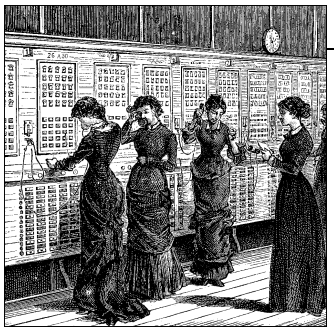

KAPITEL 3
Netzwerkgrundlagen

Ein Firewall-Buch wäre nicht komplett ohne eine Darstellung der Netzwerkprotokolle, die dazu dienen, Daten im Internet zu vermitteln. Man kann dieses Kapitel vielleicht mit der Grammatik einer Sprache vergleichen. Wie man eine Sprache nicht wirklich sprechen kann, ohne die Grammatik zu beherrschen, so kann man auch eine Firewall nicht sinnvoll konfigurieren, wenn man die Netzwerkprotokolle nicht versteht.

Firewalls funktionieren, indem sie die Daten, die sie weitervermitteln sollen, betrachten und gegebenenfalls ändern oder verwerfen. Damit die richtigen Daten weitervermittelt oder verworfen werden, müssen Sie Regeln aufstellen, welche Daten wie zu behandeln sind. Dies können Sie aber nur, wenn Sie verstehen, wie Daten in Rechnernetzen übertragen werden.

Auch die nachfolgenden Erörterungen verschiedener Angriffe erfordern ein gewisses Verständnis der Mechanismen, mit denen Daten in Netzen übertragen werden. Um zum Beispiel zu verstehen, wie es möglich ist, mit einem Netzwerkpaket eine ganze Flut von Paketen loszutreten und damit einen Zielrechner regelrecht lahmzulegen, muß man beispielsweise wissen, daß es Broadcast-Adressen gibt, mit denen man nicht nur einen einzigen, sondern eine Vielzahl von Rechnern erreicht.

Dieser Abschnitt wird daher versuchen, kurz die Frage zu klären, wie das Internet funktioniert. Die Darstellung wird sich dabei allerdings auf diejenigen Aspekte beschränken, die zum Verständnis der nachfolgenden Ausführungen nötig sind. Tiefergehende Einblicke in die Protokolle erlauben ihre Spezifikationen, die *Requests for Comments* (*RFC*s), die von manchen Linux-Distributionen unter */usr/doc/rfc* installiert werden. Alternativ können sie auch aus dem Internet heruntergeladen werden. Eine Liste mit Servern, die die aktuellen RFCs vorhalten, findet sich unter *http://dir.yahoo.com/Computers_and_Internet/Standards/RFCs*. Eine der dort aufgeführten Quellen ist *http://www.faqs.org/rfcs/*.

TCP/IP im Überblick

Um Ihnen dabei zu helfen, die Zusammenarbeit der vorgestellten Protokolle besser zu verstehen, sind in Abbildung 3-1 ihre Abhängigkeiten dargestellt.

