INSTITUT FÜR INFORMATIK
DER
LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN
TECHNISCHEN UNIVERSITÄT MÜNCHEN

IP Next Generation: IPv6

Hauptseminar im Sommersemester 1996:
Moderne Kommunikationskonzepte,
Protokolle und Dienste
Lehrstuhl Prof. Dr. H.-G. Hegering

13. Juni 1996

Marco Avitabile (avitabil@informatik.tu-muenchen.de)
Gerhard Müller (muellerg@informatik.tu-muenchen.de)

Dieses Dokument ist im Internet unter der URL
http://www.informatik.tu-muenchen.de/~muellerg/ipng/ verfügbar

Überblick

Das aktuelle *Internet Protocol* Version 4 stößt allmählich an Grenzen, die bei seiner Entwicklung noch nicht abzusehen waren. Der Adreßraum von IPv4 und seine geringe Skalierbarkeit haben sich als beschränkender Faktor für das Wachstum des Internets herausgestellt. Um die aktuellen Probleme zu lösen und neuartige Anwendungen zu ermöglichen, wird derzeit eine neue Version des Internet Protocols, die Version 6, welche in diesem Dokument vorgstellt wird, entwickelt. Damit wird ein wesentlicher Bestandteil der Internet-Protokollfamilie geändert. Der Übergang auf IPv6 wird allmählich vonstatten gehen. Spezielle Mechanismen in IPv6 erleichtern die Migration von IPv4 nach IPv6. Dieses Protokoll wird in den nächsten Jahren auf viele der im Moment an das Internet angeschlossenen Systeme Auswirkungen haben. Die Entwicklung von IPv6 ist noch nicht abgeschlossen.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einführung

1.1 Einordnung des Internet Protocols

Das *Internet* ist in den letzten Jahren äußerst populär, besonders durch die Entwicklung und den Erfolg des World Wide Web. Die Größe des Internets verdoppelt sich seit Anfang der 90er Jahre jährlich [Tan96]. Die Grundlage des Internets ist das *Internet Protocol*, welches eine zentrale Rolle in der Internet-Protokollfamilie spielt.

Die Aufgabe des aktuellen Internet Protocols wird in der formalen Definition, RFC 791, wie folgt beschrieben:

> The Internet Protocol is designed for use in interconnected systems of packet-switched computer communication networks. [...] The internet protocol provides for transmitting blocks of data called datagrams from sources to destinations, where sources and destinations are hosts identified by fixed length addresses.

Das Internet Protocol (kurz IP) stellt einen unzuverlässigen „best effort“ Datagrammdienst zwischen Systemen zur Verfügung. Im OSI-Referenzmodell ist es auf Schicht 3 anzusetzen. Abbildung 1 stellt die Einordnung des Internet Protocols in die Internet-Protokollfamilie graphisch dar.

Seit 1973 wurde das Internet Protocol nicht mehr wesentlich verändert. Im Laufe der Zeit haben sich jedoch die Anforderungen an dieses stark geändert. Ein besonders drängendes Problem ist die geringe Anzahl der noch vergebbaren Class B Netze, und auf Dauer der Mangel an IP-Adressen allgemein. Diese reichen nur noch für wenige Jahre. Aus diesem und anderen Gründen wird im Moment eine neue Version des Internet Protocols entwickelt.

Dieser Text beschreibt die Hintergründe für die Entwicklung einer neuen Version 6 des Internet Protocols (IPv6), welche neuen Möglichkeiten es bietet, und wie die Planung des Übergangs zu der neuen Version des Internet Protocols aussieht.

1.2 Der Standardisierungsprozeß

Das Internet hat eine eigene Organisation, das *Internet Architecture Board* (IAB). Diese koordiniert die Forschung und Entwicklung von neuen Internet Standards [Com95]. Wie in Abbildung 2 zu sehen, gehören zum IAB, neben dem „*Board*“, der obersten Kontrolle (eine Art Aufsichtsrat) zwei Hauptgruppen: die *Internet Research Task Force* (IRTF) und die *Internet Engineering Task Force* (IETF).

Die IETF konzentriert sich mehr auf kurz- und mittelfristige Probleme. Sie ist wiederum in mehrere Bereiche aufgeteilt. Der Vorsitzende der IETF und die Leiter der verschiedenen Bereiche bilden die *Internet Engineering Steering Group* (IESG), wobei die einzelnen Mitglieder verantwortlich sind für die Koordination der Bemühungen der IETF-Arbeitsgruppen („*Working Groups*“).

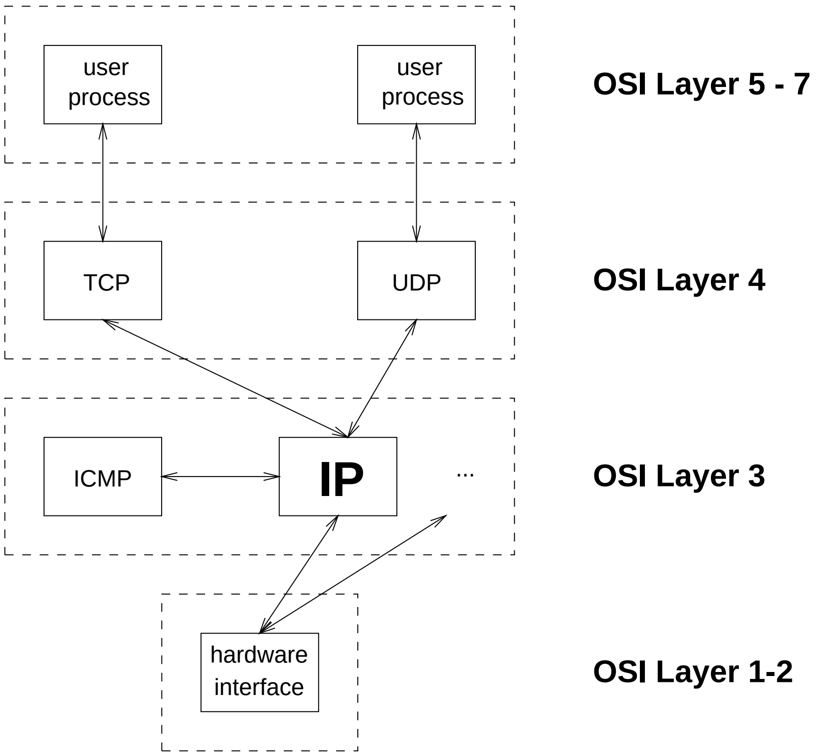

Abbildung 1: Einordnung des Internet Protocols in die Internet-Protokollfamilie

Die IRTF koordiniert die Forschungsaktivitäten in Bezug auf die Internet-Protokollfamilie und des Internets im allgemeinen. Wie die IETF, so hat auch die IRTF eine kleine Gruppe, *Internet Research Steering Group* (IRSG) genannt, welche Prioritäten setzt und die Forschungen koordiniert. Im Gegensatz zur IETF ist die IRTF wesentlich kleiner und auch weniger aktiv.

Die Mitglieder im IAB arbeiten freiwillig; wer spezielles Interesse in einem Bereich hat, kann mitarbeiten.

Vorschläge für neue oder revidierte Protokolle sowie allgemeine Dokumentation der Arbeit für das Internet erscheinen in einer Serie von technischen Berichten, welche *Request For Comments* (RFCs) genannt werden. Vorversionen von RFCs werden *Internet Drafts* genannt. Internet Drafts sind maximal 6 Monate gültig. Werden sie in dieser Zeit nicht erweitert oder ergänzt oder zu einem RFC, so werden sie ungültig. RFCs oder Internet Drafts können sehr unterschidlich lang sein, ein Detail erläutern, oder eine Übersicht über ein Gebiet geben.

RFCs werden aufsteigend durchnummeriert. Jeder RFC, neu oder revidiert, bekommt eine neue Nummer. Daher ist es wichtig, immer nur die aktuellste Nummer/Version eines Dokumentes zu betrachten.

Manche RFCs werden zu einem *Standard* erklärt. Im Moment (Anfang Juni) gibt es et-

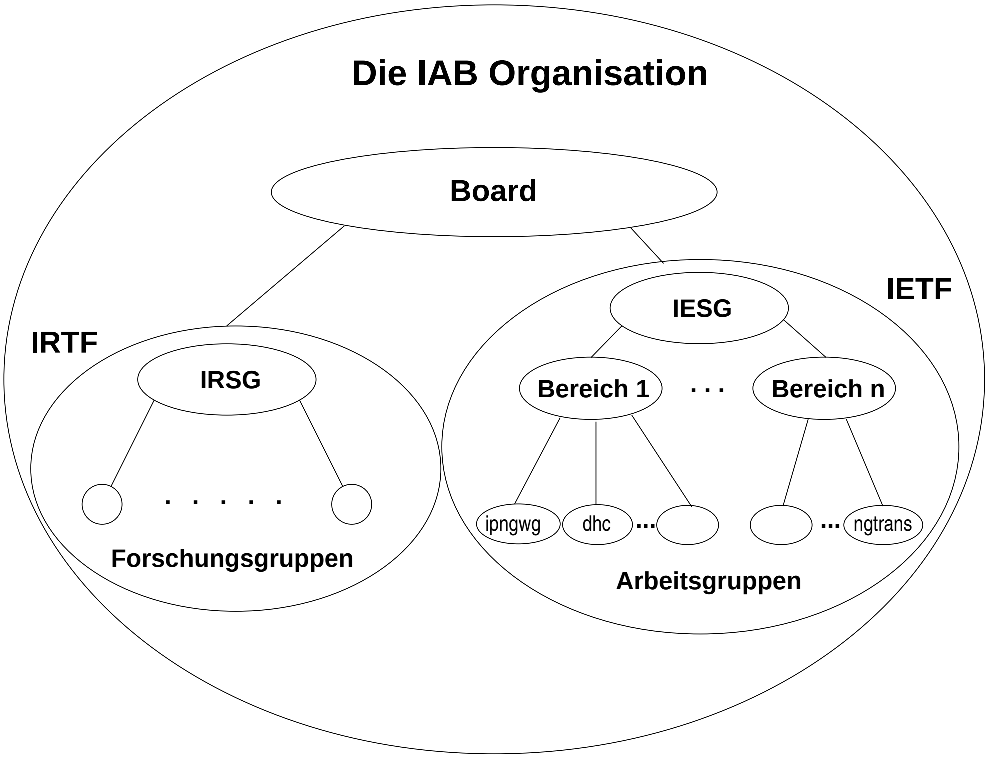

Abbildung 2: Die IAB-Organisationsstruktur

wa 1950 RFCs, 50 Standards, und rund 500 Internet Drafts. Die wenigsten Internet Drafts werden also zu einem RFC, und die wenigsten RFCs zu einem Standard.

Es gibt eine große Anzahl aktiver IETF-Arbeitsgruppen (rund 75 Arbeitsgruppen in 8 Bereichen). Sie werden unter dem Begriff „Network Working Group“ zusammengefaßt. Viele der Arbeitsgruppen sind an der Entwicklung des Internet Protocols und benachbarter Protokolle beteiligt. Die wichtigsten für die Entwicklung von IPv6 sind (mit Abkürzung):

- IP Next Generation Working Group (ipngwg)
- Dynamic Host Configuration Working Group (dhc)
- New Generation Transition Working Group (ngtrans)
- Address Autoconfiguration Working Group (addrconf)
- New Internet Routing and Adressing Architecture (nimrod)

Zusätzlich gibt es eine große Anzahl weiterer Arbeitsgruppen, die sich mit einzelnen Themengebieten, welche nicht direkt zu IPv6 gehören, intensiv beschäftigen, z.B.

- Resource Reservation Setup Protocol Working Group (rsvp)

- IPv6 MIB Working Group (ipv6mib)
- IP Routing for Wireless/Mobile Hosts Working Group (mobileip)
- IP Security Protocol Working Group (ipsec)
- Open Shortest Path First IGP Working Group (ospf)
- Routing Information Protocol Working Group (rip)

Eine Liste der aktiven IETF-Arbeitsgrupen ist unter `http://www.ietf.cnri.reston.va.us/html.charters/wg-dir.html` erhältlich.

2 Internet Protocol Version 4

2.1 Historie

Die im Moment benutzte Version des Internet Protocols, Version 4, ist schon über 20 Jahre alt. Es wurde von der Advanced Research Projects Agency (ARPA) bereits um 1973 entwickelt. Ziel war die Entwicklung eines Protokolls, welches die verschiedenen vorhandenen (physikalischen) Netzwerke verbinden konnte. Andere Ziele bei der Entwicklung waren (in absteigender Wichtigkeit) [Clark88]:

1. Die Kommunikation darf nicht durch den Verlust von einzelnen Bausteinen des Netzes nachhaltig gestört werden.
   Hier zeigt sich die militärische Herkunft; das Netz sollte auch bei Zerstörung von einzelnen Systemen weiter funktionieren; eine globale Zentrale/Kontrolle sowie starke Hirarchien verbieten sich hierdurch.

