In diesem Kapitel:

KAPITEL 2

Technische Hintergründe

In diesem Kapitel werden wir uns intensiver mit den technischen Aspekten des Internets beschäftigen und einige Grundlagen vermitteln, die für das Verständnis der anschließenden Kapitel hilfreich sind. Auf den ersten Blick mag ein Kapitel über die technische Funktionsweise des Internets etwas trocken wirken, seien Sie jedoch gewiss, dass es hinter den Kulissen viel Interessantes zu entdecken gibt.

Jede der hier besprochenen Komponenten bildet Ansatzpunkte für einen Angriff auf die beteiligten Computer: Dienste können unbemerkt im Hintergrund auf Ihrem Computer aktiv sein, Protokolle können so verwendet werden, dass Ihr Computer mit den Anfragen nicht zurechtkommt und abstürzt. Ein Angreifer kann seine IP-Adresse tarnen und sich als ein anderer ausgeben oder das Routing von Daten kann beeinflusst werden. Dies sind bei weitem nicht alle Schwachstellen, die für Angriffe ausgenutzt werden können, es ist lediglich eine Auswahl, um Ihre Sinne für mögliche Gefahren zu schärfen. Die Begriffe und Erläuterungen aus diesem Kapitel sollen es Ihnen erleichtern, diese Gefahren und die Maßnahmen zu ihrer Abwehr besser abschätzen und verstehen zu können.

Aufbau des Internets

Zunächst einmal ist das Internet nichts anderes als ein weltumspannendes Netz von Computern. Ein Computernetzwerk ist dadurch charakterisiert, dass Computer miteinander kommunizieren und sich ihre Ressourcen teilen können. Demzufolge ist auch jeder Computer im Internet für andere erreichbar und bietet oder fordert Leistungen an. Doch das Internet ist weit mehr als eine zufällige Anordnung von Maschinen. Genauer betrachtet ist es ein Netz von Netzwerken, in dem jeder Computer – als Bestandteil eines Netzes auch als *Host* bezeichnet – eine genau definierte Rolle zu erfüllen hat. Die einzelnen Hosts im Internet sind dabei keineswegs gleichwertig. So ist es theoretisch denkbar (wenn auch sehr unwahrscheinlich), dass schon der Ausfall einiger weniger zentraler Systeme (so genannter *Backbones*) den

größten Teil des Internets zum Erliegen bringen könnte. Wenn Sie hingegen an Ihrem Heim-PC den Stecker ziehen, wird das den Datentransfer im weltweiten Netz, wenn überhaupt, nur am Rande beeinflussen. Wie die einzelnen Computer in den Netzwerken, so sind darüber hinaus auch diese Netze miteinander verbunden und tauschen Informationen untereinander aus. In dem Moment, in dem Sie online gehen, wird Ihr PC Teil des Netzwerks Ihres Internet-Providers, und dieses Netz ist wiederum Teil des Internets.

Ein weiteres Charakteristikum des Internets ist, dass es ein heterogenes Netzwerk ist. Konkret bedeutet dies, dass die miteinander verbundenen Netze und Computer höchst unterschiedlich sind. Sie finden Systeme aus verschiedenen Epochen der Computergeschichte neben den allerneuesten High-Tech-Maschinen. Auch die Struktur und Anbindung der Netzwerke ist höchst verschiedenartig. Einige sind über schnelle Glasfaserkabel miteinander verbunden, andere nur über extrem langsame und störanfällige Telefonleitungen. Bereits diese Unterschiede in der beteiligten Hardware machen die Kommunikation im Internet zu einer echten Herausforderung. Noch komplizierter wird es jedoch, wenn man sich bewusst macht, dass auch die eingesetzte Software und die Betriebssysteme äußerst unterschiedlich sind. Im Gegensatz zu der überragenden Dominanz von Microsoft-Betriebssystemen im Privatkundensektor laufen viele der wichtigsten Maschinen im Internet auf anderen Plattformen wie beispielsweise Linux, Solaris, Mac OS, FreeBSD, AIX und zahlreichen anderen, meist Unix-Derivaten.

Der komplexe Aufbau des Internets und die Heterogenität innerhalb der beteiligten Systeme erfordern straffe Standards und technisch exakte Spezifikationen, um Kommunikation überhaupt zu ermöglichen. Die wichtigsten Bereiche dieser Standardisierung sind die Sprache, in der Kommunikation stattfindet, und die Namen, mit denen sich die beteiligten Rechner ansprechen. Beide Aspekte werden wir im Lauf dieses Kapitels näher betrachten.

Internetdienste

Bevor wir uns jedoch mit den zentralen Komponenten der Kommunikation im Internet befassen, müssen wir uns zunächst darüber klar werden, was Kommunikation in diesem Zusammenhang überhaupt bedeutet. Wer sind die Beteiligten in diesem Austausch, und worüber unterhalten sie sich überhaupt?

Eine der grundlegenden Strukturen des Internets ist das Client-Server-Modell. Dabei wird der Informationsaustausch immer als eine Verbindung zwischen genau zwei Kommunikationspartnern angesehen. Einer der beiden Partner – der Server – stellt Informationen zur Verfügung, die der andere – der Client – abrufen kann. Der Begriff *Informationen* steht hierbei für alle möglichen Inhalte, wie z.B. Programme, Dokumente, Bilder, Nachrichten und vieles mehr. Eine Verbindung kann dabei immer nur zwischen einem Server und einem Client, aber niemals zwischen

zwei gleichen Partnern stattfinden.[1] Eine Kommunikation, bei der beide Partner nur Informationen anbieten oder abfragen möchten, ist schon per Definition nicht möglich.

Entgegen dem allgemeinen Sprachgebrauch beziehen sich die Begriffe *Client* und *Server* eigentlich nicht auf einen kompletten Rechner, sondern stets nur auf die miteinander kommunizierenden Programme oder Dienste.[2] Als *Dienst* bezeichnet man in Netzwerken ein Programm, das, einmal gestartet, im Hintergrund wartet, bis es aufgerufen wird, und dann Informationen zur Verfügung stellt oder Befehle ausführt. Nach der Anfrage beendet sich das Programm nicht, sondern wartet wiederum auf eine weitere Anfrage durch einen Client. Dieses Verhalten, das allen Diensten gemeinsam ist, wird uns im Verlauf dieses Buches noch öfter als mögliches Einfallstor für Angriffe begegnen. Der bekannteste dieser Dienste dürfte der WWW-Dienst sein. Er liefert als Ergebnis der Anfrage Ihres Browsers (Client) die angeforderten Internetseiten zurück.

Ein Host, auf dem ein WWW-Dienst läuft, übernimmt also für alle Computer, die diesen Dienst anfragen, die Rolle des Servers. Aus der Sicht eines anderen Dienstes kann sich dieser Host aber durchaus als Client verhalten, indem er selbst Ressourcen von anderen Geräten im Internet in Anspruch nimmt. So synchronisiert ein solcher Host z.B. seine Systemzeit mit anderen Geräten im gleichen Netzwerk meist dadurch, dass er diese von einem so genannten *Timeserver* abfragt. Diese Tatsache verdeutlicht noch einmal, dass das Internet keine Einbahnstraße ist, sondern jeder auch nur temporär angeschlossene Computer vollständiger Bestandteil des Netzes wird. Auch Ihr privater Computer tritt im Internet nicht nur als Client auf, sondern bietet unter Umständen auch seine eigenen Dienste an.

