

Entwicklung der Web-Programmierung

1.1 Der Weg zum World Wide Web

Die Anfänge „des Internets“ werden üblicherweise im Jahr 1969 angesiedelt und sind damit ziemlich genau 20 Jahre älter als die *Entstehung des World Wide Web*.

Theoretische Vorarbeiten für das erste Netz lieferten Paul Baran und Donald Watts Davies anfangs der sechziger Jahre des letzten Jahrhunderts mit ihrer Idee eines paketbasierten, selbstorganisierten Netzwerkes; Hintergrund hierfür scheint die Bestrebung nach ausfallsicheren Kommunikationsnetzen zum Höhepunkt des Kalten Krieges gewesen zu sein. 1969 wurden in den USA vier Universitäten durch ein Weitverkehrsnetz (Wide Area Network, WAN) verbunden. 1972 bildeten bereits 40 Universitäten das ARPANET, 1984 wurde in Deutschland der „Verein zur Förderung eines Deutschen Forschungsnetzes e. V.“ (DFN) gegründet ^{web}, der bis heute die deutschen Hochschulen und Forschungseinrichtungen mit einem Hochleistungsnetz versorgt (aktuell das X-WiN mit einer Anschlusskapazität bis zu 10 Gigabit/s im Backbone).

Das Internet beherbergt viele Dienste wie telnet oder FTP; der heute weitaus wichtigste aber ist „das Web“ oder auch „das WWW“. Seine zentralen Vorteile sind

- die Möglichkeit, durch Hyperlinks auf andere Dokumente zu verweisen;
- die Möglichkeit des Einsatzes multimedialer Komponenten, angefangen von frei zu formatierenden Texten über Grafiken bis hin zu Ton- und Bewegtbilddaten.

Die Idee für das WWW geht auf den 1955 in London geborenen englischen Physiker Sir Timothy J. Berners-Lee zurück, der 1989 an der europäischen Großforschungseinrichtung CERN den damaligen Informationsdienst Gopher ablösen wollte; die erste „Webadresse“ überhaupt war `http://info.cern.ch`. Sein Konzept basiert auf dem Zusammenspiel mehrerer Komponenten:

- ein Webserver;
- ein Webclient;
- das Vermittlungsprotokoll HTTP;
- eine Formatierungssprache HTML.



Abbildung 1.1:
Verein zur Förderung eines Deutschen Forschungsnetzes (DFN)

Im Detail führt er aus:

Abstract:

HyperText is a way to link and access information of various kinds as a web of nodes in which the user can browse at will. Potentially, HyperText provides a single user-interface to many large classes of stored information such as reports, notes, data-bases, computer documentation and on-line systems help. We propose the implementation of a simple scheme to incorporate several different servers of machine-stored information already available at CERN, including an analysis of the requirements for information access needs by experiments.

(Tim Berners-Lee, 12. November 1990)