2. Es müssen verschiedene Kommunikations-Servicetypen unterstützt werden.
   Hier ist verbindunsorientiert und verbindungslos gemeint, diese werden auf der Basis von IP durch das Transmission Control Protocol *(TCP), und das* User Datagram Protocol *(UDP) erreicht.*

3. Verschiedene Netzwerktypen müssen unterstützt werden.
   Das Internet Protocol unterstützt alle gängigen Übertragungsmedien wie Ethernet, serielle Leitungen, Funknetze etc.

4. Die Netzressourcen müssen ein verteiltes Management unterstützen.

5. Alles sollte so kostengünstig wie möglich sein.

6. Neue Systeme sollten ohne größere Probleme in ein bestehendes Netz integriert werden können.

7. Die Ressourcen sollten erfaßbar sein.

Das Ergebnis dieser Entwicklung war das Internet Protocol Version 4.
Damals galt es ein Netz zu schaffen, um einige lokale Netzwerke zu verbinden. Die Anwendungen waren wissenschaftlicher sowie militärischer Natur.

Die Internet-Protokollfamilie mit dem zentralen Internet Protocol hat sich als äußerst erfolgreich erwiesen. Seine Einfachheit, die frühe Verfügbarkeit von freien Implementierungen sowie die Offenlegung von allen technischen Details haben die Internet-Protokollfamilie zu *der* Protokollfamilie für Netzwerke werden lassen. Trotz oder gerade wegen dieses Erfolges blieb das Internet Protocol bisher im wesentlichen unverändert.

2.2 Probleme

Das Internet Protocol ist in gewisser Weise Opfer des eigenen Erfolges geworden: niemand konnte den Erfolg und die Verbreitung der Internet-Protokollfamilie und des darauf basierenden Internets vorhersehen. Mit der massiven Benutzung von IP durch die Popularität des Internets werden die Adressen allmählich knapp. Dies mag erstaunen, gibt es doch theoretisch mehr als zwei Milliarden benutzbare Adressen. Die in der Praxis verwendete Aufteilung des Adreßraumes in Klassen (Class A-D) ist jedoch sehr verschwenderisch.

Besonders die Nachfrage nach von der Größe her günstigen Class B Netzen stellte für den IPv4-Adreßraum ein Problem dar. Als statt der Vergabe eines Class B Netzes mehrere aufeinanderfolgende Class C Netze ausgegeben wurden, wurden die extrem anwachsenden Routingtabellen zu einem Problem für die Router im Internet (es gibt theoretisch 16384 Class B Netze, aber viele Millionen Class C Netze).

Dieses Problem konnte durch die Einführung des *Classless Inter-Domain Routing* (CIDR) zumindest gemildert werden. Bei dem CIDR spielen, wie der Name schon sagt, die bekannten Klassen A bis D keine Rolle mehr. Beim CIDR werden zur Kennzeichnung eines zusammengehörigen Adreßbereiches die Anfangsadresse abgespeichert, sowie eine 32 Bit große Maske, welche indirekt die Größe des zusammengehörigen Bereiches angibt.
Beispiel: Die Adresse 194.24.8.0, binär

11000010 00011000 00010000 00000000

mit der Maske 255.255.240.0, binär

11111111 11111111 11110000 00000000

gibt einen Block von 4096 Adressen an.

Network Service Provider erhalten einen größeren Adreßbereich zugeordnet. Dieser wird von den Network Service Providern selbst verwaltet. Auf verschiedenen Ebenen ist so eine Zusammenfassung von Adressen möglich.

Angenommen, der Provider in obigem Beispiel verwaltet die Adressen von 194.0.0.0 bis 194.127.255.255. Alle Kunden des Providers seinen nur durch den Provider erreichbar. Dann reicht ein einziger Eintrag in den Routingtabellen aller Router außerhalb des Providers, der aussagt, daß alle IP-Pakete mit Adressen zwischen 194.0.0.0 und 194.127.255.255 zu dem Router des Providers geschickt werden sollen:

<194.127.0.0/255.128.0.0> routen zu <IP-Adresse des Provider-Routers>

Dies führt zu einer wesentlichen Reduktion von Informationen, die für das Routing vorgehalten und ausgetauscht werden müssen.

Die Einführung des CIDR hat wesentliche Auswirkungen auf verwendete Routing-Algorithmen. Die Zieladresse an sich wird nicht mehr interpretiert. Hier ist nun ein „longest match"-Algorithmus anzuwenden, der Pakete an die nächste Routingadresse schickt, die am weitesten mit der Zieladresse übereinstimmt.

Genauere Informationen zu CIDR und IPv4 geben die RFCs 1517-1520.

Trotz aller oben genannten Maßnahmen werden voraussichtlich um das Jahr 2005 bis 2011, gegenwärtige Geschwindigkeit der Adreßvergabe vorausgetzt, alle IPv4-Adressen ausgegeben sein [RFC 1752].

Andere aktuelle Probleme der aktuellen Version 4 des Internet Protocols sind:

- Sicherheit: Authentifizierung, Überprüfen der Integrität und Geheimhaltung werden nicht unterstützt.
- Konfiguration: Plug-and-Play ist nicht möglich, eine manuelle Konfiguration ist normalerweise nötig.
- Multimedia-Anwendungen: eine besondere Behandlung der Daten von isochronen Diensten wie Echtzeit-Audio und -Video werden nicht unterstützt.
- Skalierbarkeit: Routing, Multicast, und andere Bereiche sind nur ungenügend skalierbar.

3 Internet Protocol Version 6

3.1 Zukünftige Anforderungen

Bisher wurde IP im wesentlichen nur zur Vernetzung von fest an ein Netz angeschlossenen Computern genutzt. Damit war der Computermarkt der wesentliche Faktor für die Benutzung und das Wachstum des Internets. Dieses könnte sich in Zukunft durch neue Märkte ändern. Folgende Beispiele zeigen auf, in welchen Bereichen das Internet Protocol Verwendung finden könnte:

- **Mobile persönliche Computer (Personal Digital Assistants, PDAs)** werden in Zukunft wahrscheinlich so üblich sein wie heute ein Fernsehgerät oder das Telefon. Sie werden über verschiedene Medien wie Infrarot, Funk oder Kabel an das Internet angeschlossen werden. Hier ist ein Protokoll von Nöten, welches einen geringen Overhead besitzt, sowie automatische Adreßkonfiguration und Mobilität als elementare Elemente aufweist. Das Protokoll muß eine hohe Anzahl von Geräten, damit Adressen sowie deren Routing unterstützen.

- **Vernetzte Unterhaltungselektronik** wird ebenfalls in der Zukunft wohl üblich sein. Die Unterscheidung zwischen Fernsehgerät und Computer könnte schon bald der Vergangenheit angehören. Ein Fernsehgerät mit Internetanschluß ist schon heute erhältlich. Der Konsumgüter-Massenmarkt benötigt eine besonders preiswerte Technologie. Wie oben sind die Unterstützung einer einfachen Konfiguration, vieler Adressen und deren Routing essentiell.

- **Gerätesteuerung** könnte ein sehr großer Markt werden. Es ist denkbar, Steuerungsaufgaben, welche durch ein lokal abgeschlossenes properitäres System kontrolliert werden, über das Internet Protocol abzuwickeln. Beispiele für mögliche Kandidaten sind Klimaanlagen, Fahrstühle oder Heizungen. Auf lange Sicht wäre denkbar, jeden Schalter, jedes elektrische Gerät, jeden Motor mit einer Netzschnittstelle auszustatten. Dies erfordert einfache, robuste, einfach zu benutzende, sehr preiswerte Lösungen. Die dadurch resultierenden Möglichkeiten und Kostenreduktionspotentiale wären enorm.

IPv4 war nicht für diese neuartigen Anwendungen konzipiert und kann diese allein schon wegen der geringen Zahl von Adressen nicht unterstützen. Es wäre ein gewaltiger Vorteil, wenn alle Anwendungen ein gemeinsames Protokoll benutzen könnten, welches für all diese Aufgaben geeignet ist.

3.2 Entwicklungsziele

Sowohl die Benutzer als auch die Anwendungen haben sich seit der Entwicklung des Internet Protocols grundlegend geändert. Das Internet wird zur Zeit weitestgehend kommerzialisiert. Geschäftsleute und Privatbenutzer werden immer mehr zur wichtigsten Benutzergruppe. Mit der Veränderung des Benutzer- und Anwendungsprofils (wie oben dargestellt) muß ein Basisprotokoll folgenden Anforderungen genügen:

- Großer Adreßraum
- Skalierbarkeit
- Unterstützung von verschiedenen Servicetypen, insbesondere isochrone Diensten
- Unterstützung von sehr langsamen als auch sehr schnellen physischen Medien
- Unterstützung von preiswerten Endgeräten
- Erweiterbarkeit und Anpaßbarkeit an neue Anforderungen
- Unterstützung von Sicherheitsmaßnahmen (Authentifizierung, Geheimhaltung)
- Unterstützung von mobilen Endgeräten

Im Falle des Internet Protocols ist ein leichter, allmählicher Übergang von IPv4 auf eine neue Version des Internet Protocols notwendig. Dabei muß zu Anfang die Kommunikation zwischen den beiden Versionen möglich sein.

3.3 Historische Entwicklung

Es wurde der IETF klar, daß IPv4, auch mit kleinen Änderungen, den oben genannten Anforderungen nie würde entsprechen können. Um ein geeignetes Protokoll, das Internet Protocol der nächsten Generation (IPng) zu finden, bat die IETF in RFC 1550 (Dezember 1993) um Vorschläge. Aus den 21 eingehenden Vorschlägen kristallisierten sich bis Dezember 1992 sieben ernsthafte Vorschläge heraus [Tan96]. Einige rieten, nur kleinere Änderungen an IPv4 vorzunehemen. Andere rieten zu einer völligen Neuentwicklung. Ein weiterer Vorschlag war, das OSI-*ConnectionLess Network Protocol* (CLNP) zu verwenden.

Über die Verwendung von CLNP wurde ernsthaft nachgedacht, hätte es doch die zwei größten Protokollfamilien zusammengeführt. Die 160-Bit Adressen hätten den größeren Adreßansprüchen genügt. CLNP ist IP recht ähnlich, war es doch nach IP modelliert worden. Doch die Verwendung von CLNP als das Internet Protocol der nächsten Generation (IP Next Generation) konnte aus zwei Gründen nicht verwirklicht werden:

1. es galt in Internet-Kreisen als politisch inkorrekt zuzugeben, daß die OSI-Welt ein vernünftiges Protokoll hervorgebracht hatte,

2. die Unterstützung von verschiedenen Servicetypen war mangelhaft, welches wesentlich für die Unterstützung von Multimedia ist.

Drei der anderen besseren Vorschläge wurden nach langer Diskussion zu einem gemeinsamen Vorschlag zusammengeführt, dem *Simple Internet Protocol Plus*. Dieses wurde, angeregt durch RFC 1752, durch RFC 1883 zum Internet Protocol Version 6. Die Versionsnummer 5 war bereits einem experimentellen stream-basierten Protokoll, dem *Internet STream Protocol Version 2(+)* (ST2(+)) [RFC 1190]/[RFC 1819] zugeordnet worden.

IPv6 erfüllt die oben genannten Anforderungen recht gut. Es erhält alle wesentlichen Vorteile von IPv4 und beseitigt dessen Nachteile oder schwächt diese zumindest ab.

3.4 Design-Prinzipien von IPv6

IPv6 wurde unter folgenden Prämissen entworfen:

- IPv4 Funktionen, die selten oder nie benutzt wurden, werden nicht in den Standard-Header aufgenommen.

- Es sollte flexibler als IPv4 zu handhaben sein.

- Spätere Erweiterungen sollen möglich sein, ohne Bestehendes ändern zu müssen.

- Das Kernprotokoll sollte möglichst einfach gehalten werden. („Keep it simple")

- Fähigkeiten, die es selbst nicht realisiert, sollten auch nicht verhindert werden, wie z. B.

  - Management
  - Accounting
  - Quality of Service-Garantien

- Leichte Migration ist essentiell für den Erfolg eines neuen Internet Protocols.

3.5 Änderungen zu IPv4 im Überblick

Die Neuerungen und Änderungen von IPv6 im Vergleich zu IPv4 lassen sich im wesentlichen unter folgenden Punkten zusammenfassen [RFC 1883]:

Erweiterung der Adreßmöglichkeiten Die Adreßgröße wird von 32 Bit auf 128 Bit erweitert, um mehrere Hierachieebenen, eine größere Anzahl an adressierbaren Systemen und eine einfachere Adreßautokonfiguration zu ermöglichen. Multicast-Adressen werden um ein Bereichsfeld erweitert, mit dem die Ausbreitung des Multicasts angegeben werden kann. Außerdem wird ein neuer *Anycast*-Adreßtyp eingeführt, mit dem ein Paket zu einem System aus einer Menge von Systemen gesendet werden kann.