Wie Sie bereits gelesen haben, sind Dienste Angebote, die wir mit Hilfe der richtigen Werkzeuge (z.B. Browser oder E-Mail-Programme) nutzen können. Dabei kann man die Art der Dienste nach verschiedenen Gesichtspunkten gliedern. Eine häufig getroffene Unterscheidung ist die zwischen für uns direkt nutzbaren und so genannten systemnahen Diensten. Letztere dienen eher der Kommunikation der Maschinen untereinander, übernehmen Verwaltungsaufgaben oder sind nur auf ganz spezielle Einsatzgebiete festgelegt. Dazu gehört zum Beispiel der *Syslog*-Dienst, dessen Aufgabe es ist, systeminterne Meldungen zu speichern und zu verwalten. Er kann dabei sowohl die Meldungen der lokalen Programme und Dienste als auch diejenigen von entfernten Maschinen (die auf ihn zugreifen) handhaben. Als weite-

1 Eine der neuesten Entwicklungen im Internet sind so genannte *Peer-to-peer*-Netzwerke, die zum Austausch von Dateien ohne zwischengeschalteten Server genutzt werden (z.B. Gnutella oder Freenet). Der Name suggeriert, dass hier Kommunikation zwischen zwei gleichgearteten Systemen (engl. *peers*) stattfindet. In der Praxis ist es jedoch so, dass die entsprechende Software sich je nach Erfordernis der Situation entweder als Server oder als Client verhält.

2 Der Einfachheit halber werden wir in diesem Buch Computer, die von ihrer Grundidee her hauptsächlich Dienste anbieten, als Server und diejenigen, die hauptsächlich konsumieren, als Clients bezeichnen. Dies entspricht auch dem tatsächlichen Sprachgebrauch.

res Beispiel wäre der *Domain Name Service* (DNS) zu nennen, der meist von Programmen dazu verwendet wird, herauszufinden, welcher Rechner sich hinter einem bestimmten Internetnamen verbirgt. Näheres zu diesem Dienst erfahren Sie weiter hinten in diesem Kapitel, wenn wir uns mit der Wegesuche und der Namensgebung im Internet beschäftigen.

Im weiteren Verlauf dieses Buches werden wir uns bis auf wenige Ausnahmen hauptsächlich mit den direkt für uns nutzbaren Diensten beschäftigen, da diese für den privaten Nutzer von größerem Interesse sind. Als Beispiel für diese Dienste seien hier die drei meistgenutzten Angebote des Internets genannt.

- Der WWW-Dienst (*World Wide Web*) bietet die Webseiten an, die Sie in Ihrem Browser sehen und ist – entgegen der landläufigen Meinung – nicht mit dem eigentlichen Internet gleichzusetzen. Server, die einen solchen Dienst anbieten, werden auch als Webserver bezeichnet.
- Der E-Mail-Dienst besteht eigentlich aus zwei unterschiedlichen Teilen (siehe Tabelle 2-1). SMTP (*Simple Mail Transfer Protocol*) dient dabei dem Versand von Nachrichten, während POP (*Post Office Protocol*) zum Empfang genutzt wird.[3] Häufig sind sowohl der POP- als auch der SMTP-Server auf der gleichen Maschine installiert. Weitere Details zu diesem Thema finden Sie in Kapitel 6, *E-Mail – wer liest mit?*.
- Der dritte und auch weiterhin sehr verbreitete Dienst ist für den Austausch von Dateien zwischen Client und Server zuständig und heißt FTP (File Transfer Protocol).

Tabelle 2-1: Auswahl der am häufigsten genutzten Dienste

Dienst	Protokoll	Funktion des Dienstes
WWW	http	Anzeige von Internetseiten
DNS	DNS	Namensauflösung von IP-Adressen
E-Mail	POP SMTP	Empfangen von E-Mails Verschicken von E-Mails
FTP	ftp	Up- und Download von Daten
Telnet	Telnet	Arbeiten auf einem entfernten Server

Kommunikationswege

Nachdem wir uns im letzten Abschnitt mit dem grundsätzlichen Aufbau des Internets beschäftigt haben, werden wir uns jetzt die Kommunikation der Hosts untereinander genauer anschauen.

3 Neben POP gibt es mit IMAP (*Internet Message Access Protocol*) noch ein zweites Protokoll zum Abholen von Mails; auf IMAP werden wir in Kapitel 6, *E-Mail – wer liest mit?*, nochmals zu sprechen kommen.

Wie Sie bereits gelesen haben, bieten einige Maschinen Ressourcen an, die von anderen genutzt werden können. Da beide Kommunikationspartner aber mehrere Dienste betreiben oder mit zahlreichen Partnern gleichzeitig kommunizieren können, ergibt sich ein Zuordnungsproblem. Woher weiß beispielsweise ein Server, dass der eine Client gerade eine Webseite von ihm anfordert, während ein anderer E-Mails abrufen möchte?

Um dieses Problem zu lösen, hat man jedem Dienst eine eigene Nummer (einen *Port*) zugewiesen, die für ihn charakteristisch ist. Wenn nun also ein Client eine Webseite anfordern möchte, spricht er den Webserver unter Angabe dieses Ports an, so dass dieser weiß, dass sein WWW-Dienst gemeint ist. Etwas vereinfacht gesagt funktioniert das folgendermaßen: Wenn ein Dienst neu gestartet wird, weist man ihm einen Port zu, an dem er von nun an auf Verbindungswünsche »lauscht«. Anfragen auf andere Ports hingegen soll er gar nicht erst zur Kenntnis nehmen. Der Webdienst lauscht z.B. standardmäßig auf Port 80, während man POP unter Port 110 erreichen kann. Wenn Sie mit Ihrem Browser also eine Internetseite aufrufen möchten, müssen Sie sich im Prinzip immer die 80 hinter den Seitennamen denken. Daher könnten Sie anstatt *http://www.oreilly.de* auch (technisch exakter) *http://www.oreilly.de:80* eintippen. Dass Sie sich als normaler Nutzer diese Arbeit aber nicht zu machen brauchen und nicht die zugehörigen Ports von hunderten wichtiger Dienste im Kopf behalten müssen, verdanken Sie der Standardisierung. Da der Browser weiß, dass Webserver unter Port 80 anzusprechen sind, nimmt er die Ergänzungen im Hintergrund für Sie vor, bevor er den jeweiligen Host anspricht. Um exakt zu sein, verhält es sich noch ein wenig anders, denn es ist das *http* vor der Internetadresse, das dem Browser mitteilt, dass im folgenden das http-Protokoll zur Kommunikation mit einem WWW-Dienst benutzt wird.

Tabelle 2-2: Die Standardports einiger wichtiger Dienste

Dienst	Port
WWW	80
DNS	53
E-Mail	110 (POP) 25 (SMTP)
FTP	20, 21
Telnet	23

Obwohl alle Ports gleichwertig sind, unterscheidet man dennoch nach der Art der Verbindlichkeit, mit der einem Dienst ein bestimmter Port zugeordnet ist, drei verschiedene Gruppen:

- Well Known Ports: Portnummern zwischen 0 und 1023.
- Registered Ports: Portnummern zwischen 1024 und 49151.
- Dynamic oder Private Ports: Portnummern zwischen 49152 und 65535.

Im Bereich der *Well Known Ports* liegen historisch ältere, vor allem aber essenzielle und sehr stark verbreitete Dienste. Dieser Bereich ist besonders wichtig, da er der strengsten Standardisierung unterliegt. Die Ports in dieser Gruppe sind einem bestimmten Dienst fest zugeordnet, und sowohl der Benutzer als auch Programme wie Browser, Mail-Tools oder Newsreader können in der Regel sicher sein, den angenommenen Dienst dort zu finden. Böswillige Dienstbetreiber können allerdings auch in diesem Bereich die Zuordnungen ändern.

Da es aber weit mehr als nur 1.024 wichtige Dienste gibt, wurde die Gruppe der *Registered Ports* als ein zusätzlicher Bereich geschaffen. Auch hier ist die Zuordnung zwischen Portnummer und Angebot zwar durch die Organisation IANA (*Internet Assigned Numbers Authority*, *http://www.iana.org/assignments/port-numbers*) schriftlich geregelt, dennoch ist die Verbindlichkeit nicht mehr so stark wie im Fall der Well Known Ports. So sollten Sie als Benutzer nicht unbedingt voraussetzen, dass sich hinter einer bestimmten Nummer auch der gewünschte Dienst verbirgt.

