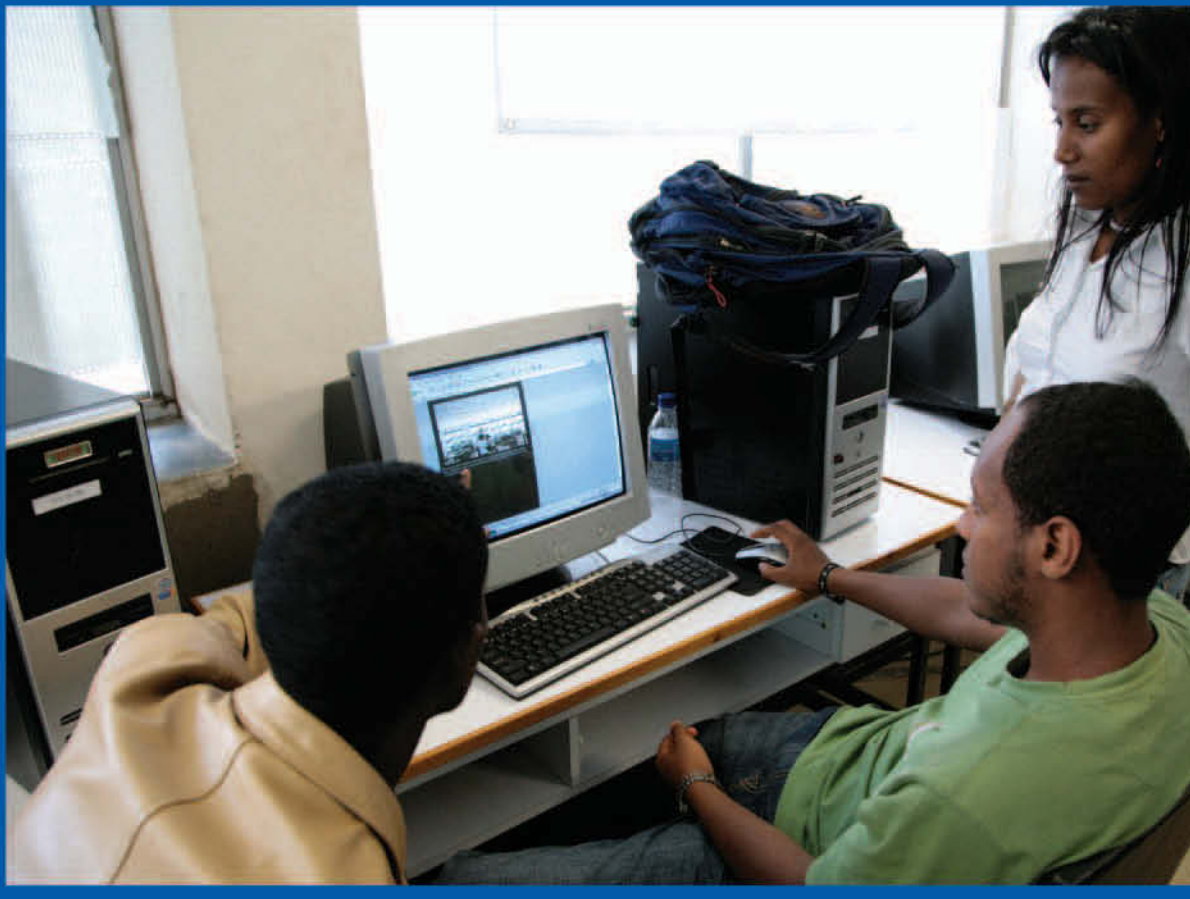
Netzwerkgrundlagen

CCNA Exploration Companion Guide



Mark A. Dye • Rick McDonald • Antoon W. Rufi

ADDISON-WESLEY

Cisco | Networking Academy
Mind Wide Open

Kapitel 3

Funktionalität und Protokolle der Anwendungsschicht

Jeder von uns erlebt das Internet, indem er das WWW, E-Mail und File-sharing-Programme nutzt. Programme wie diese stellen die Schnittstelle zum darunter liegenden Netzwerk dar und gestatten Ihnen das relativ einfache Senden und Empfangen von Daten. Die meisten Anwendungen sind intuitiv: Der Zugang und die Nutzung sind problemlos, ohne dass Sie wissen müssen, wie sie funktionieren. Je mehr Sie jedoch über die Welt der Netzwerktechnik lernen, desto wichtiger wird es zu erfahren, wie eine Anwendung Nachrichten, die über das Netzwerk gesendet und empfangen werden, formatiert, überträgt und interpretiert.

Die Veranschaulichung der Mechanismen, welche die Kommunikation über das Netzwerk ermöglichen, ist einfacher, wenn Sie das schichtorientierte Gerüst des OSI-Modells verwenden. Abbildung 3.1 zeigt dieses Gerüst. Das OSI-Modell ist ein sieben Schichten umfassendes Modell, das Ihnen den Datenfluss zwischen den einzelnen Schichten erläutern soll.

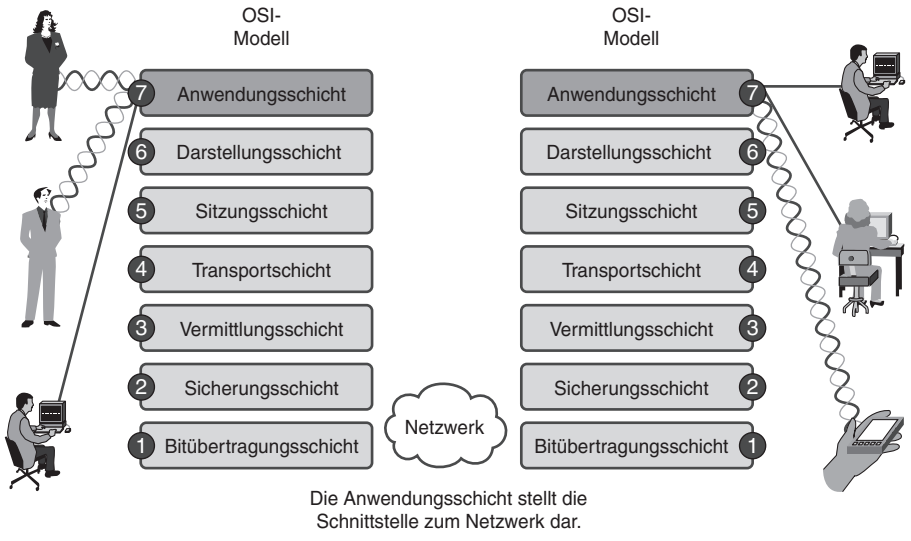


Abbildung 3.1: Menschliche Netzwerke und Datennetze miteinander verbinden

Dieses Kapitel legt den Schwerpunkt auf Schicht 7 – die Anwendungsschicht – und ihre Komponenten: Anwendungen, Dienste und Protokolle. Sie werden erfahren, wie diese drei Elemente die robuste Kommunikation über das Datennetz möglich machen.

3.1 Anwendungen: Die Schnittstelle zwischen den Netzwerken

In diesem Abschnitt werden zwei wichtige Konzepte eingeführt:

- **Anwendungsschicht.** Die Anwendungsschicht des OSI-Modells ist der erste Schritt der Daten auf dem Weg ins Netzwerk.
- **Anwendungssoftware.** Anwendungen sind Softwareprogramme, mit deren Hilfe Menschen über das Netzwerk kommunizieren. Beispiele für Anwendungssoftware (HTTP, FTP, E-Mail usw.) werden in diesem Kapitel verwendet, um die Unterschiede zwischen diesen beiden Konzepten zu erläutern.

3.1.1 Das OSI- und das TCP/IP-Modell

Das OSI-Referenzmodell ist eine schichtenbasierte, abstrakte Darstellung, deren Zweck darin besteht, als Vorlage für das Entwerfen und Lehren von Netzwerkprotokollen zu dienen. Das OSI-Modell unterteilt den Netzwerkprozess in sieben logische Schichten, die jeweils eine eigene Funktionalität aufweisen und denen bestimmte Dienste und Protokolle zugeordnet sind.

Im OSI-Modell werden die Daten von einer Schicht zur nächsten weitergegeben. Der Vorgang beginnt in der Anwendungsschicht des sendenden Hosts und arbeitet sich dann in der Hierarchie nach unten bis zur Bitübertragungsschicht vor. Danach wird der Kommunikationskanal an den Zielhost übergeben, wo die Daten die Hierarchie wieder nach oben durchlaufen und in der Anwendungsschicht enden. Abbildung 3.2 zeigt die Schritte dieses Prozesses.

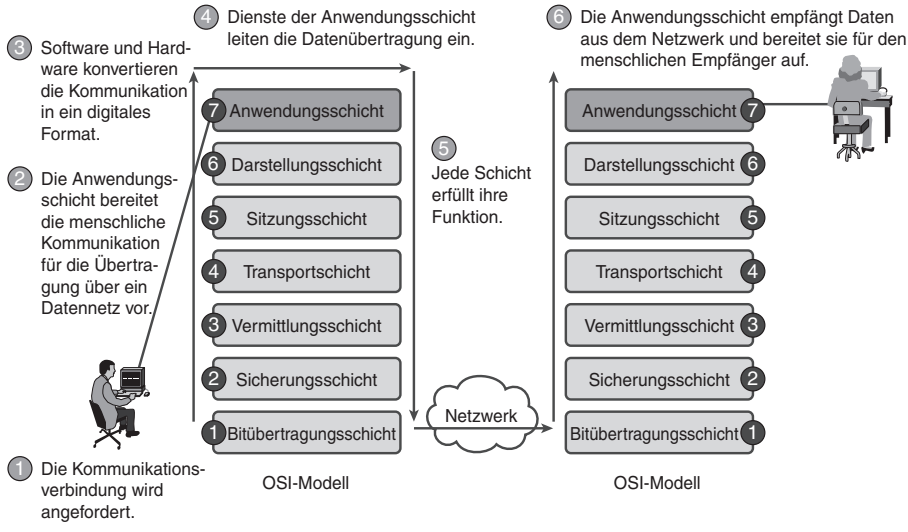


Abbildung 3.2: OSI-Kapselungsprozess

Nachfolgend werden die folgenden sechs Schritte erläutert:

1. Zunächst wird vom menschlichen Benutzer eine Kommunikationsnachricht erstellt.
2. Die Anwendungsschicht bereitet diese Kommunikationsnachricht für die Übertragung über das Datennetz vor.
3. Software und Hardware wandeln die Nachricht in ein digitales Format um.
4. Die Dienste der Anwendungsschicht leiten den Datentransfer ein.
5. Jede Schicht erfüllt nun die vorgesehene Aufgabe. Die OSI-Schichten kapseln die Daten auf ihrem Weg durch den Protokollstapel. Gekapselte Daten durchwandern dann das Medium bis zum Ziel. Dort entkapseln OSI-Schichten die Daten auf dem Weg durch den Stapel nach oben.
6. Die Anwendungsschicht empfängt die Daten aus dem Netzwerk und bereitet sie für die Verwendung durch den Menschen auf.

Die Anwendungsschicht (Schicht 7) ist sowohl im OSI- als auch im TCP/IP-Modell vorhanden. (Weitere Informationen zum TCP/IP-Modell finden Sie im Abschnitt »Protokoll- und Referenzmodelle« in Kapitel 2, »Kommunikation über das Netzwerk«.) Schicht 7 stellt die Schnittstelle zwischen den Anwendungen, mit denen Sie kommunizieren, und dem zugrunde liegenden Netzwerk dar, über das Ihre Nachrichten übertragen werden. Protokolle der Anwendungsschicht werden für den Austausch von Daten zwischen Programmen verwendet, die auf den Absender- und Zielhosts laufen. Es gibt zahlreiche Anwendungsschichtprotokolle, und fortlaufend werden weitere entwickelt. (Beispiele finden Sie weiter unten unter der Überschrift »Benutzeranwendungen, Dienste und Anwendungsschichtprotokolle«.)

Obwohl die TCP/IP-Protokollfamilie vor der Definition des OSI-Modells entwickelt wurde, entspricht die Funktionalität der Protokolle der TCP/IP-Anwendungsschicht weitgehend dem Gerüst der oberen drei Schichten des OSI-Modells: der Anwendungs-, der Darstellungs- und der Sitzungsschicht.

Die meisten Anwendungen – z. B. Webbrowser oder E-Mail-Clients – umfassen Funktionalitäten der OSI-Schichten 5, 6 und 7. Abbildung 3.3 zeigt einen Vergleich des OSI- und des TCP/IP-Modells.

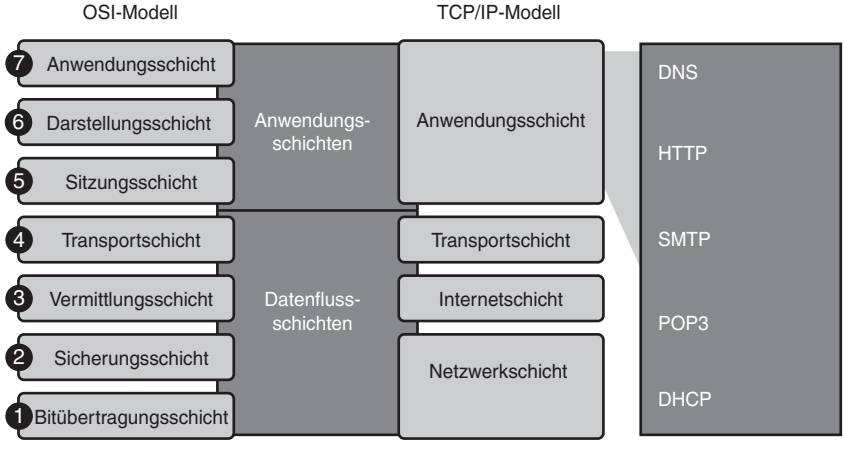


Abbildung 3.3: Das OSI- und das TCP/IP-Modell

Die meisten Protokolle der TCP/IP-Anwendungsschicht wurden vor dem Aufkommen von PCs, grafischen Oberflächen und Multimedia-Objekten entwickelt. Infolgedessen implementieren diese Protokolle nur einen geringen Teil der Funktionalität, die in der Darstellungs- und der Sitzungsschicht des OSI-Modells beschrieben wird. Diese beiden Schichten werden wir in den nächsten Abschnitten genauer betrachten.

