer über den jeweils aktuellen Access Point an eine Station gesendet.

Ein zusätzlicher Nutzen von Access Points besteht darin, dass sie in der Lage sind, Frames zu puffern, die an bestimmte Stationen adressiert sind. Gerade Notebooks schalten im Standby-Modus oft auch die WLAN-Schnittstelle ab, um Strom zu sparen; sobald die Verbindung wieder aufgebaut wird, werden die zwischengespeicherten Frames ausgeliefert.

Das ESS-Modell wird immer häufiger für öffentlich verfügbare Netzwerkzugänge eingesetzt. In Bahnhöfen, Flughäfen oder Gaststätten stehen öffentlich zunehmend WLAN-Access-Points (auch *Hotspots* genannt) zur Verfügung, in die sich Notebook-Benutzer ohne Weiteres einwählen können. Mittlerweile werden sogar die ersten Innenstädte fast flächendeckend mit einander überlappenden Access Points ausgestattet. Irgendwann könnte ein ähnlich dichtes Netz entstehen, wie es die Mobilfunkzellen inzwischen bilden.

Eine der größten Herausforderungen beim Einsatz von Wireless-Technologien bleibt die Sicherheit. Es ist zwar auch nicht weiter schwierig, das Signal von Ethernet-Kabeln abzuhören, aber immerhin ist es vergleichsweise einfach, den physikalischen Zugang zu ihnen zu kontrollieren. Bei WLAN kann dagegen im Grunde genommen jeder die Signale mit einer kompatiblen Antenne auffangen und analysieren, um unberechtigt Informationen zu erhalten oder gar zu manipulieren. Das gilt umso mehr, als man die Grenzen der Funkreichweite niemals ganz genau auf die Größe des zu vernetzenden Gebäudes oder Geländes abstimmen kann; es ist also durchaus möglich, die Funkwellen außen zu empfangen.

Um ein Mindestmaß an Sicherheit zu gewährleisten, bot die ursprüngliche 802.11-Spezifikation eine optionale Verschlüsselung der Frames an. Allerdings ist diese Methode nicht besonders sicher; Sicherheitsexperten haben bereits bewiesen, dass die Verschlüsselung verhältnismäßig leicht zu knacken ist. Schon der Name dieser Technik, *WEP* (Wired Equivalent Privacy), sagt allzu deutlich aus, dass es nicht um mehr geht, als etwa dasselbe Maß an Sicherheit zu gewährleisten wie beim rein physikalischen Schutz verkabelter Netzwerke. Der Hauptverwendungszweck besteht auch gar nicht in der Geheimhaltung, sondern in der Abgrenzung eines Wireless-Netzes von benachbarten Netzen: Es ist ärgerlich, wenn jedes vorbeifahrende Fahrzeug, in dem sich zufälligerweise ein Laptop mit 802.11-Schnittstelle befindet, diesen vorübergehend automatisch ins Netz einbucht und wieder daraus verschwindet. Dies lässt sich allerdings zuverlässiger verhindern, indem der Access Point mit einer Whitelist zugelassener MAC-Adressen konfiguriert wird.

Inzwischen stehen mit WPA und WPA2 (WiFi Protected Access) stark verbesserte WLAN-Verschlüsselungsverfahren zur Verfügung.

## 4.5 Datenfernübertragung[Zur nächsten Überschrift](https://openbook.rheinwerk-verlag.de/it_handbuch/04_005.html#dodtp8f4212dc-b99b-4d4f-8c97-b47161e53e04)

Nachdem im vorigen Abschnitt die verschiedenen Formen der LAN-Vernetzung und der WANs über Standleitungen beschrieben wurden, sollen nun diverse Verfahren der Datenfernübertragung (DFÜ) geschildert werden. Wie bereits angesprochen, wurde DFÜ bereits eingesetzt, als sie lediglich der Punkt-zu-Punkt-Kommunikation zwischen einzelnen Rechnern über eine direkte Telefonverbindung diente. Heute geht es in der Regel darum, den Zugang zu einem bestehenden Netzwerk oder (über einen kommerziellen Provider) zum Internet herzustellen.

Die erste Generation der DFÜ-Hardware, der umständliche und störanfällige Akustikkoppler, muss hier nicht mehr beschrieben werden. Die drei wesentlichen Technologien sind heute Modems für den Netzwerkzugang über analoge Telefonleitungen, der Zugang über die digitale Telefonleitung ISDN sowie Hochfrequenz-Verbindungen über verschiedene DSL-Dienste. Diese verschiedenen Zugangsverfahren werden im Folgenden dargestellt.

Eine Gemeinsamkeit aller DFÜ-Netzwerkverbindungen besteht in der Notwendigkeit, die Datenübertragung über diese Leitungen zu standardisieren und bestimmte Grundlagen für die Protokolle der Vermittlungsschicht zu schaffen. Dafür werden spezielle Protokolle verwendet, die den Netzzugang über relativ langsame serielle Leitungen ermöglichen. Das traditionelle Protokoll für die Vernetzung über Wählleitungen war SLIP (Serial Line Interface Protocol). Allerdings besitzt es eine Reihe organisatorischer und technischer Mängel und wurde deshalb weitgehend durch PPP (Point-to-Point Protocol) ersetzt.

PPP kümmert sich um die Authentifizierung des Benutzers nach der Einwahl, indem Benutzername und Passwort übermittelt werden; anschließend verhandeln die beiden direkt miteinander verbundenen Punkte die eigentlichen Netzwerkdetails. Eine der wesentlichsten Fähigkeiten des Protokolls für Internetverbindungen besteht darin, dass der Einwahlknoten dem anwählenden Rechner automatisch eine IP-Adresse zuweisen kann, über die diese Netzwerkschnittstelle im gesamten Internet identifiziert wird.

Im Einzelnen erfolgen bei PPP also die folgenden Schritte:

* Wird eine Wählleitung (analog oder ISDN) verwendet, stellt der Rechner des Benutzers über die entsprechende Schnittstelle eine Telefonverbindung her. Falls die Leitung besetzt sein sollte, werden spezielle frei konfigurierbare Maßnahmen getroffen; in der Regel erfolgt nach einer gewissen Wartezeit ein erneuter Wählversuch. Bei DSL-Leitungen wird ebenfalls die Verbindung aktiviert, auch wenn man dies nicht als *Wählen* im klassischen Sinne bezeichnen kann.
* Der Einwahlknoten verlangt eine Authentifizierung, in der Regel in Form von Benutzername und Passwort. Die meisten PPP-Implementierungen in modernen Betriebssystemen übermitteln diese Daten nach einmaliger Konfiguration automatisch, ohne Zutun des Benutzers.
* Nachdem die Daten überprüft wurden, erfolgen entweder die Ablehnung des Benutzers und der Verbindungsabbau, oder die Netzwerkparameter werden ausgehandelt. Auch wenn PPP als Netzzugangsgrundlage für alle möglichen Protokolle der Vermittlungsschicht dienen kann, wird heute fast nur noch TCP/IP aufgesetzt. Zu diesem Zweck weist der PPP-Knotenpunkt des Internetproviders der seriellen Verbindung auf der Einwahlseite eine IP-Adresse zu, eine im gesamten Internet einmalige Identifikationsnummer. Ihr Konzept wird im nächsten Abschnitt genau beschrieben.

[Rheinwerk Computing - Zum Seitenanfang](https://openbook.rheinwerk-verlag.de/it_handbuch/04_005.html#top)

### 4.5.1 Netzwerkzugang per Modem (analoge Telefonleitung)[Zur nächsten Überschrift](https://openbook.rheinwerk-verlag.de/it_handbuch/04_005.html#dodtpbf029b61-62f9-4d39-8752-c882f1868a8d)[Zur vorigen Überschrift](https://openbook.rheinwerk-verlag.de/it_handbuch/04_005.html#dodtpcf2657be-ee2c-4769-bb7d-c96cb97811d2)

Für Modems wurden im Laufe der Zeit viele verschiedene Standards entwickelt, die sich insbesondere bezüglich ihrer maximalen Datenübertragungsrate voneinander unterscheiden. Der aktuelle Standard heißt V.90 und überträgt bis zu 56.600 Bit/s. Gemessen an den üblichen Geschwindigkeiten fest verdrahteter Netzwerke ist das natürlich sehr langsam, aber kein Vergleich zu den Modem-Geschwindigkeiten vergangener Jahrzehnte. Rein physikalisch scheint mit 56,6 KBit/s die Leistungsgrenze erreicht zu sein; etwas höhere Übertragungsraten lassen sich durch die heutzutage häufig verwendete Datenkomprimierung erzielen – je nach übertragener Datenart etwa bis zur doppelten Leistung.

Das Wort »Modem« ist eine Zusammensetzung aus den Abkürzungen für »Modulator« und »Demodulator«, weil es die digitalen Signale des Computers in frequenzmodulierte Analogimpulse umwandelt, diese über die Telefonleitung überträgt und am Ziel wieder zurückverwandelt. Zu diesem Zweck muss es auf der einen Seite mit dem Computer verbunden werden, zum Beispiel über USB, klassisch auch über die alte serielle Schnittstelle oder als PCI-Steckkarte. Auf der anderen Seite wird das Modem über einen TAE-Stecker an die Telefonbuchse angeschlossen. Praktisch sind in diesem Zusammenhang Dreifach-TAE-Dosen, die leicht nachgerüstet werden können: Sie verfügen über einen speziellen Anschluss (TAE-F) für ein Telefon in der Mitte und zwei Anschlüsse (TAE-N) für Zusatzgeräte – Faxgerät, Anrufbeantworter oder eben Modem – außen.

Das Modem wird vom Computer über ein einfaches ASCII-basiertes Protokoll gesteuert. Heutzutage verwenden praktisch alle Modems den sogenannten *Hayes-Befehlssatz* (benannt nach einem längst vergessenen Modem-Hersteller). Da die Befehle dieses Protokolls alle mit der Zeichenfolge »AT« beginnen, wird er mitunter auch als *AT-Befehlssatz* bezeichnet. Wichtige Befehle sind etwa folgende:

* ATDT <Rufnummer>: DT steht für »Dial Tone« – eine Rufnummer wird im Tonwahlverfahren (Mehrfrequenzverfahren) angewählt.
* ATDP <Rufnummer>: »Dial Pulse« – eine Rufnummer wird im Pulswahlverfahren angewählt (heute sehr selten).
* ATH: »Hangup« – die Telefonverbindung wird unterbrochen, es wird »aufgelegt«.
* ATZ: Das Modem wird auf den Einschaltzustand zurückgesetzt (Reset).

Das Tonwahlverfahren verwendet mehrere Töne unterschiedlicher Frequenzen, die zusammen die verschiedenen Ziffern und Funktionen des Telefons repräsentieren. Das Pulswahlverfahren sendet dagegen eine Reihe von »Klicktönen« – einen für eine 1, zwei für eine 2 und so weiter, bis zehn für eine 0. Seitdem alle Vermittlungsstellen in den deutschen Telefonnetzen digital sind, benötigt niemand mehr das langsamere und unzuverlässigere Pulswahlverfahren. Verwechseln Sie übrigens digitale Vermittlung nicht mit digitaler Signalübertragung – Letztere findet beispielsweise bei ISDN statt, das im nächsten Abschnitt behandelt wird.

Bevor eine Datenkommunikation überhaupt denkbar ist, müssen sich beide Seiten darüber einig sein, auf welche Weise sie die aufeinanderfolgenden einzelnen Bits überhaupt als Datenbits interpretieren sollen, was Datenbits, Stoppbits und eventuelle Parity-Bits angeht. Die meisten Internet-Einwahlpunkte verwenden heutzutage den Standard 8N1 (8 Datenbits, kein Parity-Bit, 1 Stoppbit). Dies muss in den Modem-Konfigurationsdaten eingetragen werden. Eine Beschreibung der verschiedenen Formen der seriellen Datenübertragung finden Sie in [Kapitel 3](https://openbook.rheinwerk-verlag.de/it_handbuch/03_001.html#rxz1001723), »Hardware«.

Nach der Herstellung der eigentlichen Telefonverbindung findet der sogenannte *Handshake* (»Handschlag«) zwischen den beiden Gegenstellen statt. Es wird eine Übertragungskapazität ausgehandelt, die beide Seiten verwenden können. Erst nachdem die grundlegende Datenübertragung funktioniert, wird PPP eingesetzt, um die eigentliche Netzwerkverbindung über die Telefonverbindung herzustellen, wie hier bereits beschrieben wurde.

[Rheinwerk Computing - Zum Seitenanfang](https://openbook.rheinwerk-verlag.de/it_handbuch/04_005.html#top)

### 4.5.2 ISDN[Zur nächsten Überschrift](https://openbook.rheinwerk-verlag.de/it_handbuch/04_005.html#dodtp091a81f9-8fb3-4f8d-9926-5935785eb1bc)[Zur vorigen Überschrift](https://openbook.rheinwerk-verlag.de/it_handbuch/04_005.html#dodtp8f4212dc-b99b-4d4f-8c97-b47161e53e04)

Das Integrated Services Digital Network (etwa »Digitalnetzwerk mit integrierten Diensten«) oder kurz ISDN wurde in den 80er-Jahren von verschiedenen europäischen Telefongesellschaften eingeführt; die aktivste von ihnen dürfte die in Deutschland damals noch zuständige Deutsche Bundespost gewesen sein. Es handelt sich im Prinzip um die Übertragung digitaler Signale über klassische Kupferdraht-Telefonleitungen. Einem reinen Telefonkunden bietet ISDN zunächst die folgenden unmittelbaren Vorteile:

* Es werden zwei voneinander unabhängige Kanäle zur Verfügung gestellt; über beide kann gleichzeitig telefoniert, gefaxt oder Datenfernübertragung betrieben werden.
* Die Rufnummer eines Anrufers, der ebenfalls ISDN verwendet, wird übermittelt und im Display eines entsprechend ausgerüsteten Telefons angezeigt (dies funktioniert natürlich auch bei Standard-Telefonleitungen mit Digitalvermittlung).
* Während eine Verbindung besteht, kann ein weiterer Anruf angenommen werden. Entweder wird der jeweils andere Gesprächspartner in den Wartezustand versetzt (Makeln), oder der neue Anrufer wird mit dem bisherigen in dieselbe Verbindung aufgenommen (Dreierkonferenz).

Inzwischen stehen die meisten dieser Dienste dank der flächendeckend digitalen Vermittlung in Deutschland auch Analogkunden zur Verfügung; lediglich die beiden separaten Kanäle bleiben ISDN vorbehalten. Im Übrigen erhalten ISDN-Benutzer üblicherweise drei unabhängige Rufnummern (bei einigen Telefongesellschaften sogar noch mehr), die frei auf die jeweiligen Geräte verteilt werden können.

Technisch betrachtet werden sogar drei Kanäle zur Verfügung gestellt; die beiden B- Kanäle übertragen Telefongespräche, Faxe oder Daten mit jeweils 64 KBit/s, während der D-Kanal Dienstinformationen wie Rufnummernübermittlung oder Anklopfen mit 16 KBit/s überträgt. Für Internetverbindungen und andere Datenübertragungsmethoden besteht die Möglichkeit, beide B-Kanäle zu bündeln und auf diese Weise insgesamt eine Datenübertragungsrate von 128 KBit/s zu gewährleisten; natürlich entstehen dafür auch doppelte Kosten.

In der Praxis funktioniert ISDN folgendermaßen: An die normale TAE-Telefonsteckdose wird ein spezielles ISDN-Endgerät namens *NTBA* angeschlossen. Es stellt zwei sogenannte *S0-Basisanschlüsse* zur Verfügung. Diese verwenden die auch vom Twisted-Pair-Ethernet bekannten RJ-45-Stecker. An jeden dieser Anschlüsse kann ein ISDN-Endgerät angeschlossen werden, beispielsweise ein Telefon, ein Faxgerät oder ein ISDN-Adapter zur Computerdatenübertragung. Damit niemand seinen kompletten Telekommunikations-Gerätepark umstellen muss, werden als spezielle Form von ISDN-Endgeräten sogenannte *TK-Anlagen* angeboten, die wiederum den Anschluss analoger Endgeräte ermöglichen. Einige TK-Anlagen bieten auch durchgeschleifte S0-Anschlüsse an, beispielsweise um ISDN-Geräte an einer praktischeren Stelle anzuschließen.

Natürlich sollten Sie nicht versuchen, ein Analogmodem an eine TK-Anlage anzuschließen – es würde nicht etwa nur den eigentlichen Vorteil der ISDN-Datenübertragung zunichte machen, sondern funktioniert gar nicht. Da die übertragenen Daten nach außen wie ISDN aussehen, würde die Gegenstelle ihre Antworten mit einer Übertragungsgeschwindigkeit übermitteln, die das Modem nicht verarbeiten kann.

Um einen Computer mit ISDN zu verbinden, werden stattdessen verschiedene Formen von ISDN-Adaptern angeboten: als PCI-Steckkarten (früher gab es sogar ISA-Modelle), externe USB-Geräte oder PCMCIA-Geräte für Notebooks. Mittlerweile werden einige externe ISDN-Geräte auch mit integrierter TK-Anlagen-Funktion angeboten. Eine interne ISDN-Karte sieht genauso aus wie eine moderne Ethernet-Karte, und es kann leicht passieren, dass man das Twisted-Pair-Netzwerkkabel mit seinem baugleichen Stecker in die ISDN-Karte steckt und umgekehrt (selbstverständlich geschieht in diesem Fall gar nichts). Jedenfalls muss normalerweise ein ISDN-Kabel vom Anschluss des ISDN-Adapters zu einem S0-Anschluss verlaufen.

Der Unterschied zwischen den Übertragungsraten eines heutigen Modems (56,6 KBit/s) und Ein-Kanal-ISDN (64 KBit/s) mag Ihnen nicht besonders groß erscheinen. Als jedoch das Internet für Privatkunden interessant zu werden begann, lag die Übertragungsrate der meisten Modems bei 9.600 oder 14.400 Bit/s; erst allmählich kamen Geräte mit 28.800 Bit/s hinzu. Abgesehen davon besitzt ISDN noch heute einen weiteren Vorteil gegenüber Modem-Verbindungen: Der Verbindungsaufbau geht fast ohne Verzögerung vonstatten, während es bei Modems zu Wartezeiten von etlichen Sekunden kommen kann, bis die Leitung bereit ist.

Neben der Verbindung zu einem Internetprovider, die mittlerweile wohl die häufigste über ISDN genutzte Dienstleistung ist, war über Jahre hinweg auch die direkte Verbindung zwischen Computern für die ISDN-Datenübertragung üblich. Sehr viele Macintosh-Benutzer verwendeten hierfür regelmäßig die Software Leonardo, während Windows-Benutzern beispielsweise das Programm Fritz!Data zur Verfügung stand, das mit der in Deutschland besonders populären Fritz!Card der Berliner Firma AVM geliefert wurde. Auch viele Mailbox/BBS-Systeme wurden in der zweiten Hälfte der 90er-Jahre auf ISDN umgestellt oder um eine ISDN-Einwahlmöglichkeit erweitert.

Insgesamt lässt sich feststellen, dass ISDN erst mit dem Aufkommen von Internetzugängen in Firmen und Privathaushalten wirklich populär wurde. Zuvor wurde es manchmal als Telefonleitung für Firmen eingesetzt, allerdings nicht annähernd so häufig, wie die Telefongesellschaften sich dies erhofft hatten. Als ISDN dann schließlich immer öfter benutzt wurde, reichte seine Übertragungsrate immer mehr Nutzern nicht mehr aus; das Bedürfnis nach multimediafähigen Breitband-Verbindungen wuchs deutlich. Dies führte zur Einführung der im nächsten Abschnitt vorgestellten DSL-Dienste.

[Rheinwerk Computing - Zum Seitenanfang](https://openbook.rheinwerk-verlag.de/it_handbuch/04_005.html#top)

### 4.5.3 DSL-Dienste[Zur nächsten Überschrift](https://openbook.rheinwerk-verlag.de/it_handbuch/04_005.html#dodtp4112f431-eb73-4df9-87e3-1835fedc0b95)[Zur vorigen Überschrift](https://openbook.rheinwerk-verlag.de/it_handbuch/04_005.html#dodtpbf029b61-62f9-4d39-8752-c882f1868a8d)

*DSL* ist die Abkürzung für »Digital Subscriber Line« (etwa »digitale Abonnement-Leitung«). Der Name soll verdeutlichen, dass es sich de facto um eine Standleitung anstelle einer Wählleitung handelt. Zur Einführung von DSL kam es, da es durch die allmähliche Verbesserung der Qualität von Telefonleitungen möglich wurde, Signale hoher Frequenz zu übertragen. Auch die meisten DSL-Dienste verwenden also genau wie Modem- und ISDN-Verbindungen die klassischen Kupferleitungen der Telefongesellschaften, die allerdings immer häufiger durch Glasfaserleitungen ersetzt oder ergänzt werden.

# DSL-Varianten

Es existieren zwei grundsätzliche Varianten von DSL: Bei *Symmetric DSL* (SDSL) sind die Übertragungsraten für ankommende und ausgehende Daten identisch, bei *Asymmetric DSL* (ADSL) ist die ankommende Übertragungsrate höher als die ausgehende. SDSL ist eher für mittlere bis große Unternehmen geeignet, die nicht nur permanent auf das Internet zugreifen, um im Web zu recherchieren oder ihre E-Mails zu lesen, sondern bei denen auch eine Menge Datenausgänge stattfinden – beispielsweise für den Vor-Ort-Betrieb eigener Web- oder Mailserver oder für den Direktzugriff auf das Firmennetzwerk durch externe Mitarbeiter. ADSL dagegen wird häufiger von Privatkunden oder kleineren Firmen eingesetzt, die recht hohe Datenmengen aus dem Internet herunterladen, aber nur verhältnismäßig wenige und eher kleine Uploads durchführen.

Übliche ADSL-Angebote wie T-DSL der Deutschen Telekom stellten ursprünglich eine Download-Rate von 1.024 KBit/s und eine Upload-Rate von 128 KBit/s zur Verfügung, teilweise sogar noch weniger. Aktuelle ADSL-Anschlüsse des klassischen Typs sind dagegen mit Download-Geschwindigkeiten von 2 bis 8 MBit/s ausgestattet. Auch die umgekehrte Datenrate wurde entsprechend vervielfacht. Die neueren Typen ADSL2 und ADSL2+, die allerdings eine nicht überall verfügbare hohe Leitungsqualität und spezielle Hardware benötigen, schaffen sogar 16 beziehungsweise 25 MBit/s im Download.

SDSL-Lösungen werden von sehr vielen kommerziellen Providern angeboten und stellen je nach Bedarf viele verschiedene Übertragungsraten von 1.024 KBit/s bis hin zu mehreren MBit/s zur Verfügung. Sie sind deutlich teurer als ADSL-Angebote mit der gleichen oder gar mit einer höheren Übertragungsrate, weil die entsprechende Technik aufwendiger ist.

Anders als bei Modem- oder ISDN-Angeboten werden die Gebühren für DSL-Anschlüsse in der Regel nicht nach der Nutzungsdauer berechnet, sondern als sogenannte *Flatrate* für beliebig lange Onlinezeiten. Einige Provider verwenden allerdings eine Volumenbeschränkung, das heißt, ohne Aufpreis darf monatlich nur eine bestimmte Datenmenge transferiert werden.

Neben den DSL-Angeboten, die über normale Telefonleitungen laufen, werden seit einiger Zeit auch spezielle Lösungen angeboten. Eine davon ist die Internetverbindung über das Glasfaserkabel des Kabelfernsehens. Da dieses Kabel für das Passivmedium Fernsehen erfunden wurde, besitzt es in seiner ursprünglichen Version keine Rückkanal-Fähigkeit. Es können Daten empfangen, aber nicht gesendet werden; noch nicht einmal die Anforderung einer URL kann abgesetzt werden. Erst allmählich wird der Rückkanal derjenigen Kabelnetze nachgerüstet, die die Deutsche Telekom bereits verkauft hat.

Eine andere Lösung ist besonders interessant für kleine Gemeinden, die so weit von der nächsten größeren Stadt entfernt liegen, dass sich eine Nachrüstung der bestehenden Telefonleitungen oder Fernsehkabelnetze nicht lohnt: die Kommunikation mit einem Satelliten über eine Parabolantenne. Diese Lösung bietet beispielsweise die Firma Teles unter dem Namen skyDSL an. Die Datenübertragungsrate beträgt bis zu 24.000 KBit/s. Über den Satelliten ist allerdings nur der Datenempfang möglich; Anfragen und andere Sendevorgänge erfolgen über Analogmodem, ISDN oder GSM-Mobilfunk.

#### ADSL anschließen

An den TAE-Anschluss eines ADSL-Kunden wird ein sogenannter *Splitter* angeschlossen – eine Frequenzweiche, die die hochfrequenten DSL-Signale und die niedrigfrequenten normalen Telefonsignale voneinander trennt. Den Ausgang für die Telefonsignale bietet wiederum ein TAE-Anschluss, an den entweder ein Analog-Telefon oder ein NTBA angeschlossen wird, je nachdem, ob ADSL mit einem Analog- oder mit einem ISDN-Telefonanschluss kombiniert wird.

