Um das manchmal komplexe Verhalten von Informatiksystemen zu verstehen, ist eine Vorstellung von deren technischen Grundlagen erforderlich. Die Ursache der Komplexität liegt darin, dass wir wie in vielen anderen Gebieten der Informatik auch hier zahlreiche *zustandsabhängige Systeme* finden, die auf jeweils gleiche Einflüsse aus der Umgebung unterschiedlich reagieren können und die sich gegenseitig beeinflussen. Ihre Reaktionen hängen meist von der Vorgeschichte ab, also von vorhergehenden Einflüssen. Beispiele dafür finden wir reichlich:

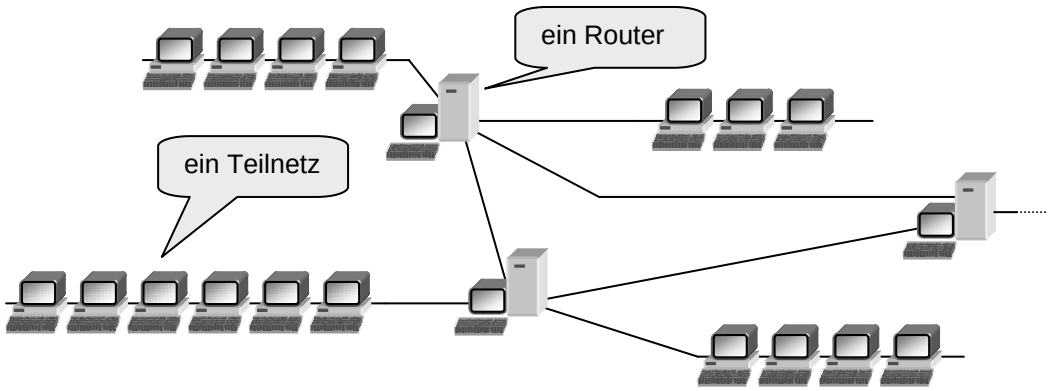
- Programme und Programmsammlungen wie eine Textverarbeitung oder ein Betriebssystem können in verschiedenen Situationen auf gleiche Eingaben verschieden reagieren, aber auch deren Bestandteile wie Objekte, Unterprogramme, … können sich so benehmen.
- Computer verhalten sich ähnlich, und das gilt auch für einfachere Schaltungen wie Ampeln, elektronische Steuergeräte, Taschenrechner, Roboter, …
- Ganze Netzwerke von Computern und/oder anderen Geräten zeigen diese Eigenschaft, wie man – manchmal leidvoll – erfahren muss.

Wir wollen in diesem Buch ein Modell für zustandsabhängige Systeme entwickeln und dieses auf unterschiedliche Gebiete anwenden. Dabei gehen wir vom größten vorhandenen System – dem Internet – aus und arbeiten uns dann zu den Details einzelner Computerkomponenten vor. Die dabei gewonnenen Erfahrungen ermöglichen es später, den Bereich der Computersprachen und der Berechenbarkeit besser zu verstehen und auch dort praktisch zu arbeiten.

1 Technische Aspekte des Internets

1.1 Aufbau, Adressierung und Protokolle

Verbindet man verschiedene Computer miteinander durch ggf. technisch unterschiedliche Datenübertragungsstrecken (Glasfaser, Modem, Funk, …), dann entsteht ein *Computer-Netzwerk*. Verbindet man wiederum die verschiedenen Netzwerke untereinander, indem einzelne Computer jeweils mit mindestens zwei Netzen verbunden werden, dann können über diese Rechner – die *Router* – Daten von einem Netz in das andere transportiert werden. Mithilfe solcher Verknüpfungen entsteht aus den Teilnetzen das Netz der Netze – das *Internet*.



Für das Funktionieren des Internets genügt es nicht, nur die Leitungen zu legen. Es müssen vor allem Verfahren entwickelt werden,

- die einzelnen Rechner des Netzes über die vielen Verbindungsknoten (die Router) hinweg zu *adressieren* – also finden zu können,
- die *Protokolle* (Übermittlungsverfahren) zu vereinbaren, mit deren Hilfe die Daten zwischen den einzelnen Rechnern – die technisch sehr unterschiedlich sein können – ausgetauscht werden,
- die *Wege* festzulegen, die die Daten im Netz nehmen sollen – denn meist gibt es dafür verschiedene Möglichkeiten,
- und schließlich *Dienste* (Email, WWW, FTP, …) und *Werkzeuge* (Browser, …) zu entwickeln, die das Internet für die unterschiedlichen Zwecke möglichst unabhängig vom eingesetzten Computer und dessen Betriebssystem nutzbar machen.