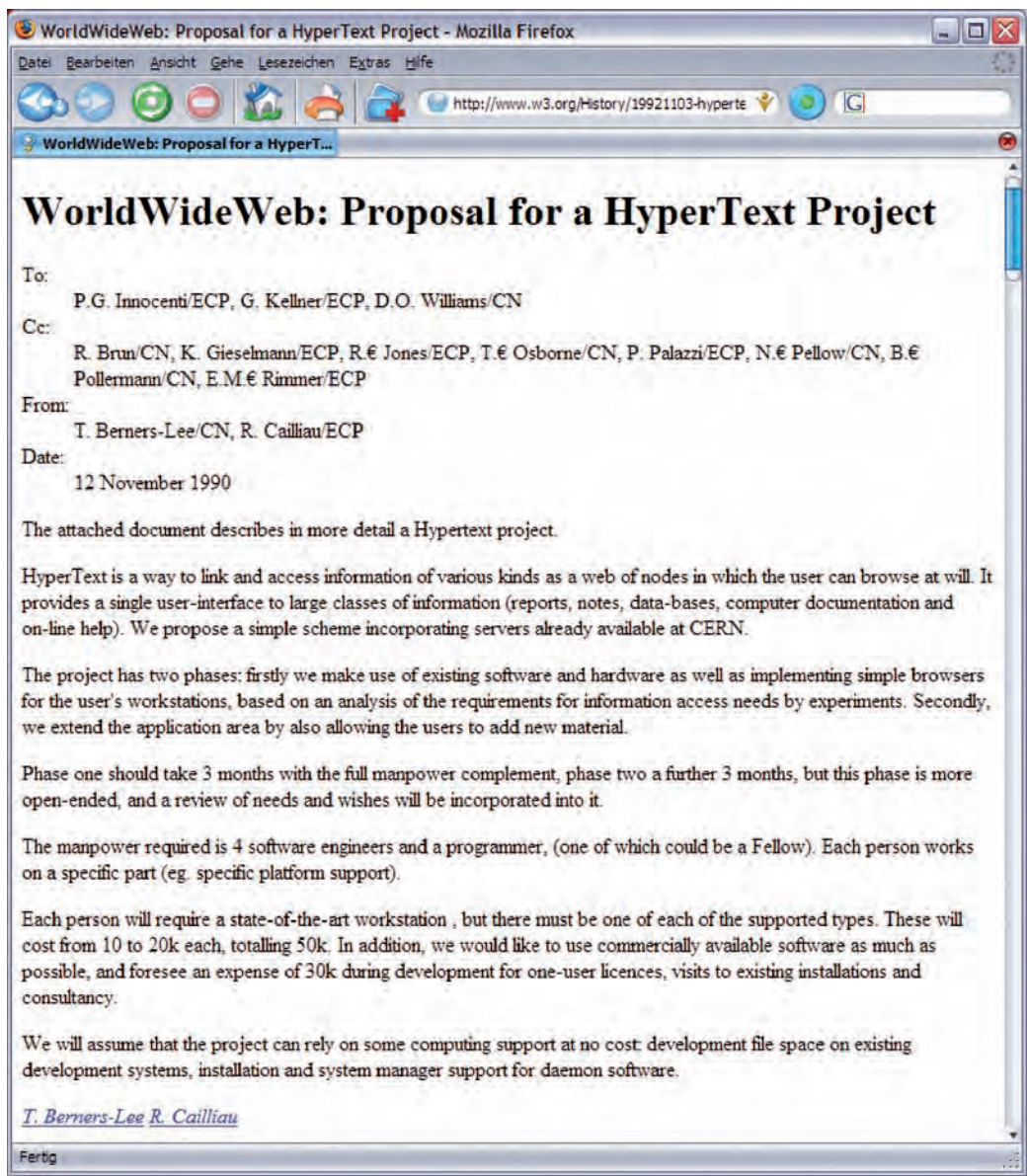


Abbildung 1.2:

Proposal von Berners-Lee, Seit dieser Initialpostulierung des WWW ist sein Wachstum ungebrochen; Ab-
November 1990 bildung 1.3 zeigt die Entwicklung seit August 1995.

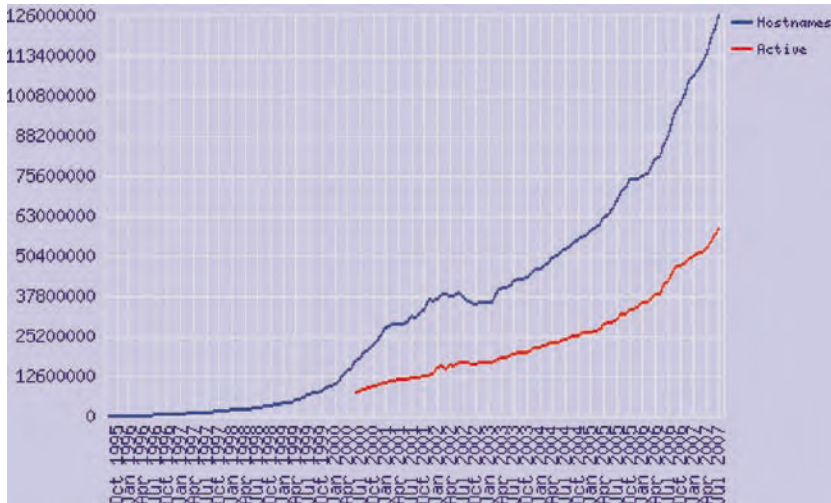


Abbildung 1.3:
Wachstum des WWW von August 1995 bis Juli 2007 (Quelle: Netcraft)

1.2 Komponenten der frühen Technik

Die ursprüngliche, auf Berners-Lee zurückgehende „WWW-Technik“ kommt mit wenigen fundamentalen Komponenten aus: ein Serverprogramm, welches angeforderte Seiten ausliefert, ein Clientprogramm, welches vom Server diese Informationen anfordert und die Rückantwort zur Anzeige bringt, eine Protokollfamilie, welche die Kommunikation zwischen Server und Client festlegt, sowie eine Auszeichnungssprache für die Codierung der anzuzeigenden Information.

