Einführung in
TCP/IP

Heiko Holtkamp
(heiko@rvs.uni-bielefeld.de)

AG Rechnernetze und Verteilte Systeme
Technische Fakultät, Universität Bielefeld

18. Juni 1997
letzte Änderung 14. Februar 2002

Inhaltsverzeichnis

1 Vorwort

„Despite its popularity and widespread use, the details of TCP/IP protocols and the structure of software that implements them remain a mystery to most computer professionals.“

(Comer D.E., Stevens D.L.: Internetworking with TCP/IP, Vol. II)

Dieses Dokument ist die Ausarbeitung des Seminarvortrages *Introduction to TCP/IP*, den ich am 22. Mai und 5. Juni 1997 im Rahmen des Seminars *UNIX System Administration* gehalten habe. Es dient auch als Grundlage für eine Einführung in Netzwerkprotokolle in den Vorlesungen *Rechnernetze, UNIX Systemadministration* und *Internet: Werkzeuge und Dienste*, die von der AG Rechnernetze und Verteilte Systeme an der Universität Bielefeld gehalten werden.

Mittlerweile hat dieses Dokument einige Versionen hinter sich und wird weiterhin in unregelmäßigen Abständen überarbeitet. Es gibt immer wieder Punkte, die besser beschrieben werden können, sich im Laufe der Zeit geändert haben oder ergänzt werden müssen. Insbesondere sind der Grundlagenteil und der Abschnitt über IPv6 immer noch nicht fertig gestellt. Ich möchte mich an dieser Stelle bei all denjenigen bedanken, die mir eine Rückmeldung und Änderungsvorschläge für diese Arbeit gegeben haben. Ich hoffe, die Vorschläge in ihrem Sinne aufgenommen zu haben.

Über Anregungen, Lob, Verbesserungsvorschläge und Kritik zu dieser Arbeit würde ich mich freuen. Das Dokument ist nicht als statisch gedacht, d. h. einmal geschrieben und damit vergessen, sondern soll, wie gesagt, weiterhin ergänzt und verbessert werden.

Mit der letzten Überarbeitung des Dokumentes habe ich mich entschieden, die Arbeit als PDF-Dokument zu veröffentlichen. Da mir leider die Zeit fehlt, die PDF-Version und die HTML-Version nebeneinander auf einem aktuellen Stand zu halten, werden Berichtigungen, Änderungen und Neuerungen vorerst nur noch in der PDF-Version durchgeführt. Eventuell ergibt sich in Zukunft ein Verfahren, mit dem beide Versionen nebeneinander erzeugt werden können.

Diese Arbeit kann jeder für *private Zwecke und für die Lehre* herunter laden und verwenden. Ich möchte aber darum bitten, falls jemand diese Arbeit auf seinen eigenen Seiten verwenden möchte, nicht einfach die Dateien zu kopieren, sondern einen Verweis auf die Seiten (bzw. auf die die Seite, http://www.rvs.uni-bielefeld.de/~heiko/tcpip) zu setzen. Wie schon gesagt, wird diese Arbeit ständig überarbeitet und es ist ärgerlich, wenn jemand eine veraltete Kopie im Netz zur Verfügung stellt. Ebenso möchte ich darum bitten, dass die Namen der ursprünglichen Autoren der Arbeit weiterhin genannt und nicht einfach aus dem Dokument gelöscht werden (ist leider auch schon vorgekommen).

Bielefeld, den 14. Februar 2002

Heiko Holtkamp

2 Grundlagen

2.1 Was ist ein Rechnernetz?

Diese Frage habe ich mir auch längere Zeit gestellt. Die Literatur äußert sich hierzu leider nicht so ganz klar. Tanenbaum „definiert“ ein Rechnernetz in [Tanenbaum1996] wie folgt:

„Das ganze Buch hindurch wird der Begriff Rechnernetze für mehrere miteinander verbundene autonome Computer verwendet. Zwei Computer gelten als miteinander verbunden, wenn sie Informationen austauschen können. Die Verbindung muß nicht aus einem Kupferkabel bestehen - es können auch Lichtwellenleiter, Mikrowellen oder Kommunikationssatelliten benutzt werden. Die Vorgabe, daß die Computer autonom sein müssen schließt Systeme, bei denen ein eindeutiges Master/Slave-Verhältnis herrscht, von vornherein aus unserer Definition aus. Kann ein Computer einen anderen beliebig ein- oder ausschalten oder steuern, besteht keine Unabhängigkeit. Ein System mit einer Steuereinheit als Master und vielen Slaves ist kein Netz, ebensowenig wie ein Großrechner mit entfernten Druckern und Terminals.“

Nach dieser Definition ist ein Netz, das aus einer Anzahl von Java-Rechnern (oder sind es doch nur Terminals) besteht kein Rechnernetz. Die Java-Rechner müssen ihr System erst aus dem Netz laden, bevor mit ihnen „autonom“ gearbeitet werden kann. Die Frage ist hier zusätzlich: arbeiten Java-Rechner wirklich autonom?

Deshalb möchte ich noch eine allgemeinere „Definition“ eines Rechnernetzes aus [Comer1998] geben:

„Das alte Modell, bei dem ein Großrechner den gesamten Rechenaufwand eines Unternehmens bewältigte, wurde durch eines ersetzt, bei dem eine große Anzahl einzelner miteinander verbundener Rechner die Arbeit übernimmt. Ein solches System nennt man Rechnernetz.“

2.2 Protokolle, Protokollhierarchien

Protokolle sind Regeln, die den Nachrichtenaustausch - oder allgemeiner das Verhalten - zwischen (Kommunikations)Partnern regeln („Protocols are formal rules of behaviour“). Die Verletzung eines vereinbarten Protokolls erschwert die Kommunikation oder macht sie sogar gänzlich unmöglich.

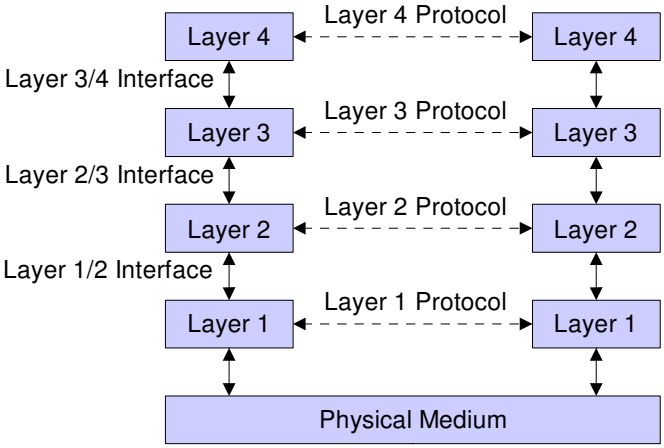


Abbildung 1 Anordnung von Protokollen zu einem Protokollstapel.

Ein Beispiel für ein Protokoll „aus dem täglichen Leben“ ist z.B. der Funkverkehr: Die Kommunikationspartner bestätigen den Empfang einer Nachricht mit Roger und leiten einen Wechsel der Sprechrichtung mit Over ein. Beendet wird die Verbindung schließlich mit Over and out.

Ähnliche Protokolle werden auch beim Datenaustausch zwischen verschiedenen Computern benötigt, auch wenn hier die Komplexität der Anforderungen etwas höher ist. Aufgrund dieser höheren Komplexität werden viele Aufgabe nicht von einem einzigen Protokoll abgewickelt. In der Regel kommen eine ganze Reihe von Protokollen, mit verschiedenen Teilaufgaben, zum Einsatz. Diese Protokolle sind dann in Form von Protokollschichten mit jeweils unterschiedlichen Funktionen angeordnet.

2.3 Eine kurze Geschichte des Internet

From small things, big things
sometimes come
(Tittel E., Robbins M.)

Gegen Ende der sechziger Jahre, als der „kalte Krieg“ seinen Höhepunkt erlangte, wurde vom US-Verteidigungsministerium (Department of Defence - DoD) eine Netzwerktechnologie gefordert, die in einem hohen Maß gegenüber Ausfällen sicher ist. Das Netz sollte dazu in der Lage sein, auch im Falle eines Atomkrieges weiter zu operieren. Eine Datenübermittlung über Telefonleitungen war zu diesem Zweck nicht geeignet, da diese gegenüber Ausfällen zu verletzlich waren (sind). Aus diesem Grund beauftragte das US-Verteidigungsministerium die *Advanced Research Projects Agency (ARPA)* mit der Entwicklung einer zuverlässigen Netztechnologie. Die ARPA wurde 1957 als Reaktion auf den Start des Sputniks durch die UdSSR gegründet. Die ARPA hatte die Aufgabe Technologien zu entwickeln, die für das Militär von Nutzen sind. Zwischenzeitlich wurde die ARPA in *Defense Advanced Research Projects Agency (DARPA)* umbenannt, da ihre Interessen primär militärischen Zwecken dienten. Die ARPA war keine Organisation, die Wissenschaftler und Forscher beschäftigte, sondern verteilte Aufträge an Universitäten und Forschungsinstitute.

Um die geforderte Zuverlässigkeit des Netzes zu erreichen, fiel die Wahl darauf, das Netz als ein *paketvermitteltes Netz (packet-switched network)* zu gestalten. Bei der Paketvermittlung werden zwei Partner während der Kommunikation nur virtuell miteinander verbunden. Die zu übertragenden Daten werden vom Absender in Stücke variabler oder fester Länge zerlegt und über die virtuelle Verbindung übertragen; vom Empfänger werden diese Stücke nach dem Eintreffen wieder zusammengesetzt. Im Gegensatz dazu werden bei der *Leitungsvermittlung (circuit switching)* für die Dauer der Datenübertragung die Kommunikationspartner fest miteinander verbunden.

Ende 1969 wurde von der *University of California Los Angeles (UCLA)*, der *University of California Santa Barbara (UCSB)*, dem *Stanford Research Institute (SRI)* und der *University of Utah* ein experimentelles Netz, das *ARPANET*, mit vier Knoten in Betrieb genommen. Diese vier Universitäten wurden von der (D)ARPA gewählt, da sie bereits eine große Anzahl von ARPA-Verträgen hatten. Das ARPA-Netz wuchs rasant (siehe Abbildung) und überspannte bald ein großes Gebiet der Vereinigten Staaten.

Mit der Zeit und dem Wachstum des ARPANET wurde klar, dass die bis dahin gewählten

Protokolle nicht mehr für den Betrieb eines größeren Netzes, das auch mehrere (Teil)Netze miteinander verband, geeignet war. Aus diesem Grund wurden schließlich weitere Forschungsarbeiten initiiert, die 1974 zur Entwicklung der *TCP/IP-Protokolle* bzw. des *TCP/IP-Modells* führten. TCP/IP wurde mit mit der Zielsetzung entwickelt, mehrere verschiedenartige Netze zur Datenübertragung miteinander zu verbinden. Um die Einbindung der TCP/IP-Protokolle in das ARPANET zu forcieren beauftragte die (D)ARPA die Firma *Bolt, Beranek & Newman (BBN)* und die *University of California at Berkeley* zur Integration von TCP/IP in Berkeley UNIX. Dies bildete auch den Grundstein des Erfolges von TCP/IP in der UNIX-Welt.

Im Jahr 1983 wurde das ARPANET schließlich von der *Defence Communications Agency (DCA)*, welche die Verwaltung des ARPANET von der (D)ARPA übernahm, aufgeteilt. Der militärische Teil des ARPANET, wurde in ein separates Teilnetz, das *MILNET*, abgetrennt, das durch streng kontrollierte Gateways vom Rest des ARPANET - dem Forschungsteil - separiert wurde.

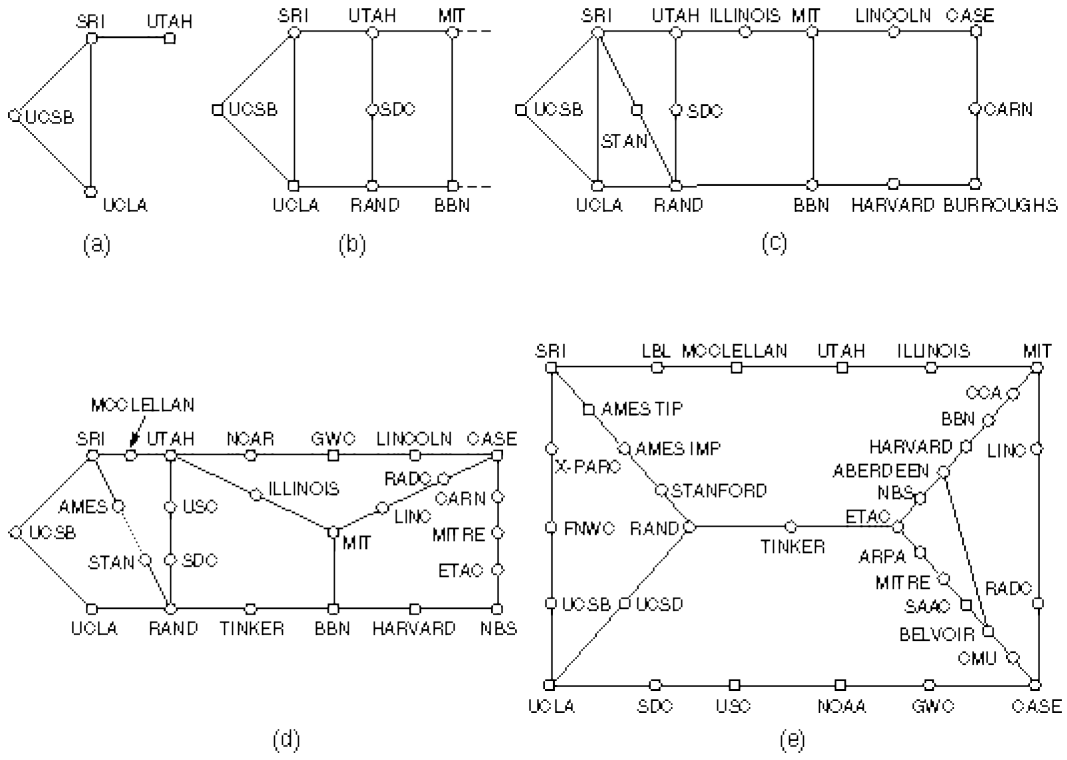


Abbildung 2 Wachstum des ARPANET a)Dezember 1969 b)Juli 1970 c)März 1971 d) April 1971 e)September 1972. (Quelle: [Tanenbaum1996])

Nachdem TCP/IP das einzige offizielle Protokoll des ARPANET wurde, nahm die Zahl der angeschlossenen Netze und Hosts rapide zu. Das ARPANET wurde von Entwicklungen, die es selber hervorgebracht hatte, überrannt. Das ARPANET in seiner ursprünglichen Form existiert heute nicht mehr, das MILNET ist aber noch in Betrieb.

> **Hinweis:** Das Denic führt Statistiken zum Wachstum des Internet in Deutschland. Die Statistiken sind unter der Adresse http://www.denic.de/de/domains/statistiken/index.html verfügbar

Die Sammlung von Netzen, die das ARPANET darstellte, wurde zunehmend als *Netzverbund*

betrachtet. Dieser Netzverbund wird heute allgemein als *das Internet* bezeichnet. Der Leim, der das Internet zusammenhält, sind die TCP/IP-Protokolle.

Hinweis: Weitere Informationen zur Geschichte des Internet sind unter folgenden Adressen zu finden:

- Internet Society - ISOC: History of the Internet (http://www.isoc.org/internet/history/)
- Musch J.: Die Geschichte des Netzes: ein historischer Abriss (http://www.psychologie.uni-bonn.de/sozial/staff/musch/history.htm)
- Hauben M.: Behind the Net: The Untold History of the ARPANET and Computer Science (http://www.columbia.edu/~rh120/ch106.x07)
- Hauben R.: The Birth and Development of the ARPANET (http://www.columbia.edu/~rh120/ch106.x08)
- w3history: Die Geschichte des World Wide Web (http://www.w3history.org/)
- Media History Project (http://www.mediahistory.com/)

2.4 Internet-Standards und Dokumentation

Eine wichtige Rolle bei der Entstehung und Entwicklung des Internet spielen die so genannten *RFCs - Request for Comments*. RFCs sind Dokumente, in denen die Standards für TCP/IP bzw. das Internet veröffentlicht werden. Einige RFCs beschreiben Dienste und Protokolle sowie deren Implementierung, andere fassen Regeln und Grundsätze (*policies*) zusammen. Standards für TCP/IP werden immer als RFCs veröffentlicht, aber nicht alle RFCs beschreiben Standards. Einige RFCs sind auch durchaus amüsant zu lesen (darauf weist schon zum Teil das Datum der Veröffentlichung , 1. April, hin):

- RFC527 – ARPAWOCKY
- RFC968 - 'Twas the Night Before Start-up
- RFC1118 – The Hitchhiker's Guide to the Internet
- RFC1149 – A Standard for the Transmission of IP Datagrams on Avian Carriers
- RFC1180 - A TCP/IP Tutorial
- RFC2324 – Hyper Text Coffe Pot Control Protocol (HTCPCP/1.0)
- RFC2795 – The Infinite Monkey Protocol Suite (IMPS)
- u.a.

Die Standards für TCP/IP werden im wesentlichen nicht durch ein Komitee entwickelt, sondern durch Diskussion und Konsens beschlossen. Jeder hat die Möglichkeit ein Dokument als RFC zu veröffentlichen und so zur Diskussion zu stellen. Ist ein Dokument veröffentlicht, wird ihm eine RFC-Nummer zugewiesen. RFCs werden fortlaufend nummeriert; zur Zeit gibt es etwa 3000. Neben der eindeutigen RFC-Nummer haben RFCs einen beschreibenden Titel.

Das erste RFC wurde am 7. April 1969 von Steve Crocker versendet (RFC 1, „Host-Software“). Steve Crocker und andere arbeiteten zusammen an dem Problem des Host-zu-Host-Dialogs in einem Netzwerk. Sie beschlossen, die Diskussionen, die sie führten,

schriftlich zu protokollieren, um später auf diese Aufzeichnungen zurückgreifen zu können. Crocker arbeitete an der Zusammenfassung der Diskussionen und bezeichnete das Dokument an dem er arbeitete, um ihm den Anschein von „Amtlichkeit“ zu nehmen, als „Request for Comment“ (Ersuchen um Stellungnahme). [HafnerLyon2000]

Die Dokumente werden von einer Arbeitsgruppe und/oder dem RFC-Editor geprüft. Dabei durchläuft das Dokument verschiedene Stufen, die Stufen der Entwicklung, Testung und Akzeptanz. Die Stufen bilden den so genannten *Standards Process*. Die Stufen werden formal als *maturity levels (Reifestufen)* bezeichnet. Zusätzlich zu seiner Stufe bekommt ein RFC einen Status.

Jon Postel (6. August 1943 – 16. Oktober 1998) gehört zu den Pionieren des Internet. Zusammen mit mit anderen (wie Bob Taylor, Vinton Cerf, Frank Heart, Larry Roberts, Leonard Kleinrock, Bob Kahn, Wesley Clark, Douglas Engelhart, Barry Wessler, Dave Walden, Severo Ornstein, Truett Thach, Roger Scantlebury, Charlie Herzfeld, Ben Barker, Steve Crocker, Bill Naylor, Roland Bryan [HafnerLyon2000]) arbeitete er an den Grundlagen des ARPANET und des Internet. Postel war 30 Jahre lang Direktor der Internet Assigned Numbers Authority (IANA). Daneben war er knapp 30 Jahre lang „der RFC-Editor“. Etwa 2.500 RFCs gingen durch seine Hände und er prägte wesentlich das System und den Erfolg der RFCs mit. In RFC1122 formulierte Postel einen zentralen Satz für das Design von Protokollen im Internet: *„Be liberal in what you accept, and conservative in what you send.“*

Ein RFC, das einmal veröffentlicht ist,wird nie verändert oder aktualisiert. Es kann nur durch ein neues RFC ersetzt werden. Bei einer Ersetzung wird das alte RFC mit der Bezeichnung „*Obsoleted by RFC xxx“* gekennzeichnet, das neue RFC beinhaltet einen Hinweis „*Obsolets RFC xxx“* auf das alte RFC. Korrekturen an einem RFC werden durch „*Updates RFC xxx“* und „*Updated by RFC xxx“* gekennzeichnet. Einige RFCs beschreiben Protokolle, die durch bessere ersetzt wurden, diese RFCs werden durch die Bezeichnung „*historic“* gekennzeichnet. Ein entsprechender Standard erhält den Status „*not recommended“*. RFCs, die sich in einer experimentellen Phase der Entwicklung befinden werden mit der Bezeichnung „*experimental“* versehen. Protokolle, die von anderen Organisationen oder von Firmen entwickelt wurden und von Interesse für das Internet sind werden zum Teil auch in RFCs veröffentlicht mit den Bezeichnungen „*informational“* oder „*best current practice“*.

Neben der RFC-Nummer können RFCs auch *STD-Nummern (Standard)*, *FYI-Nummern (For Your Interest)* oder *BCP-Nummern (Best Current Practice)* erhalten. STDs, FYIs und BCPs sind untergeordnete Kategorien von RFCs, die Dokumente mit einem bestimmten oder besonderen Inhlt kennzeichnen. Die RFCs, STDs, FYIs und BCPs sind wiederum in ihrer Kategorie eindeutig, fortlaufen nummeriert; ein Dokument kann also mehrere Nummern haben.

Der Prozess, wie RFCs veröffentlicht werden, ist ebenfalls in einem RFC, dem *Internet Official Protocol Standard (OPS)*, dokumentiert. Neben einer Erläuterung, wie RFCs veröffentlich und verfasst werden, enthält es auch Verweise auf die derzeit aktuellsten RFCs der Protokollstandards. Das OPS-RFC wird regelmäßig, jeweils nach 100 neuen RFCs, aktualisiert; RFC2600, RFC2700, RFC2800 usw. Der Internet-Standardisierungsprozess ist in RFC2026 beschrieben. Im RFC2555 (30 Years of RFCs) wird der technische und soziale Aspekt der RFCs erläutert.

Maturity Level	Bedeutung
Proposed Standard (PS)	Diese Stufe dauert mindestens 6 Monate und erfordert zwei unabhängige Implementierungen.
Draft Standard (DS)	Diese Stufe dauert mindestens 4 Monate mit Demonstrationen und einem Erfahrungsbericht mit mindestens zwei unabhängigen Implementierungen.
Standard (S oder STD)	Das RFC ist zum offiziellen Standard erhoben. Internet-Standards erhalten neben der RFC-Nummer eine so genannte *STD-Nummer* (z.B. Internet Protocol, RFC791, STD-5).

Status	Bedeutung
Required	Muss bei allen TCP/IP-basierten Hosts und Gateways implementiert werden.
Recommended	Es wird empfohlen, daß alle TCP/IP-basierten Hosts und Gateways die Spezifikationen des RFCs implementieren. Diese RFCs werden üblicherweise auch immer implementiert.
Elective	Die Implementierung ist optional. Der Anwendung wurde zugestimmt, ist aber nicht erforderlich.
Limited Use	Nicht für die generelle Nutzung gedacht.
Not recommended	Nicht zur Implementierung empfohlen.