Den letzten Bereich bilden die *Private Ports*, bei denen es keine feste Zuordnung zwischen einem Dienst und der Portnummer mehr gibt. Daher finden Sie in diesem Bereich vor allem weniger verbreitete oder sehr spezielle Dienste. Zudem können Sie sich nicht darauf verlassen, dass auf verschiedenen Systemen diese Ports gleich belegt sind.

Kommen wir nach diesem kurzen Ausflug aber zurück zum Zuordnungsproblem bei der Kommunikation von Clients und Servern. Dank den Ports auf der Serverseite kann der Client den benötigten Dienst nun einwandfrei identifizieren und ansprechen. Doch wie verhält es sich mit der Antwort? Auch auf der Seite des Clients können mehrere Verbindungen gleichzeitig gehalten werden, etwa dann, wenn Sie zwei verschiedene Internetseiten gleichzeitig aufrufen oder nebenbei einen Download ausführen. Auch hier kommt wieder das Prinzip der Ports zum Tragen. Wenn ein Client eine Verbindung zu einem bestimmten Dienst aufbauen möchte, schickt er neben der Anfrage an den Port des Servers auch einen Verweis auf einen eigenen, noch freien Port im Bereich über 1.023 mit. An diesem Port lauscht nun der Client auf die Antwort des Servers. Damit wäre eine eindeutige Zuordnung der Kommunikation zwischen genau einem Client und einem Server hergestellt.

Doch wie unterscheidet der Server nun mehrere Verbindungen voneinander? Jeder Computer im Internet kann durch eine bestimmte Adresse eindeutig identifiziert werden. Diese Adresse wird *IP-Adresse* genannt und ermöglicht zusammen mit dem Port eine genaue Zuordnung der Clients, die gleichzeitig auf den Server zugreifen. Wenn wir im vorigen Abschnitt also geschrieben haben, dass eine Verbindung immer aus zwei Kommunikationsendpunkten besteht, können wir diese nun näher beschreiben und sagen: Jede Verbindung wird durch je einen so genannten *Socket* (Zuordnung aus IP-Adresse und Port) auf Client- und Serverseite eindeutig identifiziert. Damit z. B. ein von vielen Clients gleichzeitig angesprochener Server die einzelnen Anfragen unterscheiden und jedem Partner das angeforderte Dokument

übermitteln kann, ordnet er die Systeme durch die Kombination IP-Adresse:Port eindeutig zu und kann so auch Clients unterscheiden, die denselben Dienst nutzen. Im Abschnitt »Adressierung im Internet« weiter hinten im Kapitel werden Sie mehr über dieses Thema erfahren.

Protokolle

Nachdem nun die Funktionsweise von Ports und Diensten geklärt ist, stellt sich als Nächstes die Frage, in welcher Sprache die Hosts miteinander kommunizieren. Da das Internet, wie erwähnt, sehr heterogen ist, muss auch hier auf genau spezifizierte Standards zurückgegriffen werden. Daher läuft die Datenübertragung im Internet mit Hilfe so genannter *Protokolle* ab. Diese ließen sich wohl am besten mit einer Art Regelsammlung vergleichen, in der festgelegt wird, nach welchem Muster die Kommunikation ablaufen soll. Solche Mechanismen für die Datenübertragung gibt auf der Ebene der Verkabelung, der Netzwerk-Hardware, des Betriebssystems und der konkreten Anwendungen. Die einzelnen Ebenen (*Layer*) und Protokolle werden wir uns im Abschnitt »TCP/IP-Protokollfamilie« genauer anschauen und wollen daher hier erst einmal die Idee von Protokollen im Allgemeinen betrachten.

Grundsätzlich steht ein Protokoll nie für sich allein, sondern bildet mit zahlreichen anderen eine Familie, die *Stack* (Stapel) genannt wird. Dieser Stack wird als vollständig bezeichnet, wenn er alle Ebenen der Internetkommunikation abdecken kann, also die Verkabelung, das Netzwerk, das Betriebssystem und das konkrete Programm. Der Datenverkehr in einem solchen Stapel läuft immer auf die gleiche Weise ab: Wenn ein Client beispielsweise eine E-Mail vom Server abrufen möchte, verpackt er die eigentliche Abfrage in das Protokoll, das für die Kommunikation mit Mailservern verantwortlich ist: das Post Office Protocol (POP). Anschließend wird das so gekapselte Datenpaket in den Protokollen der niedrigeren Schichten bis hin zu der eigentlichen Netzzugangsschicht immer wieder verpackt und anschließend vom tiefsten Layer des Clients an den des Servers übermittelt. Dort läuft der Weg genau andersherum, und das Paket wird immer weiter entpackt, bis die enthaltene Information schließlich bei der Mailserver-Anwendung angekommen ist.

Dieses mehrfache Verpacken mag auf den ersten Blick umständlich erscheinen. Es ist jedoch sehr effektiv und darüber hinaus auch die einzige Möglichkeit, die Heterogenität des Internets zu überwinden. Erstens werden die Kommunikationsinhalte von der eigentlichen Verbindungslogik entkoppelt und zweitens sorgt man dafür, dass jeder Layer, also jeder Teilabschnitt der Kommunikation, sich nur um den Teil der Übertragung kümmert, von dem er auch etwas versteht.

Wie Ihnen aufgefallen sein wird, sprechen wir bei der Kommunikation im Internet von *Datenpaketen*. Die Inhalte werden also nicht am Stück, sondern in Fragmenten übertragen. Der Vorteil dieser Technik liegt darin, dass erstens auf diese Weise auch die Kommunikation ultraschneller Systeme mit verhältnismäßig langsamen Part-

nern problemlos möglich wird, und zweitens bei einer Störung der Verbindung nicht der gesamte Inhalt, sondern nur das verloren gegangene Paket neu übertragen werden muss. Einen weiteren Gewinn dieser Lösung werden wir erst in einem späteren Abschnitt kennen lernen. Er betrifft die Wegesuche im Internet und eröffnet den Partnern die Möglichkeit, die einzelnen Pakete über verschiedene Routen zu verschicken und somit schnell auf Veränderungen im Netzwerk reagieren zu können.

Verbindungsorientierte Protokolle

Kommen wir mit diesem Wissen nun noch einmal zurück zu den Protokollen an sich. Im Internet unterscheidet man zwischen den so genannten verbindungs- oder zustandslosen und den verbindungsorientierten Protokollen.

Verbindungsorientierte Protokolle kommen immer dann zum Einsatz, wenn größere Datenmengen übertragen werden müssen oder die Kommunikation länger bestehen bleiben soll. Als wichtigster Vertreter dieser Protokollart ist das *Transmission Control Protocol* (TCP) zu nennen, das wir in einem späteren Abschnitt noch detaillierter besprechen werden. Hauptcharakteristikum dieser Protokolle ist, dass sie eine Verbindung vor dem eigentlichen Datenaustausch etablieren. Ähnlich wie beim Telefonieren muss der Angerufene die Verbindung durch das Abnehmen des Hörers erst bestätigen, bevor das Gespräch beginnen kann. Beide Kommunikationsendpunkte einigen sich also zuerst auf den Beginn der Übertragung und beenden diese auch erst nach gemeinsamem Einvernehmen wieder.

Jedes verschickte Datenpaket enthält zudem eine eigene Identifikationsnummer, mittels derer es im Gesamtstrom eindeutig zugeordnet und als empfangen bestätigt werden kann. Sollte daher einmal ein Paket wegen besonderer Umstände sein Ziel nicht erreichen, merkt die Gegenstelle anhand der eigentlich aufeinander folgenden Nummern, dass ein Segment verloren gegangen ist, und wird es daraufhin noch einmal anfordern. Dadurch kann zum einen garantiert werden, dass die Pakete im Endeffekt alle ans Ziel gelangen, und zum anderen, dass sie in der richtigen Reihenfolge weiterverarbeitet werden.