Den Ausgang für die speziellen ADSL-Signale bietet eine Twisted-Pair-Buchse vom Typ RJ-11. An diesen Anschluss wird in der Regel ein ADSL-Modem angeschlossen, das dann über USB oder Twisted-Pair-Ethernet mit dem Computer verbunden wird. Natürlich ist die Bezeichnung *ADSL-Modem* technisch gesehen Unfug. Bei DSL findet keinerlei Analog-Digital-Umwandlung statt. Dennoch ist der Begriff *Modem* für das Gerät weit verbreitet, weil es den Computer mit einer seriellen Fernleitung verbindet. Beim Anschluss über eine Ethernet-Schnittstelle kommt eine spezielle PPP-Variante namens *PPPoE* (PPP over Ethernet) zum Einsatz.

Statt der reinen ADSL-Modems zum Anschluss eines einzelnen Rechners werden inzwischen meist DSL-Router verwendet, die gleich einem gesamten Netzwerk per Ethernet, WLAN oder beidem den Internetzugang bereitstellen. Inzwischen erlauben auch die meisten Provider den Einsatz solcher Router; früher waren die günstigsten Tarife dagegen vielfach auf einen Einzelrechner beschränkt. Die jüngste Generation von DSL-Routern verzichtet auch gleich auf einen externen Splitter und bringt diese Funktionalität selbst mit.

[Rheinwerk Computing - Zum Seitenanfang](https://openbook.rheinwerk-verlag.de/it_handbuch/04_005.html#top)

### 4.5.4 Internetzugänge über Mobilfunk[Zur vorigen Überschrift](https://openbook.rheinwerk-verlag.de/it_handbuch/04_005.html#dodtp7825bbc7-db16-436e-af07-9207471e8618)

Neben den hier behandelten stationären DFÜ-Verbindungen werden auch diejenigen über Mobilfunk immer wichtiger. Über die seit den 90er-Jahren errichteten GSM-Netze (in Deutschland beispielsweise D1, D2, E-Plus und so weiter) kam ursprünglich vor allem ein Verfahren namens *GPRS* (General Packet Radio Service) zum Einsatz; es wurde mehrfach in Details verbessert, bietet aber noch immer keine allzu hohen Datentransferraten – sie liegen bei 53,6 KBit/s im Download und 26,8 KBit/s im Upload. Neuere Verfahren sind zum Teil erheblich schneller:

* EDGE: Download 217,6 KBit/s, Upload 108,8 KBit/s
* UMTS: Download und Upload 384 KBit/s
* HSPA: Download (HSDPA) 7,2 Mbit/s, Upload (HSUPA) 1,4 Mbit/s
* LTE (Long-Term Evolution) ist ein neuer Standard, den erst wenige Geräte und Mobilfunkanbieter unterstützen. In der ersten Ausbaustufe werden Übertragungsraten bis 100 MBit/s realisiert. Experimentell sind auch bereits Übertragungen im Gigabit-Bereich möglich; diese werden jedoch erst im Rahmen der nächsten Ausbaustufe (LTE-Advanced) ab Ende 2013 auf den Markt kommen.

Alternativ werden die verschiedenen mobilen Datenübertragungsstandards mit Generationsnummern bezeichnet: 1G bezeichnet historische Mobilfunknetze vor der Handy-Revolution der 90er-Jahre, 2G ist der GSM-Mobilfunk, 2.5G ist GPRS, und 3G steht für EDGE, UMTS und HSPA. LTE wird im Marketing zwar mit dem Modebegriff *4G* beworben, streng genommen handelt es sich jedoch um 3.9G als Weiterentwicklung von 3G. Erst LTE-Advanced wird die 4G-Spezifikationen voll erfüllen.

Mobilfunkzugänge können entweder für Web-, E-Mail- und andere Netzwerksoftware auf dem Mobiltelefon selbst verwendet werden, oder aber das Handy dient – beispielsweise über Bluetooth – als Mobilfunkmodem für einen Laptop oder ein Nur-WLAN-Tablet (der Fachbegriff dafür lautet *Tethering*). Speziell für UMTS oder HSPA gibt es auch eigenständige Netzwerkzugangsgeräte, die per USB an den Rechner angeschlossen werden und beinahe überall einen Internetzugang mit annehmbarer Geschwindigkeit bieten.

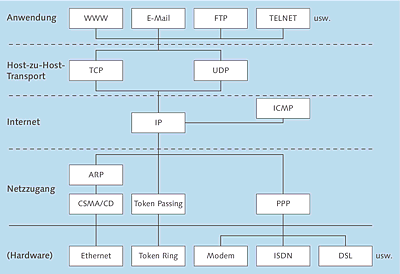
Um die Vorteile eines solchen Anschlusses wirklich zu nutzen, sollte aus Kostengründen ein Datenflatrate-Vertrag mit dem Mobilfunkanbieter abgeschlossen werden. Beachten Sie aber, dass diese Zugänge sehr oft eine Transfervolumenbeschränkung enthalten. Nachdem das Volumen für den entsprechenden Monat aufgebraucht ist, steht in der Regel nur noch eine langsamere Verbindung wie etwa EDGE zur Verfügung.[Anm.: Die Telekom kündigte im Frühjahr 2013 an, solche Regelungen künftig auch für DSL-Anschlüsse einführen zu wollen. Für Empörung sorgte dabei insbesondere, dass sie ihre eigenen Web-TV- und Unterhaltungsangebote nicht in dieses Volumen einzurechnen gedenkt (mögliche Verletzung der Netzneutralität). Nach sehr viel Kritik und Spott wurden die Pläne inzwischen zumindest aufgeweicht; Konkurrenzdruck könnte möglicherweise dazu führen, dass sie endgültig fallen gelassen werden.] Glücklicherweise verfügen praktisch alle modernen Smartphones (und Tablets sowieso) auch über WLAN, sodass zu Hause oder am Arbeitsplatz meist keine Datenübertragung über Mobilfunk erforderlich ist.

Für den Urlaub oder ähnliche Gelegenheiten werden auch Prepaid-SIM-Karten oder USB-Sticks angeboten. Auf keinen Fall sollten Sie den Fehler machen, im Ausland ungeprüft das sogenannte *Daten-Roaming* Ihres Mobiltelefons zu aktivieren, da es geradezu irrational teuer ist (zum Beispiel 1 Euro pro 50 Kilobyte übertragener Daten). Verschiedene Initiativen der EU-Kommission, diese Tarife zu senken, zeigen erst allmählich Wirkung.

## 4.6 Die TCP/IP-Protokollfamilie[Zur nächsten Überschrift](https://openbook.rheinwerk-verlag.de/it_handbuch/04_006.html#dodtpe6e706c8-0915-41ab-842f-d1c71814d4a2)

Nach einigen halbherzigen Versuchen, das OSI-Referenzmodell durch konkrete Protokolle tatsächlich zu implementieren, bemerkte man letzten Endes, dass die bereits Jahre zuvor entwickelten Internetprotokolle hervorragend als flexible, skalierbare und universelle Netzwerkprotokollfamilie einsetzbar sind. Die rasante Ausbreitung des Internets und die freie Verfügbarkeit sorgten dafür, dass diese Protokolle heute häufiger als jeder andere Protokollstapel eingesetzt werden.

[Abbildung 4.3](https://openbook.rheinwerk-verlag.de/it_handbuch/04_006.html#rxz1004532) zeigt eine konkrete Version des zuvor bereits vorgestellten TCP/IP-Protokollstapels: Auf jeder Ebene sind einige der Protokolle zu erkennen, die dort arbeiten können. Die meisten davon werden in den folgenden Abschnitten genau erläutert; die Netzzugangsprotokolle der untersten Schicht wurden ebenfalls bereits vorgestellt. Ganz zuletzt habe ich zusätzlich einige Beispiele für die Hardware angegeben, auch wenn sie kein Teil des eigentlichen TCP/IP-Stapels ist.

[](https://openbook.rheinwerk-verlag.de/it_handbuch/bilder/04_003.png)

**Abbildung 4.3**Der TCP/IP-Protokollstapel

Zwischen der Hardware und dem Netzzugang auf der einen und den anwendungsorientierten Protokollen auf der anderen Seite befinden sich die Protokolle der Vermittlungs- und der Transportschicht. Insgesamt werden alle Protokolle, die auf den verschiedenen Ebenen eines Schichtenmodells zusammenarbeiten, als *Protokollstapel* oder auch *Protokollfamilie* bezeichnet. Allerdings konzentriert sich der Schwerpunkt von TCP/IP auf die beiden mittleren Ebenen des Internet-Protokollstapels. Sie können zum einen auf fast jeden beliebigen Netzzugang aufsetzen, zum anderen wurde beinahe jede ernstzunehmende Netzwerkanwendung inzwischen für diesen Protokollstapel umgesetzt – abgesehen von den klassischen Internetanwendungen, die ohnehin dafür geschrieben wurden.

Die Protokolle der mittleren Schichten sind dafür verantwortlich, dass Daten zuverlässig über verschiedene Teilnetze oder Netzwerksegmente hinweg übertragen werden können oder auch über Netze, die verschiedene Hardware oder Netzzugangsverfahren verwenden:

* Die Protokolle der Internetschicht regeln die Adressierung der Rechner und die Übertragung der Daten an den korrekten Rechner im Netzwerk. Darüber hinaus kümmern sie sich darum, dass Daten bei Bedarf in andere Teilnetze weitergeleitet werden, übernehmen also das sogenannte *Routing*.
* Auf der Host-zu-Host-Transportschicht werden die Daten in Pakete unterteilt sowie mit der Information versehen, welche Anwendung auf dem einen Host diese Daten an welche Anwendung auf dem anderen sendet.

Die Bezeichnung *TCP/IP* kombiniert die Namen der beiden wichtigsten Bestandteile des Protokollstapels: das Internet Protocol (IP) auf der Internetschicht und das Transmission Control Protocol (TCP), das am häufigsten verwendete Protokoll der Transportebene. In den folgenden Abschnitten werden diese Protokolle näher vorgestellt, anschließend wird die technische Seite einiger wichtiger Internet-Anwendungsprotokolle beleuchtet.

[Rheinwerk Computing - Zum Seitenanfang](https://openbook.rheinwerk-verlag.de/it_handbuch/04_006.html#top)

### 4.6.1 Netzzugang in TCP/IP-Netzwerken[Zur nächsten Überschrift](https://openbook.rheinwerk-verlag.de/it_handbuch/04_006.html#dodtpe3bff1ac-8054-47ed-ac48-ca276c03909c)[Zur vorigen Überschrift](https://openbook.rheinwerk-verlag.de/it_handbuch/04_006.html#dodtpc313636c-e72e-4e0b-b33a-465f01565780)

Die unterste Ebene des Internet-Schichtenmodells ist der Netzzugang, nicht die Netzwerkhardware. Dies garantiert, dass sich die Internetprotokolle auf fast jeder beliebigen Hardware implementieren lassen, und in der Tat ist dies geschehen: In allen Formen von LANs wie Ethernet oder Token Ring, in WANs über Wähl- und Standleitungen wie auch über die meisten Formen von drahtlosen Netzen – überall laufen diese Protokolle. Dies ist ein weiterer guter Grund dafür, dass sich die Protokolle des Internets als Standard für die Netzwerkkommunikation durchsetzen konnten.

Im Grunde wird innerhalb der Spezifikation der TCP/IP-Protokolle nicht einmal der Netzzugang im OSI-Sinn beschrieben, sondern lediglich die Zusammenarbeit des IP-Protokolls, das sich innerhalb des Internet-Protokollstapels um Adressierung und Routing kümmert, mit verschiedenen Netzzugangsverfahren.

An dieser Stelle sollen nur zwei der wichtigsten Internet-Netzzugangsverfahren genannt werden: das für den Zugriff auf Ethernet verwendete Address Resolution Protocol (ARP) und das Point-to-Point Protocol (PPP), das für serielle Verbindungen über Modem, ISDN oder DSL eingesetzt wird. PPP wurde in diesem Kapitel bereits ausführlich beschrieben. Hier das Wichtigste zu ARP:

Das Address Resolution Protocol, beschrieben in RFC 826, übernimmt kurz gesagt die Umsetzung der vom Netzwerkadministrator vergebenen IP-Adressen in die vorgegebenen Hardwareadressen der Netzwerkschnittstellen.

Da die IP-Adresse den einzelnen Hosts willkürlich zugeteilt wird, kann sie auf der Netzzugangsschicht nicht bekannt sein: Auf einer bestimmten Schicht eines Protokollstapels werden die Steuerdaten der höher gelegenen Ebenen nicht ausgewertet, sondern als gewöhnliche Nutzdaten betrachtet. Deshalb kann beispielsweise eine Netzwerkkarte oder ein Hub nicht anhand der IP-Adresse entscheiden, für welche Station ein Datenpaket bestimmt ist; die Netzwerkhardware nimmt diese Adresse nicht einmal wahr.

Nachdem die IP-Software auf einem Host oder Router anhand der Empfänger-IP-Adresse festgestellt hat, dass die Daten überhaupt für das eigene Netz bestimmt sind, wird der ARP-Prozess gestartet, um diese IP-Adresse in die MAC-Adresse der Empfänger-Schnittstelle umzusetzen. Zu diesem Zweck sendet ein Rechner, der das ARP-Protokoll ausführt (beinahe jeder Rechner, der TCP/IP über Ethernet betreibt), ein sogenanntes *Broadcast-Datenpaket* in das Netzwerk. Es handelt sich um ein Datenpaket mit einer speziellen Empfängeradresse, das an alle Rechner im Netzwerk übertragen wird. Der Rechner, der seine eigene IP-Adresse im Inhalt dieses Pakets erkennt, antwortet als Einziger auf diese Anfrage und versendet seine eigene MAC-Adresse. Auf diese Weise wird ermittelt, für welchen Rechner das Datenpaket bestimmt ist.

In Ausnahmefällen kann ein Rechner auch die MAC-Adressen anderer Stationen zwischenspeichern und in Vertretung antworten.

[Rheinwerk Computing - Zum Seitenanfang](https://openbook.rheinwerk-verlag.de/it_handbuch/04_006.html#top)

### 4.6.2 IP-Adressen, Datagramme und Routing[Zur nächsten Überschrift](https://openbook.rheinwerk-verlag.de/it_handbuch/04_006.html#dodtpcd8c090c-d819-493c-b16d-cf9dff2ebdc4)[Zur vorigen Überschrift](https://openbook.rheinwerk-verlag.de/it_handbuch/04_006.html#dodtpe6e706c8-0915-41ab-842f-d1c71814d4a2)

Auf der Internet- oder Vermittlungsschicht des Internet-Protokollstapels arbeitet das Internet Protocol (IP). Als *Internet* wird in diesem Zusammenhang jedes Netzwerk bezeichnet, das diese Protokollfamilie verwendet. Dies verdeutlicht den Umstand, dass die Internetprotokolle dem Datenaustausch über mehrere physikalische Netzwerke hinweg dienen können. Spezielle Rechner, die mindestens zwei Netzwerkschnittstellen besitzen, leiten die Daten zwischen diesen Netzen weiter. Sie werden als *IP-Router* oder *IP-Gateways* bezeichnet. Im engeren Sinne ist ein Router ein Rechner, der Daten zwischen zwei Netzen des gleichen physikalischen Typs weiterleitet; ein Gateway verbindet dagegen zwei physikalisch verschiedene Netze oder arbeitet gar auf Anwendungsebene (*Application Level Gateway*). Die beiden Begriffe werden jedoch oft synonym verwendet.

Eine IP-Adresse des klassischen Typs – der Version IPv4 gemäß RFC 791 – ist eine 32 Bit lange Zahl. Sie wird üblicherweise in vier durch Punkte getrennte Dezimalzahlen zwischen 0 und 255 geschrieben. Allerdings ist die Logik, die einer solchen Adresse zugrunde liegt, besser verständlich, wenn diese binär notiert wird:

Eine typische IP-Adresse wäre etwa 11000010000100010101000111000001, in 8-Bit-Gruppen getrennt ergibt sich daraus 11000010 00010001 01010001 11000001; dies lautet in der gängigen Schreibweise dann 194.17.81.193.

# IP-Adressklassen

IP-Adressen bestehen aus zwei Komponenten: dem Netzwerkteil und dem Hostteil. Der Netzwerkteil gibt an, in welchem Netz sich der entsprechende Rechner befindet, während der Hostteil den einzelnen Rechner innerhalb dieses Netzes identifiziert.

Es gibt verschiedene Sorten von IP-Adressen, die sich bezüglich der Länge des Netzwerk- beziehungsweise Hostteils voneinander unterscheiden. Traditionell wurden die verfügbaren Adressen in feste Klassen unterteilt. Bereits 1993 wurde die Klasseneinteilung aufgegeben und durch CIDR ersetzt (RFC 1518 und 1519), da sie für die rasant steigende Anzahl von Hosts erheblich zu unflexibel war. Aus diesem Grund wurde das CIDR-Verfahren entwickelt, das dynamisch zwischen dem Netzwerk- und dem Hostteil einer Adresse trennt. Dennoch sollen an dieser Stelle zuerst die ursprünglichen Klassen vorgestellt werden, denn auf diese Weise wird vieles leichter verständlich. Beachten Sie aber, dass IP-Adressen heutzutage immer über CIDR vergeben werden. Dabei wird neben der IP-Adresse auch die im Folgenden definierte Teilnetzmaske mitgeliefert, die in der historischen Klassenlogik nicht benötigt wurde.

Zu welcher Klasse eine IP-Adresse gehörte, zeigte sich an den Bits, die am weitesten links standen:

* Klasse A**:** Das erste Bit ist 0, folglich liegt die erste 8-Bit-Gruppe zwischen 0 und 127.
* Klasse B**:** Die ersten beiden Bits lauten 10; die erste Gruppe liegt im Bereich 128 bis 191.
* Klasse C**:** Die ersten drei Bits sind 110, sodass die erste Gruppe zwischen 192 bis 223 liegt.
* Klasse D**:** Die ersten vier Bits sind 1110; die Adressen beginnen mit 224 bis 239.

Die restlichen Adressen, die mit 240 bis 255 anfangen, wurden weder im Klassenbereich noch über CIDR vergeben und sind für zukünftige Anwendungszwecke reserviert. Diejenigen mit der Anfangssequenz 11110 (Startbyte 240 bis 247) wurden manchmal trotzdem als *Klasse E* bezeichnet.

Je nach Klasse ist der Teil, der das Netzwerk kennzeichnet, unterschiedlich lang, entsprechend existieren unterschiedlich viele Netze der verschiedenen Klassen. Die Bits, die ganz rechts in der Adresse stehen und nicht zum Netzwerkteil gehören, sind die Hostbits. Je nach Länge des Netzwerkteils bleiben unterschiedlich viele Bits für den Hostteil übrig, sodass die Höchstzahl der Rechner in einem Netz variiert.

[Tabelle 4.5](https://openbook.rheinwerk-verlag.de/it_handbuch/04_006.html#rxz1004632) zeigt die wichtigsten Informationen zu den einzelnen Klassen im Überblick. In der Spalte »Netzwerkbits« stehen jeweils zwei Werte. Der erste stellt die Anzahl von Bits dar, die insgesamt den Netzwerkteil bilden. Da die Grenzen zwischen Netzwerk- und Hostteil an den Byte-Grenzen verlaufen, handelt es sich je nach Klasse um 1 bis 3 Byte. Da jedoch die Bits am Anfang der Adresse – wie zuvor gezeigt – die Klasse angeben, besteht die praktisch nutzbare Netzwerkangabe nur aus 7, 14 beziehungsweise 21 Bit. Der Rest der Adresse bildet den Hostteil, der je nach Klasse unterschiedlich groß ausfällt.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tabelle 4.5**Die IP-Adressklassen | | | | | |
| **Klasse** | **Adressbereich** | **Netzwerkbits** | **Hostbits** | **Anzahl Netze** | **Adressen pro Netz** |
| A | 0.0.0.0 bis 127.255.255.255 | 8 (7) | 24 | 128 | 16,7 Mio. |
| B | 128.0.0.0 bis 191.255.255.255 | 16 (14) | 16 | 16.384 | 65.536 |
| C | 192.0.0.0 bis 223.255.255.255 | 24 (21) | 8 | 2.097.152 | 256 |
| D | 224.0.0.0 bis 239.255.255.255 | spezieller Bereich der Multicast-Adressen | | | |

Innerhalb eines einzelnen Netzes – egal, welcher Klasse – stehen die erste und die letzte mögliche Adresse nicht als Hostadressen zur Verfügung: Die niedrigste Adresse identifiziert das gesamte Netz als solches nach außen hin, aber keinen speziellen Host; die höchste Adresse ist die sogenannte *Broadcast-Adresse*: Werden Datenpakete innerhalb des Netzes an diese Adresse gesendet, werden sie von jedem Host empfangen.

Zum Beispiel bilden die Adressen, die mit 18.x.x.x beginnen, das Klasse-A-Netzwerk 18.0.0.0 mit der Broadcast-Adresse 18.255.255.255 und Hostadressen von 18.0.0.1 bis 18.255.255.254. Dieses Netz kann theoretisch bis zu 16.777.214 Hosts beherbergen (224 – 2).

Die Adressen, die mit 162.21.x.x anfangen, befinden sich in dem Klasse-B-Netzwerk 162.21.0.0, dessen Broadcast-Adresse 162.21.255.255 lautet. Es kann bis zu 65.534 Hosts (216 – 2) mit den Adressen 162.21.0.1 bis 162.21.255.254 enthalten.

Letztes Beispiel: Adressen, die mit 201.30.9.x beginnen, liegen in dem Klasse-C-Netz 201.30.9.0 mit der Broadcast-Adresse 201.30.9.255; die 254 möglichen Hostadressen (28 – 2) sind 201.30.9.1 bis 201.30.9.254.

Die sogenannten *Multicast-Adressen* der Pseudo-Klasse D nehmen eine Sonderstellung ein: Eine Multicast-Gruppe ist eine auf beliebige Netze verteilte Gruppe von Hosts, die sich dieselbe Multicast-IP-Adresse teilen. Dies ermöglicht einen erheblich ökonomischeren Versand von Daten, da sie nicht mehr je einmal pro empfangendem Host versendet werden, sondern nur noch kopiert werden müssen, wo Empfängerrechner in unterschiedlichen Teilnetzen liegen. Aus diesem Grund ist Multicasting eine zukunftsträchtige Technologie für datenintensive Anwendungen wie etwa Videokonferenzen. Im Gegensatz dazu werden die individuellen Hostadressen als *Unicast-Adressen* bezeichnet.

#### Die Verteilung der IP-Adressen

Alle Adressen des IPv4-Adressraums werden von der Internet Assigned Numbers Association (IANA) verwaltet. Falls Sie jedoch für bestimmte Anwendungen in Ihrem Unternehmen eine oder mehrere feste IP-Adressen benötigen, dann sollten Sie sich in der Regel an einen Internetprovider und nicht an die IANA selbst wenden.

Die 128 Netze der Klasse A sind bereits alle vergeben; in der Regel an große internationale Unternehmen aus dem Elektronik- und Computerbereich sowie an US-amerikanische Staats-, Militär- und Bildungsinstitutionen. Beispielsweise gehört das Netz 17.0.0.0 der Firma Apple, 18.0.0.0 dem Massachusetts Institute of Technology (MIT) und 19.0.0.0 der Ford Motor Company.

Die 16.384 Klasse-B-Netze sind ebenfalls weitgehend vergeben, insbesondere an US-amerikanische Unternehmen und Internetprovider.

Die mehr als zwei Millionen Netze der Klasse C schließlich sind inzwischen ebenfalls überwiegend belegt. Die meisten von ihnen gehören Unternehmen und Internetprovidern, die nicht in den USA residieren, sondern etwa in Europa oder Asien. Da solche Institutionen oft mehr als 254 Hosts in ihrem Netz betreiben, wird ihnen häufig ein größerer Block von aufeinanderfolgenden Klasse-C-Netzen zugewiesen.

Die aktuelle Verteilung der IPv4-Adressen können Sie auf der Website der IANA unter [*http://www.iana.org/assignments/ipv4-address-space*](http://www.iana.org/assignments/ipv4-address-space) einsehen.

