
mitp
Dirk
Jarzyna
TCP/IP
Grundlagen, Adressierung, Subnetting

Dirk Jarzyna

TCP/IP – Grundlagen, Adressierung, Subnetting

mitp

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

ISBN 978-3-8266-9553-7
1. Auflage 2013

www.mitp.de
E-Mail: kundenbetreuung@hjr-verlag.de
Telefon: +49 6221 / 489 -555
Telefax: +49 6221 / 489 -410

Lektorat: Ernst Heinrich Pröfener
Sprachkorrektorat: Jürgen Dubau
Satz: III-satz, Husby, www.drei-satz.de

Inhaltsverzeichnis

Einführung

Über TCP/IP sind schon viele Bücher geschrieben worden, dicke und dünne, leicht verständliche, schwer verdauliche, rote und blaue. Kein origineller Einfall also, ein weiteres hinzuzufügen. So war ursprünglich auch geplant, ein Buch ausschließlich über IPv6-Grundlagen zu schreiben. Doch schon zu Beginn der Arbeit wurde klar, dass es kaum möglich ist, einfach bei IPv6 einzusteigen, ohne zuvor über IP in der Version 4 oder TCP/IP allgemein geschrieben zu haben. Denn IPv6 baut in vielerlei Hinsicht auf IPv4 auf, und es ist deutlich einfacher, IPv6 zu verstehen, wenn IPv4 verstanden ist. Da man nicht unbedingt bei jedem an IPv6 interessierten Leser ein solides Grundverständnis von IPv4 voraussetzen kann, erschien es mir notwendig, in den ersten Kapiteln dieses Buchs ein solches Grundverständnis aufzubauen. Entstanden ist so schließlich ein Buch, das Grundwissen über IPv4 bzw. TCP/IP und IPv6 zu gleichen Teilen vermittelt. Tatsächlich nimmt sich dieses Buch bestimmte IPv4-Themen sehr gründlich vor. Das betrifft beispielsweise die mit IPv4-Adressen verbundene Mathematik. Ich habe die Erfahrung gemacht, dass zwar viel über IP-Adressen und Subnetting geschrieben wird, aber der Interessierte kaum eine gründliche Anleitung findet, die ihm dabei hilft, Subnetting in der Praxis durchzuführen.

In diesem Buch geht es also generell um TCP/IP. Deshalb war der für die erste Auflage dieses Buchs genutzte Titel »IPv6 – Das Praxisbuch« unglücklich gewählt. Ich bin zwar der Auffassung, dass das Buch tatsächlich alle für die Praxis relevanten Aspekte von IPv6 behandelt, aber der Titel war definitiv irreführend. Der Titel »TCP/IP – Grundlagen, Adressierung, Subnetting trifft den Inhalt genauer. Aber natürlich befasst sich ein großer Teil des Buches mit IPv6. Und das aus gutem Grund:

Die Netzwerk- und Internetwelt hat sich im Laufe der letzten Jahre dramatisch verändert: Ethernet feierte im Mai 2013 seinen vierzigsten Geburtstag! Das Internet ist riesig geworden und gereift, die Netzwerktechnik hat sich weiterentwickelt, und neue Technologien, die vor zehn, zwanzig Jahren noch unbekannt waren, sind heute allgegenwärtig und erlauben es den Menschen, miteinander zu kommunizieren, Dokumente, Bilder, Musik, Videos und Gummibärchen auszutauschen und von fast jedem Ort der Erde aus auf Daten zuzugreifen, die an irgendeinem anderen Ort auf der Erde gespeichert sind. Noch vor zwanzig Jahren gab es kein globales Netzwerk, mit dem sich interessierte Zeitgenossen einfach so verbinden konnten. Erst vor rund zehn, zwölf Jahren war das öffentliche Internet an dem Punkt angelangt, wo sich Menschen in den meisten Teilen der Welt mit ihm verbinden konnten. Selbst zur Jahrtausendwende waren die typischen Internetbenutzer überwiegend Menschen mit einem Faible für Computertechnik oder Benutzer, die das Internet beruflich nutzten. Heute scheint praktisch jeder aufs Internet zuzugreifen – über PCs, mobile Geräte, Telefone, Fernsehgeräte, Radios und sogar Kühlschränke. Und das mit dem Kühlschrank ist ernst gemeint.