Das „System“ der RFCs leistet einen wesentlichen Beitrag zum Erfolg von TCP/IP und dem Internet. RFCs werden von zahlreichen Quellen im Internet zur Verfügung gestellt. In der Referenzliste findet sich eine Angabe zu Quellen, bei denen die RFCs bezogen werden können.

2.5 Referenzmodelle

2.5.1 Das OSI-Referenzmodell

Das *Open Systems Interconnection (OSI)-Referenzmodell* ist ein Modell, dass auf einem Vorschlag der *International Standards Organisation (ISO)* basiert. Der Aufbau des OSI-Modells ist in Abbildung 3 dargestellt.

Das Modell dient derzeit allgemein als Rahmen zur Beschreibung von Protokollcharakteristika und -funktionen [Tanenbaum1996]. Das OSI-Modell (die offizielle Bezeichnung lautet *ISO-OSI-Referenzmodell*) besteht aus sieben Schichten. Die Schichtung beruht auf dem Prinzip, dass eine Schicht der jeweils über ihr angeordneten Schicht bestimmte Dienstleistungen anbietet. Das OSI-Modell ist keine Netzarchitektur, da die Protokolle und Dienste der einzelnen Schichten vom Modell nicht definiert werden. Das Modell beschreibt lediglich, welche Aufgaben die Schichten erledigen sollen. Die folgenden Prinzipien haben zu den sieben Schichten des OSI-Modells geführt

[Tanenbaum1996]:

1. Eine neue Schicht sollte dort entstehen, wo ein neuer Abstraktionsgrad benötigt wird.
2. Jede Schicht sollte eine genau definierte Funktion erfüllen.
3. Bei der Funktionswahl sollte die Definition international genormter Protokolle berücksichtigt werden.
4. Die Grenzen zwischen den einzelnen Schichten sollten so gewählt werden, dass der Informationsfluss über die Schnittstellen möglichst gering ist.
5. Die Anzahl der Schichten sollte so groß sein, dass keine Notwendigkeit besteht, verschiedene Funktionen auf eine Schicht zu packen, aber so klein, dass die gesamte Architektur nicht unhandlich wird.

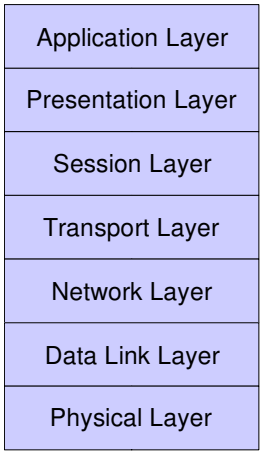


Abbildung 3 Das OSI-Referenzmodell.

Den Schichten im OSI-Modell sind die folgenden Aufgaben zugeordnet:

Anwendungsschicht (application layer): Die Anwendungsschicht enthält eine große Zahl häufig benötigter Protokolle, die einzelne Programme zur Erbringung ihrer Dienste definiert haben. Auf der Anwendungsschicht finden sich z.B. die Protokolle für die Dienste ftp, telnet, mail etc.

Darstellungsschicht (presentation layer): Die Darstellungsschicht regelt die Darstellung der Übertragungsdaten in einer von der darüber liegenden Ebene unabhängigen Form. Computersysteme verwenden z.B. oft verschiedene Codierungen für Zeichenketten (z.B. ASCII, Unicode), Zahlen usw. Damit diese Daten zwischen den Systemen ausgetauscht werden können, kodiert die Darstellungsschicht die Daten auf eine standardisierte und vereinbarte Weise.

Sitzungsschicht (session layer): Die Sitzungsschicht (oft auch Verbindungsschicht oder Kommunikationssteuerschicht genannt) ermöglicht den Verbindungsauf- und abbau. Von der Sitzungsschicht wird der Austausch von Nachrichten auf der Transportverbindung geregelt. Sitzungen können z.B. ermöglichen, ob der Transfer gleichzeitig in zwei oder nur eine Richtung erfolgen kann. Kann der Transfer jeweils in nur eine Richtung stattfinden, regelt die Sitzungsschicht, welcher der Kommunikationspartner jeweils an die Reihe kommt.

Transportschicht (transport layer): Die Transportschicht übernimmt den Transport von Nachrichten zwischen den Kommunikationspartnern. Die Transportschicht hat die grundlegende Aufgabe, den Datenfluss zu steuern und die Unverfälschtheit der Daten sicherzustellen. Beispiele für Protokolle der Transportschicht sind TCP und UDP.

Netzwerkschicht (network layer): Die Netzwerkschicht (Vermittlungsschicht) hat die

Hauptaufgabe eine Verbindung zwischen Knoten im Netzwerk herzustellen. Die Netzwerkschicht soll dabei die übergeordneten Schichten von den Details der Datenübertragung über das Netzwerk befreien. Eine der wichtigsten Aufgaben der Netzwerkschicht ist z.B. die Auswahl von Paketrouten bzw. das Routing vom Absender zum Empfänger. Das Internet Protokoll (IP) ist in der Netzwerkschicht einzuordnen.

Sicherungsschicht (data link layer): Die Aufgabe der Sicherungsschicht (Verbindungsschicht) ist die gesicherte Übertragung von Daten. Vom Sender werden hierzu die Daten in Rahmen (frames) aufgeteilt und sequentiell an den Empfänger gesendet. Vom Empfänger werden die empfangenen Daten durch Bestätigungsrahmen quittiert. Protokollbeispiele für die Sicherungsschicht sind HDLC (high-level data link control), SLIP (serial line IP) und PPP (point-to-point Protokoll).

Bitübertragungsschicht (physical layer): Die Bitübertragungsschicht regelt die Übertragung von Bits über das Übertragungsmedium. Dies betrifft die Übertragungsgeschwindigkeit, die Bit-Codierung, den Anschluss (wieviele Pins hat der Netzanschluss?) etc. Die Festlegungen auf der Bitübertragungsschicht betreffen im wesentlichen die Eigenschaften des Übertragungsmedium.

2.5.2 Das TCP/IP-Referenzmodell

Im vorhergehenden Abschnitt wurde das OSI-Referenzmodell vorgestellt. In diesem Abschnitt soll nun das Referenzmodell für die TCP/IP-Architektur vorgestellt werden. Das *TCP/IP-Referenzmodell* - benannt nach den beiden primären Protokollen TCP und IP der Netzarchitektur beruht auf den Vorschlägen, die bei der Fortentwicklung des ARPANETs gemacht wurden. Das TCP/IP-Modell ist zeitlich vor dem OSI-Referenzmodell entstanden, deshalb sind auch die Erfahrungen des TCP/IP-Modells mit in die OSI-Standardisierung eingeflossen. Das TCP/IP-Referenzmodell besteht im Gegensatz zum OSI-Modell aus nur vier Schichten: Application Layer, Transport Layer, Internet Layer, Network Layer. Als Ziele der Architektur wurden bei der Entwicklung definiert:

- Unabhängigkeit von der verwendeten Netzwerk-Technologie.
- Unabhängigkeit von der Architektur der Hostrechner.
- Universelle Verbindungsmöglichkeiten im gesamten Netzwerk.
- Ende-zu-Ende-Quittungen.
- Standardisierte Anwendungsprotokolle.

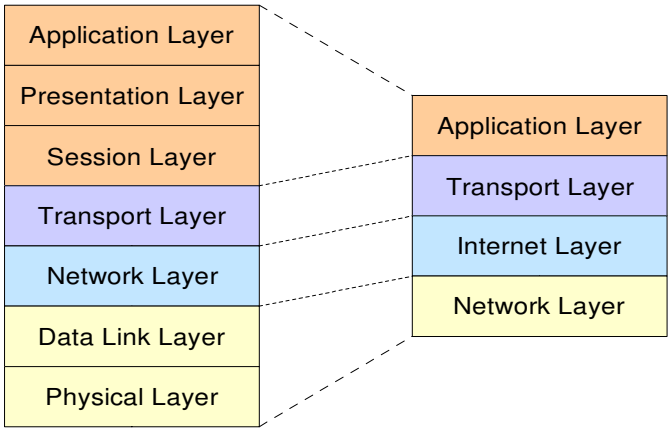


Abbildung 4 Vergleich des OSI-Referenzmodells mit dem TCP/IP-Referenzmodell.

Applikationsschicht (application layer): Die Applikationsschicht (auch Verarbeitungsschicht genannt) umfasst alle höherschichtigen Protokolle des TCP/IP-Modells. Zu den ersten Protokollen der Verarbeitungsschicht zählen TELNET (für virtuelle Terminals), FTP (Dateitransfer) und SMTP (zur Übertragung von E-Mail). Im Laufe der Zeit kamen zu den etablierten Protokollen viele weitere Protokolle wie z.B. DNS (Domain Name Service) und HTTP (Hypertext Transfer Protocol) hinzu.

Transportschicht (transport layer): Wie im OSI-Modell ermöglicht die Transportschicht die Kommunikation zwischen den Quell- und Zielhosts. Im TCP/IP-Referenzmodell wurden auf dieser Schicht zwei Ende-zu-Ende-Protokolle definiert: das Transmission Control Protocol (TCP) und das User Datagram Protocol (UDP). TCP ist ein zuverlässiges verbindungsorientiertes Protokoll, durch das ein Bytestrom fehlerfrei zu einem anderen Rechner im Internet übermittelt werden kann. UDP ist ein unzuverlässiges verbindungsloses Protokoll, das vorwiegend für Abfragen und Anwendungen in Client/Server-Umgebungen verwendet wird, in denen es in erster Linie nicht um eine exakte, sondern schnelle Datenübermittlung geht (z.B. Übertragung von Sprache und Bildsignalen).

Internetschicht (internet layer): Die Internetschicht im TCP/IP-Modell definiert nur ein Protokoll namens IP (Internet Protocol), das alle am Netzwerk beteiligten Rechner verstehen können. Die Internetschicht hat die Aufgabe IP-Pakete richtig zuzustellen. Dabei spielt das Routing der Pakete eine wichtige Rolle. Das Internet Control Message Protocol (ICMP) ist fester Bestandteil jeder IP-Implementierung und dient zur Übertragung von Diagnose- und Fehlerinformationen für das Internet Protocol.

Netzwerkschicht (network layer): Unterhalb der Internetschicht befindet sich im TCP/IP-Modell eine große Definitionslücke. Das Referenzmodell sagt auf dieser Ebene nicht viel darüber aus, was hier passieren soll. Festgelegt ist lediglich, dass zur Übermittlung von IP-Paketen ein Host über ein bestimmtes Protokoll an ein Netz angeschlossen werden muss. Dieses Protokoll ist im TCP/IP-Modell nicht weiter definiert und weicht von Netz zu Netz und Host zu Host ab. Das TCP/IP-Modell macht an dieser Stelle vielmehr Gebrauch von bereits vorhandenen Protokollen, wie z.B. Ethernet (IEEE 802.3), Serial Line IP (SLIP), etc.

3 TCP/IP im Detail

3.1 Die TCP/IP-Protokoll-Architektur

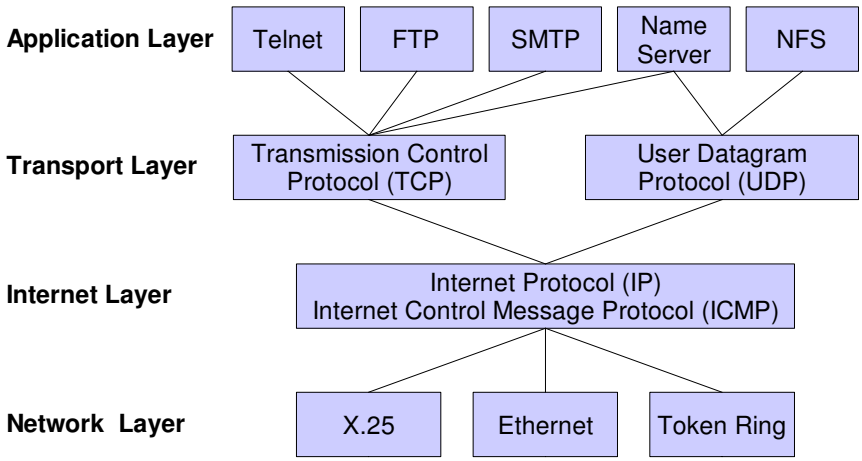


Abbildung 5 Die TCP/IP-Protokoll-Architektur.

Die TCP/IP-Architektur wird, wie im Abschnitt Referenzmodelle, Das TCP/IP-Referenzmodell gesagt, im allgemeinen als vierschichtiges Modell beschrieben. Oft wird das TCP/IP-Referenzmodell auch als fünfschichtiges Modell dargestellt. Andrew S. Tanenbaum bezeichnet das in [Tanenbaum1996] vorgestellte fünfschichtige Modell als *hybrides Referenzmodell.* Er schreibt in [Tanenbaum1996] dazu:

„(...) Viertens unterscheidet das TCP/IP-Modell nicht zwischen den Bitübertragungs- und Sicherungsschichten (erwähnt sie nicht einmal). Diese Schichten sind völlig unterschiedlich. Die Bitübertragungsschicht hat mit den Übertragungsmerkmalen von Kupferdarht, Glasfaser und drahtlosen Kommunikationsmedien zu tun. Die Sicherungsschicht ist darauf beschränkt, den Anfang und das Ende von Rahmen abzugrenzen und sie mit der gewünschten Zuverlässigkeit von einem Ende zum anderen zu befördern. Ein korrektes Modell sollte beides als separate Schichten beinhalten. Das TCP/IP-Modell tut das nicht.“

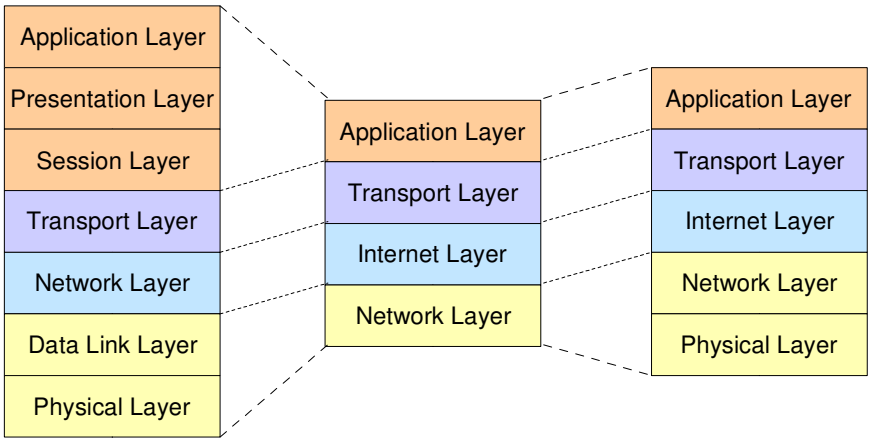


Abbildung 6 Vergleich: OSI-Referenzmodell, TCP/IP-Referenzmodell, Hybrides Referenzmodell.

Die Schichtung beruht auf dem Prinzip, dass eine Schicht die angebotenen Dienste der darunter lie-

genden Schicht in Anspruch nehmen kann. Dabei braucht die Schicht, die die Dienstleistung in Anspruch nimmt keinerlei Kenntnisse darüber haben, wie die geforderten Dienste erbracht werden bzw. implementiert sind. Auf diese Art und Weise wird eine Aufgabenteilung der Schichten erreicht (siehe dazu auch Referenzmodelle, Das TCP/IP-Referenzmodell). Daten, die von einem Applikationsprogramm über ein Netzwerk versendet werden, durchlaufen den TCP/IP-Protokollstapel von der Applikationsschicht zur Netzwerkschicht. Von jeder Schicht werden dabei Kontrollinformationen in Form eines Protokollkopfes angefügt. Diese Kontrollinformationen dienen der korrekten Zustellung der Daten. Das Zufügen von Kontrollinformationen wird als *Einkapselung (encapsulation)* bezeichnet.

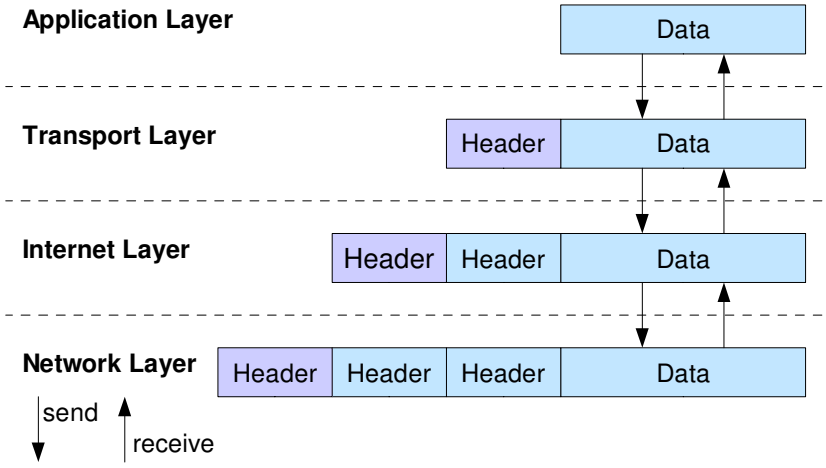


Abbildung 7 Dateneinkapselung [Hunt1995]

Innerhalb der Schichten des TCP/IP-Modells werden Daten mit verschiedenen Termini benannt, da jede Schicht auch ihre eigenen Datenstrukturen hat [Hunt1995]. Applikationen, die das Transmission Control Protocol benutzen, bezeichnen Daten als *Strom (stream)*; Applikationen, die das User Datagram Protocol verwenden, bezeichnen Daten als *Nachricht (message)*. Auf der Transportebene bezeichnen die Protokolle TCP und UDP ihre Daten als *Segment (segment)* bzw. *Paket (packet)*. Auf der Internet Schicht werden Daten allgemein als *Datengramm (datagram)* benannt. Oft werden die Daten hier aber auch als Paket bezeichnet. Auf der Netzwerkebene bezeichnen die meisten Netzwerke ihre Daten als *Pakete* oder *Rahmen (frames)*.

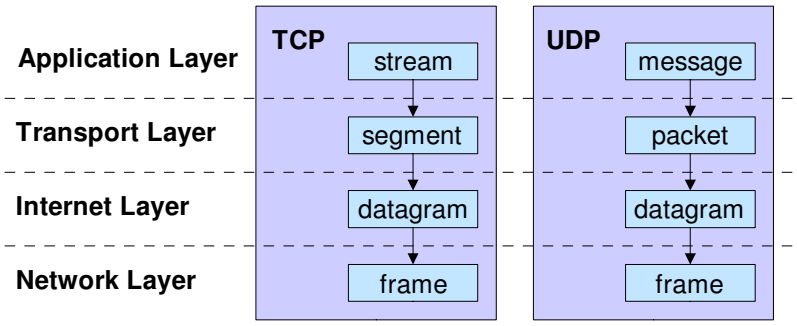


Abbildung 8 Bezeichnung der Daten auf den verschiedenen Schichten des TCP/IP-Modells [Hunt1995].

3.2 Netzwerkschicht

Die Netzwerkschicht ist die unterste Schicht des TCP/IP-Modells. Protokolle, die auf dieser Schicht angesiedelt sind, legen fest, wie ein Host an ein bestimmtes Netzwerk angeschlossen wird und wie IP-Pakete über dieses Netzwerk übertragen werden.

Im Gegensatz zu den Protokollen der höheren Schichten des TCP/IP-Modells, müssen die Protokolle der Netzwerkschicht sich auf die Details des verwendeten Netzwerks - wie z.B. Paketgrößen, Netzwerkadressierung, Anschlusscharakteristika etc. - beziehen. Die Netzwerkschicht des TCP/IP-Modells umfasst also die Aufgaben der Bitübertragungsschicht und der Sicherungsschicht im OSI-Modell.

Die Protokolle der Netzwerkschicht sind allerdings nicht im TCP/IP-Modell definiert. Wie schon gesagt, legt das Modell lediglich fest, dass zur Übermittlung von IP-Paketen ein Host über ein bestimmtes Protokoll an ein Netzwerk angeschlossen werden muss. Die Protokolle sind im Modell nicht weiter definiert. Es werden hier vielmehr bestehende Standards verwendet und in das Modell aufgenommen. Insbesondere bedeutet dies auch, dass mit neuer Hardware-Technologie auch neue Protokolle auf der Netzwerkschicht entwickelt werden müssen, so dass TCP/IP-Netzwerke diese Hardware verwenden können. Dies ist jedoch kein Nachteil, sondern eher ein Vorteil. Durch die weitgehende Unabhängigkeit vom Übertragungsmedium können neue Netzwerktechnologien schnell in das TCP/IP-Modell aufgenommen werden.

Hinweis: Eine Einführung in den Ethernet-Standard (von Michael Blume) ist unter der folgenden URL zu finden: http://www.rvs.uni-bielefeld.de/~mblume/seminar/ss97/ethernet

3.3 Internet-Schicht

Das Internet ist eine Sammlung von Teilnetzen, die miteinander verbunden sind. Es gibt keine echte Struktur des Netzes, sondern mehrere größere *Backbones*, die quasi das Rückgrat (wie der Name Backbone ja schon sagt) des Internet bilden. Die Backbones werden aus Leitungen mit sehr hoher

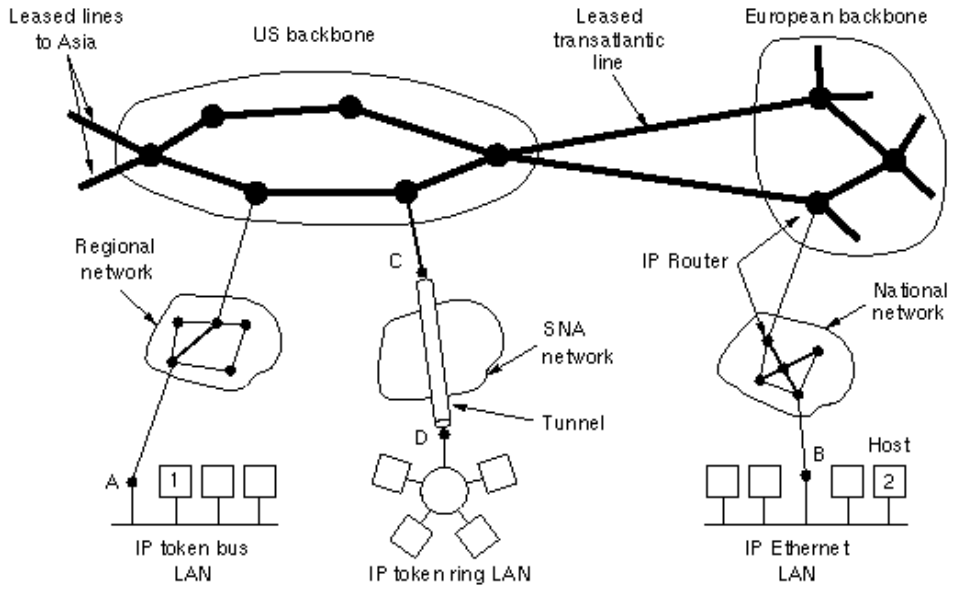


Abbildung 9 „Das Internet“ [Tanenbaum1996].

Bandbreite und schnellen Routern gebildet. An die Backbones sind wiederum größere regionale Netze angeschlossen, die lokale Netze von Universitäten, Behörden, Unternehmen und Service-Providern verbinden.