Als das Konzept der IP-Adressen entstand, konnte niemand auch nur ansatzweise erahnen, welche Dimensionen das Internet einmal annehmen würde. Deshalb glaubten die ursprünglichen Entwickler, dass sie es sich leisten könnten, den Adressraum relativ großzügig aufzuteilen – bedenken Sie etwa, dass die Hälfte des Adressraums für die überaus ineffektiven Klasse-A-Adressen vergeudet wird. Um die drohende Verknappung der IP-Adressen zu verhindern oder zumindest zu verzögern, bis eine Alternative gefunden würde, wurden einige Adressbereiche zur Verwendung in privaten Netzwerken freigegeben, die nicht (oder nicht direkt) mit dem Internet verbunden sind. Es handelt sich um die folgenden Blöcke:

* das Klasse-A-Netz 10.0.0.0
* die 16 Klasse-B-Netze 172.16.0.0 bis 172.31.0.0
* die 256 Klasse-C-Netze 192.168.0.0 bis 192.168.255.0

Ein weiterer Block, der erst später freigegeben wurde, ist das Klasse-B-Netz 169.254.0.0, das einem besonderen Verwendungszweck vorbehalten ist: Moderne TCP/IP-Implementierungen in fast allen Betriebssystemen verwenden dieses Netz für »link local« – eine Möglichkeit, sich automatisch selbst IP-Adressen zuzuweisen, falls wider Erwarten keine Verbindung zu einem DHCP-Server hergestellt werden kann, der eigentlich für die automatische Zuweisung von Adressen zuständig wäre.

Zu guter Letzt existieren noch einige Netze mit anderen speziellen Bedeutungen:

* Die Adresse 0.0.0.0 kann innerhalb eines Netzes verwendet werden, um sich auf das aktuelle Netz selbst zu beziehen.
* Das Klasse-A-Netz 127.0.0.0 beherbergt den sogenannten *Loopback-Bereich*: Über das Loopback-Interface, eine virtuelle Netzwerkschnittstelle mit der Adresse 127.0.0.1, kann ein Host mit sich selbst Netzwerkkommunikation betreiben. Dies ist zum Beispiel nützlich, um während der Programmierung von Client-Server-Anwendungen sowohl das Client- als auch das Serverprogramm auf dem lokalen Host laufen zu lassen.
* Schließlich wird die Adresse 255.255.255.255 als universelle Broadcast-Adresse verwendet: Ein Datenpaket, das an diese Adresse gesendet wird, wird wie beim normalen Broadcast von allen Hosts im Netzwerk empfangen. Nützlich ist diese Einrichtung für Schnittstellen, die ihre IP-Adresse dynamisch beziehen, da sie bei Inbetriebnahme in der Regel noch nicht einmal wissen, in welchem Netz sie sich eigentlich befinden. Auf diese Weise erhalten sie überhaupt erst die Möglichkeit, die Zuteilung einer Adresse anzufordern.

Die Vergabe der privaten Adressbereiche ist in RFC 1918 geregelt; die Festlegung der anderen speziellen Adressbereiche findet sich in RFC 3330.

#### Supernetting, Subnetting und CIDR

In der neueren Entwicklungsgeschichte des Internets hat sich herausgestellt, dass die traditionellen Adressklassen nicht für alle Anwendungsbereiche flexibel genug sind. Deshalb wurde ein neues Schema entwickelt, das die Trennlinie zwischen Netz- und Hostteil der Adressen an einer beliebigen Bit-Grenze ermöglicht. Das in RFC 1519 beschriebene Verfahren heißt Classless Inter-Domain Routing (CIDR).

Die folgenden beiden Anwendungsbeispiele verdeutlichen typische Probleme mit der alten Klassenlogik, die mithilfe von CIDR gelöst werden können:

* Ein Unternehmen besitzt das Klasse-B-Netzwerk 139.17.0.0. Es wäre jedoch wünschenswert, wenn die vier Filialen des Unternehmens jeweils unabhängige Netze betreiben könnten. Dazu soll das vorhandene Netz in vier Teile unterteilt werden – ein Fall für das sogenannte *Subnetting*.
* Ein vor kurzem neu gegründeter europäischer Internetprovider hat die 1.024 Klasse-C-Netze 203.16.0.0 bis 203.19.255.0 erhalten. Das Unternehmen möchte diese Netze als ein großes Netz verwalten, da dies die dynamische Zuteilung an Kunden bei der Einwahl erheblich vereinfacht. Eine solche Zusammenfassung von Netzen wird *Supernetting* genannt.

Das Prinzip von CIDR basiert darauf, dass die traditionellen Byte-Grenzen zwischen Netz- und Hostteil völlig aufgehoben werden. Deshalb ist die Größe des Netzes bei einem CIDR nicht mehr am Beginn der Adresse zu erkennen. Stattdessen wird die Anzahl der Bits, die den Netzwerkteil der Adresse bilden, durch einen Slash getrennt hinter der Netzwerkadresse notiert. Zum Beispiel wird das Klasse-A-Netz 14.0.0.0 zu 14.0.0.0/8.

Eine alternative Darstellungsform für die Grenze zwischen Netz- und Hostteil bei CIDR-Adressen – insbesondere in der IP-Konfiguration der meisten Betriebssysteme – stellt die Teilnetzmaske (Subnet Mask) dar. In dieser Maske werden für die Bits des Netzwerkteils am Anfang der Adresse Einsen notiert, für die Bits des Hostteils am Ende der Adresse dagegen Nullen. Genau wie die IP-Adresse selbst wird auch die Teilnetzmaske in vier dezimalen 8-Bit-Blöcken geschrieben.

Wie bereits erwähnt, wird die Klasseneinteilung bereits seit Mitte der 90er-Jahre nicht mehr verwendet. Deshalb sind alle modernen Netzwerke auf sie angewiesen, da sie allein den Netzwerkteil einer Adresse kennzeichnet. Alle modernen Betriebssysteme verwenden Protokollaufrufe mit Teilnetzmaske. Die Netzwerkadresse wird dabei mithilfe einer Bitweise-Und-Operation aus IP-Adresse und Teilnetzmaske berechnet. Hier ein Beispiel:

IP-Adresse: 192.168.0.37  
Teilnetzmaske: 255.255.255.0  
Berechnete Netzwerkadresse: 192.168.0.0

Hier zum Vergleich die Binärdarstellung:

11000000 10101000 00000000 00100101  
& 11111111 11111111 11111111 00000000  
-------------------------------------  
 11000000 10101000 00000000 00000000

[Tabelle 4.6](https://openbook.rheinwerk-verlag.de/it_handbuch/04_006.html#rxz1004740) zeigt Beispiele für die Schreibweise der ursprünglichen klassenbasierten Adressen nach CIDR-Logik sowie ihre Teilnetzmasken.

|  |  |  |  |
| --- | --- | --- | --- |
| **Tabelle 4.6**Die traditionellen IP-Adressklassen in CIDR-Darstellung | | | |
| **Klasse** | **Beispielnetz** | **CIDR-Adresse** | **Teilnetzmaske** |
| A | 17.0.0.0 | 17.0.0.0/8 | 255.0.0.0 |
| B | 167.18.0.0 | 167.18.0.0/16 | 255.255.0.0 |
| C | 195.21.92.0 | 195.21.92.0/24 | 255.255.255.0 |

Das Subnetting aus dem ersten Beispiel, die Unterteilung des Netzes 139.17.0.0/16 in vier gleich große Teilnetze, kann folgendermaßen durchgeführt werden:

* Da die 65.536 rechnerischen Adressen in vier Teile unterteilt werden sollen, sind zwei weitere Bits für den Netzwerkteil der Adresse erforderlich (4 = 22).
* Da das ursprüngliche Klasse-B-Netz einen 16 Bit (zwei Byte) langen Netzwerkteil besitzt, erfolgt die Unterteilung der vier Adressbereiche nach Bit 18, also nach dem zweiten Bit des dritten Bytes; die vier neuen Netze sind demnach 139.17.0.0/18, 139.17.64.0/18, 139.17.128.0/18 sowie 139.17.192.0/18.

[Tabelle 4.7](https://openbook.rheinwerk-verlag.de/it_handbuch/04_006.html#rxz1004807) zeigt die Eigenschaften der vier neuen Netze.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tabelle 4.7**Subnetting – Unterteilung des Netzes 139.17.0.0/16 in vier gleich große Teilnetze | | | | |
| **Netzwerk** | **1. Hostadresse** | **letzte Hostadresse** | **Broadcast-Adresse** | **Teilnetzmaske** |
| 139.17.0.0/18 | 139.17.0.1 | 139.17.63.254 | 139.17.63.255 | 255.255.192.0 |
| 139.17.64.0/18 | 139.17.64.1 | 139.17.127.254 | 139.17.127.255 | 255.255.192.0 |
| 139.17.128.0/18 | 139.17.128.1 | 139.17.191.254 | 139.17.191.255 | 255.255.192.0 |
| 139.17.192.0/18 | 139.17.192.1 | 139.17.255.254 | 139.17.255.255 | 255.255.192.0 |

Im zweiten Beispiel geht es um Supernetting, das heißt um die Zusammenfassung einzelner Netze zu einem größeren Gesamtnetz. Die Netze 203.16.0.0/24 bis 203.19.255.0/24 sollen zu einem einzigen Netz verbunden werden. Diese Aufgabe lässt sich auf folgende Weise lösen:

* Es werden 1.024 Klasse-C-Netze miteinander verbunden. 256 Netze der Klasse C ergäben einfach ein Gesamtnetz von der Größe eines Klasse-B-Netzwerks. Beispielsweise würde die Vereinigung der Netze 203.16.0.0/24 bis 203.16.255.0/24 das neue Netz 203.16.0.0/16 erzeugen. Um das gewünschte Netz der vierfachen Größe zu erhalten, muss die Grenze zwischen Netz- und Hostteil noch um zwei Bits weiter nach links verschoben werden.
* Die Adresse wird zwei Bits links von der Klasse-B-Grenze, also vor dem vorletzten Bit des zweiten Bytes, unterteilt. Daraus ergibt sich die Netzwerk-Adresse 203.16.0.0/14 mit der Teilnetzmaske 255.252.0.0. Die Broadcast-Adresse des neuen Netzes ist 203.19.255.255; die möglichen Hostadressen reichen von 203.16.0.1 bis 203.19.255.255.

Im Allgemeinen bietet es sich an, die Teilnetzmaske des ursprünglichen Netzes, das aufgeteilt oder mit mehreren verbunden werden soll, zunächst in die Binärdarstellung umzurechnen. In dieser Schreibweise fällt es am leichtesten, die Grenze zwischen Netz- und Hostteil um die gewünschte Anzahl von Bits nach links oder nach rechts zu verschieben. Anschließend können Sie die Maske wieder in die vier üblichen 8-Bit-Gruppen unterteilen und in Dezimalzahlen umrechnen.

Diese Vorgehensweise soll im Folgenden an zwei weiteren Beispielen demonstriert werden. Das Klasse-B-Netzwerk 146.20.0.0/16 soll in acht Teilnetze unterteilt werden:

* Die ursprüngliche Netzmaske ist 255.255.0.0.
* In binärer Darstellung entspricht dies 11111111 11111111 00000000 00000000.
* Eine Aufteilung in acht Netze erfolgt durch eine Verschiebung der Grenze zwischen den beiden Adressteilen um drei Stellen (8 = 23) nach rechts.
* Die neue Netzmaske in binärer Schreibweise ist 11111111 11111111 11100000 00000000.
* Nach der erneuten Umrechnung in die dezimale Vierergruppen-Darstellung ergibt sich 255.255.224.0.
* Entsprechend ergeben sich die folgenden acht Netze:
  + 146.20.0.0/19
  + 146.20.32.0/19
  + 146.20.64.0/19
  + 146.20.96.0/19
  + 146.20.128.0/19
  + 146.20.160.0/19
  + 146.20.192.0/19
  + 146.20.224.0/19

Die vier Klasse-C-Netzwerke 190.16.0.0/24 bis 190.16.3.0/24 sollen zu einem gemeinsamen Netz verbunden werden:

* Die Teilnetzmaske der vier Netze lautet jeweils 255.255.255.0.
* Binär geschrieben ergibt sich daraus 11111111 11111111 11111111 00000000.
* Die Zusammenfassung von vier solchen Netzen erfordert eine Verschiebung der Adressgrenze um zwei Bits (4 = 22) nach links.
* In Binärdarstellung lautet die neue Maske 11111111 11111111 11111100 00000000.
* Wird diese Maske wieder in Dezimalschreibweise umgerechnet, resultiert daraus 255.255.252.0.
* Das neue Netz besitzt die CIDR-Adresse 190.16.0.0/22.

Die folgenden Tabellen zeigen in übersichtlicher Form, wie die Aufteilung der alten IP-Adressklassen in verschiedene Anzahlen von Teilnetzen funktioniert. In [Tabelle 4.8](https://openbook.rheinwerk-verlag.de/it_handbuch/04_006.html#rxz1005092) wird die Klasse A behandelt. Die – rein rechnerisch mögliche – Zusammenfassung mehrerer Klasse-A-Netze durch Supernetting wird in der Praxis nicht durchgeführt, weil erstens wohl niemand mehr als 16,7 Millionen Hosts in einem Teilnetz betreiben möchte und zweitens bereits alle Klasse-A-Netze an einzelne Betreiber vergeben wurden.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tabelle 4.8**Bildung von CIDR-Teilnetzen aus einem Klasse-A-Netz | | | | |
| **Netzwerkbits** | **Hostbits** | **Anzahl Teilnetze** | **Anzahl Hosts** | **Teilnetzmaske** |
| 8 | 24 | 1 | 16.777.214 | 255.0.0.0 |
| 9 | 23 | 2 | 8.388.606 | 255.128.0.0 |
| 10 | 22 | 4 | 4.194.302 | 255.192.0.0 |
| 11 | 21 | 8 | 2.097.150 | 255.224.0.0 |
| 12 | 20 | 16 | 1.048.574 | 255.240.0.0 |
| 13 | 19 | 32 | 524.286 | 255.248.0.0 |
| 14 | 18 | 64 | 262.142 | 255.252.0.0 |
| 15 | 17 | 128 | 131.070 | 255.254.0.0 |
| 16 | 16 | 256 | 65.534 | 255.255.0.0 |
| 17 | 15 | 512 | 32.766 | 255.255.128.0 |
| 18 | 14 | 1.024 | 16.382 | 255.255.192.0 |
| 19 | 13 | 2.048 | 8.190 | 255.255.224.0 |
| 20 | 12 | 4.096 | 4.094 | 255.255.240.0 |
| 21 | 11 | 8.192 | 2.046 | 255.255.248.0 |
| 22 | 10 | 16.384 | 1.022 | 255.255.252.0 |
| 23 | 9 | 32.768 | 510 | 255.255.254.0 |
| 24 | 8 | 65.536 | 254 | 255.255.255.0 |
| 25 | 7 | 131.072 | 126 | 255.255.255.128 |
| 26 | 6 | 262.144 | 62 | 255.255.255.192 |
| 27 | 5 | 524.288 | 30 | 255.255.255.224 |
| 28 | 4 | 1.048.576 | 14 | 255.255.255.240 |
| 29 | 3 | 2.097.152 | 6 | 255.255.255.248 |
| 30 | 2 | 4.194.302 | 2 | 255.255.255.252 |

[Tabelle 4.9](https://openbook.rheinwerk-verlag.de/it_handbuch/04_006.html#rxz1005266) zeigt die Aufteilung eines Klasse-B-Netzes in beliebig kleine Teilnetze.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tabelle 4.9**Bildung von CIDR-Teilnetzen aus einem Klasse-B-Netz | | | | |
| **Netzwerkbits** | **Hostbits** | **Anzahl Teilnetze** | **Anzahl Hosts** | **Teilnetzmaske** |
| 16 | 16 | 1 | 65.534 | 255.255.0.0 |
| 17 | 15 | 2 | 32.766 | 255.255.128.0 |
| 18 | 14 | 4 | 16.382 | 255.255.192.0 |
| 19 | 13 | 8 | 8.190 | 255.255.224.0 |
| 20 | 12 | 16 | 4.094 | 255.255.240.0 |
| 21 | 11 | 32 | 2.046 | 255.255.248.0 |
| 22 | 10 | 64 | 1.022 | 255.255.252.0 |
| 23 | 9 | 128 | 510 | 255.255.254.0 |
| 24 | 8 | 256 | 254 | 255.255.255.0 |
| 25 | 7 | 512 | 126 | 255.255.255.128 |
| 26 | 6 | 1.024 | 62 | 255.255.255.192 |
| 27 | 5 | 2.048 | 30 | 255.255.255.224 |
| 28 | 4 | 4.096 | 14 | 255.255.255.240 |
| 29 | 3 | 8.192 | 6 | 255.255.255.248 |
| 30 | 2 | 16.384 | 2 | 255.255.255.252 |

[Tabelle 4.10](https://openbook.rheinwerk-verlag.de/it_handbuch/04_006.html#rxz1005360) demonstriert schließlich, wie die Unterteilung eines Klasse-C-Netzes erfolgt. In kleineren Unternehmen könnte es durchaus praktisch sein, ein solches – ohnehin kleines – Netzwerk weiter zu unterteilen.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tabelle 4.10**Bildung von CIDR-Teilnetzen aus einem Klasse-C-Netz | | | | |
| **Netzwerkbits** | **Hostbits** | **Anzahl Teilnetze** | **Anzahl Hosts** | **Teilnetzmaske** |
| 24 | 8 | 1 | 254 | 255.255.255.0 |
| 25 | 7 | 2 | 126 | 255.255.255.128 |
| 26 | 6 | 4 | 62 | 255.255.255.192 |
| 27 | 5 | 8 | 30 | 255.255.255.224 |
| 28 | 4 | 16 | 14 | 255.255.255.240 |
| 29 | 3 | 32 | 6 | 255.255.255.248 |
| 30 | 2 | 64 | 2 | 255.255.255.252 |

In der Praxis ermöglicht CIDR bereits einen erheblich flexibleren Netzwerkaufbau als die Verwendung der alten Klassen. Doch auch diese Verfahrensweise kann immer noch ungünstige Ergebnisse zur Folge haben, wenn Teilnetze mit erheblich unterschiedlichen Größen benötigt werden: Das größte benötigte Teilnetz bestimmt die Größe aller anderen; selbst das kleinste belegt eine Menge von Adressen, die es womöglich niemals benötigen wird.

Aus diesem Grund wurde das VLSM-Konzept (Variable Length Subnet Mask) eingeführt. Es handelt sich um ein spezielles Subnetting-Verfahren, bei dem ein gegebenes Netz nicht mehr in gleich große, sondern in verschieden große Teilnetze unterteilt wird. Jedem dieser Teilnetze wird eine individuelle Teilnetzmaske zugewiesen.

Das grundlegende Prinzip von VLSM besteht darin, vom kleinsten benötigten Teilnetz auszugehen und die entsprechenden größeren Netze aus Blöcken solcher kleinsten Teilnetze zu bilden, denen dann höhere Teilnetzmasken zugewiesen werden. Angenommen etwa, bei der Aufteilung eines Klasse-B-Netzes mit seinen 65.534 Hostadressen besitzt das kleinste gewünschte Teilnetz 12 Hosts, das größte etwa 500. Für die 12 Hosts ist mindestens ein Netz mit der Teilnetzmaske 255.255.255.240 erforderlich, das 14 Hostadressen bietet. Aus diesen kleinen Teilnetzen können dann entsprechend größere aufgebaut werden, wobei die Grenzen zwischen den Netzen der Logik der jeweiligen Netzmaske entsprechen müssen.

An dieser Stelle soll ein einfaches Beispiel genügen: Ein Unternehmen betreibt das öffentliche Klasse-C-Netz 196.17.41.0/24. Dieses Netz soll auf die drei Abteilungen der Firma aufgeteilt werden; die beiden Router und die drei Server sollen ein viertes separates Teilnetz bilden. [Tabelle 4.11](https://openbook.rheinwerk-verlag.de/it_handbuch/04_006.html#rxz1005430) zeigt die klassische Aufteilung des Netzes in vier gleich große Teile nach CIDR-Logik.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tabelle 4.11**Aufteilung des Netzes 196.17.41.0/24 in vier Teile nach dem CIDR-Schema | | | | |
| **Bereich** | **Anzahl Hosts** | **Teilnetz** | **maximale Hosts** | **freie Adressen** |
| Server/Router | 5 | 196.17.41.0/26 | 62 | 57 |
| Verwaltung | 20 | 196.17.41.64/26 | 62 | 42 |
| Programmierung | 61 | 196.17.41.128/26 | 62 | 1 |
| Design | 30 | 196.17.41.192/26 | 62 | 32 |

Es ist leicht zu erkennen, dass zwei der Teilnetze – Server/Router und Verwaltung – vollkommen überdimensioniert sind, während zumindest das Teilnetz der Programmierabteilung beinahe seine Belastungsgrenze erreicht hat. Stellen Sie sich vor, es werden noch zwei weitere Hosts in diese Abteilung aufgenommen: Schon wäre das Teilnetz zu klein, und es müsste über eine andere Verteilung nachgedacht werden. In diesem Beispiel könnte sie nur noch darin bestehen, zwei der anderen Bereiche zusammenzulegen, um den Programmierbereich zu vergrößern.

Eine komplexere, aber für den konkreten Anwendungsfall sinnvollere Aufteilung des Netzes mithilfe der VLSM-Technik zeigt [Tabelle 4.12](https://openbook.rheinwerk-verlag.de/it_handbuch/04_006.html#rxz1005498).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tabelle 4.12**Flexible Aufteilung des Netzes 196.17.41.0/24 in vier Teile nach dem VLSM-Schema | | | | |
| **Bereich** | **Anzahl Hosts** | **Teilnetz** | **maximale Hosts** | **freie Adressen** |
| Server/Router | 5 | 196.17.41.0/27 | 30 | 25 |
| Verwaltung | 20 | 196.17.41.32/27 | 30 | 10 |
| Design | 30 | 196.17.41.64/26 | 62 | 32 |
| Programmierung | 61 | 196.17.41.128/25 | 126 | 65 |

Für die IP-Konfiguration eines einzelnen Hosts macht es keinen Unterschied, ob das Teilnetz, in dem er sich befindet, nach der alten Klassenlogik, nach dem CIDR-Verfahren oder nach der VLSM-Methode konfiguriert wurde: In jedem Fall wird im Konfigurationsdialog des jeweiligen Betriebssystems die korrekte Teilnetzmaske eingestellt. Spezielle Unterstützung für VLSM benötigen lediglich Router, die in dem betroffenen Netz eingesetzt werden. Die meisten neueren Routing-Protokolle bieten diese Unterstützung.

#### Die Übertragung von IP-Datagrammen

Auf der Internetschicht des TCP/IP-Protokollstapels, auf der das IP-Protokoll arbeitet, werden die Datenpakete, wie bereits erwähnt, als *Datagramme* bezeichnet. Um die Datenübertragung mithilfe des IP-Protokolls genau zu erläutern, soll an dieser Stelle zunächst der IP-Header vorgestellt werden. Er enthält die Steuerdaten, die das IP-Protokoll zu einem Datenpaket hinzufügt, das ihm vom übergeordneten Transportprotokoll übergeben wird.

Der IPv4-Protokoll-Header wird wie das gesamte Protokoll in RFC 791 definiert. Seine Länge beträgt mindestens 20 Byte, dazu können bis zu 40 Byte Optionen kommen. [Tabelle 4.13](https://openbook.rheinwerk-verlag.de/it_handbuch/04_006.html#rxz1005629) zeigt den genauen Aufbau.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tabelle 4.13**Aufbau des IPv4-Datagramm-Headers | | | | | | |
| **Byte** | **0** | | **1** | **2** | | **3** |
| **0** | Version | IHL | Type of Service | Paket-Gesamtlänge | | |
| **4** | Identifikation | | | Flags | Fragment-Offset | |
| **8** | Time to Live | | Protokoll | Header-Prüfsumme | | |
| **12** | Quell-Adresse | | | | | |
| **16** | Ziel-Adresse | | | | | |
| **20** | Optionen | | | | | Padding |
| **...** | eventuell weitere Optionen | | | | | |

Die einzelnen Daten des IP-Headers sind folgende:

* Version (4 Bit): Die Versionsnummer des IP-Protokolls, die das Paket verwendet. Bei IPv4, wie der Name schon sagt, die Version 4.
* IHL (4 Bit): Internet Header Length; die Länge des Internet-Headers in 32-Bit-Wörtern (entsprechen den Zeilen in [Tabelle 4.13](https://openbook.rheinwerk-verlag.de/it_handbuch/04_006.html#rxz1005629)). Der kleinste mögliche Wert beträgt 5.
* Type of Service (8 Bit): Ein Code, der die Art des Datenpakets bestimmt. Bestimmte Sorten von Paketen, etwa für den Austausch von Routing- oder Status-Informationen, werden von bestimmten Netzen bevorzugt weitergeleitet. In ihrem 1999er-Aprilscherz bot die Computerzeitschrift c’t ein angebliches Tool zum Download an, das diese Quality-of-Service-Informationen manipulieren könne, um die Geschwindigkeit von Internetverbindungen zu erhöhen.[Anm.: Die Satire war immerhin so überzeugend gemacht, dass ein Leser per empörten Leserbrief sein Abo kündigte, weil er mit derart »unmoralischem Verhalten« im Netz nichts zu tun haben wollte.]
* Paket-Gesamtlänge (16 Bit): Die Gesamtlänge des Datagramms in Bytes, Header und Nutzdaten.
* Identifikation (16 Bit): Ein durch den Absender frei definierbarer Identifikationswert, der beispielsweise das Zusammensetzen fragmentierter Datagramme ermöglicht.
* Flags (3 Bit): Kontrollflags, die die Paketfragmentierung regeln. Das erste Bit ist reserviert und muss immer 0 sein, das zweite (DF) bestimmt, ob das Paket fragmentiert werden darf (Wert 1) oder nicht (0), das dritte (MF) regelt, ob dieses Paket das letzte Fragment (0) ist oder ob weitere Fragmente folgen (1).
* Fragment-Offset (13 Bit): Dieser Wert (angegeben in 64-Bit-Blöcken) legt fest, an welcher Stelle in einem Gesamtpaket dieses Paket steht, falls es sich um ein Fragment handelt. Das erste Fragment oder ein nicht fragmentiertes Paket erhält den Wert 0.
* Time to Live (8 Bit): Der TTL-Mechanismus sorgt dafür, dass Datagramme nicht endlos im Internet weitergeleitet werden, falls die Empfängerstation nicht gefunden wird. Jeder Router, der ein Datagramm weiterleitet, zieht von diesem Wert 1 ab; wird der Wert 0 erreicht, leitet der betreffende Router das Paket nicht mehr weiter, sondern verwirft es.
* Protokoll (8 Bit): Die hier gespeicherte Nummer legt fest, für welches Transportprotokoll der Inhalt des Datagramms bestimmt ist, zum Beispiel 6 für TCP oder 17 für UDP. Diese beiden wichtigsten Transportprotokolle werden im nächsten Abschnitt beschrieben.
* Header-Prüfsumme (16 Bit): Die Prüfsumme stellt eine einfache Plausibilitätskontrolle für den Datagramm-Header zur Verfügung. Ein Paket, dessen Header-Prüfsumme nicht korrekt ist, wird nicht akzeptiert und muss erneut versendet werden.
* Quelladresse und Zieladresse (je 32 Bit): Die IP-Adressen von Absender und Empfänger. IP-Adressen wurden zuvor bereits ausführlich behandelt.
* Optionen (variable Länge): Die meisten IP-Datagramme werden ohne zusätzliche Optionen versandt, da Absender- und Empfängerhost sowie alle auf dem Weg befindlichen Router die jeweils verwendeten Optionen unterstützen müssen. Zu den verfügbaren Optionen gehören unter anderem Sicherheitsfeatures und spezielle Streaming-Funktionen.