Header-Format-Vereinfachung Einige IPv4-Header-Felder werden zu möglichen Header-Erweiterungen oder verschwinden ganz aus dem IPv6-Standard-Header, um die bei jedem Paket anfallenden Verarbeitungskosten zu reduzieren und um die Bandbreitenkosten für den IPv6-Header zu begrenzen.

Verbesserte Unterstützung von Erweiterungen und Optionen Durch Änderung des Verfahrens IP-Header-Optionen einzufügen, können mehr und auch längere Optionen angegeben werden. Außerdem ermöglicht dieses Verfahren eine größere Flexibilität beim Einführen von neuen Erweiterungen in der Zukunft.

Möglichkeit zur Flußkennzeichnung Es wird eine neue Möglichkeit eingeführt, Pakete eines speziellen Datenflusses, für die der Sender eine besondere Behandlung wünscht, zu kennzeichnen.

Möglichkeiten zur Erkennung der Echtheit und zur Geheimhaltung Erweiterungen zur Unterstützung der Authentizität, der Datenintegrität und der Geheimhaltung werden für IPv6 spezifiziert. Diese Optionen müssen alle IPv6-Implementierungen unterstützen.

3.6 IPv6-Adressen

Eine Schicht-2-Schnittstelle wird im weiteren Text zur besseren Lesbarkeit als „Interface“ bezeichnet. Aus dem gleichen Grund wurde der englische Begriff „Link“ dem Begriff

Schicht-2-Verbindung vorgezogen. Die Verwendung der Begriffes „link-lokal“ und „site-lokal“ für „Schicht-2-beschränkt“ und „auf einen administrativen Bereich beschränkt“ wurde ebenso wegen der besseren Wiedererkennung in der englischsprachigen Literatur vorgezogen.

3.6.1 Adreßraum

Dem drängendsten Problem von IPv4, dem zu kleinen Adreßraum, wird in IPv6 durch eine Vervierfachung der Adreßbits Rechnung getragen. IPv4-Adressen sind 32 Bit lang, während IPv6-Adressen 128 Bit lang sind.

Mit 32 Bit ließen sich bei optimaler Adreßvergabe theoretisch

4 294 967 296

Interfaces adressieren, bei 128 Bit theoretisch 2^{128}, also

340 282 366 920 938 463 463 374 607 431 768 211 456

verschiedene Interfaces. Da die Adreßvergabe durch administrative, technische und andere Gründe nie optimal sein kann (Hirarchiebildung, Vorsorge für Erweiterbarkeit, vgl. mit der Vergabe von Telefonnummern bei Telefongesellschaften), muß mit einer nicht optimalen Ausnutzung des Adreßraumes gerechnet werden. Bei einer halbwegs optimistischen Rechnung stehen mit IPv6 immer noch rund $3{,}9 \times 10^{18}$ Adressen pro m^2 Erdoberfläche zur Verfügung, und selbst bei einer sehr pessimistischen Rechnung noch über 1000 Adressen pro m^2 Erdoberfläche [Hin95]. Damit ist die Benutzung von IPv6-Adressen auch in noch nicht bekannten Anwendungsgebieten ohne Probleme möglich.

3.6.2 Adreßbereiche

Der Typ (siehe Seite 16) einer IPv6-Adresse wird durch die führenden Bits in der Adresse bestimmt. Die Belegung der führenden Bits sieht zunächst wie in Tabelle 1 angegeben aus [RFC 1884].

Insgesamt sind rund 30% des gesamten Adreßraumes von Anfang an belegt oder reserviert, für zukünftige Erweiterungen stehen noch 70% des Adreßraumes zur Verfügung.

Alle Adressen mit 0000 0000-Präfix sind für verschiedene Aufgaben, z.B. Einbettung des IPv4-Adreßraumes reserviert. Es gibt Überlegungen, IPX und NSAP-Adressen direkt als IPv6-Adressen zu unterstützen. Genaueres ist hier noch nicht festgelegt. Die wichtigeren Adreßtypen werden weiter unten (Seite 16) noch genauer vorgestellt.

3.6.3 Textuelle Repräsentation von IPv6-Adressen

Die Darstellung von IPv6-Adressen erfolgt nicht mehr im IPv4-üblichen „dotted decimal“-Format, da diese Schreibweise bei längeren Adressen doch sehr unübersichtlich und fehleranfällig wird. Eine IPv6-Adresse kann auf drei unterschiedliche Arten durch einen Textstring repräsentiert werden:

Verwendung	**Präfix** (binär)	**Teil des Adreßbereiches**
Reserviert	0000 0000	1/256
Nicht zugeordnet	0000 0001	1/256
Reserviert für NSAP(OSI)-Adressen	0000 001	1/128
Reserviert für IPX-Adressen	0000 010	1/128
Nicht zugeordnet	0000 011	
bis	001	29/128
Provider-basierte Unicast-Adressen	010	1/8
Nicht zugeordnet	011	1/8
Reserviert für geographisch basierte Unicast-Adressen	100	1/8
Nicht zugeordnet	101	
bis	1111 1110 0	189/512
Link-lokale Adressen	1111 1110 10	1/1024
Site-lokale Adressen	1111 1110 11	1/1024
Multicast-Adressen	1111 1111	1/256

Tabelle 1: Belegung des IPv6-Adreßraumes

1. **Allgemeine Darstellung:** die allgemeine Dastellung `x:x:x:x:x:x:x:x` besteht aus 8 Bereichen, welche durch Doppelpunkte voneinander getrennt sind. In jedem Bereich werden 16 Bit hexadezimal dargestellt. Beispiel:

   `68E6:8C64:FFFF:FFFF:0:1180:96A:FFFF`

   Es ist nicht nötig, führende Nullen in die einzelnen Bereiche zu schreiben, aber es muß auf jeden Fall eine Ziffer in jedem der 8 Bereiche stehen.

2. **Verkürzende Darstellung:** viele Adressen werden einen oder mehrere Bereiche haben, die aus Nullen bestehen. Ein einziger Nullen-Bereich in einer Adresse läßt sich unterdrücken, wobei zwei Doppelpunkte („::") mehrere Gruppen von 16 Bit Null-Bereichen anzeigen. Diese Schreibweise ist auch für führende oder abschließende Nullen gültig. Beispiel:

   `FF05:0:0:0:0:0:0:B3`

   kann verkürzt als

   `FF05::B3`

   geschrieben werden. Andere Beispiele:

   - `1989::8:800:200C:417A`
   - `::1` (Loopback-Adresse, siehe unten)
   - `::` (unspezifizierte Adresse, siehe unten)

3. **IPv4-unterstützende Darstellung:** gerade in einer gemischten Umgebung von IPv4 und IPv6 mag es günstig sein, eine andere Darstellung für die letzten 32 Bit

zu haben. Die Schreibweise `x:x:x:x:x:x:d.d.d.d` ist ebenfalls gültig, wobei die letzten 4 ´d´s mit dazugehörigen Punkten der Schreibweise für eine IPv4-Adresse entsprechen. Beispiel:

`0:0:0:0:0:0:13.1.68.3`

oder in komprimierter Schreibweise:

`::13.1.68.3`

3.6.4 Adreßtypen

IPv6-Adressen sind 128 Bit-Bezeichner für Interfaces und Mengen von Interfaces. Einem einzelnen Interface können eine oder mehrere IPv6-Adressen verschiedener Adreßtypen zugeordnet werden. Es gibt drei verschiedene Typen von Adressen [RFC 1884]:

1. Unicast-Adressen

2. Anycast-Adressen

3. Multicast-Adressen

Im Vergleich zu IPv4 sind Anycast-Adressen neu.

Unicast: Unicast-Adressen sind Bezeichner für ein einzelnes Interface. Ein Paket, welches an eine Unicast-Adresse geschickt wird, wird **genau an dem** Interface zugestellt, welches durch die Adresse spezifiziert wird. Mehrere Adressen pro Interface müssen unterstützt werden.

Provider-basierte globale Unicast-Adressen Das Format einer IPv6-Provider-basierten Adresse ist ähnlich der Adreßform von IPv4 unter Classless Inter-Domain Routing (CIDR) aufgebaut. Sie hat folgendes Schema:

3 Bits	n Bits	m Bits	o Bits	125-n-m-o Bits
010	Registry ID	Provider ID	Subscriber ID	Intra-Subscriber

Eine jeweils variable Anzahl von Bits erlaubt flexible Größenanpassungen der Bereiche.

Lokale IPv6 Unicast-Adressen Es gibt zwei Arten von lokalen Adressen: link-lokale und site-lokale Adressen. Link-lokale Adressen sind für den Einsatz in einem einzelnen physikalischen Netz gedacht, während Site-lokale Adressen für den Einsatz in einem einzelnen administrativen Bereich gedacht sind.

Link-lokale Adressen haben folgendes Format:

10 Bits	n Bits	118-n Bits
1111111010	0..0	Token-ID

Link-lokale Adressen sind nur in dem Netz gültig, an welches ein Interface direkt angeschlossen ist. Sie ermöglichen die Adressierung in einem einzelnen physikalischen Netz. Benutzt werden sie zur automatischen Konfiguration (siehe auch Seite 19), Nachbarnentdeckung, oder wenn keine Router vorhanden sind. Das Token muß so gewählt werden, daß dieses in dem physikalischen Netz eindeutig ist. Bei Ethernet wird üblicherweise die 48-bit-IEEE-802 MAC-Adresse für diesen Zweck herangezogen. Router dürfen keine Pakete mit link-lokalen Adressen weiterleiten, da sonst Adreßkonflikte auftreten könnten.

Site-lokale Adressen haben folgendes Format:

10 Bits	n Bits	m Bits	118-n-m Bits
111111111011	0...0	Subnetz-ID	Token

Site-lokale Adressen können z.B. von Organisationen benutzt werden, welche (noch) nicht mit dem Internet verbunden sind. Wenn später die Organisation an das Internet angeschlossen wird, können globale Internet-dressen gebildet werden, indem der Site-lokale Präfix durch einen Subscriber-Präfix ersetzt wird.

Anycast: Anycast-Adressen sind Bezeichner für eine Menge von Interfaces, welche typischerweise zu verschiedenen Systemen gehören. Ein Paket, welches an eine Anycast-Adresse geschickt wird, wird an **eines** der Interfaces zugestellt, welches durch die Adresse spezifiziert wird. Vom Standpunkt des Routingprotokolls gesehen ist dieses üblicherweise das „nächste“ Interface. Eine Anycast-Adresse ist ohne zusätzliche Informationen nicht von einer Unicast-Adresse zu unterscheiden.

Im Moment dürfen Anycast-Adressen keinem Endsystem zugeordent werden, nur zu Routern, da noch keine größere Erfahrung mit Anycast-Adressen vorliegt. Eine Anycast-Adresse darf nicht Quelle eines IPv6-Paketes sein.

Verschiedene Anycast-Adressen sind vordefiniert, z.B. die Subnetz-Router Anycast-Adresse. Hier definiert das Subnetz-Präfix mit angehängten Nullen die Adresse aller Subnetz-Router. Damit erreicht jedes Endsystem einen Router, so lange ein funktionsfähiger Router im Subnetz vorhanden ist.

Multicast: Multicast-Adressen sind Bezeichner für eine Menge von Interfaces, welche typischerweise zu verschiedenen Systemen gehören. Ein Paket, welches an eine Multicast-Adresse geschickt wird, wird an **alle** Interfaces zugestellt, welche durch die Adresse spezifiziert werden. Multicast-Adressen haben ein spezielles Format (von Uni/Anycast unterscheidbar), welches unten genauer beschrieben wird. Multicast-Adressen dürfen nicht als Quelle eines IPv6-Paketes auftreten.

Multicast-Adressen haben folgendes Format:

8 Bits	4 Bits	4 Bits	112 Bits
11111111	Flags	Wirkungsbereich	Gruppen-ID

Die führenden 8 Einsen definieren eine Adresse als Multicast-Adresse.
Die oberen drei Bits des Flags-Feldes sind reserviert. Mit Hilfe des untersten Flag-Bits kann definiert werden, ob eine Multicast-Adresse nur temporär zugeordnet ist (Bit=1), oder ob es eine dauerhaft durch die *Internet Assigned Numbers Authority* (IANA) festgelegte Multicast-Adresse ist (Bit=0).

Mit Hilfe der Bits im Wirkungsbereichs-Feld kann der Wirkungsbereich einer Multicast-Gruppe definiert werden. Tabelle 2 zeigt die vorhandenen Möglichkeiten [RFC 1884]:

Wert	**Wirkungsbereich**
0	reserviert
1	node-lokal
2	link-lokal
3/4	(nicht zugeordnet)
5	site-lokal
6/7	(nicht zugeordnet)
8	organisations-lokal
9 bis D	(nicht zugeordnet)
E	global
F	reserviert

Tabelle 2: Mögliche Multicast-Wirkungsbereiche

Die Wirkung einer durch IANA festgelegten Multicast-Adresse ist unabhängig vom Bereich. Dies erlaubt eine flexible Erreichbarkeit.
Beispiel: angenommen, die *Network Time Protocol*-Servergruppe habe eine festgelegte Multicast-Adresse mit dem hexadezimalen Wert 43. Dann erreicht

- `FF01:0:0:0:0:0:0:43` alle NTP-Server in dem Rechner des Senders
- `FF02:0:0:0:0:0:0:43` alle NTP-Server in dem physikalischen Netz des Senders
- `FF05:0:0:0:0:0:0:43` alle NTP-Server in dem site-lokalen Bereich des Senders
- `FF0E:0:0:0:0:0:0:43` alle NTP-Server des Internets

Die Wirkung einer nur temporär zugeordneten Multicast-Adresse ist immer nur im angegebenen Bereich gültig.

