

TCP/IP

Grundlagen, Adressierung, Subnetting

von
Dirk Jarzyna

1. Auflage

TCP/IP – Jarzyna

schnell und portofrei erhältlich bei beck-shop.de DIE FACHBUCHHANDLUNG

Thematische Gliederung:

Netzwerkprotokolle, EDI – Netzwerkprotokolle, EDI

mitp/bhv 2013

Verlag C.H. Beck im Internet:

www.beck.de

ISBN 978 3 8266 9499 8



Dirk
Jarzyna

TCP/IP

Grundlagen, Adressierung, Subnetting

Das TCP/IP- und OSI-Netzwerkmodell

Es existiert heute kaum ein Computer, der die TCP/IP genannte Netzwerkprotokollsammlung nicht unterstützt. Das liegt daran, dass heute so gut wie jeder Computer mit dem Internet verbunden ist und deshalb gar nicht darum herum kommt. Jedes Betriebssystem, ob Windows, Linux oder Unix, unterstützt TCP/IP. Selbst die sogenannten digitalen Assistenten (PDAs) und neueren Mobiltelefone unterstützen TCP/IP. Netzwerk-Switches unterstützen TCP/IP und Router natürlich ebenfalls, denn sonst könnten sie ihre Aufgabe, Daten über lokale Netzwerke und das Internet an den richtigen Adressaten weiterzuleiten, gar nicht erledigen.

So einfach war es nicht immer. Es ist noch nicht besonders lange her, da gab es keine Netzwerkprotokolle – auch kein TCP/IP. Computerhersteller erfanden die ersten Netzwerkprotokolle, die aber zunächst nur die Systeme genau dieses Herstellers unterstützten. Die Details der Implementierung wurden als Geheimnis gehütet. Irgendwann erkannten die Hersteller aber die Notwendigkeit, ihre Computer auch mit Computern und Geräten anderer Hersteller kommunizieren zu lassen, und veröffentlichten ihre Netzwerkprotokolle. IBM veröffentlichte beispielsweise 1974 ihr Netzwerkmodell *Systems Network Architecture* (SNA). Daraufhin entwickelten andere Hersteller Produkte, mit deren Hilfe ihre Computer über SNA mit den Computern von IBM kommunizieren konnten. Das funktionierte tadellos, hatte aber unter anderem den Nachteil, dass die großen Hersteller sagen konnten, wo es im Netzwerkmarkt langgeht. Dieses Problem ist noch immer nicht so ganz gelöst ...

Eine bessere Lösung war es jedenfalls, ein offenes standardisiertes Netzwerkmodell zu schaffen, das alle Hersteller unterstützen. In den später 1970er Jahren nahm sich die *International Organization for Standardiza-*

tion (ISO) dieser Aufgabe an und begann, an etwas zu arbeiten, das wir heute als *Open-Systems-Interconnection*- oder *OSI-Netzwerkmodell* kennen. Das Ziel des OSI-Modells war von Anfang an, Netzwerkprotokolle zu standardisieren, um die Kommunikation zwischen allen Computern auf der Welt zu ermöglichen.

Dem US-Verteidigungsministerium verdanken wir nicht nur Patriot-Raketen sondern auch ein zweites standardisiertes offenes Netzwerkmodell. Verschiedene amerikanische Universitäten entwickelten (freiwillig) im Auftrag des Ministeriums Netzwerkprotokolle. Diese Arbeit resultierte in ein konkurrierendes Netzwerkmodell mit dem Namen TCP/IP.

Ende der 1980er Jahre gab es viele konkurrierende proprietäre Netzwerkmodelle, darunter beispielsweise auch Novells IPX/SPX, und zwei konkurrierende standardisierte Netzwerkmodelle (OSI und TCP/IP). Was passierte, wissen wir: Am Ende setzte sich TCP/IP durch, nicht nur unter den zwei standardisierten Modellen, sondern es verdrängte auch viele der proprietären Protokolle.

Dieses Kapitel liefert die Grundlagen zu TCP/IP. Es beschreibt, was das TCP/IP-Netzwerkmodell ist und wie es funktioniert. Da in Verbindung mit TCP/IP immer wieder Begriffe auftauchen, die sich auf OSI beziehen, ist es notwendig, auch kurz auf das OSI-Modell einzugehen.

1.1 Die TCP/IP-Architektur

TCP/IP definiert eine große Anzahl von Protokollen, die Computern erlauben, miteinander zu kommunizieren. Allerdings sind es nur einige wenige Protokolle, die tatsächlich als »Hauptprotokolle« betrachtet werden. Von diesen wenigen Schlüsselprotokollen gelten zwei Protokolle als die wichtigsten: Das *Internet Protocol* (IP) übernimmt die Adressierung, das Datagram-Routing und weitere Funktionen in einem Internetwork. Das *Transmission Control Protocol* (TCP) ist das primäre Protokoll der Transportschicht; es ist verantwortlich für den Verbindungsaufbau und das Verbindungsmanagement und sorgt für einen zuverlässigen Datentransport zwischen Softwareprozessen auf Geräten. Die Details aller Protokolle der TCP/IP-Suite sind in Dokumenten beschrieben, die *Request*

for Comments (RFCs) genannt werden. Wer die in den TCP/IP-RFCs beschriebenen Protokolle in einen Computer implementiert, kann relativ sicher sein, dass dieser Computer mit anderen Computern kommunizieren kann, die ebenfalls TCP/IP implementiert haben.