Das Problem der Paket-Fragmentierung entsteht dadurch, dass verschiedene physikalische Netzarten unterschiedliche Maximallängen für Datenpakete erlauben. Dieser Wert, der als *Maximum Transmission Unit* (MTU) bezeichnet wird, kann bei einigen Netzwerkschnittstellen per Software konfiguriert werden, bei anderen ist er vom Hersteller vorgegeben. Werden nun Datagramme aus einem Netz mit einer bestimmten MTU in ein anderes Netz mit einer kleineren MTU weitergeleitet, dann müssen die Daten in kleinere Pakete »umgepackt« werden. Wie bereits beschrieben, werden sie dazu mit Fragmentierungsinformationen versehen, damit sie später wieder richtig zusammengesetzt werden können.

Solange Quell- und Zieladresse im gleichen Netzwerk liegen, ist die Übertragung der Datagramme sehr einfach: Je nach Netzwerkart wird auf die passende Art (bei Ethernet zum Beispiel über ARP) diejenige Schnittstelle ermittelt, für die die Daten bestimmt sind. Anschließend wird das Datagramm an den korrekten Empfänger übermittelt. Dieser liest den IP-Header des Pakets, setzt eventuelle Fragmente wieder richtig zusammen und übermittelt das Paket an das Transportprotokoll, dessen Nummer im Header angegeben ist. Wie der Transportdienst mit den Daten umgeht, erfahren Sie im nächsten Abschnitt.

#### IPv6

Bereits in der ersten Hälfte der 90er-Jahre wurde damit gerechnet, dass sehr bald keine weiteren IPv4-Adressen mehr verfügbar sein würden. Dass dies viel länger als gedacht noch nicht der Fall war, lag an der Einführung von CIDR, VLSM und NAT (Letzteres wird im übernächsten Unterabschnitt, »Weitere IP-Dienste«, beschrieben). Da das Internet aber weiterhin wächst, ist es nur noch eine Frage der Zeit, bis die Anzahl der Adressen endgültig erschöpft ist; was die Zuteilung an Dienstleister angeht, ist dies sogar bereits der Fall.

Deshalb wurde schon vor einigen Jahren mit der Arbeit an einem Nachfolger für das IPv4-Protokoll begonnen, der vor allem einen größeren Adressraum durch längere IP-Adressen besitzen sollte. Letzten Endes entschieden die Entwickler sich für Adressen von 128 Bit Länge. Dies ergibt theoretisch mehr als 3,4 × 1038 verschiedene Adressen! Damit erscheint der Adressraum mehr als überdimensioniert; offensichtlich kann man damit jedem einzelnen Sandkorn auf unserem Planeten mehrere eigene IP-Adressen zuweisen. Letzten Endes geht es allerdings eher darum, beinahe beliebig viele Netze von sehr unterschiedlicher Größe einrichten zu können, und abgesehen davon werden immer mehr tragbare Geräte entwickelt, die mit Netzwerken verbunden werden – etwa dynamisch über öffentliche WLAN-Access-Points.

Die aktuelle Version des neuen IP-Protokolls wird in RFC 2460 beschrieben. Da die Version 5 für Experimente mit Multicasting verwendet wurde, lautet die Versionsnummer des Protokolls IPv6; während seiner Entwicklung wurde es auch manchmal als *IPng* (für »next generation«) bezeichnet, zum Beispiel in RFC 1752, das den ersten Arbeitsentwurf beschreibt. Die IPv6-Adresse wird nicht in 8-Bit-Dezimalgruppen geschrieben wie bei IPv4; mit 16 Gruppen wäre sie ein wenig unhandlich. Stattdessen schreibt man 8 vierstellige Hexadezimalgruppen, die durch Doppelpunkte getrennt werden. Eine IPv6-Adresse sieht zum Beispiel folgendermaßen aus:

4A29:30B4:0031:0000:0000:0092:1A3B:3394

Eine zulässige Verkürzung besteht darin, führende Nullen in einem Block wegzulassen sowie Blöcke, die nur aus Nullen bestehen, durch zwei aufeinanderfolgende Doppelpunkte zu ersetzen. Kurz gefasst lautet die Beispieladresse also 4A29:30B4:31::92:1A3B:3394. Um die Adresse eindeutig zu halten, darf eine solche Verkürzung innerhalb einer Adresse nur einmal durchgeführt werden.

Genau wie IPv4-Adressen werden auch die neuen IPv6-Adressen in zwei Teile unterteilt: Links steht ein Präfix, dahinter ein Individualteil, der dem Hostteil der IPv4-Adresse entspricht. Das Präfix gibt allerdings nicht das einzelne Netz an, zu dem die Adresse gehört, sondern informiert über den Adresstyp. Da die Präfixe wie bei IPv4 unterschiedliche Längen aufweisen können, wird das Präfix zusammen mit seiner Bit-Anzahl angegeben. [Tabelle 4.14](https://openbook.rheinwerk-verlag.de/it_handbuch/04_006.html#rxz1005768) gibt einen Überblick über die verschiedenen Adressblöcke und ihre Verwendung.

|  |  |
| --- | --- |
| **Tabelle 4.14**IPv6-Adressbereiche und -Präfixe | |
| **Präfix** | **Verwendung** |
| 0::0/8 | Reserviert für spezielle Anwendungen. |
| 100::0/8 | noch nicht zugeordnet |
| 200::0/7 | Abbildung von NSAP-Adressen |
| 400::0/7 | Abbildung von IPX-Adressen |
| 600::0/7 | noch nicht zugeordnet |
| 800::0/5 | noch nicht zugeordnet |
| 1000::0/4 | noch nicht zugeordnet |
| 2000::0/3 | global eindeutige Adressen |
| 6000::0/3 | noch nicht zugeordnet |
| 8000::0/3 | noch nicht zugeordnet |
| A000::0/3 | noch nicht zugeordnet |
| C000::0/3 | noch nicht zugeordnet |
| E000::0/4 | noch nicht zugeordnet |
| F000::0/5 | noch nicht zugeordnet |
| F800::0/6 | noch nicht zugeordnet |
| FE00::0/7 | noch nicht zugeordnet |
| FE00::0/9 | noch nicht zugeordnet |
| FE80::0/10 | auf eine Verbindung begrenzte Adressen |
| FEC0::0/10 | auf eine Einrichtung begrenzte Adressen |
| FF00::0/8 | Multicast-Adressen |

Die typischste Form von IPv6-Adressen, deren Stil am ehesten den öffentlich gerouteten IPv4-Adressen entspricht, ist die globale Unicast-Adresse. Ihre Struktur ist in RFC 2374 festgelegt und sieht folgendermaßen aus:

* externes Routing-Präfix (48 Bit)
* Site-Topologie (üblicherweise 16 Bit)
* Schnittstellen-Identifikationsnummer (normalerweise 64 Bit); wird in der Regel automatisch generiert, oft aus der MAC-Adresse der Schnittstelle oder aus der bisherigen IPv4-Adresse

Der IPv6-Datagramm-Header wurde gegenüber dem IPv4-Header erheblich vereinfacht. Durch die Auslagerung eventueller Optionen in sogenannte *Erweiterungs-Header* wird die Länge des Basis-Headers auf genau 320 Bit (40 Byte) festgelegt; einige Felder des IPv4-Headers wurden entfernt, weil sie keine Bedeutung mehr haben. [Tabelle 4.15](https://openbook.rheinwerk-verlag.de/it_handbuch/04_006.html#rxz1005857) zeigt den genauen Aufbau des IPv6-Headers.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tabelle 4.15**Aufbau des IPv6-Datagramm-Headers | | | | | | |
| **Byte** | **0** | | **1** | | **2** | **3** |
| **0** | Version | Klasse | | Flow Label | | |
| **4** | Payload Length | | | | Next Header | Hop Limit |
| **8 12 16 20** | Quelladresse | | | | | |
| **24 28 32 36** | Zieladresse | | | | | |

Hier die Bedeutung der einzelnen Felder des Headers:

* Version (4 Bit): Die Versionsnummer des IP-Protokolls; hier natürlich 6.
* Klasse (8 Bit): Dieses Feld gibt die Priorität an, mit der das Datagramm übertragen werden soll. Es ist noch nicht abschließend geklärt, wie die entsprechenden Werte aussehen sollen.
* Flow Label (20 Bit): Ein Erweiterungsfeld, in das ein von 0 verschiedener Wert eingetragen wird, wenn IPv6-Router das Datagramm auf besondere Weise behandeln sollen. Es dient vor allem der Implementierung der Quality-of-Service-Funktionalität, mit deren Hilfe Paketsorten voneinander unterschieden werden, um beispielsweise Echtzeitanwendungen wie Streaming Multimedia oder Videokonferenzen zu unterstützen.
* Payload Length (16 Bit): Die Länge der Nutzdaten, die auf den Header folgen.
* Next Header (8 Bit): Der Wert in diesem Feld gibt den Typ des ersten Erweiterungs-Headers an, falls einer vorhanden ist. Es gibt bisher sechs Arten von Erweiterungs-Headern; eine Übersicht finden Sie in [Tabelle 4.16](https://openbook.rheinwerk-verlag.de/it_handbuch/04_006.html#rxz1005933).
* Hop Limit (8 Bit): Ein neuer Name für die Time-to-Live-Funktionalität: Jeder Router zieht von dem ursprünglichen Wert 1 ab; bei Erreichen des Wertes 0 wird das Paket verworfen.
* Quell- und Zieladresse (je 128 Bit): Die Adresse des Absenders und des Empfängers; genau wie bei IPv4, nur entsprechend der Protokollspezifikation 128 statt 32 Bit lang.

|  |  |  |
| --- | --- | --- |
| **Tabelle 4.16**Die verschiedenen Typen von IPv6-Erweiterungs-Headern | | |
| **Header** | **Next-Header-Code** | **Beschreibung** |
| Hop-by-Hop Options Header | 0 | Optionen, die bei jedem Routing-Schritt ausgeführt werden müssen |
| Routing Header | 43 | Festlegung der Router, über die das Paket geleitet werden soll |
| Fragment Header | 44 | Der Absender muss bei IPv6 die MTU herausfinden und Fragmente selbst bilden; die Fragment-Informationen befinden sich hier. |
| Authentication Header | 51 | Authentifizierung des Absenders gegenüber dem Empfänger |
| Encapsulating Security Payload Header | 50 | Dient der Verschlüsselung des Datagramms (IPv6) |
| Destination Options Header | 60 | Optionen, die nur für den Zielhost bestimmt sind |
| Upper-Layer Header | 59 | Header einer höheren Schicht; aus IPv6-Sicht also Nutzdaten |

Das größte Problem, das der sofortigen Einführung von IPv6 noch im Wege steht, ist ein organisatorisches: Zum einen kann man nicht einfach über Nacht flächendeckend umsteigen, da in diesem Fall die IP-Treiber aller Hosts und Router weltweit gewechselt werden müssten, was vollkommen illusorisch ist – zumal viele ältere Hardwarekomponenten, Betriebssysteme und Programme IPv6 gar nicht unterstützen und ihre Hersteller auch nicht vorhaben, diese Unterstützung nachträglich zu implementieren. Zum anderen ist es aber auch nicht möglich, gleichzeitig einen Teil des Internets mit IPv4 und einen anderen mit IPv6 zu betreiben und auf diese Weise allmählich auf die neue Version umzusteigen, da die beiden Adressierungsschemata miteinander inkompatibel sind.

Die Lösung, die letzten Endes gefunden wurde, besteht in der Tunnelung von IPv6-Paketen durch das klassische IPv4-Netzwerk. Tunnelung bedeutet letzten Endes nichts anderes, als dass jedes IPv6-Datagramm in ein IPv4-Datagramm verpackt wird. Das heißt, das IPv6-Paket bildet aus der Sicht des IPv4-Pakets die Nutzdaten, die mit einem v4-Header versehen werden. Am jeweiligen Zielpunkt, an dem wiederum IPv6 verfügbar ist, wird das Version-4-Datagramm »ausgepackt« und den Header-Daten entsprechend weiterverarbeitet. IPv6-Tunnel-Dienste werden mittlerweile auch von mehreren kommerziellen und verschiedenen freien Anbietern, den Tunnel-Brokern, zur Verfügung gestellt.

#### IP-Routing

Komplizierter, aber auch interessanter wird es, wenn IP-Datagramme nicht für einen Host im lokalen Netz bestimmt sind, sondern für ein anderes Netzwerk. In diesem Fall muss das Paket an einen Router übergeben werden, der es weiterleitet. Die meisten Daten, die im Internet übertragen werden, passieren eine Vielzahl solcher Router, bis sie schließlich ihr Ziel erreichen. Um das Konzept des IP-Routings verstehen zu können, müssen Sie verschiedene Aspekte betrachten. Insbesondere ist die Frage von Bedeutung, auf welche Art und Weise überhaupt das korrekte Empfängernetzwerk gefunden wird.

Wichtig ist, dass man zwei Arten der Paketweiterleitung unterscheiden muss. Die reine Weiterleitung wird als *IP-Forwarding* bezeichnet; dabei sind nur zwei mögliche Netzwerkschnittstellen betroffen, sodass Quelle und Ziel jeweils feststehen. *Routing* im engeren Sinne beschreibt dagegen Verfahren, bei denen Entscheidungen zur Weiterleitung über verschiedene Wege an ein bestimmtes Ziel getroffen werden. Ein Router muss beide Verfahren beherrschen, sodass im Alltag oft nicht zwischen ihnen unterschieden wird. In LANs findet jedoch oft nur Forwarding, aber kein echtes Routing statt, da meist ohnehin nicht mehrere Router zur Auswahl stehen. Sowohl Forwarding als auch Routing lässt sich übrigens entweder statisch über festgelegte Tabellen oder dynamisch mithilfe von Protokollen erledigen.

Bei einem einzelnen Host können üblicherweise zwei Arten von Routern angegeben werden: zum einen die Router, die Daten in ein bestimmtes Fremdnetzwerk weiterleiten, und zum anderen das Default-Gateway (der Standard-Router), das alle Daten entgegennimmt, die weder für das lokale Netz noch für ein Netz mit einem speziellen Router bestimmt sind.

Beachten Sie übrigens, dass der Begriff *Gateway* zweideutig ist: Das Wort *Default-Gateway* beim IP-Forwarding oder Routing bezeichnet wie erwähnt den Standard-Router. Im Allgemeinen steht Gateway dagegen für einen Verbindungsrechner, der über sämtliche OSI-Schichten arbeitet und deshalb genauer als *Application Level Gateway* bezeichnet wird.

Bei einem privaten PC oder DSL-Router, der über eine Wählleitung mit dem Internet verbunden ist, besteht in der Regel nur eine Verbindung zu einem einzelnen Router des Providers. Welcher das ist, wird jedoch bei der Einwahl in das Netzwerk des Providers bestimmt, da auch die IP-Adresse bei jeder Einwahl dynamisch zugeteilt wird. Je nachdem, welche Adresse dem Host zugeteilt wird, ist möglicherweise ein anderer Router zuständig. Deshalb wird der Router bei der IP-Konfiguration des DFÜ-Netzwerkzugangs nicht fest angegeben, sondern durch das Einwahlprotokoll (üblicherweise PPP) mitgeteilt.

Anders sieht es dagegen oft bei Workstations in Unternehmen aus, die an ein lokales Netzwerk angeschlossen sind: Sämtliche Netzwerkkommunikation, sowohl mit dem lokalen Netz als auch mit dem Internet, findet über ein und dieselbe LAN-Schnittstelle statt, meistens über Ethernet. Innerhalb des LAN besitzt der Router für die Verknüpfung zum Internet eine bekannte IP-Adresse, die bei der IP-Konfiguration des Hosts angegeben wird. Mitunter besteht die Netzwerkinfrastruktur eines größeren Unternehmens auch aus mehreren Einzelnetzen, die über interne Router miteinander vernetzt werden. In einem solchen Fall wird häufig der Router, der zu dem anderen lokalen Netz führt, als Router für dieses konkrete Netz angegeben, während der Internetrouter (dessen Zielnetz »alle anderen Netze« sind) als Standard-Router eingerichtet wird. Für den letzteren – routingtechnisch relativ interessanten – Fall sehen Sie hier ein Beispiel:

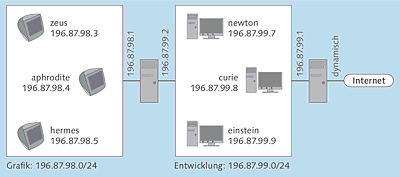
In einem Unternehmen bestehen die beiden lokalen Netze 196.87.98.0/24 und 196.87.99.0/24. Das erste Netz wird von der Grafikabteilung verwendet, das zweite von den Softwareentwicklern. In [Abbildung 4.4](https://openbook.rheinwerk-verlag.de/it_handbuch/04_006.html#rxz1119731) wird der Aufbau dieses Netzes dargestellt.

Das Netzwerk der Grafikabteilung enthält die folgenden drei Rechner:

* zeus (196.87.98.3)
* aphrodite (196.87.98.4)
* hermes (196.87.98.5)

Zum Netzwerk der Entwicklungsabteilung gehören die drei folgenden Hosts:

* newton (196.87.99.7)
* curie (196.87.99.8)
* einstein (196.87.99.9)

[](https://openbook.rheinwerk-verlag.de/it_handbuch/bilder/04_004.png)

**Abbildung 4.4**Verbindung zwischen zwei verschiedenen lokalen Netzen und dem Internet über zwei Router

Zwischen den beiden lokalen Netzen befindet sich ein Router, dessen Schnittstelle im Netz der Grafikabteilung die IP-Adresse 196.87.98.1 besitzt. Seiner anderen Schnittstelle für die Entwicklungsabteilung wurde die Adresse 196.87.99.2 zugewiesen. Ein zweiter Router verbindet die Entwicklungsabteilung mit dem Internet. Seine lokale Schnittstelle wurde mit der IP-Adresse 196.87.99.1 konfiguriert; die Adresse für die Internetschnittstelle wird vom Internetprovider dynamisch zugewiesen.

Interessant ist nun die Routing-Konfiguration der einzelnen Hosts. Die drei Rechner im Entwickler-Netzwerk kennen zwei verschiedene Router: Der Standard-Router ist 196.87.99.1, als spezieller Router für Datenpakete an das Netz 196.87.98.0 wird 196.87.99.2 angegeben. Dagegen kennen die drei Hosts im Grafik-Netzwerk nur einen einzigen Router, nämlich 196.87.98.1, der als Standard-Router eingerichtet wird. Ob Datenpakete jenseits dieses Routers für das Netz 196.87.99.0 oder für das Internet bestimmt sind, muss der Router selbst entscheiden; die Rechner schicken ihm einfach alle Datagramme, die nicht für das lokale Netz bestimmt sind.

Angenommen, »aphrodite« möchte auf Daten zugreifen, die »newton« bereitstellt. Die Daten sind offensichtlich nicht für das Netz 196.87.98.0 bestimmt, deshalb werden sie dem Router übergeben. Dieser erkennt, dass sie für das Netz 196.87.99.0 bestimmt sind, an das er unmittelbar angeschlossen ist. Er kann die Daten direkt an den Zielhost ausliefern.

Will dagegen »zeus« auf Daten aus dem Internet zugreifen, muss der Standard-Router des Grafik-Netzes erkennen, dass die Daten nicht für das andere Netz bestimmt sind, an das er selbst angeschlossen ist, und sie an den nächsten Router weiterreichen.

Ein wenig anders verhält es sich, wenn ein Rechner aus dem Entwickler-Netz wie »curie« auf »zeus« zugreifen möchte. Es ist bereits in der Routing-Konfiguration von »curie« bekannt, dass ein bestimmter Router, nämlich 196.87.99.2, zu verwenden ist. Ebenso weiß beispielsweise »einstein«, dass Zugriffe auf das Internet über den Router 196.87.99.1 erfolgen müssen.

Damit ein Host weiß, wohin er Datenpakete eigentlich schicken muss, um ein bestimmtes Netz zu erreichen, müssen die einzelnen Router in seiner Netzwerkkonfiguration angegeben werden – dies funktioniert je nach Betriebssystem unterschiedlich. Das Ergebnis dieser Konfiguration ist eine Routing-Tabelle, die ebenfalls je nach System unterschiedlich aussieht. Angenommen, alle Rechner im zuvor gezeigten Beispielnetzwerk liefen unter Unix-Varianten (die Grafik-Rechner unter Mac OS X, die Entwickler-Computer unter Linux). Dann sähe die Routing-Tabelle von »curie«, die durch den Unix-Befehl netstat -rn angezeigt werden kann[Anm.: Näheres über die einfachen TCP/IP-Dienstprogramme erfahren Sie für die jeweilige Systemplattform in [Kapitel 6](https://openbook.rheinwerk-verlag.de/it_handbuch/06_001.html#rxz1008115), »Windows«, beziehungsweise [Kapitel 7](https://openbook.rheinwerk-verlag.de/it_handbuch/07_001.html#rxz1009500), »Linux«.], so aus:

$ **netstat** **-rn**  
Routing Tables  
Destination Gateway FlagsRefcntUseInterface  
127.0.0.1 127.0.0.1 UH 1 132lo0  
196.87.99.0 196.87.99.8 U2649041le0  
196.87.98.0 196.87.99.2 UG 0 0le0  
default 196.87.99.1 UG 0 0le0

Die erste Zeile (Zieladresse 127.0.0.1) beschreibt das Erreichen der Loopback-Adresse: Das Interface (Netzwerkschnittstelle) ist lo0 (»local loopback«). Das Flag H zeigt an, dass es sich um eine Route zum Erreichen eines einzelnen Hosts handelt. Das Flag U dagegen steht für »up« und bedeutet, dass die Route zurzeit intakt ist.

In der nächsten Zeile wird das lokale Netzwerk angegeben, in dem sich »curie« selbst befindet. Deshalb wird als Gateway einfach die IP-Adresse von »curie« ausgegeben. Das Interface le0 ist die erste (und in diesem Fall einzige) Ethernet-Schnittstelle des Rechners.

Die dritte Zeile beschreibt die Route in das Grafik-Netzwerk über den Router, dessen Adresse im Entwickler-Netz 196.87.99.2 lautet. Das Flag G steht für »Gateway«, also für die Tatsache, dass für diese Route die Dienste eines Routers in Anspruch genommen werden.

In der letzten Zeile wird schließlich 196.87.99.1 als Default-Gateway angegeben, das heißt als Router für alle Ziele, die nicht explizit in der Routing-Tabelle auftauchen.

Die Routing-Tabelle von »hermes« sieht einfacher aus:

Routing Tables  
Destination GatewayFlagsRefcnt UseInterface  
127.0.0.1 127.0.0.1UH 1 132lo0  
196.87.98.0 196.87.98.5U2649041le0  
default 196.87.98.1UG 0 0le0

Da das Grafik-Netz nur einen Router kennt, gibt es nur den Loopback-Eintrag, die Information für das lokale Netz und schließlich den Default-Eintrag für alle anderen Netze.