Multicast-Gruppen werden mit Hilfe eines eigenen Protokolls, des *Internet Group Management Protocols* (IGMP) [Fen96] gebildet, verändert und aufgelöst.

3.6.5 Spezielle Adressen

Die unspezifizierte Adresse Die Adresse `0:0:0:0:0:0:0:0` wird die unspezifizierte Adresse genannt. Diese darf nie einem System zugeordnet werden. Sie zeigt die Abwesenheit einer anderen Adresse an. Sie wird z.B. von sich initialisierenden Systemen benutzt. Diese geben die unspezifizierte Adresse als Quelladresse von IPv6-Paketes an, so lange die eigenen Adresse noch nicht bekannt ist.

Loopback-Adresse Die Adresse `0:0:0:0:0:0:0:1` wird die Loopback-Adresse genannt. Diese kann von einem System genutzt werden, um IPv6-Datagramme an sich selbst zu schicken. Sie entspricht der IPv4-Adresse `127.0.0.1`. Die Loopback-Adresse darf ebenfalls nie einem System zugeordnet werden.

IPv4-kompatible IPv6-Adressen IPv6-Systeme, welche mit IPv4 kommunizieren wollen, benötigen eine IPv4-kompatible Adresse. Diese sieht wie folgt aus:

80 Bits	16 Bits	32 Bits
0000.......0000	0000	IPv4-Adresse

Wenn ein IPv4/IPv6-System mit einem nur-IPv4-System kommuniziert, werden die hintersten 32 Bit als IPv4-Adresse verwendet.

IPv4-Adressen als IPv6-Adressen Falls in einem IPv6-System Adressen eines nur-IPv4-Systemes dargestellt werden sollen, werden zur Unterscheidung von IPv4-kompatiblen IPv6-Adressen 16 Bits vor der IPv4-Adresse auf eins gesetzt.

80 Bits	16 Bits	32 Bits
0000.......0000	FFFF	IPv4-Adresse

Damit ist immer klar, ob eine Adresse eine IPv6-Adresse ist, oder ob die Adresse nur als IPv4-Adresse verwendet werden darf.

3.6.6 Autokonfiguration

Ein Ziel bei der Entwicklung von IPv6 war es, die Konfiguration von Endsystemen zu erleichtern. Zwei Verfahren, ein zustandsloses und ein dynamisches, unterstützen die automatische Konfiguration. Mit Hilfe dieser Verfahren erfahren Endsysteme ihre Adresse(n) sowie weitere Konfigurationsparameter.

Die beiden Verfahren unterscheiden sich in den externen Einflußmöglichkeiten. Das zustandslose Verfahren ermöglicht einfaches „plug and play“. Das aufwendigere dynamische Verfahren erlaubt eine genauere Konfiguration (z.B. Festlegung der verwendeten IP-Adressen) durch Systemadministratoren.

Beide Verfahren können kombiniert werden; ein Teil der benötigten Konfigurationsinformationen kann durch das zustandslose Verfahren erfahren werden, während andere Parameter durch das dynamische Verfahren konfiguriert werden können.

Das Autokonfigurationsverfahren wird durch das Setzen spezieller Felder in Router-Mitteilungen (Router Advertisements) bestimmt.

Gemeinsamkeiten IPv6-Adressen werden einem Interface für eine feste Zeitspanne zugewiesen. Wenn die Lebenszeit einer Adresse abläuft, wird die Zuordnung (und die Adresse) ungültig. Die Adresse kann nun an ein anderes Interface vergeben werden. Um dieses Ablaufen von Adreßbindungen reibungsloser zu gestalten, durchläuft eine Adresse zwei verschiedene Phasen, während sie einem Interface zugeordnet ist. Zuerst ist eine Adresse vorzuziehen (preferred), welches bedeutet, daß ihre Benutzung keiner Restriktion unterliegt. Später wird eine Adresse dann als ungünstig eingestuft (deprecated). Dies kündigt an, daß die Adresse in Zukunft ungültig werden wird. Die Benutzung der Adresse ist zwar immer noch erlaubt, aber von der Verwendung wird abgeraten. Neue Verbindungen sollten, falls möglich, immer eine vorzuziehende Adresse benutzen.

Systeme stellen durch einen speziellen Test sicher, daß Adressen einmalig sind, bevor sie sie an ein Interface vergeben. Dieser Test wird unabhängig von dem Konfigurationsverfahren durchgeführt. Falls eine doppelte Adreßbenutzung erkannt wird, stoppt die automatische Konfiguration; die Konfiguration muß manuell durchgeführt werden.

Zustandslose Autokonfiguration Bei der zustandslosen Autokonfiguration werden keine Informationen über vergebene Adressen gespeichert. Zustandslose Autokonfiguration benötigt keine manuelle Konfiguration an den Endsystemen. Die Konfiguration beschränkt sich auf eine minimale (wenn überhaupt benötigte) Konfiguration der Router. Zusätzliche Server werden nicht benötigt.

Das zustandslose Verfahren erlaubt einem Endsystem seine eigenen Adressen zu erzeugen. Dies geschieht durch eine Kombination von lokal verfügbaren Informationen und Informationen, die durch Router verkündet werden. Router verkünden regelmäßig mit ihrer Existenz Adreß-Präfixe, welche das oder die Subnetze identifizieren, an denen ein Endsystem angeschlossen ist. Systeme kombinieren ein lokal eindeutiges Token (z.B. die Ethernet-Adresse) und die Adreßpräfixe, um gültige IP-Adressen zu erzeugen. Durch die Routermitteilungen wird festgelegt, welche anderen Informationen automatisch konfiguriert werden sollen. Um nicht auf diese Nachricht warten zu müssen, kann ein System auch eine spezielle Nachricht an alle verfügbaren Router schicken.

Wenn kein Router verfügbar ist, kann ein System nur link-lokale Adressen generieren. Dies reicht jedoch aus, um mit anderen an dem gleichen Netzsegment angeschlossene Systemen zu kommunizieren.

Genauere Informationen zur statischen System-Konfiguration können [Thom96] entnommen werden.

Dynamische Autokonfiguration Mit Hilfe eines speziellen Protokolls, des *Dynamic Host Configuration Protocols* (DHCPv6) [Bou96], wird ein dynamisches Client/Server-Konfigurationsverfahren ermöglicht. Notwendig hierfür ist der Einsatz eines oder mehrer dezidierter DHCP-Server. Diese speichern Informationen über zu konfigurierende Endsysteme.

Die dynamische System-Konfiguration erfordert nicht unbedingt einen DHCP-Server in jedem Netzsegment. „Relay-Agents“ (Router) können eingesetzt werden, die Nachrichten von und zu zentralen DHCP-Servern weiterzuleiten.

Genauere Informationen zum DHCP-Protokoll für IPv6 können [Bou96] entnommen werden.

3.7 Der IPv6 Header

Die auf Seite 3.2 genannten Entwicklungsziele wirkten sich stark auf den IPv6-Header aus. Dort lassen sich folgende Änderungen zu einem IPv4-Header feststellen:

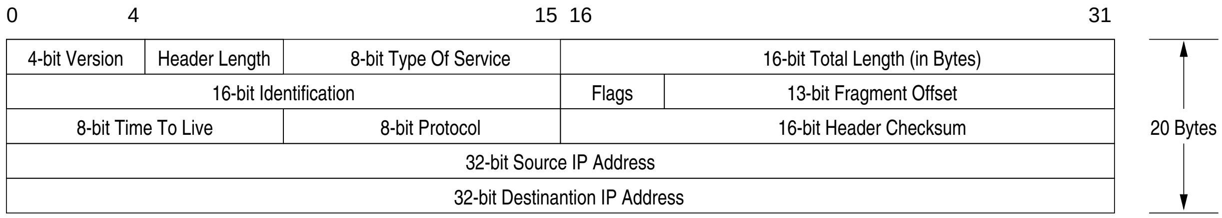

Abbildung 3: Der IPv4-Header

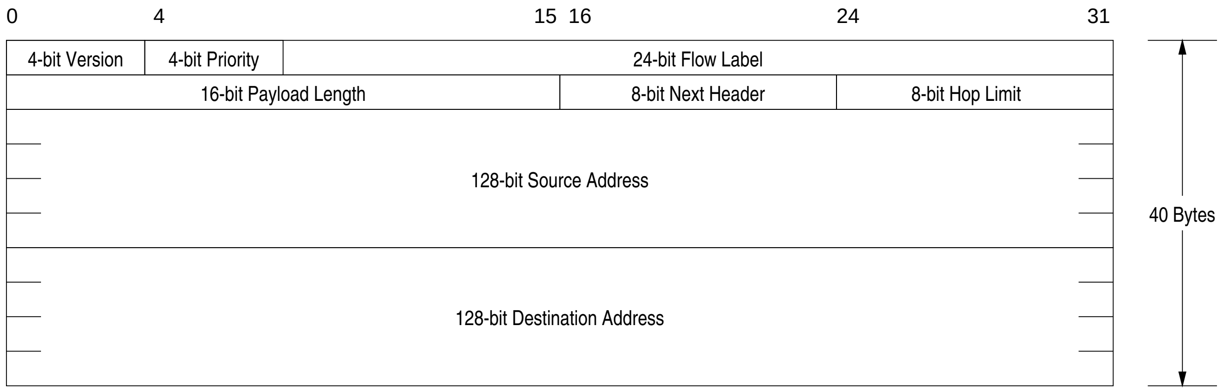

Abbildung 4: Der IPv6-Header

- Das Feld *Header Length* wurde fallen gelassen, da der Standard-Header eine feste Länge von 40 Bytes hat. An diesen können vor den Headern der höheren Protokollschichten weitere optionale Header-Erweiterungen eingefügt werden.

  Dies ermöglicht Header, die insgesammt länger als 60 Byte (IPv4) sind.

- Das Feld *Type Of Service* wurde durch das *Priority*-Feld ersetzt (siehe 3.7.1).

- Ein neues Feld *Flow Label* wurde eingeführt (siehe 3.7.2).

- Im Feld *Payload Length* wird nur die Länge der Nutzdaten ohne den Header angegeben.

- Fragmentierung sollte bei IPv6 generell vermieden werden. Sie wird jetzt in Form einer Header-Erweiterung realisiert. Dies macht die Felder *Identification*, *Flags* und *Fragment Offset* im Standard-Header überflussig.

  Weiterhin ist es im Unterschied zu IPv4 nur der Quelle, nicht aber den an der Übertragung beteiligten Routern erlaubt, eine Fragmentierung einzuleiten. Deshalb soll die *Maximum Transmission Unit* (MTU) vor dem Versenden mittels *Path MTU Discovery* [McC96] ermittelt werden. Dabei handelt es sich um ein Verfahren mit dem unter Zuhilfenahme des *Internet Control Message Protocol* (ICMP) [RFC 1885] die MTU der gesamten Übertragungsstrecke ermittelt wird. Dies ermöglicht es den meisten Protokollen auf höheren Schichten, die Paketlänge so anzupassen, daß keine Notwendigkeit besteht zu fragmentieren.

  Außerdem wird implizit in jedem Netz eine MTU von 576 Byte vorausgesetzt. Dort, wo diese Annahme nicht erfüllt ist, muß mit geeigneten Mechanismen auf Schicht 2 fragmentiert werden.

  Die hohen Verarbeitungskosten, die durch Fragmenierung entstehen, sollen auf diese Weise vermieden werden.

- Das Feld *Time To Live* wird zum *Hop Limit*-Feld. Jeder Vermittlungsrechner, welcher ein Paket weiterleitet, erniedrigt das Feld um 1. Ein Paket wird nicht weiter übertragen, falls sein Hop Limit 0 wird.

  Dies wurde unter IPv4 auch schon so gehandhabt, obwohl laut Standard [RFC 791] eigentlich die Zeit in Sekunden angegeben werden sollte, so daß es genaugenommen nur einer Umbenennung hin zur realen Verwendung des Feldes gleichkommt.

- Das Feld *Protocol* wird zum *Next Header*-Feld. Hier wird der Typ des Headers eingetragen, der dem IPv6-Standard-Header unmittelbar folgt (siehe 3.8).

- Das Feld *Header Checksum* entfällt.

  Es ist zur Zeit auch kein anderer Mechanismus, insbesondere als Header-Erweiterung, vorgesehen. Indirekt könnte man dies jedoch mit dem Authentication-Header verwirklichen (siehe 3.8.4).