Wie andere Netzwerkarchitekturen verteilt TCP/IP die verschiedenen Protokolle auf unterschiedliche Schichten oder Layers eines Architektur- oder Schichtenmodells. Ein solches Modell hilft, die einzelnen Komponenten und deren Funktionen zu beschreiben. Als klassisches Schichtenmodell wird zwar das 1979 definierte OSI-Modell angesehen. *Protokollschichtenkonzepte* existierten allerdings schon lange, bevor sie durch das OSI-Modell formalisiert wurden. Ein Beispiel dafür ist eben die TCP/IP-Protokollarchitektur. Da TCP/IP historisch eng mit dem *Department of Defence* (US-Verteidigungsministerium) verknüpft ist, wird das TCP/IP-Schichtenmodell auch als *DoD-Modell* bezeichnet.

Tabelle 1.1 zeigt die Hauptkategorien des TCP/IP-Architekturmodells.

| TCP/IP-Layer (Schicht) | Beispielprotokolle |
|-------------------------------------|-------------------------------|
| Application (Anwendungsschicht) | HTTP, POP3, SMTP, FTP, Telnet |
| Transport (Transportschicht) | TCP, UDP |
| Internet (Internetschicht) | IP (IPv4 und IPv6) |
| Network Access (Netzzugangsschicht) | Ethernet, Token-Ring, FDDI |

Tabelle 1.1: Das TCP/IP-Architekturmodell

Hinweis

Für die Bezeichnungen der einzelnen Schichten sowohl im TCP/IP- als auch im OSI-Schichtenmodell gibt es deutsche Begriffe. In Tabelle 1.1 sehen Sie die deutschen Begriffe in Klammern hinter den englischen Originalbegriffen. Es empfiehlt sich, beide Begriffe zu lernen, weil der größte Teil der Dokumentation zu TCP/IP und anderen Netzwerkthemen in englischer Sprache vorliegt und IT-Profis selbst in Deutschland häufig die englischen Originalbegriffe bevorzugen (und die deutschen Begriffe manchmal noch nicht einmal kennen – ich selbst muss sie auch immer wieder nachschlagen).

Tabelle 1.1 zeigt die vier Schichten des TCP/IP-Modells und nennt für jede Schicht beispielhaft ein paar populäre Protokolle, die eben auf der jeweiligen Schicht angesiedelt sind. Die folgenden Abschnitte beschreiben jede der vier Schichten und ihr Zusammenspiel genauer.

1.1.1 Die TCP/IP-Anwendungsschicht

Gleich das Wichtigste vorweg: Die *TCP/IP-Anwendungsschicht* definiert nicht die Anwendung selbst, sondern Dienste, die von Anwendungen benötigt werden. Das kann im Fall von HTTP beispielsweise die Fähigkeit sein, eine Datei zu übertragen. Die TCP/IP-Anwendungsschicht bietet also der auf einem Computer laufenden Anwendungssoftware Dienste. Sie bildet die Schnittstelle zwischen der Software auf dem Computer und dem Netzwerk.

Die heute populärste TCP/IP-Anwendung ist ohne Zweifel der Web-Browser. Einen Web-Browser zu benutzen ist so einfach wie Kaugummi kauen: Sie starten den Web-Browser auf Ihrem Computer und tippen den Namen der gewünschten Website ein, die daraufhin – falls nichts schiefgeht – erscheint. Hinter den Kulissen läuft dabei natürlich einiges ab.

Nehmen wir einmal an, Peter öffnet seinen Browser, der bequemerweise so konfiguriert ist, dass er automatisch die Homepage des Web-Servers Webby lädt.

Um die Webseite von Webby zu bekommen, sendet Peter einen *HTTP-Header* zu Webby. Dieser *Header* enthält den Befehl »get«, um eine Datei abzurufen. Normalerweise enthält diese Anfrage auch noch den Namen der Datei. Wird kein Name angegeben, nimmt der Web-Server an, dass die Default-Webseite gewünscht ist. Damit liegt er in der Regel richtig.

Hinweis

In deutschsprachiger Literatur über Netzwerkprotokolle liest man statt Header gelegentlich Kopf, beispielsweise statt Nachrichten-Header Nachrichtenkopf. Obwohl dies ein Buch in deutscher Sprache ist, möchte ich doch lieber beim englischen Originalbegriff Header bleiben.