Mit diesen Techniken kann zunächst nur statische Information zur Anzeige gebracht werden, etwa eine Ausgabe „Die Uhrzeit des Servers ist 12h 13min 14sek“ (mit echten, „dynamischen“ Daten) ist ursprünglich nicht möglich.

Diese grundlegenden Bestandteile der Kerntechnik sollen nun vorgestellt werden.

1.2.1 Der Webserver

Tim Berner-Lees zentraler Beitrag für das „erste WWW“ war die Bereitstellung eines *Webservers*:

Ein Webserver ist eine Serverapplikation, welche auf Anfragen im HTTP-Protokoll reagiert und die entsprechenden Antworten generiert.



Der erste Webserver war der CERN-Webserver, der über den NSCA-Webserver zum Apache weiterentwickelt wurde. Genaueres hierzu in Kapitel 6.

1.2.2 Der Webclient

Der *Webclient* hat zwei zentrale Aufgaben: Er stellt die korrekt formulierte Anfrage an den Webserver und bringt dessen Antwort (meist) grafisch formatiert zur Anzeige.

Der erste relevante Webclient war der Mosaic-Browser, welcher von Marc Andreessen entwickelt wurde; Andreessen hat damit die Firma Netscape gegründet, der Mosaic-Browser ging dann im Netscape-Browser auf. Mehr zu den aktuellen Browsern ist in Kapitel 1.2.2 zu finden.

1.2.3 Die Protokollfamilie

Für die Kommunikation zwischen Webclient und -server ist die Sprachebene festzulegen. Dies beginnt bei technischen Netzwerkbestandteilen bis hin zu den zulässigen Anweisungen wie GET für die Anforderungen eines Dokumentes und den möglichen Antwortarten. Die netzwerktechnische Ebene wird durch tcp/ip (siehe 1.6.2) festgelegt, die eigentliche Kommunikationsebene bestimmt das HTTP-Protokoll nach 1.6.5.

Hinzu kommt noch das Adressierungsschema der URL bzw. URI, welches in 1.6.9 dargestellt wird.

1.2.4 Die Auszeichnungssprache HTML

Neben den technischen Notwendigkeiten der Kommunikation zwischen Server und Client ist die Möglichkeit der Formatierung – etwa mit Grafiken und Hyperlinks – ein zentraler Grund für den unfassbaren Erfolg des World Wide Web. Diese Formatierung wird ursprünglich durch die Auszeichnungssprache HTML bereitgestellt, welche in 2.3 im Kern erläutert wird.

1.2.5 ...und die Dynamik?

Mit den bisher genannten Technikkomponenten lassen sich statische Inhalte anzeigen – aber keine dynamischen. Einfachstes Beispiel für eine Dynamik ist die Ausgabe der Uhrzeit auf dem Webserver, schon hier endet unser momentanes Wissen.

Um die zu erweitern, ist es notwendig, eine Programmierung bereitzustellen: Das ursprüngliche HTML muss um die Möglichkeit erweitert werden, dynamische Anteile zu implementieren. Hierfür wurde auch das umstrittene Schlagwort „Dynamisches HTML“ – DHTML – eingeführt, auf welches hier bewusst verzichtet wird.

Es gibt inzwischen zahllose Möglichkeiten, Dynamik im Web anzubieten. Die momentan wichtigsten werden in diesem Buch vorgestellt.

1.3 Clientseitige Web-Programmierung

Die Programmierung für das Internet kann sowohl client- als auch serverseitig erfolgen – und in vielen Fällen ist sogar die Verbindung beider Ansätze notwendig. Dabei verfolgen beide Ansätze unterschiedliche Ziele.



Abbildung 1.4:
Der Client in diesem Buch

Domäne der *clientseitigen Web-Programmierung* ist die direkte Interaktion mit dem Benutzer, die Integration von aktiven Programmen in den Browser. Hierzu zählt die Reaktion auf das Mausverhalten und die direkte Überprüfung von Eingabedaten vor Übertragung zu einem Server. Fortgeschrittenere grafische Effekte wie eine schematische Filmsequenz gehören dazu und inzwischen auch eine asynchrone Kommunikation mit Serverdiensten, um ein ähnliches Verhalten wie bei einer Desktop-Anwendung auch im Browser anbieten zu können.

Einige der Kern-Techniken in diesem Bereich sind:

- Darstellung mit (X)HTML und CSS;
- JavaScript als die grundlegende Client-Programmiersprache;
- Flash-Techniken etwa für Filmsequenzen mit ActionScript;
- Ajax für ein desktopähnliches Verhalten;

- Java Applets für vollständige Client-Applikationen, welche in einem Browser ablaufen.