  Damit werden Abarbeitungskosten, die bei IPv4 mit jedem Paket entstehen, eingespart. Das Weglassen dieser Überprüfung ist möglich, da heutzutage die Schicht-2-Verbindungen sicherer sind, oder zumindest in ihrer Schicht Verfahren implementieren, um vorhandene Übertragunsfehler zu erkennen. Zudem findet üblicherweise in einer höheren Protokollschicht die Überprüfung aller Daten statt [RFC 1883].

- Die Adreßfelder werden von 32 Bit auf 128 Bit erweitert.

- Die Ausrichtung des Headers wird von 32 Bit auf 64 Bit geändert.

  Dies soll mit neuen 64-bit-Architekturen in Netzknoten eine Effizienzsteigerung ermöglichen.

Durch diese Änderungen bzw. Vereinfachungen ist es außerdem gelungen, die Länge des Standard-IPv6-Headers nur zu verdoppeln trotz einer Vervierfachung der Adreßlänge.

3.7.1 Prioriäten

Mit dem *Priority*-Feld ist es einer Quellstation möglich, die Beförderungspriorität seiner Pakete relativ zu seinen anderen Paketen zu bestimmen. Die Prioritätswerte sind in zwei Bereiche unterteilt: Werte von 0 bis 7 werden benutzt, um Datenverkehr zu kennzeichnen, bei dem die Quelle Flußregelung bereitstellt (z. B. TCP). Werte von 8 bis 15 werden verwendet, um Datenverkehr zu kennzeichnen, bei dem Daten mit einer konstanten Übertragungsrate übertragen werden (z. B. Echtzeit-Anwendungen).

Die empfohlenen Prioritätswerte spezieller Anwendungskategorien sind für flußgesteuerte Übertragungen in Tabelle 3 aufgeführt.

Priorität	Anwendungsbereich
0	nicht charakterisierter Datenverkehr
1	„Füll“-Übertragungen (z. B. netnews)
2	Anwendungen mit kleinem Datenverkehr (z. B. email)
3	(reserviert)
4	Anwendungen mit großem Datenverkehr (z. B. FTP, NFS)
5	(reserviert)
6	interaktive Anwendungen (z. B. telnet, X)
7	Internet Kontrollverkehr (z. B. Routing-Protokolle, SNMP[1])

Tabelle 3: Empfohlene Prioritäten für verschiedene Anwendungen nach [RFC 1883]

Bei nicht flußgesteuerten Übertragungen sollte der niedrigste Prioritätswert (8) verwendet werden, falls Daten bei Überflutung verworfen werden können (z. B. bei hochwertigen Videoübertragungen). Der höchste Prioritätswert (15) soll dann angewendet werden, wenn möglichst keines der gesendeten Pakete verloren gehen soll (z. B. bei minderwertigen Audioübertragungen). Zwischen den Prioritäten für flußgesteuerte und nicht flußgesteuerte Übertragungen existiert keine implizite Ordnung. D. h. Pakete mit Prioritäten aus dem einen Bereich dürfen nicht mit Paketen, die eine Priorität aus dem anderen Bereich haben, in Relation gesetzt werden.

Durch dieses Feld im Header haben Endsysteme und vor allem Router eine Möglichkeit den Vorrang von Paketen (im gleichen Prioritätsbereich) aus dem aktuellen Datenverkehr zu ersehen.

[1]Simple Network Management Protocol

3.7.2 Flußkennzeichnung

Das Feld *Flow Label* im IPv6-Header kann von einer Quelle zur Flußkennzeichnung von Paketen benutzt werden. Als Fluß wird hierbei eine Menge von Paketen bezeichnet, die von einer einzelnen Quelle zu einem bestimmten Ziel[2] gesendet werden, für die die Quelle eine spezielle Behandlung durch die IPv6-Router wünscht. Dies kann sie z. B. bei besonderen Quality-of-Service-Merkmalen oder Echtzeit-Service verwenden. Die Art der speziellen Behandlung kann dabei über ein eigenes Protokoll, wie z. B. dem *Resource Reservation Protocol* (RSVP), oder über IPv6-Header-Erweiterungen (siehe 3.8) geschehen, wie z. B. mit dem *Hop-by-Hop Options*-Header (siehe 3.8.2).

Der Einsatz dieses Merkmals des IPv6-Protokolls wird im Moment noch experimentell erpropt. Aussagen über die Verwendung des Flow Labels sind erst möglich, wenn man genaue Aussagen über die Notwendigkeit der Flußsteuerung im Internet machen kann. Die Entwicklung des Resource Reservation Protocols, welches zur Zeit als Draft [Bra96b] vorliegt, wird hierzu einen Teil beitragen.

Systeme, die die Funktion des Flow Labels nicht unterstützen, müssen es auf null setzen, wenn sie Pakete erzeugen, bzw. unverändert lassen, wenn sie Pakete weiterleiten, und ignorieren, wenn sie Pakete empfangen.

Anwendung Das Flow Label wird von der Quelle des Datenstroms den Paketen zugewiesen. Es können mehrere aktive Flow Labels zwischen einer Quelle und einem Ziel existieren, wie auch Datenverkehr, der keinem bestimmten Fluß zugeordnet ist. Ein Datenstrom wird genau gekennzeichnet durch die Kombination der Quelladresse und dem Flow Label, welches ungleich null ist. Pakete, die zu keinem Datenfluß gehören, haben ein Flow Label von null.

Neue Flow Labels müssen pseudo-zufällig und gleichverteilt im Bereich von 1 bis FFFFFF ausgewählt werden. Dies hat den Sinn, daß es Routern möglich sein sollte, das Flow Label als Hash-Schlüssel zu benutzen, um den mit dem Fluß verbunden Status zu finden. Aus diesem Grund darf eine Quelle auch ein Flow Label nicht verwenden, solange das alte noch Gültigkeit hat. Die Gültigkeit endet 6 Sekunden nach dem letzten Paket.

Alle Pakete eines Datenflusses müssen mit der gleichen Quelladresse, Zieladresse, Priorität und Flow Label versendet werden. Enthalten Pakete einen Hop-by-Hop Options-Header, so muß dieser bei allen Paketen den gleichen Inhalt haben. Bei Verwendung eines Routing Headers müssen alle Optionen bis einschließlich des Routing Headers gleich sein. Router können, müssen aber nicht, diese Bedingungen überprüfen, und sollten bei Entdeckung eines Fehlers eine ICMP-Fehlermeldung an den Sender schicken.

3.8 Die Header-Erweiterungen

Mit IPv6 wird ein neuer Mechanismus eingeführt, mit dem zwischen den Standard-IPv6-Header und denen der höheren Protokollschichten zusätzliche Header(-Erweiterungen) pla-

[2] Im Moment sind nur Unicast oder Multicast erlaubt.

ziert werden können. Die Art des folgenden Headers wird im *Next Header*-Feld angegeben. Wie in Abbildung 5 gezeigt, kann ein IPv6-Paket keine, eine oder mehrere Header-Erweiterungen enthalten.

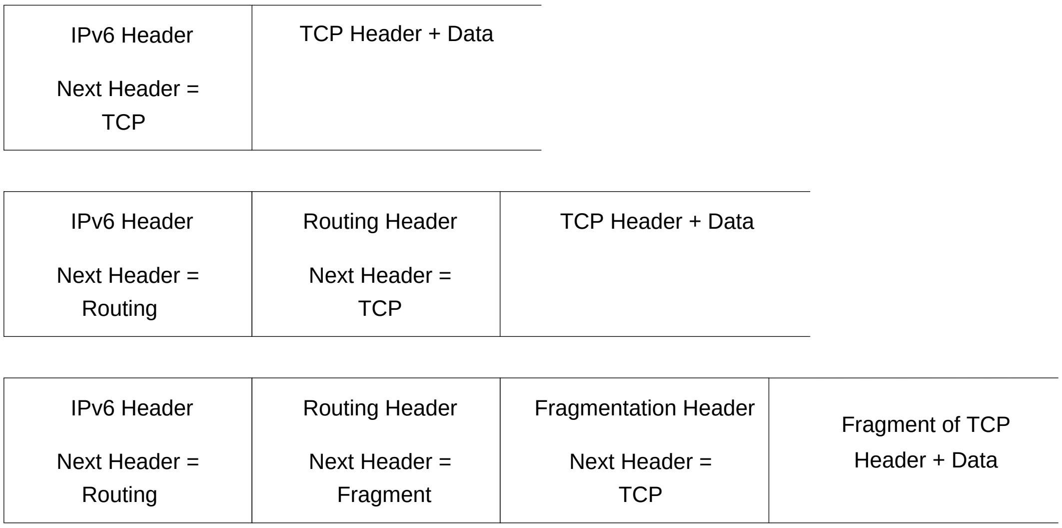

Abbildung 5: Anordung von Header-Erweiterungen

Mit Ausnahme des Hop-by-Hop Options-Headers werden die zusätzlichen Header nicht untersucht oder abgearbeitet bis das Paket an seinem Ziel angekommen ist. Ziel eines Paketes können ein oder mehrere Endgeräte sein, je nachdem ob eine Unicast-, Anycast- oder Multicast-Adresse angegeben wurde. Durch diese Vorgabe werden die Prozeßkosten für Pakete mit Header-Erweiterungen niedrig gehalten.

Normalerweise wird das Next Header-Feld des IPv6-Headers entschlüsselt und dann die erste Header-Erweiterung verarbeitet bzw. der Header einer höherliegenden Protokollschicht, falls keine Option angegeben wurde. Anhand des Inhalts und der Semantik jeder Header-Erweiterung wird entschieden, ob der nächste Header abgearbeitet wird oder nicht. Daher müssen die Header-Erweiterungen immer in der Reihenfolge ihres Auftretens abgearbeitet werden. Es ist nicht erlaubt, alle Header zu lesen und die Ausführungsfolge zu verändern.

Eine andere Verbesserung zu IPv4 ist, daß die IPv6-Header-Erweiterungen eine beliebige Länge haben können und daß es keine Beschränkung bezüglich der gesamten Länge von Optionen auf insgesamt 40 Bytes gibt.

Durch diese Verbesserungen können mit IPv6 Optionen verwendet werden, die in IPv4 in der Praxis nicht, oder zumindest nicht zusammen, eingesetzt werden können. Ein Beispiel hierfür sind die IPv6 Authentifizierungs- und Geheimhaltungsoptionen und die Routing Option.

Um die Abarbeitungsgeschwindigkeit von nachfolgenden Headern und des darauffolgenden Transportprotokolls zu erhöhen, sind alle Erweiterungen ein Vielfaches von 64 Bits lang.

Im Augenblick sind für IPv6-konforme Implementierungen die in Tabelle 4 aufgelisteten Header-Erweiterungen definiert [RFC 1883].

Header-Erweiterung	**Funktion**
Routing (Type 0)	Erweitertes Routing (wie IPv4 Source Routing)
Fragmentation	Fragmentierung und Reassemblierung
Authentication	Authentifizierung und Integrität
Encapsulating Security Payload	Geheimhaltung
Hop-by-Hop Options	Spezielle Optionen, welche von jedem Rechner abgearbeitet werden müssen
Destination Options	Optionale Informationen, welche nur vom Zielrechner untersucht werden müssen

Tabelle 4: Standard-Header-Erweiterungen

3.8.1 Verwendung der Erweiterungen

Enthält ein Paket mehr als eine Header-Erweiterung, so sollten diese in folgender Reihenfolge verwendet werden:

1. IPv6-Header
2. Hop-by-Hop Options-Header
3. Destination Options-Header
4. Routing-Header
5. Fragmentation-Header
6. Authentication-Header
7. Encapsulating Security Payload-Header
8. Destination Options-Header
9. Header höherer Protokollschichten

Jeder Header-Typ, bis auf die Ausnahme des Destination Options-Header, sollte nur einmal in einem Paket enthalten sein. Dieser kann maximal zweimal verwendet werden, einmal vor dem Routing-Header und einmal vor dem Header der höheren Protokollschicht.

Bei Verwendung vor dem Routing-Header muß jedes System, welches durch eine Adresse dort referenziert wird, diese Option (es können auch mehrere sein) verarbeiten. Das Paket kann darüberhinaus noch über weitere Vermittlungsysteme zugestellt werden (Loose Source

Routing), die nicht im Routing-Header aufgeführt sind. Diese dürfen die Option jedoch nicht verarbeiten.

Beim Hop-by-Hop Options-Header hingegen muß jedes System, welches das Paket empfängt, die dort angegebene Option beacheten. Wird der Destination Options-Header vor der höheren Protokollschicht plaziert, so muß nur das Zielsystem die Option verarbeiten.

Wird eine neue Header-Erweiterung definiert, so muß für diese die Einordnung relativ zu den bisher definierten spezifiert werden.

