Die Internet-Protokollfamilie der nächsten Generation

Dr. Torsten Braun

IBM Europäisches Zentrum für Netzwerkforschung
Vangerowstr. 18, D-69115 Heidelberg

email: braun@heidelbg.ibm.com

Telefon: 06221/59-4352
Fax: 06221/59-3300

Kurzzusammenfassung

IP ist das zentrale Protokoll der Internet-Familie und bildet die Basis sämtlicher Internet-Anwendungen wie electronic Mail, FTP-Dateitransfer oder World Wide Web. Wegen der Engpässe im zur Verfügung stehenden Internet-Adreßraum wurde 1992 von der IETF (Internet Engineering Task Force) entschieden, eine neue Version des Internet-Protokolls (IP) zu entwickeln. Die Notwendigkeit einer neuen IP-Version wurde zu einem kompletten Neuentwurf des Protokolls genutzt. Der Artikel gibt einen Überblick über die wichtigsten Eigenschaften des Protokolls wie neue Adreß- und Datenformate, Unterstützung von Dienstgüten sowie Sicherheitsaspekten. Daraus ergeben sich nicht zuletzt Möglichkeiten wie die automatische Systemkonfiguration sowie die Unterstützung mobiler und multimedialer Endsysteme.

1 Einführung

Aufgrund der steigenden Popularität des Internet und seiner angebotenen Dienste steigt seit Jahren die Anzahl der Internet-Knoten stark an. Es ist abzusehen, daß zwischen den Jahren 2000 und 2010 der derzeit verfügbare IPv4-Adreßraum den Bedarf nicht mehr abdecken kann. Aus diesem Grund wurde Anfang des Jahres 1992 entschieden, eine neue Version des IP-Protokolls zu entwickeln. Das neue Internet Protocol Version 6 (IPv6) basiert auf dem Protokoll SIPP (Simple Internet Protocol Plus) [26], welches sich wiederum aus den Ansätzen ENCAPS, SIP (Simple Internet Protocol) und PIP (The P Internet Protocol) entwickelte [10][12].

Das folgende Kapitel beschreibt die wichtigsten Änderungen von IPv6 [11][21] gegenüber der Vorgängerversion IPv4 [14]. Im Vordergrund stehen die neuen Daten- und Adressierungsformate, die sich daraus ergebenden Möglichkeiten zur automatischen Systemkonfiguration sowie verschiedene Neuerungen zur Unterstützung von Dienstgüten und Sicherheit. Anschließend werden die Auswirkungen der IPv6-Entwicklung auf damit in Beziehung stehende Protokolle behandelt und Strategien zur Migration von IPv4 nach IPv6 diskutiert.

2 IP Version 6

Die wichtigsten neuen Eigenschaften von IPv6 betreffen die folgenden Aspekte, welche im weiteren detaillierter diskutiert werden:

- Erweiterte Adressierungsmöglichkeiten
- Vereinfachungen des grundlegenden Datenformats und verbesserte Unterstützung von Erweiterungen und Optionen
- Dienstgütenunterstützung

- Automatische Systemkonfiguration
- Sicherheitsaspekte
- Unterstützung mobiler Endsysteme

2.1 Adressierung

Die Adressen wurden in IPv6 auf 128 Bits erweitert, was pro Quadratmeter der Erdoberfläche ca. $6.65 \cdot 10^{23}$ Adressen erlaubt. Grundsätzlich werden

- Unicast-,
- Anycast- und
- Multicast-Adressen

unterschieden. Der Typ einer IPv6-Adresse wird durch ein Format-Präfix festgelegt, welches aus den führenden Bits einer IPv6-Adresse besteht [27]. Anycast-Adressen werden aus dem Unicast-Bereich gebildet, so daß Unicast- und Anycast-Adressen syntaktisch nicht unterschieden werden können. Die höchsten Bits des Format-Präfix unterscheiden lediglich, ob es sich bei einer Adresse um eine Unicast/Anycast- oder um eine Multicast-Adresse handelt.

2.1.1 Unicast-Adressen

Eine Unicast-Adresse identifiziert ein einziges Interface eines Zwischen- bzw. Endsystems. Ein an eine Unicast-Adresse gesendetes Paket wird daher von genau einem, d.h. dem explizit adressierten, Interface empfangen. Unicast-Adressen lassen sich weiter mit Hilfe des Format-Präfix in mehrere Typen unterscheiden. Die Struktur einer Unicast-Adresse und damit die Anzahl der Hierarchiestufen hängt vom jeweiligen Präfix ab. Tabelle 1 zeigt die derzeit definierten Präfix-Werte und die dazugehörigen Adreßtypen.

Tabelle 1: Präfix-Werte und Adreßtypen

Adreßtyp	Präfix	Anteil am Adreßraum
reserviert (z.B. für Loopback-, IPv4-kompatible Adressen)	0000 0000	1/256
NSAP-Adressen	0000 001	1/128
IPX-Adressen	0000 010	1/128
Provider-basierte Unicast-Adressen	010	1/8
Geographisch-basierte Unicast-Adressen	100	1/8
Link-lokale Unicast-Adressen	1111 1110 10	1/1024
Standort-lokale Unicast-Adressen	1111 1110 11	1/1024

Neben speziellen Adreßformaten zur Integration von IPX- und OSI-NSAP-Adressen [8] existieren Adressen für reine IPv4-Knoten sowie für IPv6-Knoten, die ein Einkapseln von IPv6-Paketen in IPv4-Paketen erlauben (IPv4-kompatible IPv6-Adressen). Die Einkapselungstechnik wird für den Übergang von IPv4 nach IPv6 benötigt (vgl. Kap. 4).