3.3.1 Internet Protokoll (IP)

Das *Internet Protokoll (Internet Protocol - IP)* ist der Leim, der dies alles zusammenhält. IP stellt die Basisdienste für die Übermittlung von Daten in TCP/IP-Netzen bereit und ist im RFC 791 spezifiziert. Hauptaufgaben des Internet Protokolls sind die Adressierung von Hosts und das Fragmentieren von Paketen. Diese Pakete werden von IP nach bestem Bemühen („best effort“) von der Quelle zum Ziel befördert, unabhängig davon, ob sich die Hosts im gleichen Netz befinden oder andere Netze dazwischen liegen. Garantiert ist die Zustellung allerdings nicht. Das Internet Protokoll enthält keine Funktionen für die Ende-zu-Ende-Sicherung oder für die Flusskontrolle.

Die Funktionen von IP umfassen:

- Die Definition von Datengrammen, welche die Basiseinheiten für die Übermittlung von Daten im Internet bilden.
- Definition des Adressierungsschemas.
- Übermittlung der Daten von der Transportebene zur Netzwerkschicht.
- Routing von Datengrammen durch das Netz.
- Fragmentierung und Zusammensetzen von Datengrammen.

IP ist ein *verbindungsloses* Protokoll, d.h. zur Datenübertragung wird keine Ende-zu-Ende-Verbindung der Kommunikationspartner etabliert. Ferner ist IP ein *unzuverlässiges* Protokoll, da es über keine Mechanismen zur Fehlererkennung und -behebung verfügt. Unzuverlässig bedeutet aber keinesfalls, dass man sich auf das IP Protokoll nicht verlassen kann. Unzuverlässig bedeutet in diesem Zusammenhang lediglich, dass IP die Zustellung der Daten nicht garantieren kann. Sind die Daten aber beim Zielhost angelangt, sind diese Daten auch korrekt.

IP-Datengramm

Die TCP/IP-Protokolle wurden entwickelt, um Daten über ein paketvermittelndes Netz (wie dem ARPANET) zu übertragen. Ein *Paket* ist ein Datenblock zusammen mit den Informationen, die notwendig sind, um sie dem Empfänger zuzustellen (ein Paket ist also nichts anderes als ein Paket im herkömmliche Sinn bei der Post - das Paket enthält die Daten, auf dem Paket ist die Adresse des Empfängers notiert). Das *Datengramm (datagram)* ist das Paketformat, das vom Internet Protokoll definiert ist. Ein IP-Datengramm besteht aus einem Header und den zu übertragenden Daten. Der Header hat einen festen 20 Byte großen Teil, gefolgt von einem optionalen Teil variabler Länge. Der Header umfasst alle Informationen, die notwendig sind, um das Datengramm dem Empfänger zuzustellen. Ein Datengramm kann theoretisch maximal 64 KByte groß sein, in der Praxis liegt die Größe ungefähr bei 1500 Byte (das hängt mit der maximalen Rahmengröße des Ethernet-Protokolls zusammen).

Die Felder des in der Abbildung dargestellten Protokollkopfes haben die folgende Bedeutung:

Version:

Das *Versions-Feld* enthält die Versionsnummer des IP-Protokolls. Durch die Einbindung der Versionsnummer besteht die Möglichkeit über eine längere Zeit mit verschiedenen Versionen des IP Protokolls zu arbeiten. Einige Hosts können mit der alten und andere mit der neuen

Version arbeiten. Die derzeitige Versionsnummer ist 4, aber die Version 6 des IP Protokolls befindet sich bereits in der Erprobung (siehe [Hartjes1997], [Hinden], [Hosenfeld1996], [Kuschke1994], [Tanenbaum1996], [WiN]).

Length:

Das Feld *Length (Internet Header Length - IHL)* enthält die Länge des Protokollkopfs, da diese nicht konstant ist. Die Länge wird in 32-Bit-Worten angegeben. Der kleinste zulässige Wert ist 5 - das entspricht also 20 Byte; in diesem Fall sind im Header keine Optionen gesetzt. Die Länge des Headers kann sich durch Anfügen von Optionen aber bis auf 60 Byte erhöhen (der Maximalwert für das 4-Bit-Feld ist 15).

Type of Servive:

Über das Feld *Type of Service* kann IP angewiesen werden Nachrichten nach bestimmten Kriterien zu behandeln. Als Dienste sind hier verschiedene Kombinationen aus Zuverlässigkeit und Geschwindigkeit möglich. In der Praxis wird dieses Feld aber ignoriert, hat also den Wert 0. Das Feld selbst hat den folgenden Aufbau:

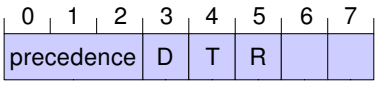


Abbildung 10 Aufbau des Type of Service Feldes.

Precedence (Bits 0-2) gibt die Priorität von 0 (normal) bis 7 (Steuerungspaket) an. Die drei Flags (D,T,R) ermöglichen es dem Host anzugeben, worauf er bei der Datenübertragung am meisten Wert legt: Verzögerung (Delay - D), Durchsatz (Throughput - T), Zuverlässigkeit (Reliability - R). Die beiden anderen Bit-Felder sind reserviert.

Total Length:

Enthält die gesamte *Paketlänge*, d.h. Header und Daten. Da es sich hierbei um ein 16-Bit-Feld handelt ist die Maximallänge eines Datengramms auf 65.535 Byte begrenzt. In der Spezifikation von IP (RFC 791) ist festgelegt, dass jeder Host in der Lage sein muss, Pakete bis zu einer Länge von 576 Bytes zu verarbeiten. In der Regel können von den Host aber Pakete größerer Länge verarbeitet werden.

Identification:

Über das *Identifikationsfeld* kann der Zielhost feststellen, zu welchem Datengramm ein neu angekommenes Fragment gehört. Alle Fragmente eines Datengramms enthalten die gleiche Identifikationsnummer, die vom Absender vergeben wird.

Flags:

Das Flags-Feld ist drei Bit lang. Die Flags bestehen aus zwei Bits namens *DF - Don't Fragment* und *MF - More Fragments*. Das erste Bit des Flags-Feldes ist ungenutzt bzw. reserviert. Die beiden Bits DF und MF steuern die Behandlung eines Pakets im Falle einer Fragmentierung. Mit dem DF-Bit wird signalisiert, dass das Datengramm nicht fragmentiert werden darf. Auch dann nicht, wenn das Paket dann evtl. nicht mehr weiter transportiert werden kann und verworfen werden muss. Alle Hosts müssen, wie schon gesagt Fragmente bzw. Datengramme mit einer Größe von 576 Bytes oder weniger verarbeiten können. Mit dem MF-Bit wird angezeigt, ob einem IP-Paket weitere Teilpakete nachfolgen. Diese Bit ist bei

allen Fragmenten außer dem letzten gesetzt.

Fragment Offset:

Der *Fragmentabstand* bezeichnet, an welcher Stelle relativ zum Beginn des gesamten Datengramms ein Fragment gehört. Mit Hilfe dieser Angabe kann der Zielhost das Originalpaket wieder aus den Fragmenten zusammensetzen. Da dieses Feld nur 13 Bit groß ist, können maximal 8192 Fragmente pro Datengramm erstellt werden. Alle Fragmente, außer dem letzten, müssen ein Vielfaches von 8 Byte sein. Dies ist die elementare Fragmenteinheit.

Time to Live:

Das Feld *Time to Live* ist ein Zähler, mit dem die Lebensdauer von IP-Paketen begrenzt wird. Im RFC 791 ist für dieses Feld als Einheit Sekunden spezifiziert. Zulässig ist eine maximale Lebensdauer von 255 Sekunden (8 Bit). Der Zähler muss von jedem Netzknoten, der durchlaufen wird um mindestens 1 verringert werden. Bei einer längeren Zwischenspeicherung in einem Router muss der Inhalt sogar mehrmals verringert werden. Enthält das Feld den Wert 0, muss das Paket verworfen werden: damit wird verhindert, dass ein Paket endlos in einem Netz umherwandert. Der Absender wird in einem solchen Fall durch eine Warnmeldung in Form einer *ICMP-Nachricht* (siehe weiter unten) informiert.

Protocol:

Enthält die Nummer des Transportprotokolls, an das das Paket weitergeleitet werden muss. Die Numerierung von Protokollen ist im gesamten Internet einheitlich. Bisher wurden die Protokollnummern im RFC 1700 definiert. Diese Aufgabe ist nun von der *Internet Assigned Numbers Authority (IANA)* [http://www.iana.org] übernommen worden. Bei UNIX-Systemen sind die Protokollnummern in der Datei `/etc/protocols` abgelegt.

Header Checksum:

Dieses Feld enthält die Prüfsumme der Felder im IP-Header. Die Nutzdaten des IP-Datengramms werden aus Effizienzgründen nicht mit geprüft. Diese Prüfung findet beim Empfänger innerhalb des Transportprotokolls statt. Die Prüfsumme muss von jedem Netzknoten, der durchlaufen wird, neu berechnet werden, da der IP-Header durch das Feld Time-to-Live sich bei jeder Teilstrecke verändert. Aus diesem Grund ist auch eine sehr effiziente Bildung der Prüfsumme wichtig. Als Prüfsumme wird *das 1er-Komplement der Summe aller 16-Bit-Halbwörter der zu überprüfenden Daten* verwendet. Zum Zweck dieses Algorithmus wird angenommen, dass die Prüfsumme zu Beginn der Berechnung Null ist.

Source Address, Destination Address:

In diese Felder werden die 32-Bit langen Internet-Adressen eingetragen. Die Internet-Adressen werden im Abschnitt Adressierung auf der Internet-Schicht näher betrachtet.

Options und Padding:

Das Feld *Options* wurde im Protokollkopf aufgenommen, um die Möglichkeit zu bieten das IP-Protokoll um weitere Informationen zu ergänzen, die im ursprünglichen Design nicht berücksichtigt wurden. Das Optionsfeld hat eine variable Länge. Jede Option beginnt mit einem Code von einem Byte, über den die Option identifiziert wird. Manchen Optionen folgt ein weiteres Optionsfeld von 1 Byte und dann ein oder mehrere Datenbytes für die Option.

Das Feld Options wird über das *Padding* auf ein Vielfaches von 4 Byte aufgefüllt. Derzeit sind die folgenden Optionen bekannt:

End of Option List
: Kennzeichnet das Ende der Optionsliste.

No Option
: Kann zum Auffüllen von Bits zwischen Optionen verwendet werden.

Security
: Bezeichnet, wie geheim ein Datengramm ist. In der Praxis wird diese Option jedoch fast immer ignoriert.

Loose Source-Routing, Strict Source-Routing
: Diese Option enthält eine Liste von Internet-Adressen, die das Datagramm durchlaufen soll. Auf diese Weise kann dem Datenpaket vorgeschrieben werden eine bestimmte Route durch das Internet zu nehmen. Beim Source-Routing wird zwischen *Strict Source and Record Route* und *Loose Source and Record Route* unterschieden. Im ersten Fall wird verlangt, dass das Paket diese Route genau einhalten muss. Desweiteren wird die genommene Route aufgezeichnet. Die zweite Variante schreibt vor, dass die angegebenen Router nicht umgangen werden dürfen. Auf dem Weg können aber auch andere Router besucht werden.

Record Route
: Die Knoten, die dieses Datengramm durchläuft, werden angewiesen ihre IP-Adresse an das Optionsfeld anzuhängen. Damit lässt sich ermitteln, welche Route ein Datengramm genommen hat. Wie anfangs schon gesagt, ist die Größe für das Optionsfeld auf 40 Byte beschränkt. Deshalb kommt es heute auch oftmals zu Problemen mit dieser Option, da weit mehr Router durchlaufen werden, als dies zu Beginn des ARPANET der Fall war.

Time Stamp
: Diese Option ist mit der Option Record Route vergleichbar. Zusätzlich zur IP-Adresse wird bei dieser Option die Uhrzeit des Durchlaufs durch den Knoten vermerkt. Auch diese Option dient hauptsächlich zur Fehlerbehandlung, wobei zusätzlich z.B. Verzögerungen auf den Netzstrecken erfasst werden können.

Weitere Details zu den Optionen sind in RFC 791 zu finden.

3.3.2 Adressierung auf der Internet-Schicht

Zur Adressierung eines Kommunikationspartners in Form eines Applikationsprogramms müssen beim Durchlaufen der vier TCP/IP-Schichten auch vier verschiedene Adressen angegeben werden.

1. Eine Netzwerkadresse (z.B. eine Ethernet-Adresse, MAC-Adresse)
2. Eine Internet-Adresse
3. Eine Transportprotokoll-Adresse
4. Eine Portnummer

Zwei dieser Adressen finden sich als Felder im IP-Header: die *Internet-Adresse* und die *Transportprotokoll-Adresse*.

Protokollnummern

IP verwendet *Protokollnummern*, um empfangene Daten an das richtige Transportprotokoll weiterzuleiten. Die Protokollnummer ist ein einzelnes Byte im IP-Header. Die Protokollnummern sind im gesamten Internet einheitlich. Protokollnummern wurden im RFC 1700 definiert. Diese Aufgabe ist nun von der *Internet Assigned Numbers Authority (IANA)* [http://www.iana.org] bzw. der *Internet Corporation for Assigned Names and Numbers (ICANN)* [http://www.icann.org] übernommen worden. Die Nummern werden in einer Datenbank unter http://www.iana.org/numbers.htm verwaltet.

Auf UNIX-Systemen sind die Protokollnummern in der Datei `/etc/protocols` abgelegt. Diese Datei ist eine einfache Tabelle, die einen Protokollnamen und die damit verbundene Protokollnummer enthält. Nachfolgend ist der Inhalt der Datei `/etc/protocols` einer aktuellen LINUX-Maschine abgebildet:

```
heiko@phoenix:~> more /etc/protocols
#
# protocols     This file describes the various protocols that are
#               available from the TCP/IP subsystem.  It should be
#               consulted instead of using the numbers in the ARPA
#               include files, or, worse, just guessing them.
#

ip      0       IP      # internet protocol, pseudo protocol number
icmp    1       ICMP    # internet control message protocol
igmp    2       IGMP    # internet group multicast protocol
ggp     3       GGP     # gateway-gateway protocol
tcp     6       TCP     # transmission control protocol
pup     12      PUP     # PARC universal packet protocol
udp     17      UDP     # user datagram protocol
idp     22      IDP     # WhatsThis?
raw     255     RAW     # RAW IP interface

# End.
```

Empfängt IP ein Datengramm, in dessen Header als Protokollnummer 6 eingetragen ist, so werden diese Daten an das Transmission Control Protocol weitergeleitet; ist die Nummer 17, werden die Daten an das User Datagram Protocol weitergeleitet etc.

IP-Adressen

Jeder Host und Router im Internet hat (mindestens) eine 32-Bit lange IP-Adresse. Eine IP-Adresse ist eindeutig, kein Knoten im Internet hat die gleiche IP-Adresse wie ein anderer. Knoten, die an mehrere Netze angeschlossen sind, haben in jedem Netz eine eigene IP-Adresse. Es gilt: *Jede Schnittstelle in jedem Host und Router muss im Internet eine global eindeutige IP-Adresse haben* [KuroseRoss2002]. Die Vergabe von IP-Adressen wird zentral organisiert, um Adresskonflikte zu vermeiden. Diese Aufgabe wurde eine längere Zeit vom *Network Information Center (NIC)* [http://www.internic.net] wahrgenommen und ist nun an die *Internet Assigned Numbers Authority (IANA)* bzw. an die *Internet Corporation for Assigned Names and Numbers (ICANN)* übergegangen.

Die ICANN hat Teile des Adressraums an ihre Vertreter in den verschiedenen regionalen Gebieten - *Asia Pacific Network Information Center (APNIC)*, *American Registry for Internet Numbers (ARIN)*, *Réseaux IP Européens (RIPE)* – vergeben, die die IP-Adressen dann an Unternehmen, Behörden und *Internet Service Provider (ISP)* vergeben. Die Adressen werden nicht einzeln zugeordnet, sondern nach Netzklassen vergeben. Beantragt man IP-Adressen für ein Netz, so erhält man nicht für jeden Rechner eine einzelne Adresse zugeteilt, sondern einen Bereich von Adressen, der selbst zu verwalten ist. Nur Internet Service Provider, die sehr viel Adressraum benötigen, wenden sich noch direkt an die jeweiligen Vertreter der IANA bzw. ICANN in ihren Gebieten. Alle anderen Unternehmen wenden sich für die Zuordnung von IP-Adressen bzw. Adressbereichen an ihren ISP. Die Grundlage für die Vergabe von IP-Adressen bilden die Richtlinien, die in RFC 2050

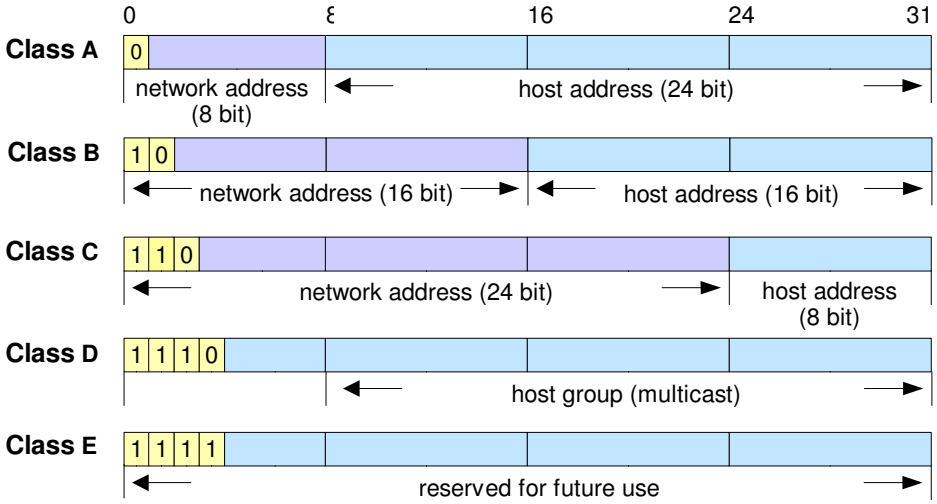


Abbildung 11 IP-Adressformate.

(*Internet Registry IP Allocation Guidelines*) festgehalten sind.

Wie die Abbildung „IP-Adressformate“ zeigt, sind IP-Adressen in verschiedene Klassen, mit unterschiedlich langer Netzwerk- und Hostadresse, eingeteilt. Die *Netzwerkadresse* definiert das Netzwerk, in dem sich ein Host befindet: alle Hosts eines Netzes haben die gleiche Netzwerkadresse. Die *Hostadresse* identifiziert einen bestimmten Rechner bzw. eine Schnittstelle innerhalb eines Netzes.

Abbildung 12 IP-Adressen bestehen aus einer Netzwerkadresse und einer Hostadresse.

Ist ein Host an mehrere Netze angeschlossen, so hat er für jedes Netz eine eigene IP-Adresse. *"Eine IP-Adresse identifiziert keinen bestimmten Computer (Host), sondern eine Verbindung zwischen einem Computer (Host) und einem Netz. Ein Computer (Host) mit mehreren Netzanschlüssen (z.B. ein Router) muss für jeden Anschluss eine IP-Adresse zugewiesen werden."* [Comer1998].

IP-Adressen sind 32-Bit große Zahlen, die normalerweise nicht als Binärzahl, sondern in

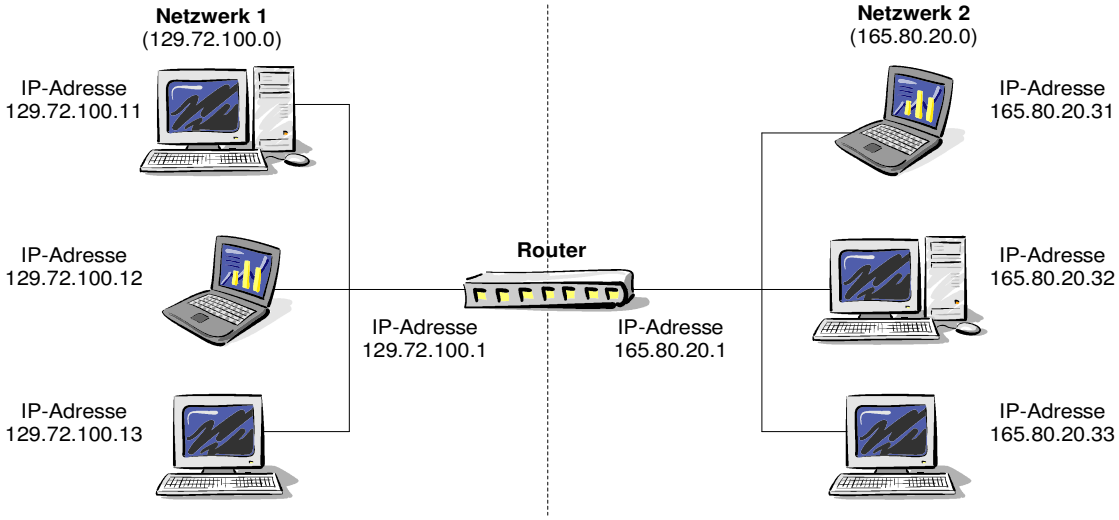


Abbildung 13 Jedes Netzwerk hat eine verschiedene Netzwerkadresse und jeder Host im Netzwerk hat eine andere Hostadresse.

gepunkteter Dezimalnotation geschrieben werden. In diesem Format wird die 32-Bit große Zahl in 4 Byte-Blöcke aufgeteilt, die mit Punkten voneinander getrennt sind. Die Adresse 01111111 11111111 11111111 11111111 wird so als 127.255.255.255 geschrieben. Die niedrigste IP-Adresse ist 0.0.0.0., die höchste 255.255.255.255.

Wie zuvor gesagt, sind IP-Adressen in *Klassen* unterteilt. Der Wert des ersten Bytes gibt die Adressklasse an:

Adressklasse	Erstes Byte	Bytes für die Netzadresse	Bytes für die Hostadresse	Adressformat*	Anzahl Hosts
Klasse A	1-126	1	3	N.H.H.H	2^{24} (~16 Mio.)
Klasse B	128-191	2	2	N.N.H.H.	2^{16} (~64.000)
Klasse C	192-223	3	1	N.N.N.H	254
Klasse D	224-239	Multicast-Adressen			
Klasse E	240-254	Experimentelle Adressen bzw. für zukünftige Nutzung reserviert			

*N steht für einen Teil der Netzadresse, H für einen Teil der Hostadresse.

Klasse A:

Das erste Byte hat einen Wert kleiner als 128, d.h. das erste Bit der Adresse ist 0. Das erste Byte ist Netzwerknummer, die letzten drei Bytes identifizieren einen Host im Netz. Es gibt demzufolge also 126 Klasse A Netze, die bis zu 16 Millionen Host in einem Netz.

Klasse B:

Ein Wert von 128 bis 191 für das erste Byte (das erste Bit ist gleich 1, Bit 2 gleich 0) identifiziert eine Klasse B Adresse. Die ersten beiden Bytes identifizieren das Netzwerk, die letzen beiden Bytes einen Host. Das ergibt 16.382 Klasse B Netze mit bis zu 64.000 Hosts in einem Netz.

Klasse C:

Klasse C Netze werden über Werte von 192 bis 223 für das erste Byte (die ersten beiden Bits sind gleich 1, Bit 3 gleich 0) identifiziert. Es gibt 2 Millionen Klasse C Netze, d.h. die ersten drei Bytes werden für die Netzwerkadresse verwendet. Ein Klasse C Netz kann bis zu 254 Host beinhalten.