Mit diesen Techniken ist auch eine direkte Kommunikation mit einem Datenbanksystem denkbar – insbesondere mit dem Applet –, wird aber aus Gründen der Sicherheit heute nicht mehr angewendet.

Eine typische Client-Programmierung ist der Farbwähler von Selfhtml unter <http://de.selfhtml.org/helferlein/farben.htm>: In einem Web-Formular sind Farbwerte auszuwählen, durch Druck auf den Button „Farben anzeigen“ werden die zugehörigen Hex-Farbwerte berechnet (unten links) und das rechte Fenster wird in den gewählten Farben neu gezeichnet. Alles geschieht direkt auf dem Client.



Abbildung 1.5:
Exemplarische clientbasierte
Web-Programmierung

1.4 Serverseitige Web-Programmierung

Wesentlich komplexere Programmierungen finden sich serverseitig, insbesondere, weil hier Datenbankanbindungen genutzt werden können. Typische Beispiele für solche serverseitigen Web-Programmierungen sind Google, Ebay und Amazon.

Hier ist das Spektrum der eingesetzten Techniken groß:

- CGI-Programmierungen mit ausführbaren Programmen wie kompiliertes C oder mit Scriptsprachen wie Perl, Python oder Ruby;
- PHP;
- Server Side Includes (SSI);
- fastCGI-Anwendungen;
- spezielle Module für den Webserver;
- Frameworks wie Ruby on Rails;
- Java-Entwicklungen auf dem J2EE-Framework wie Servlets und Java Server Pages;

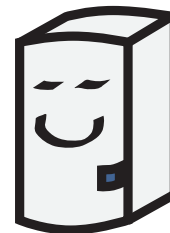


Abbildung 1.6:
Der Server in diesem Buch

- Content-Management-Systeme wie TYPO3 mit TypoScript.

1.5 Sprachen für die Web-Programmierung

In 1.3 und 1.4 wurden die gängigen Sprachen für den Einsatz in der Web-Programmierung genannt. Einige davon, etwa Perl und Java, sind Sprachen, welche auch unabhängig von der Web-Programmierung eingesetzt werden; andere wie PHP und ActionScript werden faktisch ausschließlich in diesem Bereich verwendet.

Viele dieser Sprachen sind Scriptsprachen, weshalb diese genauer betrachtet werden.

1.5.1 Scriptsprachen

Scriptsprachen sind einfache Programmiersprachen, die sich meistens durch folgende Eigenschaften auszeichnen:

- sie werden interpretiert und nicht kompiliert;
- Variablen in Scriptsprachen sind nicht typisiert: Eine Variable wird nicht deklariert und hat keinen festgelegten Typ wie „Zeichenkette“ oder „Gleitkommazahl“, sondern nimmt beliebige Werte auf und wandelt diese je nach Bedarf in einen anderen Typ um.

Die meisten der in diesem Buch behandelten Sprachen gehören zu der Gruppe der Scriptsprachen, etwa Perl, PHP, JavaScript und Ruby.

1.5.2 Java



Abbildung 1.7:
Java-Logo

Die populäre Sprache *Java* ist keine Scriptsprache, sondern eine moderne Hochsprache, welche typisch für die aktuelle Programmierung ist.

Java wurde um 1995 von James Gosling bei Sun Microsystems entwickelt und hat seither einen beispiellosen Erfolg.

Wesentlich für den Erfolg von Java ist das hier erstmals erfolgreich umgesetzte Konzept der plattformunabhängigen Programmierung: Java Sourcecode wird durch den Java-Compiler `javac` in eine plattformunabhängige `class`-Datei übersetzt; diese kann direkt auf verschiedenen Betriebssystemen durch die Java Virtual Machine (JVM) ausgeführt werden.

Die Plattformunabhängigkeit von Java hat die Verbreitung der Sprache wesentlich beschleunigt, da sie eine neue Technik in der Web-Programmierung ermöglicht: das Java Applet, ein im Web-Browser ausgeführtes Java-Programm. Dieser Weg hat sich allerdings nicht durchgesetzt, heute ist dafür serverseitiges Java wie Servlets und Java Server Pages eine wesentliche Technik.

Java ist heute in der Anwendung und genauso in der Lehre/Ausbildung eine vorherrschende Sprache. Aus diesem Grund wird in diesem Buch an vielen Stellen auf Java Bezug genommen.



Dieser spezielle Hinweis zeigt Parallelen und Unterschiede der vorgestellten Sprachen zu Java auf.