Ein Beispiel für eine hierarchisch strukturierte IPv6-Adresse ist die Provider-basierte Unicast-Adresse (s. Abbildung 1), die sich neben einem festen Präfix aus einer jeweils variablen Anzahl von Bits für Registrierungsinstanz-, Provider-, Subscriber-, Subnetz- und Interface-Kennung zusammensetzen kann [37].

Link-lokale Adressen enthalten neben dem Präfix lediglich eine Interface-Kennung in den niederwertigsten Bits. Link-lokale Adressen werden während der automatischen Konfiguration oder, wenn in einem Netz keine Router vorhanden sind, verwendet. Router dürfen Pakete mit Link-lokalen Adressen als Quelladressen nicht weiterleiten.

3 bit	n bit	m bit	o bit	p bit	125-m-n-o-p bit
010	Registry-ID	Provider-ID	Subscriber-ID	Subnetz-ID	Interface-ID

Abbildung 1: Provider-basierte globale Unicast-Adresse

Standort-lokale Adressen werden benutzt, falls ein Standort bzw. eine Organisation noch nicht an das Internet angeschlossen ist. In diesem Fall müssen solche Organisationen nicht eine von einem anderen Internet-Knoten belegte Adresse verwenden, sondern erhalten eine speziell für diesen Zweck vorgesehene Adresse. Bei einem späteren Internet-Anschluß muß dann lediglich das Standort-lokale Präfix durch ein Provider-basiertes Präfix ersetzt werden. Subnetz- und Interface-Kennung bleiben unverändert.

Bei Benutzung einer IEEE-802-MAC-Adresse als Interface-Kennung ist diese typischerweise 48 bit lang. Falls kürzere Interface-Kennungen gewünscht werden, können diese auch lokal, z.B. durch einfaches Numerieren, konfiguriert werden. Bei der automatischen Adreßkonfiguration (s. Abs. 2.4) setzen die an einen Link angeschlossenen Knoten die vom entsprechenden Router verteilten Adreßinformationen und die Interface-Kennung zu einer IPv6-Adresse zusammen.

2.1.2 Anycast-Adressen

Eine Anycast-Adresse kennzeichnet eine Menge von Interfaces, die typischerweise zu verschiedenen Zwischensystemen gehören. Ein an eine Anycast-Adresse gesendetes Paket wird dabei genau an ein Interface dieser Menge, in der Regel dem nächst gelegenen Interface gemäß der Routing-Metrik, ausgeliefert.

Anycast-Adressen werden aus dem Unicast-Adreßbereich allokiert und können daher syntaktisch nicht von einer Unicast-Adresse unterschieden werden. Knoten, die zu einer Anycast-Adresse gehören, müssen explizit so konfiguriert werden, daß sie die Anycast-Adresse als solche erkennen. Üblicherweise befinden sich Knoten einer Anycast-Adresse in einer topologischen Region, die durch ein Präfix identifiziert wird. Allerdings kann es auch vorkommen, daß sich Knoten einer Anycast-Adresse in völlig unterschiedlichen Regionen befinden. In diesem Fall müssen Anycast-Adressen als getrennte Routing-Einträge im Internet ausgewiesen werden.

Eine beispielhafte Anwendung von Anycast-Adressen ist der Wunsch, Pakete über ein Subnetz eines bestimmten Netzanbieters zu übertragen. Dies erfolgt dann damit, daß im Routing-Kopf eine Anycast-Adresse, die einen oder mehrere Knoten eines bestimmten Netzanbieters beschreibt, angegeben werden. Ein durch die Anycast-Adresse beschriebener Knoten muß dann beim Übertragen der Dateneinheit zum endgültigen Ziel besucht werden. Weitere Anwendungen von Anycast-Adressen werden noch untersucht. Anycast-Adressen sollen zunächst nicht in Quelladressen auftreten und auch nur Zwischensysteme identifizieren.

2.1.3 Multicast-Adressen

Eine Multicast-Adresse besitzt ein Flag-Feld, ein Scope-Feld und die Gruppenkennung (s. Abbildung 2). Das Flag-Feld zeigt an, ob es sich bei der Gruppe um eine transiente oder permanente Gruppe handelt. Permanente Gruppen sind bekannte Gruppen mit einer durch die Internet Assigned Numbering Authority (IANA) registrierten Adresse. Transiente Adressen sind nicht permanent eingerichtete Adressen. Das Scope-Feld kodiert die Reichweite, d.h. den Gültigkeitsbereich der Multicast-Gruppe. Dieser kann auf den Knoten, den Link, den Standort oder die Organisation beschränkt sein oder eine globale Gültigkeit besitzen. Permanente Adressen sind im Gegensatz von transienten Adressen unabhängig vom Gültigkeitsbereich. Beispielsweise haben zwei in unterschiedlichen Organisationen eingerichtete, transiente Gruppen mit Gültigkeitsbereich Organisation keine Beziehung zueinander.