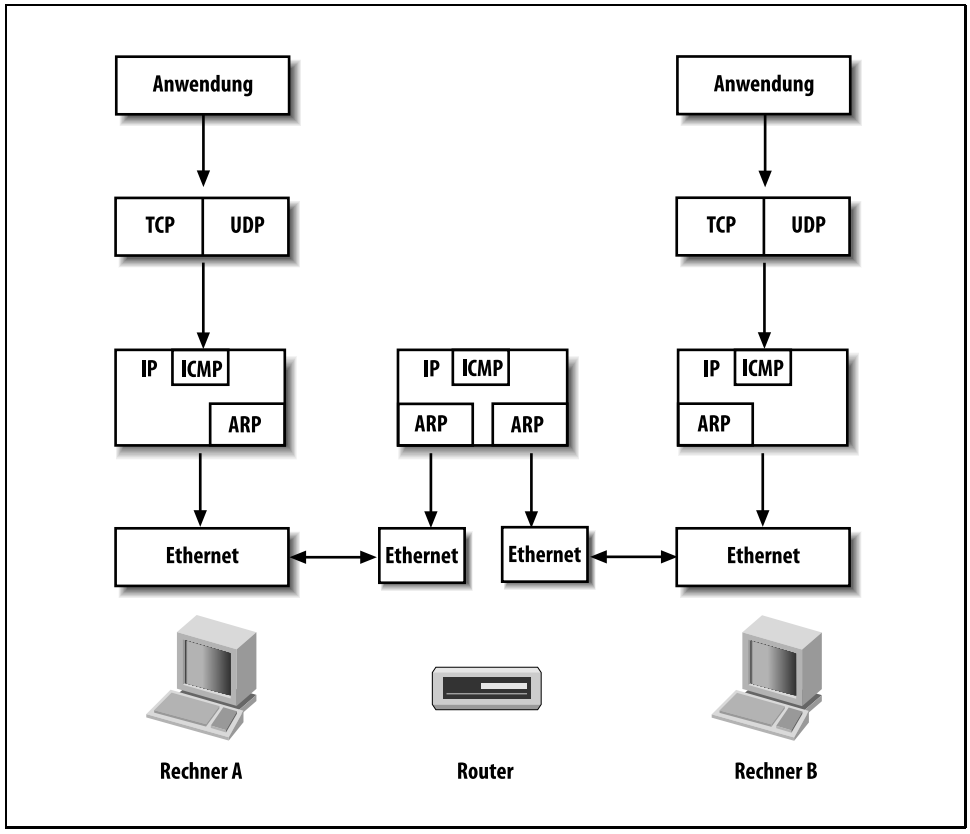

Abbildung 3-1: Protokollstack und Routing in TCP/IP

Beginnen wir mit unserer Betrachtung des Diagramms am unteren Bildrand. Dort befinden wir uns auf der Ebene der Hardware. In den beteiligten Rechnern befinden sich Netzwerkkarten, die über Kabel miteinander verbunden sind. Sie übertragen untereinander Daten, indem sie zu genau definierten Zeiten Spannungen auf den Kabeln erzeugen, die jeweils den zu übertragenden Nullen und Einsen entsprechen. Damit beide Karten miteinander kommunizieren können, ist es notwendig, im Detail zu definieren, wie die Datenübermittlung im einzelnen ablaufen soll. Es ist zum Beispiel zu klären, welche Spannungen Nullen bzw. Einsen darstellen, wie lange eine Spannung angelegt wird, um ein einzelnes Bit zu übertragen, und was geschieht, wenn mehrere an das gleiche Kabel angeschlossene Karten gleichzeitig senden.

Hat man all dies geklärt, so ist man in der Lage, einzelne Nullen und Einsen zu übertragen. In der Regel leisten Netzwerkkarten aber mehr als das. Netzwerke bestehen oft aus mehr als zwei Rechnern. Daher ist es notwendig, die zu übertragenden Daten in Blöcke, sogenannte *Pakete*, aufzuteilen, die zusätzlich Informationen wie z. B. Sender und Empfänger des Paketes enthalten. Man muß also definieren, wie groß die Pakete maximal sein dürfen, woran man gegebenenfalls den Anfang und das Ende eines Paketes erkennt und welche Bits des Paketes welche Bedeutung haben.

Eine solche Definition, wie die Kommunikation zwischen mehreren Partnern abläuft, nennt man *Protokoll*. Das Protokoll, in dem sich die momentan üblichen Netzwerkkarten unterhalten, nennt sich Ethernet. Allerdings existieren auch andere Protokolle.

Nun bietet das Ethernet-Protokoll bei weitem nicht alle Eigenschaften, die eine Anwendung benötigt. So kann es z. B. nur Pakete an Rechner übertragen, die direkt über ein BNC-Kabel oder einen Hub mit dem Sender verbunden sind. Da die Länge der einzelnen Netzwerkkabel begrenzt ist, bedeutet dies, daß Pakete nur innerhalb eines kleinen lokalen Netzes übertragen werden können. Dieses Problem wurde gelöst, indem man Software entwickelte, die Pakete aus einem lokalen Netz von einer Netzwerkkarte entgegennahm und sie über eine weitere Netzwerkkarte in ein anderes Netz weiterleitete. Diese Software benötigte aber zusätzliche Angaben. Die im Ethernet-Protokoll angegebene Empfängeradresse gibt nur den jeweils letzten Absender bzw. den jeweils nächsten Empfänger an, für eine Übermittlung über mehrere Zwischenstationen ist es aber auch nötig, den eigentlichen Sender und den letztendlichen Empfänger zu kennen.

Nun hätte man zu diesem Zweck sicherlich das Ethernet-Protokoll erweitern können. Allerdings war Ethernet zu dieser Zeit nicht das einzige verwendete Protokoll. Es konkurrierten diverse heute teilweise vergessene Protokolle, die sich sehr stark unterschieden. Sie benutzten z. B. vollkommen inkompatible Arten, Netzwerkadressen anzugeben, und besaßen unterschiedliche Maximalgrößen für die zu übertragenden Pakete. Man entschied sich deshalb dafür, ein neues Protokoll namens IP zu entwickeln, dessen Nachrichten im Datenteil von Paketen beliebiger Protokolle übertragen werden konnten.

Dieser Trend setzte sich später fort, so daß immer wieder neue Protokolle definiert wurden, welche die Funktionalität von bereits vorhandenen Protokollen nutzten, indem sie auf ihnen aufsetzten. Dazu definierten sie jeweils einen Kopf (*Header*) und einen Datenteil. Der Kopf enthält jeweils alle Angaben, die das Protokoll für seine Funktion benötigt, während der Datenteil die eigentlich zu übertragenden Daten enthält. Diese können ihrerseits wieder aus dem Header eines höheren Protokolls bestehen sowie aus einem Datenteil, der wieder einen Header eines noch höheren Protokolls enthält ...

Abbildung 3-2 zeigt dies an einem konkreten Beispiel. Hier wurde eine Webseite angefordert. Übertragen wird sie mit dem HTTP-Protokoll. Dieses versieht die Seite mit einem HTTP-Header und sendet sie an den Zielrechner. Dazu benutzt HTTP das TCP-Protokoll. Aus Sicht des HTTP-Protokolls besteht damit die Übertragung einer Webseite darin, ein Paket auf dem sendenden Rechner an TCP zu übergeben und es dann am empfangenden Rechner von ihm entgegenzunehmen.