Auf diese Weise werden Daten durch das gesamte Internet geroutet. Jedes Mal, wenn ein Router passiert wird, erfolgt ein sogenannter *Hop* der Daten. Wegen des TTL-Felds von 8 Bit Größe, das im IP-Header enthalten ist und bereits beschrieben wurde, erreicht ein Datagramm sein Ziel stets mit höchstens 255 Hops – oder eben gar nicht.

Damit IP-Datenpakete ihr Ziel überhaupt erreichen können, muss im Prinzip jeder einzelne Router im gesamten Internet darüber Bescheid wissen, wie er jedes beliebige Netz erreichen kann. Zu diesem Zweck unterhält auch jeder Router Routing-Tabellen, die den bereits für die einzelnen Hosts gezeigten ähnlich sehen. Da das Internet ein Zusammenschluss aus vielen einzelnen Netzwerken ist, müssen diese Tabellen jedoch ständig aktualisiert werden, denn es ergeben sich häufig Konfigurationsänderungen, weil neue Netze hinzukommen oder vorhandene geändert oder aufgegeben werden. Es wäre absolut unzumutbar, diese Konfigurationsänderungen ständig manuell auf dem aktuellen Stand zu halten, was deshalb auch seit vielen Jahren nicht mehr üblich ist (außer innerhalb sehr kleiner Netze wie in dem Beispiel zuvor, in denen sich die Routing-Einstellungen selten ändern müssen).

Die Router im Internet müssen deshalb ständig Informationen darüber austauschen, an welche anderen Netzwerke sie jeweils Daten vermitteln. Sie müssen komplexe Routing-Entscheidungen treffen, indem sie den Aufwand und die Kosten verschiedener Routen vergleichen und die Pakete eben nicht direkt ans Ziel, sondern auf dem derzeit günstigsten Weg weiterleiten, damit diese nicht nur sicher, sondern auch möglichst schnell ihr Ziel erreichen. Dieses eigentliche Routing ist erheblich dynamischer als das einfache Forwarding, sodass die Routing-Informationen ständig aktualisiert werden müssen. Auf diese Weise kann ein Paket bei Ausfall oder auch nur starker Belastung einer bestimmten Route über eine andere Route umgeleitet werden. Zu diesem Zweck wurden eine Reihe verschiedener Routing-Protokolle entwickelt, mit deren Hilfe dies bewerkstelligt wird. Jedes dieser Routing-Protokolle besitzt andere Eigenschaften, außerdem wird nicht jedes dieser Protokolle von jedem Hersteller unterstützt.

Zunächst muss zwischen zwei Arten von Routing unterschieden werden: dem Routing innerhalb zusammenhängender Netze eines einzelnen Betreibers (Interior Routing), der innerhalb dieses Bereiches frei über die Konfiguration entscheiden kann, und dem Routing zwischen voneinander unabhängigen derartigen Bereichen (Exterior Routing). Alle zusammenhängenden Netze eines Betreibers werden als *autonome Systeme* (Autonomous Systems, abgekürzt AS) bezeichnet. Einige Routing-Protokolle, etwa das veraltete RIP oder das aktuellere OSPF, dienen dem Routing innerhalb von autonomen Systemen, während andere, vor allem BGP, für das Routing zwischen den Grenzen autonomer Systeme zuständig sind. Diese drei genannten Routing-Protokolle werden im weiteren Verlauf des Kapitels kurz vorgestellt.

Wenn ein Router ein Routing-Protokoll ausführt, dann teilt er den benachbarten Routern mit, an welche Netze er Daten weiterleitet. Die meisten Routing-Protokolle machen außerdem Angaben über die »Kosten«, die für das Erreichen eines bestimmten Netzes kalkuliert werden müssen. Der Begriff *Kosten* hat nichts mit dem Preis zu tun, sondern bestimmt vor allem, über wie viele Hops ein bestimmtes Netzwerk durch den jeweiligen Router erreicht werden kann. Allerdings gibt es auch die Möglichkeit, die Kostenangaben willkürlich zu manipulieren – je nachdem, wie »gern« ein Router Daten an ein bestimmtes Netzwerk übermitteln soll. Wenn ein Router bestimmen muss, an welchen benachbarten Router er die Daten für ein bestimmtes Netz übergeben soll, sucht er sich denjenigen aus, der für dieses Netz geringere Kosten angibt. Diese Kostendaten werden auch als die *Metrik* des Routings bezeichnet.

Auf diese Weise wird versucht, die Datenströme zwischen den verschiedenen Backbone-Netzwerken möglichst gleichmäßig zu verteilen, außerdem bestehen verschiedene Arten von Verträgen oder Vereinbarungen zwischen den Netzbetreibern, was die Weiterleitung von Daten bestimmter anderer Netzwerke betrifft. Beispielsweise bestand in Deutschland in den 90er-Jahren ein mehrjähriger Streit zwischen dem Deutschen Forschungsnetz (DFN), dem Betreiber der deutschen Universitätsnetze, und den kommerziellen Internetprovidern. Es ging um die Frage, wer wem mehr Datenverkehr aus dem jeweils anderen Netz zumutete. Erst durch die Einführung neuer zentraler Datenaustauschpunkte wie dem DE-CIX konnte der Konflikt beigelegt werden.

Hier einige wichtige Routing-Protokolle im Überblick:

* Routing Information Protocol (RIP)  
  Das Routing Information Protocol (RIP) wird auf Unix-Routern durch den Routing Daemon (routed) ausgeführt. Beim Start von routed wird eine Anfrage ausgesendet. Alle anderen Router, die innerhalb desselben autonomen Systems ebenfalls routed ausführen, beantworten diese Anfrage durch Update-Pakete. Darin sind die Zieladressen aus den Routing-Tabellen der anderen Router und deren jeweilige Metrik enthalten.

Enthält ein Update-Paket die Routen zu Netzen, die noch gar nicht bekannt sind, fügt der Router sie zu seiner Routing-Tabelle hinzu. Außerdem werden Routen ersetzt, falls ein Update-Paket die Information enthält, dass ein bestimmtes Netzwerk über einen anderen Router mit geringeren Kosten zu erreichen ist.

Ein Router, auf dem routed läuft, sendet ebenfalls Update-Pakete, und zwar in der Regel alle 30 Sekunden. Erhält ein Router von einem anderen Router mehrere Male keine Update-Pakete mehr (häufig beträgt die Wartezeit 180 Sekunden), löscht er alle Einträge aus seiner Routing-Tabelle, die diesen Router verwenden. Außerdem werden diejenigen Einträge gelöscht, deren Kosten mehr als 15 Hops betragen. Letzteres beschränkt RIP auf kleinere autonome Systeme.

RIP interpretiert IP-Adressen streng nach der alten Klassen-Logik und beherrscht weder CIDR noch VLSM. Dies ist der Hauptgrund, warum es immer seltener verwendet wird.

Außerdem besteht das Problem, dass durch den plötzlichen Ausfall von Routern Konfigurationsfehler entstehen können: Alle Netze, die ursprünglich nur durch den ausgefallenen Router erreicht werden konnten, können nun gar nicht mehr erreicht werden. Dies spricht sich jedoch nur allmählich herum, da ein Router zwar zunächst alle Routen entfernt, die durch den ausgefallenen Router führten, von den anderen jedoch wieder die Route zu dem Netz lernt, das nun nicht mehr erreichbar ist. Bei einem Update-Intervall von 30 Sekunden kann es recht lange dauern, bis die Router die Entfernung zu dem nicht mehr verfügbaren Netz auf die nicht mehr relevanten 16 Hops »hochgeschaukelt« haben.

Um dieses Szenario zu verhindern, wird eine Technik namens *Split Horizon* verwendet: Ein Router bietet Routing-Informationen nicht über die Verbindung an, über die er sie gelernt hat. Eine Erweiterung dieses Verfahrens ist *Poison Reverse*; hier wird den Routern, von denen eine bestimmte Verbindung gelernt wurde, aktiv die »Unendlich-Metrik« 16 angegeben.

Einige Probleme von RIP werden in der neueren Version RIP-2, die in RFC 1723 beschrieben wird, beseitigt; vor allem arbeitet diese Version mit CIDR-Adressierung.

* Open Shortest Path First (OSPF)  
  Das in RFC 2178 beschriebene Open-Shortest-Path-First-Protokoll (OSPF) ist ein sogenanntes *Link-State-Protokoll*: Der einzelne Router speichert einen gerichteten Graphen des Netzwerks aus seiner jeweiligen Sicht. Ein gerichteter Graph ist eine Art Baumdiagramm mit dem lokalen Router als Wurzel; sein Aufbau geschieht nach dem Shortest-Path-First-Algorithmus von Dijkstra: Die Kosten des lokalen Routers selbst werden mit 0 angegeben; von diesem zweigen die Routen zu den Nachbarn baumförmig ab, dann wiederum zu deren Nachbarn und so weiter. In einem zweiten Schritt wird der Link-State-Graph optimiert. Falls mehrere Routen zu einem Ziel bestehen, beispielsweise eine direkte und eine indirekte, wird jeweils die weniger kostengünstige Route entfernt.

Um die Link-State-Datenbank klein zu halten, werden größere autonome Systeme in kleinere Einheiten unterteilt, die *Areas*. Nur vereinzelte Router, die sogenannten *Bereichsgrenzrouter*, werden von den Routern innerhalb einer Area als Verbindung in andere Areas betrachtet.

Ein OSPF-Router gewinnt seine Erkenntnisse über die benachbarten Router, indem er sogenannte *Hello-Pakete* aussendet. Diese enthalten seine eigene Adresse und die Information, von welchen benachbarten Routern er bereits Routing-Daten erhalten hat. Ein Router, der ein Hello-Paket erhält, trägt den Absender dieses Pakets als Nachbarn in seinen eigenen Link-State-Graphen ein. Die Hello-Pakete werden in regelmäßigen Abständen ausgesandt, um den Nachbarn mitzuteilen, dass der Router noch bereit ist. Erhält ein Router keine weiteren Pakete von einem bestimmten Nachbarn, geht er davon aus, dass dieser nicht mehr zur Verfügung steht, entfernt ihn aus seiner Link-State-Datenbank und informiert das Netzwerk darüber.

OSPF-Router geben Daten über ihre Nachbarn an das gesamte Netzwerk weiter, indem sie Link State Advertisements (LSA) über alle ihre Netzwerkschnittstellen versenden. Der Empfänger eines LSA-Pakets leitet es weiter, indem er es ebenfalls über alle seine Schnittstellen versendet – mit Ausnahme derjenigen, über die er es empfangen hat. Dieses Verfahren der schnellen Verbreitung von Informationen über ein Netzwerk wird als *Flooding* bezeichnet.

* Border Gateway Protocol (BGP)  
  Anders als bei den beiden zuvor behandelten Routing-Protokollen handelt es sich beim Border Gateway Protocol (BGP) um ein externes Routing-Protokoll, das Verbindungen zwischen verschiedenen autonomen Systemen regelt. Vom Standpunkt des externen Routings aus erscheinen die autonomen Systeme selbst als in sich geschlossene Gebilde, die nicht näher differenziert werden. BGP wird nur von den Bereichsgrenzroutern der autonomen Systeme ausgeführt, also in der Regel lediglich bei Internetprovidern oder großen Backbone-Netzbetreibern. Die meisten Firmennetze sind dagegen Teil eines autonomen Systems, das von einem Provider betrieben wird, führen also lediglich interne Routing-Protokolle wie OSPF aus.

Die benachbarten BGP-Router, *Peers* genannt, kommunizieren über eine zuverlässige TCP-Verbindung, die über den dafür vorgesehenen TCP-Port 179 abgewickelt wird. Es wird stets eine vollständige Route mit allen ihren Knotenpunkten angegeben. Dies unterscheidet BGP von den meisten internen Routing-Protokollen, die nur die Verbindungen zu ihren unmittelbaren Nachbarn angeben. Aus diesem Grund wird BGP als *Pfadvektor-Protokoll* bezeichnet.

Wird das erste Mal eine Verbindung zu einem Peer hergestellt, dann werden über sogenannte *UPDATE-Pakete* die vollständigen Routing-Tabellen ausgetauscht, danach werden nur noch Änderungen mitgeteilt. Außerdem werden in regelmäßigen Abständen KEEPALIVE-Pakete versandt, falls keine Änderungen vorliegen, um den Peers mitzuteilen, dass der Router noch einsatzbereit ist.

# Weitere IP-Dienste

In fast allen modernen TCP/IP-Netzwerken – insbesondere in lokalen Firmennetzen, die mit dem Internet verbunden sind – spielen zwei weitere Protokolle eine wichtige Rolle: DHCP dient dazu, den Rechnern im Netzwerk automatisch IP-Adressen zuzuweisen, während das NAT-Protokoll meist vom Standard-Router ausgeführt wird und die im Internet unbrauchbaren privaten IP-Adressen mit öffentlichen überschreibt und umgekehrt. Diese beiden Protokolle sollen hier näher vorgestellt werden.

Das in RFC 2131 und 2132 definierte Dynamic Host Configuration Protocol (DHCP) dient dazu, einem Host automatisch TCP/IP-Konfigurationsdaten zuzuweisen. Es ist eine Erweiterung des älteren Bootstrap Protocols (BOOTP). Ein Host, der seine Netzwerkparameter über DHCP beziehen möchte, sendet bei Inbetriebnahme eine Broadcast-Anfrage namens *BOOTREQUEST* an die allgemeine Broadcast-Adresse 255.255.255.255. Der Rechner muss also noch nicht einmal wissen, in welchem Netzwerk er sich befindet – das ist beispielsweise ideal für ein Notebook, das manchmal an ein Heim- und manchmal an ein Büro-Netzwerk angeschlossen wird. Läuft in dem Netz ein DHCP-Server, antwortet er mit einem Satz von Konfigurationsparametern, mit denen der Host seine TCP/IP-Konfiguration durchführt.

Das wichtigste Merkmal von DHCP besteht in der dynamischen Vergabe von IP-Adressen, die Netzwerkadministratoren das Leben erheblich erleichtert – insbesondere in solchen Netzwerken, in denen häufig Änderungen auftreten. Diese automatische Vergabe erfolgt in Form einer »Lease« (Pacht) mit beschränkter Gültigkeit. Ein Host, der ordnungsgemäß vom Netz abgemeldet wird (ein normaler Vorgang beim Herunterfahren moderner Betriebssysteme), gibt seine IP-Adresse selbst an den DHCP-Server zurück. Das Lease-Verfahren sorgt dagegen dafür, dass IP-Adressen auch dann wieder für den Server verfügbar werden, wenn ein Host unerwartet vom Netz getrennt oder unsachgemäß abgeschaltet wird. Bleibt ein Rechner über den Lease-Zeitraum hinaus im Netz aktiv, erfolgt in der Regel eine Verlängerung der Lease.

Auf dem DHCP-Server muss ein Teil der Adressen des Netzwerks, in dem er sich befindet, als DHCP-Pool konfiguriert werden, aus dem die Adressen automatisch an die anfragenden DHCP-Clients vergeben werden. Es muss darauf geachtet werden, genügend Adressen aus diesem Pool auszuschließen, weil eine Reihe von Internetdiensten eine feste IP-Adresse benötigt oder zumindest besser damit funktioniert.

Network Address Translation (NAT) ist eine relativ neue Entwicklung und löst dementsprechend ein modernes Problem: Immer mehr Netzwerke benötigen permanenten oder auch nur temporären Zugang zum Internet, obwohl sie mit den zuvor vorgestellten privaten IP-Adressen konfiguriert wurden. Es wäre bei der heutigen Anzahl von Internethosts und angeschlossenen Netzen auch gar nicht mehr möglich, allen angeschlossenen Netzwerken öffentliche IP-Adressen zuzuweisen. Da die privaten IP-Adressen jedoch nicht eindeutig sind, müssen sie beim Übergang ins Internet mit einer öffentlichen Adresse überschrieben werden und umgekehrt.

Eine aktuelle Form von NAT, die im Kernel moderner Unix-Systeme konfiguriert werden kann, wird auch als *IP-Masquerading* bezeichnet und geht noch einen Schritt weiter als NAT: Es ist nur eine externe IP-Adresse erforderlich; alle lokalen Adressen werden auf diese eine Adresse abgebildet. Unterschieden werden die Rechner in diesem Fall anhand der Client-Portnummer der Datenpakete, die zur Transportebene gehört und im nächsten Abschnitt näher beschrieben wird. Aus diesem Grund wird das echte Masquerading manchmal auch als *PAT* (Port Address Translation) bezeichnet. Diese spezielle Form von NAT verbirgt die Details des internen Netzwerks vor dem Internet, die einzelnen Rechner sind von außen nicht erreichbar. Dies ist ein angenehmer Nebeneffekt dieses Verfahrens, der zusätzlich der Sicherheit im Netzwerk dient.

[Tabelle 4.17](https://openbook.rheinwerk-verlag.de/it_handbuch/04_006.html#rxz1006082) zeigt ein Beispiel für klassisches NAT in einem privaten Netzwerk mit der Adresse 192.168.1.0/24. Jede interne IP-Adresse wird auf eine individuelle externe Adresse abgebildet.

|  |  |  |
| --- | --- | --- |
| **Tabelle 4.17**Beispiel für klassisches NAT | | |
| **Hostname** | **interne IP-Adresse** | **externe IP-Adresse** |
| gandalf | 192.168.1.4 | 204.81.92.6 |
| Frodo | 192.168.1.5 | 204.81.92.3 |
| Bilbo | 192.168.1.6 | 204.81.92.5 |

In [Tabelle 4.18](https://openbook.rheinwerk-verlag.de/it_handbuch/04_006.html#rxz1006124) wird dagegen für dasselbe Netzwerk ein Beispiel für IP-Masquerading (PAT) gezeigt. Das Konzept der Portnummern wird im weiteren Verlauf des Kapitels noch genauer beschrieben.

|  |  |  |  |
| --- | --- | --- | --- |
| **Tabelle 4.18**Beispiel für IP-Masquerading | | | |
| **Hostname** | **interne IP-Adresse** | **externe IP-Adresse** | **externe Portnummer** |
| Gandalf | 192.168.1.4 | 204.81.92.4 | 22.191 |
| Frodo | 192.168.1.5 | 22.192 |
| Bilbo | 192.168.1.6 | 22.193 |

Der Rechner, der NAT ausführt, ist üblicherweise derjenige Router, der das lokale Netz mit dem Netzwerk eines Providers und demzufolge mit dem Internet verbindet. NAT wird von allen gängigen Unix-Versionen sowie von Windows NT und seinen Nachfolgern unterstützt. Außerdem können die meisten ISDN- oder DSL-Kompakt-Router NAT ausführen. Eine nähere Beschreibung vieler Aspekte von NAT findet sich in RFC 3022.

Auf einem Linux-System kann NAT beispielsweise durch die Kernel-Firewall Netfilter/iptables bereitgestellt werden. Dieser Aspekt wird auch in [Kapitel 20](https://openbook.rheinwerk-verlag.de/it_handbuch/20_001.html#rxz1026482), »Computer- und Netzwerksicherheit«, erwähnt.

Die Windows-Desktopsysteme enthalten einen eigenen NAT-Dienst namens *Internet Connection Sharing* (ICS). Dadurch fungiert eine Workstation als NAT-Router und ermöglich so die Nutzung ihres Internetzugangs durch andere Maschinen. Diese NAT-Variante gilt allerdings als unsicher und ist zudem erheblich unflexibler als IP-Masquerading, da sie automatisch erfolgt und keinerlei Einstellungen ermöglicht.

In manchen Netzwerken erfolgt der Zugriff auf Internetdienste nicht über Router, sondern über Gateways auf Anwendungsebene, die in der Öffentlichkeit als Stellvertreter (Proxys) des eigenen Rechners arbeiten. Solche Proxyserver gibt es für fast alle Dienste. Am bekanntesten sind Webproxys, die oft auch als Cache (Zwischenspeicher) für häufig aufgerufene Websites dienen. Die bekannteste Proxy- und Webcache-Software ist der Open-Source-Proxy *squid*; Microsoft bietet ebenfalls ein entsprechendes Produkt namens *ISA Server* an.

[Rheinwerk Computing - Zum Seitenanfang](https://openbook.rheinwerk-verlag.de/it_handbuch/04_006.html#top)

### 4.6.3 Transportprotokolle[Zur nächsten Überschrift](https://openbook.rheinwerk-verlag.de/it_handbuch/04_006.html#dodtpd5d73d94-a81f-44c8-8c80-14bbc8f9ca46)[Zur vorigen Überschrift](https://openbook.rheinwerk-verlag.de/it_handbuch/04_006.html#dodtp33548c8e-1929-412a-a10b-498b54d4b395)

Eine Anwendung, die Daten über ein TCP/IP-Netzwerk wie das Internet übertragen möchte, beauftragt zu diesem Zweck ein Transportprotokoll, also ein Protokoll der Host-zu-Host-Transportschicht des Internet-Schichtenmodells. Nachdem im vorigen Abschnitt das IP-Routing erläutert wurde, sollten Sie auch verstehen, warum der Vorgang als *Host-zu-Host-Transport* bezeichnet wird: Router betrachten von den Datenpaketen, die sie weiterleiten sollen, immer nur den IP-Header und werten dessen Informationen aus. Aus der Sicht des IP-Protokolls existieren die Daten der Transportschicht nicht. Umgekehrt ist also das Routing ein Implementierungsdetail, das für die Protokolle der Transportschicht nicht sichtbar ist. Aus ihrer Sicht kann der Zielhost immer unmittelbar erreicht werden.

Um den Bedürfnissen verschiedener Anwendungen gerecht zu werden, wurden zwei verschiedene wichtige Transportprotokolle definiert. Das häufiger verwendete *TCP-Protokoll* (definiert in RFC 793), das einen Teil des Namens der Protokollfamilie ausmacht, stellt den zuverlässigen Transport von Datenpaketen in einer definierten Reihenfolge zur Verfügung. Dagegen bietet das *UDP-Protokoll* (RFC 768) die Möglichkeit, Daten auf Kosten der Zuverlässigkeit möglichst schnell zu transportieren.

Ein wenig zwischen Vermittlungs- und Transportschicht liegt das ICMP-Protokoll (Internet Control Message Protocol), das für den Versand spezieller Datagramme verwendet wird, mit deren Hilfe überprüft werden kann, ob ein entfernter Host im Netzwerk aktiv ist. Das entsprechende Dienstprogramm heißt *ping* und wird in [Kapitel 6](https://openbook.rheinwerk-verlag.de/it_handbuch/06_001.html#rxz1008115), »Windows«, und in [Kapitel 7](https://openbook.rheinwerk-verlag.de/it_handbuch/07_001.html#rxz1009500), »Linux«, für die jeweilige Systemplattform vorgestellt.

Die beiden Protokolle werden in den folgenden Abschnitten genauer beschrieben.

# Das Transmission Control Protocol (TCP)

Wie Sie im vorigen Abschnitt erfahren haben, werden IP-Datagramme jeweils individuell durch das Netzwerk geleitet. Deshalb kann auf der Basis von Datagrammen kein zuverlässiger Transport kontinuierlicher Datenströme stattfinden, weil es vollkommen normal ist, dass Datagramme nicht in der Reihenfolge ankommen, in der sie abgeschickt wurden. Außerdem ist es auch möglich, dass sie gar nicht ankommen, weil auf der Ebene des IP-Protokolls keine entsprechende Kontrolle durchgeführt wird.

Um nun über den potenziell unsicheren Weg der IP-Datagramme Daten zuverlässig durch das Netzwerk zu transportieren, wird auf dieser höher gelegenen Ebene eine Flusskontrolle implementiert: Im Wesentlichen werden die Datenpakete durch das TCP-Protokoll durchnummeriert, um die korrekte Reihenfolge aufrechtzuerhalten. Im Übrigen erwartet der ursprüngliche Absender für jedes einzelne Datenpaket eine Bestätigung; bleibt sie zu lange aus, versendet der Absender das entsprechende Paket einfach erneut.