Darstellungsschicht

Die Darstellungsschicht bietet im Wesentlichen drei Funktionen:

- Codierung und Konvertierung der Anwendungsschichtdaten, um sicherzustellen, dass die Daten des Absendergeräts von der jeweiligen Anwendung auf dem Zielgerät interpretiert werden können
- Komprimierung der Daten auf eine Weise, dass sie auf dem Zielgerät entkomprimiert werden können
- Verschlüsselung der Daten für die Übertragung und Entschlüsselung der Daten beim Empfang auf dem Zielgerät

Implementierungen der Darstellungsschicht sind nicht unbedingt mit einem bestimmten Protokollstapel verknüpft. Exemplarisch seien hier Video und Grafikanwendungen genannt:

- Zu den bekanntesten Standards für Videoanwendungen gehören QuickTime und MPEG (Motion Picture Experts Group). QuickTime ist eine von Apple Computer stammende Spezifikation für Video- und Audiodaten, während MPEG ein verbreiteter Standard zur Videokomprimierung und -codierung ist.
- Bekannte Grafikformate sind GIF (Graphics Interchange Format), JPEG (Joint Photographic Experts Group) und TIFF (Tagged Image File Format). GIF und JPEG sind Komprimierungs- und Codierungsstandards, und TIFF ist ein Standardcodierungsformat für Grafiken.

Sitzungsschicht

Die Funktionen der Sitzungsschicht erstellen Dialoge zwischen Absender- und Zielanwendungen und erhalten diese aufrecht. Die Sitzungsschicht behandelt den Austausch von Daten zur Einleitung und Aufrechterhaltung von Dialogen und den Neustart von Sitzungen, die unterbrochen oder längere Zeit nicht verwendet wurden.

Protokolle der TCP/IP-Anwendungsschicht

Die bekanntesten Protokolle der TCP/IP-Anwendungsschicht sind diejenigen, die den Austausch der Benutzerdaten ermöglichen. Diese Protokolle geben das Format und die Steuerinformationen an, die für viele gängige Internetkommunikationsfunktionen erforderlich sind. Zu den TCP/IP-Protokollen gehören die folgenden:

- **DNS (Domain Name System).** Wird verwendet, um Internetnamen in IP-Adressen aufzulösen.
- **HTTP (Hypertext Transfer Protocol).** Dient der Übertragung von Dateien, aus denen die Webseiten des WWW bestehen.

- **SMTP (Simple Mail Transfer Protocol).** Wird zur Übertragung von Mail-Nachrichten und -anhängen verwendet.
- **Telnet.** Ein Protokoll zur Terminal-Emulation, das den Fernzugriff auf Server und Netzwerkgeräte gestattet.
- **FTP (File Transfer Protocol).** Wird zur interaktiven Übertragung von Dateien zwischen Systemen verwendet.

Die Protokolle in der TCP/IP-Familie sind im Allgemeinen durch RFCs (Requests for Comments) definiert. Die IETF (Internet Engineering Task Force) ist für die Pflege der RFCs als Normen für die TCP/IP-Familie zuständig.

3.1.2 Software der Anwendungsschicht

Die mit den Protokollen der Anwendungsschicht im OSI- und TCP/IP-Modell verknüpften Funktionen ermöglichen eine Anbindung des menschlichen Netzwerks an das zugrunde liegende Datennetz. Wenn Sie einen Webbrowser oder ein Instant Messaging-Fenster öffnen, wird eine Anwendung gestartet, und das Programm wird in den Speicher des Geräts kopiert und dort ausgeführt. Jedes ausgeführte Programm, das auf einem Gerät geladen ist, wird als Prozess bezeichnet.

Innerhalb der Anwendungsschicht gibt es zwei Arten von Softwareprogrammen oder Prozessen, die Zugriff auf das Netzwerk bieten: Anwendungen und Dienste. Abbildung 3.4 zeigt dieses Konzept.

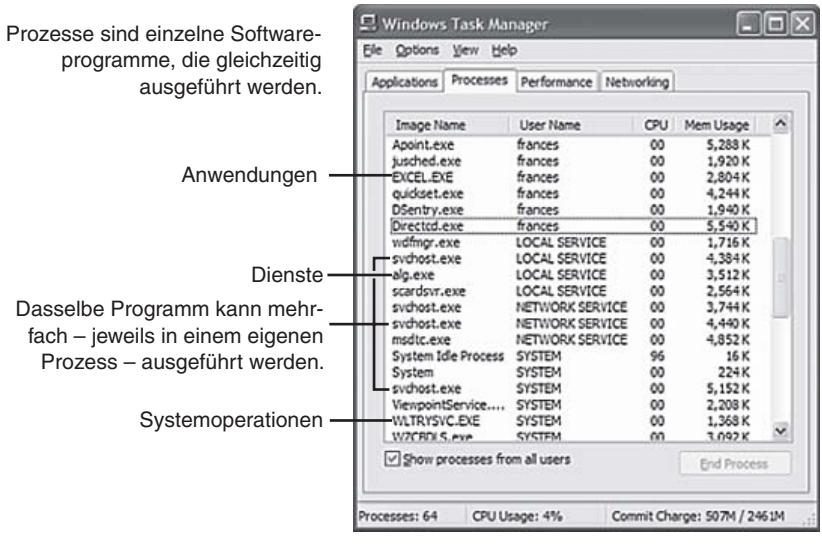


Abbildung 3.4: Softwareprozesse

Anwendung mit Netzwerkunterstützung

Einige Anwendungen für Endbenutzer bieten direkte Netzwerkunterstützung, das heißt, sie implementieren die Protokolle der Anwendungsschicht und können direkt mit den unteren Schichten des Protokollstapels kommunizieren. Mailclients und Webbrowser sind Beispiele für diese Anwendungstypen.

Dienste der Anwendungsschicht

Andere Programme – z. B. Dateiübertragungssoftware oder Spooler für den Netzwerkdruck – benötigen für die Verwendung von Netzwerkressourcen die Hilfe von Diensten der Anwendungsschicht. Wiewohl für den Benutzer transparent, sind diese Dienste an das Netzwerk angebunden und bereiten die Daten für die Übertragung vor. Verschiedene Datenarten – Text, Grafiken oder Video – benötigen unterschiedliche Netzwerkdienste, um sicherzustellen, dass sie angemessen auf die Verarbeitung durch die Funktionen vorbereitet sind, die in den unteren Schichten des OSI-Modells vorhanden sind.

Alle Anwendungen und Netzwerkdienste verwenden Protokolle, die die zu verwendenden Standards und Datenformate definieren. Ein Dienst stellt eine bestimmte Funktion bereit, und ein Protokoll gibt die Regeln an, die der Dienst verwendet. Um die *Funktion* verschiedener Netzwerkdienste zu verstehen, müssen Sie sich mit den ihnen zugrunde liegenden Protokollen vertraut machen, denn diese bestimmen die Abläufe.

3.1.3 Benutzeranwendungen, Dienste und Anwendungsschichtprotokolle

Die Anwendungsschicht verwendet Protokolle, die innerhalb von Anwendungen und Diensten implementiert sind. Anwendungen bieten Benutzern eine Möglichkeit, Nachrichten zu erstellen, Dienste der Anwendungsschicht richten eine Schnittstelle zum Netzwerk ein, und Protokolle definieren die Regeln und Formate, die festlegen, wie die Daten behandelt werden (siehe Abbildung 3.5). Ein einzelnes ausführbares Programm kann alle drei Komponenten verwenden. Wenn Sie beispielsweise »Telnet« sagen, könnten Sie die Anwendung Telnet, den Telnet-Dienst oder das Telnet-Protokoll meinen.

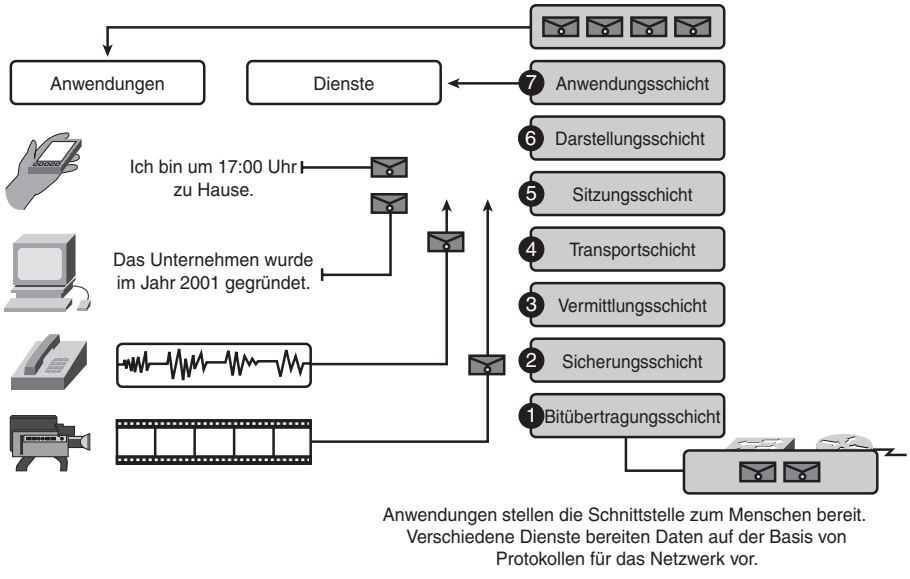


Abbildung 3.5: Menschliche Netzwerke und Datennetze verbinden

Im OSI-Modell befinden sich Anwendungen, die direkt mit menschlichen Benutzern interagieren, ganz oben im Protokollstapel. Gleiches gilt auch für die Benutzer selbst. Wie alle Schichten im OSI-Modell ist die Anwendungsschicht auf Funktionen der untergeordneten Schichten angewiesen, um den Kommunikationsprozess auszuführen. Innerhalb der Anwendungsschicht legen Protokolle fest, welche Nachrichten zwischen Absender- und Zielhosts ausgetauscht werden, wie die Syntax der Steuerbefehle sowie Typ und Format der zu übertragenden Daten aussehen und welche Methoden zur Meldung von und zur Wiederherstellung nach Fehlern angemessen sind.

3.1.4 Funktionen der Anwendungsschichtprotokolle

Absender- und Empfängergeräte verwenden die Protokolle der Anwendungsschicht während einer Sitzung. Damit die Kommunikation erfolgreich verläuft, müssen die Protokolle in der Anwendungsschicht beim Absender und Empfänger übereinstimmen.

Protokolle führen die folgenden Aufgaben aus:

- Sie definieren konsistente Regel für den Datenaustausch zwischen Anwendungen und Diensten, die auf den beteiligten Geräten geladen sind.
- Sie legen fest, wie Daten innerhalb der Nachrichten strukturiert sind und welche Nachrichtentypen zwischen Absender und Empfänger ausge-

tauscht werden. Diese Nachrichten können Dienstanforderungen, Bestätigungen, Daten- oder Statusnachrichten oder Fehlermeldungen sein.

- Sie legen Nachrichtendialoge fest und gewährleisten so, dass für eine gesendete Nachricht die passende Antwort erstellt wird und bei der Datenübertragung die passenden Dienste aufgerufen werden.

Viele Arten von Anwendungen kommunizieren über Datennetze. Aus diesem Grund müssen die Dienste der Anwendungsschicht viele Protokolle implementieren, um alle erforderlichen Kommunikationsausprägungen vermitteln zu können. Jedes Protokoll hat einen bestimmten Zweck und umfasst alle Eigenschaften, die erforderlich sind, um diesen Zweck zu erfüllen. Die Protokollmerkmale in den einzelnen Schichten müssen beachtet werden, damit die Funktionen an der Schnittstelle einer Schicht zu den Diensten der ihr untergeordneten Schicht passen.

Anwendungen und Dienste können im Verlauf eines einzelnen Kommunikationsprozesses auch mehrere Protokolle einsetzen. So kann etwa ein Protokoll festlegen, wie die Netzwerkverbindung herzustellen ist, während ein anderes den Vorgang zur Datenübertragung beschreibt, wenn die Nachricht an die nächstniedrigere Schicht übergeben wird.

3.2 Vorkehrungen für Anwendungen und Dienste treffen

Wenn ein Benutzer versucht, auf Daten mit seinem Gerät zuzugreifen – sei es ein PC, ein Laptop, ein PDA, ein Handy oder irgendein anderes an ein Netzwerk angeschlossenes Gerät –, dann sind diese Daten unter Umständen nicht physisch auf dem betreffenden Gerät gespeichert. Ist dies der Fall, so wird an das Gerät, auf dem die Daten vorhanden sind, eine Anforderung gesendet, um darauf zugreifen zu dürfen. In den folgenden Abschnitten behandeln wir drei Themen, die das Verständnis dafür erleichtert sollen, wie eine Datenanforderung auftritt und erfüllt wird:

- Client/Server-Modell
- Dienste und Protokolle der Anwendungsschicht
- Peer-to-Peer-Netzwerke und -Anwendungen

3.2.1 Das Client/Server-Modell

Im Client/Server-Modell wird das Gerät, welches die Daten anfordert, als Client und das die Anforderung beantwortende Gerät als Server bezeichnet. Client- und Serverprozesse finden in der Anwendungsschicht statt. Der

Client leitet den Kommunikationsprozess ein, indem er Daten beim Server anfordert. Dieser beantwortet die Anforderung durch Senden eines oder mehrerer Datenströme an den Client. Protokolle der Anwendungsschicht beschreiben den Aufbau von Anforderungen und Antworten, die zwischen Clients und Servern ausgetauscht werden. Zusätzlich zu den eigentlichen Daten kann dieser Datenaustausch auch Steuerdaten umfassen, zum Beispiel zur Benutzerauthentifizierung oder zur Bezeichnung einer zu übertragenden Datei.