Diese Transportsicherheit bringt aber nicht nur Vorteile mit sich, sondern hat auch ihre Schattenseite. Da jedes Datenpaket neben dem eigentlichen Inhalt alle für die Übertragung nötigen Informationen enthält, ist der so genannte *Overhead*, also der Datenüberschuss an Verwaltungsinformationen, sehr groß, worunter vor allem die Geschwindigkeit leidet.

Zustandslose Protokolle

Im Gegensatz zu den verbindungsorientierten Protokollen könnte man die *zustandslosen Protokolle* eher mit einem Fax oder dem klassischen Paperboy vergleichen. So wie dieser seine Zeitung vom Fahrrad in Richtung Haus wirft, stellt auch ein zustandsloses Protokoll die Übertragung nicht sicher, kann also nicht für den Emp-

fang der Daten garantieren. Da verbindungslose Protokolle weder über Identifikationsnummern noch über Mechanismen zur Aushandlung von Verbindungen verfügen, kann es vorkommen, dass einzelne Pakete verloren gehen und die Übertragung daher unvollständig ist, da die fehlenden Segmente nicht nachgeliefert werden. Ebenso ist es möglich, dass die Daten ihr Ziel überhaupt nicht erreichen. Das Resultat ist ein zwar unzuverlässiges, aber sehr schnelles Übertragungsverfahren. ››Unzuverlässig‹‹ bedeutet in diesem Zusammenhang nicht, dass man solchen Protokollen nicht trauen sollte oder sie für die Kommunikation nicht geeignet wären, sondern nur, dass es keine Übertragungsgarantie gibt. Zustandslose Protokolle werden meist für die Übertragung kurzer Inhalte verwendet. Ein weiterer Verwendungszweck sind so genannte *Streaming Media*, wie z.B. Video oder Internet-Radio, wo bereits während der Übertragung der Daten mit dem Abspielen begonnen wird. Bei der Übertragung dieser Inhalte ist es zum einen zu verkraften, wenn das eine oder andere Datenpaket unterwegs verloren geht, zum anderen wäre eine Nachlieferung verlorener Teile hier sinnlos.

Zustandslose Protokolle kommen vermehrt innerhalb von LANs (*Local Area Networks*) vor, da hier der Übertragungsweg als sicher angesehen werden kann. Die Art und Anzahl der Rechner sowie die Route sind bekannt, und Hardware- und Verkabelungsfehler können ausgeschlossen oder zumindest schnell eliminiert werden. Aber auch im Internet spielen verbindungslose Protokolle eine wichtige Rolle. So basiert der für die gesamte Kommunikation essenzielle *Domain Name Service* (DNS) auf dem zustandslosen *User Datagramm Protocol* (UDP).

TCP/IP-Protokollfamilie

Ohne zu tief ins Detail zu gehen, werfen wir nun einen Blick auf die zentrale Protokollfamilie des Internets, die TCP/IP-Protokollfamilie. Dieses Wissen ist für das Verständnis der Sicherheitsproblematik im Internet essenziell. Sie brauchen sich die Feinheiten jedoch nicht einzuprägen – was zählt ist, ein Gefühl dafür zu entwickeln, wie das Netz arbeitet.

TCP/IP ist eigentlich der Name zweier verschiedener Protokolle. Da diese aber von zentraler Bedeutung für die Struktur des Internets sind, wurde nach ihnen der gesamte Protokollstapel benannt. Das Schichtenmodell von TCP/IP kennt vier Ebenen (Layer) und deckt diese mit jeweils mindestens einem Protokoll ab (siehe Abbildung 2-1).

Network Layer (Netzzugang)
: Die Netzzugangsschicht bildet die unterste Ebene der Kommunikation in der TCP/IP-Welt, sie sorgt für den Versand der Daten über die jeweils zugrunde liegende Netzwerkarchitektur. So müssen die Daten für eine ISDN-Übertragung beispielsweise anders verpackt werden, als wenn sie über Ethernet laufen würden.

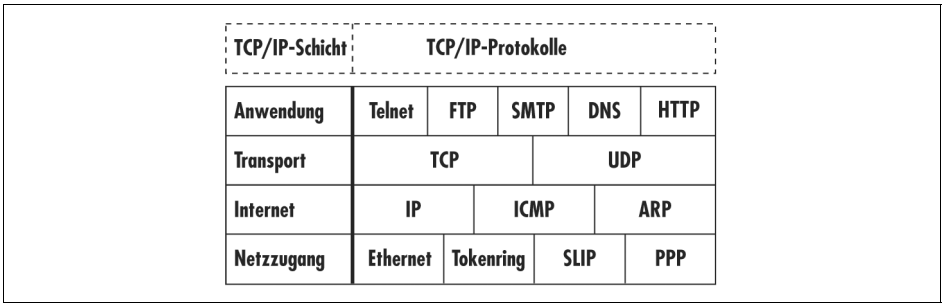

Abbildung 2-1: Auswahl einiger Protokolle mit Angabe der Schicht, in der sie gültig sind

In dieser Schicht wird außerdem die endgültige Paketgröße festgelegt, mit der die Daten über das Netz wandern. Diese Größe liegt z.B. im Fall von Ethernet bei 1.500 Bytes pro Frame (Paket); sie wird auch als *Maximum Transfer Unit* (MTU) bezeichnet.[4] Größere Pakete aus anderen Schichten oder Protokollen werden folglich zurechtgestutzt und auf mehrere Frames aufgeteilt.

Internet Layer (Internet)

Die Internetschicht liegt direkt über dem Network Layer und ist vor allem für das *Routing*, also die Wegewahl, zuständig. Hier wird das Paket für den Transport von Computer zu Computer vorbereitet, und die Daten werden mittels einer so genannten *IP-Nummer* adressiert. Das zentrale, zustandslose Protokoll auf dieser Ebene trägt den Namen *Internet Protocol* (IP). Wir werden ihm seiner großen Bedeutung wegen noch öfter begegnen.

Transmission Layer (Transport)

Während der Internet Layer auf der Ebene der Computer arbeitet, erfüllt die Transportschicht ihre Aufgaben auf der Ebene der Prozesse. Sie sorgt für die Kommunikation der einzelnen Anwendungen untereinander. Als Protokolle sind hier TCP und UDP zu nennen. Auf dieser Ebene werden im Fall von TCP beispielsweise die Identifikationsnummern und weitere Verbindungsinformationen angehängt.

Application Layer (Anwendung)

Auf der obersten Ebene befinden sich die eigentlichen Anwendungen wie der bereits erwähnte *Domain Name Service* (DNS) oder das *Hypertext Transfer Protocol* (HTTP), das für die Übertragung von Webseiten verantwortlich ist. Als Benutzer haben Sie meist nur Kontakt mit dieser Ebene. Zu dem Zeitpunkt, zu dem sich die Daten auf dieser Ebene befinden, sind sie nur in einer Hülle verpackt und können daher bei den meisten Protokollen noch von Menschen im Klartext gelesen und verstanden werden.

4 Die MTU ist kein fester Wert, sondern von Architektur zu Architektur verschieden. So kann es also durchaus dazu kommen, dass ein Paket auf seinem Weg von Host zu Host mehrmals das Übertragungsmedium und somit auch die Paketgröße wechselt. Wenn Sie z.B. von zu Hause aus per ISDN Daten senden, können die einmal beim Provider angekommen Informationen fortan über Ethernet transportiert werden.

Transmission Control Protocol (TCP)

Wie bereits beschrieben, liegt TCP im Transmission Layer der TCP/IP-Familie. TCP ist verbindungsorientiert und verursacht daher einen sehr großen Overhead; die damit erkaufte Übertragungssicherheit macht es jedoch zu einem der wichtigsten Protokolle des Internets. So nutzen zum Beispiel der Mail-, der WWW- und der File Transfer-Dienst dieses Protokoll. All diesen ist gemeinsam, dass eine verhältnismäßig große Datenmenge übertragen werden muss und der Ausfall von einzelnen Datenpaketen den gesamten Inhalt zunichte machen würde. Um etwas genauer zu verstehen, wie ein solches Protokoll überhaupt aufgebaut ist und wie Verbindungen ausgehandelt werden, wollen wir im Folgenden näher auf die Eigenarten des Transmission Control Protocol eingehen.