Die Entwickler des Internets hatten das Ziel, dieses Netzwerk dezentral so einzurichten, dass es auch beim Ausfall von Teilen des Netzes immer noch funktioniert. Anfangs gab es dafür militärische Gründe, denn die Ursprünge des Internets liegen in Netzwerken in US-amerikanischen Universitäten und/oder militärischen Einrichtungen. Heute verhindert schon die schiere Größe des Netzes eine rein hierarchische Organisation, und auch die Anfälligkeit gegenüber absichtlichen oder technisch bedingten Störungen eines Gebildes, das vom Funktionieren einiger weniger Computer abhängt, spricht gegen eine rein baumartige Struktur.

Sollen Daten in einem Netz verteilt werden, das seine Struktur z. B. durch Ausfall oder Überlastung von Teilen oder durch Ergänzungen ständig ändert, dann dürfen die Datenwege nicht vorab festgelegt werden, sondern die Daten müssen sich ihren Weg vom Absender zum Adressaten „selbst suchen“. Dazu müssen Start- und Zielrechner eindeutig identifizierbar sein, und die Router müssen entscheiden können, welche der freien Wege zwischen ihnen und den nächsten Routern die Daten „in die richtige Richtung“ befördern.

Beginnen wir mit der Adressierung:

Adressen werden derzeit noch im Internet als *IP*-(Internet-Protocol)-*Adressen* der Version *IPv4* vergeben, die aus vier Teilen bestehen, die jeweils einer ganzen Zahl zwischen 0 und 255 entsprechen. Da jede dieser Zahlen eine 8-Bit-Dualzahl darstellt (s. Anhang A1), bestehen IP-Adressen aus insgesamt 32 Bits. (Die Einteilung in vier Zahlen dient vor allem der Lesbarkeit.) Da es Netze sehr unterschiedlicher Größe gibt, teilt man diese (derzeit noch) in fünf Klassen ein, die über unterschiedlich viele Rechner verfügen können. (Die neue Adressierungsart *IPv6* ermöglicht sehr viel mehr Adressen als bisher.)

Eine IP-Adresse besteht aus drei Teilen:

- Das erste Bit bzw. die ersten Bits geben die Netzklasse an (A bis E).
- Je nach Netzklasse folgt eine unterschiedlich lange Netzadresse (*net-id*). Je länger diese ist, desto mehr Netze dieser Klasse gibt es.
- Danach kommt die Rechneradresse (*host-id*), die jeweils einen bestimmten Rechner im Teilnetz identifiziert. Auch hier gilt: Je länger die Rechneradresse, desto mehr Rechner können sich im Teilnetz befinden. (Dabei kann ein Rechner durchaus mehrere host-ids besitzen, z. B. wenn er als Router zu verschiedenen Teilnetzen gehört.)

Damit haben wir einen „Interessenkonflikt“ zwischen Netzadresse und Rechneradresse: Weil die IP-Adresse auf 32 Bits beschränkt ist, muss die Rechneradresse kürzer werden, wenn die Netzadresse wächst – und umgekehrt. Je mehr Netze zu einer Klasse gehören, desto weniger Rechner passen jeweils in die Teilnetze dieser Klasse.

Bit	0	1	2	3	4	5	6	7	8 ….. 15	16	17 …… 23	24 ……..…… 31
Klasse A	**0**	Netzadresse (7 Bit)							Rechneradresse (24 Bit)			
Klasse B	**1**	**0**	Netzadresse (14 Bit)							Rechneradresse (16 Bit)		
Klasse C	**1**	**1**	**0**	Netzadresse (21 Bit)								Rechneradresse (8 Bit)
Klasse D	**1**	**1**	**1**	**0**	Gruppenadresse (multicast)							
Klasse E	**1**	**1**	**1**	**1**	**0**	reserviert für zukünftige Zwecke						

Man kann leicht nachrechnen, dass es bei diesen Daten prinzipiell

- $128 = 2^7$ unterschiedliche Netze der Klasse A gibt, die jeweils aus bis zu 2^{24}, also 16777216 Computern bestehen können,
- $16384 = 2^{14}$ unterschiedliche Netze der Klasse B gibt, die jeweils aus bis zu 2^{16}, also 65536 Computern bestehen können,
- $2097152 = 2^{21}$ unterschiedliche Netze der Klasse C gibt, die jeweils aus bis zu 2^8, also 256 Computern bestehen können.