Exemplarisch für ein Client/Server-Netzwerk sei eine Unternehmensumgebung genannt, in der Mitarbeiter einen Mailserver zum Senden, Empfangen und Speichern von E-Mail verwenden. Der Mailclient auf dem Computer eines Mitarbeiters sendet eine Anforderung an den Mailserver ab, in der er um die Übermittlung ungelesener Mails ersucht. Der Server antwortet dann, indem er die angeforderte E-Mail an den Client schickt.

Zwar wird ein Datenfluss meist so beschrieben, dass Daten vom Server an den Client fließen, doch gibt es immer auch einen Datenfluss vom Client an den Server. Das Volumen der ausgetauschten Daten kann dabei in beiden Richtungen gleichgroß sein, und manchmal schickt der Client sogar mehr Daten an den Server als umgekehrt. Dies ist etwa dann der Fall, wenn der Client eine Datei zu Speicherzwecken an den Server überträgt. Der Datentransfer vom Client zum Server wird gemeinhin als *Upload* (Hochladen) bezeichnet, die Übertragung vom Server zum Client hingegen als *Download* (Herunterladen). Abbildung 3.6 zeigt das Prinzip des Client/Server-Modells.

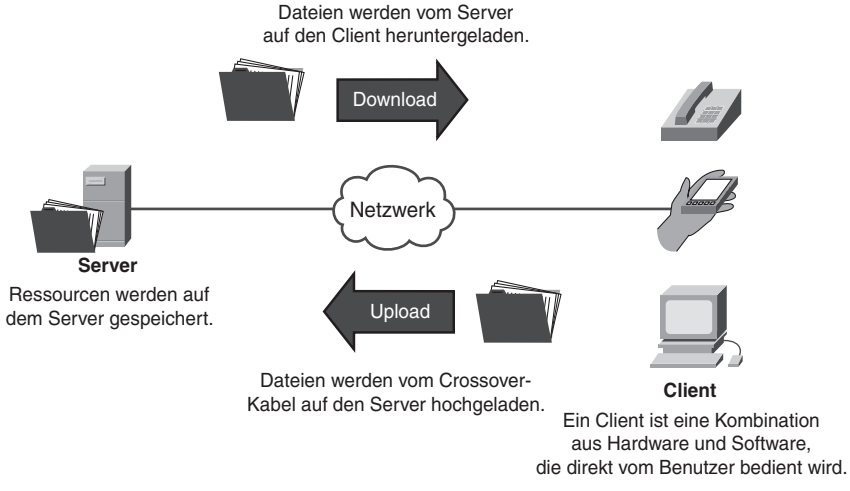


Abbildung 3.6: Das Client/Server-Modell

3.2.2 Server

Im allgemeinen Netzwerkkontext ist jedes Gerät, das auf Anforderungen von Clientanwendungen reagiert, ein Server. Ein Server ist gemeinhin ein Computer, der über Daten oder Dienste verfügt, die für viele Client-Systeme freigegeben sind. So können etwa Webseiten, Dokumente, Datenbanken, Bilder, Video- und Audiodateien auf einem Server gespeichert sein und den anfragenden Clients zur Verfügung gestellt werden. In anderen Fällen – etwa bei einem Netzwerkdrucker – leitet der Druckserver die Druckanforderungen von den Clients an den angegebenen Drucker weiter.

Verschiedene Arten von Server-Anwendungen stellen unterschiedliche Bedingungen für den Client-Zugriff. Einige Server können die Authentifizierung von Anmeldeinformationen verlangen, um zu überprüfen, ob der Benutzer die Berechtigung hat, auf die angeforderten Daten zuzugreifen oder einen bestimmten Vorgang durchzuführen. Solche Server benötigen eine zentrale Liste der Benutzerkonten und der den Benutzern gewährten Autorisierungen oder Berechtigungen (sowohl für den Datenzugriff als auch zur Durchführung von Vorgängen). Wenn Sie beispielsweise einen FTP-Client verwenden, um Daten auf einen FTP-Server hochzuladen, verfügen Sie möglicherweise zwar über Berechtigungen zum Schreiben in Ihren eigenen Ordner, nicht aber zum Lesen anderer Dateien auf dem Server.

In einem Client/Server-Netzwerk führt der Server einen Dienst oder Prozess aus, der auch als Server-Daemon bezeichnet wird. Wie die meisten Dienste laufen Daemons gewöhnlich im Hintergrund und unterliegen nicht der direkten Kontrolle eines Benutzers. Daemons »horchen« auf Anforderungen eines Clients, denn ihre Aufgabe besteht darin, immer dann zu reagieren, wenn der Server eine Anforderung für einen Dienst erhält, der vom jeweiligen Daemon vermittelt wird. »Hört« ein Daemon eine passende Clientanforderung, dann tauscht er die entsprechenden, vom Protokoll vorgesehenen Nachrichten mit dem Client aus und sendet die angeforderten Daten im passenden Format an den Client.

Abbildung 3.7 zeigt, wie Clients Dienste vom Server anfordern: Ein Client fordert eine Audiodatei (WAV) an, ein anderer eine Videodatei (AVI). Der Server reagiert, indem er die angeforderten Dateien an die Clients schickt.

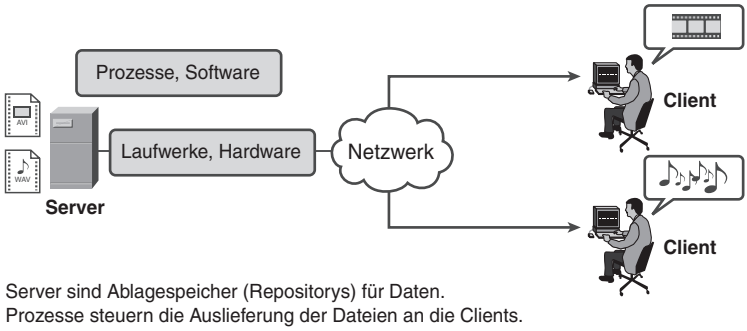


Abbildung 3.7: Server

3.2.3 Dienste und Protokolle der Anwendungsschicht

Eine einzelne Anwendung kann viele verschiedene unterstützende Dienste der Anwendungsschicht nutzen. Insofern ist, was dem Benutzer als einzelne Anforderung einer Webseite erscheint, in Wirklichkeit oft ein Dutzend oder mehr Einzelanforderungen. Für jede Anforderung können mehrere Prozesse ausgeführt werden. So verlangt beispielsweise FTP, dass ein Client einen Steuer- und einen Datenstromprozess zum Server einleitet.

Hinzu kommt, dass Server in der Regel mehrere Clientanforderungen gleichzeitig erhalten (siehe Abbildung 3.8). Beispielsweise kann ein Telnet-Server zur selben Zeit mehrere Verbindungsanforderungen von Clients erhalten. Diese einzelnen Clientanforderungen müssen gleichzeitig und separat verarbeitet werden, damit die Verbindungen hergestellt werden können. Die Prozesse und Dienste der Anwendungsschicht sind auf Unterstützung durch Funktionen untergeordneter Schichten angewiesen, um mehrere Kommunikationsprozesse erfolgreich verwalten zu können.

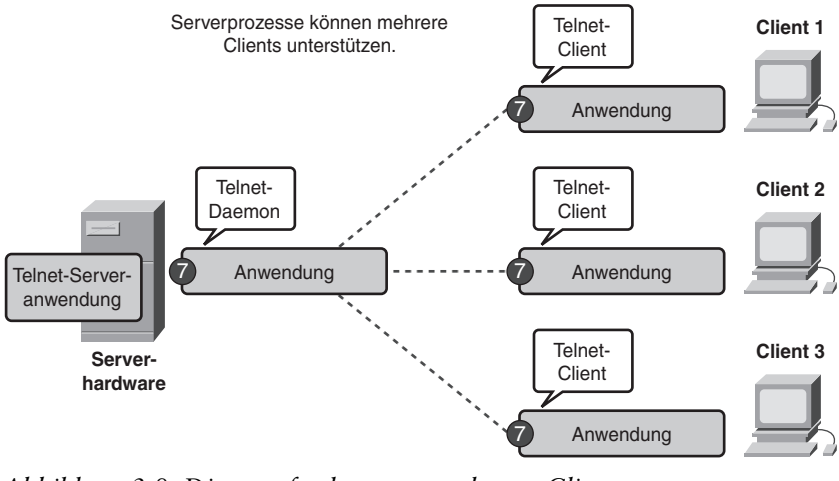


Abbildung 3.8: Dienstanforderungen mehrerer Clients

Client/Server-Interaktion (3.2.3.2)

Packet Tracer
Aktivität

Bei dieser Aktivität untersuchen Sie ein einfaches Beispiel für die Client/Server-Interaktion, welches als Modell für komplexere Interaktionen im weiteren Verlauf des Kurses dienen kann. Zur Durchführung der Aktivität verwenden Sie den Packet Tracer und die Datei *e1-3232.pka* auf der Begleit-CD-ROM zu diesem Buch.

3.2.4 Peer-to-Peer-Netzwerke und -Anwendungen

Neben dem Client/Server-Modell gibt es in der Netzwerktechnik noch ein zweites Modell: das Peer-to-Peer-Modell (kurz auch als P2P bezeichnet). Dieses existiert auf zwei Ebenen: Peer-to-Peer-Netzdesign und Peer-to-Peer-Anwendungen. Beide Formen weisen ähnliche Eigenschaften auf, funktionieren in der Praxis aber unterschiedlich.

Peer-to-Peer-Netzwerke

In einem Peer-to-Peer-Netzwerk sind zwei oder mehr Computer über ein Netzwerk miteinander verbunden und nutzen Ressourcen wie Drucker und Dateien gemeinsam, ohne dass ein dedizierter Server vorhanden wäre. Jedes angeschlossene Endgerät – der sogenannte Peer – kann als Server oder als Client agieren. Ein Computer kann in einer Transaktion die Rolle eines Servers übernommen haben, während er gleichzeitig in einer anderen Transaktion als Client agiert. Die Rollen von Client und Server werden anforderungsbezogen festgelegt (siehe Abbildung 3.9). Die Abbildung zeigt, wie ein Peer bei einem anderen Peer die Bereitstellung von Druckdiensten anfordert und gleichzeitig als Dateiserver fungiert, der eine seiner Dateien freigibt.

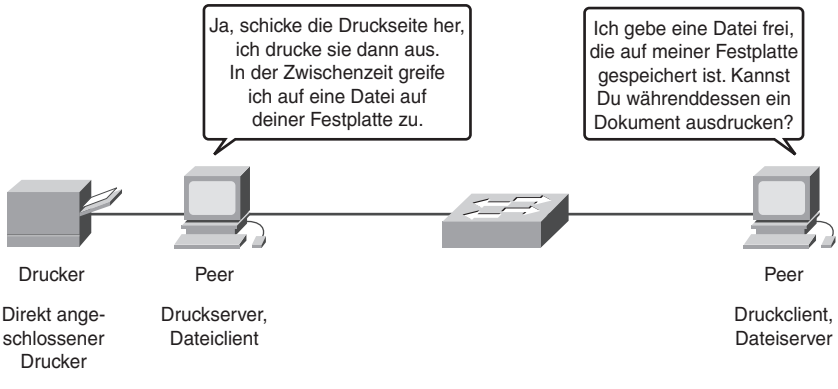


Abbildung 3.9: Peer-to-Peer-Netzwerk

Ein einfaches Heimnetz mit zwei verbundenen Computern, die gemeinsam einen Drucker verwenden, ist ein Beispiel für ein Peer-to-Peer-Netzwerk. Jeder Benutzer kann für seinen Computer festlegen, ob Dateien gemeinsam verwendet oder Netzwerkspiele aktiviert werden oder eine Internetverbindung freigegeben wird. Ein weiteres Beispiel für diese Funktionalität sind zwei Computer, die an ein großes Netzwerk angeschlossen sind und einander mithilfe von Softwareanwendungen Ressourcen über das Netzwerk freigeben.

Anders als das Client/Server-Modell, das dedizierte Server benutzt, werden die Ressourcen bei Peer-to-Peer-Netzwerken dezentralisiert. Die freizugebenden Daten werden nicht auf dedizierten Servern, sondern an einem beliebigen Speicherort auf einem beliebigen angeschlossenen Gerät gespeichert. Viele aktuelle Betriebssysteme unterstützen die Datei- und Druckfreigabe ohne zusätzliche Serversoftware. Da Peer-to-Peer-Netzwerke gewöhnlich keine zentralisierten Benutzerkonten, Berechtigungen oder Überwachungsfunktionen verwenden, ist die Durchsetzung von Sicherheitsfunktionen und Zugriffsrichtlinien in Netzwerken mit mehr als nur ein paar Computern schwierig. Benutzerkonten und Zugriffsrechte müssen auf jedem Peer-Gerät separat konfiguriert werden.