8bit	4bit	4 bit	112 bit
11111111	flags	scope	Gruppen-ID

Abbildung 2: Multicast-Adressen

Folgende Multicast-Adressen sind vordefiniert:

- All-Nodes-Adresse für die Gruppe aller IPv6-Knoten (Knoten- und Link-lokal)
- All Routers-Adresse für die Gruppe aller IPv6-Router (Knoten- und Link-lokal)
- DHCP(Dynamic Host Configuration Protocol)-Server/Relay-Agent-Adresse für die Gruppe aller DHCP-Server und Relay-Agenten (Link-lokal)
- Solicited-Nodes-Adresse (Link-lokal)

Die Link-lokale Solicited-Nodes-Adresse besteht aus einem 96 bit langen Präfix und den unteren 32 Bits der Unicast- bzw. Anycast-Adresse. Jeder Knoten ist automatisch Mitglied in der mit der Solicited-Nodes-Multicast-Adresse verbundenen Gruppe. Die vordefinierten Adressen werden insbesondere von mit IPv6 verbundenen Kontrollprotokollen wie Neighbor Discovery (ND) und DHCPv6 verwendet (s. Kap. 2.4). Multicast-Kommunikation bildet damit eine wichtige Grundlage für mehrere grundlegende Kontrollmechanismen im IPv6-Umfeld.

IPv6-Multicast-Adressen werden auf Ethernet- und FDDI-Adressen abgebildet, die sich aus dem Präfix 0x3333 und den vier hinteren Bytes der IPv6-Adresse zusammensetzen [18][19]. Bei Token-Ring-Netzen werden alle Multicast-Adressen auf eine feste Adresse abgebildet [44].

2.2 Datenformate

2.2.1 Allgemeines Paketformat

Im Vergleich zu IPv4 wurden bei IPv6 einige gravierende Änderungen beim Paketformat vorgenommen. Insgesamt fällt auf, daß das minimale IPv6-Paketformat weniger Felder enthält als das minimale IPv4-Paketformat.

Verschiedene Felder wie Header Length, Header Checksum, Type Of Service und Identification wurden eliminiert (in Abbildung 3 durchgestrichene Felder), andere IPv4-Felder wie Fragmentierungs-Information (Flags, Fragment Offset) und Optionen wanderten in Erweiterungs-Header (fett umrandete Felder). Dadurch sollen die Verarbeitungskosten für die IP-Dateneinheiten ohne optionale Informationen, die in der Regel den Normalfall darstellen, minimiert werden und die für den minimalen Header erforderliche Bandbreite gering gehalten werden.

Die IPv4-Felder Precedence, Total Length, Time To Live, Precedence und Protocol werden durch Priority, Payload Length, Hop Limit und Next Header ersetzt. Diese Felder sind in Abbildung 3 und Abbildung 4 schattiert. Hop Limit gibt die Anzahl der maximalen Anzahl von Hops, die ein Paket passieren darf, an. Hop Limit ersetzt das Feld Time To Live, welches allerdings in IPv4 üblicherweise als Hop-Zähler implementiert wurde, da ein Hop-Zähler wesentlich einfacher zu implementieren ist als eine Zeitangabe [41].

Das neue Format für Optionen soll auch flexiblere Options-Längen und die inkrementelle Einführung neuer Optionen besser unterstützen. Außerdem werden die Optionen bis auf eine Ausnahme nur in Endsystemen verarbeitet, was zu einer Vereinfachung der Paketverarbeitung in den Routern führt.