Klasse D:

Klasse D Adressen, so genannte *Multicast-Adressen*, werden dazu verwendet ein Datengramm an mehrere Hostadressen gleichzeitig zu versenden. Das erste Byte einer Multicast-Adresse hat den Wertebereich von 224 bis 239, d.h. die ersten drei Bit sind gesetzt und Bit 4 ist gleich 0. Sendet ein Prozeß eine Nachricht an eine Adresse der Klasse D, wird die Nachricht an alle Mitglieder der adressierten Gruppe versendet. Die Übermittlung der Nachricht erfolgt, wie bei IP üblich, nach bestem Bemühen, d.h. ohne Garantie, dass die Daten auch tatsächlich alle Mitglieder einer Gruppe erreichen.

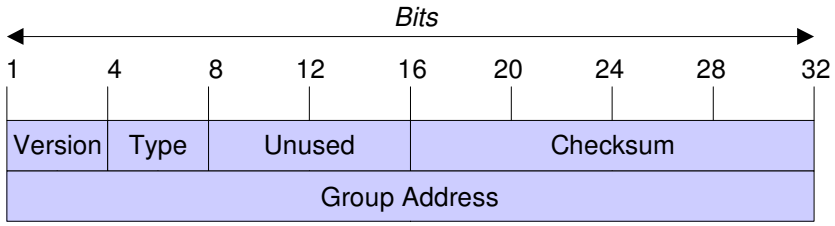


Abbildung 14 Der IGMP-Header.

Für das Multicasting wird ein spezielles Protokoll namens *Internet Group Management Protocol (IGMP)* verwendet. IGMP entspricht grob ICMP, mit dem Unterschied, dass es nur zwei Arten von Paketen kennt: Anfragen und Antworten. Anfragen werden dazu verwendet, zu ermitteln welche Hosts Mitglieder einer Gruppe sind. Antworten informieren darüber, zu welchen Gruppen ein Host gehört. Jedes IGMP-Paket hat ein festes Format und wird zur Übertragung in IP-Pakete eingekapselt.

Version

In RFC1112 ist die aktuelle Version 1 des IGMP Protokolls spezifiziert. Version 0, die in RFC998 beschrieben wird, ist obsolet.

Type

Wie zuvor gesagt, kennt IGMP zwei Nachrichtentypen: Anfragen und Antorten:
1 = Host Membership Query (Anfrage)
2 = Host Membership Report (Antwort)

Unused

Dieses Feld wird derzeit nicht benutzt.

Checksum

Der Algorithmus zur Berechnung der Checksumme entspricht dem des IP-Protokolls.

Group Address

Bei einer Anfrage zur Gruppenzugehörigkeit wird das Gruppenadressenfeld mit Nullen gefüllt. Ein Host, der eine Anfrage erhält, ignoriert dieses Feld. Bei einer IGMP-

Antwort enthält das Gruppenadressenfeld die Adresse der Gruppe, zu der der sendende Host gehört.

Eine genaue Beschreibung des Internet Group Management Protocol ist in RFC1112 zu finden (RFC1054 und RFC 998 beschreiben ältere Versionen von IGMP).

Der weitere Bereich der IP-Adressen von 240 bis 254 im ersten Byte ist für zukünftige Nutzungen reserviert. In der Literatur wird dieser Bereich oft auch als Klasse E bezeichnet (vgl. [Comer1998]).

Im Internet müssen die Adressen eindeutig sein. Aus diesem Grund werden die IP-Adressen, wie bereits gesagt, von einer zentralen Organisation vergeben bzw. deren Vergabe koordiniert. Dabei ist sichergestellt, dass die Adressen eindeutig und auch im Internet sichtbar sind.

Private IP-Adressen

Dies ist aber nicht immer notwendig. Netze, die keinen Kontakt zum globalen Internet haben („geschlossene Netzwerke“), benötigen keine Adresse, die auch im Internet sichtbar ist. Es ist auch nicht notwendig, dass sichergestellt ist, das diese Adressen in keinem anderen, privaten Netz eingesetzt werden. Aus diesem Grund wurden Adressbereiche festgelegt, die nur für *private Netze* bestimmt sind. Diese Bereiche sind in RFC 1918 (*Address Allocation for Private Internets*) festgelegt (RFC 1597, auf das sich oft auch neuere Literatur bezieht, ist durch RFC 1918 ersetzt). IP-Nummern aus dem privaten Adressenbereich dürfen im globalen Internet nicht weitergeleitet werden. Dadurch ist es möglich, diese Adressen in beliebig vielen, nicht-öffentlichen Netzen, einzusetzen.

Die folgenden Adressbereiche sind für die Nutzung in privaten Netzen reserviert (nach RFC 1918):

Klasse A: 10.0.0.0
: Für ein privates Klasse A-Netz ist der Adressbereich von 10.0.0.0 bis 10.255.255.254 reserviert.

Klasse B: 172.16.0.0 bis 172.31.0.0
: Für die private Nutzung sind 16 Klasse B-Netze reserviert. Jedes dieser Netze kann aus bis zu 65.000 Hosts bestehen (also z.B. ein Netz mit den Adressen von 172.17.0.1 bis 172.17.255.254).

Klasse C: 192.168.0.0 bis 192.168.255.0
: 256 Klasse C-Netzte stehen zur privaten Nutzung zur Verfügung. Jedes dieser Netze kann jeweils 254 Hosts enthalten (z.B. ein Netz mit den Adressen 192.168.0.1 bis 192.168.0.254).

Jeder kann aus diesen Bereichen den Adressbereich für sein eigenes privates Netz auswählen. Die Zuteilung dieser Adressen bedarf nicht die Koordination mit der IANA bzw. ICANN oder einer anderen Organisation, die für die Zuordnung von IP-Adressen verantwortlich ist.

Spezielle IP-Adressen

Neben den IP-Adressen aus dem privaten Adressbereich gibt es weitere IP-Adressen, die eine bestimmte Bedeutung haben.

Adressen mit der Netznummer 0 beziehen sich auf das aktuelle Netz. Mit einer solchen Adresse

können sich Hosts auf ihr eigenes Netz beziehen, ohne die Netzadresse zu kennen (allerdings muss bekannt sein, um welche Netzklasse es sich handelt, damit die passende Anzahl Null-Bytes gesetzt wird). Insbesondere hat die Adresse 0.0.0.0, bei der alle Bits 0 sind, die Bedeutung „Dieser Host“ („This Host“).

Der Wert 127 im ersten Byte steht für das *Loopback Device* eines Hosts. Pakete, die an eine Adresse der Form 127.x.y.z gesendet werden, werden nicht auf einer Leitung bzw. Schnittstelle ausgegeben, sondern lokal verarbeitet. Dieses Merkmal wird häufig zur Fehlerbehandlung benutzt. In der Literatur wird häufig die Adresse 127.0.0.1 als Adresse für das Loopback-Device angegeben, es gilt aber, das alle Adressen der Form 127.x.y.z das Loopback-Device adressieren sollten.

Neben einigen Netzadressen sind auch bestimmte Hostadressen für spezielle Zwecke reserviert. Bei allen Netzwerkklassen sind die Werte 0 und 255 bei den Hostadressen reserviert.

Eine IP-Adresse, bei der alle Hostbits auf Null gesetzt sind, identifiziert das Netz selbst. Die Adresse 80.0.0.0 bezieht sich so z.B. auf das Klasse-A-Netz 80, die Adresse 128.66.0.0 bezieht sich auf das Klasse-B-Netz 128.66.

Eine IP-Adresse, bei der alle Host-Bytes den Wert 255 haben, ist eine *Broadcast-Adresse*. Eine Broadcast-Adresse wird benutzt, um *alle* Hosts in einem Netzwerk zu adressieren. Die Adresse, bei der alle Bits gesetzt sind (255.255.255.255) wird als *eingeschränkte Broadcast-Adresse* (loakler Broadcast) bezeichnet, die nur im lokalen Netz verarbeitet und nicht über Router weitergeleitet wird.

Subnetting (Teilnetzwerke)

Mit der ursprünglichen Einteilung von IP-Adressen in Adressklassen wurde beabsichtigt, durch den Netzwerkteil ein physikalisches Netzwerk eindeutig zu erkennen und die Weiterleitung von Paketen zu diesem Netz zu vereinfachen. Dieser Ansatz birgt aber einen großen Nachteil in sich: der verfügbare Adressraum ist verschwenderisch und ungünstig aufgeteilt. Bei der Vergabe einer Netzwerkadresse pro physikalischen Netz wird der IP-Adressraum viel schneller als notwendig aufgebraucht. Das Problem liegt im wesentlichen darin, dass sich die Adresse eines Klasse A, B oder C Netzes auf *ein* physikalisches Netz bezieht und nicht auf eine Gruppe von LANs.

Als Beispiel, um dieses Problem zu verdeutlichen, stelle man sich ein großes Unternehmen oder eine Universität vor, die ihre internen Netze an das Internet anschließen möchte. Ungeachtet der Größe bzw. der Anzahl der vorhandenen Hosts, müsste das Unternehmen für jedes Netzwerk mindestens eine Netzwerkadresse der Klasse C nutzen. Besteht eines der Netze z.B. nur aus 10 Knoten würden dabei 244 Adressen (256-2-10 = 244) vergeudet. Schlimmer noch, für ein Netz mit mehr als 255 Hosts würde eine Klasse-B-Adresse benötigt; sind an dieses Netzt z.B. nur 300 Hosts angeschlossen würden über 16.000 Adressen verschwendet.

Um den theoretisch vorhandenen IP-Adressraum aufzubrauchen, müssten über 4 Milliarden Hosts an das Internet angeschlossen werden (verfügbarer IP-Adressraum $\approx 2^{32}$), um aber z.B. den Adressraum, der für Klasse-B-Netze reserviert ist, aufzubrauchen, genügen 16.000 physikalische Netze (2^{14} = verfügbarer Adressraum für die Netzwerkadresse), die eine Klasse-B-Netzadresse verwenden.

Als Lösung des Problems kann man ein Netz für die interne Verwendung in mehrere Teilnetze aufteilen, wobei das Netz nach außen weiterhin als ein einzelnes Netz erscheint. Der Mechanismus, der diese Verfeinerung in so genannte Teilnetze oder Subnetze (subnets, subnetworks) ermöglicht,

wird als *subnetting* bezeichnet. Subnetze sind in RFC 950 (*Internet Standard Subnetting Procedure*) definiert. Beim Subnetting wird ein Teil der Hostadresse dazu genutzt, um den Netzwerkteil zu erweitern.

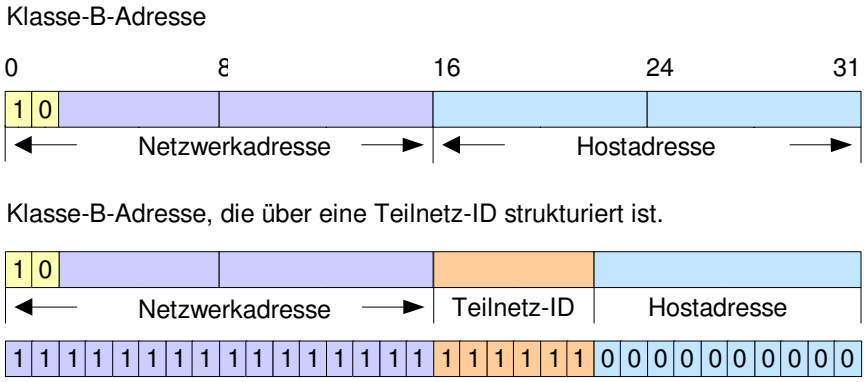


Abbildung 15 Klasse-B-Netz, das über eine Subnetzmaske in 64 Teilnetze unterteilt ist.

Anstatt beispielsweise eine einzelne Adresse der Klasse B mit 14 Bit (bzw. 16 Bit) für die Netzwerknummer und 16 Bit für die Hostnummer zu haben, werden aus der Hostnummer Bits entnommen, um damit eine *Teilnetz-ID* zu erstellen. Ein Unternehmen könnte so z.B. eine Klasse-B-Adresse beantragen und diese Adresse weiter unterteilen. Von den Hostbits könnten 6 Bit genutzt werden, um Teilnetze zu identifizieren, zur Adressierung der Hosts blieben dann 10 Bit. Damit ist es dem Unternehmen möglich, 64 Subnetzwerke mit bis zu 1.022 Host über eine Klasse-B-Adresse zu betreiben.

Um den Mechanismus für Subnetzwerke zu implementieren, müssen Hosts und Router um eine *Teilnetzmaske (Subnet Mask)*, die die Grenze zwischen der Netwerk-ID, der Teilnetz-ID und der Host-ID angibt, erweitert werden. Wie viele Bit für die Teilnetz-ID genutzt werden, bleibt dem Administrator überlassen. Die Teilnetzmaske ist eine 32 Bit große „Adresse“, die dazu benutzt wird, die Netzwerk-ID von der Host-ID in einer bestimmten IP-Adresse zu unterscheiden. Alle Bits der IP-Adresse, die zur Netzwerk-ID gehören, haben in der Teilnetzmaske den Wert 1; alle Bits, die zur Host-ID gehören, haben den Wert 0. Teilnetzmasken werden, ebenso wie IP-Adressen, üblicherweise in der Punktdezimalnotation geschrieben. Für die oben genannte Aufteilung würde die Subnetzmaske 255.255.252.0 (binär 11111111.11111111.11111100.00000000) lauten. Alternativ werden Subnetzmasken auch oft in der Form */xx* angegeben, wobei xx die Anzahl der Bits für den Netzwerkteil der Adresse angibt. Für obiges Beispiel würde die Notation also /22 lauten (22 Bit werden für den Netzwerkteil der Adresse genutzt, die Teilnetzmaske ist 22 Bit lang). Außerhalb des so aufgeteilten Netzes sind die Teilnetze nicht erkennbar, die Aufteilung in Subnetze ist für den Rest des Internets transparent, die innere Strukturierung bleibt verborgen. Deshalb ist es auch nicht notwenig, dass die Aufteilung eines Netzes in Subnetze mit der ICANN koordiniert wird oder andere Router im Internet rekonfiguriert werden müssen.

Als Beispiel sei angenommen, dass das Klasse-C-Netzwerk 192.96.34.0 in zwei Subnetzwerke aufgeteilt werden soll. Alles was dafür getan werden muss ist, statt der Standard-Subnetzmaske /24 (255.255.255.0) für ein Klasse-C-Netzwerk die Subnetzmaske /25 (255.255.255.128) zu wählen. Diese Subnetzmaske teilt das Klasse-C-Netz in zwei Teilnetze mit jeweils 128 möglichen Host-Adressen (tatsächlich sind es aufgrund der speziellen IP-Adressen jeweils 126 Hostadressen). Das erste Netz hat einen Adressraum von 192.96.34.0 bis 192.96.34.127, das zweite Netz einen Adressraum von 192.96.34.128 bis 192.96.34.255.

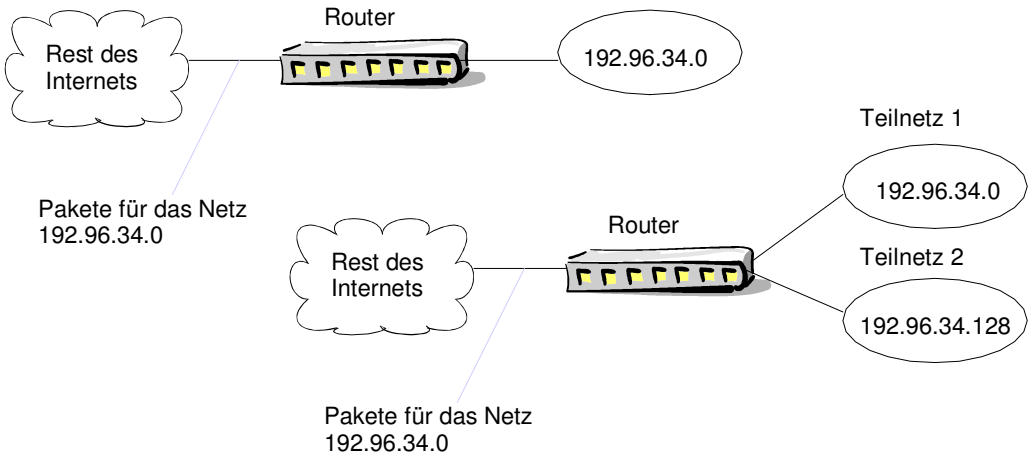


Abbildung 16 Aufteilung eines Netzes in Subnetze.

Die nachfolgende Tabelle zeigt die gebräuchlichsten Netzmasken für Klasse-B- und Klasse-C-Netze mit ihrer Anzahl Teilnetze und der Anzahl Hosts in einem Teilnetz:

Subnetzmaske		Netzwerk Bits	Netze per Subnetzmaske	Host Bits	Hosts per Subnetz	Netzwerkklasse
255.255.0.0	/16	16	1	16	65.534	Klasse B Maske (Standard)
255.255.128.0	/17	17	2	15	32.766	Klasse B Maske
255.255.192.0	/18	18	4	14	16.382	Klasse B Maske
255.255.224.0	/19	19	8	13	8.190	Klasse B Maske
255.255.240.0	/20	20	16	12	4.094	Klasse B Maske
255.255.248	/21	21	32	11	2.046	Klasse B Maske
255.255.252.0	/22	22	64	10	1.022	Klasse B Maske
255.255.254	/23	23	128	9	510	Klasse B Maske
255.255.255.0	/24	24	1	8	254	Klasse C Maske (Standard)
255.255.255.128	/25	25	2	7	126	Klasse C Maske
255.255.255.192	/26	26	4	6	62	Klasse C Maske
255.255.255.224	/27	27	8	5	30	Klasse C Maske
255.255.255.240	/28	28	16	4	14	Klasse C Maske
255.255.255.248	/29	29	32	3	6	Klasse C Maske
255.255.255.252	/30	30	64	2	2	Klasse C Maske

Mit RFC 950 ist festgelegt, dass jeder Host in einem IP-Netzwerk eine Subnetzmaske haben muss, entweder eine *Standard-Subnetzmaske (default-Subnetzmaske)* oder eine *angepasste Subnetzmaske (custom-Subnetzmaske)*. Wird ein Netz in mehrere Teilnetze aufgeteilt, muss jeder Host innerhalb eines Netzes die gleiche Subnetzmaske haben, damit er mit den anderen Hosts im Netz kommunizieren kann. Haben die Hosts unterschiedliche Subnetzmasken, „denken“ sie, sie wären in unterschiedlichen Netzen und müssen erst einen Router kontaktieren, um miteinander zu kommunizieren.

Alle Geräte in einem (IP-)Netzwerk müssen Routing-Entscheidungen (wie werden die Pakete weitergeleitet) treffen. Die Routing-Entscheidungen sind in den meisten Fällen sehr einfach:

- Liegt der Zielhost im lokalen Netz, werden die Daten direkt an den Zielhost geliefert.
- Liegt der Zielhost in einem entfernten Netzwerk, werden die Daten an einen lokalen Router weitergeleitet (oft auch Gateway genannt), der ausführlichere Routing-Tabellen hat.

Jeder Router führt intern eine Tabelle, aus der hervorgeht, wie IP-Pakete zu Hosts weitergeleitet werden müssen. Kommt ein IP-Paket bei einem Router an, wird seine Zieladresse in der *Routing-Tabelle (Weiterleitungstabelle)* nachgeschlagen. Die Routing-Tabelle setzt sich aus Einträgen im Format <Netzwerk-ID, NächsterHop> zusammen. Ist das Paket an einen Host adressiert, der im selben lokalen Netz liegt, wird das Paket direkt an das Ziel übertragen. Ist das Paket für einen Host in einem entfernten Netz bestimmt und dieses entfernte Netz in der Tabelle bekannt ist, wird es an den nächsten Router, der zu diesem Netz führt, an der in der Tabelle aufgeführten Schnittstelle weitergegeben. Ist das Netz an das das Paket adressiert ist nicht bekannt, wird das Paket an einen *Standard-Router (Standard-Gateway)* weitergeleitet, der ausführlichere Tabellen hat. Dieser Algorithmus bedeutet, dass jeder Router nur andere entfernte Netzwerke und lokale Hosts kennen muss, nicht aber Netzwerk/Host-Paare. Dadurch wird der Umfang der Routing-Tabellen stark reduziert.

Für die Unterstützung von Subnetting haben die Einträge in den Routing-Tabellen das Format <Subnetz-ID, Subnetzmaske, NächsterHop>. Um den richtigen Eintrag in der Tabelle zu finden, führt der Router ein *boolsches UND* zwischen der Zieladresse des Pakets und der Subnetzmaske jeden Eintrags in der Routing-Tabelle nacheinander durch. Stimmt das Ergebnis mit der Subnetz-ID eines Eintrags überein, ist dies der zu verwendende Eintrag und das Paket wird an den dadruch bezeichneten Router weitergeleitet.

> **Hinweis:** Der Inhalt der Routing-Tabelle kann mit den Befehlen `netstat -nr` oder `route print` auf den meisten UNIX und Windows-Systemen angezeigt werden.

Um dies zu verdeutlichen, sei das oben genannte Beispiel des Klasse-C-Netzes aufgegriffen. Ein entfernter Host adressiert ein IP-Paket an die IP-Adresse 192.96.34.160. Der Zielhost liegt also in dem Klasse-C-Netzwerk 192.96.34.0. Bei Erreichen des Routers, der die Pakete an Hosts im Netz 192.96.34.0 weiterleitet, das intern über die Subnetzmaske 255.255.255.128 in zwei Teilnetze aufgeteilt ist, führt nun der Router ein boolsches UND der Zieladresse mit der Subnetzwerkmaske des Netzwerkes durch.

`11000000.01100000.00100010.10100000`	`192.96.34.160`	Klasse-C-Adresse
`11111111.11111111.11111111.10000000`	`255.255.255.128`	Subnetzmaske für das Teilnetz 192.96.34.0
`11000000.01100000.00100010.10000000`	`192.96.34.128`	Netzwerkadresse des Teilnetzes

Die Berechnung ergibt, dass das Paket an den Host 192.96.34.160 in Teilnetz 2 (192.96.34.128) weitergeleitet werden muss.

Der Mechanismus des Subnetting hilft, die Zuweisung von IP-Adressen effizienter durchzuführen. Es muss nicht jedes mal eine ganze Adresse der Klasse B oder C aufgebraucht werden, wenn es neues physikalisches Netzwerk an das Internet angeschlossen wird. Stattdessen, kann ein bestehendes Netz strukturiert und besser ausgenutzt werden. Zusätzlich ist Subnetting bei der

Aggregation von Informationen nützlich. Eine komplexe Sammlung physikalischer Netze kann nach außen wie ein einzelnes Netzwerk dargestellt werden. Damit lässt sich der Umfang der Informationen, die in den Routern gespeichert werden müssen, um einen Host in einem dieser Netze Pakete zu übermitteln, erheblich reduzieren.