So gut wie jedes Mobiltelefon unterstützt Internetverkehr und benötigt deshalb eine IP-Adresse. Dies gilt auch für moderne TV-Geräte, für Internet-Radios sowieso. Viele neue Autos können eine IP-Adresse beziehen und nutzen. Einige Hardwarehersteller sind der Meinung, wirklich jede ihrer Appliances benötige unbedingt die Fähigkeit, sich mit dem Internet zu verbinden. Selbst Nintendo hat den Gameboy zum IP-Adressen-Konsumenten weiterentwickelt, indem das Unternehmen ihm einen kleinen Web-Browser und ein paar weitere Funktionen eingepflanzt hat. Die riesige Zahl der Internetbenutzer und fast explosionsartig zunehmende Anzahl internetfähiger Endgeräte hat Auswirkungen: Die verfügbaren IP-Adressen (IPv4) sind so gut wie erschöpft. Das bedeutet nicht, dass sich Netzwerkfachleute nicht mehr mit IPv4 beschäftigen müssen, denn die riesige Zahl IPv4 nutzender Geräte will ja nach wie vor gemanagt werden, sondern es bedeutet, dass nun zusätzlich IPv6-Kenntnisse erforderlich sind.

»Irgendwann innerhalb der nächsten sechs Jahre wird die Menge der noch zuteilbaren IPv4-Adressen verbraucht sein.« Dies schrieb in ähnlicher Form die *Information Week*, und zwar bereits am 21. Mai 2007 (`www.informationweek.com`, »The Impending Internet Address Shortage«). Diese Situation ist inzwischen eingetreten.

Das bedeutet, dass die Migration zum neuen Internet-Protokoll IPv6 beschleunigt werden muss, um die sich abzeichnende Katastrophe ein für alle Mal zu stoppen.

Die Migration zu IPv6 wird durch den Bedarf nach immer mehr Adressen getrieben werden. Außerdem steigt die Nachfrage nach IPv6 durch den Druck der Behörden: Die US-Regierung setzte bereits ein Datum im Jahr 2008, bis zu dem sämtliche Einrichtungen, Behörden, Ämter und Agenturen der Regierung ihre Core-IP-Netzwerke auf IPv6 umgestellt haben sollten. Inzwischen haben wir 2013, die Sache dürft also längst erledigt sein. Ob dies tatsächlich so ist, war mir nicht wichtig genug, um es zu recherchieren. Als Signal taugt diese Regierungsinitiative auf jeden Fall.

Die Frage, die sich IT-Verantwortlichen stellt, lautet nicht mehr, ob sie zu IPv6 migrieren, sondern *wann* sie dies tun.

Die zwei wichtigsten Gründe für die Migration zu IPv6 wurden genannt: der Bedarf nach mehr Adressen und die Vorgaben durch Behörden. Daneben gibt es aber noch viele weitere Gründe, die IP-Verantwortliche eine Migration in Angriff nehmen lassen sollten, darunter folgende:

- Adresszuweisungsfunktionen: Die IPv6-Adresszuweisung erlaubt dynamische Zuteilung, leichtere Änderung und die Wiederherstellung von Adressen.
- Kein Bedarf für NAT und PAT: Durch Nutzung öffentlich registrierter eindeutiger Adressen auf allen Geräten entfällt die Notwendigkeit von Netzwerkadress- und Port-Übersetzungen. Ein angenehmer Nebeneffekt ist die Beseitigung einiger Anwendungsschicht- und VPN-Tunneling-Probleme, die NAT sonst bereitet.

- Aggregation: Der riesige Adressbereich von IPv6 erlaubt eine viel leichtere Zusammenfassung von Adressblöcken im Internet.
- IPSec: IPSec funktioniert natürlich sowohl mit IPv4 als auch mit IPv6, ist auf IPv6-Hosts jedoch zwingend erforderlich. Man kann also darauf vertrauen, dass IPSec vorhanden ist, beispielsweise für VPN-Tunneling.
- Header-Verbesserungen: Router müssen nicht mehr für jedes Paket eine Header-Prüfsumme berechnen, was natürlich den Overhead pro Paket reduziert. Außerdem enthält der Header ein Flow-Label, das die leichte Identifizierung von Paketen erlaubt, die über dieselbe einzelne TCP- oder UDP-Verbindung gesendet werden.