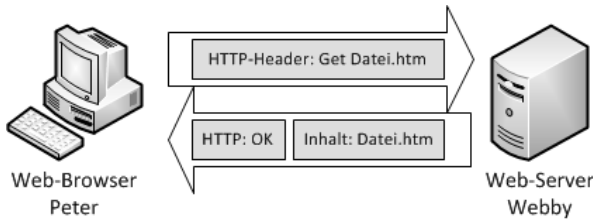


Abb. 1.1: Eine HTTP-Get-Anfrage und die HTTP-Antwort

Die Antwort des Web-Servers enthält ebenfalls einen HTTP-Header, der aber gerade mal ein »OK« zurück liefert. In der Realität enthält der Header natürlich einen HTTP-Return-Code, der sagt, ob die Anfrage bedient werden kann. Kann der Web-Server die gewünschte Datei nicht finden, sendet er einen HTTP-404-Fehler, »not found«. Findet er die Datei, dann sendet er den Return-Code 200: »Alles klar, ich bearbeite die Anfrage.«

Dieses einfache Beispiel zeigt eines der wichtigsten Konzepte von Netzwerkmodellen: Wenn eine bestimmte Schicht auf einem Computer mit derselben Schicht auf einem anderen Computer kommuniziert, dann nutzen die beiden Computer *Header*, welche die zu kommunizierenden Informationen enthalten. Die Header sind ein Teil dessen, was zwischen den Computern übertragen wird. Dieser Prozess wird »*same-layer interaction*« genannt, übersetzt etwa »Interaktion gleicher Schichten«.

Das Anwendungsschichtprotokoll (HTTP in unserem Beispiel) auf Peters Computer kommuniziert mit der Anwendungsschicht auf dem Web-Server Webby. Diese Kommunikation erfolgt durch das Erzeugen und Senden von Anwendungsschicht-Headern. Egal um welches Anwendungsschichtprotokoll es sich handelt, sie alle nutzen dasselbe Konzept der Kommunikation.

Neben HTTP umfassen die Protokolle dieser Schicht u.a. noch die Applikationsprotokolle FTP und SMTP sowie die administrativen Protokolle SNMP, DHCP und DNS.

1.1.2 Die TCP/IP-Transportschicht

Während zur TCP/IP-Anwendungsschicht relativ viele Protokolle zählen – HTTP ist ja nur eines davon –, gibt es auf der *TCP/IP-Transportschicht* eigentlich nur zwei Hauptprotokolle, die der Rede wert sind: das *Transmission Control Protocol* (TCP) und das *User Datagram Protocol* (UDP). Eine detaillierte Beschreibung der Transportprotokolle erfolgt später, und in diesem Abschnitt konzentrieren wir uns auf eine Schlüsselfunktion von TCP, die gut geeignet ist, etwas mehr über das generelle Konzept von Netzwerkmodellen zu erklären.

Um zu verstehen, was *Transportprotokolle* leisten, müssen wir an die Schicht direkt oberhalb der Transportschicht denken, die Anwendungsschicht. Jede Schicht stellt der direkt oberhalb liegenden Schicht einen Dienst zur Verfügung. Kehren wir noch einmal zurück zu unserem Beispiel mit Peter und Webby. Was wäre passiert, wenn Peters HTTP-Anfrage oder die Antwort des Web-Servers während der Übertragung irgendwo im TCP/IP-Netzwerk verloren gegangen wäre? Klar, die Seite wäre nicht im Browser erschienen.

TCP/IP benötigt also einen Mechanismus, der die Lieferung von Daten über ein Netzwerk garantiert. Da natürlich sehr viele Anwendungsschichtprotokolle eine garantierte, also zuverlässige Datenübertragung über ein Netzwerk wünschen, bietet TCP ihnen eine Error-Recovery-, also Fehlerbehebungsfunktion, die sich Acknowledgments (Bestätigungsnummern) bedient.

Betrachten Sie Abbildung 1.2: Der Web-Browser beauftragt TCP, die HTTP-Get-Anfrage zuverlässig auszuliefern. TCP sendet die Daten von Peter zum Web-Server – die Daten treffen fehlerfrei beim Web-Server ein, was dieser umgehend durch ein Acknowledgment bestätigt. Außerdem reicht der Web-Server die Daten an die Web-Server-Software weiter, die sie verarbeitet. Dasselbe geschieht in umgekehrter Richtung mit der Antwort des Web-Servers, die ebenso erfolgreich bei Peter eintrifft.

Welche Vorteile die TCP-Fehlerbehebung bietet, stellt man natürlich erst dann fest, wenn die Daten unterwegs verloren gehen. Gehen wir einstweilen davon aus, dass bei einem Datenverlust nicht etwa HTTP

eingreift, sondern TCP die Daten erneut sendet und gewährleistet, dass sie erfolgreich empfangen werden.

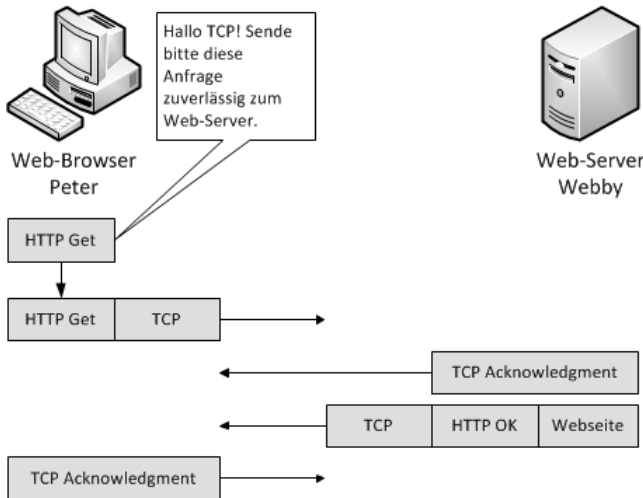


Abb. 1.2: TCP stellt HTTP seine Dienste zur Verfügung.