3.3.3 Fragmentierung

Damit Datengramme über jede Art von Netzwerk verschickt werden können, muss das Internet Protokoll dazu in der Lage sein, die Größe der Datengramme dem jeweiligen Netz anzupassen. Jedes Netzwerk besitzt eine so genannte *maximale Paketgröße (Maximum Transfer Unit - MTU)*, die bezeichnet, dass nur Pakete bis zu dieser Größe über das Netz verschickt werden können. So dürfen z.B. Pakete, die über ein X.25-Netz verschickt werden sollen nicht größer als 128 Byte sein. Ein Ethernet-Paket darf die Größe von 1500 Byte nicht überschreiten. Falls die MTU eines Übertragungsmediums kleiner ist als die Größe eines versendeten Pakets, so muss dieses Paket in kleinere Pakete aufgeteilt werden.

Es genügt allerdings nicht, dass die Protokolle der Transportschicht nun von sich aus einfach kleinere Pakete versenden. Ein Paket kann auf dem Weg vom Quell- zum Zielhost mehrere unterschiedliche Netzwerke mit unterschiedlichen MTUs durchlaufen. Aus diesem Grund muss ein flexibleres Verfahren angewendet werden, dass bereits auf der Internet-Schicht kleiner Pakete erzeugen kann. Dieses Verfahren wird *Fragmentierung* genannt.

Unter Fragmentierung wird verstanden, dass das IP-Protokoll eines jeden Netzwerkknotens (sei es ein Router, ein Host o.ä.) in der Lage ist (sein sollte), empfangene Pakete gegebenenfalls zu zerteilen, um sie weiter über ein Teilnetz bis zum Zielhost zu übertragen. Jedes empfangende IP muss dazu in der Lage sein, diese Fragmente wieder zum ursprünglichen Paket zusammenzusetzen.

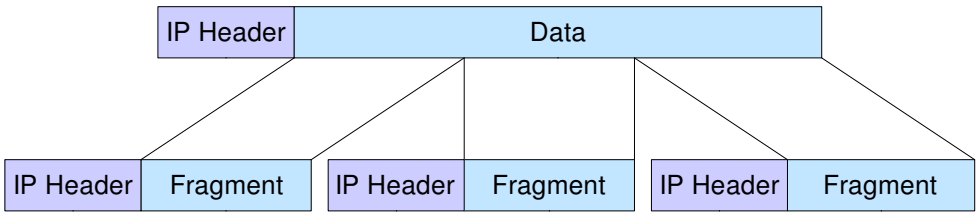


Abbildung 17 Fragementierung eines IP-Pakets.

Jedes Fragment eines zerteilten Pakets erhält einen eigenen, vollständigen IP-Header. Über das Identifikationsfeld im Header können alle Fragmente eines Pakets wiedererkannt werden. Die einzelnen Fragmente eines Pakets können durchaus unterschiedliche Wege auf dem Weg zum Zielhost nehmen. Die Lage der Daten eines Fragments innerhalb der Gesamtnachricht wird mit Hilfe des Fragment Offset-Feldes ermittelt.

3.3.4 Internet Control Message Protocol (ICMP)

Das *Internet Control Message Protocol (ICMP)* ist Bestandteil jeder IP-Implementierung und hat die Aufgabe Fehler- und Diagnoseinformationen für IP zu transportieren. ICMP ist im RFC 792 spezifiziert. Oft wird ICMP auch für Testzwecke verwendet, etwa um zu ermitteln, ob ein Host derzeit empfangsbereit ist.

Durch seine Vielseitigkeit bietet ICMP aber auch leider die Möglichkeit versteckte Nachrichten zu übermitteln. Ein Stichwort ist hier das so genannte „ICMP-Tunneling“. Beim ICMP-Tunneling wird das Datenfeld eines ICMP-Paketes genutzt, um Informationen zwischen Rechnern auszutauschen.

ICMP-Tunneling ist aber keine Technik, die es eventuellen Datenspionen ermöglicht, in einen Rechner oder ein Netz einzubrechen. Dennoch stellt das Tunneling eine Bedrohung für das Sicherheitskonzept eines Netzes dar. In [Schmidt1997a] ist ein ausführlicher Bericht über das ICMP-Tunneling zu finden.

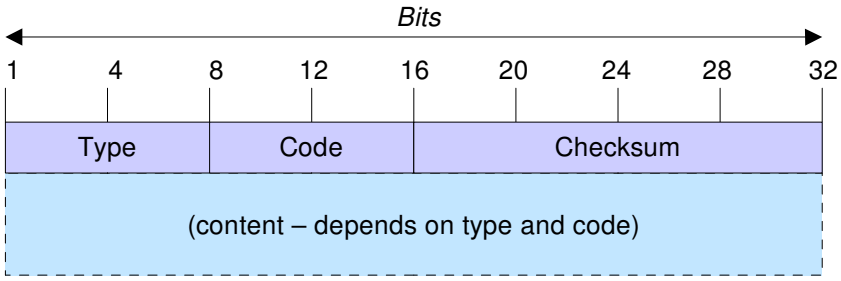


Abbildung 18 Der ICMP-Header (allgemeiner Aufbau).

ICMP hat sehr unterschiedliche Informationen zu transportieren. Deshalb ist nur der Grundaufbau des ICMP-Headers immer gleich, die Bedeutung der einzelnen Felder im Protokollkopf wechselt jedoch. Jeder ICMP-Nachrichtentyp wird in einem IP-Datengramm eingekapselt.

Die derzeit wichtigsten ICMP-Nachrichtentypen sind:

Destination Unreachable (Ziel nicht erreichbar):

Diese Nachricht wird verwendet, wenn:

- ein Netzwerk, Host, Protokoll oder Port nicht erreichbar ist,
- ein Paket nicht fragmentiert werden kann, weil das DF-Bit gesetz ist,
- die Source Route Option nicht erfolgreich ist.

Source Quench (Quelle löschen):

Wird ausgesendet, wenn ein Host zu viele Pakete verschickt, die aus Kapazitätsmangel nicht mehr verarbeitet werden können. Der sendende Host muss dann die Rate zum Aussenden von Nachrichten verringern.

Parameter Problem:

Verständigt den Absender eines Datengramms darüber, dass das Paket aufgrund einer fehlerhaften Angabe im IP-Header verworfen werden musste.

Redirect:

Wird ausgesendet, wenn ein Router feststellt, dass ein Paket falsch weitergeleitet wurde. Der sendende Host wird damit aufgefordert, die angegebene Route zu ändern.

Time Exceeded (Zeit verstrichen):

Diese Nachricht wird an den Absender eines Datengramms gesendet, dessen Lebensdauer den Wert 0 erreicht hat. Diese Nachricht ist ein Zeichen dafür, dass Pakete in einem Zyklus wandern, dass Netz überlastet ist oder die Lebensdauer für das Paket zu gering eingestellt wurde.

Echo Reply, Echo Request:

Mit diesen Nachrichten kann festgestellt werden, ob ein bestimmtes Ziel erreichbar ist. Ein Echo Request wird an einen Host gesendet und hat einen Echo Reply zur Folge (falls der Host erreicht wird).

Timestamp Request, Timestamp Reply:
Diese beiden Nachrichten sind ähnlich den zuvor beschriebenen Nachrichten, außer das die Ankunftszeit der Nachricht und die Sendezeit der Antwort mit erfasst werden. Mit diesen Nachrichtentypen kann die Netzleistung gemessen werden.

IP verwendet ICMP zum versenden von Fehler- und Diagnosemeldungen, während ICMP zur Übertragung seiner Nachrichten IP benutzt. Das bedeutet, wenn eine ICMP-Nachricht verschickt werden muss, wird ein IP-Datengramm erzeugt und die ICMP-Meldung in den Datenbereich des IP-Datengramms eingekapselt.

Das Datengramm wird dann wie üblich versendet. Eine ICMP-Nachricht wird immer als Antwort auf ein Datengramm verschickt. Entweder ist ein Datengramm auf ein Problem gestoßen, oder das Datengramm enthält eine ICMP-Anfrage, auf die eine Antwort versendet versendet werden muss. In beiden Fällen sendet ein Host oder Router eine ICMP-Nachricht an die Quelle des Datengramms zurück.

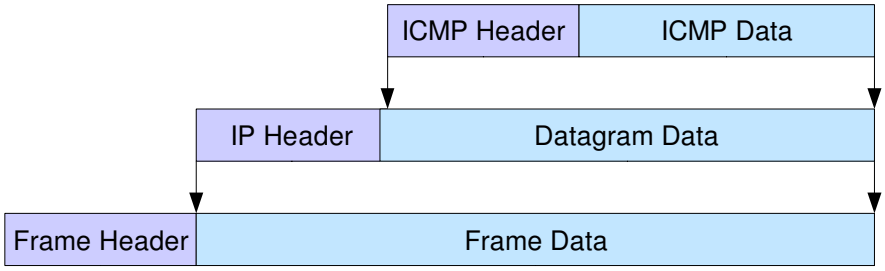


Abbildung 19 ICMP-Nachrichten-Einkapselung.

3.4 Transportschicht

Über der Internet-Schicht befindet sich die *Transportschicht (Host-to-Host-Transport Layer)*. Die beiden wichtigsten Protokolle der Transportschicht sind das *Transmission Control Protocol (TCP)* und das *User Datagram Protocol (UDP)*. Die Aufgabe von TCP besteht in der Bereitstellung eines sicheren und zuverlässigen Ende-zu-Ende-Transports von Daten durch ein Netzwerk. UDP ist im Gegensatz dazu ein verbindungsloses Transportprotokoll, das Anwendungen die Möglichkeit bietet, eingekapselte rohe IP-Pakete zu übertragen.

3.4.1 Transmission Control Protocol (TCP)

Das *Transmission Control Protocol (TCP)* ist ein *zuverlässiges*, *verbindungsorientiertes*, *Bytestrom* Protokoll. Die Hauptaufgabe von TCP besteht in der Bereitstellung eines sicheren Transports von Daten durch das Netzwerk. TCP ist im RFC 793 definiert. Diese Definitionen wurden im Laufe der Zeit von Fehlern und Inkonsistenzen befreit (RFC 1122) und um einige Anforderungen ergänzt (RFC 1323).

Im weiteren sollen nun die oben genannten Eigenschaften des Transmission Control Protocol - zuverlässig (reliable), verbindungsorientiert (connection-oriented), Bytestrom (byte-stream) - näher betrachtet werden.

Das Transmission Control Protocol stellt die *Zuverlässigkeit* der Datenübertragung mit einem Mechanismus, der als *Positive Acknowledgement with Re-Transmission (PAR)* bezeichnet wird, bereit. Dies bedeutet nichts anderes als das, dass das System, welches Daten sendet, die Übertragung der Daten solange wiederholt, bis vom Empfänger der Erhalt der Daten quittiert bzw. positiv bestätigt wird. Die Dateneinheiten, die zwischen den sendenden und empfangenden TCP-Einheiten ausgetauscht werden, heißen *Segmente*. Ein TCP-Segment besteht aus einem mindestens 20 Byte großen Protokollkopf (siehe den Abschnitt Der TCP-Header) und den zu übertragenden Daten. In jedem dieser Segmente ist eine Prüfsumme enthalten, anhand derer der Empfänger prüfen kann, ob die Daten fehlerfrei sind. Im Falle einer fehlerfreien Übertragung sendet der Empfänger eine Empfangsbestätigung an den Sender. Andernfalls wird das Datengramm verworfen und keine Empfangsbestätigung verschickt. Ist nach einer bestimmten Zeitperiode (timeout-period) beim Sender keine Empfangsbestätigung eingetroffen, verschickt der Sender das betreffende Segment erneut. Näheres zur Zeitüberwachung siehe [Santifaller1998].

TCP ist ein verbindungsorientiertes Protokoll. Verbindungen werden über ein *Dreiwege-Handshake (three-way handshake)* aufgebaut. Über das Dreiwege-Handshake werden Steuerinformationen ausgetauscht, die die logische *Ende-zu-Ende-Verbindung* etablieren. Zum Aufbau einer Verbindung sendet ein Host (Host 1) einem anderen Host (Host 2), mit dem er eine Verbindung aufbauen will, ein Segment, in dem das SYN-Flag (siehe Der TCP-Header, Flags) gesetzt ist. Mit diesem Segment teilt Host 1 Host 2 mit, das der Aufbau einer Verbindung gewünscht wird. Die Sequenznummer des von Host 1 gesendeten Segments gibt Host 2 außerdem an, welche Sequenznummer Host 1 zur Datenübertragung verwendet. Sequenznummern sind notwendig, um sicherzustellen, dass die Daten vom Sender in der richtigen Reihenfolge beim Empfänger ankommen. Der empfangende Host 2 kann die Verbindung nun annehmen oder ablehnen. Nimmt er die Verbindung an, wird ein Bestätigungssegment gesendet. In diesem Segment sind das SYN-Bit und das ACK-Bit (siehe Der TCP-Header, Flags) gesetzt. Im Feld für die Quittungsnummer bestätigt Host 2 die Sequenznummer von Host 1, dadurch, dass die um Eins erhöhte Sequenznummer von Host 1 gesendet wird. Die Sequenznummer des Bestätigungssegments von Host 2 an Host 1 informiert Host 1 darüber, mit welcher Sequenznummer beginnend Host 2 die Daten empfängt. Schlußendlich bestätigt Host 1 den Empfang des Bestätigungssegments von Host 2 mit einem Segment, in dem das ACK-Flag gesetzt ist und die um Eins erhöhte Sequenznummer von Host 2 im Quittungsnummernfeld eingetragen ist. Mit diesem Segment können auch gleichzeitig die ersten Daten an Host 2 übertragen werden. Nach dem Austausch dieser Informationen hat Host 1 die Bestätigung, dass Host 2 bereit ist Daten zu empfangen. Die Datenübertragung kann nun stattfinden. Eine TCP-Verbindung besteht immer aus genau zwei Endpunkten (Punkt-zu-Punkt-Verbindung).

Zum Beenden der Verbindung tauschen die beiden Host wiederum einen Dreiwege-Handshake aus, bei dem das FIN-Bit (siehe Der TCP-Header, Flags) zum Beenden der Verbindung gesetzt ist. Natürlich verläuft der Verbindungsaufbau nicht immer ohne Probleme. Eine Reihe interessanter Betrachtungen ist zu finden in [Tanenbaum1996].

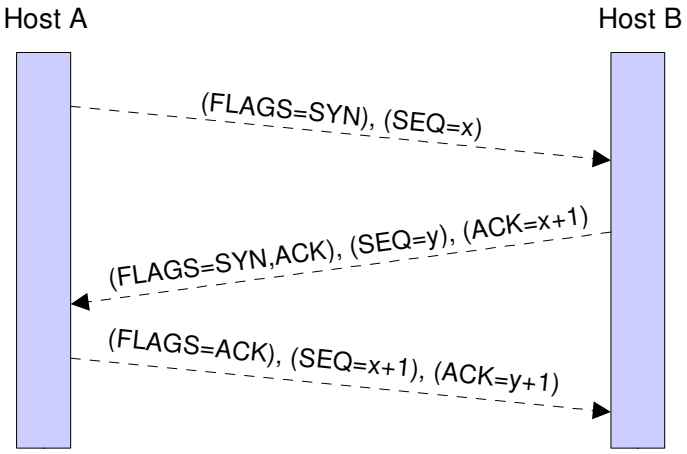


Abbildung 20 Dreiwege-Handshake (am Beispiel Verbindungsaufbau).

TCP nimmt *Datenströme* von Applikationen an und teilt diese in höchsten 64 KByte große Segmente auf (üblich sind ungefähr 1.500 Byte). Jedes dieser Segmente wird als IP-Datengramm verschickt. Kommen IP-Datengramme mit TCP-Daten bei einer Maschine an, werden diese an TCP weitergeleitet und wieder zu den ursprünglichen Bytesströmen zusammengesetzt. Die IP-Schicht gibt allerdings keine Gewähr dafür, dass die Datengramme richtig zugestellt werden. Es ist deshalb, wie oben bereits gesagt, die Aufgabe von TCP für eine erneute Übertragung der Daten zu sorgen. Es ist aber auch möglich, dass die IP-Datengramme zwar korrekt ankommen, aber in der falschen Reihenfolge sind. In diesem Fall muss TCP dafür sorgen, dass die Daten wieder in die richtige Reihenfolge gebracht werden. Hierfür verwendet TCP eine *Sequenznummer* und eine *Bestätigungsnummer* (siehe: TCP Sequence Number, Acknowledgement Number).

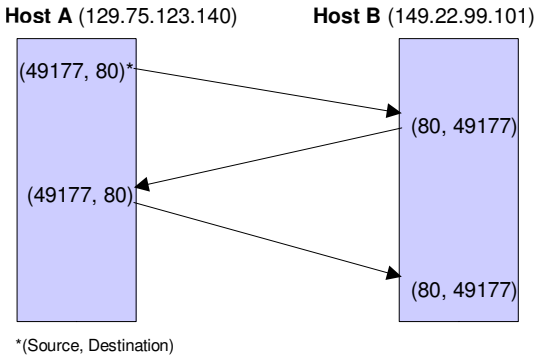
Portnummern

TCP ist außerdem dafür verantwortlich, die empfangenen Daten an die korrekte Applikation weiterzuleiten. Zur Adressierung der Anwendungen werden auf der Transportebene deshalb sogenannte *Portnummern (Kanalnummern)* verwendet. Portnummern sind 16 Bit groß; theoretisch kann ein Host somit bis zu 65.535 verschiedene TCP-Verbindungen aufbauen. Auch UDP verwendet Portnummern zur Adressierung. Portnummern sind nicht einzigartig zwischen den Transportprotokollen - die Transportprotokolle haben jeweils eigene Adressräume. Das bedeutet TCP und UDP können die gleichen Portnummern belegen. Das heißt, dass die Portnummer 53 in TCP nicht identisch mit der Portnummer 53 in UDP ist. Der Gültigkeitsbereich einer Portnummer ist auf einen Host beschränkt.

Eine IP-Adresse zusammen mit der Portnummer spezifiziert einen Kommunikationsendpunkt, einen so genannten *Socket* (ein Socket kann als 48-Bit Endpunkt betrachtet werden). Die Socketnummern von Quelle und Ziel identifizieren die Verbindung (socket1, socket2). Eine Verbindung ist durch die Angabe dieses Paares eindeutig identifiziert.

Möchte z.B. ein Host A (129.75.123.140) eine Verbindung zu einem entfernten Host B (149.22.99.101) aufnehmen, z.B. um den Inhalt einer Webseite anzuzeigen, so wird auf der TCP-Schicht als Zielport die Portnummer 80 für das Hypertext Transfer Protocol (http) angegeben. Host A, der den Dienst auf Port 80 von Host B in Anspruch nehmen möchte gibt als Quellport eine dynamische Portnummer (s.u.) aus dem Bereich 49.152 - 65.535 an, damit die von ihm gewünschten Daten von Host B an ihn zurückgeliefert werden können, z.B. 49.177. Damit ist die

Verbindung auf der TCP-Schicht über die Angabe von Quell- und Zielport eindeutig identifiziert. Zusammen mit den IP-Adressen bilden die Portnummern die beiden Sockets, die die Kommunikation zwischen Host A und Host B eindeutig kennzeichnen.



Bis 1992 waren Portnummern unter 256 für *gut bekannte Ports (well-known ports)* reserviert. Well-known Ports werden für Standarddienste, wie z.B telnet, ftp etc. genutzt. Ports zwischen 256 und 1023 wurden im allgemeinen für UNIX-spezifische Dienste (wie z.B. `rlogin`) benutzt. Ein Beispiel für den Unterschied zwischen einem Internet-weiten Dienst und einem UNIX-spezifischen Dienst ist der Unterschied zwischen Telnet und RLogin. Beide Dienste erlauben es, sich über das Netz auf einem entfernten Host einzuloggen. Telnet ist ein TCP/IP-Standard mit der Portnummer 23 und kann von so gut wie auf allen Betriebssystemen implementiert werden. RLogin ist im Gegensatz dazu ein UNIX-spezifischer Dienst, dessen Portnummer 53 ist.

Die Verwaltung der Portnummern ist nun von der *Internet Assigned Numbers Authority (IANA)* [http://www.iana.org] übernommen worden. Portnummern sind dabei in drei Bereiche aufgeteilt worden: *well-known ports*, *registered ports* und *dynamic ports*.

0 - 1023	Well-known ports (von der IANA verwaltet). Der Bereich der well-known ports ist bis 1023 erweitert worden, damit sind auch die UNIX-spezifischen Dienste als Standarddienste festgelegt.
1024 - 49151	Registered ports. Registrierte Ports dienen für Dienste, die üblicher Weise auf bestimmten Ports laufen. Ein Beispiel ist hier der Port 8080, der als "zweiter" bzw. alternativer Port für das http dient.
49152 - 65535	Dynamic and/or private ports. Dieser Bereich ist für die sogenannten dynamischen Ports festgelegt. Dynamische Ports dienen zur Kommunikation zwischen den beiden TCP-Schichten, die an einer Kommunikation beteiligt sind. Ein dynamischer Port wird nicht von bestimmten Standarddiensten belegt.

Auf UNIX-Systemen sind Portnummern in der Datei `/etc/services` definiert. Auszug aus der Datei `/etc/services` eines Linux-Systems:

```
heiko@phoenix:~> more /etc/services
#
# Network services, Internet style
#
# Note that it is presently the policy of IANA to assign a single well-
# known port number for both TCP and UDP; hence, most entries here have
# two entries even if the protocol doesn't support UDP operations.
# Updated from RFC 1340, ``Assigned Numbers'' (July 1992).  Not all ports
# are included, only the more common ones.
#
```

```
#       from: @(#)services      5.8 (Berkeley) 5/9/91
#       $Id: services,v 1.9 1993/11/08 19:49:15 cgd Exp $
#
tcpmux          1/tcp           # TCP port service multiplexer
echo            7/tcp
echo            7/udp
systat          11/tcp          users
daytime         13/tcp
daytime         13/udp
netstat         15/tcp
qotd            17/tcp          quote
msp             18/tcp          # message send protocol
msp             18/udp          # message send protocol
chargen         19/tcp          ttytst source
chargen         19/udp          ttytst source
ftp             21/tcp
# 22 - unassigned
telnet          23/tcp
# 24 - private
smtp            25/tcp          mail
# 26 - unassigned
time            37/tcp          timserver
time            37/udp          timserver
rlp             39/udp          resource        # resource location
nameserver      42/tcp          name            # IEN 116
whois           43/tcp          nicname
domain          53/tcp          nameserver      # name-domain server
domain          53/udp          nameserver
#
# UNIX specific services
#
exec            512/tcp
biff            512/udp         comsat
login           513/tcp
who             513/udp         whod
shell           514/tcp         cmd             # no passwords used
syslog          514/udp
printer         515/tcp         spooler         # line printer spooler
talk            517/udp
ntalk           518/udp
route           520/udp         router routed   # RIP
timed           525/udp         timeserver
```

Der TCP-Header

Die sendende und die empfangende TCP-Einheit tauschen Daten in Form von Segmenten aus. Ein Segment ist nichts anderes als die zu übertragenden Daten, versehen mit „Steuerinformationen“. Jedes Segment beginnt mit einem 20-Byte-Header, auf den Header-Optionen folgen können. Den Optionen folgen schließlich die zu übertragenden Daten. Die Segmentgröße wird durch zwei Faktoren begrenzt: erstens muss jedes Segment, einschließlich des TCP-Headers, in das Nutzdatenfeld des IP-Protokolls passen (65.535 Byte); zweitens hat jedes Netz eine *maximale Transfereinheit (MTU - Maximum Transfer Unit)*, in die das Segment passen muss. In der Regel ist die MTU einige tausend Byte groß und gibt die obere Grenze der Segmentgröße vor (z.B. Ethernet 1.500 Bytes). Läuft ein Segment durch eine Anzahl von Netzen und trifft dabei auf ein Netz mit einer kleineren MTU, so muss das Segment vom Router in kleinere Segmente aufgeteilt (*fragmentiert*) werden. Unabhängig von der Größe der MTU können dem TCP-Header und den

Optionen maximal 65.535-20-20 = 65.495 Datenbyte folgen (die ersten 20 Byte beziehen sich auf den IP-Header, die zweiten auf den TCP-Header; die Länge der Optionen wird mit zu den Datenbytes gezählt). TCP-Segmente ohne Daten sind zulässig und dienen der Übermittlung von Bestätigungen und Steuernachrichten.