Selbstverständlich wird die weltweite Migration von IPv4 zu IPv6 kein einmaliges Ereignis sein. Ja selbst innerhalb von ein, zwei Jahren wird sie nicht erledigt sein. Stattdessen wird sie ein sehr langfristiger Prozess sein, der – siehe die Regierung der USA – bereits begonnen hat. IT-Verantwortliche, Netzwerkadministratoren und System Engineers werden zunehmend mehr über IPv6 lernen müssen. Dieses Buch ist eine Informationsquelle dafür.

IPv6 wurde nicht von Grund auf neu erfunden, sondern bedient sich bei einigen Konzepten, Methoden und Strategien durchaus bei IPv4. Andererseits unterscheidet es sich in vielen Punkten signifikant von IPv4. Um IPv6 zu verstehen, bietet es sich deshalb an, beide Protokollfamilien miteinander zu vergleichen. Dieses Buch tut genau dies an vielen Stellen. Im ersten Teil des Buchs findet der Leser alles wirklich Wissenswerte zu TCP/IP und IPv4, im zweiten Teil geht es dann ausschließlich um IPv6.

Dieses Buch erhebt keinen Anspruch auf Vollständigkeit; es beschreibt Dinge, die für die Praxis relevant sind. Sie finden hier also beispielsweise keine Listen, die jedes einzelne Bit einer DHCP-Acknowledgment-Nachricht beschreiben, sondern Sie erfahren hier, wozu DHCP dient, wie es grundsätzlich funktioniert und wo Sie es bei Bedarf herbekommen.

Teil I

TCP/IP-Grundlagen

Kapitel 1

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 Internetzwerk. 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.

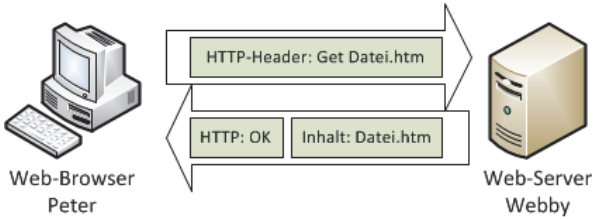

Abb. 1.1: Eine HTTP-Get-Anfrage und die HTTP-Antwort

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.

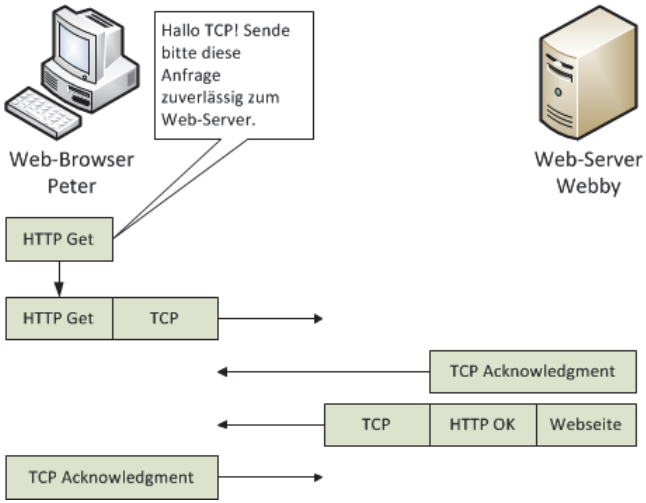

Abb. 1.2: TCP stellt HTTP seine Dienste zur Verfügung.

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 ignorierten 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.

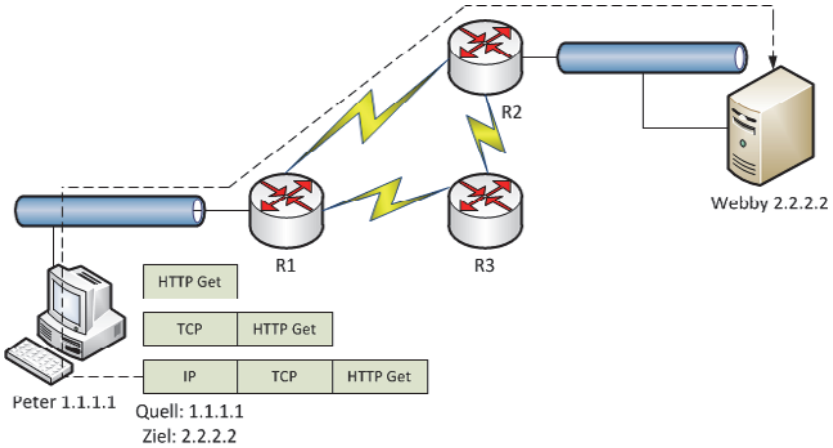

Abb. 1.3: IP-Dienste

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).