Dieses zweite Beispiel veranschaulicht ein Konzept, das »*adjacent-layer interaction*« genannt wird, übersetzt etwa »Interaktion benachbarter oder angrenzender Schichten«. Dieses Konzept beschreibt, wie die benachbarten Schichten eines Netzwerkmodells auf demselben Computer zusammenarbeiten. Das Protokoll der höheren Schicht, in diesem Fall HTTP, muss etwas tun, was es nicht kann (Fehlerbehebung). Also beauftragt das Protokoll der höheren Schicht das Protokoll der eine Stufe tiefer liegenden Schicht (TCP) damit, diese Aufgabe zu übernehmen. Das Protokoll der tieferen Schicht bietet dem Protokoll der höheren Schicht also einen Dienst.

Die beiden Beispiele zur Anwendungs- und Transportschicht ignorieren viele Details des physischen Netzwerks. Beide Schichten funktionieren immer genau gleich, unabhängig davon, ob sich die involvierten Endpunkt-Computer im selben LAN befinden oder ob sie durch das komplette Internet voneinander getrennt sind. Die zwei verbleibenden

Schichten aber, die Internet- und die Netzzugangsschicht, müssen das zugrunde liegende physische Netzwerk verstehen, denn sie definieren die Protokolle, die benutzt werden, um die Daten von einem Host zum anderen zu liefern.

1.1.3 Die TCP/IP-Internetschicht

Die *Internetschicht* ist zuständig für die logische Adressierung der physischen Netzwerkschnittstelle. Das klingt kompliziert, ist aber ganz einfach. Sehen wir uns noch einmal die Anfrage an, die Peter zum Web-Server sendet, diesmal mit einigen Details über das *Internetprotokoll* (IP). Die Linien bei Peters Arbeitsstation und dem Web-Server repräsentieren einfach zwei LANs, deren Details nicht wichtig sind. Wenn Peter die Daten sendet, dann sendet er tatsächlich ein IP-Paket. Dieses IP-Paket enthält einen IP-Header, den Transportschicht-Header (in diesem Fall einen TCP-Header), den Anwendungsschicht-Header (HTTP) und Anwendungsdaten (in diesem Fall keine). Der IP-Header enthält jeweils ein Quell- und ein Ziel-IP-Adressfeld. Das Quell-IP-Adressfeld enthält Peters IP-Adresse (1.1.1.1), das Ziel-IP-Adressfeld die IP-Adresse des Web-Servers (2.2.2.2).

Peter sendet das Paket zum Router R₁. R₁ untersucht die Ziel-IP-Adresse (2.2.2.2) und fällt die Routing-Entscheidung, das Paket zum Router R₂ zu senden. Das funktioniert, weil R₁ genug von der Netzwerktopologie kennt, um zu wissen, dass der Web-Server (2.2.2.2) auf der anderen Seite von R₂ liegt. Wenn R₂ das Paket empfängt, leitet dieser Router es über das Ethernet an den Web-Server weiter. Sollte die Verbindung zwischen R₁ und R₂ ausfallen, dann erlaubt IP R₁, eine neue Route zu lernen, die den Web-Server über R₃ erreicht.

IP definiert also IP-Adressen für jedes TCP/IP-fähige Gerät (IP-Host genannt). Diese Adressen erlauben IP-Hosts zu kommunizieren. Außerdem definiert IP *Routing*. Routing beschreibt, wie ein Router Datenpakete weiterleiten oder *routen* sollte.

Auf dieser Schicht finden wir also IP, außerdem unterstützende Protokolle wie ICMP und diverse Routing-Protokolle wie OSPF, RIP oder BGP.

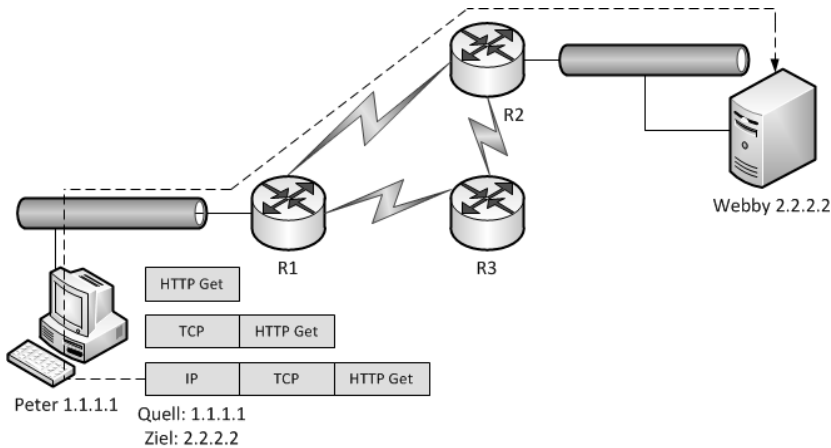


Abb. 1.3: IP-Dienste

1.1.4 Die TCP/IP-Netzzugangsschicht

Die *Netzzugangsschicht* definiert die für die Lieferung von Daten über ein physisches Netzwerk notwendigen Protokolle und Geräte. Der Begriff *Netzzugang* oder *Network Access* lässt bereits daran denken, dass diese Schicht definiert, wie ein Host-Computer zu verbinden ist mit dem physischen Medium, über das Daten transportiert werden können. Ethernet ist beispielsweise ein Protokoll, das auf der Netzzugangsschicht angesiedelt ist. Es definiert die notwendige Verkabelung, die Adressierung und Protokolle für ein Ethernet-LAN. Andere Protokolle der Netzzugangsschicht definieren Stecker, Kabel und Stromstärken für Protokolle, die Daten über WAN-Verbindungen übertragen.