Die Netzklassen D und E wollen wir hier nicht betrachten.

Aus diesen 32-Bit-Adressen erhält man die „gepunktete“ Schreibweise aus vier Zahlen, indem man die IP-Adresse in vier Teile zu je acht Bits aufteilt. Diese rechnet man dann in Dezimalzahlen um. Für eine Adresse in einem Klasse-A-Netz erhält man z. B.:

Netzklasse	net-id	host-id
0	0000011	00000000000000000000000001

→ 00000011 00000000 00000000 00000001 (vier 8-Bit-Gruppen)

→ 3 0 0 1

→ 3.0.0.1 (IP-Adresse)

Rechnet man für die Netze der Klasse A die IP-Netzadressenbereiche aus, dann erhält man IP-Adressen zwischen 1.0.0.0 und 126.0.0.0. (Das Netz 127.0.0.0 wird für andere Zwecke benutzt.)

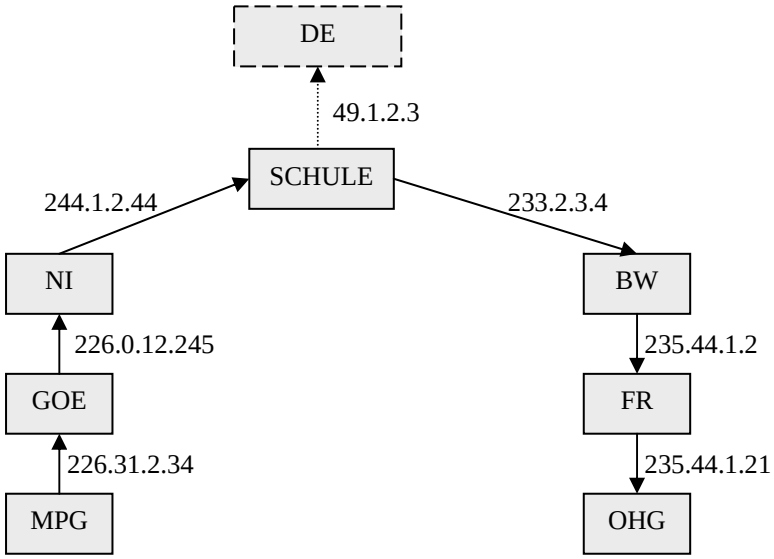
Einige Adressbereiche dienen besonderen Zwecken:

- Eine Adresse, die nur aus Einsen besteht (111…1), adressiert alle Rechner im Netz.
- Eine Adresse, die nur aus Nullen besteht (000…0), adressiert den eigenen Rechner (genauer: die eigene host-id).
- Die Adresse 127.0.0.1 wird als *Loopback*-Adresse benutzt. Der Rechner sendet die so adressierten Daten nicht ins Netz, sondern behandelt sie so, als ob sie aus dem Netz gekommen wären. Sie dient also vor allem zu Testzwecken.
- Einige Adressen sind für spezielle Zwecke reserviert. Dazu gehören z. B. die Adressen 192.168.0.0 bis 192.168.255.255, die innerhalb verschiedener lokaler Netze gleichzeitig benutzt werden können. Solche *Intranets* sind über je ein *Gateway* (einen Rechner, der sowohl mit dem Internet wie dem lokalen Intranet verbunden ist) vom Internet abgeschottet.

Da für Menschen Zahlen nur schlecht zu merken und zu unterscheiden sind, hat man eine zweite Art der Adressierung eingeführt, die von den IP-Adressen weitgehend unabhängig ist: das *Domain Name System DNS*. Hier werden in einer Baumstruktur gestaffelte Namensbereiche benutzt, die durch Punkte von einander getrennt und „von rechts nach links“ zu lesen sind. Ausgegangen wird von so genannten *top-level-domains*, die für bestimmte Zwecke vorgesehen sind, z. B.