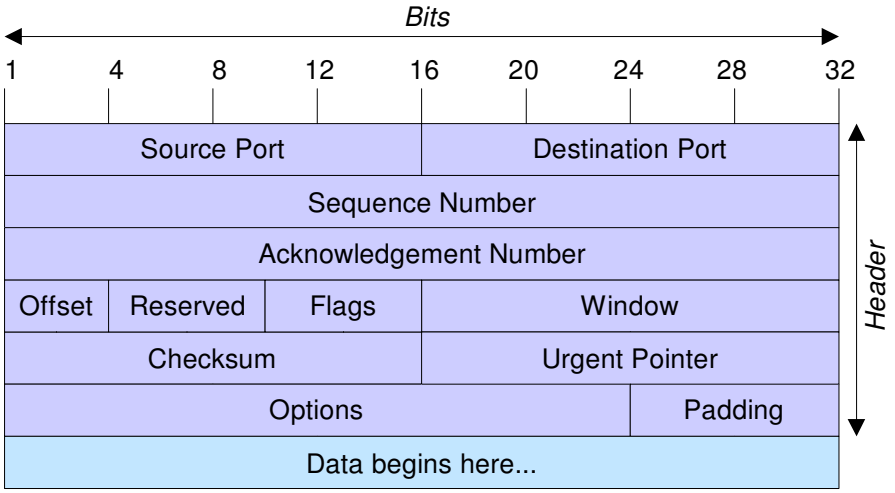


Abbildung 21 Der TCP-Header.

Die Felder des TCP-Headers haben die folgende Bedeutung:

Source-/Destination-Port:

Die Felder *Source Port (Quellport)* und *Destination Port (Zielport)* adressieren die Endpunkte der Verbindung. Die Größe für die beiden Felder beträgt 16 Bit (siehe auch den Abschnitt Portnummern).

Sequence Number, Acknowledgement Number:

Die *Sequenznummer* und die *Bestätigungsnummer* sind jeweils 32-Bit-Zahlen. Die Nummern geben die Stellung der Daten des Segments innerhalb des in der Verbindung ausgetauschten Datenstroms an. Die Sequenznummer gilt in Senderichtung, die Bestätigungsnummer für Empfangsquittungen. Jeder der beiden TCP-Verbindungspartner generiert beim Verbindungsaufbau eine Sequenznummer, die sich während des Zeitraums der Verbindung **nicht** wiederholen darf. Dies ist allerdings durch den großen Zahlenraum von 2^{32} wohl ausreichend gesichert (RFC1323 stellt allerdings dar, dass die Größe der Sequenznummer bei schnellen Netztechnologien zu einem Problem werden kann, so liegt die Durchlaufzeit bei 10-Mbit/s-Ethernet durch alle Folgenummern noch bei 57 Minuten, während die Durchlaufzeit bei 1-Gbit/s-Ethernet bereits nur noch 34 Sekunden beträgt). Diese Nummern werden beim Verbindungsaufbau ausgetauscht und gegenseitig quittiert. Bei der Datenübertragung wird die Sequenznummer vom Absender jeweils um die Anzahl der bereits gesendeten Bytes erhöht. Mit der Quittungsnummer gibt der Empfänger an, bis zu welchem Byte er die Daten bereits korrekt empfangen hat. Die Nummer gibt allerdings nicht an, welches Byte zuletzt korrekt empfangen wurde, sondern welches Byte als nächstes zu erwarten ist.

Offset:

Das Feld *Offset* (oder auch *Header Length*) gibt die Länge des TCP-Headers in 32-Bit Worten an. Dies entspricht dem Anfang der Daten im TCP-Segment. Das Feld ist notwendig, da der

Header durch das Optionsfeld eine variable Länge hat.

Flags:

Mit den sechs 1-Bit-Flags im *Flags*-Feld werden bestimmte Aktionen im TCP-Protokoll aktiviert:

URG

Wird das Flag *URG* auf 1 gesetzt, so bedeutet dies, dass der *Urgent Pointer (Dringendzeiger)* verwendet wird.

ACK

Das *ACK-Bit* wird gesetzt, um anzugeben, dass die Bestätigungsnummer im Feld *Acknowledgement Number* gültig ist. Ist das Bit auf 0 gesetzt, enthält das TCP-Segment keine Bestätigung, das Feld Acknowledgement Number wird ignoriert.

PSH

Ist das PSH-Bit gesetzt, so werden die Daten in dem entsprechenden Segment sofort bei Ankunft der adressierten Anwendung bereitgestellt ohne sie zu puffern.

RST

Das *RST-Bit* dient dazu eine Verbindung zurückzusetzen, falls ein Fehler bei Übertragung aufgetreten ist. Dies kann sowohl der Fall sein, wenn ein ungültiges Segment übertragen wurde, ein Host abgestürzt ist oder der Versuch eines Verbindungsaufbaus abgewiesen werden soll.

SYN

Das *SYN-Flag (Synchronize Sequenze Numbers)* wird verwendet, um Verbindungen aufzubauen. Zusammen mit der Acknowledgement Number und dem ACK-Bit wird die Verbindung im Form eines *Dreiwege-Handshake* aufgebaut (siehe oben).

FIN

Das *FIN-Bit* dient zum Beenden einer Verbindung. Ist das Bit gesetzt, gibt dies an, dass der Sender keine weiteren Daten zu Übertragen hat. Das Segment mit gesetztem FIN-Bit muss quittiert werden.

Window:

Das Feld *Fenstergröße* enthält die Anzahl Bytes, die der Empfänger ab dem bereits bestätigten Byte empfangen kann. Mit der Angabe der Fenstergröße erfolgt in TCP die Flußsteuerung. Das TCP-Protokoll arbeitet nach dem Prinzip eines *Schiebefensters mit variabler Größe (Sliding Window)*. Jede Seite einer Verbindung darf die Anzahl Bytes senden, die im Feld für die Fenstergröße angegeben ist, ohne auf eine Quittung von der Empfängerseite zu warten. Währen des Sendens können gleichzeitug Quittungen für die von der anderen Seite empfangenen Daten eintreffen (diese Quittungen können wiederum neue Fenstergrößen einstellen). Eine Fenstergröße von 0 besagt, dass die Bytes bis einschließlich der Acknowledgement Number minus Eins empfangen wurden, der Empfänger momentan aber keine weiteren Daten empfangen kann. Die Erlaubnis zum weiteren Senden von Daten erfolgt durch das versenden eines Segments mit gleicher Bestätigungsnummer und einer Fenstergröße ungleich Null.

Checksum:

Die *Prüfsumme* prüft den Protokollkopf, die Daten und den *Pseudo-Header* (siehe Abbildung 20).

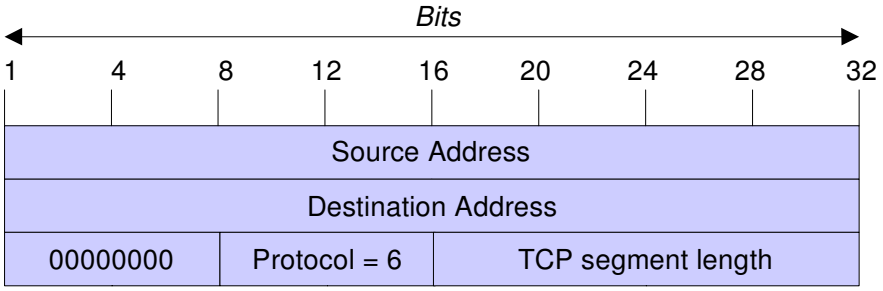


Abbildung 22 Der Pseudo-Header in der Prüfsumme.

Der Algorithmus für die Bildung der Prüfsumme ist einfach: alle 16-Bit Wörter werden im 1er-Komplement addiert und die Summe ermittelt. Bei der Berechnung ist das Feld Checksum auf Null gesetzt und das Datenfeld wird bei ungerader Länge um ein Nullbyte aufgefüllt. Führt der Empfänger des Segments die Berechnung auf das gesamte Segment aus - inklusive des Feldes für die Prüfsumme - sollte das Ergebnis 0 sein [Tanenbaum1996]. Der Pseudo-Header enthält die 32-bit großen IP-Adressen der Quell- und Zielmaschine sowie die Protokollnummer (für TCP 6) und die Länge des TCP-Segments. Die Einbeziehung der Felder des Pseudo-Headers in die Prüfsummenberechnung hilft, durch IP falsch zugeteilte Pakete zu erkennen. Die Verwendung von IP-Adressen auf der Transportebene stellt allerdings eine Verletzung der Protokollhierarchie dar.

Urgent Pointer:

Der *Urgent-Zeiger* ergibt zusammen mit der Sequenznummer einen Zeiger auf ein Datenbyte. Dies entspricht einem Byteversatz zu einer Stelle, an der dringende Daten vorgefunden werden. TCP signalisiert damit, dass sich an einer bestimmten Stelle im Datenstrom wichtige Daten befinden, die sofort gelesen werden sollten. Das Feld wird nur gelesen, wenn auch das Urgent-Flag (siehe oben) gesetzt ist.

Options:

Das *Options-Feld* soll eine Möglichkeit bieten Funktionen bereitzustellen, die im normalen TCP-Protokollkopf nicht vorgesehen sind. In TCP sind drei Optionen definiert: *End of Option List, No-Operation und Maximum Segment Size*. Die wichtigste dieser drei Optionen ist die Maximale Segmentgröße. Mit dieser Option kann ein Host die maximale Anzahl Nutzdaten übermitteln, die er annehmen will bzw. annehmen kann. Während eines Verbindungsaufbaus kann jede Seite ihr Maximum an Nutzdaten übermitteln, die kleinere der beiden Zahlen wird als maximale Nutzdatengröße für die Übertragung übernommen. Wird diese Option von einem Host nicht unterstützt wird als Standard die Vorgabe von 536 Byte verwendet.

Padding:

Das Feld *Padding* wird verwendet, um sicherzustellen, dass der Header an einer 32-Bit Grenze endet und die Daten an einer 32-Bit Grenze beginnen. Das Füllfeld wird mit Nullen gefüllt.

3.4.2 User Datagram Protocol (UDP)

Das *User Datagram Protocol (UDP)* ist im RFC 768 definiert. UDP ist ein unzuverlässiges, verbindungsloses Protokoll. Wie zuvor schon gesagt, bedeutet unzuverlässig in diesem Zusammenhang nicht, dass die Daten evtl. fehlerhaft beim Zielrechner ankommen, sondern, dass das Protokoll keinerlei Mechanismen zur Verfügung stellt, die sichern, dass die Daten auch tatsächlich beim Zielrechner ankommen. Sind die Daten aber beim Zielrechner angekommen, so sind sie auch korrekt. UDP bietet gegenüber TCP den Vorteil eines geringen Protokoll-Overheads. Viele Anwendungen, bei denen nur eine geringen Anzahl von Daten übertragen wird (z.B. Client/Server-Anwendungen, die auf der Grundlage einer Anfrage und einer Antwort laufen), verwenden UDP als Transportprotokoll, da unter Umständen der Aufwand zur Herstellung einer Verbindung und einer zuverlässigen Datenübermittlung größer ist als die wiederholte Übertragung der Daten.

Ein UDP-Segment besteht aus einem Header von 8 Byte, gefolgt von den Daten.

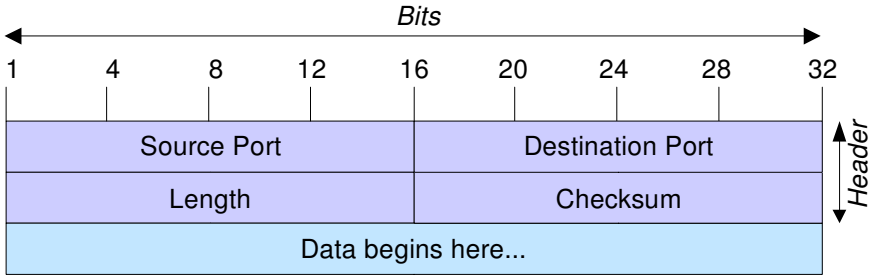


Abbildung 23 Der UDP-Header.

Die Sender- und Empfänger-Portnummern erfüllen den gleichen Zweck wie beim Transmission Control Protocol. Sie identifizieren die Endpunkte der Quell- und Zielmaschine. Das Feld für die Länge enthält die Länge des gesamten Datengramms, inklusive der Länge des Protokollkopfes. Die Prüfsumme enthält die Internet-Prüfsumme der UDP-Daten, des Protokollkopfs und des Pseudo-Headers. Das Prüfsummenfeld ist optional. Enthält das Feld eine 0, wurde vom Absender keine Prüfsumme eingetragen und somit findet beim Empfänger keine Überprüfung statt.

Das User Datagram Protocol liefert über die Leistungen des Internet Protokolls hinaus nur Portnummern für die Adressierung der Kommunikationsendpunkte und eine optionale Prüfsumme. Das Protokoll beinhaltet keine Transportquittungen oder andere Mechanismen für die Bereitstellung einer zuverlässigen Ende-zu-Ende-Verbindung. Hierdurch wird UDP allerdings sehr effizient und eignet sich somit besonders für Anwendungen, bei denen es in erster Linie auf die Geschwindigkeit der Datenübertragung ankommt (z.B. verteilte Dateisysteme wie NFS).

3.5 Applikationsschicht

Die oberste Schicht des TCP/IP-Modells ist die Applikationsschicht. Diese Schicht bietet eine Reihe standardisierter Anwendungsprotokolle, auf die eine Vielzahl von Anwendungsprogrammen aufsetzen. An dieser Stelle seien nur einige Protokolle, die auf der Anwendungsschicht angeboten werden, genannt:

TELNET:

> *TELNET* ist das Protokoll für virtuelle Terminals. Es dient dazu, Zugriff auf einen am Netz angeschlossenen Rechner in Form einer Terminalsitzung (auch *remote login* genannt) zu

liefern. Der TELNET-Dienst benutzt den TCP-Port 23. TELNET ist im RFC 854 spezifiziert.

FTP:

Mit dem *File Transfer Protocol - FTP* lassen sich Dateien externer Rechner übertragen, löschen, ändern oder umbenennen. FTP ist im RFC 959 definiert. Von FTP werden die Ports 20 und 21 benutzt. Port 21 wird als Kommandokanal verwendet und Port 20 dient als Datenkanal.

SMTP:

Das *Simple Mail Transfer Protocol - SMTP* ist das Protokoll für die elektronische Post im Internet. Das Übertragungsprotokoll für elektronische Post ist im RFC 821 und das Nachrichtenformat im RFC 822 spezifiziert.

DNS:

Der *Domain Name Service - DNS* dient dazu ASCII-Zeichenketten in Internet-Adressen und umgekehrt zu wandeln. DNS ist ein hierarchisches Benennungssystem, das auf Domänen basiert und ein verteiltes Datenbanksystem zur Implementierung des Benennungsschemas. Es wird im wesentlichen dazu benutzt Hostnamen und E-Mailadressen (mit denen Menschen nun einmal besser umgehen können) in IP-Adressen umzuwandeln. DNS ist in den RFCs 1034 und 1035 definiert.

NFS:

Mit dem *Network File System - NFS* lassen sich mehrere Rechner auf transparente Weise miteinander verbinden. Der NFS-Dienst stellt eine virtuelle Verbindung von Laufwerken und Festplatten her, so daß sich entfernte Dateisysteme als Erweiterung des eigenen lokalen Dateisystems darstellen.

4 IP Version 6

Es wird wohl noch etwas länger dauern, bis dieser Abschnitt der Arbeit endlich fertig ist, da ich im Moment ziemlich viele andere Dinge zu tun habe. Für alle, die aber jetzt schon mehr wissen wollen gebe ich an dieser Stelle einen kurzen Hinweis auf Literatur (weitere Quellen sind in der Literaturliste zu finden):

- Hinden R.: IP Next Generation (IPng). http://playground.sun.com/pub/ipng/html/ipng-main.html.
- Huitema, C.: IPv6 - die neue Generation, Architektur und Implementierung. Addison-Wesley, München, 2000.

4.1 Die Zukunft

Das rasche (exponentielle) Wachstum des Internet zwingt dazu, das Internet Protokoll in der Version 4 (IPv4) zu ändern oder durch ein Nachfolgeprotokoll zu ersetzen .

Bis zu Beginn 90er Jahre wurde das Internet größtenteils nur von Universitäten, Regierungsbehörden (dies aber auch fast nur in den USA und hier vor allem vom Verteidigungsministerium) und einigen Firmen aus der Industrie genutzt. Seit der Einführung des *World Wide Web (WWW)* ist das Internet aber auch zunehmend für Privatpersonen, kleinere Firmen etc. interessant. Das Internet wandelt sich von einem "Spielplatz für Akademiker" zu einem weltweiten Informations- und Unterhaltungssystem. Mit der ständig steigenden Anzahl von Benutzern des Internet werden sich auch die Anforderungen an das Netz ändern bzw. haben sich bereits geändert. Genannt sei hier nur als Beispiel das angestrebte Zusammenwachsen der Computer-, Unterhaltungs- und Telekommunikationsbranchen. Den Anforderungen, die z.B. *Video-on-demand* stellt, ist das Internet bzw. das Internet Protokoll in der Version 4 nicht gewachsen.

Vinton Cerf (der „Vater“ des Internet) bezeichnet in einem Interview mit der Zeitschrift c't [Krempl1998] das Internet *"(...) als die wichtigste Infrastruktur für alle Arten von Kommunikation."*. Auf die Frage, wie man sich die neuen Kommunikationsdienste des Internet vorstellen könne, antwortete Cerf:

> *"Am spannendsten finde ich es, die ganzen Haushaltsgeräte ans Netz anzuschließen. Ich denke dabei nicht nur daran, dass der Kühlschrank sich in Zukunft mit der Heizung austauscht, ob es in der Küche zu warm ist. Stromgesellschaften könnten beispielsweise Geräte wie Geschirrspülmaschinen kontrollieren und ihnen Strom genau dann zur Verfügung stellen, wenn gerade keine Spitzennachfrage herrscht. Derartige Anwendungen hängen allerdings davon ab, dass sie zu einem erschwinglichen Preis angeboten werden. Das ist nicht unbedingt ferne Zukunftsmusik; die Programmierer müßten eigentlich nur damit anfangen, endlich Software für intelligente Netzwerkanwendungen zu schreiben. Und natürlich muß die Sicherheit derartiger Systeme garantiert sein. Schließlich möchte ich nicht, dass die Nachbarkinder mein Haus programmieren!"*

Auf die Internet Protokolle kommen in der nächsten Zeit also völlig neue Anforderungen zu. Wie versucht wird, diese Anforderungen zu erfüllen, wird in den nächsten Abschnitten beschrieben.

4.2 Classless InterDomain Routing (CIDR)

Der Verknappung der Internet-Adressen durch die ständig steigende Benutzerzahl wird zunächst versucht, mit dem *Classless InterDomain Routing (CIDR)* entgegen zu wirken.

Durch die Vergabe von Internet-Adressen in Klassen (Netze der Klassen A,B,C,...) wird eine große Anzahl von Adressen verschwendet. Hierbei stellt sich vor allem die Klasse B als Problem dar. Viele Firmen nehmen ein Netz der Klasse B für sich in Anspruch, da ein Klasse A Netz mit bis zu 16 Mio. Hosts selbst für eine sehr große Firma überdimensioniert scheint. Tatsächlich ist aber oft auch ein Klasse B Netz zu groß. Für viele Firmen würde ein Netz der Klasse C ausreichen. Dies wurde aber nicht verlangt, da viele Unternehmen die Befürchtung hatten, dass ein Klasse C Netz mit seinen bis zu 254 möglichen Hosts nicht ausreichen würde.

Ein größeres Hostfeld für Netze der Klasse C (z.B. 10 Bit, das entspricht 1022 Host pro Netz) hätte das Problem der knapper werdenden IP-Adressen vermutlich gemildert. Ein anderes Problem wäre dadurch allerdings entstanden: die Einträge der Routing-Tabellen wären explodiert.

Ein anderes Konzept ist das Classless InterDomain Routing (RFC 1519): die verbleibenden Netze der Klasse C werden in Blöcken variabler Größe zugewiesen. Werden beispielsweise 2000 Adressen benötigt, so können einfach acht aufeinander folgende Netze der Klasse C vergeben werden; das entspricht einem Block von 2048 Adressen. Zusätzlich werden die verbliebenen Klasse C Adressen restriktiver und strukturierter vergeben (RFC 1519). Die Welt ist dabei in vier Zonen, von denen jede einen Teil des verbliebenen Klasse C Adressraums erhält, aufgeteilt:

194.0.0.0 - 195.255.255.255	Europa
198.0.0.0 - 199.255.255.255	Nordamerika
200.0.0.0 - 201.255.255.255	Mittel- und Südamerika
202.0.0.0 - 203.255.255.255	Asien und pazifischer Raum
204.0.0.0 - 223.255.255.255	Reserviert für zukünftige Nutzung

Jede der Zonen erhält dadurch in etwa 32 Millionen Adressen zugewiesen. Vorteil bei diesem Vorgehen ist, dass die 32 Millionen Adressen einer Region im Prinzip zu einem Eintrag in den Routing-Tabellen komprimiert worden sind. Der Vorteil der dadurch entsteht ist, dass z.B. jeder Router, der eine Adresse außerhalb seiner Region zugesandt bekommt...

4.3 Internet Protokoll Version 6 (IPv6)

Der vorrangige Grund für eine Änderung des IP-Protokolls ist auf den begrenzten Adreßraum zurückzuführen. CIDR schafft hier zwar wieder etwas Luft, dennoch ist klar absehbar, dass auch diese Maßnahmen nicht ausreichen, um die Verknappung der Adressen für eine längere Zeit in den Griff zu bekommen.