Hier ist anzumerken, dass auf dieser Schicht sehr häufig überhaupt kein Protokoll läuft, das zur TCP/IP-Familie gehört. Wird TCP/IP zum Beispiel über ein Ethernet ausgeführt, dann ist es Ethernet, das sich um die Funktionen dieser Schicht kümmert. Trotzdem definiert der TCP/IP-Standard natürlich Protokolle für TCP/IP-Netzwerke, die keine eigenen Netzzugangsschichtprotokolle haben. Diese Protokolle sind das *Serial Line Internet Protocol* (SLIP) und das *Point-to-Point Protocol* (PPP).

Wie jede Schicht in jedem beliebigen Netzwerkmodell bietet die Netzzugangsschicht der im Modell über ihr liegenden Schicht Dienste. Sehen wir uns an, welche Dienste sie IP bietet. IP nutzt die Netzzugangsschicht, um IP-Pakete über das physische Netzwerk zu übertragen. Nun kennt IP zwar die Netzwerktopologie und weiß, welche Router miteinander und welche Computer mit welchen physischen Netzwerken verbunden sind, aber Details des zugrunde liegenden physischen Netzwerks kennt IP nicht. Deshalb nutzt IP die Dienste der Netzzugangsschicht.

Die Netzzugangsschicht umfasst natürlich eine Menge unterschiedlicher Protokolle. Dazu zählen beispielsweise sämtliche Variationen von Ethernet-Protokollen und anderen LAN-Standards, außerdem WAN-Standards wie ATM und das *Point-to-Point Protocol* (PPP).

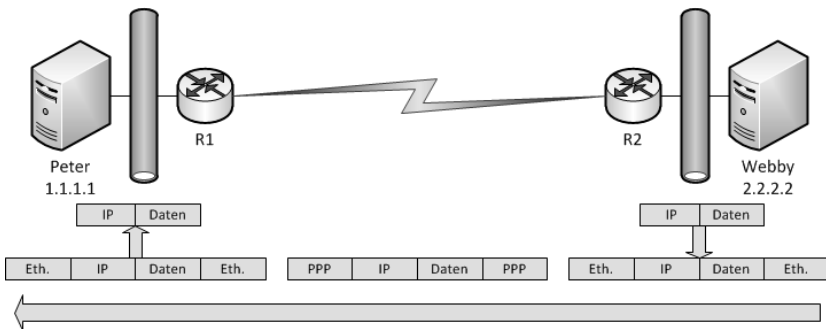


Abb. 1.4: Ethernet- und PPP-Dienste für IP

Sehen wir uns Abbildung 1.4 an. Web-Server Webby (rechts in der Abbildung) sendet seine Antwort an Peter. Dazu nutzt der Web-Server zunächst Ethernet, um das Paket an Router R2 zu senden. Dieser Prozess verlangt die Einhaltung der Ethernet-Protokollregeln. Spezifisch bedeutet dies, das IP-Paket (den IP-Header und die Daten) zwischen einen Ethernet-Header und einen Ethernet-Trailer zu stecken. Diesen generellen Vorgang nennt man *Einkapselung* oder *Kapselung* (Encapsulation).

Das Paket ist nun bei R2 angekommen, es muss nun von R2 zu R1 übertragen bzw. geroutet werden. Zwischen den beiden Routern wird

das WAN-Protokoll PPP genutzt. Aufgabe des IP-Routings ist es, IP-Pakete auszuliefern, das heißt IP-Header und Daten. R₂ benötigt den von Webby übermittelten Ethernet-Header und -Trailer nicht mehr. Also entfernt er den Ethernet-Header und -Trailer, übrig bleibt das Original-IP-Paket. Nun geht es mittels PPP weiter. R₂ kapselt das IP-Paket also zwischen einem PPP-Header und PPP-Trailer ein und sendet den entstandenen Daten-Frame über die WAN-Verbindung zum Router R₁.

Wichtig

Die Datengruppe einschließlich der Ethernet- und/oder PPP-Header und -Trailer auf Ebene der Netzzugangsschicht nennt man *Frame*. Auf Ebene der Internetschicht spricht man von einem *Paket*, und die Transportschicht überträgt *Segmente*.

Auf der Seite von Peter dreht sich der Vorgang um: R₁ entfernt den PPP-Header und -Trailer, weil die Übertragung über die serielle Verbindung erledigt ist. Dann setzt R₁ einen neuen Ethernet-Header vor und einen neuen Ethernet-Trailer hinter das IP-Paket, um es nun via Ethernet zu Peters Computer weiterleiten zu können.