3.8.2 Optionen für Hop-by-Hop Options- und Destination Options-Header

Im Hop-by-Hop Options-Header und Destination Options-Header werden optionale Informationen transportiert, die von bestimmten Systemen auf dem Weg zum Zielsystem untersucht werden müssen. Beide haben folgendes Format:

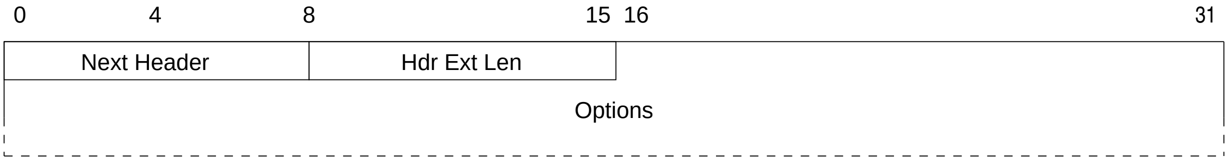

Abbildung 6: Hop-by-Hop und Destination Options-Header

Das *Next Header*-Feld hat hier die übliche Bedeutung, indem es den Typ des folgenden Headers spezifiziert, und das *Hdr Ext Len*-Feld gibt die Länge des Headers in 64-bit-Einheiten an. Da die Header immer auf 64 Bit ausgerichtet werden, werden die ersten 64 Bit nicht gezählt.

Danach werden eine oder mehrere Optionen mittels des *type-length-value*-Verfahrens (TLV) angegeben (Abbildung 7).

Abbildung 7: Angabe von Optionen mittels des TLV-Verfahrens

Wie der Name schon suggeriert, wird zuerst im *Option Type*-Feld die Art der Option, im *Option Data Length*-Feld die Länge dieser in Bytes und schließlich noch die eigentliche Option angegeben.

Im Option Type-Feld ist noch implizit mit den höchstwertigen 3 Bits angegeben, wie ein System verfahren soll, wenn es die angegebene Option nicht versteht, und ob sich in dieser Option die Daten während einer Übertragung ändern können. Dies ist wichtig für Verschlüsselungsalgorithmen, die gegebenenfalls dann diese Option aussparen müssen.

Zur Zeit sind für beide nur zwei *Padding*-Optionen, welche eine Ausrichtung in Byte-Schrittweite erlauben, und für den Hop-by-Hop Options-Header noch zusätzlich die *Jumbo Payload*-Option (Abbildung 8) spezifiziert.

0 4 8	15	16	31
		194	4
Jumbo Payload Length			

Abbildung 8: Jumbo Payload Option

Mit dieser ist es möglich Pakete zu verschicken, die mehr als 65535 Byte Nutzdaten enthalten. Dies stellt eine sinnvolle Erweiterung für die Übertragung von sehr großen Datenmengen dar. Eine gleichzeitige Verwendung der Jumbo Payload Option und eines Fragmentation-Headers ist nicht erlaubt, da sich dies widerspricht.

Konkret würde ein Hop-by-Hop Options-Header mit Jumbo Payload Option, wie in Abbildung 9 dargestellt, verwendet werden.

0 4	8 15	16	31
Next Header	0	194	4
Jumbo Payload Length			

Abbildung 9: Hop-by-Hop Options-Header mit Jumbo Payload Option

3.8.3 Routing

Das Routing von IPv6 ist fast identisch zu dem in IPv4 unter CIDR (siehe 2.2). Ausgenommen ist die Tatsache, daß der Adreßraum 128 Bit groß ist. Mit wenigen Erweiterungen sollten alle IPv4-Routingalgorithmen (OSPF, RIP, IDRP, ISIS usw.) auch unter IPv6 verwendet werden können. IPv6 beinhaltet auch einfache Routingerweiterungen, mit denen neue Funtionalitäten erreicht werden. Diese Möglichkeiten beinhalten:

- Auswahl von Dienstanbietern (Provider Selection) (z.B. anhand der Kosten, Geschwindigkeit usw.)
- Systemmobilität (routen zur aktuellen Lokation)
- Automatische Readressierung (routen zur neuen Adresse)

Die neue Routing-Funktionalität wird erreicht, indem eine Sequenz von IPv6-Adressen mittels des *IPv6 Routing*-Headers angegeben wird. Der Routing Header wird von einer Quelle benutzt, um zwischengelagerte Systeme (oder topologische Gruppen) anzugeben, welche von einem Paket auf dem Weg zum Zielsystem „besucht“ werden müssen. Diese Funktionalität ist sehr ähnlich der von IPv4 *Loose Source and Record Route Option* und *Strict Source and Record Route Option* [RFC 791]. Der Unterschied besteht darin, daß nicht für die Routing-Option generell angegeben werden muß, ob das Loose oder Strict Verfahren verwendet werden soll, sondern daß die Angabe für jede Adresse gesondert möglich ist.

Oft ist ein System daran interessiert auf Pakete zu antworten. Daher ist es meistens erforderlich, daß ein System die angegebene Reihenfolge der Adressen umkehrt, um so Pakete zum Absender zurückschicken zu können. Dieses Vorgehen ist in IPv6 der Standardfall. Will man sichergehen, daß die Adressen während der Übertragung nicht verändert worden sind, so kann man zusätzlich den *Authentication*-Header (siehe 3.8.4) benutzen.

Diese grundsätzliche Vorgehensweise ist nötig, damit Systeme solche neuen Möglichkeiten wie die Auswahl von Dienstanbietern und Verwendung der neuen Adreßmöglichkeiten anwenden können.

Beispiele In den drei folgenden Beispielen soll gezeigt werden, wie man die Adreßsequenzen verwenden kann. Hierbei wird eine Adreßsequenz durch einzelne Adressen, getrennt durch Kommata, dargestellt. Zum Beispiel:

SRC, I1, I2, I3, DST

Die erste Adresse ist dabei die Quelladresse, die letzte die Zieladresse und die mittleren sind Adressen von Zwischenstationen.

Nehmen wir an, zwei Systeme H1 und H2 (Abbildung 10) wollen miteinander kommunizieren und sind mit den Providern P1 und P2 verbunden. Ein dritter Provider PR, für drahtlose Systeme, ist mit den beiden anderen Providern P1 und P2 verbunden.

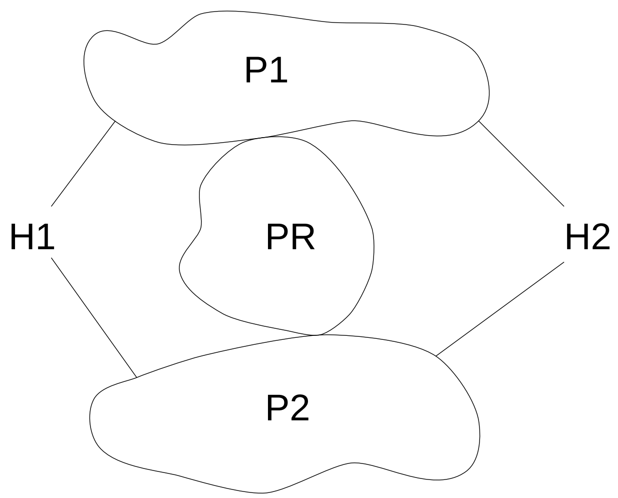

Abbildung 10: Beispiel-Netzwerk

Im einfachsten Fall, also ohne Verwendung von Adreßsequenzen, enthält ein Paket die Adressen:

H1, H2

Antwortet H2 auf die Pakete, so dreht H2 die Adreßfolge um und schickt ein Paket mit den Adressen

H2, H1

zurück. In diesem Beispiel könnte irgendein Provider für die Übertragung der Pakte verwendet werden, ohne daß H1 oder H2 eine Möglichkeit hätten zu bestimmen, wie die Pakete geroutet werden.
Will H1 bestimmen, daß jede Kommunikation zu und von H2 nur über den Provider P1 erfolgen soll, so konstruiert er ein Paket mit folgender Adreßsequenz:

H1, P1, H2

Dabei ist sichergestellt, daß H2 bei einer Antwort zu H1 die umgekehrte Route verwendet und die Pakete auch über P1 schickt. Die Adreßfolge in den Antwortpaketen würde folgendermaßen aussehen:

H2, P1, H1

Allerdings könnte es auch Fälle geben, in denen H2 zuerst die Pakete über den Provider P2 schicken möchte, bevor sie zu P1 und schließlich zu H1 gelangen. Hierbei würde H2 abweichend vom Standardfall die Adressen folgendermaßen angeben:

H2, P2, P1, H1

Wird H1 mobil und wechselt zum Provider PR, so kann er die Kommunikation mit H2 aufrecht erhalten, sogar ohne eine Verbindung zu unterbrechen, indem er Pakete mit der neuen Adreßsequenz schickt:

H1, PR, P1, H2

Diese Adreßsequenz stellt bei einer Antwort von H2 sicher, daß die Verfahrensweise von H1 übernommen wird, erst die Pakete zu P1 und dann zum neuen Provider PR von H1 zu schicken. Dies geschieht mit der Adreßfolge:

H2, P1, PR, H1

Wie man aus den Beispielen entnehmen kann, wird durch die Angabe von Adreßsequenzen mit dem IPv6 Routing Option Header die Auswahl von Dienstanbietern, Mobilität und Readressierung ermöglicht. Dies ist ein einfaches, aber trotzdem mächtiges Verfahren. Hierbei könnte insbesondere der neue Anycast-Adreßtyp verwendet werden, mit dem es Dienstanbietern ermöglicht wird, die Adressen für Gateways zu oder von anderen Netzen zu vereinheitlichen.

3.8.4 Sicherheit

Es werden zwei Möglichkeiten angeboten um Sicherheitsaspekte zu unterstützen. Jede wird durch einen eigenen Header realisiert.

Der Authentication-Header (AH) ist entwickelt worden, um Integrität und Echtheit eines IP-Datagrammes überprüfen zu können [RFC 1826]. Geheimhaltung der Daten ist damit nicht realisiert. Dadurch wird erhofft, daß der Authentication-Header in weiten Teilen des Internets Verbreitung findet, auch in Gebieten und Ländern, in denen Geheimhaltung mittels Verschlüsselung verboten ist.

Mit dem Authentication-Header kann man Sicherheit zwischen Endsystemen, zwischen zwei oder mehreren Gateways und einer Menge von Endsystemen oder Gateways realisieren.

Ein *Security Gateway* ist ein System, welches den Kommunikationszugang zu externen nicht glaubwürdigen und internen glaubwürdigen Systemen herstellt. Hierbei ist gedacht, daß das Security Gateway den vertrauenswürdigen Systemen einen Sicherheitsdienst anbietet, um mit nicht vertrauenswürdigen externen Partnern zu kommunizieren. Von dem vertrauenswürdigen Teilnetz muß natürlich angenommen werden können, daß sich alle Systeme (wie Endgerät und Router) vertrauen, aber auch, daß die unterliegenden Kommunikationkanäle nicht angegriffen werden können. Ist ein solches Securtiy Gateway vorhanden, können alle internen Systeme den Sicherheitsdienst in Anspruch nehmen und brauchen daher den Authentication-Header nicht zu implementieren.

Dies wird wohl am häufigsten bei IPv4-Systemen der Fall sein, da IPv6 die Unterstützung des Authentication-Headers verpflichtend vorschreibt. Andererseits kann durch das Gateway im internen Netz auf die Verwendung des Authentication-Headers verzichtet werden, um dadurch die Protokollast zu veringern. Trotzdem besteht so bei Kommunikation nach außen über das Gateway die Möglichkeit, sichere Verbindungen zu unterhalten.

Der Encapsulating Security Payload-Header (ESP) sichert in erster Linie die Geheimhaltung, aber mit gewissen Algorithmen kann er zusätzlich auch Integrität und Echtheit der Daten sichern [RFC 1827]. Aus gesetzlichen Gründen darf dieser Mechanismus oder der verwendete kryptographische Algorithmus aus manchen Staaten nicht exportiert und in anderen Staaten nicht importiert verwendet werden.

Auch hier bietet es sich an, Security Gateways zu verwenden. Damit kann man über ein unsicheres Weitverkehrsnetz, wie das Internet, ein virtuelles privates Netz bilden. Dies stellt zwar keinen Ersatz für die Geheimhaltung von End- zu Endsystem dar, aber dazu kann der Encapsulating Security Payload-Header zusätzlich von diesen verwendet werden.

Allgemeines Beide Header sind fester Bestandteil der IPv6-Spezifikation. Trotzdem ist man bemüht, eine Lösung zu finden, mit der auch IPv4-Implementation diese Funktionalität benutzen können. In beiden Möglichkeiten kann der kryptographische Alogrithmus ausgetauscht werden, ohne daß andere Teile der Implementation verändert werden müssen.

Ein *Key Management*-Protokoll ist noch nicht vereinbart, aber Vorschläge existieren bereits. Die beiden Header Typen sind nur durch den *Security Parameters Index* (SPI) mit dem Key-Management verbunden. Daher kann man sie defineren, ohne den Mechanismus zu kennen [RFC 1825].

Beide Header-Typen können kombiniert werden. Im Unterschied zu IPv4 müssen alle

IPv6-Systeme diesen Dienst mindestens mit dem MD5-Algorithmus mit einem 128-bit-Schlüssel unterstützen. Optional ist zur Zeit die Unterstützung für andere Algorithmen, obwohl sicher ist, daß MD5-Verschlüssselung sehr hohe Verarbeitungskosten erzeugt und somit die Übertragung merklich verlangsamt.