TCP verfährt ähnlich. Es versieht das Paket, das aus der eigentlichen Webseite und dem HTTP-Header besteht, mit einem TCP-Header und übergibt es an IP. Es ist nun die

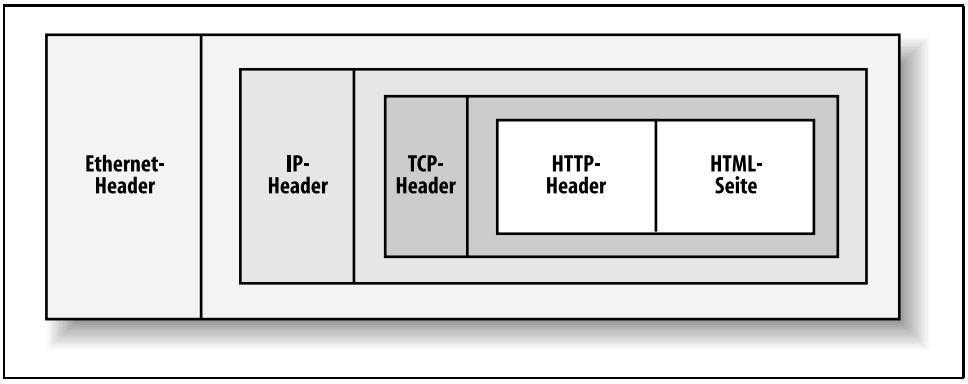

Abbildung 3-2: Paketstruktur am Beispiel eines HTTP-Paketes

Aufgabe von IP, dafür zu sorgen, daß das Paket den Zielrechner erreicht. Dort wird der IP-Header wieder entfernt und das Paket an TCP übergeben.

Aus Sicht von IP sieht die Übertragung etwas komplizierter aus. IP vermittelt Pakete, indem es sie mit einem IP-Header versieht und sie mittels Ethernet an den nächsten Router verschickt. Dort nimmt die dortige Instanz des IP-Protokolls das Paket entgegen und entscheidet anhand des IP-Headers, an welchen Rechner es weitervermittelt wird. Dazu wird wiederum das Ethernet-Protokoll verwendet. Am Zielrechner nimmt IP schließlich das Paket entgegen, entfernt den IP-Header und übergibt das Paket an TCP.

Wenn Ethernet das IP-Paket erhält, fügt es ebenfalls einen Header an. Das Paket, das dann tatsächlich über die Netzwerkleitung übermittelt wird, sieht damit schematisch wie in Abbildung 3-2 dargestellt aus.

Im folgenden werden wir uns nun etwas näher mit den einzelnen Protokollen oberhalb von Ethernet auseinandersetzen.

IP

IP, das Internet-Protokoll, bildet die Grundlage jeglicher Übertragung im Internet. Es wird im RFC 791 beschrieben und dient dazu, einzelne Datenpakete zwischen zwei Rechnern zu befördern. Allerdings ist es normalerweise nötig, Pakete über mehrere Zwischenstationen (Router) zu senden, bevor sie ihr Ziel erreichen. Diese Router haben die Aufgabe zu entscheiden, an welchen Rechner ein Paket weitervermittelt werden soll. Einfache Router treffen diese Entscheidung anhand einer fest eingestellten Tabelle, während komplexere Router mit Nachbarroutern Nachrichten austauschen, um ihre Tabellen dynamisch an die Gegebenheiten im Netz anzupassen. Auf diese Weise ist es möglich, alternative Wege zu finden, wenn ein Teil des Netzes ausgefallen ist.

Soll nun ein Rechner eine IP-Nachricht senden, so muß er als erstes entscheiden, ob das Paket für einen Rechner im lokalen Netz bestimmt ist. Ist dies der Fall, so kann es

direkt an ihn gesendet werden. Dazu bedient sich IP darunterliegender Protokolle wie z. B. Ethernet oder Token Ring. Diese sind spezifisch auf die Hardware abgestimmt und können in der Regel Pakete nur dann vermitteln, wenn der Zielrechner physikalisch (z. B. über einen BNC-Strang) mit dem Sender verbunden ist.

Läßt sich das Paket dagegen nicht direkt vermitteln, so muß der Sender es an einen zuständigen Router schicken. Hierbei handelt es sich um einen Rechner, der an mehrere Netzwerkstränge angeschlossen ist und Pakete zwischen ihnen vermittelt. Dieser muß dann seinerseits wieder entscheiden, auf welche Weise er es weitersenden kann. So wird das Paket dann Schritt für Schritt weitervermittelt, bis es seinen Empfänger erreicht.

Bei dieser Vermittlung kann es geschehen, daß das Paket in eine Schleife gerät. Dies bedeutet, daß es immer wieder zwischen denselben Routern hin und her gereicht wird, ohne jemals sein Ziel zu erreichen. Um in so einem Fall zu verhindern, daß das Paket bis in alle Ewigkeit weiterkreist, existiert in jedem IP-Paket ein Feld »Time to Live« (TTL). Dieses kann einen Wert zwischen 0 und 255 annehmen. An jedem Router wird der Wert in dem Feld mindestens um 1 verringert. Hat er 0 erreicht, wird das Paket verworfen und eine Fehlermeldung an den Empfänger gesendet.

Protokolle, die auf IP aufsetzen (z. B. TCP und UDP), können also Pakete an entfernte Rechner zustellen lassen, ohne dabei die Vermittlung der Pakete durch die Router berücksichtigen zu müssen. Aus ihrer Sicht stellt IP eine direkte Vermittlung zwischen zwei Rechnern her.