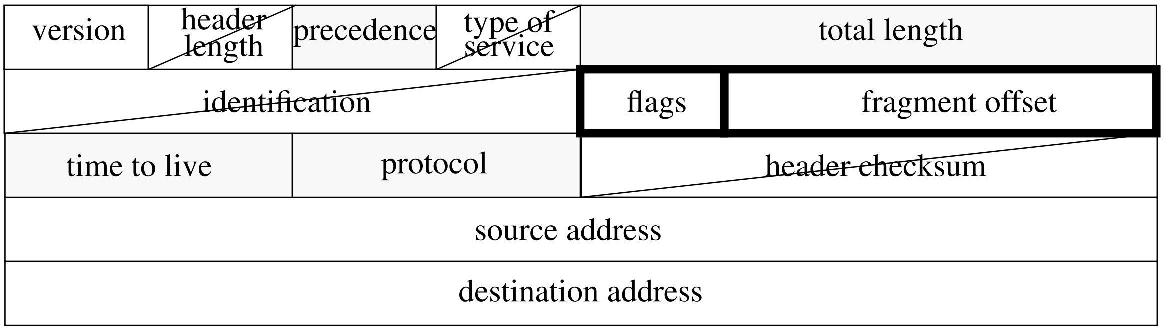

Abbildung 3: IPv4-Header

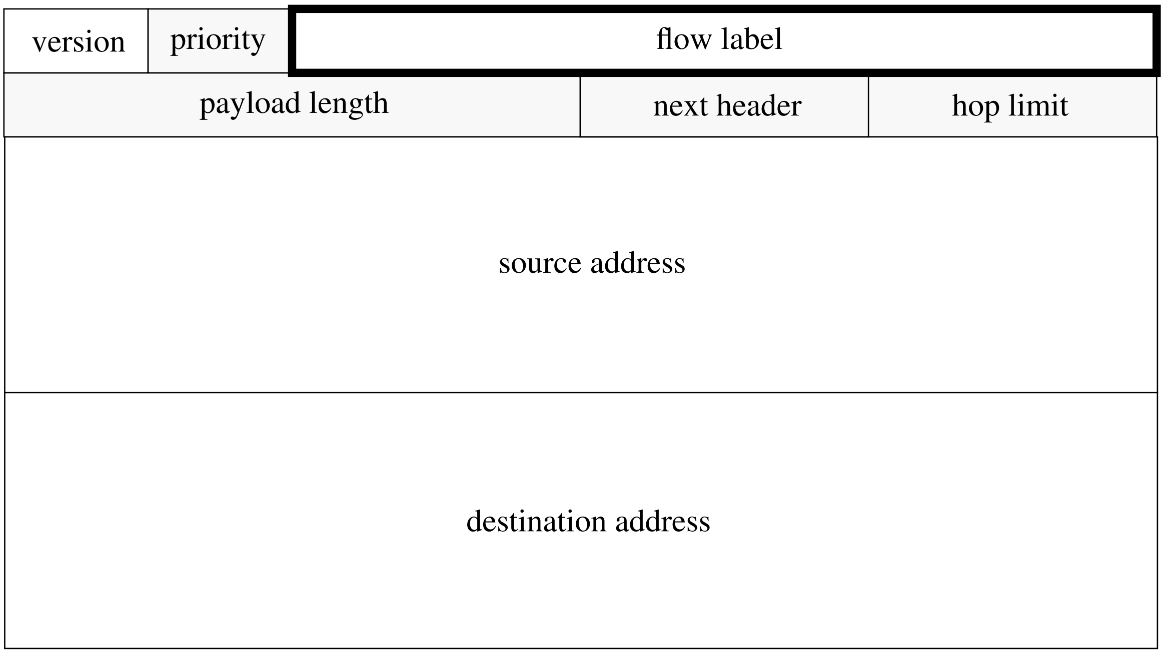

Abbildung 4: IPv6-Header

2.2.2 Header-Erweiterungen

In IPv6 werden optionale Informationen in separaten Header-Erweiterungen kodiert (Abbildung 5). Diese Header werden zwischen dem IPv6-Header und den Headern der darüberliegenden Schichten plaziert. Jeder dieser verschiedenen Header-Erweiterungen wird durch einen speziellen Wert gekennzeichnet. Header-Erweiterungen werden bis auf eine Ausnahme nicht in den Routern entlang eines Pfades bearbeitet, d.h. sie werden nur in den Endsystemen ausgewertet. Die Ausnahme besteht in den Hop-by-hop-Optionen, die in den Zwischensystemen ausgewertet werden müssen. Diese Header-Erweiterung muß unmittelbar dem IPv6-Kopf folgen. Auf die Hop-by-hop-Optionen folgende Erweiterungs-Header können damit von den Routern ignoriert werden. Die Existenz der Hop-by-hop-Option wird durch den Wert 0 im next-Header-Feld des IPv6-Headers angezeigt.

Jede Header-Erweiterung besitzt eine Länge mit einem Vielfachen von 8 Bytes, um eine Ausrichtung der folgenden Kopf-Erweiterungen auf 8-Byte-Grenzen zu gewährleisten. Jede Erweiterung darf höchstens einmal in einem Paket vorkommen. Falls mehr als eine Header-Erweiterung existiert, so müssen diese in einer bestimmten, im folgenden dargestellten Reihenfolge auftreten. Andere Reihenfolgen sind zulässig und müssen bearbeitet werden.

- IPv6-Header
- Hop-by-hop-Optionen
- Ziel-Optionen zur Verarbeitung im ersten durch die IPv6-Zieladresse gekennzeichneten Zielknoten und in den im Routing-Header angegebenen Knoten
- Routing-Header
- Fragment-Header
- Authentifizierungs-Header
- Verschlüsselung-Header
- Ziel-Optionen zur Verarbeitung im endgültigen Zielknoten

IPv6 Header next header = TCP	TCP Header + Daten

IPv6 Header next header = Routing	Routing Header next header = TCP	TCP Header + Daten

IPv6 Header next header = Routing	Routing Header next header = Fragment	Fragment Header next header = TCP	TCP Header + Daten (Fragment)

Abbildung 5: Header-Erweiterungen

Die *Hop-by-hop-Optionen* werden dazu verwendet, optionale Informationen auszutauschen, die in jedem Knoten eines Pfades ausgewertet werden müssen. Eine Hop-by-Hop-Option ist die Jumbo-Paket-Option, die aus einer 4 Oktetts langen Längenangabe besteht. Dadurch hat man die Möglichkeit, Längen bis zu 2^{32}-1 zu definieren und damit längere Pakete als 65535 Oktetts zu generieren. Die Nutzlastlänge im IPv6-Header muß bei verwendeter Jumbo-Paket-

Option auf 0 gesetzt sein. Die Jumbo-Paket-Option darf nicht in Paketen mit Fragment-Kopf verwendet werden.

Der *Routing-Header* wird von einer IPv6-Quelle dazu verwendet, einen oder mehrere Router bzw. Topologiebereiche bei der Angabe von Anycast-Adressen zu spezifizieren, die auf dem Weg eines Pakets zum endgültigen Zielknoten passiert werden sollen. Damit kann der Weg eines Pakets zu seinem Ziel beeinflußt werden um geringere Kosten oder eine bessere Dienstqualität zu erreichen. Die Funktion des Routing-Headers entspricht der Source-Routing-Funktion von IPv4. Der Routing-Header wird erst dann bearbeitet, wenn er den nächsten in der Adressenliste angegebenen Knoten erreicht hat. Falls dies der Fall ist, wird die nächste im Routing-Header enthaltene Adresse in das Zieladresse-Feld eingetragen.