1.6 Technische Grundlage: die Internetprotokolle

In technischer Sicht ist „das Internet“ letztlich nichts anderes als die physikalische Infrastruktur mit einer ganzen Familie an *Protokollen*, die den Daten- und Kommunikationsaustausch über das physikalische Netz regeln. Die Gesamtzahl dieser Protokolle wird mit über 500 angegeben.

Welche Protokolle nun genau zu den Definitionen des Internets zählen, ist in letzter Konsequenz nicht zu sagen; die wichtigsten werden hier kurz vorgestellt. Eine umfassendere Darstellung der wichtigsten Protokolle ist in [MS04, S. 236 ff] enthalten.

1.6.1 Das Schichtmodell

Das „Netzwerk“ stellt letztlich eine verteilte Anwendung bereit, die zum Teil im Browser des Clientrechners und zum anderen Teil auf einem Webserver abläuft; dazwischen regelt das Netzwerk die Kommunikation zwischen den beteiligten Komponenten.

Um die Arbeitsweise des Netzwerks zu verstehen, ist es sinnvoll, das Netzwerk in verschiedene Hierarchieebenen mit klar definierten Schnittstellen zu zerlegen: das *Schichtmodell* (vgl. [MS04, S. 239 f]).

Das verbreitete ISO/OSI-Schichtmodell geht von sieben aufeinander aufbauenden Netzwerkschichten aus; die ersten vier davon zählen zum Transportsystem.

- **Schicht 1: Bitübertragungsschicht**
Diese unterste, auch als physikalisch bezeichnete Schicht stellt die technische Bitübertragung bereit.
- **Schicht 2: Sicherungsschicht**
Die Sicherungsschicht ist für eine fehlerfreie Übertragung verantwortlich und stellt die Verbindung zum Übertragungsmedium her.
- **Schicht 3: Vermittlungsschicht**
Diese Schicht regelt die Adressierung innerhalb des Netzwerks, also die Zuordnung von Netzwerkadressen zu physikalischen Endpunkten des Netzwerkes.
- **Schicht 4: Transportschicht**
Diese stellt einen sicheren und korrekten Datenaustausch sicher.

Die oberen drei Schichten bilden das Anwendungssystem; sie bestehen aus:

- **Schicht 5: Sitzungsschicht**
Diese stellt Dienste für einen synchronisierten Datenaustausch und für die Prozesskommunikation bereit, um den Abbruch einer Sitzung zu verhindern.
- **Schicht 6: Darstellungsschicht**
Hier werden systemabhängig codierte Daten in unabhängiger Form dargestellt.
- **Schicht 7: Anwendungsschicht**
Diese oberste Schicht stellt den darüber liegenden Anwendungen die benötigten Funktionalitäten wie E-Mail und Remote Login zur Verfügung.

Die Internet-Programmierung betrifft mindestens ab der dritten Schicht alle vorhandenen.

1.6.2 tcp/ip

Bei dem *tcp/ip-Protokoll* handelt es sich um die wesentliche Grundlage des Internets schlechthin, und es sind zwei aufeinander aufbauende Protokolle. Sie betreffen die dritte und vierte Schicht des ISO/OSI-Modells nach 1.6.1.

1.6.3 Internet Protocol (ip)

Das *Internet Protocol* ip regelt die Kommunikation zwischen zwei Rechnern ohne vorherigen Verbindungsaufbau auf der Vermittlungsschicht (Schicht drei im ISO/OSI-Modell). Dabei sind keine Mechanismen der Fehlererkennung und -korrektur implementiert, dies bleibt der höheren tcp-Schicht überlassen.

Zwei Versionen des ip sind momentan verbreitet, ip v4 und ip v6.

1.6.3.1 ip v4

Die *Version 4* des ip ist das verbreitetste Protokoll. Bekannt ist es insbesondere durch das Schema der ip-Adressen bestehend aus vier Gruppen zu je 8 bit für die Netzwerkadresse, etwa

`143.93.17.94`

für den Rechner mit dem Namen `ceres.informatik.fh-kl.de`. Die Beziehung zwischen Namen und Netzwerkadresse regelt der Domain Name Service nach 1.6.6.

Da für jede Zahl nur 8 bit zur Verfügung stehen, gibt es insgesamt

$$N = 24 \cdot 8 = 2^{32} = 4.294.967.296$$

Netzwerkadressen, wobei einige zusätzlich nur für besondere Zwecke bereitstehen. Dieser Adressraum erweist sich zunehmend als zu klein, weshalb eine neue Version des ip notwendig wird.