Will man die Echtheit der Datagramme auch von Zwischenstationen überprüfen lassen, so muß beachtet werden, daß die Pakete nicht fragmentiert werden, z. B. bei zwischenliegenden IPv4 Stationen. (Anwendung von Path MTU Discovery)

Bei IPv6 ist der Authentication-Header normalerweise nach dem Hop-by-Hop und vor dem IPv6 Destination Options angeordnet.

4 Migration von IPv4 nach IPv6

4.1 Anforderungen

Das Internet Protocol ist einer der wichtigsten, wenn nicht der wichtigste Bestandteil der Internet-Protokollfamilie. Daher ist hier eine Änderung besonders schwer zu verwirklichen. Auf lange Sicht ist wünschenswert, daß alle an das Internet angeschlossenen Systeme auf IPv6 umgestellt werden. Dies kann natürlich nicht von heute auf morgen erreicht werden. Ein allmächlicher, möglichst einfacher Übergang ist nötig. Um den Übergang so einfach wie möglich zu machen, ist es notwendig, daß eine beidseitige Kommunikation zwischen IPv6-Systemen und nur-IPv4-Systemen möglich ist.

Schon bei der Entwicklung von IPv6 wurde großen Wert auf eine leichte Migration von IPv4 auf IPv6 geachtet. Folgende Anforderungen wurden schon bei der IPv6-Entwicklung berücksichtigt [RFC 1752]:

- Ein allmählicher Übergang von IPv4 nach IPv6 muß möglich sein
- Wenige oder keine Abhängigkeiten beim „Upgraden" dürfen vorhanden sein
- Es darf keine Umstellungsschalter geben um IPv6 benutzen zu können.

Der Übergang von IPv4 auf IPv6 sollte abgeschlossen sein, bevor der Mangel von IPv4-Adressen zu einem ernsthaften Problem wird.

Der Übergang von IPv4 nach IPv6 findet, wie in Abbildung 11 zu sehen, in zwei Phasen statt. Am Ende der ersten Phase beherrschen alle bestehenden Systeme sowohl IPv4 als auch IPv6. Erst wenn alle Systeme IPv6 beherrschen kann die Unterstützung von IPv4 aufgegeben werden.

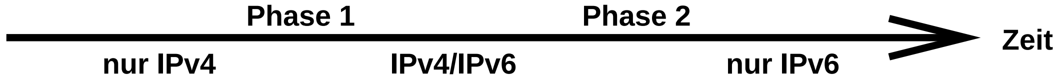

Abbildung 11: Migrationsphasen von IPv4 nach IPv6

4.2 Migrationsmechanismen

Durch die in [RFC 1933] beschriebenen Mechanismen wird ein allmähliche Übergang ermöglicht:

- Paralleler Einsatz von IPv4 und IPv6 („Dual IP Layer").
  Alle Systeme unterstützen während der Übergangszeit IPv4 und IPv6 (siehe Bild 12).

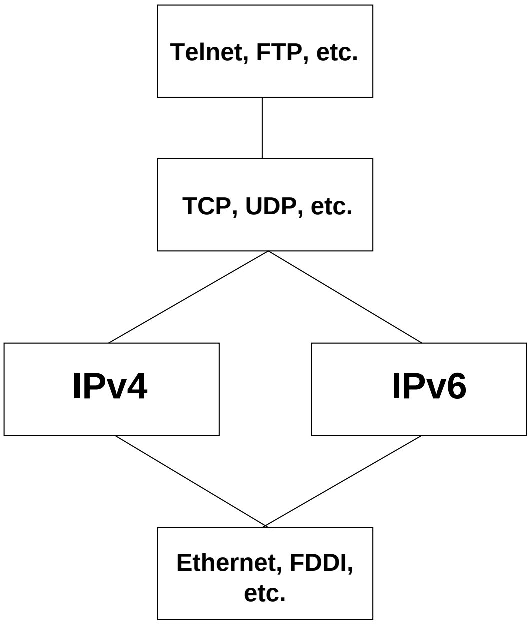

Abbildung 12: IPv4 und IPv6 werden in der Migrationsphase gleichzeitig unterstützt

- Tunneln
  IPv6-Datagramme können bei Bedarf in IPv4-Datagramme verpackt und über bestehende IPv4-Topologien getunnelt werden.

Die Upgrade-Abhängigkeiten sehen wie folgt aus:

- Bevor IPv4-Endsysteme oder Router auf IPv6 umgestellt werden können, ist es notwendig, Domain-Name-Server (DNS, setzt symbolische Namen in IP-Adressen um) so umzurüsten, daß sie zusätzlich zu den IPv4-„A"(dreß)-Einträgen „AAAA"-Einträge (die entsprechenden Einträge für IPv6-Adressen) unterstützen.

- Bevor auf nur-IPv6 umgestellt werden kann, ist es notwendig, daß alle Router (in dem betreffenden Bereich, mit dem man kommunizieren möchte) IPv6 unterstützen.

Nur-IPv6-Systeme können nicht mit nur-IPv4-Systemen kommunizieren.

4.3 Verfahrensweisen

Wesentlich für die vorgesehenen Verfahrensweisen ist, daß während der Migrationsphase Systeme IPv4- als auch IPv6-fähig sind, IPv4-kompatible IPv6-Adressen verwendet werden und vorhandene IPv4-Infrastruktur ohne Änderungen genutzt werden kann.

4.3.1 Domain Name Server

Domain Name Server spielen eine zentrale Rolle in der Migration von IPv4 aud IPv6. Abhängig von den dortigen Adreßeinträgen können Systeme entscheiden, welche Version des Internet Protocols verwendet werden soll.

Wichtig dabei ist, daß von nur-IPv4-Systemen im DNS nur ein A-Eintrag vorhanden ist, von IPv4/IPv6-Systemen sowohl ein A-Eintrag, als auch ein AAAA-Eintrag. Nur-IPv6-Systeme erhalten lediglich einen AAAA-Eintrag. Mit Hilfe des DNS ist es nun möglich zu entscheiden, auf welche Art und Weise mit einem anderen System kommuniziert werden soll: bekommt ein IPv4/IPv6-Server eine IPv6-kompatible Adresse als Ergebnis einer Anfrage an einen DNS zurück, so wird IPv6 verwendet. Andernfalls läuft die Kommunikation über IPv4.

Die Szenarien im einzelnen:

- IPv4-System A(lt) möchte mit IPv4/IPv6-System N(eu) kommunizieren:
  A schickt eine Anfrage an den DNS-Server und erhält (nur) eine IPv4-Adresse (der DNS weiß, daß das fragende System ein nur-IPv4-System ist). Damit kommunizieren A und N mittels IPv4.

- IPv4/IPv6-System N möchte mit IPv4-System A kommunizieren:
  N schickt eine Anfrage an den DNS-Server und erhält eine IPv4-Adresse. Damit kommunizieren N und A mittels IPv4.

- IPv4/IPv6-System $N1$ und IPv4/IPv6-System $N2$ kommunizieren:
  N1 schickt eine Anfrage an den DNS-Server und erhält sowohl eine IPv4, als auch eine IPv6-Adresse. Damit weiß $N1$, daß $N2$ auch über IPv6 zu erreichen ist und kommuniziert mit $N2$ mittels IPv6.

Damit wird klar, daß der Einsatz von AAAA-fähigen DNS essentiell für den Einsatz von IPv6 ist. Dabei muß der DNS selbst nicht unbedingt das IPv6-Protokoll zur Kommunikation benutzen; IPv4 ist für die Kommunikation mit den Anfragenden ebenfalls möglich. Die Umrüstung des DNS ist deswegen relativ einfach, lediglich ein neuer Adreßtyp muß unterstützt werden.

4.3.2 Tunneln

Durch den Einsatz entsprechender Adressen und der DNS wissen die beteiligten Kommunikationspartner zwar welches Protokoll sie verwenden sollen, aber dies reicht zur Kommunikation noch nicht aus. Selbst wenn beide beteiligten Partner IPv6 unterstützen, ist noch nicht sichergestellt, daß dies auch für alle Router zwischen den Kommunikationspartnern der Fall ist.

Um die bestehende IPv4-Infrastuktur auch weiterhin nutzen zu können, werden bei Bedarf IPv6-Datagramme **getunnelt**. Dabei werden, wie in Abbildung 13 zu sehen, IPv6-Datagramme in IPv4-Datagramme „eingepackt". Beim Tunnel-Endpunkt wird dann das IPv6-Datagramm aus dem IPv4-Datagramm „ausgepackt" und das IPv6-Datagramm ganz normal als solches behandelt. Falls nötig, ist auch der umgekehrte Fall denkbar, daß IPv4-Datagramme in IPv6-Datagrammen getunnelt werden.

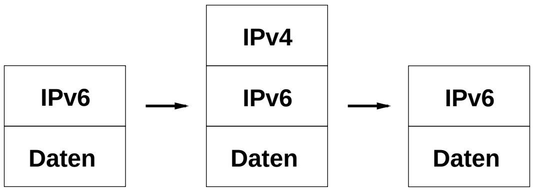

Abbildung 13: Tunneln von IPv6-Paketen über eine existierende IPv4-Infrastruktur

Falls ein IPv4/IPv6-Host direkt an einen IPv6-Router angeschlossen ist, werden IPv6-Datagramme direkt als solche versendet. Ist dies nicht der Fall, so werden IPv6-Datagramme entweder zu einem IPv6-fähigen Router oder gleich bis zum Endgerät getunnelt.

Es sind die folgende Verfahren denkbar, jeweils abhänig davon, wo der Anfang eines Tunnels und dessen Ende ist (hier: Host=Endsystem):

- Router-zu-Router-Tunnel. IPv4/IPv6-Router, welche durch eine IPv4-Infrastruktur verbunden sind, können IPv6-Pakete zwischen sich tunneln.
- Host-zu-Router. IPv4/IPv6-Hosts können IPv6-Pakete zu einem IPv4/IPv6-Router, der über IPv4 erreichbar ist, tunneln.
- Host-zu-Host. IPv4/IPv6-Hosts, welche über eine IPv4-Infrastruktur verbunden sind, können IPv6-Pakete zwischen sich selbst tunneln.
- Router-zu-Host. IPv4/IPv6-Router können IPv6-Pakete zu ihrem IPv4/IPv6-Ziel tunneln.

Bis ein Datagramm sein Ziel erreicht, ist der mehrfache Einsatz von Tunneln denkbar.

Beim Tunneln unterscheidet man zwischen „automatischem Tunneln" und „konfiguriertem Tunneln". Falls das Ziel eines IPv6-Paketes eine IPv4-kompatible Adresse hat, kann diese automatisch in eine IPv4-Adresse umgewandelt werden. Diese Adresse kann dann zum automatischen Tunneln herangezogen werden.

Der Tunnel-Mechanismus erscheint einfach, bei der Implementation muß jedoch auf einige Gegebenheiten geachtet werden, z.B.:

- Wie werden ICMP-Fehlermeldungen an den richtigen Empfänger geleitet?

- Welche Auswirkungen hat die MTU, wie kann diese effizient ausgenützt werden?
- Wie wird TTL bzw. das Hop Limit beim Tunneln geändert? Welche Auswirkungen hat diese?

Die Behandlung dieser Themen sprengt den Rahmen dieses Dokuments. Genauere Informationen sind zu diesem Thema in [RFC 1933] zu finden.

5 Ausblick

5.1 Weitere Änderungen

Mit der Änderung des Basisprotokolls IP alleine ist es nicht getan. Neben den Änderungen im Basisprotokoll kommt es, wie zum Teil schon angesprochen, zu Änderungen in folgenden Bereichen:

- Anpassung bestehender Protokolle
  - an 128 Bit-Adressen (TCP, UDP etc.)
  - an neue Möglichkeiten (Authentifikation etc.)
- Neuentwicklung von Protokollen
  - neue Möglichkeiten werden durch neue Protokolle genutzt (RSVP etc.)
- Anpassung größerer Teile von Betriebssystemen
  - viele interne Änderungen nötig
  - verändertes API (z.B. Socket-Interface)
  - überarbeitete und neue Netzwerkbefehle
  - Unterstützung von Transitionsmechanismen (Dual Layer, Tunneln)
  - Unterstützung von neuen Möglichkeiten (Autokonfiguration, Sicherheit etc.)
- Applikationsbereich
  - leichte Änderungen und Neukompilierung für Unterstützung von längeren Adressen nötig
  - völlige Überarbeitung zur Unterstützung von neuen Möglichkeiten

5.2 Probleme von IPv6 und Alternativen

Die Entwicklung von IPv6 ist noch nicht abgeschlossen. Auf lange Sicht gibt es wohl zu IPv6 keine Alternative. Ein schneller Umstieg von IPv4 auf IPv6 ist nicht zu erwarten, so lange nicht die neuen Möglichkeiten von IPv6 interessant sind.