Schauen wir uns dazu zuerst das in Abbildung 2-2 dargestellte TCP-Segment an (so werden die TCP-Pakete genannt), um zu verstehen, welche Informationen in der Übertragungssicherheit spezifiziert werden. Dabei betrachten wir nicht den eigentlichen Inhalt des Pakets, sondern nur die Verpackung (Header), in der dieses gekapselt ist.

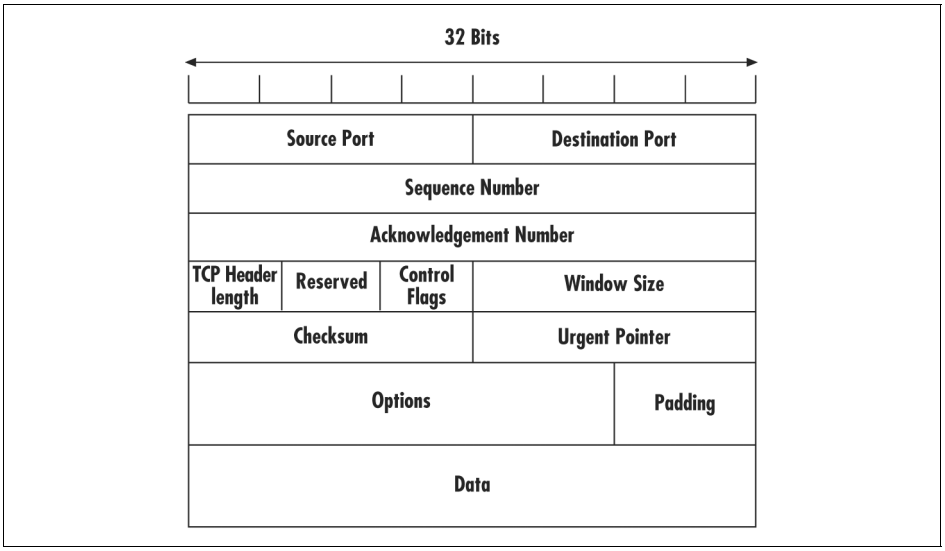

Abbildung 2-2: Der Header eines TCP-Segments

Source & Destination Port
: Diese beiden Felder beschreiben die Ports, zwischen denen die Kommunikation ablaufen soll. Dabei steht zum Beispiel bei der Übertragung einer Internetseite das erste Feld für den Port auf der Servermaschine, und das zweite beschreibt den Port auf dem Client. Bei der Anfrage einer neuen Seite verhält es sich umgekehrt.

Sequence & Acknowledgement Number
: Sie sind der Schlüssel für die Zuverlässigkeit von TCP. In der Phase des Verbindungsaufbaus werden hier die Startnummern für die Verbindung festgelegt. Während der Datenübertragung wird mittels dieser Nummern überprüft, ob Segmente tatsächlich beim Empfänger eingetroffen sind.

Flags
: Im TCP-Header gibt es sechs verschiedene *Flags*; sie drücken die Aufgabe des entsprechenden Segments aus und spielen unter anderem beim Verbindungsaufbau eine große Rolle. Die Flags bestehen aus einem Bit und können somit nur den Wert 0 oder 1 annehmen. Der Wert 1 bedeutet dabei, dass dieses Flag aktiv ist. So signalisiert beispielsweise ein gesetztes SYN-Flag, dass es sich bei dem Paket um einen Verbindungsaufbauwunsch handelt, während das FIN-Flag für einen Abbauwunsch steht.

Data
: Das Data-Feld gehört nicht mehr zum Header, es enthält die eigentlichen Informationen (Body).

Verbindungsaufbau

Wie beschrieben, handelt es sich bei TCP um ein verbindungsorientiertes Protokoll, bei dem sich die beiden Partner erst auf den Aufbau einer Verbindung einigen müssen. Dieser Vorgang wird als *Three-Way-Handshake* bezeichnet und basiert auf dem Zusammenspiel der oben erwähnten Flags (siehe Abbildung 2-3). Der Client sendet einen Verbindungsaufbauwunsch an den Server. Er zeichnet sich durch ein gesetztes SYN-Flag und die Sequenznummer aus, mit der die Verbindung initialisiert werden soll.

Der Server antwortet seinerseits mit einem gesetzten SYN- und ACK-Flag sowie seiner eigenen Sequenznummer. Zudem bestätigt er die Nummer des Clients, indem er sie um 1 hochzählt und als Acknowledgement Number zurückschickt. Die Verbindung gilt jetzt als halb offen. Der Client antwortet nun wiederum mit einem ACK-Flag und der erhöhten Sequenznummer des Servers sowie seiner eigenen nächsten Sequenznummer. Damit gilt die Verbindung als offen. Alle weiteren Pakete, die zwischen den Kommunikationspartnern ausgetauscht werden, enthalten ein gesetztes ACK-Flag. Die SYN-Flags werden während der gesamten Verbindung nicht mehr benutzt.Bei Interesse finden Sie die genauen Spezifikationen des TCP in den *Requests for Comment* (RFC) 793, 1122 und 1323.[5]

5 Ein *Request for Comment* ist ein Dokument, in dem Standards und Protokolle veröffentlicht werden. Entgegen seiner Bezeichnung wird ein RFC erst nach Abschluss der Diskussion herausgegeben und fungiert von da an als Standard für das Internet. RFCs können z.B. von der Seite *http://www.rfc-editor.org* bezogen werden.

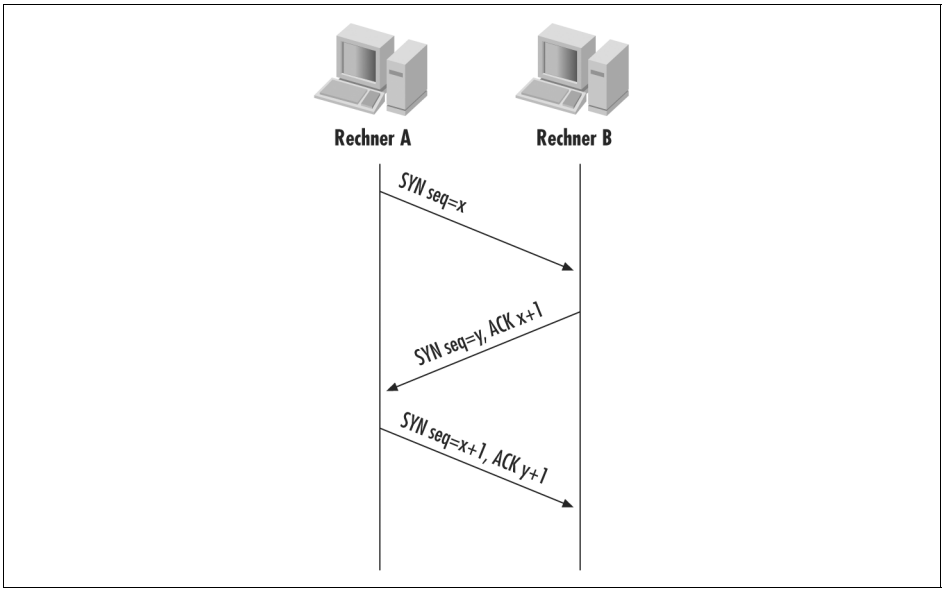

Abbildung 2-3: Three-Way-Handshake bei TCP

User Datagram Protocol (UDP)

Im Gegensatz zu TCP handelt es sich bei UDP um ein zustandsloses Protokoll, das vor allem durch seinen geringen Overhead und die damit verbundene hohe Übertragungsgeschwindigkeit glänzt. UDP kommt daher bei der Übertragung von Video-Streams und Internet-Radio zum Einsatz, wo zum einen viel Inhalt in wenig Zeit übertragen werden muss und zum anderen den einzelnen Paketen keine zentrale Bedeutung für den Gesamtinhalt zukommt.