Da IP dazu konzipiert wurde, beliebige Protokolle zu benutzen, um die Daten zum jeweils nächsten Rechner zu übertragen, ist es nicht immer möglich, im vorhinein herauszufinden, welche maximale Paketgröße die auf den jeweiligen Teilstrecken verwendeten Protokolle erlauben. Damit besteht die Möglichkeit, daß ein IP-Paket für eine bestimmte Teilstrecke zu groß ist.

IP bietet daher die Möglichkeit, Pakete in kleine Teilpakete aufzusplitten, die dann beim Empfänger wieder zusammengesetzt werden. Diese Funktionalität wird *Fragmentierung* genannt.

Ein Überbleibsel aus der Zeit, als sich das Netz noch in seiner Entstehung befand, ist das *Source Routing*. Obwohl der Sender normalerweise nicht wissen wird, welche Route seine Pakete durch das Internet nehmen werden, besteht prinzipiell die Möglichkeit, mehrere Stationen vorzugeben, über die das Paket geroutet werden soll. Diese Funktionalität wird heute eigentlich nicht mehr benötigt. Auch sind viele Router so konfiguriert, daß sie derartige Vorgaben ignorieren. Allerdings wird noch immer versucht, diese Funktionalität für Angriffe zu nutzen.

Adressen werden IP in der Form w.x.y.z übergeben, wobei w, x, y und z für Zahlen zwischen 0 und 255 stehen (z. B. 10.0.0.1 oder 127.0.0.1). Dabei ist es üblich, Rechner in Subnetze einzuteilen. So könnten z. B. alle Rechner eines lokalen Netzes Adressen besitzen, die mit 192.168.0 beginnen. Beliebt ist dabei, die ersten 8, 16 oder 24 Bits der Netzwerkadresse als Netzwerkanteil zu nehmen. Prinzipiell ist dies aber nicht die einzige Möglichkeit. Auch ein Subnetz mit z. B. 17 Bits wäre möglich, es könnte z. B. alle Rechneradressen zwischen 192.168.128.0 und 192.168.255.255 enthalten.

Einige Adressen haben eine Sonderrolle. So beinhaltet 0.0.0.0 als Absenderadresse die Aussage: »Ich kenne meine eigene Adresse nicht«. Dies wird von Protokollen wie etwa DHCP verwendet, die dazu dienen, Rechnern automatisch eine Adresse zuzuweisen. Die Adresse 255.255.255.255 dagegen dient dazu, alle Rechner am selben Netzwerkstrang zu erreichen, man nennt sie auch *Broadcast-Adresse*. Meistens werden dabei allerdings nur alle Bits des Rechneranteils der Adresse auf 1 gesetzt. Für das Subnetz 192.168.0 wäre die Broadcast-Adresse damit 192.168.0.255. Die Adresse 192.168.0.0 würde soviel besagen wie: »Ich weiß, daß ich mich im Netz 192.168.0 befinde, aber wer bin ich?« Sie wird aber eher selten benutzt. Diese Adresse wird auch oft als »Adresse des Netzwerks« oder *Netzwerk-Adresse* bezeichnet.

Bevor eine IP-Adresse benutzt werden darf, muß sie zentral registriert werden. Dies ist nötig, um zu verhindern, daß mehrere Rechner im Internet dieselbe Adresse benutzen. Eine Ausnahme bilden Adressen aus den folgenden Bereichen, die frei in lokalen Netzen verwendet werden dürfen.

10.0.0.0 bis 10.255.255.255
172.16.0.0 bis 172.31.255.255 (oft unterteilt in 16 Subnetze 172.x)
192.168.0.0 bis 192.168.255.255 (oft unterteilt in 256 Subnetze 192.168.x)

Pakete mit Quell- oder Zieladressen aus diesen Bereichen werden von Routern im Internet üblicherweise ignoriert. Rechner, die solche Adressen benutzen, sind damit aus dem Internet nicht direkt erreichbar. Wir werden später sehen, wie wir diesen Rechnern trotzdem die Möglichkeit verschaffen können, auf Server im Internet zuzugreifen.

ARP

IP benutzt untergeordnete Protokolle, um Pakete zur jeweils nächsten Station des Weges weiterzuleiten. In lokalen Netzen wird dafür in der Regel das Ethernet-Protokoll eingesetzt. Allerdings kennt Ethernet keine IP-Adressen, sondern verwendet eigene Adressen, *MAC-Adressen* genannt. Diese sind anders aufgebaut und verwenden 6 statt 4 Bytes (z. B. 00:80:AD:18:AC:94). Um nun eine IP-Adresse in eine MAC-Adresse übersetzen zu können, benutzt IP das Address Resolution Protocol (ARP), das im RFC 826 definiert wird.

Hierbei wird eine Anfrage auf Ethernet-Ebene an alle Rechner gestellt. Diese enthält die MAC- und die IP-Adresse des Senders sowie die IP-Adresse des gewünschten Empfängers[1]. Dieser wird dann seine eigene MAC-Adresse in das Paket eintragen und es direkt an den Absender zurücksenden.