Peer-to-Peer-Anwendungen

Anders als ein Peer-to-Peer-Netzwerk ermöglicht eine Peer-to-Peer-Anwendung es einem Gerät, innerhalb derselben Kommunikationssitzung als Client *und* als Server zu agieren. In diesem Modell ist jeder Client ein Server und jeder Server ein Client. Abbildung 3.10 veranschaulicht dies anhand zweier Telefone, die zum selben Netzwerk gehören und eine Instant Messaging-Nachricht versenden. Die Linien am oberen Rand der Abbildung stellen den digitalen Datenaustausch zwischen den beiden Geräten dar. Beide können einen Kommunikationsvorgang einleiten und gelten dabei als gleichberechtigt. Allerdings ist es für Peer-to-Peer-Anwendungen notwendig, dass jedes Endgerät eine Benutzeroberfläche bereitstellt und einen Hintergrunddienst ausführt. Wenn Sie eine bestimmte Peer-to-Peer-Anwendung starten, werden Benutzeroberfläche und Hintergrunddienste aufgerufen. Danach können die Geräte direkt miteinander kommunizieren.

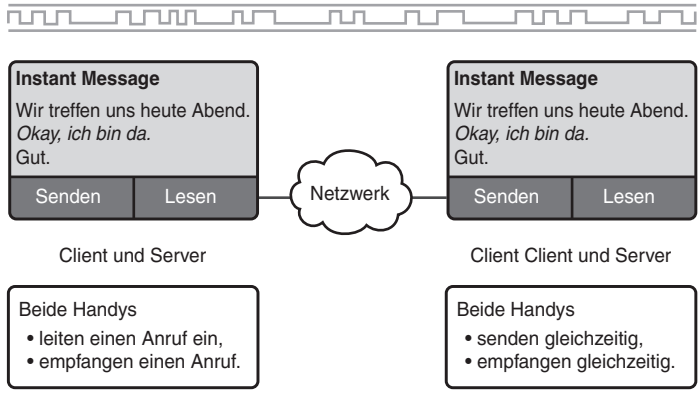


Abbildung 3.10: Peer-to-Peer-Anwendungen

Eine Form der Peer-to-Peer-Anwendung ist ein sogenanntes P2P-Hybridsystem, welches – obwohl sich die freigegebenen Dateien auf separaten Host-Computern befinden – ein zentralisiertes Verzeichnis verwendet: den Indexserver. Jeder Peer muss auf den Indexserver zugreifen, um die Position einer Ressource zu ermitteln, die auf einem anderen Peer gespeichert ist. Der Indexserver kann auch dabei helfen, zwei Peers miteinander zu verbinden; danach jedoch erfolgt die Kommunikation zwischen diesen beiden Peers, ohne dass der Indexserver noch einmal kontaktiert würde.

Peer-to-Peer-Anwendungen können in Peer-to-Peer-Netzwerken, Client/Server-Netzwerken und über das Internet verwendet werden.

3.3 Beispiele für Protokolle und Dienste der Anwendungsschicht

Nachdem Sie nun einen Überblick über die Frage erhalten haben, wie Anwendungen eine Schnittstelle für den Benutzer bereitstellen und Zugriff auf ein Netzwerk vermitteln, wollen wir uns nun einige häufig verwendete Protokolle näher ansehen.

Wie Sie im weiteren Verlauf dieses Buches noch erfahren werden, verwendet die Transportschicht ein Adressierungsschema: die Port-Nummer. Port-Nummern bezeichnen Anwendungen und Dienste der Anwendungsschicht, die Absender und Empfänger der Daten sind. Serverprogramme verwenden gemeinhin vordefinierte Port-Nummern, die den meisten Clients bekannt sind. Wenn Sie die verschiedenen Protokolle und Dienste der TCP/IP-Anwendungsschicht untersuchen, werden Sie auch die TCP- und UDP-Port-Nummern verwenden, die gewöhnlich mit diesen Diensten verknüpft sind. Tabelle 3.1 zeigt einige dieser Dienste.

Tabelle 3.1: Dienste und Port-Nummern

Dienst	Port-Nummern
DNS	TCP-/UDP-Port 53
HTTP	TCP-Port 80
SMTP	TCP-Port 25
POP3	UDP-Port 110
Telnet	TCP-Port 23
DHCP	UDP-Port 67
FTP	TCP-Ports 20 und 21

Im nächsten Abschnitt wollen wir DNS, WWW-Dienste und HTTP genauer unter die Lupe nehmen.

3.3.1 DNS-Dienste und DNS-Protokoll

In Datennetzen erhalten Geräte IP-Adressen, um Nachrichten über das Netzwerk senden und empfangen zu können. Allerdings fällt es den meisten Menschen schwer, sich diese numerischen Adressen zu merken. Aus diesem Grund wird das Prinzip der Domänen-Namen benutzt: Die numerischen Adresswerte werden in einen einfachen, leicht zu merkenden Namen konvertiert.

Im Internet sind Domänen-Namen wie *http://www.cisco.com* weitaus leichter zu merken als 198.132.219.25 – der zum gegenwärtigen Zeitpunkt gültigen IP-Adresse des unter dem genannten Domänen-Namen erreichbaren Servers. Zudem wäre ein Adresswechsel durch Cisco für die Benutzer transparent, denn der Domänen-Name lautet weiterhin *http://www.cisco.com* – schließlich wird einfach nur die neue Adresse mit dem vorhandenen Domänen-Namen verknüpft, und die Konnektivität bleibt vorhanden (siehe Abbildung 3.11). Als Netzwerke noch klein waren, war die Zuordnung von Domänen-Namen und zugehörigen Adressen eine einfache Angelegenheit. Mittlerweile jedoch haben die Netzwerke an Größe zugenommen, und folglich ist auch die Anzahl der Geräte angestiegen, weswegen eine manuelle Verwaltung heute nicht mehr in Frage kommt.

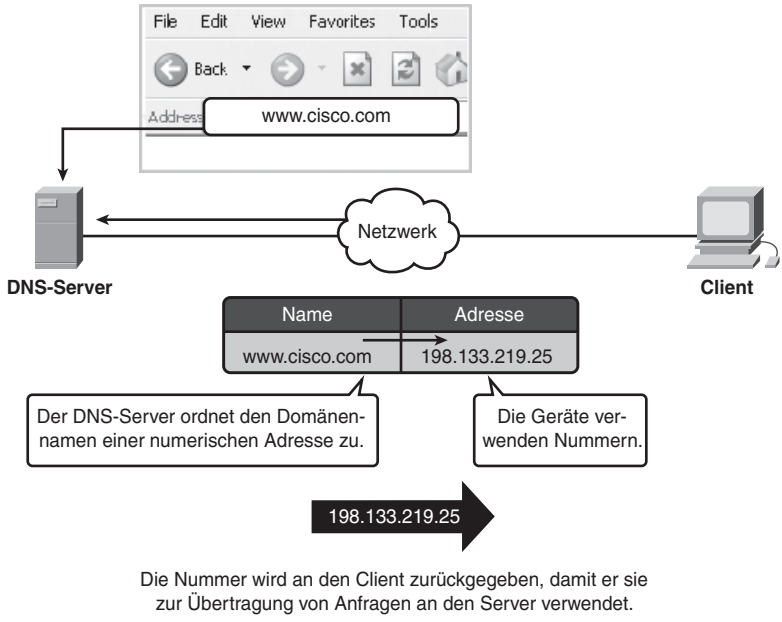


Abbildung 3.11: Auflösung einer DNS-Adresse

DNS wurde zur Auflösung von Namen in IP-Adressen entwickelt. Hierzu wird eine Anzahl verteilter Server verwendet, deren Aufgabe die Auflösung der mit den Adressen verknüpften Namen ist.

Wie DNS funktioniert

Das DNS-Protokoll definiert einen automatisierten Dienst, der einen Ressourcennamen mit der jeweils zugehörigen numerischen Netzadresse verknüpft. Es umfasst das Format für Abfragen und Antworten sowie Datenformate. Die DNS-Kommunikation verwendet ein Nachrichtenformat, welches für alle Arten von Clientanforderungen und Server-Antworten, Fehlermeldungen und die Übertragung von Ressource-Records (dt. *Resourceneinträge*) zwischen Servern verwendet wird.

DNS ist ein Client/Server-Dienst, unterscheidet sich jedoch von den anderen Client/Server-Diensten, die Sie kennengelernt haben oder noch kennenlernen werden. Während andere Dienste einen Client verwenden, der eine Anwendung ist (Webbrowser, Mailclient usw.), wird der DNS-Dienst selbst als Client ausgeführt. Der DNS-Client – manchmal auch DNS-Resolver genannt – unterstützt die Namensauflösung für andere Netzwerkanwendungen und Dienste, die eine solche benötigen.

Wenn Sie ein Netzwerkgerät konfigurieren, geben Sie meist eine oder mehrere DNS-Server-Adressen an, die der DNS-Client zur Namensauflösung verwenden kann. Gewöhnlich erhalten Sie die Adressangaben der DNS-Server von Ihrem Internetprovider. Wenn die Anwendung eines Benutzers die Herstellung der Verbindung zu einem Ziel im Internet verlangt und dabei einen Namen nennt, fragt der anfordernde DNS-Client einen dieser DNS-Server ab, um den Namen in eine numerische Adresse aufzulösen.

Computerbetriebssysteme verfügen auch über ein Utility namens Nslookup, mit dem der Benutzer an den Namensserver manuell eine Abfrage stellen kann, um einen gegebenen Hostnamen aufzulösen. Dieses Utility können Sie auch verwenden, um Probleme in Verbindung mit der Namensauflösung zu beheben und den aktuellen Status der Namensserver zu überprüfen.

In Listing 3.1 wird nach Absetzen des Befehls `nslookup` der für Ihren Host vorkonfigurierte DNS-Server angezeigt. Hier heißt der Server *dns-sjk.cisco.com* und hat die Adresse 171.68.226.120.

Listing 3.1: Nslookup

```
Microsoft Windows XP [Version 5.1.2600]
 Copyright 1985-2001 Microsoft Corp.

C:\> nslookup

Default Server: dns-sjk.cisco.com
Address: 171.68.226.120
>www.cisco.com
Server: dns-sj.cisco.com
Address: 171.70.168.183

Name:  www.cisco.com
Address: 198.133.219.25
```

Sie können nachfolgend den Namen eines Hosts oder einer Domäne eingeben, deren Adresse Sie erhalten wollen. Die erste Abfrage in Listing 3.1 bezieht sich auf *www.cisco.com*. Der antwortende Namensserver gibt die Adresse 198.133.219.25 zurück.

Zwar zeigt das Beispiel nur einfache Abfragen, doch der Befehl `nslookup` verfügt über zahlreiche Optionen für umfassende Tests und die Überprüfung des DNS-Prozesses.

Namensauflösung und Caches

Ein DNS-Server ermöglicht die Namensauflösung mithilfe des Namens-Daemons, der meist kurz als *named* (ausgesprochen »näim-dih«) bezeichnet wird. Der DNS-Server agiert als »Telefonbuch« des Internets: Er übersetzt für das menschliche Auge bestimmte Hostnamen (z. B. *http://www.cisco.com*) in die IP-Adressen, die Netzwerkgeräte benötigen, um Daten zu übermitteln.

Der DNS-Server speichert verschiedene Arten von Ressource-Records, um Namen aufzulösen. Diese Einträge enthalten den Namen, die Adresse und den Typ des Eintrags.

Zu den Typen von Einträgen gehören beispielsweise die folgenden:

- **A:** Adresse eines Hosts
- **NS:** Maßgeblicher Namensserver
- **CNAME:** Der kanonische Name (oder vollqualifizierte Domänen-Name, kurz FQDN) eines Alias; wird verwendet, wenn mehrere Dienste dieselbe Netzwerkadresse haben, aber jeder Dienst über einen eigenen Eintrag im DNS verfügt
- **MX:** Mail-Exchange-Eintrag, der einen Domänen-Name mit einer Liste von Mail-Exchange-Servern der betreffenden Domäne verknüpft

Wenn ein Client eine Abfrage absetzt, schlägt der *named*-Prozess zunächst in seinen eigenen Einträgen nach, um festzustellen, ob er den Namen so auflösen kann. Ist dies nicht möglich, kontaktiert er andere Server, um die Namensauflösung durchzuführen.

Die Anforderung kann über mehrere Server weitergeleitet werden, was Zeit und Bandbreite kostet. Deswegen legt, wenn ein Ergebnis gefunden und an den ursprünglichen Server zurückgegeben wird, der Server die IP-Adresse, die dem Namen entspricht, in einem als Cache bezeichneten Zwischenspeicher ab. Wird derselbe Name später noch einmal angefordert, dann kann der erste Server als Adresse den im Cache gespeicherten Wert zurückgeben. Diese Zwischenspeicherung verringert sowohl das Datenaufkommen für DNS-Abfragen im Netzwerk als auch die Belastung von Servern weiter oben in der Hierarchie. Der DNS-Client-Dienst auf Windows-PCs optimiert die Leistung der DNS-Namensauflösung, indem er zuvor aufgelöste Namen auch im Speicher ablegt. Der Befehl `ipconfig /displaydns` zeigt alle im Cache vorhandenen DNS-Einträge auf einem Windows XP- oder Windows 2000-Computersystem an.

DNS-Hierarchie

DNS verwendet ein hierarchisches System zur Erstellung einer Namensdatenbank, um Namen aufzulösen. Die Hierarchie sieht wie ein auf dem Kopf stehender Baum aus, bei dem sich der Stamm oben und die Äste unten befinden.