Lokale Netzwerke (LAN) bestehen aus einem beschränkten Adressraum. Typisch ist hier die Unterscheidung in Class-A-, Class-B- und Class-C-Netzwerke. Class-C-Netzwerke

- `123.123.123.abc`

und damit aus $2^8 = 256$ vielen Adressen. Per Konvention bekommt der lokale Übergangspunkt in andere Netze, der Router, eine Adresse am Ende des Zahlenbereichs, also im Beispiel `123.123.123.255`. Class-B-Netze schränken die Adressen nur nach dem Schema

`123.123.abc.def`

ein und bestehen somit aus 65.536 Einzeladressen, Class-A-Netzwerke sind nochmals nach dem Schema

`123.abc.def.ghi`

größer und kommen somit auf 16.777.216 einzelne Netzwerkadressen.

1.6.3.2 Sonderadressen in ip v4

Einigen ip-Adressen kommt eine besondere Bedeutung zu, die auch in der Web-Programmierung von großer Bedeutung sind.

Zunächst ist hier der Bezug zum eigenen Netzwerkinterface genannt, die Adresse `127.0.0.1`, die auch als `loopback` oder `localhost` bezeichnet wird. Wenn der Webserver auf dem lokalen Rechner zu testen ist, dann genügt die URL

`http://127.0.0.1`

zum Testen. Das gesamte Class-C-Netzwerk

127.0.0.abc

ist dem Bezug zum lokalen Rechnersystem vorbehalten und wird nicht über das Netzwerk geleitet.

Für Multicast-Dienste gibt es einen eigenen Adressraum von 224.0.0.0 bis 239.255.255.255.

Wichtig für den Aufbau sicherer Netzwerke sind nichtroutbare Netzwerkadressen. Dieser Adressbereich ist in RFC 1918 definiert; Tabelle 1.1 gibt eine Übersicht über den privaten Netzwerkadressbereich. Diese Adressen können in dieser Form nur lokal verwendet werden.

1.6.3.3 Network Address Translation (NAT)

Um mit nichtroutbaren internen Adressen nach außen arbeiten zu können – etwa um „zu surfen“ – bedient man sich der Adressumsetzung *Network Address Translation* (NAT); hier wird jede ausgehende Netzwerkverbindung auf eine offizielle, routbare Adresse umgesetzt. Der Vorteil davon ist, dass somit keinerlei Informationen über die tatsächlichen internen Adressen nach außen gelangt. Üblicherweise werden alle internen Adressen dabei auf die gleiche externe Adresse umgeleitet.

Vergleichbar zu NAT ist Port Address Translation (PAT); hierbei werden die Netzwerk-Ports umgesetzt.

Adressbereich	Anzahl Adressen
10.0.0.0 – 10.255.255.255	16.777.214
172.16.0.0 – 172.31.255.255	1.048.544
192.168.0.0 – 192.168.255.255	65.536
169.254.0.0 – 169.254.255.255	65.536

Tabelle 1.1:
Private Netzwerkadressen

1.6.3.4 ip v6

Die neuere *Version 6* des ip beinhaltet wesentliche Fortschritte gegenüber der Version 4. Hierzu zählen insbesondere:

- Der Adressraum in IPv6 ist deutlich erhöht bei Beibehaltung der Kompatibilität zu den IPv4-Adressen.
- IPv6 beinhaltet die Sicherungsmaßnahmen, die als IPSec bezeichnet werden. Damit wird eine netzwerkseitig integrierte Verschlüsselung des Datentransfers ermöglicht.

1.6.3.5 Transmission Control Protocol (tcp)

Das mit dem Internet Protocol 1982 eingeführte Transmission Control Protocol (tcp) ist ein verbindungsorientiertes Protokoll auf der ISO/OSI-Schicht vier (Transportschicht). Es setzt auf dem ip auf und sichert die wichtige Korrektheit der Datenübertragung.

Konkret bedeutet das tcp, dass die einzelnen Datenpakete, aus denen ein Datenstrom besteht, überprüft werden auf

- die Vollständigkeit;
- die richtige Reihenfolge des Empfangs.

Somit ist eine auf tcp/ip basierende Netzwerkverbindung die ideale geprüfte Verbindung für die Daten einer Web-Site. Wir werden vorwiegend bei der Web-Programmierung deshalb auf tcp/ip aufbauen, benötigen aber mitunter auch andere Protokolle, die hier noch kurz vorgestellt werden sollen.