Als Erstes sollten Sie sich den TCP-Paket-Header ansehen, der in [Tabelle 4.19](https://openbook.rheinwerk-verlag.de/it_handbuch/04_006.html#rxz1006260) gezeigt wird.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tabelle 4.19**Aufbau des TCP-Datenpaket-Headers | | | | | | |
| **Byte** | **0** | | **1** | | **2** | **3** |
| **0** | Quellport | | | | Zielport | |
| **4** | Sequenznummer | | | | | |
| **8** | Bestätigungsnummer | | | | | |
| **12** | Offset | reserviert | | Flags | Fenster | |
| **16** | Prüfsumme | | | | Urgent-Zeiger | |
| **20** | Optionen | | | | | Padding |

Ein TCP-Datenpaket-Header besteht aus den folgenden Bestandteilen:

* Quellport (16 Bit): Ports stellen eine Methode zur Identifikation der konkreten Anwendungen zur Verfügung, die auf den beteiligten Hosts miteinander kommunizieren. Der Quellport ist die Portnummer des Absenders.
* Zielport (16 Bit): Dies ist entsprechend die Portnummer des Empfängers.
* Sequenznummer (32 Bit): Normalerweise gibt diese Nummer an, dem wievielten Byte der zu übertragenden Sequenz das erste Nutzdaten-Byte des Pakets entspricht. Ausnahme: Ist das SYN-Flag gesetzt, wird die Anfangs-Sequenznummer (Initial Sequence Number, ISN) angegeben.
* Bestätigungsnummer (32 Bit): Das Startbyte der Sequenz, deren Übertragung als Nächstes erwartet wird; ist nur bei gesetztem ACK-Bit von Bedeutung.
* Offset (4 Bit): Anzahl der 32-Bit-Wörter, aus denen der Header besteht; gibt entsprechend den Beginn der Nutzdaten im Paket an.
* Reserviert (6 Bit): Reserviert für zukünftige Anwendungen; muss 0 sein.
* Flags (6 Bit): Verschiedene Statusbits; im Einzelnen:
  + URG: Urgent Data wird versandt; der Inhalt des Urgent-Zeigers muss beachtet werden.
  + ACK: Acknowledgement – das Bestätigungsfeld muss berücksichtigt werden.
  + PSH: Push-Funktion – ist dieses Bit gesetzt, wird die Pufferung des Pakets verhindert; es wird unmittelbar gesendet.
  + RST: Reset – Verbindung zurücksetzen
  + SYN: Sequenznummern synchronisieren
  + FIN: Ende der Sequenz; keine weiteren Daten vom Absender
* Fenster (16 Bit): Die Anzahl von Datenbytes, die der Absender des Pakets zu empfangen bereit ist; basiert unter anderem auf der IP-MTU der verwendeten Schnittstelle.
* Prüfsumme (16 Bit): Anhand dieser einfacheren Plausibilitätskontrolle kann die Korrektheit der übertragenen Daten überprüft werden.
* Urgent-Zeiger (16 Bit): Ein Zeiger auf das Byte der aktuellen Sequenz, das Urgent Data enthält. Wird nur ausgewertet, wenn das URG-Flag gesetzt ist.
* Optionen (variable Länge): Enthält verschiedene hersteller- und implementierungsabhängige Zusatzinformationen; stets ein Vielfaches von 8 Bit lang.

Zwischen den beiden Hosts, die über TCP kommunizieren, wird eine virtuelle Punkt-zu-Punkt-Verbindung hergestellt; aus diesem Grund wird TCP auch als *verbindungsorientiertes Protokoll* bezeichnet. Dies ermöglicht den Transport eines kontinuierlichen Datenstroms über die potenziell unzuverlässigen IP-Datagramme, in die die TCP-Pakete verpackt werden. Um die Datenübertragung einzuleiten, findet zunächst ein sogenannter *Drei-Wege-Handshake* statt: Drei spezielle Datenpakete ohne Nutzdateninhalt werden ausgetauscht. Der Host, der die Verbindung initiiert, sendet ein Paket mit gesetztem SYN-Bit an den Empfänger. Dieser schickt ein Paket zurück, bei dem SYN und ACK gesetzt sind, und erwartet wiederum eine Antwort, bei der nur das ACK-Flag gesetzt ist. Erst nachdem dies geschehen ist, beginnt die eigentliche Übertragung von Nutzdaten. Dieses Vorgehen garantiert, dass beide Hosts bereit sind, miteinander zu kommunizieren.

Anschließend sendet der Absender ein Paket nach dem anderen an den Empfängerhost, wobei die Sequenznummer stets um die im vorigen Paket versandte Nutzdatenmenge erhöht wird. Der Empfänger beantwortet jedes empfangene Paket, dessen Prüfsumme mit dem Inhalt übereinstimmt, mit einem Bestätigungspaket, dessen ACK-Flag also gesetzt ist. Der Wert des Bestätigungsfelds ist der Byte-Offset der nächsten Datensequenz, die der Empfänger erwartet, ist also die Summe aus Sequenznummer und Nutzdatenlänge des soeben empfangenen Pakets.

Erhält der Absender die Bestätigung nicht innerhalb einer definierten Zeit (Timeout), sendet er das entsprechende Paket unaufgefordert erneut. Dieses Verfahren wird *positive Bestätigung* (Positive Acknowledgement) genannt, da lediglich der Erfolg gemeldet wird; von einem Misserfolg wird automatisch ausgegangen, wenn keine Meldung erfolgt. Dieses Verfahren ist zuverlässiger als das Arbeiten mit Misserfolgsmeldungen: Kommt die Erfolgsmeldung nicht an, wird das Paket einfach erneut versandt, ansonsten gibt es aber keine schädlichen Folgen (außer dem geringen Mehraufwand für ein überflüssig versandtes Paket, falls einmal lediglich die Bestätigung verloren gegangen ist). Käme dagegen eine Misserfolgsmeldung nicht an, würde das betreffende Paket nicht erneut versandt und würde den Empfänger niemals erreichen.

Ein weiterer wichtiger Bestandteil von TCP-Paketen sind die beiden 16 Bit langen Portnummern. Jedes Paket kann anhand des Portnummern-Paares als zu einer bestimmten Sequenz und Anwendung gehörig identifiziert werden. Das ist auch absolut notwendig: Stellen Sie sich vor, Sie haben zwei Browserfenster geöffnet; in beiden werden gleichzeitig verschiedene Seiten von *www.galileocomputing.de* geladen. Anhand der IP-Adressen können die beiden Datenübertragungen nicht voneinander unterschieden werden, da die beiden Hosts, die hier miteinander kommunizieren, identisch sind. Es könnte also sehr leicht passieren, dass die Daten fehlerhaft zugeordnet werden und Sie zwei seltsame Mischungen der beiden Dokumente erhalten – ein Effekt wie in dem Horror-Klassiker »Die Fliege«!

Dieses Szenario kann deshalb nicht eintreten, weil die beiden Datenübertragungen nicht über dasselbe Paar von Portnummern erfolgen. In der Regel ist die Portnummer des Servers festgelegt, während der Client irgendeinen Port wählt, der gerade frei ist. Die untersten 1.024 Portnummern sind als sogenannte *Well-known Ports* für Standard-Serverdienste fest vergeben; für Clients wird eine zufällige Nummer (ein sogenannter *Ephemeral Port*) zwischen 1.024 und 65.535 verwendet. Beispielsweise benutzen Webserver, also HTTP-Server, üblicherweise den TCP-Port 80, FTP-Server den Port 21 und Telnet-Server den Port 23. Eine kleine Liste, die auch UDP betrifft, finden Sie in [Tabelle 4.21](https://openbook.rheinwerk-verlag.de/it_handbuch/04_006.html#rxz1006482).

In dem Beispiel mit den beiden Browserfenstern ist der Server-Port jeweils 80; die Client-Ports sind dagegen unterschiedlich, beispielsweise 16832 und 16723. Dies ist eine Verdeutlichung der Formulierung, dass nur die Portpaare und nicht die beiden einzelnen Ports unterschiedlich sein müssen, um Sequenzen voneinander abzugrenzen.

Gewöhnlich »lauscht« ein TCP-Serverdienst an seinem speziellen Port auf ankommende Verbindungsversuche. Unternimmt ein Client den Versuch, eine TCP-Verbindung mit dieser speziellen Portnummer als Empfängerport und einer zufälligen Nummer als Absender herzustellen, akzeptiert der Server dies nach den Regeln des Drei-Wege-Handshakes; eine Verbindung für den gegenseitigen Datenaustausch ist hergestellt.

Interessant ist schließlich das Thema Urgent Data: Manchmal muss ein Host einen anderen über einen besonderen Zustand informieren, beispielsweise eine Konfigurationsänderung oder einen vom Benutzer initiierten Abbruch mitteilen. Zu diesem Zweck wird das URG-Flag gesetzt; der Empfänger ermittelt daraufhin aus dem Urgent-Zeiger-Feld des Paket-Headers die Bytenummer innerhalb der Sequenz, in der sich diese dringlichen Daten befinden. Es handelt sich stets nur um ein einziges Byte, das auch als *Out-of-Bound-Byte* bezeichnet wird, weil es nicht zum gewöhnlichen Datenstrom gehört. Es ist also unmöglich, auf diesem Weg eine längere dringende Mitteilung zu versenden, aber immerhin besteht die Möglichkeit, bestimmte zwischen den Anwendungen vereinbarte Signale auszutauschen.

# Das User Datagram Protocol (UDP)

Manche Anwendungen möchten auf den Komfort und die Sicherheit von TCP getrost verzichten, wenn sie die Daten dafür schneller ans Ziel befördern können. Die Möglichkeit eines solchen möglichst schnellen Versandes bietet das UDP-Protokoll. Ob eine Anwendung für ihre Datenübertragung nun TCP, UDP oder beide verwenden möchte, entscheidet sie selbst.

Stellen Sie sich als Beispiel ein Netzwerkspiel vor, eine virtuelle 3D-Umgebung, in der Sie gegen Ihre Mitspieler »kämpfen« können. Ein solches Spiel ist ideal für die Erklärung des Nutzens beider Übertragungsarten geeignet: Bestimmte grundlegende Konfigurationsdaten (Lebt der Gegner überhaupt noch? Hat er auf mich geschossen?) sind entscheidend für den eigentlichen Spielverlauf und sollten deshalb zuverlässig über TCP übertragen werden. Dagegen sind bestimmte Details (Pose und Gesichtsausdruck der gegnerischen Spielfigur; die Position von Gegnern außerhalb des »Gesichtsfelds« und so weiter) nicht so wichtig. Wenn überhaupt, sollten sie möglichst schnell übertragen werden. Fallen sie vorübergehend aus, schadet das auch nichts – ideale Kandidaten für die Übertragung mithilfe des schnelleren, aber weniger zuverlässigen UDP-Protokolls.

Der Hauptgrund, warum sich Daten über UDP schneller übertragen lassen als über TCP, ist der erheblich kleinere und weniger komplexe Paket-Header. Der Aufbau dieses Headers, der gerade einmal (unveränderlich) 64 Bit groß ist, wird in [Tabelle 4.20](https://openbook.rheinwerk-verlag.de/it_handbuch/04_006.html#rxz1006358) dargestellt.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tabelle 4.20**Aufbau des UDP-Headers | | | | |
| **Byte** | **0** | **1** | **2** | **3** |
| **0** | Quellport | | Zielport | |
| **4** | Länge | | Prüfsumme | |

Die einzelnen Header-Felder haben dieselbe Bedeutung wie die gleichnamigen Felder beim TCP-Protokoll. Mit der Länge ist hier die Länge des gesamten Pakets inklusive dieses Headers gemeint. Der Quellport wird häufig einfach auf 0 gesetzt: Da UDP dem schnellen Versand einer einzelnen Nachricht dient, auf die in der Regel keine Antwort erwartet wird, ist es nicht nötig, diese Information festzulegen. Der Zielport ist dagegen meist der festgelegte Port des UDP-Servers, an den das Paket verschickt wird. UDP wird für viele einfache Internetdienste verwendet: die Uhrzeitsynchronisation über ein Netzwerk, den Echo-Dienst zur Kontrolle der Funktionstüchtigkeit von Verbindungen oder entfernten Hosts und so weiter.

Im Gegensatz zu dem verbindungsorientierten TCP wird UDP als *nachrichtenorientiertes Protokoll* bezeichnet, da es dem schnellen, verbindungslosen Versand einzelner Pakete in Form kurzer Meldungen dient. Dies erklärt auch den Namen des Protokolls: Einer Anwendung, die von diesem Transportdienst Gebrauch macht, wird der unmittelbare und leichtgewichtige Zugriff auf IP-Datagramme ermöglicht.

Die Portnummern für gängige Serverdienste (bei UDP spricht man häufig auch von Servicenummern) liegen wie bei TCP zwischen 0 und 1023. Sie werden genau wie öffentliche IP-Adressen von der IANA vergeben. In der Regel wird dieselbe Portnummer für beide Transportprotokolle verwendet, obwohl die meisten Anwendungen nur auf jeweils einem der beiden Protokolle laufen. [Tabelle 4.21](https://openbook.rheinwerk-verlag.de/it_handbuch/04_006.html#rxz1006482) zeigt einige häufig verwendete Beispiele mit ihrem offiziellen Namen und dem am häufigsten verwendeten Transportprotokoll. Die vollständige Liste aller öffentlichen Serverdienste finden Sie unter [*http://www.iana.org/assignments/port-numbers*](http://www.iana.org/assignments/port-numbers). Falls Sie ein Unix-System verwenden, steht eine ähnliche, möglicherweise weniger vollständige Liste in der Konfigurationsdatei */etc/services*.

|  |  |  |  |
| --- | --- | --- | --- |
| **Tabelle 4.21**Einige TCP/UDP-Portnummern für gängige Dienste | | | |
| **Nummer** | **Transport- protokoll** | **Name** | **Beschreibung** |
| 7 | TCP, UDP | echo | genaue Rückgabe der übermittelten Daten zur Kontrolle |
| 13 | TCP, UDP | daytime | Datum und Uhrzeit (RFC 867) |
| 20 | TCP | ftp | FTP-Datenstrom |
| 21 | TCP | ftp | FTP-Steuerung |
| 22 | TCP | ssh | Secure Shell – Telnet-Alternative mit Verschlüsselung |
| 23 | TCP | telnet | Terminal-Emulation |
| 25 | TCP | smtp | E-Mail-Versand |
| 53 | TCP, UDP | domain | Nameserver-Abfragen |
| 80 | TCP | http | Webserver |
| 110 | TCP | pop3 | E-Mail-Postfach-Server (klassisch) |
| 142 | TCP | imap | E-Mail-Postfach-Server (modern) |
| 443 | TCP, UDP | https | SSL-verschlüsselte Webserver-Kommunikation |

[Rheinwerk Computing - Zum Seitenanfang](https://openbook.rheinwerk-verlag.de/it_handbuch/04_006.html#top)

# 4.6.4 Das Domain Name System (DNS)[Zur nächsten Überschrift](https://openbook.rheinwerk-verlag.de/it_handbuch/04_006.html#dodtp827791ac-b465-4f3f-8157-2b6b50c8174e)[Zur vorigen Überschrift](https://openbook.rheinwerk-verlag.de/it_handbuch/04_006.html#dodtp8ca634a2-918d-4dcf-b22e-077911da245a)

Die Verwendung von IP-Adressen zum Erreichen entfernter Rechner ist ideal, solange in die Datenübertragung nur Computer involviert sind. Für die Verwendung durch Menschen sind sie weniger gut geeignet (es gibt zum Beispiel nur wenige Menschen, die sich Telefonnummern auf Anhieb besser merken können als die zugehörigen Namen). Aus diesem Grund ist es seit Beginn des Internets und seiner Vorläufer üblich, einen Mechanismus einzurichten, der den beteiligten Menschen die Verwendung benutzerfreundlicher Namen statt der unhandlichen IP-Adressen ermöglicht.

Als das ARPANET entwickelt wurde, behalf man sich mit einer einfachen Textdatei, die pro Zeile einen Hostnamen und eine IP-Adresse nebeneinander auflistete. Noch heute verwenden Unix-Rechner eine ähnliche Datei namens */etc/hosts*. Auch unter Windows ist das Verfahren bekannt. Hier befindet sich die Datei in *<Windows-Verzeichnis>\system32\drivers\etc* und heißt – untypisch für Windows – ebenfalls nur *hosts*, ohne Dateiendung. Allerdings wird dieses Verfahren heute immer seltener für die Namenszuordnung in lokalen Netzen eingesetzt, weil in immer mehr Firmennetzen DHCP verwendet wird.

Findet der Rechner einen Namenseintrag in seiner */etc/hosts*-Datei, wird er die entsprechende Adresse nicht mehr bei einem Nameserver nachfragen. Auf diese Weise können Sie natürlich Ihre Kollegen ein wenig ärgern: Machen Sie beispielsweise die IP-Adresse ausfindig, die zu [*www.yahoo.de*](http://www.yahoo.de/) gehört (aktuell 77.238.178.122), und tragen Sie in die Datei */etc/hosts* eines Kollegen etwa folgende Zeile ein:

77.238.178.122 www.google.de

Jedes Mal, wenn der Kollege nun die Suchmaschine Google anwenden möchte, wird er stattdessen bei Yahoo! landen, kann sich das aber zunächst beim besten Willen nicht erklären.

Früher wurde die Datei namens *hosts.txt* zentral verwaltet und regelmäßig unter den teilnehmenden Hosts im ARPANET ausgetauscht, um die Namensdaten aktuell zu halten. Als das Netz jedoch immer größer wurde, funktionierte dieses System nicht mehr, weil man mit den häufigen Änderungen nicht mehr nachkam und weil die gesamte Datei außerdem sehr umfangreich war und ihr Versand eine erhebliche Menge an Netzwerkverkehr erzeugte.

Schließlich wurde statt der einfachen Textdatei eine hierarchische, vernetzte Datenbank eingeführt, die bis heute ein verteiltes Netz von Nameservern bildet. Diese Server geben auf Anfrage Auskunft über die zu einem Hostnamen gehörende IP-Adresse oder umgekehrt. Außerdem leiten sie die Anfrage automatisch weiter, wenn sie selbst keine Antwort wissen. Das System wird als *Domain Name System* (DNS) bezeichnet und ist Thema einer ganzen Reihe von RFCs. Die wesentlichen Grundlagen werden in RFC 1034 und 1035 beschrieben.

Damit Hostnamen im gesamten Internet eindeutig sind, werden sie hierarchisch als sogenannte *Domainnamen* vergeben. Zu diesem Zweck wird ein Name aus immer spezialisierteren Bestandteilen zusammengesetzt; das System lässt sich mit einem Pfad in einem Dateisystem vergleichen. Allerdings besteht ein wesentlicher Unterschied: Beim Dateisystempfad steht der allgemeinste Name vorn und der speziellste hinten, während es beim Domainnamen genau umgekehrt ist.

Beispielsweise bedeutet der Unix-Pfad */home/sascha/hb\_fachinfo/netzwerk/protokolle.txt*, dass sich die Datei *protokolle.txt* im Verzeichnis *netzwerk* befindet, einem Unterverzeichnis von *hb\_fachinfo*, das wiederum dem Verzeichnis *sascha* untergeordnet ist. *sascha* ist seinerseits ein Unterverzeichnis von *home*, das schließlich direkt unter der Wurzel des Dateisystems ( */* ) liegt.

Dagegen ist der Domainname [*www.buecher.lingoworld.de*](http://www.buecher.lingoworld.de/) genau umgekehrt aufgebaut: Der Host/Dienst *www* liegt in der Domain *buecher*, einer Subdomain von *lingoworld* in der Top-Level-Domain *de*. Die Wurzel des DNS-Systems selbst ist nicht sichtbar, weil ihr Name der leere String ist.

Auf der jeweiligen Ebene des DNS-Systems muss ein bestimmter Name einmalig sein. Beispielsweise kann es *buecher.lingoworld.de* nur einmal geben. Unterhalb dieser Domain können untergeordnete Domains (Subdomains) oder die Namen einzelner Hosts oder Serverdienste bestehen, beispielsweise *www.buecher.lingoworld.de*, *ftp.buecher.lingoworld.de* oder *neuheiten.buecher.lingoworld.de*. Im Übrigen dürfen dieselben Namen natürlich auf über- oder untergeordneten oder auch auf »Geschwister-Ebenen« existieren: Es kann die Website *www.lingoworld.de* ebenso geben wie etwa *www.download.lingoworld.de*. Selbstverständlich ist auch *www.buecher.de* kein Problem – es handelt sich um eine andere Domain unterhalb der Top-Level-Domain *de*.

Aus der Sicht der DNS-Administration wird jede Ebene eines solchen Namens auch als *Zone* bezeichnet, weil eine solche Ebene jeweils unabhängig von den übergeordneten Ebenen verwaltet wird. Beispielsweise kann der Administrator der Domain *lingoworld.de* Subdomains wie *buecher.lingoworld.de* oder *download.lingoworld.de* einrichten. Er kann die Verantwortung für eine Subdomain auch an jemand anderen delegieren, für den dann beispielsweise *buecher.lingoworld.de* wieder eine unabhängige Zone darstellt. Andererseits kann der Zonenverantwortliche für *lingoworld.de* nicht auf andere Zonen in der Domain *.de* zugreifen; beispielsweise geht ihn die Konfiguration der Zone *google.de* nichts an.

Die Infrastruktur der Domainnamen wird von den über das gesamte Internet verbreiteten Nameservern verwaltet. Diese führen alle ein Programm aus, das Anfragen nach Name-Adresse-Zuordnungen beantwortet, unbekannte Zuordnungen bei anderen Nameservern erfragt und dann meistens dauerhaft speichert. Die am häufigsten verwendete derartige Software heißt BIND (Berkeley Internet Name Domain) und läuft unter allen Unix-Varianten; sie wird in [Kapitel 14](https://openbook.rheinwerk-verlag.de/it_handbuch/14_001.html#rxz1020525), »Weitere Internet Serverdienste«, vorgestellt.

Auf der obersten Ebene des DNS existiert die spezielle Zone, deren Name der leere String ist. Diese Zone wird durch die Root-Nameserver der ICANN verwaltet und enthält Verweise auf alle Top-Level-Domains. Davon gibt es organisatorisch gesehen zwei Sorten (auch wenn es keinen technischen Unterschied gibt):

* Die »Generic TLDs« (allgemeine Top-Level-Domains) wie *.com* oder *.org* unterteilen die jeweiligen Domains, die unter ihnen liegen, nach der Funktion ihrer Betreiber.
* Die »Country TLDs« oder »ccTLDs« (Länder-TLDs) sind dagegen für eine geografische Einteilung vorgesehen.

In der Praxis kommt es ohnehin zu einer Vermischung: Zum einen sind viele Generic TLDs mittlerweile für beliebige Betreiber verwendbar, zum anderen gibt es einige Länder-TLDs, die wegen ihrer spezifischen Abkürzung für bestimmte Branchen interessant sind – am bekanntesten ist in diesem Zusammenhang wohl der Südsee-Inselstaat Tuvalu mit seiner bei Fernsehsendern beliebten TLD *.tv*.

[Tabelle 4.22](https://openbook.rheinwerk-verlag.de/it_handbuch/04_006.html#rxz1006629) listet einige häufig verwendete Top-Level-Domains auf. Die mit Abstand meisten Betreiber-Domains enthält die Generic-TLD *.com*, gefolgt von der länderspezifischen Domain *.de* (Deutschland).

|  |  |
| --- | --- |
| **Tabelle 4.22**Übersicht über einige wichtige Top-Level-Domains | |
| **Top-Level-Domain** | **Bedeutung** |
| **Generic Top-Level-Domains** | |
| *.com* | *commercial* (Firmen) |
| *.org* | *organization* (Organisationen und Vereine) |
| *.net* | *network* (Netzwerkbetreiber; Internet-Infrastruktur) |
| *.edu* | *educational* (US-Schulen und -Universitäten) |
| *.gov* | *government* (US-Regierung, US-Behörden, öffentlicher Dienst) |
| *.mil* | *military* (US-Militär) |
| *.info* | *information* (allgemeine Informationsdienste) |
| *.aero* | *aeronautics* (Luftfahrtindustrie, Fluggesellschaften) |
| **Länder-Top-Level-Domains** | |
| *.at* | Österreich |
| *.ch* | Schweiz |
| *.cn* | Volksrepublik China |
| *.de* | Deutschland |
| *.es* | Spanien |
| *.fr* | Frankreich |
| *.it* | Italien |
| *.jp* | Japan |
| *.ru* | Russland |
| *.tr* | Türkei |
| *.uk* | Vereinigtes Königreich |
| *.us* | USA |
| *.va* | Vatikanstadt |

Der jeweilige Haupt-Nameserver einer Top-Level-Domain enthält Verweise auf sämtliche unterhalb dieser Domain befindlichen Second-Level-Domains. Je nach konkreter TLD handelt es sich dabei entweder unmittelbar um die einzelnen Domains, die von Betreibern angemeldet werden können, oder eine Domain ist in sich noch einmal in Organisationsstrukturen unterteilt. Bei allen Generic-TLDs und den meisten Länder-TLDs ist Ersteres der Fall. Nur einige Länder-TLDs werden noch einmal organisatorisch unterteilt: Zum Beispiel verwendet das Vereinigte Königreich Unterteilungen wie *.co.uk* für Firmen, *.ac.uk* für Universitäten oder *.org.uk* für Vereine und Organisationen; auch die Türkei verwendet die entsprechenden Unterteilungen *.com.tr*, *.edu.tr* und *.org.tr*.