Der *Fragment-Header* wird dazu benutzt, um größere Nutzlastpakete zu senden als in die Pfad-MTU (MTU: maximum transmission unit) paßt. Die *Pfad-MTU* ist das Minimum aller Link-MTUs auf dem Pfad zwischen Quell- und Zielknoten. Eine *Link-MTU* ist die maximale Größe einer Übertragungseinheit auf einem Link. Die Link-MTU hängt vom Netzwerktyp ab [18][19][44]. Im Gegensatz zu IPv4 wird bei IPv6 nur in Quellknoten segmentiert und nur im Zielknoten reassembliert. Router führen keine Segmentier- und Reassemblierfunktionen durch. IPv6 erfordert, daß jeder Link im Internet mindestens eine MTU von 576 Oktetts unterstützt. Falls dies nicht der Fall ist, müssen auf der Link-Ebene Segmentier- und Reassemblierfunktionen angeboten werden. Beispielsweise müssen ATM-Adaptionsschichten Segmentier- und Reassemblierfunktionen anbieten, um die MTUs in ATM-Zellen zu segmentieren und die Zellen schließlich wieder zu MTUs zu reassemblieren.

Es wird erwartet, daß IPv6-Knoten eine Pfad-MTU-Erkennung (path MTU discovery) [32] durchführen, und daß die Pakete von der Anwendung, d.h. dem darüber liegenden Protokoll, kleiner als die Pfad-MTU gewählt werden. Bei der Pfad-MTU-Erkennung sendet der Sender das erste Paket an eine gegebene Zieladresse mit der Größe der Link-MTU des ersten Links. Falls auf dem Weg zum Ziel das Paket die Link-MTU eines Netzes übersteigt, so daß das Paket nicht weitergeleitet werden kann, sendet der betreffende Router eine ICMP-Nachricht “message too big” mit Angabe der Link-MTU an den Sender zurück. Der Sender bildet die folgenden Pakete, so daß diese Link-MTU nicht überschritten wird.

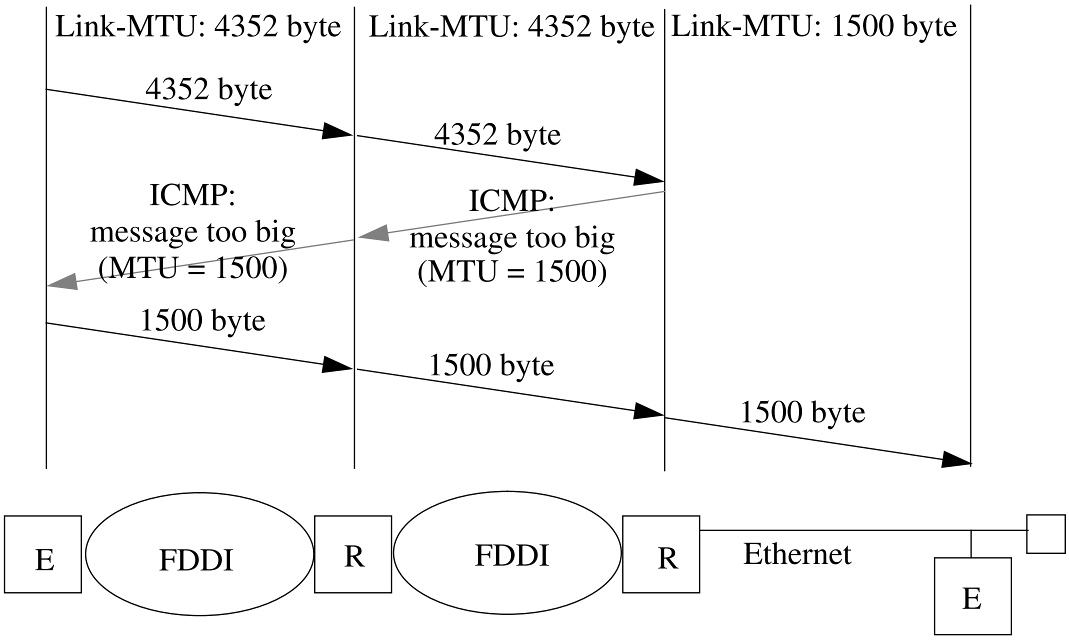

Abbildung 6: Pfad-MTU-Erkennung

Falls keine Pfad-MTU-Erkennung durchgeführt werden kann, werden lediglich Pakete mit einer minimalen MTU-Größe gesendet, um die Benutzung von Segmentier- und Reassemblierfunktionen sowie von Fragment-Köpfen zu vermeiden. Dieses Vorgehen entspricht auch dem Konzept des Application Level Framing [13], welches unter anderem vorschlägt, daß bereits auf Anwendungsebene eine für alle darunter liegenden Schichten passende Paketgröße gewählt wird und die Anwendung ihre Daten bereits in die passende Einheit unterteilen. Lediglich in Fällen, in denen die Länge des zu sendenden Pakets die Pfad-MTU übertrifft, wird in IP segmentiert und ein Fragment-Kopf benötigt. Der Fragment-Header enthält Informationen um die relative Position des Segments im ursprünglichen, nicht segmentierten Paket anzuzeigen und das letzte Segment zu markieren. Sämtliche Segmente werden durch einen innerhalb der Paketlebensdauer eindeutigen Identifikator gekennzeichnet. Typischerweise wird man bei einer Implementierung einen Zähler verwenden, der für jedes Segment inkrementiert wird. Weitere Header-Erweiterungen dienen zu Authentifizierungs- und Verschlüsselungs-Funktionen. Diese Sicherheitsaspekte betreffenden Header-Erweiterungen werden in Kap. 2.5 behandelt.