Die Unterstützung von ATM durch das Resource Reservation Protocol ist nicht optimal; hier zeigen sich die prinzipiellen Probleme, einen leitungsvermittelter Dienst als Basis für einen verbindungslosen Dienst zu benutzen. Hier bietet das *Internet STream Protocol Version 2(+)* (ST2(+)) [RFC 1190]/[RFC 1819] deutliche Vorteile. Daher kann es sein, daß für Video und Audiodaten eher dieses Protokoll verwendet werden wird.

Bei einigen Anwendungen kann, wie im Vortrag „IP über ATM“ [Gar96] gezeigt, eventuell auch ganz auf IP verzichtet werden.

Die nicht mehr vorhandene Prüfsumme im IPv6-Header führt dazu, daß Übertragungsfehler im Kopf erst später entdeckt werden. Mit einer zudehmenden Fehlerfreiheit der im Moment hauptsächich verwendeten physikalischen Medien (Ethernet etc.) stellt dies kein größeres Problem dar. Da aber IPv6 auch oder gerade über andere Medien übertragen werden soll (Funk, Infrarot), kann dies zu einem Problem werden. Es wird erwartet, daß die Fehlererkennung durch ein höheres Protokoll wie UDP und TCP vorgenommen wird.

5.3 Aktueller Entwicklungsstand von IPv6

Das Kernprotokoll von IPv6 wird wohl nicht mehr geändert werden. Bei den wichtigen Protokollen aus der Transportschicht, wie TCP und UDP, ändert sich nur sehr wenig. Die Entwicklung der Protokolle um IPv6 herum, wie z.B. der dynamischen Konfigation ist noch in keiner Weise abgeschlossen. Es ist der Trend festzustellen, daß viele der Protokolle, die neue Funktionalität ermöglichen (Security, RSVP etc.), sowohl für IPv4 als auch für IPv6 entwickelt werden.

Mehrere Organisationen arbeiten momentan an IPv6-Implementationen. Es wird auch hier wieder sogenannte „freie“ Implementierungen von IPv6 geben, z.B. durch das US Naval Research Laboratory. Der Quelltext wird unter Berkeley-ähnlichen Lizenzbestimmungen „free distributable“ sein.

Verschiedene Firmen arbeiten ebenso an der Implementierung von IPv6, Tabelle 5 gibt eine Übersicht.

IPv6-Router-Impementierungen werden von folgenden Firmen vorgenommen:

- Bay Networks
- Cisco Systems
- Penril Datability Networks
- Telebit Communications A/S

Firma/Organisation	**Betriebssystem**
Digital Equipment Corporation	Digital UNIX
FTP Software	DOS/Windows
Swedish Institute of Computer Science	HP-UX
Apple Computer	MacOS
INRIA Rocquencourt	NetBSD
BULL SA	AIX
Siemens Nixdorf Informationssysteme AG	BS2000
Sun Microsystems Inc.	Solaris 2
Mentat	STREAMS

Tabelle 5: IPv6-Implementierungen

- Ipsilon Networks, Inc.

Eine genauere Übersicht über die schon vorhandenen Implementierungen bzw. geplante Implementierungen ist unter der URL `http://playground.sun.com/pub/ipnghtml/ipng-implementations.html` zu finden.

5.4 Neue Möglichkeiten

Das Internet Protocol Version 6 ermöglicht interessante neue Anwendungen. Um die Abrechnung zu erleichtern, wäre es durchaus möglich, eine IP-Adresse für einzelne Personen zu verwenden, anstatt für einzelne Interfaces.

Eine andere interessante Idee ist es, bereits in die Adresse verschiedene Informationen fest hineinzukodieren. Anwendung wäre z.B. das jeder Dienst, der ein Sicherheitsrisiko darstellen könnte (WWW, FTP etc.) eine eigene Internet-Adresse bekäme. Ebenfalls möglich wäre die Verwendung von einzelnen Adreßbits für die Information über den Preis eines Services, und wie dieser abgerechnet wird (vgl. bestimmte Telefonnummern wie 0130 oder 0190).

Die auf Seite 10 bereits vorgestellten Ideen stellen weitere interessante Anwendungsmöglichkeiten dar. Fernsehgeräte mit eigener IP-Adresse zur Benutzung des Internets sind keine Zukunftsvision mehr, es gibt die ersten entsprechenden Geräte zu einem nur unwesentlich höhererm Preis als vergleichbare Geräte ohne Internet-Anschluß zu kaufen.

6 Literatur

[Hin96] Robert Hinden, *IP Next Generation (IPng) Homepage*, `http://playground.sun.com/pub/ipng/html/ipng-main.html`, 22 Januar 1996

Bücher und Zeitschriften stellen im schnellebigen Internet keine gute Grundlage für Informationen zu IPv6. Die wesentliche Grundlage für Informationen

im Internet-Bereich stellen RFCs sowie Internet Drafts dar. Die weiter unten angegebenen RFCs und Internet Drafts waren bei der Erstellung dieses Dokuments aktuell; der Leser möge bitte selber auf neuere Versionen sowie neue Dokumente achten.

Bücher und Zeitschriften:

[Bra96a] Scott O. Bradner, Allison Mankin (Ed.), *IPng, Internet Protocol Next Generation*, Addison-Wesley, 1996, ISBN 0-201-63395-7

[Clark88] David D. Clark, *The Design Philosophy of the DARPA Internet Protocols*, Computer Communications Review, Volume 18, Nummer 4, August 1988, Seiten 106-114, ACM Press

[Com95] Douglas E. Comer, *Internetworking with TCP/IP, Vol. I: Principles, Protocols, and Architecture* (Third Edition), Prentice-Hall, 1995, ISBN: 0-13-216987-8

[Gar96] Markus Garschhammer, Armin Gruner, Votrag *IP über ATM*, Hauptseminar *Moderne Kommunikationskonzepte, Protokolle und Dienste*, Lehrstuhl Prof. Hegering, Technische Universität/Ludwig Maximilians-Universität München, 1996

[Hin95] Robert M. Hinden, *IP Next Generation Overview*, ConneXions, Volume 9, No. 3, 1995, pages 2-18, ISSN: 0894-5926

[Tan96] Andrew S. Tanenbaum, *Computer Networks*, 3. Auflage, Prentice-Hall, März 1996, ISBN: 0-13-394248-1

Request For Comments:

RFCs sind auf verschiedenen Servern zu finden, unter anderem auf `ftp.leo.org` unter `pub/comp/doc/standards/rfc`. Ein guter Index über alle RFCs erhält man unter `http://ds.internic.net/ds/rfc-index.html`

[RFC 768] J. Postel, *User Datagram Protocol*, RFC 768, USC/Information Sciences Institute, August 1980

[RFC 791] J. Postel (Ed.), *Internet Protocol - DARPA Internet Program Protocol Specification*, RFC 791, USC/Information Sciences Institute, September 1981

[RFC 793] J. Postel (Ed.), *Transmission Control Protocol - DARPA Internet Program Protocol Specification*, RFC 793, USC/Information Sciences Institute, September 1981

[RFC 1190] C. Topolcic, *Experimental Internet Stream Protocol: Version 2 (ST-II)*, RFC 1190, Oktober 1990

[RFC 1517] R. Hinden (Ed.), *Applicability Statement for the Implementation of Classless Inter-Domain Routing (CIDR)*, RFC 1517, Internet Engineering Steering Group, September 1993

[RFC 1518] Y. Rekhter (Ed.), T. Li (Ed.), *An Architecture for IP Address Allocation with CIDR*, RFC 1518, Network Working Group, September 1993

[RFC 1519] V. Fuller, T. Li, J. Yu, K. Varadhan, *Classless Inter-Domain Routing (CIDR):an Address Assignment and Aggregation Strategy*, RFC 1519, Network Working Group, September 1993

[RFC 1520] Y. Rekhter, C. Topolcic, *Exchanging Routing Information Across Provider Boundaries in the CIDR Environment*, RFC 1520, Network Working Group, September 1993

[RFC 1550] S. Bradner, A. Mankin, *IP: Next Generation (IPng) White Paper Solicitation*, RFC 1550, Network Working Group, Dezember 1993

[RFC 1752] S. Bradner, A. Mankin, *The Recommendation for the IP Next Generation Protocol*, RFC 1752, Network Working Group, Januar 1995

[RFC 1819] L. DelGrossi, L. Berger, *Internet STream Protocol Version 2+*, RFC 1819, Juli 1995

[RFC 1825] R. Atkinson, *Security Architecture for the Internet Protocol*, RFC 1825, Network Working Group, August 1995

[RFC 1826] R. Atkinson, *IP Authentication Header*, RFC 1826, Network Working Group, August 1995

[RFC 1827] R. Atkinson, *IP Encapsulating Security Payload (ESP)*, RFC 1827, Network Working Group, August 1995

[RFC 1828] P. Metzger, W. Simpson, *IP Authentication using Keyed MD5*, RFC 1828, Network Working Group, August 1995

[RFC 1883] S. Deering, R. Hinden, *Internet Protocol, Version 6 (IPv6) Specification*, RFC 1883, Network Working Group, Dezember 1995

[RFC 1884] R. Hinden (Ed.), S. Deering (Ed.), *IP Version 6 Addressing Architecture*, RFC 1884, Network Working Group, Dezember 1995

[RFC 1885] A. Conta, S. Deering, *Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification*, RFC 1885, Network Working Group, Dezember 1995

[RFC 1886] S. Thomson, C. Huitema, *DNS Extensions to support IP version 6*, RFC 1886, Network Working Group, Dezember 1995

[RFC 1887] Y. Rekhter (Ed.), T. Li (Ed.), *An Architecture for IPv6 Unicast Address Allocation* , RFC 1887, Network Working Group, Dezember 1995

[RFC 1920] J. Postel (Ed.), *Internet Official Protocol Standards*, RFC 1920, Internet Architecture Board, März 1996

[RFC 1933] R.Gilligan, E. Nordmark, *Transition Mechanisms for IPv6 Hosts and Routers*, RFC 1933, Network Working Group, April 1996

[RFC 1949] A. Ballardie, *Scalable Multicast Key Distribution*, RFC 1949, Network Working Group, Mai 1996

[RFC 1958] B. Carpenter (Ed.), *Architectural Principles of the Internet*, RFC 1958, Network Working Group, Juni 1996

Drafts:

Internet Drafts sind Draft-Dokumente, die maximal für 6 Monate gültig und verfügbar sind. Internet Drafts können jederzeit verbessert, ersetzt, oder duch andere Dokumente obsolet werden. Es ist nicht korrekt, Internet Drafts als Referenzmaterial anzugeben oder aus ihnen zu zitieren außer als „working draft“ oder „work in progress“.

Der jeweils aktuelle Status eines Internet Drafts erfährt man unter `http://www.ietf.cnri.reston.va.us/1id-abstracts.html`.

Einen Überblick über vorhandene Internet Drafts in München erhält man unter der URL `http://www.leo.org/archiv/doc/standards/internet-drafts/`.

In München sind Internet Drafts über ftp auf `ftp.leo.org` unter `pub/comp/doc/standards/internet-drafts` zu erhalten.

[Mal96] G. Malkin, *Internet Users' Glossary*, draft-ietf-userglos-glossary2-01.txt, Mai 1996
Eine gute und umfangreiche Erklärung aller Internet-spezifischen Abkürzungen und Begriffe.

[Thom96] Susan Thomson, Thomas Narten, *IPv6 Stateless Address Autoconfiguration*, ADDRCONF Working Group, draft-ietf-addrconf-ipv6-auto-07.txt, Dezember 1995

[Bou96] J. Bound, C. Perkins, *Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*, DHC Working Group, draft-ietf-dhc-dhcpv6-04.txt, Februar 1996

[Elz96] Robert Elz, *Identifying IPv6 Interfaces in Link-Local Addresses*, Network Working Group, draft-ietf-ipngwg-iid-02.txt, Mai 1996

[Fen96] W. Fenner, *Internet Group Management Protocol, Version 2*, Inter-Domain Multicast Routing Working Group, draft-ietf-idmr-igmp-v2-02.txt, Februar 1996

[Per96] Charles Perkins, David B. Johnson, *Mobility Support in IPv6*, IPv6 Working Group, draft-ietf-mobileip-ipv6-00.txt, Januar 1996

[Cra96] Matt Crawford, *A Method for the Transmission of IPv6 Packets over Ethernet Networks*, IETF, draft-ietf-ipngwg-ethernet-ntwrks-02.txt, Februar 1996

[Nar96] Thomas Narten, Erik Nordmark, W. A. Simpson, *Neighbor Discovery for IP Version 6 (IPv6)*, IETF, draft-ietf-ipngwg-discovery-06.txt, März 1996

[McC96] J. McCann, S. Deering, J. Mogul, *Path MTU Discovery for IP version 6*, IETF, draft-ietf-ipngwg-pmtuv6-02.txt, April 1996

[Bra96b] R. Braden (Ed.) et al., *Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification*, IETF, draft-ietf-rsvp-spec-12.txt, Mai 1996