Weitere Gründe für eine Änderung des IP-Protokolls sind die oben schon erwähnten neuen Anforderungen an das Internet. Diesen Anforderungen ist IP in der Version 4 nicht gewachsen. Die *IETF (Internet Engineering Task Force)* begann deshalb 1990 mit der Arbeit an einer neuen Version von IP. Die wesentlichen Ziele des Projekts sind [Tanenbaum1996]:

- Unterstützung von Milliarden von Hosts, auch bei ineffizienter Nutzung des Adressraums

- Reduzierung des Umfangs der Routing-Tabellen
- Vereinfachung des Protokolls, damit Router Pakete schneller abwickeln können
- Höhere Sicherheit (Authentifikation und Datenschutz) als das heutige IP
- Mehr Gewicht auf Dienstarten, insbesondere für Echtzeitanwendungen
- Unterstützung von Multicasting durch die Möglichkeit, den Umfang zu definieren
- Möglichkeit für Hosts, ohne Adressänderung auf Reise zu gehen
- Möglichkeit für das Protokoll, sich zukünftig weiterzuentwickeln
- Unterstützung der alten und neuen Protokolle in Koexistenz für Jahre

Im Dezember 1993 forderte die IETF mit RFC 1550 [IP: Next Generation (IPnG) White Paper Solicitation, Dec. 1993] die Internet-Gemeinde dazu auf, Vorschläge für ein neues Internet Protokoll zu machen. Auf die Anfrage wurde eine Vielzahl von Vorschlägen eingereicht. Diese reichten von nur geringfügigen Änderungen am bestehenden IPv4 bis zur vollständigen Ablösung durch ein neues Protokoll. Die drei besten Vorschläge wurden im *IEEE Network Magazine* veröffentlicht ([Deering1993], [Francis1993], [KatzFord1993]). Aus diesen Vorschlägen wurde von der IETF das *Simple Internet Protocol Plus (SIPP)* als Grundlage für die neue IP-Version ausgewählt. SIPP ist eine Kombination aus den Vorschlägen von Deering [Deering1993] und Francis [Francis1993].

Als die Entwickler mit den Arbeiten an der neuen Version des Internet Protokolls begannen, wurde natürlich auch ein Name für das Projekt bzw. das neue Protokoll benötigt. Wohl angeregt durch eine gleichnamige Fernsehsendung, wurde als Arbeitsname *IP - Next Generation (IPnG)* gewählt. Schließlich bekam das neue IP eine offizielle Versionsnummer zugewiesen: IP Version 6 oder kurz IPv6. Die Protokollnummer 5 (IPv5) wurde bereits für ein experimentelles Protokoll verwendet.

Die folgende Beschreibung von IPv6 orientiert sich an RFC 2460 [Internet Protocol, Version 6 (IPv6) Specification, Dec. 1998]. Dieses Dokument gibt den neuesten Stand der Spezifikation des Internet Protokolls in der Version 6 wieder. RFC 2460 enthält einige wesentliche Änderungen der Spezifikation gegenüber RFC 1883 [Internet Protocol, Version 6 (IPv6) Specification, Dec. 1995].

4.3.1 Die Merkmale von IPv6

Viele der als erfolgreich betrachteten Merkmale von IPv4 bleiben in IPv6 voll erhalten. Trotzdem ist IPv6 im allgemeinen nicht mit IPv4 kompatibel, wohl aber zu den weiteren Internet-Protokollen, insbesondere den Protokollen der Transportschicht (TCP, UDP); eventuell nach geringfügigen Modifikationen. Die Modifikationen betreffen im wesentlichen die erweiterte Adressgröße (bisher 32 Bit auf nun 128 Bit).

Die wesentlichen Merkmale von IPv6 sind:

- **Adressgröße:** Als wichtigstes Merkmal hat IPv6 gegenüber IPv4 größere Adressen. Statt bisher 32 Bit stehen nun 128 Bit für die Adressen bereit. Theoretisch lassen sich damit $2^{128} = 3.4*10^{38}$ Adressen vergeben.
- **Header-Format:** Der IPv6 (Basis)Header wurde vollständig geändert. Der Header enthält nur 7 statt bisher 13 Felder. Diese Änderung ermöglicht Routern, Pakete schneller zu verarbeiten. Im Gegensatz zu IPv4 gibt es bei IPv6 nicht mehr nur einen Header, sondern mehrere Header. Ein Datengramm besteht aus einem Basis-Header, sowie einem oder mehreren Zusatz-Headern, gefolgt von den Nutzdaten.
- **Erweiterte Unterstützung von Optionen und Erweiterungen:** Die Erweiterung der

Optionen ist notwendig geworden, da einige, bei IPv4 notwendige Felder nun optional sind. Darüber hinaus unterscheidet sich auch die Art, wie die Optionen dargestellt werden. Für Router wird es damit einfacher, Optionen, die nicht für sie bestimmt sind, zu überspringen. Dies ermöglicht ebenfalls eine schnellere Verarbeitung von Paketen.

- **Dienstarten:** IPv6 legt mehr Gewicht auf die Unterstützung von Dienstarten. Damit kommt IPv6 den Forderungen nach einer verbesserten Unterstützung der Übertragung von Video- und Audiodaten entgegen. IPv6 bietet hierzu eine Option zur Echtzeitübertragung.
- **Sicherheit:** IPv6 beinhaltet nun im Protokoll selbst Mechanismen zur sicheren Datenübertragung. Wichtige neue Merkmale von IPv6 sind hier Authentifikation (authentication), Datenintegrität (data integrity) und Datenverlässlichkeit (data confidentiality).
- **Erweiterbarkeit:** IPv6 ist ein erweiterbares Protokoll. Bei der Spezifikation des Protokolls wurde nicht versucht alle potentiell möglichen Einsatzfelder für das Protokoll in die Spezifikation zu integrieren. Vielmehr bietet IPv6 die Möglichkeit über IPv6-Erweiterungs-Header das Protokoll zu erweitern. Damit ist das Protokoll offen für zukünftige Verbesserungen.

4.3.2 Das IPv6 Datengrammformat

Ein IPv6-Datengramm besteht aus dem *Basis-Header*, gefolgt von den optionalen IPv6-Erweiterungs-Headern und den Nutzdaten.

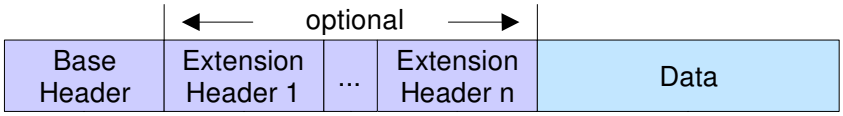


Abbildung 24 Allgemeine Form eines IPv6-Datengramms.

4.3.3 Der IPv6-Basis-Header

Der IPv6-Basis-Header ist doppelt so groß wie der IPv4-Header. Der IPv6-Basis-Header enthält weniger Felder als der IPv4-Header, dafür ist aber die Adressgröße für die Quell- und Zieladresse von bisher 32-Bit auf nunmehr 128-Bit erweitert worden.

Version

Mit dem Feld *Version* können Router überprüfen, um welche Version des Protokolls es sich handelt. Für ein IPv6-Datengramm ist dieses Feld immer 6 und für ein IPv4-Datengramm dementsprechend immer 4. Mit diesem Feld ist es möglich für eine lange Zeit die unterschiedlichen Protokollversionen IPv4 und IPv6 nebeneinander zu verwenden. Über die Prüfung des Feldes Version können die Daten an das jeweils richtige „Verarbeitungsprogramm“ weitergeleitet werden.

Priority:

Das Feld *Priority* (oder *Traffic Class*) ...

Flow Label

Das Feld *Flow Label*...

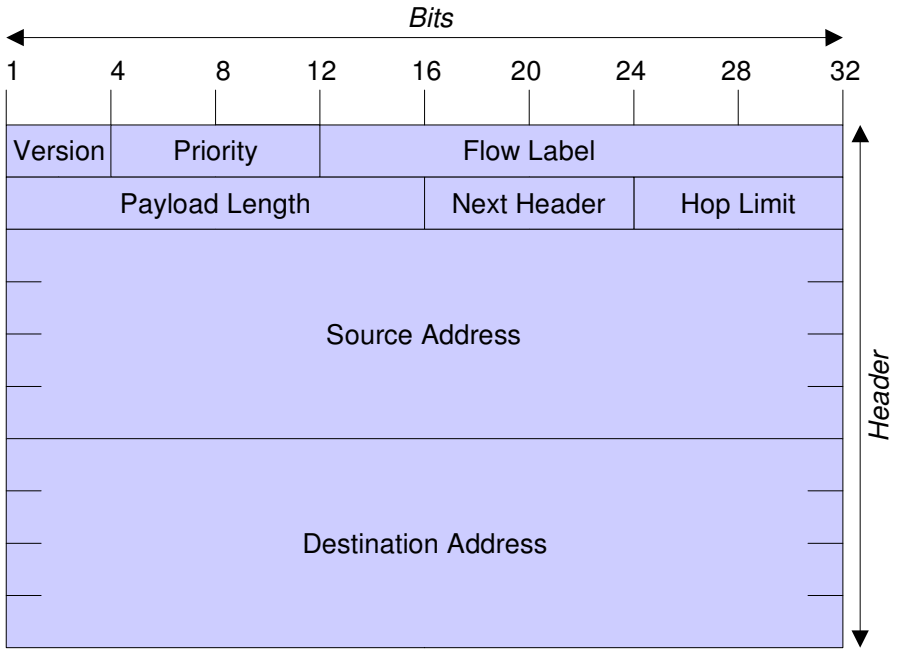


Abbildung 25 IPv6-Basis-Header.

Payload Length

Das Feld *Payload Length (Nutzdatenlänge)* gibt an, wie viele Bytes dem IPv6-Basis-Header folgen, der IPv6-Basis-Header ist ausgeschlossen. Die Erweiterungs-Header werden bei der Berechnung der Nutzdatenlänge mit einbezogen. Das entsprechende Feld wird in der Protokollversion 4 mit *Total Length* bezeichnet. Allerdings bezieht IPv4 den 20 Byte großen Header auch mit in die Berechnung ein, wodurch die Bezeichnung „total length“ gerechtfertigt ist.

Next Header

Das Feld *Next Header* gibt an, welcher Erweiterungs-Header dem IPv6-Basis-Header folgt. Jeder folgende Erweiterungs-Header beinhaltet ebenfalls ein Feld Next Header, das auf den nachfolgenden Header verweist. Ist dies der letzte zu IPv6 zugehörige Header, so gibt das Feld an, welches Transportprotokoll (z.B. TCP oder UDP) folgt. Eine genauere Beschreibung des Konzepts mehrerer Header folgt im Abschnitt IPv6-Erweiterungs-Header

Hop Limit

Mit dem Feld *Hop Limit* wird festgelegt, wie lange ein Paket überleben darf. Der Wert des Feldes wird nach jeder Teilstrecke gesenkt. Ein Datengramm wird dann verworfen, wenn das Feld Hop Limit auf Null herunter gezählt ist, bevor das Datengramm sein Ziel erreicht hat. IPv4 verwendet hierzu das Feld *Time to Live*, welches die Zeit in Sekunden angibt, die ein Paket überleben darf. Allerdings wird dieses Feld von den meisten Routern nicht so interpretiert. In IPv6 wurde das Feld deshalb umbenannt, um die tatsächliche Nutzung wiederzugeben.

Source Address, Destination Address

Die beiden Felder *Quell-* und *Zieladresse* dienen zur Identifizierung des Senders und Empfängers eines IP-Datengramms. IPv6 verwendet zur Adressierung 4 mal so große Adressen wie IPv4: 128 Bit statt 32 Bit. Eine genaue Beschreibung der IPv6-Adressen folgt

im Abschnitt IPv6-Adressierung.

Ein Vergleich des IPv4-Headers mit dem IPv6-Basis-Header veranschaulicht, welche Felder bei IPv6 weggelassen wurden:

- Das Feld *Length (Internet Header Length - IHL)* ist nicht mehr vorhanden, da der IPv6-Basis-Header eine feste Länge von 40 Byte hat. Bei IPv4 ist dieses Feld notwendig, da der Header aufgrund der Optionen eine variable Länge hat.
- Das Feld *Protocol* wird nicht mehr benötigt, da das Feld *Next Header* angibt, was nach dem letzten IP-Header folgt (z.B. TCP oder UDP).
- Alle Felder die bisher zur Fragmentierung eines IP-Datengramms benötigt wurden (*Identification, Flags, Fragment Offset*), sind im IPv6-Basis-Header nicht mehr vorhanden, da die Fragmentierung in IPv6 gegenüber IPv4 anders gehandhabt wird. Alle IPv6 kompatiblen Hosts und Router müssen Pakete mit einer Größe von 1280 Byte (RFC 1883 legte diese Größe noch auf 576 Byte fest) unterstützen. Durch diese Regel wird eine Fragmentierung im Prinzip nicht notwendig. Empfängt ein Router ein zu großes Paket, so führt er keine Fragmentierung mehr durch, sondern sendet eine Nachricht an den Absender des Pakets zurück. In dieser Nachricht wird der sendende Host angewiesen, alle weiteren Pakete zu diesem Ziel aufzuteilen. Das bedeutet, dass von den Hosts „erwartet“ wird, dass sie von vornherein eine Datengrammgröße wählen, die keine Fragmentierung voraussetzt. Dadurch wird eine größere Effizienz bei der Übertragung erreicht, als wenn Pakete von Routern auf dem Weg fragmentiert werden müssen. Die Steuerung der Fragmentierung erfolgt bei IPv6 über den *Fragment Header*.
- Das Feld *Checksum* ist nicht mehr vorhanden, da die Berechnung der Prüfsumme sich nachteilig auf die Leistung der Datenübertragung ausgewirkt hat. Das entfernen der Prüfsumme aus dem Internet Protokoll hat zu heftigen Diskussionen geführt [Tanenbaum1996]. Die eine Seite kritisierte heftig das entfernen der Prüfsumme, während die andere Seite argumentierte, dass Prüfsummen etwas sind, das auch von Anwendungen übernommen werden kann, sofern sich die Anwendung tatsächlich um Datenintegrität kümmert. Ein weiteres Gegenargument war, dass eine Prüfsumme auf der Transportschicht bereits vorhanden ist, weshalb innerhalb der Vermittlungsschicht keine weitere Prüfsumme notwendig sei. Letztendlich fiel die Entscheidung, dass IPv6 keine Prüfsumme enthält.

4.3.4 IPv6-Erweiterungs-Header

IPv6 nutzt das Konzept der Erweiterungs-Header, um a) eine effiziente Datenübertragung und b) eine Erweiterung des Protokolls zu ermöglichen. Der erste Punkt ist leicht ersichtlich: Der Basis-Header enthält nur Felder, die unbedingt für die Übermittlung eines Datengramms notwendig sind, erfordert die Übertragung weitere Optionen, so können diese über einen Erweiterungs-Header angegeben werden. IPv6 sieht vor, das einige Merkmale des Protokolls nur gezielt benutzt werden. Ein gutes Beispiel ist hier die Fragmentierung von Datengrammen. Obwohl viele IPv4-Datengramme nicht fragmentiert werden müssen, enthält der IPv4-Header Felder, für die Fragmentierung. IPv6 gliedert die Felder für die Fragmentierung in einen separaten Header aus, der wirklich nur dann verwendet werden muß, wenn das Datengramm tatsächlich fragmentiert werden muß. Ein weiterer wesentlicher Vorteil des Konzepts der Erweiterungs-Header ist, dass das Protokoll um neue Funktionen erweitert werden kann. Es genügt, für das Feld *Next Header* einen neuen Typ und ein neues Header-Format zu definieren. IPv4 erfordert hierzu eine vollständige Änderung des Headers.

Derzeit sind 6 Erweiterungs-Header definiert. Alle Erweiterungs-Header sind optional. Werden mehrere Erweiterungs-Header verwendet, so ist es erforderlich, sie in einer festen Reihenfolge anzugeben.

Header	**Beschreibung**
IPv6-Basis-Header	Zwingend erforderlicher IPv6-Basis-Header
Optionen für Teilstrecken (Hop-by-Hop Options Header)	Verschiedene Informationen für Router
Optionen für Ziele (Destination Options Header)	Zusätzliche Informationen für das Ziel
Routing (Routing Header)	Definition einer vollständigen oder teilweisen Route
Fragmentierung (Fragment Header)	Verwaltung von Datengrammfragmenten
Authentifikation (Authentication Header)	Echtheitsüberprüfung des Senders
Verschlüsselte Sicherheitsdaten (Encapsulating Security Payload Header)	Informationen über den verschlüsselten Inhalt
Optionen für Ziele (Destination Options Header)	Zusätzliche Informationen für das Ziel (für Optionen, die nur vom endgültigen Ziel des Pakets verarbeitet werden müssen)
Header der höheren Schichten (Upper Layer Header)	Header der höheren Protokollschichten (TCP, UDP etc.)

Die ersten 5 Header sind in RFC 2460 (bzw. RFC 1883). Der Authentifikations-Header sowie der Header für Sicherheitsdaten werden in RFC 2402 (RFC 1826) und RFC 2406 (RFC 1827) beschrieben.

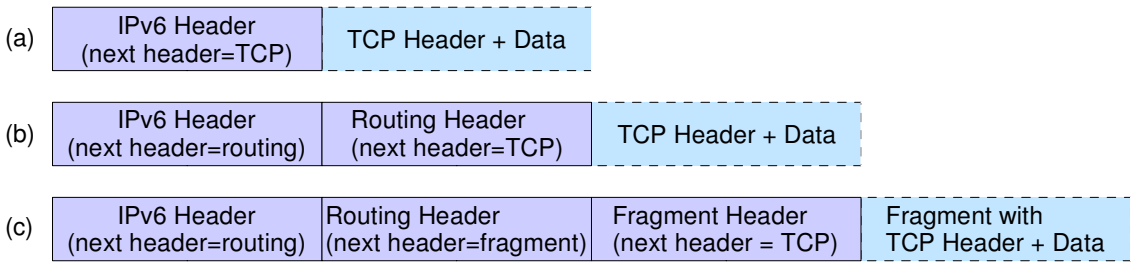


Abbildung 26 IPv6-Datengramme. (a) IPv6-Basis-Header und Nutzdaten (b) IPv6-Basis-Header mit einem Zusatz-Header für Routing-Informationen und Nutzdaten (c) IPv6-Basis-Header mit einem Zusatz-Header für Routing-Informationen, einem Zusatz-Header für Fragmentierung und Nutzdaten.

Hop-by-Hop Options Header

Routing Header

Fragment Header

Destination Options Header

4.3.5 IPv6-Adressierung

5 Quellenverzeichnis

5.1 Anmerkungen zur Literatur

Es gibt eine große Zahl an Literatur, die sich mit TCP/IP und Computernetzwerken im allgemeinen befasst. Ich möchte an dieser Stelle einige Werke hervorheben, die mir bei der Erstellung des Seminarvortrags und dieses Textes besonders hilfreich waren.

Comer D.E.: Internetworking with TCP/IP, Volumes I-III

Die drei Bände von Comer über TCP/IP sind wohl **die** Werke über TCP/IP schlechthin - wird von einigen auch als „TCP/IP-Bibel“ bezeichnet.

Stevens R. W.: TCP/IP Illustrated, Volumes I-III

Stevens hat mit seinem dreibändigen Werk über TCP/IP wohl die zweite Bibel zum Thema TCP/IP verfasst (oder nennen wir es das Neue Testament). Mir persönlich gefallen die drei Bände von Stevens noch besser als die Bücher von Comer. Stevens hat für die Darstellung der TCP/IP-Protokolle einen anderen Weg, als die bloße theoretische Vorstellung der einzelnen Protokolle gewählt, die Protokolle werden anhand eines Analyse-Tools untersucht und ihre Funktionsweise so anschaulich vermittelt.

Tanenbaum A.S.: Computernetzwerke

Wer allgemein etwas über Computernetzwerke erfahren möchte, der sollte in dieses Buch schauen. Das Buch deckt so ziemlich alles ab, was mit Computernetzen zu tun hat und ist dabei sehr gut geschrieben. Im WWW werden vom Autor und dem Verlag Leseproben und alle Abbildungen des Buches zur Verfügung gestellt (siehe die Angaben in der Literaturliste).

Kurose J.F., Ross K.W.: Computernetze

Normalerweise werden Computernetze in der Literatur immer von unten nach oben, d.h. von der Netzwerkschicht hinauf zur Applikationsschicht beschrieben. Kurose und Ross wählen den umgekehrten Weg und beschreiben Computernetze in einem Top-Down-Ansatz von der Applikationsschicht hinunter zur Netzwerkschicht. Das Buch enthält zahlreiche Interviews mit Fachleuten und gibt damit einen interessanten Einblick in die Praxis. Das Buch steht in der englischsprachigen Ausgabe auch komplett im WWW zur Verfügung.

Comer D.E.: Computernetzwerke und Internets

Noch ein gutes Buch von Douglas Comer. Das Buch gibt eine sehr gute und umfassende Darstellung aller wichtigen Themen im Bereich Netzwerke und Internets. Es geht dabei nicht stark in die Tiefe, sondern gibt eher einen ersten Einstieg in das Thema. Zu dem Buch gibt es auch eine WWW-Seite, auf der weitere Informationen zum Buch (insbesondere alle Abbildungen und zahlreiche Animationen) vorhanden sind. Dem Buch ist eine Kopie der WWW-Seiten als CD-ROM beigelegt.

Huitema C.: IPv6 die neue Generation

Eine hervorragende Übersicht über das neue IP. Es wird verdeutlicht, wie sich IPv6 von IPv4 unterscheidet und welche Vorbereitungen für den Einsatz von IPv6 getroffen werden müssen. Neben allen wichtigen Erläuterungen zum neuen Internet Protokoll wird fast nebenher auch ein geschichtlicher Überblick über die Entwicklung von IPv6 gegeben.

Holzmann G.J.: Design and Validation of Computer Protocols

Dieses Buch ist eine sehr gute Einführung in (Computer)Protokolle, den Entwurf von Protokollen und Verifikation von Protokollen. Und das beste ist, dass das Buch vollständig als Postscript- oder PDF-Datei im Internet verfügbar ist!

Hunt C.: TCP/IP Network Administration

Ein Buch, das die Grundlagen von TCP/IP bespricht und vor allem für die Administration von TCP/IP Netzwerken in einer UNIX Umgebung gedacht ist. Neben einer Version, die sich speziell mit der Administration von TCP/IP-Netzwerken in UNIX-Umgebungen befasst ist auch eine Version für TCP/IP unter Windows verfügbar.

Hafner K., Lyon M.: ARPA KADABRA - Die Geschichte des Internet

Das Buch erzählt die Geschichte des Internet und der Gruppe von Wissenschaftlern, die maßgeblich an der Entwicklung des Internet bzw. seines Vorläufern beteiligt waren. Sehr kurzweilig geschrieben und mehr eine Art Erzählung als ein Sachbuch, aber dennoch eine hervorragende sachliche Darstellung der Internet-Geschichte.