Es soll noch erwähnt werden, daß Netzwerkkarten normalerweise nur Pakete annehmen, die direkt an ihre MAC-Adresse oder an die Broadcast-Adresse unter Ethernet gesendet wurden. Pakete, die für andere Netzwerkkarten bestimmt sind, werden ignoriert. Um Eindeutigkeit sicherzustellen, wird jeder Karte bei der Herstellung eine eindeutige MAC-Adresse mitgegeben. Allerdings existieren Karten, bei denen diese später mittels Software verändert werden kann.

1 Wenn sich der eigentliche Empfänger nicht im selben Subnetz befindet, so wird hier die IP-Adresse des Routers genommen, der das Paket weitervermitteln soll.

Einen Sonderfall stellt der *Promiscuous Mode* dar. Wird eine Karte in diesen Zustand geschaltet, wird sie alle Pakete annehmen, gleichgültig für wen sie bestimmt sind. Dies wird insbesondere von speziellen Programmen zur Netzwerkdiagnose, den sogenannten *Sniffern*, genutzt. Wir werden später noch sehen, daß diese Programme auch recht gut dazu geeignet sind, Paßwörter zu protokollieren. Deswegen sind diese Programme nicht nur bei Netzwerkadministratoren, sondern auch bei Crackern überaus beliebt.

ICMP

ICMP, das Internet Control Message Protocol, erlaubt IP, Fehler- und Kontrollnachrichten zu übertragen. Es wird in RFC 792 beschrieben. Wichtige Nachrichten sind z. B. »Echo Request«, »Echo Reply«, »Destination Unreachable«, »Source Quench«, »Redirect«, »Router Solicitation«, »Router Advertisement«, »Time Exceeded« und »Parameter Problem«.

»Echo Request« bewirkt, daß mitgeschickte Daten vom Empfänger in einer ICMP-Nachricht vom Typ »Echo Reply« wieder zurückgeschickt werden. Dies dient in erster Linie der Kontrolle, ob ein Rechner erreichbar ist.

»Destination Unreachable« signalisiert, daß ein Paket nicht zugestellt werden konnte. Ein zusätzlicher Parameter gibt an, ob das Zielnetz, der Zielrechner oder der angesprochene Dienst auf dem Zielrechner nicht verfügbar ist.

»Source Quench« wird benutzt, wenn Pakete schneller ankommen, als der Zielrechner sie verarbeiten kann. Damit wird der Sender aufgefordert, das Senden vorübergehend einzustellen.

»Redirect« wird von Routern verwendet, um dem Sender mitzuteilen, daß er einen anderen Router verwenden soll. Hierbei muß sich der umleitende Router im selben Subnetz wie der Sender und der vorgeschlagene Alternativrouter befinden.

»Router Solicitation« erlaubt es einem Rechner nachzufragen, welche Router im lokalen Netz existieren. Jeder Router im Subnetz, der diesen Mechanismus unterstützt, wird dann mit einer »Router Advertisement«-Nachricht antworten. Den gesamten Vorgang bezeichnet man auch als ICMP Router Discovery Protocol (IRDP).

»Time Exceeded« wird gesendet, wenn der TTL-Zähler in einem Paket 0 erreicht hat und das Paket deswegen verworfen wurde.

»Parameter Problem« ist schließlich eine allgemeine Fehlermeldung für den Fall, daß ein beschädigtes oder ungültiges IP-Paket empfangen wurde.

TCP

TCP, das Transmission Control Protocol, wird in RFC 793 beschrieben und benutzt IP, um Pakete zwischen Anwendungen zu transportieren und sicherzustellen, daß definierte Paketfolgen in der richtigen Reihenfolge beim Empfänger eintreffen, ohne daß Pakete doppelt gesendet oder ausgelassen werden.

Der Sinn einer Adressierung von Anwendungen wird klar, wenn man sich vergegenwärtigt, daß auf einem Rechner verschiedene Serverdienste gleichzeitig aktiv sein können. Ohne ein Konzept, wie man diese adressieren kann, müßte jeder Dienst jedes Paket entgegennehmen und dann entscheiden, ob es für ihn bestimmt ist.

TCP führt hierzu das Konzept der *Portnummern* ein. Sowohl Senderprozeß als auch Empfängerprozeß reservieren sich beim Betriebssystem eine Portnummer. Auf diese Weise können sowohl die Anfragen zugestellt als auch die Antworten an den Absender zurückbefördert werden, da die Angabe von Serveradresse, Serverportnummer, Klientenadresse und Klientenportnummer eine Verbindung eindeutig beschreibt. Auch wenn ein Server auf demselben Port Anfragen von diversen Klientenprogrammen entgegennimmt, von denen einige vielleicht sogar denselben Rechner benutzen (wenn z. B. gleichzeitig Klienten verschiedener Hersteller eingesetzt werden), kann die Antwort immer eindeutig zugestellt werden (weil z. B. ein Klient Port 1024 zugewiesen bekam, während der andere Port 1025 benutzt).

Als Beispiel für das Portkonzept soll Abbildung 3-3 dienen. Hier haben wir einen HTTP- und FTP-Server, auf den mit drei Anwendungen vom selben Klienten aus zugegriffen wird. Es ergeben sich folgende Verbindungen:

Klient	Server	Verbindung
dummy:1024	wonderland:80	HTTP-Verbindung mit Firefox
dummy:1027	wonderland:21	FTP-Verbindung mit Firefox
dummy:1025	wonderland:80	HTTP-Verbindung mit kfm
dummy:1026	wonderland:21	FTP-Verbindung mit ncftp

Jedes Paket kann eindeutig zugestellt werden. Würde man eine der vier Angaben weglassen, so wäre dies nicht mehr möglich.