2.3 Unterstützung von Dienstgüten

Dienstgüten werden durch

- die Angabe einer Priorität im IP-Header und durch
- die spezielle Kennzeichnung einzelner Datenströme durch Flußmarken

unterstützt. Die durch die Flußmarken gekennzeichneten IP-Pakete können eine Sonderbehandlung bei der Verarbeitung in den IP-Routern erfahren. Die Unterstützung von Dienstgüten ist insbesondere für Realzeitanwendungen von großer Wichtigkeit.

2.3.1 Prioritätsstufen

Das Prioritätsfeld erlaubt einer Datenquelle, die gewünschte Priorität seiner Pakete zu beschreiben. Die Prioritätswerte sind in zwei Bereiche gegliedert:

- Die Werte 0-7 sind entsprechend der Tabelle 2 für Verkehrsarten vorbehalten, bei denen der Sender eine Staukontrolle durchführt, d.h. bei Verkehrsarten, bei denen in Stausituationen die Verkehrslast reduziert wird. Beispielsweise ist dies bei TCP-Verkehr durch das Slow-Start-Verfahren der Fall.
- Die Werte 8-15 werden für Verkehrsarten ohne Staukontrolle verwendet, z.B. bei mit einer konstanten Rate gesendeten Realzeitdaten. Der niedrigste Wert (8) soll dabei für Pakete verwendet werden, bei denen der Sender unter bestimmten Umständen, z.B. Stausituationen, einen Verlust tolerieren kann.

Tabelle 2: Prioritäten für Verkehrsarten mit Staukontrolle

Wert	Verkehrsart
0	nicht näher charakterisierter Verkehr
1	Füllverkehr (z.B. news)
2	Datenverkehr ohne spezielle Behandlung (z.B. electronic mail)
3	reserviert
4	Datenverkehr mit spezieller Behandlung (z.B. ftp, NFS)
5	reserviert
6	interaktiver Verkehr (z.B. telnet, X)
7	Kontrollverkehr (z.B. Routing-Protokolle, ICMP)

2.3.2 Flußmarken

Eines der interessantesten neuen Merkmale von IPv6 sind die sogenannten Flußmarken (flow labels). Diese Flußmarken werden von einem Quellknoten dazu verwendet, Pakete zu kennzeichnen, die eine besondere Behandlung durch IPv6-Router benötigen. Beispielsweise können damit Pakete gekennzeichnet werden, die Realzeitdaten enthalten oder zu Verbindungen höherer Protokolle mit bestimmten Dienstgüteanforderungen gehören.

Unter einem Fluß (flow) wird in der Internet-Welt eine Folge von Paketen von einer speziellen Quelle zu einem speziellen (Unicast- oder Multicast-) Ziel verstanden. Pakete eines Flusses müssen daher die selben Ziel- und Quelladressen enthalten. Ein Fluß ist durch eine eindeutige Kombination von Quelladresse und einer von 0 verschiedenen Flußmarke gekennzeichnet. Flußmarken erlauben die gleichzeitige Existenz mehrere Flüsse zwischen einem Quell- und einem Zielknoten. Die Flußmarke sollte generell zufällig und uniform aus dem dafür vorgesehenen Intervall gewählt werden, um ihn als Schlüssel in den Hash-Tabellen der Router einsetzen zu können.