RFCs - Request for Comments

Ein RFC ist ein Papier, das sich in irgendeiner Form mit Verfahren, die im Internet verwendet werden, beschäftigt. Dabei kann es sich um einen Verbesserungsvorschlag oder eine Anmerkung für ein bestehendes Verfahren handeln oder einem Vorschlag zu einem neuen Verfahren. Ein Vorschlag zu einem neuen Verfahren kann nach einiger Zeit (und Prüfung) dann zu einem Standard erklärt werden. Jedem vorgeschlagenen Standard wird eine Nummer zugeordnet. Die folgende Liste gibt eine Reihe von RFCs wieder, die sich mit Themen des Vortrags bzw. dieser Arbeit befassen:

- RFC768 - UDP
- RFC783 - TFTP
- RFC791 - IP
- RFC792 - ICMP
- RFC793 - TCP
- RFC814 - Name, adresses, ports and routes
- RFC821/2 - Mail
- RFC825 - Specification for RFC's
- RFC826 - ARP
- RFC854 - TELNET
- RFC894 - A Standard for the Transmission of IP Datagrams over Ethernet
- RFC903 - RARP
- RFC950 - Internet Standard Subnetting Procedure (Subnets)
- RFC959 - FTP
- RFC1009 - Requirements for Internet Gateways
- RFC1011 - Official Internet Protocols
- RFC1013 - X Window System Protocol, Version 11 (Alpha Update)

- RFC1032/3/4/5 - Domains (Domain Administration & Domain Names)
- RFC1042 - Transmission of IP Datagrams over IEEE 802 Networks
- RFC1058 - Routing Information Protocol
- RFC1112 - Host Extensions for IP Multicasting
- RFC1117 - Internet numbers
- RFC1118 - Hitchhikers guide to the Internet
- RFC1180 - A TCP/IP Tutorial
- RFC1208 - Networking glossary of terms
- RFC1310 - The Internet Standards Process
- RFC1323 - TCP Extensions for High Performance
- RFC1521 - MIME
- RFC1550 - IPng White Paper Solicitation
- RFC1597 - Address Allocation for Private Internets
- RFC1700 - Assigned Numbers (Well-known Ports etc.)
- RFC1752 - Recommedation for the IP Next Generation Protocol
- RFC1825 - Security Architecture for the Internet Protocol
- RFC1826 - IP Authentication Header
- RFC1883 - Internet Protocol, Version 6 (IPv6)
- RFC1884 - IP Version 6 Addressing Architecture
- RFC1885 - Internet Control Message Protocol (ICMPv6)
- RFC1886 - DNS Extensions to support IP version 6
- RFC1918 - Address Allocation for Private Internets
- RFC1972 - Transmission of IPv6 Packets over Ethernet Networks
- RFC2019 - Transmission of IPv6 Packets over FDDI Networks
- RFC2200 - Internet Official Protocol Standards

5.2 Literaturliste

Im Gegensatz dazu, wie es bei wissenschaftlichen Veröffentlichungen üblich ist, nenne ich in der Literaturliste auch Quellen, die nicht unmittelbar im Text zitiert sind. Viele der hier angegebenen Quellen habe ich zum Thema gelesen und indirekt Wissen aus ihnen in den Text einfließen lassen. Ich fand es immer hilfreich einen Überblick über die Literatur zu einem Gebiet zu bekommen und möchte deshalb eine Literaturliste mit einer Vielzahl hilfreicher Quellen angeben.

[Black1999] Black U.: Internet-Technologien der Zukunft. Addison-Wesley, München, 1999.

[ComerStevens1995] Comer D.E., Stevens D.L.: Internetworking with TCP/IP, Vol. I - Priciples, Protocols and Internals. Prentice Hall, Englewood Cliffs, New Jersey, 1995, 3rd ed.

[ComerStevens1994] Comer D.E., Stevens D.L.: Internetworking with TCP/IP, Vol. II - Design, Implementation and Internals. Prentice Hall, Englewood Cliffs, New Jersey, 1994, 2nd ed.

[ComerStevens1996] Comer D.E., Stevens D.L.: Internetworking with TCP/IP, Vol. III - Client-Server Programming and Applications for the BSD Socket Version. Prentice Hall, Englewood Cliffs, New Jersey, 1996, 2nd ed.

[Comer1998] Comer D.E.: Computernetzwerke und Internets. Prentice Hall, München, 1998. http://www.netbook.cs.purdue.edu.

[Davidson1988] Davidson J.: Introduction to TCP/IP. Springer, New York, 1988

[DE-NIC] DE-NIC: Deutsches Network Information Center. http://www.nic.de/

[Deering1993] Deering S.E.: SIP - Simple Internet Protocol. IEEE Network Magazine, Band 7, S. 16-28, Mai/Juni 1993.

[Francis1993] Francis P.: A Near-Termn Architecture for Deploying Pip. IEEE Network Magazine, Band 7, S. 30-37, Mai/Juni 1993.

[Hasenstein1997] Hasenstein M.: IP Network Address Translation. Diplomarbeit an der Technischen Universität Chemnitz, 1997. Online verfügbar unter: http://www.suse.de/~mha/HyperNews/get/linux-ip-net.html.

[HafnerLyon2000] Hafner K., Lyon M.: ARPA Kadabra - oder die Geschichte des Internet. DPunkt-Verlag, Heidelberg, 2000, 2. Auflage.

[Hartjes1997] Hartjes K., Löffler H., Wessendorf G.: Fortsetzung folgt: Aktuelles zur IPv6-Einführung. ix 4/97, S. 97ff.

[Hedrick1987] Hedrick C.L.: Introduction to the Internet Protocols . Computer Science Facilities Group, Rutgers The State University of New Jersey, 1987.

[Hinden] Hinden R.: IP Next Generation (IPng). http://playground.sun.com/pub/ipng/html/ipng-main.html.

[Holzmann1991] Holzmann G.J.: Design and Validation of Computer Protocols. Prentice Hall, Englewood Cliffs, New Jersey, 1991.
Im Internet unter: http://cm.bell-labs.com/cm/cs/what/spin/Doc/Book91.html.

[HosenfeldBrauer1995] Hosenfeld F., Brauer K.: Kommunikation ohne Grenzen: TCP/IP - Informationsübermittlung im Internet. c't 12/95, S. 330ff.

[Hosenfeld1996] Hosenfeld F.: Next Generation - IPv6: ein neues Kommunikationszeitalter?. c't 11/96, S. 380ff.

[Huitema2000] Huitema, C.: IPv6 - die neue Generation, Architektur und Implementierung. Addison Wesley, München, 2000.

[Hunt1995] Hunt C.: TCP/IP Network Administration. O'Reilly & Assoc., Sebastopol, CA, 1995

[KatzFord1993] Katz D., Ford P.S.: TUBA - Replacing IP with CLNP. IEEE Network Magazine, Band 7, S. 38-47, Mai/Juni 1993.

[Kirch] Kirch O.: LINUX Network Administrators Guide. http://metalab.unc.edu/mdw/LDP/nag/nag.html.

[Köhntopp1993a] Köhntopp K.: Weltweit vernetzt - Struktur und Dienste des Internet. c't 2/93, S. 82ff.

[Köhntopp1993b] Köhntopp K.: Einheitliche Sicht - Netzwerkprotokolle im Internet. c't 3/93, S. 232ff.

[Krempl1998] Krempl S.: Das Internet bleibt spannend! Im Gespräch mit 'Internet-Vater' Vinton G. Cerf. c't 3/98. S. 44ff.

[Kuri1996] Kuri J.: Wenn der Postmann zweimal klingelt - Namen und Adressen im TCP/IP-Netzwerk und im Internet. c't 12/96, S. 334ff.

[Kuri1997a] Kuri J.: Böhmische Dörfer - Vom Kabel zum Netzwerk. c't 1/97, S. 245ff.

[Kuri1997b] Kuri J.: Da geht's lang! - Routing, oder: wie die Daten im Internet ihren Weg finden. c't 6/97, S. 380ff.

[KuroseRoss2002] Kurose J.F., Ross K.W.: Computernetze, Ein Top-Down-Ansatz mit Schwerpunkt Internet, Addison-Wesley, 2002.

Unter http://www.awl.com/kurose-ross steht das gesamte Buch online zur Verfügung, ebenso weitere ergänzende Materialien.

[Kuschke1994] Kuschke M.: Vervierfacht - IPv6: neue Spezifikationen zur Lösung des Adreßdilemmas. ix 10/94, S. 132ff.

[Microsoft1998] Microsoft: Windows NT Server - Introduction to TCP/IP. White Paper, Microsoft Corporation, Redmond, 1998. Online verfügbar unter: http://www.microsoft.com/NTServer/nts/techdetails/compares/TCPIntrowp.asp.

[Micrsoft2001] Microsoft: Introduction to IP Version 6. White Paper, Microsoft Corporation, Redmond, 2001. Online verfügbar unter: http://www.microsoft.com/technet/network/ipvers6.aps.

[OberschelpVossen1995] Oberschelp W., Vossen G.: Rechneraufbau und Rechnerstrukturen. Oldenbourg, München, 1995, 6. Auflg.

[RFC] RFCs - Requests For Comments: Liste aller RFC http://www.RFC-editor.org oder: http://internic.net/ds/RFC-index.txt oder: ftp://ftp.nic.de/RFC/pub/doc/RFC
Es gibt aber noch viele andere Adressen, unter denen die RFCs zu bekommen sind, auch als Zustellung per E-Mail.

[Santifaller1998] Santifaller M.: TCP/IP und ONC/NFS - Internetworking mit UNIX. Addison Wesley, Bonn, Reding Massachucetts, 1998, 4. Auflage.

[Schmidt1997a] Schmidt J.: Firewall getunnelt - Geheimer Datenaustausch über ICMP-Pakete. c't 11/97, S. 332ff.

[Schmidt1997b] Schmidt J.: Falsche Fährten - Mißbrauchsmöglichkeiten von ARP und ICMP. c't 12/97, S. 246ff.

[Sietmann2000] Sietmann R.: Mobilmachung - Internetprotokoll Version 6. iX 4/2000, S. 100ff.

[Stainov2000] Stainov, R.: IPnG - Das Internet-Protokoll der nächsten Generation. Thomson Publishing, Bonn, 1997.

[Stallings1987] Stallings W.: Handbook of Computer Communications Standards, Vol. I: The Open Systems Interconnection (OSI) Model and OSI-Related Standards. MacMillan Pub. Company, Indianapolis, 1987

[Stallings1988a] Stallings W.: Handbook of Computer Communications Standards, Vol. II: Local Network Standards. MacMillan Pub. Company, New York, London, 1988

[Stallings1988b] Stallings W.: Handbook of Computer Communications Standards, Vol. III: Department of Defense (DoD) Protocol Standards. MacMillan Pub. Company, New York, London, 1988

[Stevens1999a] Stevens R. W.: TCP/IP Illustrated, Vol. 1: The Protocols. Addison Wesley, 1999, 14th ed.

[StevensWright1999b] Stevens R. W., Wright G. R.: TCP/IP Illustrated, Vol. 2: The Implementation. Addison Wesley, 1999, 7th ed.

[StevensWright1999c] Stevens R. W., Wright G. R.: TCP/IP Illustrated, Vol. 3: TCP for transactions, HTTP, NNTP, and the UNIX Domain protocols. Addison Wesley, 1998, 4th ed.

[Tanenbaum1996] Tanenbaum A.S.: Computernetzwerke. Prentice Hall, München, 1997, 3. Auflage.)

Zu dem Buch werden im WWW auch zusätzliche Informationen vom Verlag unter http://www.prenhall.com/divisions/ptr/tanenbaum/book.html und vom Autor unter http://www.cs.vu.nl/~ast zur Verfügung gestellt.

[Weihrich1997] Weihrich T.: Filofax fürs Internet - Der Domain Name Service von TCP/IP. c't 10/97, S. 346ff.

[WiN] WiN/DFN: IP Version 6 im WiN - Ein DFN Projekt. http://www.join.uni-muenster.de.

5.3 Wichtige Organisationen

Internet Architecture Board – IAB (http://www.iab.org)

Das **Internet Architecture Board (IAB)** ist eine technische Beratungsgruppe der Internet Society, die sich der organisatorisch-technischen Entwicklung des Internets widmet. Vom IAB werden die gültigen Standards und Protokolle des Internets in so genannten RFCs veröffentlicht. Die beiden

wichtigsten Gremien des IAB sind die Internet Engineering Task Force (IETF) und die Internet Research Task Force (IRTF).

Internet Society – ISOC (http://www.isoc.org)

Die **Internet Society (ISOC)** ist eine 1992 gegründete NGO (Non Government Organization) für die Pflege und Weiterentwicklung der Internetinfrastruktur und der Art, wie das Internet genutzt wird. Neben sozialen, politischen und technischen Fragen befasst sich die ISOC auch mit PR-Arbeiten. Die ISOC beherbergt die für die Standards im Internet zuständigen Gremien Internet Engineering Task Force (IETF) und Internet Architecture Board (IAB). Die ISOC ernennt die Mitarbeiter des IAB, die vom Nominierungsausschuss der IETF vorgeschlagen werden.

Internet Engineering Task Force – IETF (http://www.ietf.org)

Die **Internet Engineering Task Force (IETF)** ist neben der Internet Research Task Force (IRTF) eine von zwei Arbeitsgruppen des Internet Architecture Board (IAB). Sie ist eine offene, internationale Vereinigung von Netzwerktechnikern, Herstellern und Anwendern, die für Vorschläge zur Standardisierung des Internets zuständig ist und wurde 1986 gegründet. Die IETF gliedert sich in verschiedene Bereiche (Anwendungen (Applications), Internet-Dienste, IP Next Generation (IPnG), Netzwerkmanagement, Betriebliche Anforderungen (Operational Requirements),. Routing, Sicherheit (Security), Transportdienste und Benutzerdienste (Transport and User Services)). Jeder Bereich hat ein oder zwei Direktoren, die zusammen mit dem IETF-Vorsitzendem bilden die Internet Engineering Steering Group (IESG). Die IESG ist verantwortlich für das technische Management der IETF Aktivitäten und für die Internet-Standards. Eine detaillierte Übersicht zur IETF ist in *RFC 3160 The TAO of IETF* zu finden.

Internet Research Task Force – IRTF (http://www.irtf.org)

Die **Internet Research Task Force (IRTF)** ist neben der Internet Engineering Task Force (IETF) die zweite Arbeitsgruppe des Internet Architecture Board (IAB). Sie wurde 1998 gegründet um die Forschung und Entwicklung im Bereich der Netzwerke und deren Techniken zu fördern. Sie besteht aus Forschern im Bereich der Netzwerktechnik mit dem Schwerpunkt Internet. Die Internet Research Steering Group (IRSG) leitet und koordiniert die Forschungsarbeiten der IRTF. Dabei kommt es mitunter zu Überschneidungen mit den Arbeiten der IETF. Auch bei den Mitgliedern der Gruppen gibt es Überschneidungen. Ähnlich wie die IETF ist die IRTF in Arbeits- bzw. Forschungsgruppen organisiert, die sich mit unterschiedlichen Themen befassen. Weitere Informationen sind in *RFC 2014 Research Group Guidelines and Procedures* zu finden.

Internet Assignes Numbers Authority – IANA (http://www.iana.org)

Die **Internet Assigned Numbers Authority (IANA)** ist eine Organisation, die die Vergabe von IP-Adressen, Top Level Domains und IP-Protokollnummern regelt. Die IANA delegiert die lokale Registration von IP- Adressen an Regionale Internet-Registries (RIRs). Jede RIR ist für einen bestimmten Teil der Welt verantwortlich, im einzelnen:

- ARIN - American Registry for Internet Numbers (http://www.arin.net)

- RIPE - Réseaux IP Européens (http://www.ripe.net)
- APNIC - Asia Pacific Network Information Center (http://www.apnic.net)
- LACNIC - Latin American and Caribbean Internet Address Registry (http://www.lacnic.net)
- AfriNIC – African Internet Numbers Registry (http://www.afrinic.net)

Die IANA verteilt IPv4-Adressen in großen Blöcken (typischerweise /8 in CIDR-Notation), und die RIRs folgen dann ihren eigenen Regelungen für die Zuweisung von Adressen an Endkunden (in diesem Sinne Provider oder Organisationen, die ihre IP-Adressen selbst verwalten), wobei dann meist /19er oder /20er Blöcke zugeteilt werden. Die IANA ist auch für die Delegation und Zuweisung von IPv6-Adressen zuständig Die IANA ist organisatorisch eine Unterabteilung der Internet Corporation for Assigned Names and Numbers (ICANN) bzw. ist in der ICANN aufgegangen.

Internet Corporation for Assigned Names and Numbers – ICANN (http://www.icann.org)

Die **Internet Corporation for Assigned Names and Numbers (ICANN)** verwaltet Namen und Adressen im Internet und koordiniert somit technische Aspekte des Internet. Sie wird oft auch, nicht ohne Wertung, als eine Art „Weltregierung des Internets“ bezeichnet. Die ICANN wurde im Oktober 1998 von einem Zusammenschluss verschiedener Interessenverbände (Wirtschaft, Technik, Wissenschaft und Nutzer) gegründet. Die ICANN hat die Verantwortung für eine Reihe technischer Vorgaben, die zuvor von der IANA und verschiedenen anderen Gruppen getragen wurden. Das Direktorium der ICANN besteht aus verschiedenen Mitgliedern aus aller Welt. Die ICANN koordiniert:

- Internet-Domain-Namen (Domain Name System, speziell die Root-Server),
- IP-Adressen,
- Protokoll-Parameter und Port-Adressen der Internet- Protokoll-Familie.

World Wide Web Consortium – W3C (http://www.w3c.org)

Das **World Wide Web Consortium (W3C)** wurde 1994 gegründet und ist das Gremium, das speziell die Weiterentwicklung technischer Standards des World Wide Web (WWW) koordiniert. Gründer und Vorsitzender des W3C ist Tim Berners-Lee, der auch als „Erfinder“ des World Wide Web gilt. Beispiele für vom W3C maßgeblich entwickelte und verabschiedete Standards sind:

- Hypertext Markup Language (HTML)
- Extensible Markup Language (XML)
- Extensible Hypertext Markup Language (XHTML)
- Synchronized Multimedia Integration Language (SMIL)
- Scalable Vector Graphics (SVG)
- Mathematical Markup Language (MathML)
- Portable Network Graphics (PNG)

Das W3C und seine Mitglieder beschäftigen sich auch mit der Weiterentwicklung von Standards

oder mit der Entwicklung neuer Standards. Bei der Entwicklung neuer Standards hat sich das W3C verpflichtet, nur noch Technologien zu verwenden, die frei von Patentgebühren sind.

Institute of Electrical and Electronics Engineers – IEEE (http://www.ieee.org)

Das **Institute of Electrical and Electronics Engineers (IEEE - „ei tripel ih“)** ist ein weltweiter Berufsverband von Ingenieuren aus den Bereichen Elektrotechnik und Informatik. Das IEEE ist Veranstalter von Fachtagungen und bildet Gremien für die Standardisierung von Technologien, Hardware und Software. Das IEEE entstand am 1. Januar 1963 aus dem Zusammenschluss der beiden amerikanischen Ingenieursverbände American Institute of Electrical Engineers (AIEE) und Institute of Radio Engineers (IRE). Wichtige IEEE-Standards sind u.a.

- IEEE 488 (Bussystem für Peripheriegeräte)
- IEEE 754 (Fließkomma-Arithmetik-Spezifikationen)
- IEEE 802 (LAN/MAN)
- IEEE 1284 (Parallele Schnittstelle)
- IEEE 1003 (POSIX)
- IEEE 1394 (FireWire)
- IEEE 802.11 (Wireless LAN)

International Telecommunication Union – ITU (http://www.itu.org)

Der Status von Telefongesellschaften ist den verschiedenen Ländern der Welt äußerst unterschiedlich. Häufig haben Landesregierungen ein Monopol auf nahezu alle Kommunikationsmedien (wie z.B. Post, Telegrammdienst, Telefon, Radio und Fernsehen). In einigen Fällen ist die Behörde, die für die Regelung der Telekommunikation zuständig ist, ein staatliches Unternehmen, in anderen Fällen ist es direkt eine Regierungsorganisation. Es zeichnet sich weltweit jedoch ein Trend ab, diese Monopolstellungen aufzuheben und den Telekommunikationsbereich für private Unternehmen zu öffnen. Angesichts der Vielzahl verschiedener Betreiber und Anbieter liegt es nahe, für eine globale Kompatibilität zu sorgen, damit Menschen und technische Geräte in allen Ländern der Erde untereinander genutzt werden können. Die **International Telecommunication Union (ITU, frz: Union Internationale des Télécommunications, UIT)** mit Sitz in Genf ist die Organisation, die sich offiziell und weltweit mit technischen und rechtlichen Aspekten der Telekommunikation beschäftigt. Die ITU geht zurück auf den 1865 gegründeten Internationalen Telegraphenverein. Heute ist sie eine Teilorganisation der Vereinten Nationen (UNO). Die Ziele der ITU sind Abstimmung und Förderung der internationalen Zusammenarbeit im Nachrichtenwesen. In ihrem Rahmen arbeiten Landesregierungen, Unternehmen des privaten Sektors, sowie weitere regionale und nationale Organisationen zusammen. Grundlage der ITU ist der Internationale Fernmeldevertrag, der Aufgaben, Rechte und Pflichten der ITU-Organe festlegt. Wichtig ist festzuhalten, dass die Empfehlungen der ITU nur als Empfehlungen zu betrachten sind, die von den einzelnen Landesregierungen angenommen oder ignoriert werden können.

International Organization for Standardization – ISO (http://www.iso.org)

Die **International Organization for Standardization (ISO, auch als International Standards Organization bezeichnet)**, erarbeitet internationale Normen (engl. Standards) in so gut wie allen Bereichen (Ausnahme ist z.B. der Bereich Elektrik und Elektronik, für den die International Electrotechnical Commission) zuständig ist), wie z.B. technischen Standards (Normung von Schrauben und Muttern, Leitungsbeschichtungen, Datenformate), klassifikatorischen Standards (die berühmten Ländercodes für Domains .de, .nl, .jp etc.) und Verfahrensstandards (ISO9000 Qualitätsmanagement). Die ISO wurde 1946 gegründet und ist eine freiwillige, nicht per Staatsvertrag geregelte Organisation. Mitglieder der ISO sind die nationalen Normungsinstitute der jeweiligen Mitgliedsländer (z.B. ANSI (USA), BSI (Großbritannien), AFNOR (Frankreich)). Das deutsche Institut für Normung e.V. ist seit 1951 Mitglied der ISO. In Bezug auf Telekommunikationsstandards arbeitet die ISO eng mit der ITU zusammen.

Deutsches Institut für Normung e.V. - DIN (http://www.din.de)

Das **Deutsches Institut für Normung e.V. (DIN)** mit Sitz in Berlin ist die nationale Normungsorganisation in Deutschland. Der Verein entwickelt in Zusammenarbeit mit Handel, Industrie, Wissenschaft, Verbrauchern und Behörden technische Standards (Normen) zur Rationalisierung und Qualitätssicherung. Das DIN vertritt die deutschen Interessen in den internationalen Normungsgremien (wie ISO und IEC). Durch die Festlegung der Normen soll sichergestellt werden, dass die Inhalte und Verfahrenstechniken den allgemein anerkannten Regeln der Technik entsprechen. Dem DIN angegliedert ist der Beuth-Verlag, der den Vertrieb der vom DIN herausgegebenen Normen, Normen anderer Normungsstellen und ausländischer Normen übernimmt. Gegründet wurde das DIN 1917 als Normenausschuss der deutschen Industrie (NADI). Die erste Norm (DIN 1 Kegelstifte) erschien im Jahr 1918. Seit 1920 ist das DIN ein eingetragener Verein. 1926 wurde das DIN von Normenausschuss der deutschen Industrie in Deutscher Normenausschuss (DNA) umbenannt. 1975 erfolgte schließlich die Umbenennung auf den heutigen Namen im Rahmen eines Vertrages zwischen der Bundesregierung und dem Verein, der die Zusammenarbeit zwischen beiden festlegt.