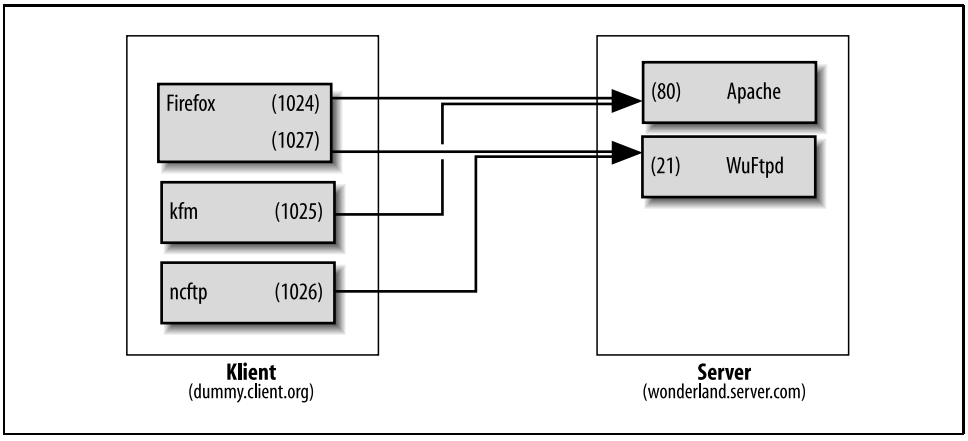

Abbildung 3-3: Eindeutige Adressierung durch Ports

Im Internet sind bestimmte Portnummern fest vergeben. So benutzt DNS die Portnummer 53. Dabei gilt auf Unix-Systemen, daß Klienten in der Regel Portnummern grö-

ßer 1023 benutzen, während die meisten Server auf Portnummern kleiner als 1024 auf Anfragen warten. Leider ist es so, daß sich nicht alle Server an diese Regel halten. So benutzt z. B. das X Window-System standardmäßig die Portnummer 6000. Dies werden wir später bei der Aufstellung der Filterregeln berücksichtigen müssen.

TCP soll darüber hinaus aber auch eine gewisse Verläßlichkeit der Übertragung sicherstellen. Dies ist nötig, da IP nur die Übertragung einzelner Pakete regelt, aber nicht deren Lieferung garantiert. So kann es passieren, daß Datenpakete verlorengehen, doppelt gesendet werden oder in der falschen Reihenfolge beim Empfänger ankommen.

Genau diese Probleme soll TCP beheben. Zu diesem Zweck werden alle gesendeten Bytes numeriert. Im Header jeden Paketes wird dabei die Nummer des ersten enthaltenen Datenbytes eingetragen (*Folgenummer*). Trifft nun ein Paket beim Empfänger ein, so überprüft dieser die Folgenummer des Paketes. Ist sie niedriger als der erwartete Wert, so wurde das Paket schon empfangen, und der Empfänger kann es entsorgen. Ist sie zu hoch, so muß ein Paket verlorengegangen sein. In diesem Fall wird der Empfänger den Absender darum bitten, noch einmal alle Pakete zu senden, deren Folgenummer größer ist als die des letzten Paketes mit einer erwarteten Folgenummer. Trifft schließlich ein Paket mit einer korrekten Folgenummer ein, so sendet der Empfänger als Bestätigung die Nummer des nächsten erwarteten Paketes (Nummer des letzten empfangenen Datenbytes + 1). Empfängt der Sender nach Ablauf einer gewissen Zeit keine Bestätigung, so wird er die unbestätigten Pakete noch einmal senden, da er annimmt, daß sie den Empfänger nicht erreicht haben. Dabei ist es durchaus möglich, daß mehrere Pakete gesendet werden, bevor eine Bestätigung erwartet wird.

Die Folgenummern werden nicht für jede Verbindung wieder auf Null gesetzt. Statt dessen wird bei jedem Verbindungsaufbau auf beiden Seiten ein zufälliger Wert gewählt, der dann als Folgenummer für die eigenen Pakete dient. Dies hat den Vorteil, daß zufällig eintreffende Pakete aus früheren Verbindungen daran erkannt werden können, daß ihre Folgenummern außerhalb eines zulässigen Bereiches liegen.

Um der Gegenstelle mitzuteilen, welche Folgenummern man zu benutzen gedenkt, findet ein Verbindungsaufbau statt, der aus drei Paketen besteht. Im Header dieser Pakete sind dabei drei Felder für diese Darstellung interessant:

SEQ Folgenummer des Paketes (Sequence Number).

ACK Folgenummer, die von der Gegenstelle erwartet wird (Acknowledgement Number).

CTL Feld mit mehreren Bits, *Flags* genannt, die Kontrollinformationen signalisieren (Control Flags):

- **SYN** Dieses Paket gibt die neue Folgenummer des Senders bekannt. Wird nur beim Verbindungsaufbau benutzt (synchronize).
- **ACK** Dieses Paket bestätigt ein vorangegangenes Paket (acknowledge).
- **FIN** Der Sender wird keine weiteren Daten mehr schicken. Haben beide Seiten ein FIN gesendet, ist die Verbindung beendet (finish).
- **RST** Die Verbindung wird sofort beendet (reset).