Aus Sicherheitsgründen sollten die Zonendaten für die Domain eines einzelnen Betreibers auf mindestens zwei voneinander unabhängigen (das heißt in verschiedenen autonomen Systemen befindlichen) Nameservern vorliegen. Die Daten müssen dafür nur auf einem der beiden Server erstellt werden, der als *primärer Master-Nameserver* bezeichnet wird; der andere – *Slave-Nameserver* genannt – repliziert sie automatisch. Größere Unternehmen und Institutionen verwalten in der Regel ihre eigenen Zonen. Der externe Slave-Nameserver mit denselben Zonendaten befindet sich in diesem Fall meist beim zuständigen Backbone-Provider, über den diese Betreiber mit dem Internet verbunden sind. Privatanwender oder kleinere Firmen besitzen dagegen zwar häufig eine eigene Domain (*www.meinname.de* ist werbewirksamer als so etwas wie *home.t-online.de/users/meinname*), unter dieser Domain laufen allerdings oft lediglich eine beim Provider gehostete Website und einige E-Mail-Adressen. In diesem Fall werden die Zonendaten meist beim Hosting-Provider und einem anderen Provider verwaltet; die beiden Provider stellen sich den Slave-Name-Service dann gegenseitig zur Verfügung.

Bei den meisten Einzelrechnern oder kleineren Firmennetzwerken besteht die ganze DNS-Konfiguration oft lediglich aus der Eingabe der IP-Adresse eines Nameservers des eigenen Providers; in vielen Fällen ist sogar dies unnötig, weil die Standard-Nameserver beim Verbindungsaufbau bekanntgegeben werden. Die Nameserver werden stets befragt, wenn Namensdaten erforderlich sind.

Gebe ich zum Beispiel in meinem Webbrowser [*www.google.de*](http://www.google.de/) ein, überprüft dieser zunächst, ob er die IP-Adresse vielleicht bereits kennt. Andernfalls fragt er den Standard-Nameserver des Providers. Dieser weiß die Antwort entweder selbst und liefert sie unmittelbar zurück oder wendet sich an den übergeordneten Nameserver – in diesem Fall den für die Top-Level-Domain *.de* zuständigen Server. Der wiederum kennt die für die Domain *google.de* zuständigen Nameserver und leitet die Anfrage an den ersten von ihnen weiter. Dieser ermittelt die IP-Adresse des Dienstes [*www.google.de*](http://www.google.de/) und gibt sie zurück. Nun weiß der Browser, welche IP-Adresse er verwenden muss. Außerdem speichert der Nameserver des Providers die gefundene Adresse ebenfalls in seinem Cache ab, um die nächste entsprechende Anfrage schneller beantworten zu können.

Insgesamt stellt das DNS ein leistungsfähiges, flexibles und effizientes System zur Verwaltung benutzerfreundlicher Hostnamen zur Verfügung. Es wird im gesamten Internet eingesetzt und zumindest clientseitig von jedem beliebigen Betriebssystem unterstützt. Allerdings handelt es sich nicht um die einzige Art und Weise der Namensverwaltung. Gerade in herstellerabhängigen lokalen Netzen werden Dienste wie der Windows-Namensdienst WINS oder das Network Information System (NIS) von Sun eingesetzt. Letzteres ist nicht nur ein Namens-, sondern ein einfacher Verzeichnisdienst.

[Rheinwerk Computing - Zum Seitenanfang](https://openbook.rheinwerk-verlag.de/it_handbuch/04_006.html#top)

### 4.6.5 Verschiedene Internet-Anwendungsprotokolle[Zur nächsten Überschrift](https://openbook.rheinwerk-verlag.de/it_handbuch/04_006.html)[Zur vorigen Überschrift](https://openbook.rheinwerk-verlag.de/it_handbuch/04_006.html#dodtpd5d73d94-a81f-44c8-8c80-14bbc8f9ca46)

Genau wie beinahe jede Hardware die TCP/IP-Protokolle unterstützt, laufen auf der Anwendungsschicht des Internet-Protokollstapels auch fast alle Arten von Netzwerkanwendungen. Dazu gehören unter anderem auch Anwendungsprotokolle, die ursprünglich für bestimmte herstellerabhängige Netzwerke konzipiert wurden, beispielsweise die Standard-Fileserver-Protokolle der diversen Betriebssysteme: Das unter Windows verwendete SMB-Protokoll (Server Message Blocks) wurde zunächst auf Microsofts eigenes NetBEUI-Netzwerk aufgesetzt; das von Apple konzipierte AppleShare lief ursprünglich nur unter AppleTalk. Mittlerweile können diese speziellen Anwendungsprotokolle alternativ auch auf TCP/IP aufgesetzt werden, was in der Praxis immer häufiger der Fall ist. Solche nativen Spezialprotokolle sind allerdings nicht Thema dieses Abschnitts, sondern werden am Ende des Kapitels im Abschnitt »Andere Protokollstapel« kurz angerissen.

An dieser Stelle geht es lediglich darum, die grundlegende Funktionsweise einiger typischer Internetdienste auf der Ebene ihrer Protokolle zu beschreiben. Falls Sie also Details über die Verwendung von Internet-Client-Server-Diensten oder deren Konfiguration unter einem bestimmten Betriebssystem suchen, sollten Sie [Kapitel 13](https://openbook.rheinwerk-verlag.de/it_handbuch/13_001.html#rxz1018567), »Server für Webanwendungen«, und [Kapitel 14](https://openbook.rheinwerk-verlag.de/it_handbuch/14_001.html#rxz1020525), »Weitere Internet-Serverdienste«, lesen. Hier erfahren Sie dagegen eher, was hinter den Kulissen wirklich passiert, wenn Sie eine E-Mail versenden oder eine Webseite anfordern.

Das (für Administratoren und Programmierer) Angenehme an den meisten Internet-Anwendungsprotokollen ist, dass die Protokollbefehle in Form von Klartextwörtern in Englisch verschickt werden. Wenn Sie mit einem Packet-Sniffer oder einfach mit *telnet* die Inhalte der über das Netzwerk übertragenen Datenpakete kontrollieren, können Sie deshalb unmittelbar verstehen, was die verschiedenen Hosts miteinander »reden«. Auf diese Weise ist es verhältnismäßig einfach, Konfigurations- oder Programmierfehler auf der Ebene der Anwendungsprotokolle zu entdecken und zu beseitigen.

In der Regel bestehen die Anforderungen eines Internet-Anwendungsclients aus einzeiligen Befehlen, die vom Server mit einer Statusmeldung und manchmal auch mit der Lieferung konkreter Daten beantwortet werden. Sie können das Verhalten eines Clients simulieren, indem Sie mit einem Terminal-Programm wie *telnet* eine Verbindung zu dem passenden Host und Port aufbauen und die entsprechenden Befehle von Hand eintippen.

# Telnet

Eine der ältesten Anwendungen des Internets ist die Terminal-Emulation: Ein Programm ermöglicht Ihnen über ein Terminal, das an Ihren Computer angeschlossen ist, die Arbeit an einem anderen Computer, zu dem eine Netzwerkverbindung besteht. Telnet ist eines der wichtigsten Werkzeuge für Systemadministratoren, die auf diese Weise entfernte Rechner verwalten, ohne sich physikalisch dorthin zu begeben (insbesondere an Wochenenden oder nach Feierabend schätzen Admins diese Möglichkeit, weil sie eventuelle Pannenhilfe von zu Hause aus erledigen können). Die Telnet-Spezifikation ist in RFC 854 festgelegt.

Fast alles, was an dieser Stelle über Telnet gesagt wird, gilt sinngemäß auch für SSH, die Secure Shell. Im Grunde genommen handelt es sich dabei um eine sichere Variante von Telnet, die mit Verschlüsselung arbeitet. Der gravierendste Schönheitsfehler des klassischen Telnets besteht nämlich darin, dass es Daten im Klartext überträgt – und das gilt unter anderem auch für Passwörter und ähnlich sensible Daten. SSH ist nicht in einem RFC spezifiziert, denn obwohl es sich aus frei verfügbaren Komponenten zusammensetzt, ist die ursprüngliche Implementierung kommerziell. Nähere Informationen erhalten Sie auf der Website [*http://www.ssh.com*](http://www.ssh.com/). Eine freie Implementierung, die in den meisten Linux- und anderen Unix-Systemen zum Einsatz kommt, ist OpenSSH ([*http://www.openssh.com*](http://www.openssh.com/)).

Der Telnet-Server lauscht auf dem TCP-Port 23 auf eingehende Verbindungen (SSH verwendet Port 22). Wenn ein TCP-Verbindungsversuch erfolgt, wird der Benutzer am entfernten Host zunächst nach Benutzername und Passwort gefragt, bevor er tatsächlich arbeiten kann. Im Grunde genommen wird dem jeweiligen Benutzer seine Standard-Unix-Shell zur Verfügung gestellt, an der er auch lokal auf dem entsprechenden Rechner arbeiten würde.

Telnet und SSH sind daher beliebig flexibel, was den Inhalt der in beide Richtungen übermittelten Daten angeht. Wenn Sie erst einmal mit dem Telnet-Server verbunden sind, können Sie jedes beliebige Programm auf dem entfernten Host ausführen, für das Sie Benutzerrechte besitzen. Dazu gehören auch solche Programme, die nicht zeilenorientiert, sondern mit einer Vollbildmaske arbeiten – zum Beispiel die klassischen Unix-Texteditoren *vi* und *Emacs*. Deshalb genügt die zeilenorientierte Kommunikation zwischen Client und Server bei Telnet nicht: In einem Vollbildprogramm kann jeder einzelne Tastendruck eine Bedeutung haben, die unmittelbar umgesetzt werden muss. Falls Sie Beispiele dafür sehen möchten, was Sie in einer SSH- oder Telnet-Sitzung eingeben können, lesen Sie einfach die Abschnitte über die Shell in [Kapitel 7](https://openbook.rheinwerk-verlag.de/it_handbuch/07_001.html#rxz1009500), »Linux«*.* Alles, was dort steht, gilt auch für den Fernzugriff.

Die einzige Art von Programmen, die nicht über Telnet ausgeführt werden können, sind solche, die auf einer grafischen Benutzeroberfläche laufen. Allerdings bietet die Unix-Welt auch für dieses Problem die passende Lösung: Sie benötigen auf Ihrem lokalen Rechner zusätzlich zu dem Telnet-Client einen X-Window-Server, der die grundlegenden Zeichenfunktionen für die GUI zur Verfügung stellt. Sobald dieser X-Server läuft, können Sie ein X-basiertes Anwendungsprogramm auf dem entfernten Server starten und Ihren eigenen Rechner als Ziel der grafischen Darstellung angeben (in der Regel mit dem Parameter -display). Angenommen, Ihr eigener Rechner besitzt im lokalen Netz die IP-Adresse 192.168.0.9. Dann können Sie in das Telnet-Programm, in dem eine Sitzung auf einem anderen Rechner läuft, Folgendes eingeben, um in Ihrem X-Server ein *xterm* (ein X-basiertes Terminal) zu starten:

**# xterm -display 192.168.0.9:0.0**

Der Zusatz 0.0 hinter der IP-Adresse bedeutet sinngemäß »erster X-Server, erster Bildschirm«. Wichtig ist, dass Sie den Begriff *X-Server* nicht falsch verstehen: Hier läuft der Server, der die grafische Oberfläche als Dienstleistung zur Verfügung stellt, auf Ihrem eigenen Rechner, während der Client das auf dem entfernten Rechner laufende Programm ist, dessen Ausgabe in Ihrem lokalen X-Window-System erfolgt. Näheres über die Konfiguration von X-Servern unter Unix erfahren Sie in [Kapitel 7](https://openbook.rheinwerk-verlag.de/it_handbuch/07_001.html#rxz1009500), »Linux*«*. Allerdings gibt es auch X-Server für andere Systeme, beispielsweise für Windows oder klassisches Mac OS. Sie sind natürlich nicht für die grafische Darstellung lokaler Programme gedacht (das können die eingebauten GUIs von Mac OS oder Windows selbst gut genug), sondern für grafisch orientierte Programme, die auf entfernten Unix-Rechnern laufen.

Im Übrigen sollten Sie das Telnet-Protokoll, das die Terminal-Emulation bereitstellt, nicht mit dem Unix- und Windows-Dienstprogramm *telnet* verwechseln. Letzteres kann nämlich – wie bereits erwähnt – mit fast jedem Internetserver kommunizieren, wenn Sie die passenden Parameter (IP-Adresse beziehungsweise Hostname und Portnummer beziehungsweise Standard-Dienstname) eingeben.

# FTP

Das File Transfer Protocol (FTP) ist beinahe das genaue Gegenteil von Telnet: ein aus ganz wenigen Befehlen bestehendes, klartextbasiertes, zeilenorientiertes Protokoll. Es gehört zu den frühesten Internetanwendungen überhaupt. Seine erste Definition steht in RFC 172 von 1971, die aktuelle Spezifikation befindet sich in RFC 959. Den reinen Datei-Download über FTP beherrscht heutzutage fast jeder Webbrowser; die meisten stellen auch die Verzeichnisansicht des entfernten Rechners übersichtlich und angenehm dar.

In der Praxis wird jedoch überwiegend ein grafisch orientierter FTP-Client verwendet, der dem lokalen Dateinavigator Ihres Betriebssystems idealerweise möglichst ähnlich sieht. Die häufigste Anwendung für ein solches Programm dürfte die Pflege einer eigenen Website sein. Dabei bearbeiten Sie die Daten in der Regel auf Ihrem eigenen Rechner und laden sie anschließend mithilfe eines solchen FTP-Programms auf den Server Ihres Hosting-Providers hoch, um sie zu veröffentlichen. Bekannte FTP-Clients sind beispielsweise WS\_FTP für Windows oder Fetch für Mac OS. Auch in gängige Website-Editoren wie Adobe Dreamweaver sind FTP-Module eingebaut.

Falls Sie jedoch genau sehen möchten, wie FTP-Client (auf Ihrem eigenen Rechner) und -Server (auf dem entfernten Rechner) miteinander kommunizieren, können Sie das in Unix und Windows eingebaute Konsolenprogramm *ftp* verwenden. Die Befehle, die Sie auf der Clientseite eingeben, sind jeweils einzeilig und bestehen aus einem Schlüsselwort mit eventuellen Parametern, gefolgt von einem Zeilenumbruch. Die Antwort des Servers ist zunächst eine Statusmeldung, die aus einer dreistelligen dezimalen Codenummer und einem Meldungstext besteht; häufig folgen auf die Statusmeldung zusätzliche Datenzeilen. Um dem Client das Ende einer solchen Datensequenz zu signalisieren, beginnt die letzte Zeile der Antwort des Servers wieder mit derselben Codenummer wie die erste.

Die folgenden Zeilen zeigen den Mitschnitt einer FTP-Sitzung mit dem Host [*www.lingoworld.de*](http://www.lingoworld.de/). Der Name *www* besagt natürlich, dass es sich eigentlich um einen Webserver handelt. Es ist durchaus üblich, dass Hosting-Provider den Webserver unmittelbar per FTP zugänglich machen, um die eigene Website hochzuladen:

> **ftp** **www.lingoworld.de**  
Verbunden zu www.lingoworld.de.  
220 FTP Server ready.  
Benutzer (www.lingoworld.de:(none)): **XXXXX**  
331 Password required for XXXXX.  
Kennwort: [Eingabe wird nicht angezeigt]  
230 User XXXXX logged in.  
Ftp> **pwd**  
257 "/" is current directory.  
Ftp> **cd** **extra**  
250 CWD command successful.  
Ftp> **ls**  
200 PORT command successful.  
150 Opening ASCII mode data connection for file list.  
test.txt  
info.txt  
226-Transfer complete.  
226 Quotas off  
21 Bytes empfangen in 0,01 Sekunden (2,10 KB/s)  
Ftp> **get** **test.txt**  
200 PORT command successful.  
150 Opening ASCII mode data connection for test.txt (2589 bytes).  
226 Transfer complete.  
2718 Bytes empfangen in 0,04 Sekunden (67,95 KB/s)  
Ftp> **quit**  
221 Goodbye.

In dieser Sitzung wird zunächst die Anmeldung durchgeführt (der Benutzername wird angezeigt, allerdings habe ich ihn hier geändert; die Passworteingabe hat kein grafisches Feedback), anschließend werden die folgenden Befehle eingesetzt (die ersten drei entsprechen gleichnamigen Unix-Shell-Befehlen, siehe [Kapitel 7](https://openbook.rheinwerk-verlag.de/it_handbuch/07_001.html#rxz1009500), »Linux*«*):

* pwd: *print working directory* – aktuelles Arbeitsverzeichnis ausgeben
* cd: *change directory* – Verzeichnis auf dem entfernten Server wechseln
* ls: *list* – Verzeichnisinhalt anzeigen
* get: die angegebene Datei in das aktuelle lokale Verzeichnis herunterladen
* quit: die Sitzung und das FTP-Programm beenden

Weitere wichtige Befehle sind folgende:

* put: die angegebene Datei in das aktuelle entfernte Verzeichnis hochladen
* binary: umschalten in den Binärmodus
* ascii: umschalten in den ASCII-Modus
* help: eine Liste aller verfügbaren Befehle anzeigen

Es ist wichtig, dass Sie den Unterschied zwischen dem ASCII- und dem Binärmodus verstehen. Das ganze Problem hat damit zu tun, dass die verschiedenen Betriebssystementwickler sich nicht auf einen gemeinsamen Standard für Zeilenumbrüche in Textdateien einigen konnten. Wie in [Kapitel 16](https://openbook.rheinwerk-verlag.de/it_handbuch/16_001.html#rxz1021683), »Weitere Datei- und Datenformate«, genau erläutert wird, verwendet Unix das ASCII-Zeichen mit dem Code 10 (LF, Line Feed oder Zeilenvorschub), klassisches Mac OS das ASCII-Zeichen 13 (CR, Carriage Return oder Wagenrücklauf), und Windows sowie die meisten Netzwerk-Anwendungsprotokolle benutzen beide Zeichen hintereinander.

Im ASCII-Modus werden die Zeilenumbrüche innerhalb einer Datei bei der Übertragung jeweils umgewandelt, sodass beispielsweise die auf Ihrem Windows-Rechner gespeicherten Textdateien mit CR/LF auf dem entfernten Unix-Server mit dem für dessen Verhältnisse korrekten (Nur-)LF ankommen und umgekehrt. Sie sollten jedoch begreifen, dass dieses bei Textdateien recht segensreiche Feature bei Binärdateien wie Bildern oder Programmen den sicheren Tod zur Folge hat. Wird jedes Vorkommen des Byte-Wertes 10 durch die beiden Bytes 13 und 10 ersetzt oder umgekehrt, werden die Bytes in einer solchen Datei verändert und planlos verschoben! Natürlich ist eine auf diese Weise behandelte Bild-, Audio- oder Programmdatei unbrauchbar.

Die meisten grafisch orientierten FTP-Programme entscheiden je nach Dateityp passend selbst, ob sie ASCII- oder Binär-Übertragung verwenden sollen. Bei dem Konsolen-FTP-Programm müssen Sie für jede einzelne Datei selbst in den richtigen Modus umschalten. Das ist ein – aber nicht der einzige – Grund dafür, dass die Arbeit mit der Konsolenversion von FTP in der Praxis fast unzumutbar ist.

#### E-Mail

Die E-Mail, die sich unter dem Dach eines Mailclients wie Thunderbird, Outlook Express oder Apple Mail so einheitlich präsentiert, bedarf in Wirklichkeit der Zusammenarbeit mit mindestens zwei verschiedenen Servern. Der eine ist für den Versand von E-Mails zuständig und führt zu diesem Zweck das Protokoll SMTP (Simple Mail Transport Protocol) aus. Ein anderer enthält das E-Mail-Postfach, in dem an Sie adressierte Nachrichten ankommen; dieser Dienst wird entweder von dem weitverbreiteten POP3-Protokoll (Post Office Protocol Version 3) oder dem komfortableren IMAP (Internet Message Access Protocol) versehen.

Wenn Sie eine E-Mail versenden möchten, wird diese an einen SMTP-Server übermittelt, der sich um die Weiterleitung der Nachricht an den Empfänger kümmert. SMTP, definiert in RFC 2821 (Neufassung von RFC 821), ist ähnlich wie FTP ein einfaches, textbasiertes Protokoll aus wenigen Befehlen; der zuständige Server wartet am TCP-Port 25 auf Verbindungen.

Viele SMTP-Server von Internetprovidern kontrollieren bis heute nicht die Identität des Absenders. Das Problem dabei ist, dass solche Server dadurch leicht für das Versenden von Spam verwendet werden können oder dass sogar jemand eine falsche Identität vortäuschen kann. Dabei sieht die SMTP-Spezifikation durchaus mehrere mögliche Authentifizierungsverfahren vor:

* Manche SMTP-Server überprüfen die IP-Adresse des Hosts, von dem die Verbindung initiiert wurde – ein ideales Verfahren für normale Internetprovider, die nur ihren eigenen Kunden Zugriff auf ihre SMTP-Server gewähren möchten.
* Eine andere Möglichkeit besteht darin, die Anmeldung am E-Mail-Empfangsserver desselben Providers als Voraussetzung für den E-Mail-Versand zu verlangen. Dies Verfahren wird *SMTP after POP* genannt. Nachteil: Manche E-Mail-Clients können nicht damit umgehen, sodass man jedes Mal vor dem E-Mail-Versand auf »Mail empfangen« klicken muss.
* Das sicherste Verfahren wurde erst nachträglich zu SMTP hinzugefügt (inzwischen ist es aber glücklicherweise fast flächendeckend verbreitet): die persönliche Anmeldung beim SMTP-Server mit Benutzername und Passwort.

Sie können die Kommunikation mit einem SMTP-Server über das Programm *telnet* abwickeln. Eine solche Sitzung sieht beispielsweise folgendermaßen aus (die konkreten Namens- und Adressdaten habe ich anonymisiert):

> **telnet** **smtp.myprovider.de** **smtp**  
220 smtp.myprovider.de ESMTP Fri, 13 Apr 2007 12:37:21 +0100 **HELO**  
250 smtp.myprovider.de Hello[203.51.81.17] **MAIL** **From:** **absender@myprovider.de**  
250 <absender@myprovider.de> is syntactically correct **RCPT** **To:** **empfaenger@elsewhere.com**  
250 <empfaenger@elsewhere.com> verified **DATA**  
354 Enter message, ending with "." on a line by itself **FROM:** **Sascha** **<absender@myprovider.de>  
To:** **Jack** **<empfaenger@elsewhere.com>  
Subject:** **Gruesse**  
 **Hallo** **Jack,  
hier** **ist** **wieder** **einmal** **Post** **für** **dich.  
Viel** **Spaß** **damit!  
Gruss,** **Sascha  
.**  
250 OK id=18QdIY-00048Y-00 **QUIT**  
221 smtp.myprovider.de closing connection.

In dieser kurzen Konversation werden die folgenden SMTP-Befehle verwendet:

* HELO: Mit diesem Befehl meldet sich der Client beim Server an; eventuell findet in diesem Zusammenhang die bereits beschriebene Überprüfung der Client-IP-Adresse statt. Manche SMTP-Server verlangen auch die Angabe eines Domainnamens hinter dem Befehl.
* MAIL: Dieser Befehl leitet die Erzeugung einer neuen Nachricht ein; der Empfänger muss im Format From: E-Mail-Adresse oder From: Name <E-Mail-Adresse> angegeben werden.
* RCPT: Gibt einen Empfänger im Format To: E-Mail-Adresse oder To: Name <E-Mail-Adresse> an.
* DATA: Alle folgenden Zeilen des Clients werden als Teil der eigentlichen E-Mail-Nachricht aufgefasst, bis eine Zeile folgt, die nur einen Punkt (.) enthält.
* QUIT: Die Sitzung wird hiermit beendet; alle bis zu diesem Zeitpunkt erzeugten E-Mail-Nachrichten werden versandt.

Die E-Mail-Nachricht selbst (zwischen DATA und der Abschlusszeile mit dem Punkt) ist eine klassische Textnachricht, deren Aufbau in RFC 822 (aktualisiert in RFC 2822) beschrieben wird. Prinzipiell besteht sie aus mehreren Header-Zeilen im Format Feldname: Wert, gefolgt von einer Leerzeile und dem eigentlichen Text. Der minimale Header enthält den Absender (From), den Empfänger (To) und einen Betreff (Subject). Absender und Empfänger dürfen wie bei den SMTP-Befehlen MAIL und RCPT diverse Formate besitzen. Weitere häufige Header-Felder sind die Kopien-Empfänger (Cc für »Carbon Copy«) sowie die unsichtbaren Kopien-Empfänger (Bcc für »Blind Carbon Copy«). Die normalen Kopien-Empfänger werden in der Nachricht angezeigt, die unsichtbaren nicht.

Ein alternatives Format für E-Mails, das heutzutage bereits häufiger verwendet wird als RFC 822, ist das MIME-Format. Die verschiedenen Aspekte von MIME werden in RFC 2045 bis 2049 dargelegt. Die Abkürzung MIME steht für »Multipurpose Internet Mail Extensions«. Es handelt sich um ein Format, das für den Versand beliebiger Text- und Binärdaten geeignet ist, sogar von verschiedenen Datentypen innerhalb derselben Nachricht.