Ganz oben in der Hierarchie befinden sich die Root-Server, die Daten dazu speichern, wie man die Top-Level-Domänenserver erreicht; diese wiederum verfügen über Einträge, die auf Second-Level-Domänenserver verweisen usw.

Die verschiedenen Top-Level-Domänen (d. h. Domänen oberster Ebene) bezeichnen entweder ein Land oder einen Organisationstyp. Beispiele für Top-Level-Domänen sind etwa:

- **.au:** Australien
- **.co:** Kolumbien
- **.com:** Unternehmen oder Industriezweig
- **.jp:** Japan
- **.org:** gemeinnützige Organisation

Auf die Top-Level-Domänen folgen die Second-Level-Domänen und diesen weitere untergeordnete Domänen. Ein gutes Beispiel hierfür ist der Domänen-Name *http://www.cisco.netacad.net*. Hierbei ist *.net* die Top-Level-Domäne, *.netacad* die Second-Level-Domäne und *.cisco* eine weitere untergeordnete Ebene.

Jeder Domänen-Name ist ein Pfad, der sich beginnend beim Stamm von oben nach unten durch die Baumstruktur schlängelt. Betrachten Sie etwa Abbildung 3.12: Unter Umständen wissen die DNS-Root-Server nicht genau, wo sich der Mailserver *mail.cisco.com* befindet, aber sie verfügen über einen Eintrag für die Domäne *.com* in der Top-Level-Domäne. Analog haben die Server in der Domäne *.com* keinen Eintrag für *mail.cisco.com*, wohl aber für die Second-Level-Domäne *cisco.com*. Die Server in der Domäne *cisco.com* verfügen dann über den notwendigen Eintrag für *mail.cisco.com* (bei dem es sich genau genommen um einen MX-Eintrag handelt).

DNS bedient sich zur Speicherung und Pflege dieser Ressourceneinträge der beschriebenen Hierarchie dezentraler Server. Die Ressourceneinträge geben Domänen-Namen, die der Server auflösen kann, sowie alternative Server an, die Anforderungen bearbeiten können. Wenn ein gegebener Server über Ressourceneinträge verfügt, die seiner Stufe in der Domänen-Hierarchie entsprechen, dann bezeichnet man ihn als autoritativ (maßgeblich) für diese Einträge.

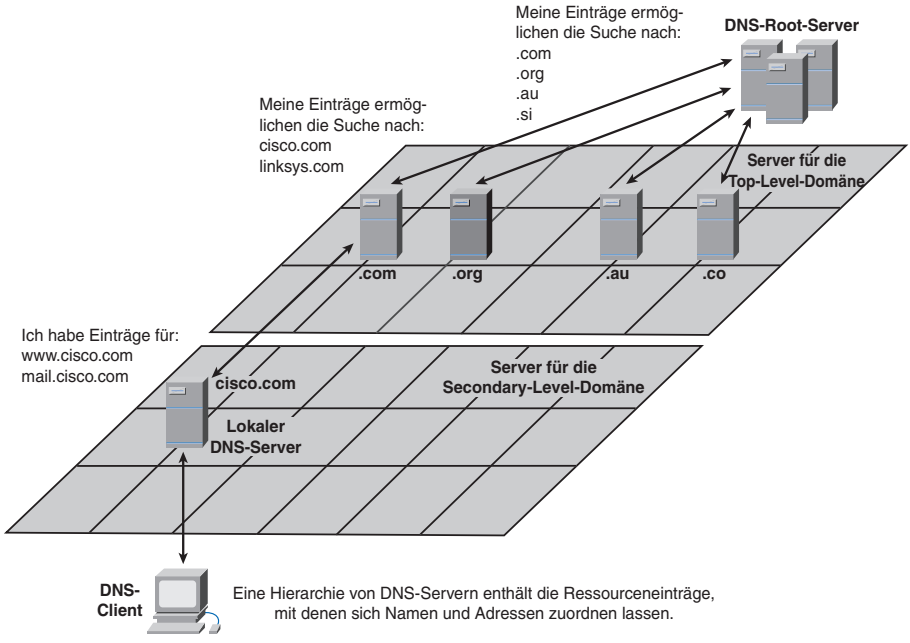


Abbildung 3.12: DNS-Serverhierarchie

So wäre beispielsweise ein Namensserver in der Domäne *cisco.netacad.net* nicht maßgeblich für den Eintrag *mail.cisco.com*, weil dieser Eintrag sich auf einen Server auf einer höheren Domänen-Stufe bezieht (nämlich die Domäne *cisco.com*).

Anmerkung

Zwei wichtige RFCs zum DNS-Protokoll finden Sie unter den folgenden Adressen:

- *http://www.ietf.org/rfc/rfc1034.txt*
- *http://www.ietf.org/rfc/rfc1035.txt*

RFCs sind Standarddokumente, die neue Forschungen, Erfindungen und Methoden im Zusammenhang mit Internettechnologien beschreiben. Diese RFCs sind ausgesprochen technisch gehalten, bieten Ihnen aber einen tieferen Einblick in die Details dieser Standards.

3.3.2 Der WWW-Dienst und HTTP

Wenn ein URL (Uniform Ressource Locator, d. h. eine Webadresse) in einen Browser eingegeben wird, stellt dieser über HTTP eine Verbindung zu dem Webdienst her, der auf dem Server ausgeführt wird. URLs und URIs (Uniform Resource Identifier) sind Bezeichnungen für das Prinzip, das die meisten Menschen Webadresse nennen.

Der URL *http://www.cisco.com/index.html* bezeichnet eine bestimmte Ressource, nämlich eine Webseite namens *index.html* auf einem Server mit der Bezeichnung *cisco.com*.

Webbrowser sind die Clientanwendungen, die Computer benutzen, um eine Verbindung mit dem World Wide Web herzustellen und auf Ressourcen zuzugreifen, die auf einem Webserver gespeichert sind. Wie die meisten Serverprozesse wird auch der Webserver als Hintergrunddienst ausgeführt und stellt unterschiedliche Dateitypen bereit.

Um auf Inhalte zugreifen zu können, stellen Webclients Verbindungen zum Server her und fordern die gewünschten Ressourcen an. Der Server reagiert mit einer Übermittlung der Ressourcen. Werden diese vom Browser empfangen, so interpretiert dieser die Daten und stellt sie für den Benutzer dar.

Browser können viele Datentypen interpretieren und darstellen, z. B. einfache Textdateien oder auch HTML-Dateien (HTML ist die Sprache, mit der Webseiten erstellt werden). Andere Datentypen hingegen erfordern einen anderen Dienst oder ein anderes Programm, das meist Plug-in oder Add-on genannt wird. Damit der Browser ermitteln kann, welcher Art die empfangene Datei ist, gibt der Server an, welchen Typ von Daten die Datei enthält.

Um besser verstehen zu können, wie Webbrowser und Webclient interagieren, wollen wir uns einmal ansehen, wie eine Webseite in einem Browser geöffnet wird. Betrachten Sie zu diesem Zweck den URL *http://www.cisco.com/web-server.htm*.

Zunächst interpretiert der Browser die drei Bestandteile des URL:

- *http*: Protokoll oder Schema
- *www.cisco.com*: Servername
- *web-server.htm*: Angeforderter Dateiname

Der Browser fragt dann einen Namensserver ab, um *http://www.cisco.com* in eine numerische Adresse umzuwandeln und mit dieser den Server zu kontaktieren. Entsprechend den Vorgaben von HTTP sendet der Browser eine GET-Anforderung an den Server und verlangt die Übertragung der Datei *web-server.htm*. Der Server seinerseits sendet den HTML-Code dieser Webs-

eite an den Browser. Schließlich entschlüsselt der Browser den empfangenen HTML-Code und formatiert die Seite für das Browserfenster.

HTTP ist eines der Protokolle in der TCP/IP-Protokollfamilie. Es wurde ursprünglich zum Veröffentlichen und Abrufen von HTML-Seiten geschrieben und wird nun für verteilte, kollaborative Datensysteme verwendet. HTTP wird im gesamten World Wide Web zur Datenübertragung benutzt und ist eines der meistverwendeten Anwendungsprotokolle.

HTTP beschreibt ein Anforderungs-Antwort-Protokoll. Wenn ein Client (meist ein Webbrowser) eine Anforderungsnachricht an einen Server sendet, definiert das HTTP-Protokoll die Nachrichtentypen, die der Client zum Anfordern der Webseite verwendet, sowie die Nachrichtentypen, die der Server zur Beantwortung der Anforderungen benutzt. Es gibt drei gängige Nachrichtentypen:

- GET
- POST
- PUT

GET ist eine Clientanforderung von Daten. Ein Webbrowser sendet die GET-Nachricht, um Seiten bei einem Webserver anzufordern. Wie Abbildung 3.13 zeigt, antwortet der Server, wenn er die GET-Anforderung erhält, mit einer Statuszeile (z. B. *HTTP/1.1 200 OK*) und einer eigenen Nachricht, deren Nachrichtenkörper die angeforderte Datei, eine Fehlermeldung oder andere Daten sein können.

POST und PUT dienen der Übermittlung von Nachrichten an den Webserver, mit denen Daten hochgeladen werden. Wenn der Benutzer beispielsweise Daten in ein Formular eingibt, das in eine Webseite integriert ist, schließt POST diese Daten in die an den Server gesendete Nachricht mit ein. PUT hingegen lädt Ressourcen oder Inhalte auf den Webserver.

HTTP ist zwar ein bemerkenswert flexibles, aber keineswegs sicheres Protokoll. POST-Nachrichten werden in unverschlüsselter Form auf den Server hochgeladen und können abgefangen und mitgelesen werden. Auch die Antworten des Servers – meist HTML-Seiten – sind unverschlüsselt.

Deswegen wird zur sicheren Kommunikation über das Internet das HTTPS-Protokoll (Secure HTTP, sicheres HTTP) eingesetzt, um auf Webserverdaten zugreifen und Daten an den Webserver senden zu können. HTTPS verwendet Authentifizierungs- und Verschlüsselungsverfahren zur Absicherung von Daten, wenn diese zwischen Server und Client ausgetauscht werden. Dabei beschreibt HTTPS zusätzliche Regeln zur Übermittlung von Daten zwischen der Anwendungs- und der Transportschicht.

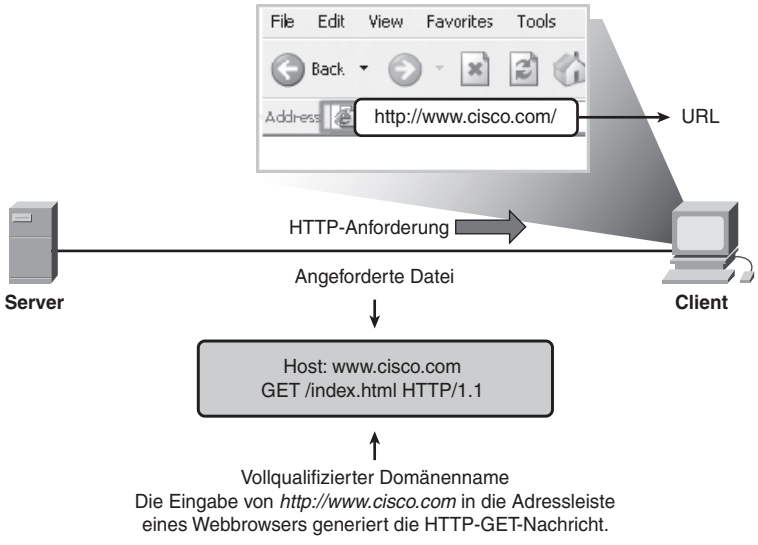


Abbildung 3.13: Die Methode GET des HTTP-Protokolls

Packet Tracer Aktivität

Netzwerkdarstellungen (3.3.2.3)

Bei dieser Aktivität konfigurieren Sie DNS- und HTTP-Dienste und analysieren die Pakete, die infolge der Anforderung einer Webseite durch Eingabe eines URL entstehen. Zur Durchführung der Aktivität verwenden Sie den Packet Tracer und die Datei *e1-3323.pka* auf der Begleit-CD-ROM zu diesem Buch.

3.3.3 E-Mail-Dienste und SMTP-/POP-Protokolle

E-Mail – der wohl beliebteste aller Netzwerkdienste – hat die Kommunikation der Menschen durch seine Einfachheit und Geschwindigkeit revolutioniert. Damit E-Mail aber auf einem Computer oder einem anderen Endgerät ausgeführt werden kann, sind diverse Anwendungen und Dienste erforderlich. Zwei Beispiele für Anwendungsschichtprotokolle sind POP (Post Office Protocol) und SMTP (Simple Mail Transfer Protocol). Wie HTTP definieren auch diese Protokolle Client/Server-Prozesse.

POP und POP3 (POP Version 3) sind Zustellungsprotokolle für eingehende Mail. Hierbei handelt es sich um typische Client/Server-Protokolle, die E-Mails vom Mailserver zum Client übertragen.

SMTP hingegen regelt die Übertragung ausgehender Mail vom Client zum Mailserver sowie den Transport von Mail zwischen Mailservern. SMTP gestattet das Übermitteln von Mail über Datennetze zwischen unterschiedlichen Server- und Clientanwendungen und macht den Austausch von E-Mail über das Internet erst möglich.

Wenn Benutzer E-Mails verfassen, verwenden sie hierzu gewöhnlich einen MUA (Mail User Agent) – einfach gesagt einen Mailclient. Der MUA gestattet das Versenden von Nachrichten und legt empfangene Nachrichten im Postfach des Clients ab. Hierbei handelt es sich um separate Prozesse (siehe Abbildung 3.14).