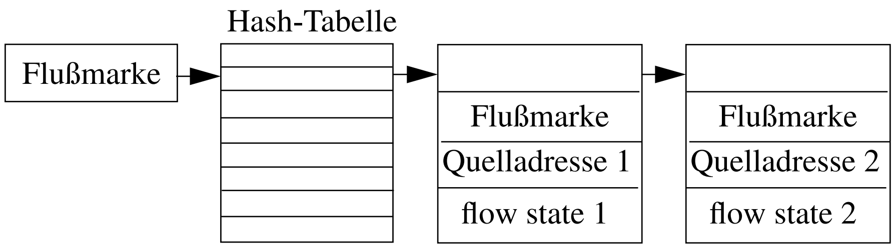

Abbildung 7: Hash-Tabelle in Routern

Die Art der speziellen Behandlung für Pakete eines Flusses kann den Routern explizit mit Hilfe eines Kontrollprotokolls (z.B. Ressourcenreservierungsprotokolle) oder durch in den Paketen direkt enthaltenen Informationen (z.B. in den Hop-by-hop-Optionen) mitgeteilt werden. In diesem Zusammenhang bietet sich insbesondere eine Integration von IPv6 mit dem Ressourcenreservierungsprotokoll RSVP (Resource Reservation Protocol) [9] an, welches derzeit innerhalb der Internet Engineering Task Force (IETF) spezifiziert wird. RSVP basiert ebenfalls auf dem Flußkonzept und erlaubt Empfängern das Reservieren von Ressourcen für einen bestimmten Fluß. In Abhängigkeit der Reservierungsnachrichten werden in den zwischen Empfänger und Sender-Endsystem liegenden Routern Systemressourcen wie CPU-Rechenzeiten oder Pufferspeicher und Netzwerkressourcen wie die Bandbreite der angeschlossenen Netze reserviert. Während bei der Integration von RSVP und IPv4 die Analyse der TCP/UDP-Port-Nummern zur Identifikation von RSVP-Flüssen unbedingt notwendig ist, machen IPv6-Flußmarken die Analyse von Port-Nummern überflüssig. Dies ist insbesondere dann wichtig, wenn die IP-Nutzlast und damit die Port-Nummern, wie in Abs. 2.5.2 beschrieben, verschlüsselt werden. Die Gültigkeitsdauer der Flußmarken wird durch die Kontrollprotokolle bzw. durch die Angaben in den Optionen festgelegt.

Flußmarken können auch ohne das Aufsetzen durch Kontrollprotokolle oder Optionen zur Vereinfachung der Paketverarbeitung durch einen Router beitragen. Empfängt ein Router ein Paket mit einer unbekannten Flußmarke, so kann der Router zunächst die erforderlichen Verarbeitungsschritte durchführen und die Ergebnisse (z.B. die Kennung des Ausgangs-Netzwerk-Interfaces oder die MAC-Adresse des nächsten Knotens) in einem Cache-Speicher ablegen. Für nachfolgende Pakete des gleichen Flusses können die bereits berechneten Ergebnisse wiederverwendet werden. Auf eine erneute komplette Analyse des Pakets kann dann verzichtet werden, die Paketverarbeitung durch den Router wird dadurch beschleunigt. Eventuelle Hop-by-hop-Optionen und Routing-Header müssen aber einen identischen Inhalt haben. Router können diese Bedingungen überprüfen und ICMP-Fehlermeldungen senden, falls diese Regeln verletzt werden. Flüsse sind in diesem Fall lediglich 6 Sekunden gültig, danach muß der Router das nächste Paket wie ein Paket mit unbekannter Flußmarke behandeln und die Ergebnisse neu berechnen. Zukünftige Router- oder Endsystemarchitekturen dürfen die Existenz von Flußmarken in allen Paketen eines Flusses nicht voraussetzen und müssen auch ohne Flußmarken leistungsfähig genug implementiert sein. In diesem Zusammenhang ist wichtig, daß nicht sämtliche IP-Pakete mit von 0 verschiedenen Flußmarken versehen werden. Ansonsten ist eine zu große Anzahl von in Routern zu unterstützenden Flüssen zu erwarten, was in gegebenfalls zu kleinen Cache-Speichern oder in zu langen Suchzeiten nach den Flußmarken resultieren kann [34].

2.4 Automatische Systemkonfiguration

2.4.1 Neighbor Discovery

Das Neighbor Discovery Protokoll (ND) [1][33] ist eine Sammlung von verschiedenen Protokollmechanismen. Die ND-Nachrichten werden über zusätzlich definierte ICMPv6-Pakete transportiert. ICMP [15] wird allgemein zum Kontrollnachrichtenaustausch zwischen IP-Instanzen verwendet. Während in ICMPv4 im wesentlichen Fehler- und Echo-Nachrichten ausgetauscht wurden, wird ICMPv6 auch für die Gruppenverwaltung eingesetzt, die bislang vom IGMP (Internet Group Management Protocol) [22] ausgeführt wurde. IGMP-Pakete werden in einer IPv6-Umgebung durch entsprechende ICMPv6-Pakete ersetzt. ND definiert ICMPv6-Pakete, welche die Grundlage für die im folgenden beschriebenen ND-Funktionen bilden.

- Router-Erkennung
  ND unterstützt die an einem Link angeschlossenen Knoten bei der Erkennung der angeschlossenen Router. Aktive Router werden durch deren Router-Advertisement-Pakete identifiziert. Router-Advertisement-Pakete werden periodisch an die All-Hosts-Multicast-Adresse gesendet. Sie können auch von einem Endsystem durch ein Router-Solicitation-Paket an die All-Routers-Multicast-Adresse angefordert werden. In diesem Fall wird ein Router-Advertisement-Paket an das anfordernde Endsystem adressiert. Als Quelladresse

enthalten Router-Advertisement- und Router-Solicitation-Pakete jeweils die Link-lokale Unicast-Adresse des Senders.
Ein Router-Advertisement-Paket kann neben den IP- und Link-Adressen des Routers auch Präfix-, Link- und Internet-spezifische Informationen enthalten. Flags geben an, auf welche Art ein Endsystem seine Adresse (zustandslos oder zustandsbehaftet) zu konfigurieren hat.

- Präfix-Erkennung
  Die Router Advertisement-Pakete enthalten Präfix-Listen für den Link, auf dem das Router-Advertisement-Paket gesendet wurde. Knoten mit gemeinsamen Präfixen befinden sich an einem gemeinsamen Link, d.h. ein Knoten kann durch Vergleichen der Präfixe mit einer Adresse eines anderen Knotens feststellen, ob sich dieser Knoten am gleichen Link befindet oder nicht. Knoten-Adressen mit nicht passenden Präfixen sind in der Regel nicht am selben Link angeschlossen und daher nur über einen Router erreichbar. Des weiteren werden Präfixe zur Adreßkonfiguration (s.unten) verwendet.