1.2 Das OSI-Referenzmodell

Grundlage einer standardisierten Kommunikation zwischen Computern ist das *OSI-Referenzmodell* der ISO (OSI = *Open Systems Interconnection*, ISO = *International Organization for Standardization*). Das OSI-Modell ist für die Informations- und Kommunikationstechnologie wichtig, weil es ein komplettes Modell der Funktionen eines Kommunikationssystems zur Verfügung stellt. Halten sich also die verschiedenen Hersteller von Systemen an dieses Modell, dann können sich die Systeme untereinander verstehen. Allerdings passt das Modell nicht immer und ist auch nicht auf allen Computern verfügbar. Nichtsdestotrotz eignet es sich prima zur Beschreibung, zur Analyse und zum Vergleich verschiedener netzwerk- und kommunikationsrelevanter Angelegenheiten.

Ich beschreibe das OSI-Referenzmodell hier recht oberflächlich, da nahezu jedes Buch, das von Netzwerken allgemein handelt, Beschreibungen dieses Modells beinhaltet – der interessierte Leser sei darauf verwiesen. Die ISO hat für das OSI-Referenzmodell sieben Schichten definiert, die klar voneinander getrennt sind. Jede Schicht beschreibt detailliert bestimmte Aufgaben und Funktionen des Kommunikationsprozesses.

Das OSI-Referenzmodell beschreibt den Datenfluss in einem Netzwerk, angefangen bei der niedrigsten Schicht, die die physikalische Verbindung definiert, bis zu der Schicht, die die Anwendung des Benutzers definiert. Jede Information, die über das Netzwerk übertragen wird, wird wie beim zuvor besprochenen TCP/IP-Netzwerkmodell von Schicht zu Schicht weitergereicht. Jede einzelne Schicht kommuniziert mit der jeweils unter- und übergeordneten Schicht (*adjacent-layer interaction*) über definierte Schnittstellen. Empfängt eine Schicht ein Datenpaket, so prüft sie die Zieladresse des Pakets. Ist die eigene Adresse nicht vorhanden, so wird das Paket zur nächsten Schicht weitergereicht.

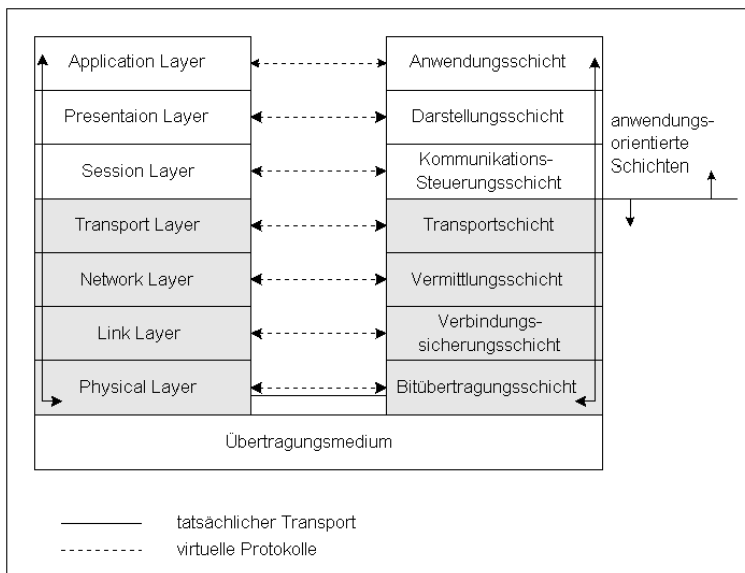


Abb. 1.5: Die sieben Schichten des OSI-Referenzmodells

Wenn zwei Computer in einem Netzwerk miteinander kommunizieren, so unterstellt die Software jeder einzelnen Schicht eines Computers, dass sie mit genau derselben Schicht des anderen Computers kommuniziert (*same-layer interaction*). So kommuniziert beispielsweise die Transportschicht des ersten Computers mit der Transportschicht des zweiten Computers. Dabei hat die Transportschicht des ersten Computers überhaupt keine Ahnung davon, wie die Kommunikation die tieferen Schichten durchläuft, über das physikalische Medium übertragen wird und schließlich durch die einzelnen Schichten des zweiten Computers fließt.

Wie erwähnt, besitzt jede einzelne Schicht fest umrissene Aufgaben, die nachfolgend kurz beschrieben werden:

Die *Anwendungsschicht* repräsentiert die Dienste, die Anwendungssoftware direkt unterstützen, beispielsweise Software für Filetransfer, Datenbankzugriffe oder E-Mail. Diese Schicht ist der Punkt, an dem Anwendungsprogramme, die außerhalb des Computers kommunizieren müssen, auf Netzwerkdienste zugreifen. Die Anwendungsschicht definiert außerdem Prozesse für die Benutzerauthentifizierung.

Die *Darstellungsschicht* ist zuständig für die Transformation der Daten in Standardformate, beispielsweise ASCII-Text, EBCDIC-Text, binär, BCD oder JPEG. Neben der Interpretation und Umsetzung der Daten bietet die Darstellungsschicht Dienste wie Verschlüsselung und Datenkompression.