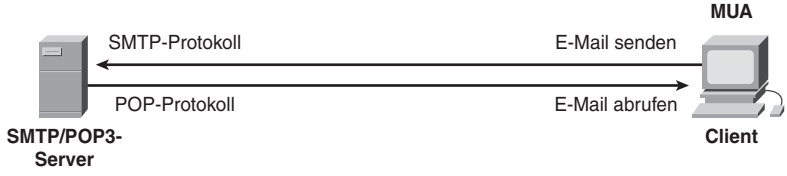


Abbildung 3.14: E-Mail-Client (MUA)

Um Mails von einem Mailserver zu erhalten, kann der Mailclient POP verwenden. Beim Versenden von Mail von einem Client oder einem Server kommen Nachrichtenformate und Befehls-Strings zum Einsatz, die im SMTP-Protokoll definiert sind. Gewöhnlich fasst ein Mailclient die Funktionalitäten beider Protokolle in einer Anwendung zusammen.

3.3.4 E-Mail-Server-Prozesse: MTA und MDA

Der E-Mail-Server führt zwei separate Prozesse aus:

- MTA (Mail Transfer Agent)
- MDA (Mail Delivery Agent)

Der MTA-Prozess wird zur Weiterleitung von E-Mail verwendet. Wie Abbildung 3.15 zeigt, empfängt der MTA Nachrichten vom MUA oder von einem anderen MTA auf einem anderen Mailserver. Dem Nachrichten-Header entnimmt er, wie eine Nachricht weitergeleitet werden muss, um ihr Ziel zu erreichen. Ist diese Mail an einen Benutzer gerichtet, dessen Postfach sich auf dem lokalen Server befindet, dann wird sie an den MDA übergeben. Befindet sich das Postfach des Mail-Empfängers hingegen nicht auf dem lokalen Server, dann leitet der MTA die Mail an den MTA des betreffenden Servers weiter.

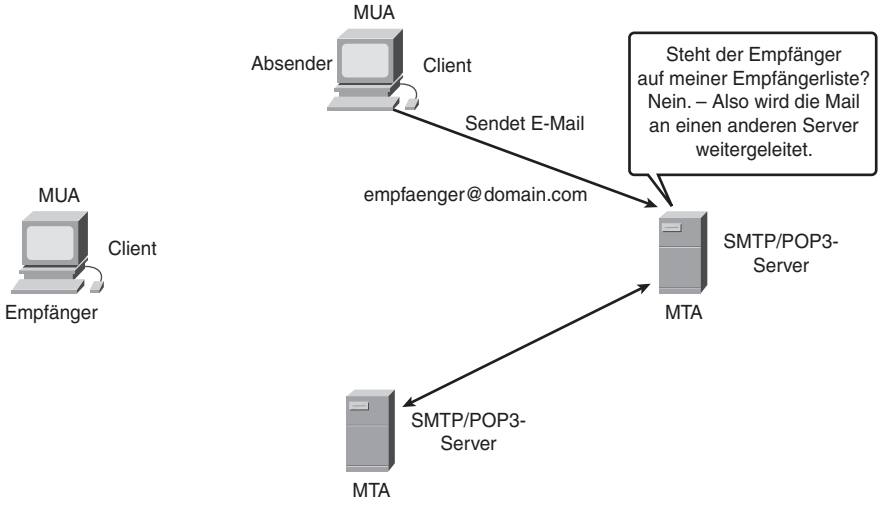


Abbildung 3.15: E-Mail-Server: MTA

In Abbildung 3.16 sehen Sie, dass der MDA eine E-Mail von einem MTA entgegennimmt und die Zustellung durchführt. Der MDA empfängt alle eingehenden Mails vom MTA und legt diese in den Postfächern der jeweiligen Benutzer ab. Der MDA beseitigt auch abschließende Zustellungsprobleme, z. B. Virenprüfung, Spam-Filterung und die Verarbeitung von Empfangsbestätigungen.

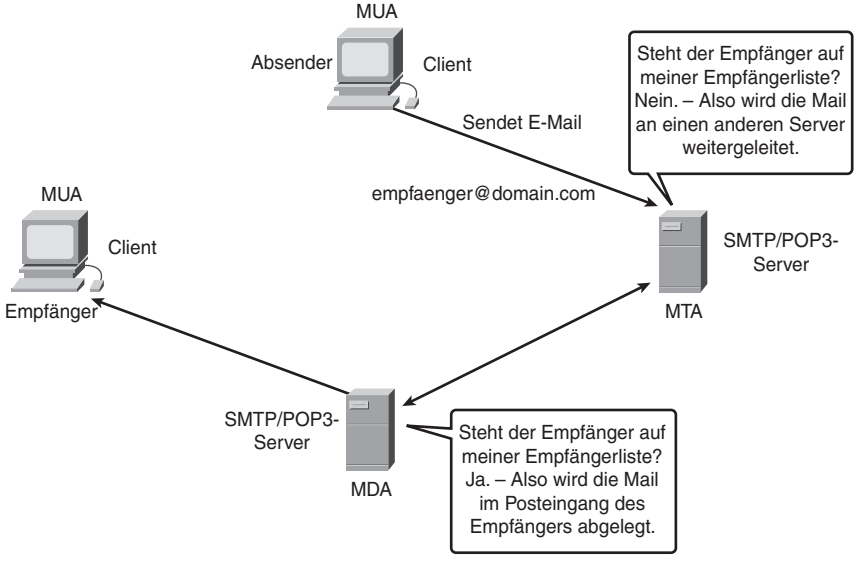


Abbildung 3.16: E-Mail-Server: MDA

Bei den meisten E-Mail-bezogenen Kommunikationsvorgängen kommen MUA-, MTA- und MDA-Anwendungen zum Einsatz. Es gibt aber auch Alternativen zur Mail-Zustellung. Ein Client kann beispielsweise an ein firmeninternes E-Mail-System wie IBM Lotus Notes, Novell Groupwise oder Microsoft Exchange angeschlossen sein. Diese Systeme weisen häufig ein eigenes internes E-Mail-Format auf, und die Clients kommunizieren mit dem Mailserver in der Regel über ein proprietäres Protokoll.

Der Server sendet und empfängt E-Mails über das Internet mithilfe eines Mail-Gateways; das Gateway übernimmt alle erforderlichen Formatierungsarbeiten. Wenn beispielsweise zwei Benutzer, die für dasselbe Unternehmen tätig sind, E-Mails miteinander austauschen wollen und hierfür ein proprietäres Protokoll verwenden, verbleiben ihre Nachrichten vollständig innerhalb des firmeninternen E-Mail-Systems.

Eine andere Alternative für Computer, auf denen kein MUA ausgeführt wird, besteht darin, eine Verbindung mit einem Maildienst über einen Webbrowser herzustellen und auf diese Weise Nachrichten zu empfangen und zu senden. Einige Computer verwenden einen eigenen MTA und verwalten Domänen übergreifende E-Mail selbst.

Das SMTP-Protokollnachrichtenformat verwendet eine feste Anzahl von Befehlen und Antworten. Diese Befehle unterstützen die in SMTP verwendeten Prozeduren, so etwa die Sitzungseinleitung, Mailtransaktionen, Mailweiterleitung, die Überprüfung von Postfachnamen, die Erweiterung von Mailinglisten und alle einleitenden und abschließenden Vorgänge für den Austausch.

Es folgen einige der im SMTP-Protokoll spezifizierten Befehle:

- **HELO:** Weist den SMTP-Clientprozess gegenüber dem SMTP-Serverprozess aus.
- **EHLO:** Ist eine neue HELO-Version, die Diensterweiterungen enthält.
- **MAIL FROM:** Bezeichnet den Absender.
- **RCPT TO:** Bezeichnet den Empfänger.
- **DATA:** Bezeichnet den Nachrichtenkörper.

3.3.5 FTP

Ein anderes häufig verwendetes Protokoll der Anwendungsschicht ist FTP (File Transfer Protocol). FTP wurde entwickelt, um Dateiübertragungen zwischen einem Client und einem Server zu ermöglichen. Ein FTP-Client ist eine Anwendung, die auf einem Computer läuft, der zum Hoch- und Herunter-

laden von Daten auf bzw. von einem Server dient, auf dem der FTP-Daemon (FTPd) ausgeführt wird.

Um Dateien erfolgreich zu übertragen, erfordert FTP zwei Verbindungen zwischen dem Client und dem Server: eine für Befehle und Antworten, und eine zweite für die eigentliche Dateiübertragung.

Der Client stellt die erste Verbindung zum Server auf TCP-Port 21 her. Diese Verbindung besteht aus Clientbefehlen und Serverantworten und dient zur Steuerung des Datenverkehrs.

Dann stellt der Client die zweite Verbindung zum Server auf TCP-Port 20 her. Diese Verbindung ist für den eigentlichen Dateitransfer vorgesehen und wird für jede übertragene Datei neu erstellt.

Die Datenübertragung kann in beiden Richtungen erfolgen (siehe Abbildung 3.17). Der Client kann eine Datei vom Server herunterladen (Pull) oder auf den Server hochladen (Push).

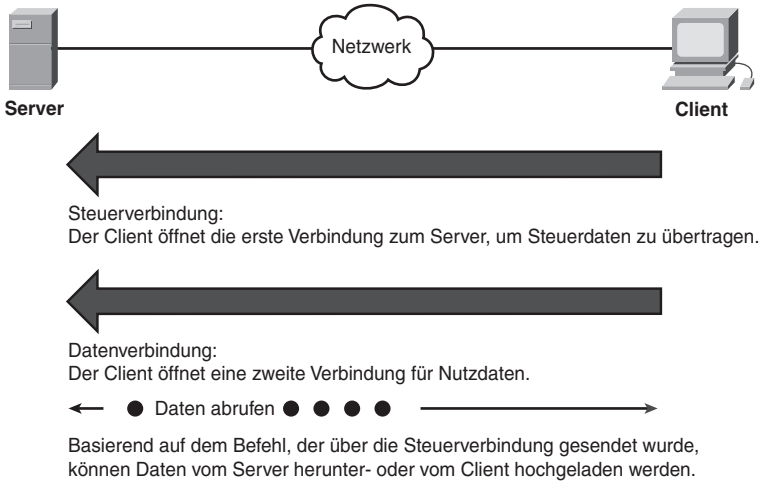


Abbildung 3.17: FTP-Prozess

3.3.6 DHCP

DHCP (Dynamic Host Configuration Protocol) ermöglicht es Clients in einem Netzwerk, IP-Adressen und andere Angaben über einen DHCP-Server zu beziehen. Das Protokoll automatisiert die Zuweisung von IP-Adressen, Subnetzmasken, Gateways und anderen IP-spezifischen Netzwerkparametern.

Mit DHCP kann ein Host eine IP-Adresse dynamisch beziehen, sobald er eine Verbindung zum Netzwerk herstellt. Der DHCP-Server wird dabei kon-

taktiert, indem eine Anforderung für eine IP-Adresse gesendet wird. Der DHCP-Server wählt dann eine Adresse aus einem als *Pool* bezeichneten vorkonfigurierten Adressbereich aus und weist sie dem Hostclient für eine bestimmte Zeit zu.

In größeren Netzwerken, LANs und immer dort, wo die Benutzer häufig wechseln, ist DHCP zu bevorzugen. Neue Benutzer erscheinen mit Laptops und benötigen eine Verbindung. Andere erhalten neue Workstations, die angeschlossen werden müssen. Statt nun einen Netzwerkadministrator jedes Mal neue IP-Adressen für diese neuen Ressourcen konfigurieren zu lassen, ist es weitaus effizienter, dies automatisch mithilfe von DHCP erledigen zu lassen.

Wenn ein DHCP-konfiguriertes Gerät gestartet wird oder sich mit dem Netzwerk verbindet, versendet der Client ein Broadcast-Paket namens DHCP DISCOVER, um im Netzwerk eventuell vorhandene DHCP-Server zu ermitteln. Ein DHCP-Server antwortet in einem solchen Fall mit einem DHCP OFFER-Paket; dies ist eine Nachricht, die ein Lease-Angebot mit reservierter IP-Adresse, Subnetzmaske, DNS-Serveradresse und Angaben zum Default-Gateway sowie zur Gültigkeitsdauer der Lease enthält.

Via DHCP verteilte Adressen werden den Hosts nicht dauerhaft zugewiesen, sondern nur für eine bestimmte Zeit. Wird der Host heruntergefahren oder vom Netzwerk getrennt, so wird die Adresse dem Pool zur erneuten Verwendung wieder zugeführt. Dies ist besonders praktisch bei mobilen Benutzern, die sich immer wieder neu mit dem Netzwerk verbinden. Benutzer können ihren Standort nach Belieben wechseln und die Netzwerkverbindungen dann neu herstellen. Der Host kann nach dem Herstellen der Hardwareverbindung eine IP-Adresse entweder über ein kabelgestütztes oder drahtloses LAN beziehen.

DHCP ermöglicht auch den Zugriff auf das Internet über Funk-Hotspots am Flughafen oder im Café. Sobald Sie in die Reichweite des Hotspots gelangen, kontaktiert der DHCP-Client auf Ihrem Laptop den lokalen DHCP-Server über eine Funkverbindung. Der Server weist Ihrem Laptop dann eine IP-Adresse zu.

Verschiedene Gerätearten kommen als DHCP-Server in Frage, sofern sie eine DHCP-Dienst-Software ausführen. In den meisten mittelgroßen bis großen Netzwerken ist der DHCP-Server gewöhnlich ein lokaler dedizierter Server, der auf einem PC läuft.

Bei Heimnetzen befindet er sich hingegen meist beim Internetprovider, und ein Host in diesem Heimnetz bezieht seine IP-Konfiguration direkt von diesem Provider.