Der Header ist bei UDP wesentlich schlanker als bei TCP und besteht im Prinzip nur aus den Angaben der beiden Ports sowie der Länge und Prüfsumme des Pakets. Mechanismen wie Flags und Sequenznummern fehlen hier vollständig, da UDP nicht zwischen verschiedenen Zuständen unterscheiden kann und auch nicht überprüft, ob alle Pakete angekommen sind.

Internet Protocol (IP)

Das verbindungslose Internet Protocol ist im Internet-Layer angesiedelt und hier für die Adressierung und das Routing (die Wegewahl) zuständig. Während also die Protokolle auf der Transportebene für die Kommunikation zwischen einzelnen Prozessen (z.B. Diensten) zuständig sind, sorgt IP für die Übermittlung der Daten von Computer zu Computer. Begriffe wie Ports, die bereits mit der Anwendungslogik zusammenhängen, spielen beim IP keine Rolle. Stattdessen finden sich aber Infor-

mationen über den Quell- und Zielhost im Kopf des Datenpakets. Abbildung 2-4 zeigt den Header eines IP-Segments.

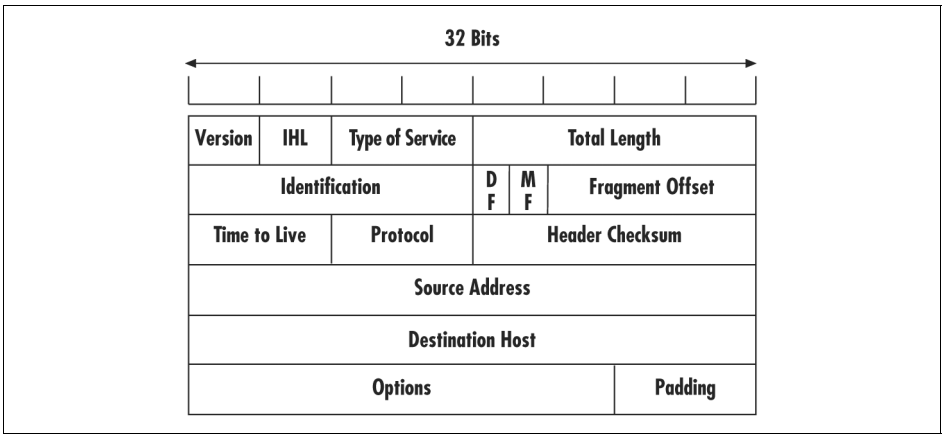

Abbildung 2-4: Der Aufbau des IP-Headers

Adressierung im Internet

Damit ein Computer im Internet überhaupt erreicht werden kann, muss er über eine Adresse verfügen. Da das Internet aber rund um die Welt aus vielen einzelnen Netzwerken besteht, ergibt sich hier ein Problem der Eindeutigkeit. So ist es beispielsweise möglich, dass zwei verschiedene Hosts in unterschiedlichen Netzen den gleichen Namen tragen. Versucht man einen der beiden zu erreichen, stellt sich die Frage, welcher nun konkret gemeint ist. Daher muss es neben dem für Menschen leicht zu merkenden Namen noch eine andere Adressierungsmöglichkeit geben, die für jeden Rechner eindeutig ist. Aus dieser Adresse muss nicht nur der Host selbst, sondern auch das Netz, in dem er sich befindet, hervorgehen, denn sonst gäbe es keine Möglichkeit, ihn unter Millionen von anderen Systemen ausfindig zu machen.

Da Computer mit Zahlen wesentlich besser umgehen können als mit alphanumerischen Ausdrücken und sich auf Basis der Zahlen z.B. auch Berechnungen über den »Ort« eines Hosts in einem Netzwerk durchführen lassen, hat man sich für eine Adresse entschieden, die aus insgesamt zwölf numerischen Stellen besteht. Diese so genannte *IP-Adresse* oder *IP-Nummer* ist in vier Blöcke gegliedert, die jeweils aus einer Zahl zwischen 0 und 255 bestehen und durch Punkte voneinander getrennt sind. Das erlaubt sowohl die eindeutige Identifizierung eines einzelnen Hosts als auch die des dazugehörigen Netzwerks. Zu diesem Zweck charakterisiert nur ein Teil der Blöcke den Rechner selbst, während der andere stellvertretend für das Netz steht. Wie groß der jeweilige Teil ist, hängt dabei von der Gesamtanzahl an Systemen ab, die in einem gemeinsamen Bereich zusammengefasst werden sollen.

Abbildung 2-5: Der Host 192.168.13.12 aus dem privaten Klasse-C-Netz 192.168.13.0

So reicht bei einem verhältnismäßig kleinen Netz mit einer Maximalanzahl von 254 Hosts der letzte der vier Ziffernbereiche aus, während die drei vorderen für die Adressierung des Netzwerks zuständig sind. Diese Zahl ergibt sich daraus, dass jeder einzelne Block genau 8 Bit groß ist und somit nur Zahlen bis 256 dargestellt werden können, wobei die erste und letzte aus Verwaltungsgründen nicht vergeben werden dürfen. Ein Netz, bei dem nur die beiden ersten Bytes das Netzwerk charakterisieren, kann also 65.534 [(256 x 256) -2] Computer eindeutig identifizieren.

Der Größe nach unterscheidet man daher drei verschiedene Typen von Netzen. Bei Klasse-A-Netzen beschreibt nur das erste Oktett das Netzwerk, so dass hier insgesamt über 16 Millionen Einzelsysteme adressiert werden könnten. Es gibt aber weltweit nur 125 Netzwerke dieser Größe. Eine mögliche Adresse aus diesem Bereich wäre die 126.10.10.10, wobei die 126 das Netz angibt, während die 10.10.10 einen speziellen Computer identifiziert. Neben diesen riesigen Netzen gibt es noch die bereits indirekt genannten B- und C-Klasse-Netzwerke mit maximal je 65.534 Rechnern in 16.336 verschiedenen Netzen beziehungsweise je 254 Hosts in über 2 Millionen Netzwerken. Aus technischen und administrativen Gründen werden die hier genannten Netze wiederum in kleinere Subnetze aufgeteilt, die meist aus nicht mehr als ein paar Dutzend Rechnern bestehen.

Zwei weitere wichtige Aspekte sollen hier nicht unerwähnt bleiben. Eine direkte Kommunikation der Hosts untereinander kann immer nur im selben Netz stattfinden. Das heißt, dass ein Computer, der sich in einem anderen Netzwerk befindet als Ihr System, nicht direkt von Ihnen angesprochen werden kann. Daher muss es Vermittler geben, die sowohl Mitglied des einen wie auch des anderen Netzwerks sind und Ihre Daten an den jeweiligen Computer weiterleiten. Diese Vermittler werden *Router* oder auch *Gateway* genannt (mehr dazu folgt im Abschnitt »Routing« weiter hinten in diesem Kapitel). Dies hat aber zur Folge, dass es zwangsläufig Hosts geben muss, die über mehr als nur eine IP-Adresse und Netzwerkkarte verfügen, da sie ja Teil von mindestens zwei Netzen sein müssen. Solange jedoch jede Adresse eindeutig einer bestimmten Maschine zugeordnet ist, ergeben sich hieraus keine Probleme.

Des Weiteren unterschiedet man zwischen öffentlichen und privaten Adressbereichen. Zu Anfang hatten wir ja gesagt, dass die IP-Adresse der eindeutigen Identifizierung von Hosts im Internet dient. Das trifft jedoch nur auf die so genannten *öffentlichen* Adressen zu. Diese sind im gesamten Internet eindeutig und können daher auch von jedem mit dem Internet verbundenen Computer aus erreicht werden.