Die *Kommunikationssteuerungs- oder Sitzungsschicht* erlaubt zwei Anwendungen, auf verschiedenen Konversationen oder Sitzungen (Sessions) aufzubauen, zu verwenden und abzubauen. Man redet hierbei von Prozess-zu-Prozess-Verbindungen. Die Kommunikationssteuerungsschicht ist ferner zuständig für die Prozesssynchronisation. Diese Schicht ist die unterste Ebene der anwendungsorientierten Schichten des Referenzmodells.

Zu den Aufgaben der *Transportschicht* zählt u. a. Fehlererkennung und -behebung. Lange Nachrichten werden, falls notwendig, in dieser Schicht für die Übertragung in kleinere Segmente zerlegt und auf Seiten des Empfängers wieder zu den Originalnachrichten zusammenge-

setzt. Die empfangende Transportschicht kann Empfangsbestätigungen senden. Die Transportschicht bildet logische Ende-zu-Ende-Verbindungen in Abstraktion der technischen Übertragungssysteme.

Die *Vermittlungsschicht* adressiert Nachrichten und übersetzt logische Adressen und Namen in physikalische Adressen. Sie bestimmt die Route vom Quell- zum Zielcomputer. Hier erfolgt also die Wegbestimmung im Netz, das Routing.

Die *Verbindungssicherungs- oder kurz Sicherungsschicht* bildet aus den Bits der Bitübertragungsschicht und den Paketen der Vermittlungsschicht Frames (logisch strukturierte Datenpakete). Die Schicht ist zuständig für den Transport der Frames von einem Computer zu einem anderen.

Die *Bitübertragungsschicht* überträgt Bits von einem Computer zu einem anderen. Hinter dieser Schicht verbergen sich die nachrichtentechnischen Hilfsmittel für die Übertragung der Bits.

Wichtig

Was Sie sich über das OSI-Referenzmodell unbedingt merken sollten, ist die Tatsache, dass das Modell beschreibt, was getan werden soll, nicht jedoch, wie es getan wird. Wichtig ist das insofern, als dass die Details der Implementierungen des Systems den jeweiligen Herstellern überlassen bleiben.

1.2.1 Einordnung der Komponenten und Protokolle ins OSI-Referenzmodell

Der große Vorteil des OSI-Referenzmodells ist dessen klare Strukturierung: Jeder einzelnen Schicht des Modells sind klare Aufgaben zugeordnet. Die Kommunikation zwischen den Schichten läuft über ebenso klar definierte Schnittstellen.

Was jetzt noch fehlt, ist die Zuordnung der Bestandteile eines Kommunikationssystems zu den einzelnen Schichten des Modells. In der folgenden Tabelle sehen Sie, auf welchen Ebenen sich bereits bekannte

Dinge wie Verkabelung und Netzwerkkarten befinden. Weiterhin wird dargestellt, auf welchen Ebenen sich welche Protokolle befinden.

| OSI-Schicht | Bestandteil |
|---------------------------|---|
| 7. Anwendungsschicht | HTTP, Telnet, FTP, POP3, X.400, FTAM, VTAM, X.500 |
| 6. Darstellungsschicht | ASN.1, SMB |
| 5. Sitzungsschicht | NCP, NLSP |
| 4. Transportschicht | TCP, SPX |
| 3. Vermittlungsschicht | IP, IPX, Router |
| 2. Sicherungsschicht | Ethernet (IEEE 802.3), Token-Ring (IEEE 802.5), PPP, Frame-Relay, ATM, LAN-Switch, Wireless-Access-Point, DSL-Modem, Kabelmodem |
| 1. Bitübertragungsschicht | Ethernet (IEEE 802.3), RJ-45, EIA/TIA-232, V.35, Hub, Repeater, Verkabelung, Netzwerkkarte |

Tabelle 1.2: OSI-Schichten und Bestandteile

1.2.2 OSI und TCP/IP

Das OSI-Modell wird heute als Standard für Vergleiche mit anderen Netzwerkmodellen genutzt. Tabelle 1.3 vergleicht die sieben Schichten des OSI-Modells mit den vier Schichten des TCP/IP-Modells.

| OSI | TCP/IP |
|------------------------|--------------------|
| Anwendungsschicht | Anwendungsschicht |
| Darstellungsschicht | |
| Sitzungsschicht | |
| Transportschicht | Transportschicht |
| Vermittlungsschicht | Internetschicht |
| Sicherungsschicht | Netzzugangsschicht |
| Bitübertragungsschicht | |