PSH Normalerweise sammelt der Kernel Daten erst einmal und gibt sie dann in größeren Blöcken an die Anwendung weiter. Dieses Flag soll das verhindern und bewirken, daß die Daten sofort weitergegeben werden (push). Außerdem garantiert das Setzen dieses Bits, daß das Paket bestätigt wird, unabhängig davon, ob dies normalerweise geschehen würde oder nicht. Das Flag wird unter Linux kaum verwendet.

URG Dient dazu, einen Teil der Daten im Paket als besonders eilig zu markieren (urgent). Ein zusätzliches Feld im TCP-Header namens »Urgent Pointer« gibt den Offset der betreffenden Daten innerhalb des Paketes an.

Der Verbindungsaufbau erfolgt nun in der Form:

	Rechner A				Rechner B
1.	⟶	SEQ=100		CTL=SYN	⟶
2.	⟵	SEQ=300	ACK=101	CTL=SYN,ACK	⟵
3.	⟶	SEQ=101	ACK=301	CTL=ACK	⟶

In diesem Beispiel beginnt Rechner A, indem er im ersten Schritt Rechner B mitteilt, daß er für seine Pakete mit der Folgenummer 100 beginnt. Rechner B bestätigt dies, indem er mitteilt, er erwarte nun ein Paket mit der Folgenummer 101. Zusätzlich gibt er an, er werde mit der Folgenummer 300 beginnen. Im dritten Schritt bestätigt Rechner A dies, indem er als Nummer des nächsten von ihm erwarteten Paketes die 301 angibt.

Für die Konfiguration unserer Firewall sollten wir uns merken, daß ein ankommendes Paket, in dem das SYN-Bit gesetzt ist, das ACK-Bit aber nicht, darauf hinweist, daß eine Verbindung zu unserem Rechner aufgebaut werden soll.

UDP

UDP, das User Datagram Protocol, wird in RFC 768 beschrieben. Bei ihm handelt es sich um den »kleinen Bruder« von TCP. Auch hier werden Anwendungen über Ports adressiert. Die weitergehende Verbindungslogik von TCP mit seinen Folgenummern und Bestätigungen fehlt dagegen. Bei UDP werden einzelne Pakete, sogenannte *Datagramme*, abgeschickt in der Hoffnung, der Empfänger möge sie irgendwann irgendwie erhalten. Diese Art von Protokollen wird in der Fachwelt oft auch als »Send and Pray« bezeichnet.

Nun sollte man aber nicht den Fehler begehen, UDP für antiquiert und obsolet zu erachten. Tatsächlich gibt es eine Reihe von Anwendungen, bei denen UDP deutliche Vorteile gegenüber dem verbindungsorientierten TCP aufweist. Dies ist insbesondere im Multimedia-Bereich der Fall. Wenn Töne oder Filme in Echtzeit über das Netz übertragen werden sollen, stellt sich der Verwaltungsaufwand von TCP als spürbarer Nachteil heraus. Gehen z.B. bei einer Musikübertragung über das Internet einige wenige Pakete verloren, so können heutige Fehlerkorrekturverfahren die fehlenden Informationen genau genug aus den vorangegangenen Daten »schätzen«, daß es dem Hörer nicht auffällt. Ein Warten auf das erneute Senden von Daten würde zu Pausen führen und den Hörgenuß empfindlich beeinträchtigen.

Andere Anwendungen von UDP betreffen Fälle, in denen eine Anfrage und die dazugehörige Antwort in ein Paket passen, womit die drei Pakete für den Verbindungsaufbau zu einem spürbaren Overhead werden. Dies war z. B. der Grund, warum DNS für kleinere Anfragen, die den Großteil der Fälle im normalen Betrieb ausmachen, UDP statt TCP benutzt.

Schließlich existieren noch Fälle, in denen zwar Informationen verschickt werden, jedoch keine Antwort erwartet wird. Hier wird z. T. nicht gewünscht, daß zusätzliche Systemlast erzeugt wird, indem auf eine Empfangsbestätigung gewartet wird, die die sendende Anwendung nicht interessiert. Ein Beispiel hierfür ist das Schreiben von Systemprotokollen über das Netzwerk. Hier werden Fehler- und Statusmeldungen, anstatt sie in die lokale Protokolldatei zu schreiben, über das Netz an einen dedizierten Protokollierungsrechner geschickt. Das dabei verwendete Syslog-Protokoll sendet UDP-Pakete an Port 514 des Protokollierungsrechners, ohne daß jemals eine Antwort erfolgt.

DNS

Bei DNS handelt es sich um eine Art dezentrales Telefonbuch des Internets, das in den RFCs 1034 und 1035 beschrieben wird. Ein DNS-Server ist dafür zuständig, logische Adressen der Art *www.example.com* in IP-Adressen der Art *192.0.34.166* umzusetzen. Dies ist notwendig, da IP keine logischen Adressen kennt, der menschliche Benutzer dagegen ein schlechtes Gedächtnis für Zahlen hat.