Der MIME-Header ist eine Erweiterung des RFC-822-Headers. Die wichtigsten neuen Felder sind Content-type, das den Datentyp angibt, und Content-Transfer-Encoding, mit dessen Hilfe das Datenformat festgelegt wird. Ersteres beschreibt also den Inhalt der Nachricht, Letzteres die Form, in der sie versandt wird. Der Inhaltstyp (Content-Type), meist *MIME-Type* genannt, besteht aus zwei Bestandteilen, die durch einen Slash (/) voneinander getrennt werden: dem Haupttyp und dem genaueren Untertyp. Haupttypen sind beispielsweise text (ASCII-Text), image (Bilddaten), audio (Audiodaten), video (Digitalvideo) oder application (proprietäres Datenformat eines bestimmten Anwendungsprogramms). [Tabelle 4.23](https://openbook.rheinwerk-verlag.de/it_handbuch/04_006.html#rxz1006823) listet einige gängige MIME-Types auf. Die vollständige Liste aller registrierten Typen finden Sie online unter [*http://www.iana.org/assignments/media-types/index.html*](http://www.iana.org/assignments/media-types/index.html).

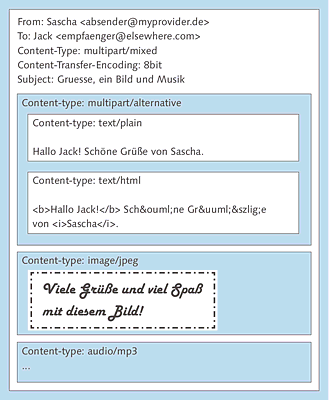
|  |  |
| --- | --- |
| **Tabelle 4.23**Einige gängige MIME-Datentypen | |
| **Typ** | **Beschreibung** |
| text/plain | reiner Text ohne Formatierungsbefehle |
| text/html | HTML-Code |
| text/xml | XML-Code |
| image/gif | Bild vom Dateityp GIF |
| image/jpeg | Bild vom Dateityp JPEG |
| image/png | Bild vom Dateityp PNG |
| audio/wav | Sounddatei vom Typ Microsoft Wave |
| audio/aiff | Sounddatei vom Typ Apple AIFF |
| audio/mpeg | komprimierte Sounddatei vom Typ MP3 |
| video/avi | Digitalvideo vom Typ Microsoft Video for Windows |
| video/mov | Digitalvideo vom Typ Apple QuickTime |
| video/mpeg | Digitalvideo vom Typ MPEG |
| application/ x-shockwave-flash | komprimierter Adobe-Flash-Film (Dateiendung*.swf*) |
| application/ x-director | komprimierter Adobe-Director-Film (Dateiendung *.dcr*) |
| application/ x-www-form-urlencoded | POST-Forumlardaten bei HTTP-Anfragen an Webserver (Näheres siehe [Kapitel 13](https://openbook.rheinwerk-verlag.de/it_handbuch/13_001.html#rxz1018567), 17 und 18) |
| multipart/mixed | »Umschlag« für mehrere MIME-Unterabschnitte |
| multipart/alternative | »Umschlag« für denselben Inhalt in mehreren Alternativformaten |
| multipart/form-data | »Umschlag« für POST-Formulardaten einschließlich Datei-Uploads (siehe [Kapitel 18](https://openbook.rheinwerk-verlag.de/it_handbuch/18_001.html#rxz1024475)) |

Die drei letzten Typen in der Tabelle machen MIME besonders interessant: Ein MIME-Dokument vom Typ multipart/mixed kann beliebig viele Teile enthalten, die jeweils einen vollständigen MIME-Header besitzen und wiederum beliebige MIME-Types aufweisen können. Mithilfe dieser Technik werden in modernen E-Mail-Programmen Attachments (Dateianhänge) der Mail hinzugefügt. Dagegen wird ein Abschnitt vom Typ multipart/alternative eingesetzt, um denselben Inhalt in verschiedenen alternativen Darstellungsformen zu umschließen, beispielsweise ein Bild im GIF- und im PNG-Format oder (wahrscheinlich die häufigste Anwendung) den Text einer E-Mail-Nachricht einmal im einfachen Text- und einmal im HTML-Format. zeigt beispielhaft, wie eine MIME-Nachricht mit zwei Dateianhängen aufgebaut sein könnte.

Dasselbe Verfahren – in diesem Fall mit dem MIME-Type multipart/form-data – kommt bei Webformularen zum Einsatz, wenn diese außer gewöhnlichen Auswahl- oder Eingabefeldern auch Datei-Uploads unterstützen.

Der Content-Transfer-Encoding-Header gibt dem Empfängerclient einen Hinweis, auf welche Weise die ankommenden Daten zu interpretieren sind. Häufig verwendete Werte sind etwa folgende:

* 7bit: Keine Codierung; eignet sich für 7-Bit-ASCII (englischer Text). Automatischer Zeilenumbruch nach spätestens 1.000 Zeichen.
* 8bit: Keine Codierung; eignet sich für 8-Bit-Text (internationaler Text). Ebenfalls automatischer Zeilenumbruch nach spätestens 1.000 Zeichen.
* binary: Keine Codierung, es erfolgt kein automatischer Zeilenumbruch.
* quoted-printable: Spezielle Codierung von Sonderzeichen, die über 7-Bit-ASCII hinausgehen. Beispiel: »größer« wird zu »gr=FC=DFer« (die Codierung besteht aus einem Gleichheitszeichen, gefolgt von hexadezimalem Zeichencode).
* base64: Bevorzugte Codierung für Binärdateien. Ein spezieller Algorithmus packt die Daten 7-Bit-kompatibel um. Dieses Format ist auch dann nicht von Menschen lesbar, wenn Klartext codiert wird – aus »Hallo Welt!« wird beispielsweise SGFsbG8gV2VsdCE=.

[](https://openbook.rheinwerk-verlag.de/it_handbuch/bilder/04_005.png)

**Abbildung 4.5**Beispiel für eine E-Mail im MIME-Multipart-Format

Die Codierungsformen quoted-printable und base64 besitzen den Vorteil, dass die Mailnachricht formal kompatibel mit RFC 822 bleibt und entsprechend auch über alte Mailserver versandt und empfangen werden kann.

Der E-Mail-Empfang über einen POP3-Server erfolgt auf textbasierte Art, ähnlich wie bei SMTP. Der Server kommuniziert über den TCP-Port 110. Die Beschreibung von POP3 steht in RFC 1939. Zur Verdeutlichung hier wiederum eine Telnet-basierte Konversation mit einem (unkenntlich gemachten) POP3-Server:

# **telnet** **pop.myprovider.de** **pop3**  
+OK POP3 server ready **USER** **absender**  
+OK **PASS** **XXXXX**  
+OK **LIST**  
1898  
. **RETR** **1**  
+OK 953 octets  
Return-path: <empfanger@elsewhere.com>  
Envelope-to: absender@myprovider.de  
Delivery-date: Tue, 24 Dec 2002 03:08:24 +0100  
Received: from [207.18.31.76] (helo=smtp.elsewhere.com)  
 by mxng13.myprovider.de with esmtp (Exim 3.35 #1)  
 id 18QeUU-00027O-00  
 for absender@myprovider.de; Tue, 24 Dec 2002  
 03:08:18 +0100  
Received: from box (xdsl-202-21-109-17.elsewhere.com  
[202.21.109.17])  
by smtp.elsewhere.com (Postfix) with SMTP id  
CA500866C1  
for <absender@myprovider.de>; Tue, 24 Dec 2002  
03:08:14 +0100 (MET)  
Message-ID: <001901c2aaf2$31ce81e0$0200a8c0@box>  
From: "Jack" <empfaenger@elsewhere.com>  
To: "Sascha" <absender@provider.de>  
Subject: Gruesse  
Date: Tue, 24 Dec 2002 03:14:30 +0100  
MIME-Version: 1.0  
Content-Type: text/plain;  
charset="utf-8"  
Content-Transfer-Encoding: quoted-printable  
  
Hi!  
Wie geht's?  
Alles klar?  
Ciao.  
. **DELE** **1**  
+OK **QUIT**  
+OK

In dieser Sitzung kommen die folgenden POP3-Befehle zum Einsatz:

* USER: Angabe des Benutzernamens für die Anmeldung
* PASS: Angabe des Passworts für die Anmeldung
* LIST: nummerierte Liste der verfügbaren E-Mails mit der jeweiligen Länge in Byte
* RETR: E-Mail mit der angegebenen Nummer empfangen
* DELE: E-Mail mit der angegebenen Nummer vom Server löschen
* QUIT: Sitzung beenden

Die meisten E-Mail-Programme führen RETR und DELE standardmäßig unmittelbar nacheinander durch, die Nachrichten verbleiben also in der Regel nicht auf dem Server. Bei IMAP-Servern ist es dagegen meist anders: Der besondere Vorteil des IMAP-Protokolls besteht darin, dass auf dem Mailserver selbst verschiedene Ordner eingerichtet werden können, um Mails dort zu verwalten. Auf diese Weise erleichtert IMAP die E-Mail-Verwaltung für mobile Benutzer. Die aktuelle Version von IMAP ist das in RFC 2060 dargestellte IMAP4. Ein IMAP-Server funktioniert ähnlich wie ein POP3-Server, verwendet allerdings den TCP-Port 142.

Eine weitere beliebte Form der E-Mail-Nutzung sind webbasierte Freemail-Dienste wie GMX oder Hotmail. Dabei handelt es sich um gewöhnliche POP-SMTP-Kombinationen, die über eine Website mit persönlicher Anmeldung zugänglich gemacht werden. Das Programm, das mit den E-Mail-Servern kommuniziert, läuft auf dem Webserver und wird dem Kunden per Browser zur Verfügung gestellt.

#### Newsgroups

Newsgroups als virtuelle »Schwarze Bretter« wurden 1979 eingeführt, um Gruppendiskussionen zwischen der Duke University und der University of North Carolina zu ermöglichen. Das System entwickelte sich im Laufe der Jahre zum weltweiten *Usenet* mit mehreren Zehntausend Newsgroups.

Das Usenet besteht aus einem losen Verbund von weltweit verteilten Newsservern, die das in RFC 977 festgelegte Protokoll NNTP (Network News Transport Protocol) ausführen. Wer einen Artikel in einer Newsgroup veröffentlichen möchte, sendet diesen an den TCP-Port 119 des nächstgelegenen Newsservers (in der Regel den des eigenen Providers); innerhalb von spätestens 24 Stunden dürfte die Nachricht jeden Newsserver weltweit, der die betreffende Newsgroup bereitstellt, erreicht haben.

Um Nachrichten in einer bestimmten Newsgroup zu lesen, müssen Sie diese abonnieren: Sie verbinden sich über Ihren Newsreader mit einem Newsserver, der diese Group anbietet, und aktivieren das Abonnement für die Group. Bei jedem Start Ihres Newsreaders werden nun zunächst die Header-Daten aller aktuellen Nachrichten in der Group heruntergeladen und angezeigt. Sobald Sie eine solche Kopfzeile anklicken, wird der eigentliche Inhalt der Nachricht geladen.

Eine Newsgroup-Nachricht ist ein RFC-822-kompatibles Dokument. Allerdings definiert das NNTP-Protokoll einige spezielle Header-Felder, die erforderlich sind, um die Nachrichten den verschiedenen Groups zuzuordnen und ihre Position in einem Thread, einem Diskussionsstrang, festzulegen. Die meisten Newsserver beherrschen im Übrigen die MIME-Erweiterungen, allerdings sind MIME-basierte HTML- oder gar Multimedia-Nachrichten in traditionellen Usenet-Newsgroups verpönt; es ist üblich, nur reinen Text zu verwenden.

Zu einer Newsgroup-Nachricht gehören die folgenden wichtigen Header-Felder (abgesehen von denjenigen, die bereits beim Thema SMTP beschrieben wurden):

* Article: Eine ID des Beitrags, bestehend aus einer Nummer und dem Namen der Newsgroup, in die der Beitrag gepostet wird. Diese Nummern können auf verschiedenen Newsservern unterschiedlich sein, da sie in der Reihenfolge vergeben werden, in der Nachrichten eintreffen.
* Message-ID: Eine unveränderliche und weltweit einmalige ID für diesen einen Beitrag über alle Newsgroups hinweg. Ermöglicht das quellgenaue Zitieren und Verlinken eines Postings.
* Referrers: Die Message-ID des ursprünglichen Newsgroup-Beitrags, auf den geantwortet wurde. Anhand dieses Felds wird die Nachricht in einen Thread einsortiert.

Das Usenet besteht aus einer Reihe von Newsgroups mit hierarchisch gegliederten Namen. Ganz links steht dabei die allgemeine Oberkategorie, die nach rechts immer weiter spezialisiert wird (ähnlich wie in einem Dateisystem und andersherum als bei Domainnamen). Die Hauptkategorien sind beispielsweise *comp* für computerbezogene Themen, *rec* (*recreation*) für Freizeit, Sport und Spiel, *soc* für gesellschaftspolitische Themen oder *sci* (*science*) für die Welt der Naturwissenschaft und Technik. Innerhalb dieser Kategorien bestehen einzelne Groups wie *comp.lang.perl.modules* (Computer – Programmiersprachen – Perl – Module), *rec.autos.vw* (Freizeit – Autos – Volkswagen), *soc.religion.islam* (Gesellschaft – Religion – Islam) oder *sci.crypt.random-numbers* (Wissenschaft – Kryptografie – Zufallsgeneratoren). Im Übrigen gibt es Hauptkategorien, die auf ein bestimmtes Länderkürzel lauten, für Newsgroups, in denen eine bestimmte Sprache gesprochen wird (etwa die *de.\**-Hierarchie für deutschsprachige Groups).

Da das traditionelle Usenet recht schwerfällig und konservativ ist, bedarf es beinahe endloser Diskussionen, bevor unter einer der klassischen Hauptkategorien eine neue Newsgroup eingerichtet wird. Aus diesem Grund wurde die spezielle *alt.\**-Hierarchie eingeführt, unter der jeder neue Groups anlegen kann. Diese Hierarchie gehört nicht zum eigentlichen Usenet und enthält die bizarrsten, aber auch einige der interessantesten Newsgroups.

Die Beliebtheit der Newsgroups unter den Internetnutzern scheint allerdings bereits vor einigen Jahren ihren Zenit überschritten zu haben. Die größte Konkurrenz bilden heutzutage webbasierte Foren, in denen über speziellere Themen diskutiert wird und die nicht ganz so strenge Verhaltensregeln besitzen wie die Newsgroups oder insbesondere das klassische Usenet. Dennoch besteht die Möglichkeit, ohne spezielles News-Programm auf beliebige Newsgroups zuzugreifen: Der Suchmaschinenbetreiber Google kaufte vor einigen Jahren den webbasierten News-Dienst Deja.com und dessen Usenet-Archiv auf. Unter [*http://groups.google.com*](http://groups.google.com/) können Sie in fast allen jemals geschriebenen Newsgroup-Beiträgen recherchieren und sich darüber hinaus anmelden, um aktiv an Newsgroup-Diskussionen teilzunehmen.

#### Das World Wide Web

Das Web ist heute die dominierende Internetanwendung überhaupt, und zwar in einem solchen Maße, dass viele Leute das WWW mit dem gesamten Internet gleichsetzen. Wer auf das Web zugreifen möchte, verwendet dazu eine spezielle Clientsoftware, den sogenannten *Webbrowser*. Nach der Eingabe einer Dokumentadresse stellt der Browser eine TCP-Verbindung zu dem gewünschten Webserver her und fordert über das HTTP-Protokoll das gewünschte Dokument an. Das Dokument ist üblicherweise in der Seitenbeschreibungssprache HTML verfasst (unterstützt durch CSS und JavaScript), die der Browser interpretiert und in eine auf bestimmte Art und Weise formatierte Webseite umwandelt.

Eine solche Seite kann außerdem Verweise auf eingebettete Dateien wie Bilder oder Multimedia enthalten, die der Browser auf dieselbe Art anfordert wie das HTML-Dokument selbst und an der passenden Stelle auf der Seite platziert. Ein weiteres wichtiges Element von Webseiten sind die Hyperlinks, anklickbare Verknüpfungen zu anderen Dokumenten. Wenn Sie einen Hyperlink aktivieren, wird die entsprechende Datei angefordert und in den Browser geladen.

Damit das Web funktionieren kann, wirken einige wesentliche Konzepte zusammen:

* Das Anwendungsprotokoll HTTP, über das Dokumente beim Server angefordert und von diesem ausgeliefert werden. Die aktuelle Version des Protokolls, HTTP 1.1, wird in RFC 2616 beschrieben; in [Kapitel 13](https://openbook.rheinwerk-verlag.de/it_handbuch/13_001.html#rxz1018567), »Server für Webanwendungen«, erhalten Sie genauere Informationen darüber.
* Ein spezielles Format für Dokumentadressen, das als *Uniform Resource Locator* (URL) bezeichnet wird und dessen Definition sich in RFC 1738 befindet. Die URL wird beispielsweise in die Adresszeile des Browsers eingegeben; sie sieht beispielsweise so aus: [*http://www.galileocomputing.com/*](http://www.galileocomputing.com/).
* Die Seitenbeschreibungssprache HTML, in der Hypertext-Dokumente für das WWW geschrieben werden. Neuere HTML-Versionen werden nicht mehr in RFCs definiert; eine genaue Beschreibung von HTML finden Sie in [Kapitel 17](https://openbook.rheinwerk-verlag.de/it_handbuch/17_001.html#rxz1022720), »Webseitenerstellung mit (X)HTML und CSS«.

HTTP ist ein ähnlich einfaches, klartextbasiertes Protokoll wie etwa FTP. Ein Webserver wartet üblicherweise auf dem TCP-Port 80 auf Verbindungen. Der folgende Mitschnitt einer Dokumentanforderung über *telnet* zeigt einige wesentliche Bestandteile:

> **telnet** **www.galileocomputing.de** **http  
GET** **/** **HTTP/1.1  
Host:** **www.galileocomputing.de  
  
HTTP/1.1** **200** **OK  
Date:** **Tue,** **14** **Apr** **2009** **05:06:43** **GMT  
Server:** **Zope/(Zope** **2.7.6-final,** **python** **2.3.5,** **linux2)** **ZServer/1.1  
Content-Length:** **7730  
Accept-Ranges:** **none  
Last-Modified:** **Sat,** **18** **Jan** **2008** **17:16:04** **GMT  
Content-Type:** **text/html  
Cache-Control:** **no-store,** **no-cache,** **must-revalidate,** **post-check=0,  
pre-check=0  
Expires:** **Sun,** **19** **Apr** **2009** **05:06:43** **GMT  
  
<!DOCTYPE** **HTML** **PUBLIC** **"-//W3C//DTD** **HTML** **4.0** **Transitional//EN"  
"http://www.w3.org/TR/REC-html40/loose.dtd">  
<html** **lang="de">  
[...]**

Der wichtigste HTTP-Client-Befehl, mit dessen Hilfe hier das Dokument angefordert wird, lautet GET. Die Parameter sind der Dokument-Pfad (hier das Wurzeldokument /) und die HTTP-Protokoll-Version. Anschließend kann der Client beliebig viele Header im Format RFC-822-Format Feldname: Wert senden; der einzige Pflicht-Header ist bei HTTP 1.1 Host, weil auf einem Server mit ein und derselben IP-Adresse unterschiedliche Websites mit eigenen Hostnamen liegen können (sogenannte *virtuelle Hosts*). Abgeschlossen wird die Anfrage durch einen doppelten Zeilenumbruch.

Die Antwort des Servers besteht ebenfalls aus der HTTP-Version, darauf folgen eine Statuscode-Nummer wie bei FTP und ein entsprechender Meldungstext. In diesem Fall bedeutet 200 OK, dass das Dokument gefunden wurde und nun übertragen wird (im vorigen Beispiel wurde es aus Platzgründen nach der zweiten Zeile abgebrochen).

Die eigentliche Datenübertragung besteht aus einer Reihe von Header-Feldern, darauf folgen eine Leerzeile und dann der eigentliche Dokumentinhalt.

Das einzige unbedingt erforderliche Header-Feld ist Content-type – wie bei E-Mails handelt es sich um die MIME-Formatangabe des Dokumentinhalts. Der Webserver ermittelt den Wert, den er hier eintragen muss, anhand der jeweiligen Dateiendung. Der Browser interpretiert den Wert dieses Felds und nicht etwa die Dateiendung des übermittelten Dokuments. Daher toleriert der Browser auch dynamisch erzeugte Dokumente mit Dateiendungen wie *.php*, *.pl* oder *.jsp* als HTML. Ein Browser benötigt diese Information dringend, damit er weiß, auf welche Weise er die ankommenden Daten interpretieren soll.

Die aktuelle Protokollversion HTTP 1.1 benötigt zusätzlich den Header Host: Da auf einem Rechner unterschiedliche Webdomains betrieben werden können, muss angegeben werden, welche Website der Client anfordert.

Ein weiterer HTTP-Client-Befehl ist beispielsweise POST, durch den Benutzereingaben aus HTML-Formularen an den Server übertragen werden; in diesem Fall enthält die Anfrage nach dem Header und der Leerzeile einen Body mit den Daten, genau wie die meisten HTTP-Antworten. Eine andere nützliche Methode ist aber auch HEAD, mit deren Hilfe nur die Header-Daten ohne das eigentliche Dokument angefordert werden (zum Beispiel, damit der Browser überprüfen kann, ob er bereits die aktuelle Version eines zum wiederholten Mal geladenen Dokuments in seinem Cache gespeichert hat).

Einige wichtige Statuscodes, die der Server zurückgeben kann, werden in [Tabelle 4.24](https://openbook.rheinwerk-verlag.de/it_handbuch/04_006.html#rxz1006985) gezeigt.

|  |  |  |
| --- | --- | --- |
| **Tabelle 4.24**Einige wichtige HTTP-Statuscodes | | |
| **Code** | **Meldungstext** | **Beschreibung** |
| 200 | OK | Dokument gefunden; wird geliefert |
| 204 | No Content | Anfrage akzeptiert, kein Dateninhalt |
| 301 | Moved Permanently | Die Dateien befinden sich unter einer neuen URL, die mitgeliefert wird. |
| 304 | Not Modified | Seit dem letzten Aufruf wurde der Dokumentinhalt nicht geändert. Der Browser oder der Proxy kann seine lokale Kopie verwenden. |
| 401 | Unauthorized | Es ist eine persönliche Anmeldung mit Benutzername und Kennwort erforderlich. |
| 403 | Forbidden | Zugriff verweigert; eine persönliche Anmeldung nützt auch nichts. |
| 404 | Not Found | Dokument existiert nicht; meist Folge eines fehlerhaften oder veralteten Links. |
| 500 | Internal Server Error | Ein serverseitiges Skript enthält einen schwerwiegenden Fehler. |

Um die Anfrage, die zuvor per telnet manuell erstellt wurde, von einem Browser durchführen zu lassen, müssen Sie die URL [*https://www.galileo-press.de*](https://www.rheinwerk-verlag.de/?GPP=openbook) in sein Adressfeld eintippen. Der Browser stellt eine Verbindung mit dem Standard-HTTP-Port des Hosts *www.galileocomputing.de* her und fordert, da kein konkreter Pfad angegeben wurde, das Wurzeldokument */* an. Damit der Webserver auch dann ein Dokument ausliefern kann, wenn der Browser lediglich einen Verzeichnis-, aber keinen Dateinamen übermittelt hat, verwenden Webserver das Konzept des Standard- oder Index-Dokuments: Ein Dokument im angeforderten Verzeichnis, das einen speziellen Dateinamen besitzt, wird automatisch ausgeliefert. Der häufigste Name für ein solches Dokument ist *index.html* oder *index.htm*.

Allgemein setzt sich eine URL aus dem Zugriffsverfahren beziehungsweise Anwendungsprotokoll, dem Hostnamen oder der IP-Adresse sowie dem Pfad des gewünschten Dokuments zusammen. Die Verwendung ist nicht auf HTTP beschränkt; es gibt beispielsweise auch FTP-URLs wie *ftp://ftp.uni-koeln.de*. Daneben existieren auch speziellere URL-Schemata wie diejenigen für E-Mail-Verweise in der Form mailto: E-Mail-Empfänger oder für Newsgroups, zum Beispiel *news:comp.lang.java.beans*. Solche besonderen URLs werden häufig als Link in HTML-Dokumenten verwendet. Die meisten Browser können entweder selbst damit umgehen, oder es ist ein Zusatzprogramm konfiguriert, das automatisch geöffnet wird, wenn man einen solchen Link anklickt.

Die konkrete Einrichtung eines Webservers (des Marktführers Apache) wird in [Kapitel 13](https://openbook.rheinwerk-verlag.de/it_handbuch/13_001.html#rxz1018567), »Server für Webanwendungen«, besprochen; dort erfahren Sie auch noch mehr Details zu HTTP. Die Erstellung von Webanwendungen mit HTML, CSS, PHP, Ruby on Rails, JavaScript und Ajax kommt dagegen in den Kapiteln 17 bis 19 zur Sprache.