Viele Heimnetze und kleine Unternehmen verwenden auch ein ISR-Gerät (Integrated Services Router) zur Verbindung mit dem Provider. In diesem Fall ist der ISR sowohl ein DHCP-Client als auch ein DHCP-Server. Als Client bezieht er seine IP-Konfiguration vom Provider und agiert dann als DHCP-Server für die internen Hosts des lokalen Netzwerks.

Abbildung 3.18 zeigt die verschiedenen Möglichkeiten zur Aufstellung von DHCP-Servern.

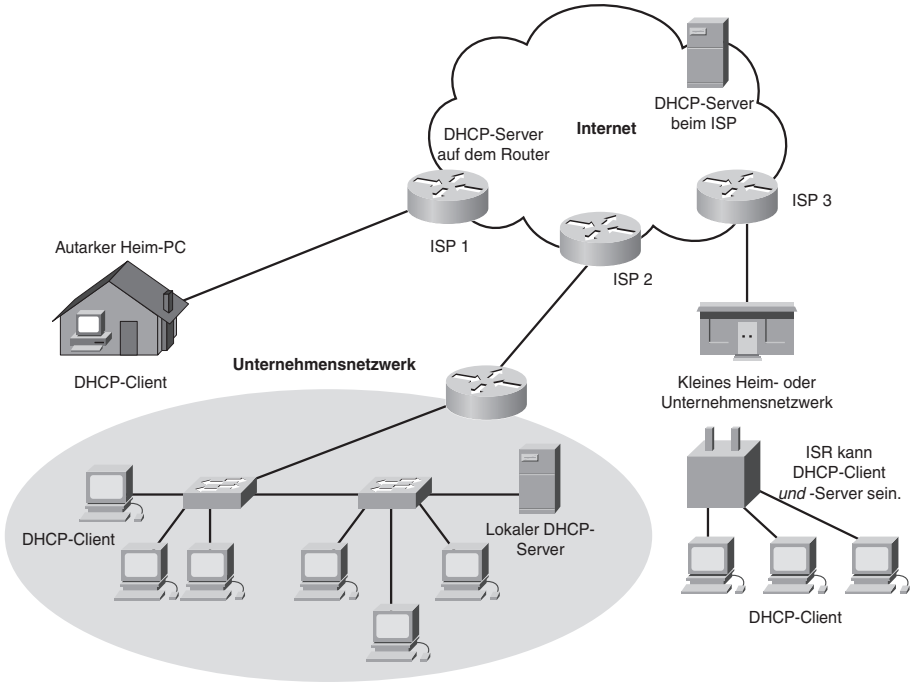


Abbildung 3.18: DHCP-Server

DHCP kann ein Sicherheitsrisiko darstellen, weil jedes Gerät, das an das Netzwerk angeschlossen wird, eine Adresse erhält. Dieses Risiko macht die physische Absicherung zu einem wichtigen Faktor bei der Entscheidung, ob eine dynamische oder eine statische (manuelle) Adressierung verwendet werden soll.

Dynamische und statische Adressierungsmethoden haben jeweils ihren Platz im Netzdesign. Viele Netzwerke verwenden sowohl DHCP als auch die statische Adressierung. DHCP wird dabei oft für die normalen Hosts der Benutzer verwendet; statische (feste) Adressen hingegen kommen für Netzwerkgeräte wie Gateways, Switches, Server und Drucker zum Einsatz.

Der Client kann mehrere DHCP OFFER-Pakete empfangen, sofern sich mehrere DHCP-Server im lokalen Netzwerk befinden. Der Client muss zwischen diesen wählen und dann als Broadcast ein DHCP REQUEST-Paket senden, welches den gewählten Server mit der jeweiligen Lease angibt.

Ein Client kann außerdem fordern, dass er eine Adresse, die ihm bereits zuvor zugewiesen wurde, erneut erhält. Sofern die vom Client beantragte oder vom Server angebotene IP-Adresse noch gültig ist, sendet der betreffende Server eine DHCP ACK-Nachricht (Bestätigungsnachricht) zurück. Hierdurch weiß der Client, dass die Lease verlängert wurde. Ist das Angebot nicht mehr gültig, weil etwa eine Zeitüberschreitung stattgefunden oder ein anderer Client die Lease erhalten hat, muss der ausgewählte Server dem Client eine DHCP NAK-Nachricht (Negativbestätigung) zuschicken. Hält der Client die Lease, so muss diese vor ihrem Ablauf mithilfe einer weiteren DHCP REQUEST-Nachricht erneuert werden. Der DHCP-Server trägt dafür Sorge, dass alle IP-Adressen eindeutig sind. (Eine IP-Adresse darf nicht gleichzeitig zwei verschiedenen Netzwerkgeräten zugewiesen werden.)

3.3.7 Dateifreigabe und das SMB-Protokoll

SMB (Server Message Block) ist ein Client/Server-basiertes Dateifreigabeprotokoll. IBM hat SMB in den spätern 1980er-Jahren entwickelt, um die Struktur freigegebener Netzwerkressourcen (Verzeichnisse, Dateien, Drucker und serielle Ports) zu beschreiben. Es handelt sich um ein Anfrage-Antwort-Protokoll. Anders als via FTP unterstützte Dateifreigaben stellen Clients hier eine längerfristige Verbindung zu Servern her. Nachdem die Verbindung hergestellt ist, kann der Benutzer des Clients auf die Serverressourcen so zugreifen, als ob sich die Ressource lokal auf dem Host des Clients befände.

SMB-Dateidienste und SMB-Druckdienste sind zur wichtigsten Stütze von Microsoft-Netzwerken geworden. Mit der Einführung der Windows 2000-Betriebssysteme änderte Microsoft die zugrunde liegende Struktur für die Verwendung von SMB. In früheren Versionen von Microsoft-Produkten verwendeten die SMB-Dienste ein Nicht-TCP/IP-Protokoll zur Implementierung der Namensauflösung. Seit Windows 2000 jedoch verwenden alle Microsoft-Produkte die DNS-Namensauflösung. Auf diese Weise können TCP/IP-Protokolle SMB-Ressourcenfreigaben direkt unterstützen (siehe Abbildung 3.19).

Die Betriebssysteme Linux und UNIX bieten ebenfalls eine Methode zur gemeinsamen Nutzung von Ressourcen mit Microsoft-Netzwerken an. Hier kommt eine SMB-Version namens SAMBA zum Einsatz. Auch die Apple Macintosh-Betriebssysteme unterstützen die Ressourcenfreigabe mithilfe des SMB-Protokolls.

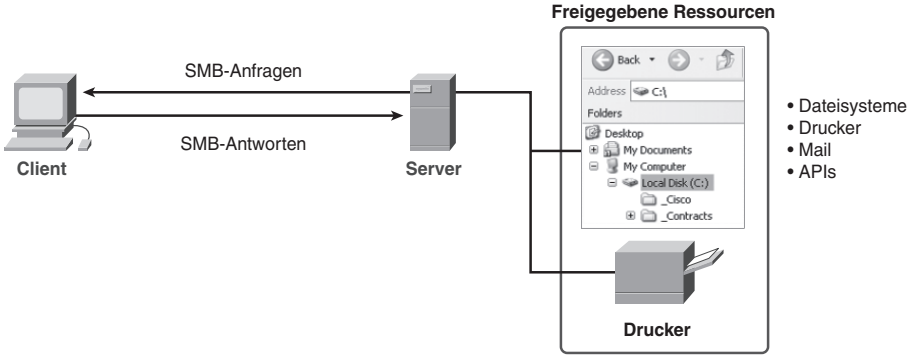


Abbildung 3.19: Dateifreigaben mit dem SMB-Protokoll

Das SMB-Protokoll beschreibt den Zugriff auf das Dateisystem und gibt an, wie Clients Dateien anfordern können. Ferner definiert SMB die prozessinterne Kommunikation. Alle SMB-Nachrichten weisen dasselbe Format auf. Dieses verwendet einen Header fester Größe, gefolgt von einer Parameter- und Datenkomponente variabler Größe.

SMB-Nachrichten führen die folgenden Aufgaben aus:

- Starten, Authentifizieren und Beenden von Sitzungen
- Steuern des Datei- und Druckerzugriffs
- Senden oder Empfangen von Nachrichten an bzw. von einem anderen Gerät durch die Anwendung

3.3.8 Peer-to-Peer-Dienste und das Gnutella-Protokoll

Sie haben FTP und SMB als Möglichkeiten zur Dateiübertragung bereits kennengelernt. In diesem Abschnitt beschreiben wir mit Gnutella ein weiteres Anwendungsprotokoll. Die Weitergabe von Dateien über das Internet – das Filesharing – ist extrem populär geworden. Mit Peer-to-Peer-Anwendungen, die auf dem Gnutella-Protokoll basieren, können Benutzer die Dateien auf ihren Festplatten anderen zum Download anbieten. Gnutella-kompatible Clientsoftware gestattet Benutzern das Herstellen einer Verbindung mit Gnutella-Diensten über das Internet und das Suchen von und Zugreifen auf Ressourcen, die von anderen Gnutella-Peers freigegeben wurden.

Es gibt viele Clientanwendungen, über die der Zugriff auf das Gnutella-Netzwerk möglich ist – BearShare, Gnucleus, LimeWire, Morpheus, WinMX und XoloX sind nur einige davon. Zwar pflegt das Gnutella Developer Forum das Basisprotokoll, doch entwickeln Drittanbieter häufig Erweiterungen, um das Protokoll besser an ihre Anwendungen anzupassen.

Viele Peer-to-Peer-Anwendungen verwenden keine zentrale Datenbank, in der alle Dateien aufgeführt sind, die den Peers zur Verfügung stehen. Stattdessen teilen die Geräte im Netzwerk auf Anfrage einander mit, welche Dateien vorhanden sind, und unterstützen das Auffinden von Ressourcen mithilfe des Gnutella-Protokolls und der zugehörigen Dienste (siehe Abbildung 3.20). Wenn ein Benutzer mit einem Gnutella-Dienst verbunden ist, suchen die Clientanwendungen nach anderen Gnutella-Knoten, mit denen sie Verbindungen herstellen können. Diese Knoten behandeln Anfragen nach dem Ort von Ressourcen und Antworten auf diese Anfragen. Zudem versenden sie Steuernachrichten, die den Dienst beim Entdecken weiterer Knoten unterstützen. Die eigentlichen Dateiübertragungen nutzen dann in der Regel HTTP-Dienste.

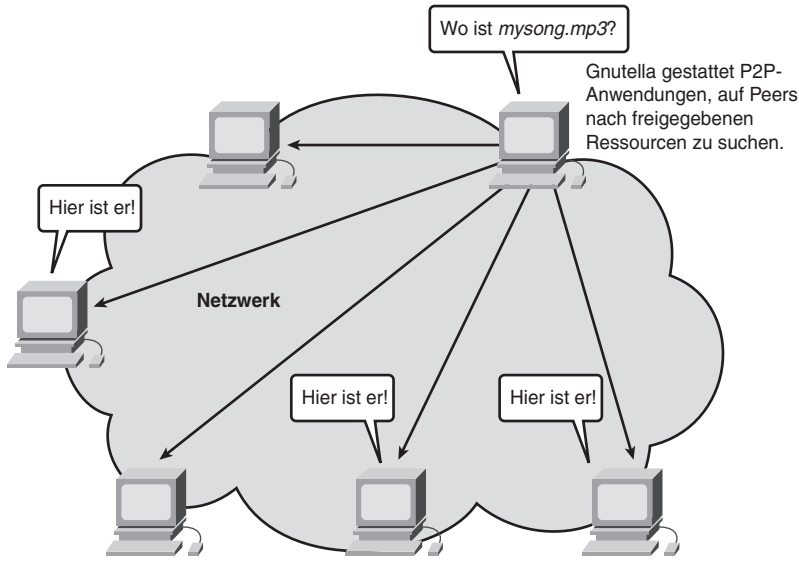


Abbildung 3.20: Das Gnutella-Protokoll

Das Gnutella-Protokoll definiert fünf verschiedene Pakettypen:

- ***ping.*** Dient der Entdeckung von Geräten.
- ***pong.*** Antwort auf *ping*
- ***query.*** Fragt den Dateispeicherort ab.
- ***query hit.*** Antwort auf *query*
- ***push.*** Antwort auf eine Download-Anforderung

3.3.9 Telnet-Dienste und -Protokoll

Lange bevor Desktopcomputer mit ausgefeilten grafischen Oberflächen existierten, verwendeten Benutzer textbasierte Systeme, die häufig nur an einen Zentralcomputer angeschlossene Anzeigeterminals waren. Als Netzwerke erfunden wurden, benötigten die Benutzer eine Möglichkeit des Fernzugriffs auf Computersysteme in einer Weise, wie ihn auch direkt angeschlossene Terminals boten.

Um diese Anforderung zu erfüllen, wurde Telnet erfunden. Das war in den frühen 1970er-Jahren – Telnet gehört damit zu den ältesten Anwendungsschichtprotokollen und -diensten in der TCP/IP-Familie. Telnet ist ein Client/Server-Protokoll, das eine Standardmethode der Emulation textbasierter Terminalgeräte über das Datennetz bereitstellt. Sowohl das Protokoll selbst als auch die Clientsoftware, die es implementiert, wird gemeinhin als Telnet bezeichnet. Der Telnet-Dienst wird in Abbildung 3.21 gezeigt.