Darüber hinaus gibt es *private* IP-Adressen, die nur in ihrem jeweiligen lokalen Netz gültig sind. Ein solches privates Netz kann der Eindeutigkeit wegen daher niemals direkt ans Internet angeschlossen werden. Ein privater Adressbereich wären beispielsweise alle Klasse-C-Netze, die sich in 192.168.x.y bilden lassen, also beispielsweise 192.168.13.0 mit Platz für 254 Computer. Es kann daher ohne Probleme diverse solche Netze mit denselben Netz- und Hostnummern geben, ohne dass es zu Konflikten kommen würde. Soll solch ein Netzwerk aber an das Internet angeschlossen werden, müssen entweder die IP-Adressen aller Hosts und des Netzes in öffentliche geändert werden, oder ein Router muss vor das private Netzwerk montiert werden und fortan alle Datenpakete, die ins Internet gelangen sollen, so umschreiben, dass sie über seine öffentliche Adresse verfügen. Dieser Vorgang wird *Masquerading* (eine öffentliche IP-Adresse für n verschiedene Rechner) oder *Network Address Translation* (NAT – mehrere (m) öffentliche IP-Adressen für n verschiedene Rechner) genannt. Wir werden in späteren Kapiteln darauf zurückkommen. Für die Rechner im Internet erscheint dann der Router als Endpunkt der Kommunikation, von seiner Rolle als Vermittler erfahren sie nichts.

Welchen Vorteil haben dann aber diese privaten Adressbereiche? Erstens ist die Anzahl an öffentlichen IP-Adressen und vor allem an öffentlichen Netzen langsam aber sicher erschöpft, was sich aber mit der nächsten IP-Version (IPv6) ändern wird. Zweitens kosten öffentliche Adressen zum Teil viel Geld. Für kleinere Unternehmen oder Heimnetzwerke macht es daher durchaus Sinn, die privaten Bereiche zu benutzen. Ein weiterer Grund liegt aber auch darin, dass man durch das Masquerading einen gewissen Schutz vor Angriffen aus dem Internet erhält, da Angreifer nicht direkt auf den PC hinter dem Router zugreifen können. Wie Angreifer diesen Schutz dennoch aushebeln, werden wir in den kommenden Kapiteln besprechen.

Wie verhält es sich nun aber mit Ihrem Computer zu Hause? Schließlich hatten wir ja gesagt, dass jeder, sogar nur kurz über eine Telefonleitung teilnehmende PC vollständig Teil des Internets wird und somit eine eindeutige Adresse haben muss. Um dies zu gewährleisten, teilt Ihnen der Provider, über den Sie sich ins Internet einwählen, eine gültige IP-Adresse mit. Während der anschließenden Online-Sitzung sind Sie dann unter dieser Adresse erreichbar. Da der Anbieter aber keine unbeschränkte Anzahl an Adressen zur Verfügung hat, ist diese Zuordnung nur zeitweilig gültig, also dynamisch. Die IP-Adresse ist also nicht für Ihren Computer reserviert, sondern wird Ihnen bei jeder Einwahl aufs Neue zugeteilt. Daher ist es zwar möglich, aber sehr unwahrscheinlich, dass Sie bei einer erneuten Einwahl dieselbe IP-Adresse noch einmal bekommen. Diese dynamische Vergabe hat den Vorteil, dass es Angreifern nicht unmittelbar möglich ist, herauszufinden, unter welcher Adresse Sie gerade im Internet surfen.

Domain Name Service (DNS)

Wie bereits erwähnt, sind Zahlenkolonnen für die Verarbeitung durch EDV-Systeme besser geeignet als alphanumerische Ausdrücke. Für uns Nutzer gilt jedoch das genaue Gegenteil, und so hat man sich entschieden, neben der numerischen Adressierung noch eine nominale zu wählen. Die Computernamen, die Sie aus der Adressleiste Ihres Browsers kennen, sind also nichts anderes als ein Kompromiss zu unseren Gunsten. Im Folgenden wollen wir näher darauf eingehen, wie solche Namen aufgebaut sind und in welchem Zusammenhang sie mit den IP-Adressen stehen.

Der Begriff der *Domain* bezeichnet die Gesamtheit von Hosts oder Netzen. Hinter einer Domain können sich also auch mehrere Netze verbergen. An der obersten Stelle im Domainkonzept steht die *Root*-Domain (Wurzeldomain). Eine Hierarchiestufe tiefer finden wir die so genannten *Top-Level-Domains*. Diese teilen den Namensraum in Länder oder Nutzungsfelder auf. Beispiele für Top-Level-Domains sind »de« für deutsche Domains oder »fr« für französische.

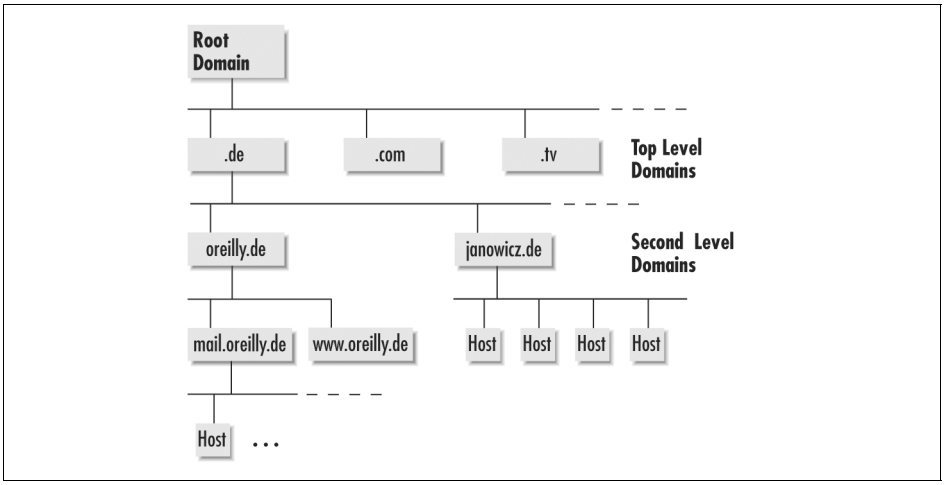

Abbildung 2-6: Die Domainstruktur im Internet

Die Namen der *Second-Level-Domains* dürften Ihnen vertraut sein, denn dabei handelt es sich um (nahezu) frei wählbare Namen wie beispielsweise »oreilly«. Darauf folgen entweder schon einzelne Hosts oder weitere Subdomains, in diesem Fall also *Third-Level-Domains*. Ein Beispiel wäre »java« aus der Domain *sun.com*. So entstehen also die im Internet allgegenwärtigen Namen wie *www.oreilly.de* oder das eben angesprochene *java.sun.com*. Im erstgenannten Fall ist also »www« der Name des Rechners, auf dem sich der Webdienst für den O'Reilly-Verlag befindet, während »oreilly.de« das Netz adressiert. An dieser Stelle werden die Parallelen zur eigentlichen IP-Adressierung sehr deutlich, da auch bei dieser aus dem vollständigen Sys-

temnamen sowohl Host als auch Netz abgeleitet werden können. Gleichzeitig löst sich damit auch das Problem der Eindeutigkeit von Gerätenamen, denn es gibt zwar Millionen Systeme, die allesamt »www« heißen, aber nur eines davon liegt im O'Reilly-Netz. So wie eine einzelne Maschine mehr als eine IP-Adresse besitzen kann, kann sie auch mehrere Namen haben; diese werden als *Alias* bezeichnet und dienen hauptsächlich administrativen Zwecken, z.B. wenn mehrere wichtige Dienste auf dem gleichen Rechner laufen. So wäre es für einen Benutzer verwirrend, wenn er seinen Mailserver unter *www.mein-server.de* ansprechen müsste. Daher verfügt der Server zusätzlich über den Alias »mail« und wird dann unter *mail.mein-server.de* angesprochen. Dabei steht der Name jedoch in keinem verpflichtenden Verhältnis zum angebotenen Dienst und kann neben *www* ebenso *java* oder *xyz* heißen.