Das ganze Internet ist zu diesem Zweck in *Zonen* eingeteilt. Jede dieser Zonen besitzt mindestens zwei Server, die für die Rechner ihrer Zone die Umwandlung von einer logischen in eine IP-Adresse durchführen können. Man kann diese Zonen auch wieder in Unterzonen unterteilen, die eigene DNS-Server besitzen und gegebenenfalls von jemand anders administriert werden. In engem Zusammenhang zu den Zonen stehen die *Domänen*. Unter einer Domäne versteht man üblicherweise den Teil der logischen Adresse, der nicht den Rechnernamen darstellt. Im Fall von *www.oreilly.de* ist *www* der eigentliche Rechnername, und *oreilly.de* ist die Domäne. Innerhalb von *oreilly.de* gibt es noch weitere Rechner. Zum Beispiel wird dort auch ein Mailserver betrieben, der unter dem Namen *mail.oreilly.de* angesprochen werden kann[2].

Der Zusammenhang zwischen Rechnernamen, Domänen und Zonen ist im Internet so gelöst, daß die sogenannten *Top Level Domains (TLDs)* wie *.com*, *.org*, *.net*, *.edu*, *.de* ... in einer Zone zusammengefaßt sind, die durch die sogenannten *Root-Server* verwaltet werden. Diese Root-Server sind weltweit verteilt und bilden den Ausgangspunkt einer Adressumsetzung.

Diese Root-Zone enthält aber nicht etwa alle Zuordnungen zu allen Rechneradressen im Internet. Vielmehr stellen die Top Level Domains eigene Unterzonen dar, die eigene DNS-Server besitzen. In der Root-Zone sind nur die Adressen dieser TLD-Server gespeichert. Um nun eine Adresse wie *www.oreilly.de* aufzulösen, muß man sich als nächstes an die Server der Unterzone *.de* wenden. Auch hier ist nur eingetragen, daß unsere gesuch-

2 Technisch handelt es sich bei dem Namen *mail* um einen Alias, der auf den tatsächlichen Namen des Mailservers verweist.

te Domäne *oreilly.de* wiederum eine Unterzone mit eigenen DNS-Servern ist. Erst in den Servern der Zone *oreilly.de* steht, daß der gesuchte Rechner die IP-Adresse *193.99.144.71* hat.

Stellt man nun an einen Server eine DNS-Anfrage nach einem Rechner, der nicht in der von ihm verwalteten Zone liegt, so gibt es zwei mögliche Reaktionen. Entweder wird der befragte Rechner auf einen anderen Rechner hinweisen, der die Antwort kennen könnte, oder er fragt selbst bei einem anderen Rechner nach und teilt dem Anfragenden dann das Ergebnis mit. Die explizite Aufforderung an einen DNS-Server, gegebenenfalls auch bei anderen Rechnern nachzufragen, nennt man eine *rekursive Anfrage*.

Üblicherweise erlaubt der DNS-Server Ihres Providers rekursive Anfragen von den Rechnern seiner Kunden. Ihr Rechner braucht sich also nicht selber von den Rootservern zum DNS-Server des Zielrechners vorzuhangeln, sondern überläßt es ihm.

Da jede Zone aus Sicherheitsgründen mehr als einen DNS-Server besitzt, existiert mit dem *Zone Transfer Request* eine Methode, mit der der ganze Inhalt der DNS-Datenbank in einem Vorgang heruntergeladen werden kann. So kann man den Abgleich der einzelnen Server automatisieren, und neue Informationen brauchen nur auf einem Server manuell eingetragen zu werden.

DNS-Server können normalerweise sowohl auf Port 53 UDP als auch auf Port 53 TCP angesprochen werden. Für normale Anfragen wird dabei in der Regel UDP verwendet. Für Anfragen, bei denen die Antwort einen bestimmten Umfang überschreitet (z. B. Zone Transfer Requests), wird dagegen TCP verwendet.

Schließlich soll noch erwähnt werden, daß Domänen existieren, die im Internet nicht vergeben werden. Diese kann man im lokalen Netz zu Testzwecken verwenden, ohne dabei befürchten zu müssen, daß es zu Konflikten mit tatsächlichen Adressen im Internet kommt.

RFC 2606 definiert zu diesem Zweck die folgenden TLDs und Domänen:

.test ist gedacht für das Testen von aktuellem oder neuem DNS-bezogenem Programmcode.

.example wird für die Verwendung in Programmdokumentationen und als Beispiele empfohlen.

.invalid dient dazu, ungültige Domänennamen zu konstruieren, die auch auf Anhieb als ungültig erkennbar sind.

.localhost wird von einigen DNS-Implementationen fest mit der Adresse 127.0.0.1 versehen. Sie sollte daher nicht zu anderen Zwecken verwendet werden, um keine Probleme mit bestehenden DNS-Installationen zu verursachen.

example.com, example.net, example.org Diese drei Domänen wurden ebenfalls reserviert, um als Beipiele in Dokumentationen und zu Testzwecken zu dienen. Allerdings existiert im Internet tatsächlich ein Rechner *www.example.com*. Auf diesem liegt aber nur eine Webseite, die auf RFC 2606 verweist. Sie verlieren also nichts, wenn Sie in Ihrem lokalen Netz eine Domäne *example.com* definieren und damit den Zugriff aus dem lokalen Netz auf den Server blockieren.