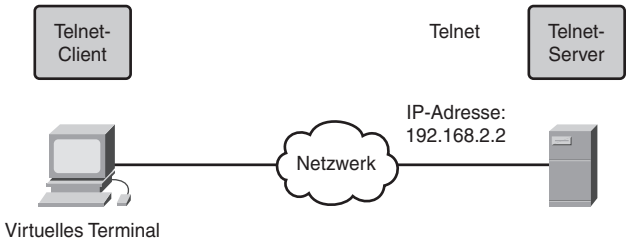


Abbildung 3.21: Telnet-Dienst

Für eine Telnet-Verbindung gibt es mehrere Bezeichnungen: *VTY-(*Virtual Terminal), *Sitzung* oder *Verbindung*. Telnet spezifiziert, wie eine VTY-Sitzung aufgebaut und beendet wird. Ferner beschreibt es die Syntax und Reihenfolge der Befehle, mit denen die Telnet-Sitzung gestartet wird, und stellt Steuerbefehle bereit, die während einer Sitzung abgesetzt werden können. Jeder Telnet-Befehl umfasst mindestens zwei Bytes. Das erste Byte ist ein Sonderzeichen namens IAC (Interpret as Command). Wie der Name bereits sagt, definiert das IAC-Zeichen das nächste Byte als Befehl (und nicht als Text). Statt eines physischen Geräts zur Verbindung mit dem Server verwendet Telnet eine Software zur Erstellung eines virtuellen Geräts, welches dieselben Eigenschaften bietet wie eine Terminal-Sitzung mit Direktzugriff auf die Befehlszeile des Servers.

Um Verbindungen über Telnet zu ermöglichen, führt der Server einen Dienst namens Telnet-Daemon aus. Von einem Endgerät mit der Telnet-Clientanwendung wird eine virtuelle Terminalverbindung zum Server hergestellt. Die meisten Betriebssysteme verfügen bereits über einen Telnet-Client für die Anwendungsschicht. Auf einem Microsoft Windows-PC kann Telnet über die Eingabeaufforderung ausgeführt werden. Andere gängige Terminalanwendungen, die als Telnet-Clients ausgeführt werden, sind HyperTerminal, Minicom und TeraTerm.

Nachdem eine Telnet-Verbindung hergestellt wurde, können Benutzer jede autorisierte Funktion auf dem Server ausführen – gerade so, als ob sie eine Befehlszeilensitzung direkt auf dem Server durchführten. Bei vorhandener Autorisierung können sie Prozesse starten und beenden, das Gerät konfigurieren und das System sogar herunterfahren.

Es folgen exemplarisch einige Telnet-Protokollbefehle:

- **AYT (Are You There).** Ermöglicht dem Benutzer die Anforderung einer Antwort (meist eines Eingabeaufforderungssymbols), das auf dem Terminalbildschirm erscheint, damit der Benutzer weiß, dass die VTY-Sitzung aktiv ist.
- **EL (Erase Line).** Löscht den gesamten Text der aktuellen Zeile.
- **IP (Interrupt Process).** Ermöglicht das Sperren, Unterbrechen, Abbrechen oder Beenden des Prozesses, mit dem das virtuelle Terminal verbunden ist. Wenn beispielsweise ein Benutzer über VTY ein Programm auf dem Telnet-Server gestartet hat, kann er einen IP-Befehl absetzen, um das Programm zu beenden.

Zwar unterstützt das Telnet-Protokoll die Benutzerauthentifizierung, nicht jedoch die verschlüsselte Übertragung von Daten. Alle Daten, die während einer Telnet-Sitzung übermittelt werden, werden als Klartext über das Netzwerk übertragen, können also leicht abgefangen und ausgelesen werden.

Das SSH-Protokoll (Secure Shell) bietet eine alternative und sichere Methode für den Serverzugriff. SSH stellt die Struktur für eine sichere Anmeldung und andere sichere Netzwerkdienste bereit. Ferner unterstützt SSH eine stärkere Authentifizierung als Telnet und zudem die Übertragung von Sitzungsdaten in verschlüsselter Form. Netzwerktechnikern wird empfohlen, statt Telnet stets SSH zu verwenden, wann immer dies möglich ist.

3.4 Zusammenfassung

Die Anwendungsschicht ist für den direkten Zugriff auf die zugrunde liegenden Prozesse zuständig, die die Kommunikation mit dem menschlichen Netzwerk verwalten und ermöglichen. Diese Schicht dient als Absender und Empfänger der Kommunikation über Datennetze. Die Anwendungen, Protokolle und Dienste der Anwendungsschicht ermöglichen Benutzern, interaktiv mit dem Datennetz auf eine sinnvolle und effiziente Weise zu kommunizieren.

Anwendungen sind Computerprogramme, mit denen der Benutzer interaktiv arbeitet und die den Datenübertragungsprozess auf Anforderung des Benutzers einleiten.

Dienste sind Hintergrundprogramme, die eine Verbindung zwischen der Anwendungsschicht und den untergeordneten Schichten des Netzwerkmodells herstellen.

Protokolle stellen eine Struktur anerkannter Regeln dar, ähnlich wie Grammatik und Interpunktionsregeln Richtlinien einer Sprache sind. Diese Protokollregeln stellen sicher, dass Dienste, die auf einem bestimmten Gerät ausgeführt werden, Daten an viele andere Netzwerkgeräte senden und von diesen empfangen können.

Die Auslieferung von Daten eines Servers über das Netzwerk kann von einem Client angefordert werden. In einem Peer-to-Peer-Kontext kann jedes Gerät als Client oder Server fungieren, und die Daten werden abhängig von der hergestellten Client/Server-Beziehung übermittelt. Nachrichten werden zwischen den Diensten der Anwendungsschicht auf jedem Endgerät entsprechend den Protokollspezifikationen ausgetauscht, welche die Herstellung und Verwendung solcher Beziehungen beschreiben.

Protokolle wie beispielsweise HTTP unterstützen die Zustellung von Webseiten an Endgeräte. SMTP und POP unterstützen das Senden und Empfangen von E-Mail. SMB ermöglicht Benutzern die Freigabe von Dateien. DNS löst einfach lesbare Namen, die zur Bezeichnung von Netzwerkressourcen verwendet werden, in numerische Adressen auf, die im Netzwerk verstanden werden. Telnet ermöglicht einen textbasierten Fernzugriff auf Geräte. DHCP weist Ressourcen IP-Adressen und andere Netzwerkparameter dynamisch zu. Peer-to-Peer schließlich ermöglicht zwei oder mehr Computern die Weitergabe von Ressourcen über das Netzwerk.

3.5 Aktivitäten und Übungen

Die Aktivitäten und Übungen im Begleitbuch »Network Fundamentals, CCNA Exploration Labs and Study Guide« (ISBN 1-58713-203-6) ermöglichen ein praxisbezogenes Üben der folgenden in diesem Kapitel vorgestellten Themen:

Übungen

Aktivität 3.1: Datenstrom aufzeichnen (3.4.1.1)

Bei dieser Aktivität verwenden Sie einen Computer, der entweder über ein Mikrofon und den Microsoft Audiorecorder oder aber über einen Internetzugang verfügt, um eine Audiodatei herunterzuladen.

Übung 3.1: Webserver administrieren(3.4.2.1)

In dieser Übung werden Sie den beliebten Apache-Webserver herunterladen, installieren und konfigurieren. Dann werden Sie mit einem Webbrowser eine Verbindung zum Server herstellen und den Kommunikationsvorgang mit Wireshark aufzeichnen. Die Analyse Ihrer Aufzeichnung wird Ihnen das Verständnis der Funktionsweise von HTTP erleichtern.

Übung 3.2: E-Mail-Dienste und -Protokolle (3.4.3.1)

In dieser Übung konfigurieren und verwenden Sie eine Mail-Clientanwendung, um eine Verbindung mit den Eagle-Server-Netzwerkdiensten herzustellen. Dann werden Sie die Kommunikation mit Wireshark aufzeichnen und die erfassten Pakete analysieren.

Packet Tracer Companion

Viele Praxisübungen enthalten Packet Tracer-Begleitaktivitäten, in denen Sie Packet Tracer zur Simulation der Übung verwenden können. Schlagen Sie dazu in »Network Fundamentals, CCNA Exploration Labs and Study Guide« (ISBN 1-58713-203-6) Praxisübungen mit Packet Tracer nach.

3.6 Lernzielkontrolle

Beantworten Sie die folgenden Fragen, um Ihren Kenntnisstand bezüglich der in diesem Kapitel beschriebenen Themen und Konzepte zu überprüfen. Die Antworten finden Sie in Anhang A, »Antworten zu Lernzielkontrollen und weiterführenden Fragen«.

1. Die Anwendungsschicht ist _____ des OSI-Modells.
   a) Schicht 1
   b) Schicht 3
   c) Schicht 4
   d) Schicht 7
2. Welche drei OSI-Schichten umfasst in etwa die TCP/IP-Anwendungsschicht?
   a) Anwendungsschicht, Sitzungsschicht, Transportschicht
   b) Anwendungsschicht, Darstellungsschicht, Sitzungsschicht
   c) Anwendungsschicht, Transportschicht, Vermittlungsschicht
   d) Anwendungsschicht, Vermittlungsschicht, Sicherungsschicht
3. Für welche der folgenden Aufgaben wird HTTP verwendet?
   a) Internetnamen in IP-Adressen auflösen
   b) Fernzugriff auf Server und Netzwerkgeräte vermitteln
   c) Dateien übertragen, die die Webseiten des World Wide Web bilden
   d) Mailnachrichten und Anhänge übertragen
4. Welchen Port verwendet das POP-Protokoll?
   a) TCP-/UDP-Port 53
   b) TCP-Port 80
   c) TCP-Port 25
   d) UDP-Port 110

5. Was ist GET?
   a) Eine clientseitige Datenanforderung
   b) Ein Protokoll, das Ressourcen oder Inhalte auf den Webserver lädt
   c) Ein Protokoll, das Daten in unverschlüsselter Form auf den Server hochlädt, die unter Umständen abgefangen und mitgelesen werden können
   d) Eine Serverantwort
6. Welches ist der beliebteste Netzwerkdienst?
   a) HTTP
   b) FTP
   c) Telnet
   d) E-Mail
7. FTP benötigt _____ Verbindung(en) zwischen Client und Server, um erfolgreich Daten übertragen zu können.
   a) 1
   b) 2
   c) 3
   d) 4
8. Was ermöglicht das Aktivieren von DHCP auf den Clients in einem Netzwerk?
   a) uneingeschränkte Telefonkommunikation
   b) Wiedergabe von Videostreams
   c) Erhalt von IP-Adressen
   d) Nachverfolgung sporadischer DoS-Angriffe
9. Die Betriebssysteme Linux und UNIX verwenden SAMBA. SAMBA ist eine Version welches Protokolls?
   a) SMB
   b) HTTP
   c) FTP
   d) SMTP

10. Welche der folgenden Sitzungen ist eine Telnet-Sitzung?
    a) FTP-Sitzung (File Transfer Protocol)
    b) TFTP-Sitzung (Trivial File Transfer Protocol)
    c) VTY-Sitzung (Virtual Terminal)
    d) AUX-Sitzung (Auxiliary)
11. Ist eBay eine Peer-to-Peer- oder eine Client/Server-Anwendung?
12. Im Client/Server-Modell bezeichnet man das Gerät, welches den Dienst anfordert, als _____.
13. HTTP wird als Anfrage-Antwort-Protokoll bezeichnet. Wie sehen drei typische Nachrichtenformate aus?
14. Was automatisiert DHCP?
15. Wofür steht das Akronym FTP, und wofür wird FTP verwendet?

3.7 Weiterführende Fragen und Aktivitäten

Die folgenden Fragen setzen ein tieferes Verständnis der in diesem Kapitel behandelten Konzepte voraus. Sie finden die Antworten in Anhang A.

1. Listen Sie den sechs Schritte umfassenden Prozess der Konvertierung menschlicher Kommunikationsvorgänge in Daten auf.
2. Beschreiben Sie die beiden Formen von Anwendungssoftware und deren jeweilige Aufgaben.
3. Arbeiten Sie die Bedeutung der Begriffe »Server« und »Client« im Kontext von Datennetzen aus.
4. Stellen Sie Client/Server- und Peer-to-Peer-Datenübertragung über Netzwerke einander vergleichend gegenüber.
5. Geben Sie fünf allgemeine Funktionen an, die Protokolle der Anwendungsschicht ausführen.
6. Nennen Sie den jeweiligen Zweck der Anwendungsschichtprotokolle DNS, HTTP, SMB und SMTP/POP.
7. Stellen Sie die Nachrichten, die Anwendungsschichtprotokolle wie DNS, HTTP, SMB und SMTP/POP zwischen Geräten austauschen, damit Datenübertragungen stattfinden können, einander vergleichend gegenüber.

3.8 Weitere Informationen

Die folgenden Fragen haben den Zweck, Sie zur Wiederholung der in diesem Kapitel behandelten Themen zu ermutigen. Ihr Schulungsleiter wird Sie möglicherweise auffordern, die Fragen zu beantworten und die Ergebnisse im Kurs zu erläutern.

1. Warum ist es wichtig, zwischen einer bestimmten Anweisung der Anwendungsschicht, dem zugehörigen Dienst und dem Protokoll zu unterscheiden? Diskutieren Sie die Frage im Kontext der Netzwerkreferenzmodelle.
2. Angenommen, es wäre möglich, alle Anwendungsschichtdienste in ein einziges allumfassendes Protokoll einzubinden? Behandeln Sie die Vor- und Nachteile eines solchen Protokolls.
3. Wie würden Sie ein neues Protokoll für einen neuen Anwendungsschichtdienst entwickeln? Was müsste Bestandteil dieses Protokolls sein? Wer müsste in den Prozess eingebunden werden, und wie würden die Informationen verbreitet werden?