1.6.4 User Datagram Protocol (udp)

Das *User Datagram Protocol* (udp) ist die Alternative zu tcp, wenn die Schnelligkeit der Übertragung wichtiger ist als die Korrektheit. Dies ist etwa im Bereich von Streaming-Diensten der Fall.

udp ist ein verbindungsloses Protokoll. Es verzichtet auf die Kontrolle der Korrektheit der Datenlieferung, die wesentlicher Bestandteil von tcp ist.

1.6.5 Hypertext Transfer Protocol (HTTP)

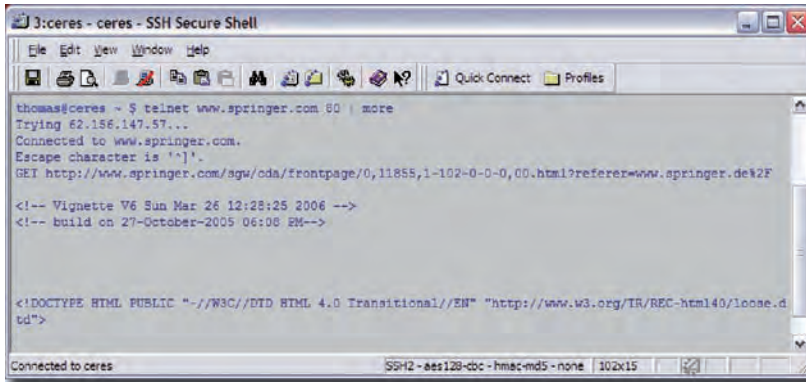
Das *Hypertext Transfer Protocol* (HTTP) nimmt für die Thematik dieses Buches eine ganz entscheidende Rolle ein, da es regelt, wie zwischen Webclient und Webserver Informationen ausgetauscht werden.

HTTP ist Kern der „Erfindung des Webs“ durch Tim Berners-Lee (vgl. 1.1). 1989 wurde Version 0.9 von HTTP vorgestellt, ihr folgten 1992 die Versionen 1.0 und die seit 1997 aufgrund von RFC 2068 verwendete Version 1.1.

HTTP ist ein sehr einfaches Protokoll. Die drei Versionen lassen sich folgendermaßen abgrenzen:

- HTTP Version 0.9 (Abbildung 1.8)
 - vollständig textbasiert (keine Multimedia-Daten)
 - der Server kann nur angefordertes Dokument senden, keine zusätzlichen Informationen
 - der Client kann keine Daten an den Server übertragen
 - HTTP 0.9 ist nicht mehr aktuell
- HTTP Version 1.0 (Abbildung 1.9)
 - bietet neben der Methode GET auch die Methode POST
 - implementiert HEADER-Informationen mit Meta-Daten
 - stellt Internet Medientypen (MIME) zur Verfügung
 - unterstützt mit der Information last-modified Caching-Mechanismen
 - bietet Authentifizierungsmöglichkeiten.
- HTTP Version 1.1 (Abbildung 1.10)
 - Entitäts-Tag: eindeutige Kennung für jedes Dokument (erleichtert Caching auf mehreren Servern)
 - feste Verbindungen (Zeitbegrenzung durch keep-alive), dadurch verbessertes Netzwerkmanagement
 - verbesserte Authentifizierung (DIGEST)
 - Multihoming: ein Server verwaltet mehrere WWW-Domänen mit unterschiedlichem Dokumentenstamm