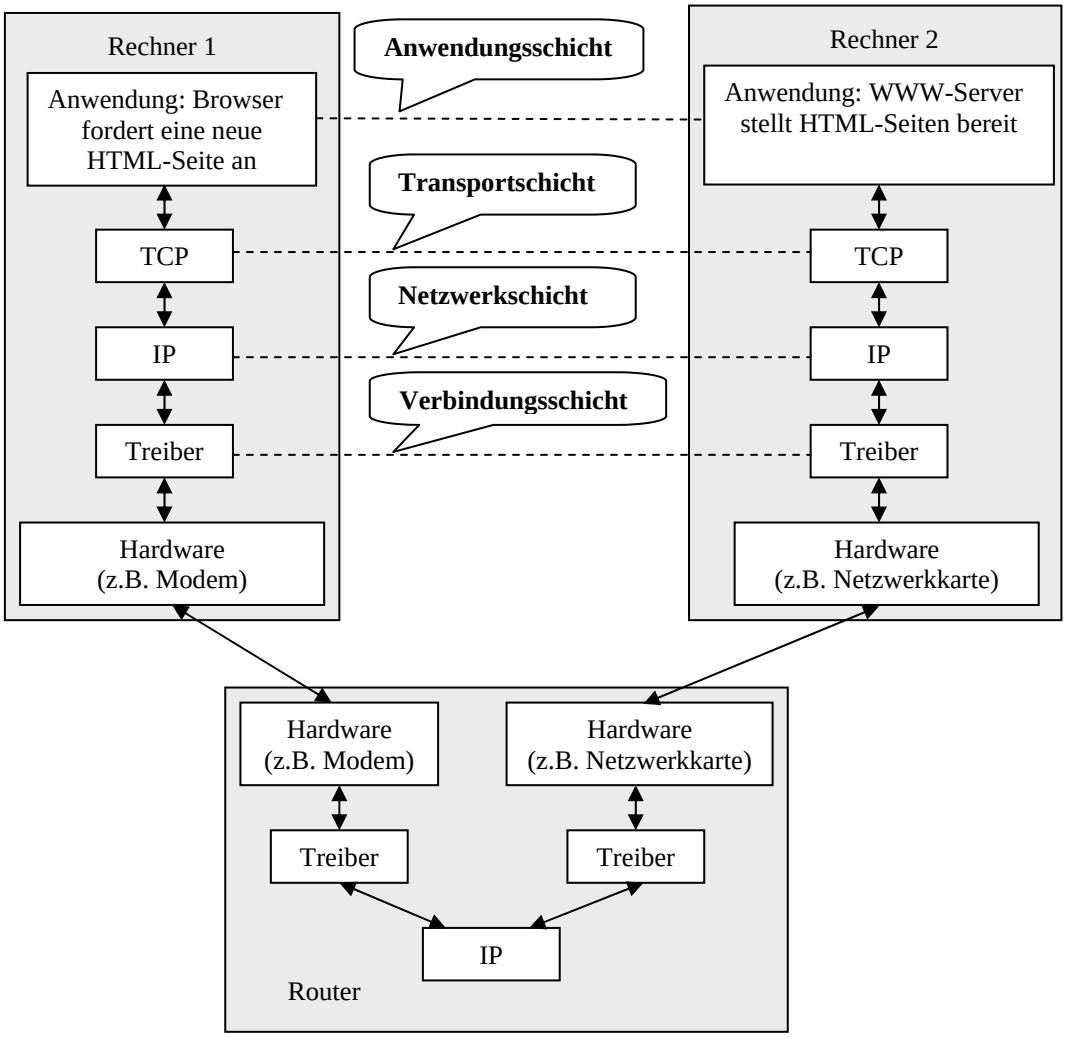
- edu — für den Erziehungsbereich (Schulen und Universitäten),
- com, mil, org — für kommerzielle oder militärische Zwecke bzw. für größere Organisationen,
- de, uk, fr, it, … — als Länderkennungen.

Für die einzelnen Domains existieren jeweils Organisationen, die Namen für Unterdomänen vergeben können; in Deutschland z. B. die DENIC, bei der man sich über die Besitzer von de-Domänen informieren kann: http://www.denic.de/servlet/Whois. Diese Organisationen betreiben auch jeweils mehrere Name-Server, die zwischen den Bereichsnamen und den zugehörigen IP-Adressen vermitteln. Sendet also z. B. eine Schule in Göttingen/Niedersachsen mit der Adresse mpg.goe.ni.schule.de eine Nachricht an eine Schule in Freiburg/Baden-Würtemberg mit der Adresse ohg.fr.bw.schule.de, dann müssen mehrere Nameserver eingeschaltet werden, um diesen Adressen gültige IP-Werte zuzuordnen. Jeder Domain Rechner verfügt über eine Tabelle, in der sich mindestens die IP-Adresse des Computers befindet, der die übergeordnete Domäne verwaltet (die unten angegebenen IP-Adressen sind fiktiv). Der Nameserver der DE-Domäne wird nicht angesprochen, da schon die SCHULE-Domäne die Nachricht in das richtige Teilnetz weiterleiten kann.