Kommen wir nach diesem Exkurs zur Benennung von Computersystemen zurück zum eigentlichen Domain Name Service (DNS). Dabei handelt es sich im Prinzip um eine große Datenbank, in der die Zuordnungen von Namen und IP-Adressen gespeichert und verwaltet werden. Man kann bei diesem Dienst also erfragen, welche Adresse zu welchem Namen gehört und umgekehrt. Erinnern Sie sich an den Aufbau des IP-Headers im Abschnitt »TCP/IP-Protokollfamilie«: Dort befinden sich zwei Felder, in denen die IP-Adressen von Ziel- und Quellhost verzeichnet sind. Da der Benutzer aber beim Surfen den Namen eingibt, also zum Beispiel *www.oreilly.de*, muss der Computer diesen in die IP-Adresse umwandeln, um anschließend damit weiterarbeiten zu können. Dieser Vorgang wird als *Namensauflösung* bezeichnet. Man spricht davon, dass mittels des Domain Name Service Gerätenamen in IP-Adressen aufgelöst werden. Folglich muss der Computer wenigstens die IP-Adresse des DNS-Servers kennen. Daher wird diese fest in die Netzwerkeinstellungen des Betriebssystems eingegeben oder bei der Einwahl vom Provider mitgeteilt.

Bei Millionen von IP-Adressen und noch mehr Aliasen ist es aber undenkbar, dass ein einzelner DNS-Server diese verwalten oder gar aktuell halten könnte. Daher hat man die Zuordnungstabelle auf zahlreiche Systeme verteilt und geht dabei nach dem Prinzip vor, dass jeder Server für diejenigen Daten verantwortlich ist, die in seinem Verwaltungsbereich liegen. Der DNS-Server von O'Reilly beherbergt also zum Beispiel alle Hostnamen und IP-Adressen aus dem dazugehörigen Netzwerk. Damit ist erstens sichergestellt, dass die Daten vollständig sind, und zweitens, dass diese sich auch auf einem aktuellen Stand befinden. Schließlich hat der Netzbetreiber ja ein Interesse daran, dass sein eigenes Netz sauber funktioniert. Der Nachteil dieser Einteilung ist jedoch, dass nicht jeder DNS-Server eine Anfrage auflösen kann und daher einen weiteren Server zu Rate ziehen muss, um zu erfahren, wo er den für die gesuchte Domain verantwortlichen Server findet. Dafür fragt er einfach beim DNS-Server nach, der für die gesamte Top-Level-Domain (z.B. *.de*) oder sogar für die Root-Domain zuständig ist.

Da es sich hierbei um einen langwierigen Prozess handelt, wäre es uneffektiv, diese Daten verfallen zu lassen, und so merkt sich der DNS-Server die IP-Adresse samt Namen für einige Zeit, so dass er bei der nächsten Anfrage nach dem gleichen Sys-

tem nicht noch einmal nachfragen muss. Damit die Daten dennoch aktuell genug sind, verwirft er die Zuordnung nach einigen Stunden oder Tagen wieder. Schauen wir uns dazu ein kurzes Beispiel an.

Ein Benutzer möchte sich die Webseiten auf *www.beispiel.de* anschauen und tippt daher diesen Namen in die Adresszeile seines Browsers ein. Daraufhin wird sein Computer den ihm bekannten Nameserver des Providers ansprechen und nach der IP-Adresse von *www.beispiel.de* fragen. Da der DNS-Server des Providers aber weder für die Daten zuständig ist noch diese gespeichert hat, wendet er sich an den Nameserver der *.de*-Domain. Dieser kann die IP-Adresse zwar auch nicht auflösen, kennt aber die Adresse des *beispiel.de*-Nameservers und fragt bei diesem an. Hier erfährt er endlich die korrekte IP-Adresse, so dass der Computer unseres Benutzers nun den gewünschten Server ansprechen kann.

Routing

Bereits mehrfach ist in diesem Kapitel der Begriff *Routing* gefallen. Dabei handelt es sich um die Wahl der Strecke, die ein Datenpaket von Computer zu Computer zurücklegen muss, bevor es sein Ziel erreicht. Nötig wird dieser Vorgang dadurch, dass Hosts immer nur im eigenen Netz sichtbar sind, also von außen nicht unmittelbar angesprochen werden können.

Ein fremder Rechner kann zwar von der Existenz eines Systems wissen und sogar dessen IP-Adresse kennen, solange er aber nicht im selben Netz ist, kann er keinen Kontakt aufnehmen. Damit das Paket dennoch zugestellt werden kann, muss sich der Quellhost an den für sein Netz zuständigen Router wenden, also eine Maschine, die Mitglied zweier oder mehrerer verschiedener Netzwerke ist (siehe Abbildung 2-7).

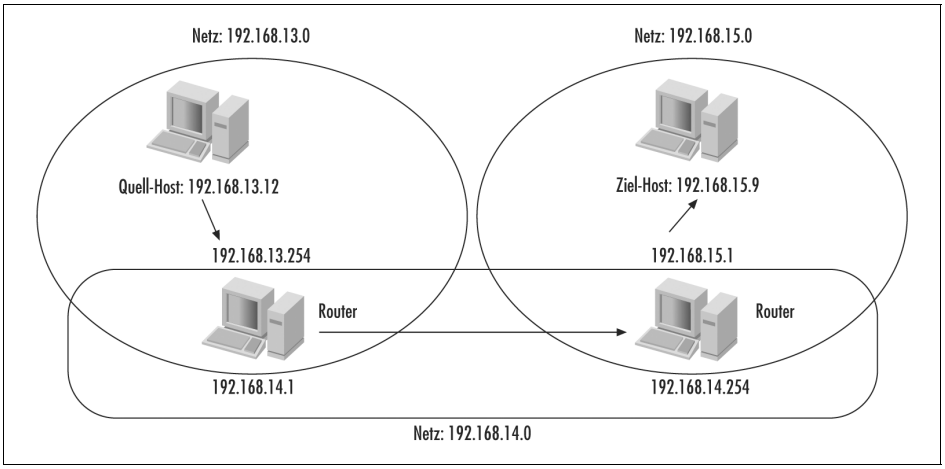

Abbildung 2-7: Die Route zwischen drei Netzwerken

Der Router übernimmt nun das Paket und überprüft, ob sich das Zielsystem in einem seiner eigenen Netze befindet. Wenn dem so ist, also der Netzteil der IP-Adresse von Router und Ziel übereinstimmt, kann er das Datenpaket direkt zustellen. Ansonsten sendet er es an einen anderen Router weiter, der wiederum prüft, ob das Paket das Zielnetz erreicht hat oder nicht.

Sie sehen also, dass das Vorhandensein einer Identifikationsmöglichkeit, die neben dem Host auch das zugehörige Netz adressiert, Routing überhaupt erst ermöglicht.

Zusammenfassung

Sie haben in diesem Kapitel die grundlegenden Komponenten der Kommunikation im Internet kennengelernt. Wir haben festgestellt, dass der Datenaustausch zwischen zwei Partnern im Internet mittels so genannter Dienste stattfindet. Die Partner lassen sich dabei nach ihrer Funktion in Clients und Server aufteilen. Die eigentliche Kommunikation funktioniert mit Hilfe von Protokollen, jenen Regelsammlungen, nach denen die Informationen verpackt und verschickt werden. Am Beispiel von TCP/IP wurde deutlich, welche Ebenen der Kommunikation es gibt und wie eine Kommunikationssituation zustande kommt (Three-Way-Handshake). Als Abschluss des Kapitels haben wir die Benennung der beteiligten Partner behandelt und die Wegewahl, die zum Transport der einzelnen Datensegmente notwendig ist.