Die Bedeutung der wichtigsten HTTP-Sprachbestandteile ist in Tabelle 1.2 aufgeführt.



```

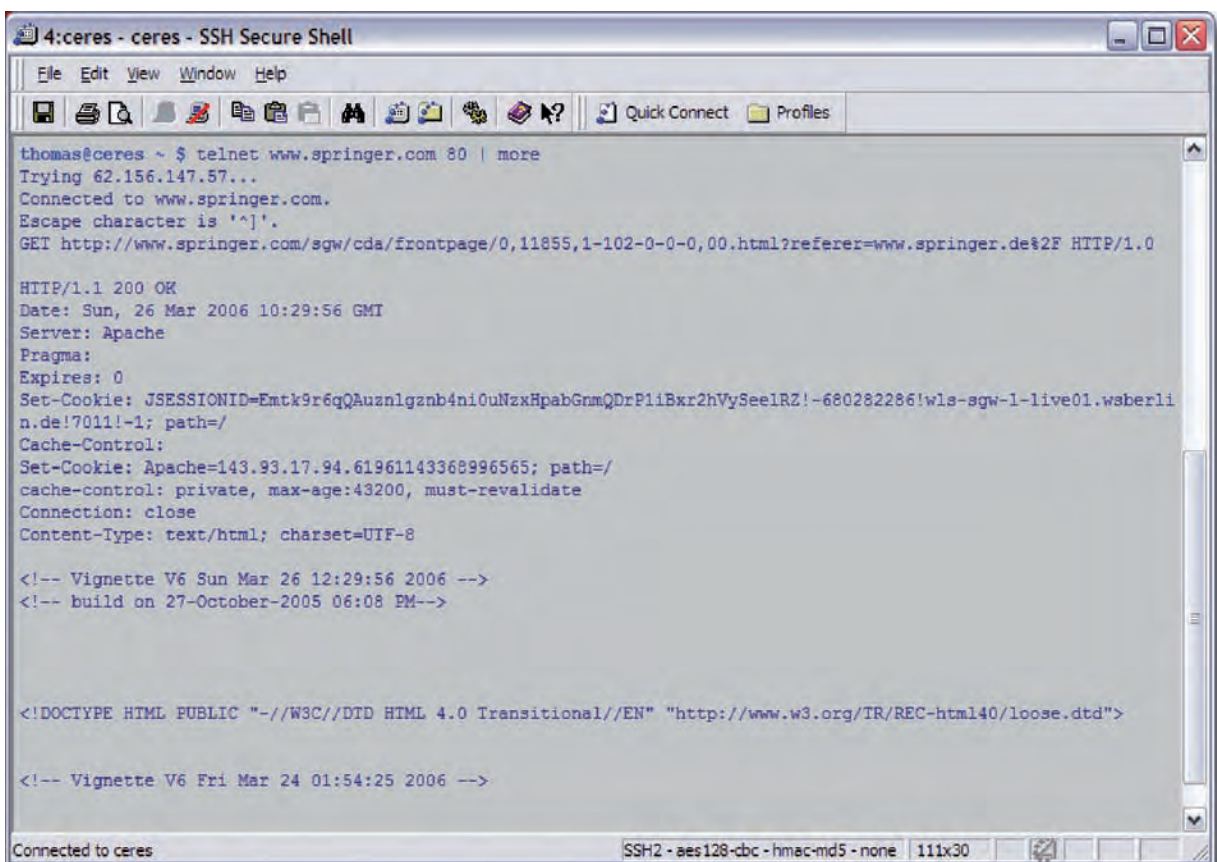
3:ceres - ceres - SSH Secure Shell
File Edit View Window Help
Quick Connect Profiles
thomas@ceres ~ $ telnet www.springer.com 80 | more
Trying 62.156.147.57...
Connected to www.springer.com.
Escape character is '^]'.
GET http://www.springer.com/sgw/cda/frontpage/0,11855,1-102-0-0-0,00.html?referer=www.springer.de%2F

<!-- Vignette V6 Sun Mar 26 12:28:25 2006 -->
<!-- build on 27-October-2005 06:08 EM-->

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN" "http://www.w3.org/TR/REC-html40/loose.d
td">
Connected to ceres          SSH2 - aes128-cbc - hmac-md5 - none 102x15

```

Abbildung 1.8:
GET-Request zur
Springer-Homepage in HTTP 0.9



```

4:ceres - ceres - SSH Secure Shell
File Edit View Window Help
Quick Connect Profiles
thomas@ceres ~ $ telnet www.springer.com 80 | more
Trying 62.156.147.57...
Connected to www.springer.com.
Escape character is '^]'.
GET http://www.springer.com/sgw/cda/frontpage/0,11855,1-102-0-0-0,00.html?referer=www.springer.de%2F HTTP/1.0

HTTP/1.1 200 OK
Date: Sun, 26 Mar 2006 10:29:56 GMT
Server: Apache
Pragma:
Expires: 0
Set-Cookie: JSESSIONID=Emtk9r6qQAuznlgznb4ni0uNzxHpabGnmQDrPliBxr2hVySee1RZ!-680282286!wls-sgw-1-live01.wsberli
n.de!7011!-1; path=/
Cache-Control:
Set-Cookie: Apache=143.93.17.94.61961143368996565; path=/
cache-control: private, max-age=43200, must-revalidate
Connection: close
Content-Type: text/html; charset=UTF-8

<!-- Vignette V6 Sun Mar 26 12:29:56 2006 -->
<!-- build on 27-October-2005 06:08 EM-->

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN" "http://www.w3.org/TR/REC-html40/loose.dtd">

<!-- Vignette V6 Fri Mar 24 01:54:25 2006 -->
Connected to ceres          SSH2 - aes128-cbc - hmac-md5 - none 111x30

```

Abbildung 1.9:
GET-Request zur
Springer-Homepage in HTTP 1.0