Der Transport von Daten zwischen den Rechnern erfolgt mithilfe eines *Schichtenmodells*, in dem die Daten von Schicht zu Schicht „nach unten“ weiter gereicht werden. Auf der „niedrigsten“ Ebene befindet sich die Netzwerk-Hardware, die Bitfolgen zwischen den Rechnern überträgt. Haben die Daten ihren Bestimmungsort erreicht, werden sie wieder zusammengesetzt und einem Anwendungsprogramm zur Verfügung gestellt. Bei diesem Verfahren übernimmt jede Schicht spezifische Aufgaben in Form eines *Protokolls*, das von der „darüber“ liegenden Schicht benutzt werden kann, ohne dass diese weiß, wie genau die darunter liegende Schicht ihre Aufgaben löst. Damit können unterschiedliche Hard- und Softwaresysteme für den gleichen Zweck eingesetzt werden. Z. B. kann die physikalische Bit-Ebene der Netzwerke auf sehr unterschiedlichen Techniken beruhen: auf Glasfaser-, Funk- oder sonstiger Technologie.

Das Internet benutzt ein 4-Schichten Modell:



Im oben dargestellten Fall fordert ein Browser eine HTML-Seite von einem anderen Rechner an. Er benutzt dafür in der *Anwendungsschicht* das *HTTP-Protokoll* (Hypertext Transport Protocol), das seinen Auftrag an die *Transportschicht* weitergibt, die das *TCP* (Transmission Control Protocol) bereitstellt. Diese Schicht zerlegt die zu übermittelnden Daten in Pakete (*Datagramme*), die nummeriert und an die *Netzwerkschicht* mit ihrem *IP* weitergereicht werden. Dabei werden die Datenpakete mit den Adressen von Absender und Adressaten und weiteren Informationen, z. B. einer Prüfsumme versehen, anhand der

der Empfänger überprüfen kann, ob die Daten unversehrt angekommen sind. Die Paketnummerierung erfolgt fortlaufend, beginnend mit einer Zufallszahl. So können zwar im Netz gleichzeitig verschiedene Pakete mit der gleichen Nummer unterwegs sein. Dass diese aber auch beim Absender und Adressaten übereinstimmen, ist extrem unwahrscheinlich. Die Datagramme werden an die Treibersoftware der *Verbindungsschicht* weitergereicht, die spezifisch für die benutzte Hardware (Modem, ISDN-Karte, Ethernetkarte, …) ist und mit deren Hilfe sie die Daten auf ihren Weg schickt. Treffen die Datenpakete auf ihrem Weg auf einen Router, dann „schaufelt" dieser sie von einem Teilnetz in das andere – mithilfe von IP. Beim Zielrechner angekommen, werden die Datenpakete, die in zufälliger Reihenfolge auf unterschiedlichen Wegen gereist sein können, zwischengespeichert und von der TCP-Schicht anhand ihrer Nummer richtig zusammengesetzt. Danach werden sie an die Anwendungsschicht dieses Rechners durchgereicht – usw. Die Aufteilung der erforderlichen Funktionen auf unterschiedliche Ebenen ermöglicht es, dass andere Protokolle der Anwendungsschicht, z. B. *FTP*, ebenfalls TCP/IP benutzen, ohne die erforderlichen Details neu implementieren zu müssen.

Damit heutige Rechner, auf denen gleichzeitig mehrere Programme laufen, auf mehreren Kanälen gleichzeitig auf das Internet zugreifen können, werden zusätzlich zur IP-Adresse so genannte *Ports* benutzt, die bestimmten Programmen zugeordnet werden. Portnummern sind ganze Zahlen und liegen zwischen 0 und 65535. Das HTTP-Protokoll benutzt z. B. (meist) den Port 80. Auf diese Weise können bunt gemischt Datenpakete für unterschiedliche Anwendungen den gleichen Rechner auf der gleichen Leitung erreichen. Dort werden sie dann anhand der Portnummer einer Anwendung zugeordnet.