Tabelle 1.3: OSI und TCP/IP

Jede der sieben Schichten des OSI-Modells besitzt eine Sammlung genau definierter Funktionen. Somit ist es möglich, jedes Netzwerkprotokoll und jede Netzwerkspezifikation zu untersuchen und festzustellen, welcher der sieben Schichten das Protokoll oder die Spezifikation entspricht. Die Internetschicht von TCP/IP, überwiegend durch IP implementiert, entspricht ziemlich direkt der Vermittlungsschicht von OSI. Viele Netzwerkprofis sagen deshalb auch, IP sei ein *Schicht-3- oder Layer-3-Protokoll*. Nummeriert man die OSI-Schichten mit der Bitübertragungsschicht beginnend durch, landet man tatsächlich bei Schicht 3. Im TCP/IP-Modell wäre es allerdings Schicht 2 – schön blöd. Da sich aber bei der Beschreibung von Protokollen jeder auf das OSI-Modell bezieht, kann man IP tatsächlich als Schicht-3-Protokoll bezeichnen – alles wieder im Lot. Statt Schicht-3-Protokoll kann man natürlich auch Vermittlungsschichtprotokoll oder Network-Layer-Protokoll sagen. Habe ich Ihnen schon gesagt, dass ich diese deutschen Übersetzungen der Schichtbezeichnungen ziemlich künstlerisch und wenig zungenfreundlich finde? So sehen das auch andere Zeitgenossen – ich habe jedenfalls noch nie jemanden von einem Vermittlungsschichtprotokoll oder gar Kommunikationssteuerungsschichtprotokoll reden gehört.

Beim Betrachten der Tabelle 1.3 könnte man den Eindruck gewinnen, die Vermittlungsschicht des OSI-Modells wäre mit der Internetschicht des TCP/IP-Modells identisch. Das ist nicht der Fall, aber beide Schichten sind sich immerhin sehr ähnlich. Die OSI-Vermittlungsschicht definiert die logische Adressierung und das Routing. Das tut die TCP/IP-Internetschicht ebenfalls. Die Details unterscheiden sich, aber da beide Schichten dieselben Dinge spezifizieren, entspricht die TCP/IP-Internetschicht der OSI-Vermittlungsschicht doch sehr genau. Dasselbe gilt auch für die TCP/IP-Transportschicht. Hier heißt die entsprechende OSI-Schicht sogar genau so. TCP ein Transportschicht- oder Schicht-4-Protokoll zu nennen, ist also völlig korrekt.

Aber nicht alle Schichten des TCP/IP-Modells entsprechen genau einer Schicht des OSI-Modells. Beispielsweise definiert die TCP/IP-Netzzugangsschicht Protokolle und Spezifikationen, die OSI in den Schichten 1 und 2, also der Bitübertragungs- und der Verbindungssicherungsschicht definiert.

1.2.3 OSI-Einkapselung

Etwas weiter oben wurde kurz das Konzept der Kapselung oder Einkapselung erklärt. Darunter versteht man das Hinzufügen immer neuer Header- und gegebenenfalls Trailer-Informationen zu einem Datenpaket. Jede Schicht eines Netzwerkmodells fügt den Daten einen neuen Header hinzu, einige Schichten auch einen Trailer. Je nachdem, um welche Schicht im TCP/IP-Modell es geht, besitzen die entstandenen Datenpakete andere, eindeutige Namen: *Segmente* auf der TCP/IP-Transportschicht, *Pakete* auf der TCP/IP-Internetschicht und *Frames* auf der TCP/IP-Netzzugangsschicht.

Bei OSI sieht das ganz ähnlich aus, denn auch OSI funktioniert mit dem Prinzip der Kapselung und bezeichnet die entstehenden Datenpakete auf jeder Schicht anders. Allerdings nutzt OSI nicht wirklich originelle eindeutige Namen für die Pakete, sondern nummeriert sie eigentlich nur durch. Bei OSI bezeichnet man ein Datenpaket als *Protocol Data Unit* (etwa Protokolldateneinheit). Um zu kennzeichnen, auf welcher Schicht das jeweilige Datenpaket anzutreffen ist, setzt OSI einfach ein *Layer*, gefolgt von der Nummer der Schicht, davor. Ein Datenpaket der Vermittlungsschicht heißt also *Layer 3 Protocol Data Unit*, abgekürzt *L3PDU*.

1.3 Das weiß ich nun

1. TCP und UDP sind Protokolle welcher OSI-Schicht?
 - a. Sitzungsschicht
 - b. Darstellungsschicht
 - c. Verbindungsschicht
 - d. Sicherungsschicht
 - e. Transportschicht
2. Wie nennt man die über das Netzwerk zu sendende Dateneinheit auf der TCP/IP-Internetschicht?
 - a. Frame
 - b. Päckchen

- c. Segment
 - d. Postbrief
 - e. Paket
3. Welche der folgenden Protokolle sind Protokolle der TCP/IP-Internetschicht?
- a. SMTP
 - b. Ethernet
 - c. UDP
 - d. IP
 - e. HTTP
 - f. TCP
4. Welche OSI-Schicht definiert die Standards für Datenformate und Verschlüsselung?
- a. Sicherungsschicht
 - b. Darstellungsschicht
 - c. Vermittlungsschicht
 - d. Transportschicht
5. Welcher OSI-Schicht entspricht die TCP/IP-Internetschicht?
- a. Vermittlungsschicht
 - b. Darstellungsschicht
 - c. Sicherungsschicht
 - d. Nachtschicht
 - e. Transportschicht
6. Welche TCP/IP-Schicht definiert logische Adressen und Routing?
- a. Netzzugangsschicht
 - b. Internetschicht
 - c. Transportschicht
 - d. Anwendungsschicht

Auflösung im Anhang A.