- Parameter-Erkennung
  Durch die Analyse der Router-Advertisement-Pakete lernen die Knoten Link-spezifische Parameter wie die MTU-Größe oder Internet-spezifischen Parameter wie der maximale Hop-Limit-Wert. Des weiteren wird in den Router-Advertisement-Paketen angegeben, auf welche Art die Adreßkonfiguration erfolgen soll. Die Parameter-Erkennung erleichtert sehr stark die Konfiguration von IP-Knoten. Während zum Installieren von IPv4-Knoten bislang umfangreiche Mengen von Parametern vor dem Anschluß des Knotens an das Netz zu konfigurieren waren, können diese in einer IPv6-Umgebung automatisch konfiguriert werden.

- zustandslose (stateless) automatische Adreßkonfiguration [42]
  Die zustandslose automatische Konfiguration von IPv6-Adressen erleichtert die Konfiguration einer eindeutigen Adresse eines Knoten-Interfaces. Die IP-Adresse setzt sich bei der Benutzung dieses Mechanismus aus einem Link-spezifischen Präfix und einer individuellen Interface-Kennung zusammen. Beispiele für Interface-Kennungen bestehen in MAC-Adressen gemäß IEEE802 oder in der einfachen Durchnumerierung von Interfaces. Zur Konfiguration wartet ein Knoten auf die periodisch gesendeten Router-Advertisement-Pakete oder er fordert ein solches mit einem Router-Solicitation-Paket an. Die zustandslose automatische Adreßkonfiguration kann nur dann erfolgen, falls an einem Link Router angeschlossen sind. Befindet sich kein Router in einem Subnetz oder befindet sich ein Knoten in der Adreßkonfigurationsphase, so wird zur Kommunikation mit Knoten am selben Link die Link-lokale Adresse benutzt.

- Adreßauflösung
  Die Adreßauflösung dient zum Bestimmen einer Link-Adresse eines am gleichen Link angeschlossenen Knotens, falls lediglich dessen IP-Adresse bekannt ist. Dazu wird ein Neighbor-Solicitation-Paket an die Solicited-Nodes-Multicast-Adresse gesendet. Die Antwort in Form eines Neighbor-Advertisement-Pakets enthält die Link-Adresse des gesuchten Knotens. Üblicherweise wird nicht bei jedem zu sendenden Paket eine Adreßauflösung durchgeführt, sondern die aufgelösten Adressen werden in einem Cache-Speicher abgelegt. Ein an die All-Nodes-Multicast-Adresse gerichtetes Neighbor-Advertisement-Paket kann auch eine Änderung der Link-Adresse ankündigen.
  Dadurch daß Neighbor-Solicitation-Pakete an die Link-lokale Solicited-Nodes-Multicast-Adresse gesendet werden, werden nicht alle Knoten eines Links, sondern in der Regel nur kleine Teilmenge der an ein Link angeschlossenen Knoten adressiert. Beim Address Resolution Protocol (ARP) in einer IPv4-Umgebung werden Adreßanfragen per Broadcast übertragen. Durch die Solicited-Nodes-Multicast-Adresse werden hingegen nur Knoten angesprochen, deren 32 niederwertigste Adreßbits der Unicast-Adresse mit der Solicited-

Nodes-Multicast-Adresse identisch sind.

- Bestimmung des nächsten Knotens
  Zum Senden eines IP-Pakets an einen bestimmten Knoten muß durch den Vergleich der Zieladresse und der gelernten Adreß-Präfix-Werte bestimmt werden, ob das Paket direkt an den Knoten geschickt werden kann, d.h. ob der Knoten am selben Link angeschlossen ist, oder ob das Paket zunächst an einen Router zur Weitervermittlung gesendet werden muß.
- Unerreichbarkeitsbestimmung von Nachbarknoten
  Mit dem Senden eines Neighbor-Solicitation-Pakets wird der adressierte Knoten zum Senden einer entsprechenden Antwort in Form einer Neighbor-Advertisement-Pakets aufgefordert. Ein Knoten kann damit feststellen, ob der Nachbarknoten noch erreichbar ist oder nicht.
- Erkennung von Adreßduplikaten
  Adreßduplikate, d.h. die doppelte Benutzung einer einzigen IP-Adresse, werden durch die zustandslose automatische Adreßkonfiguration erkannt. Um festzustellen, daß die Adresse eindeutig ist, reicht es aus, die Eindeutigkeit der Interface-Kennung im Bereich des Links zu überprüfen. Hierzu wird ein Neighbor-Solicitation-Paket an die Solicited-Nodes-Multicast-Adresse gesendet. Die in möglichen Antworten enthaltenen Adressen werden anschließend überprüft.
- Redirect-Funktion, d.h. Lernen eines besseren ersten Knotens für ein bestimmtes Ziel
  In bestimmten Fällen ist es möglich, daß ein an den selben Link angeschlossener Knoten X ein Adreß-Präfix benutzt, welches nicht in der von einem Router verbreiteten Präfix-Liste enthalten ist. In diesem Fall geht ein Knoten beim Senden eines Pakets an X davon aus, das Paket an den Router senden zu müssen, damit dieser das Weiterleiten übernimmt. Ein Redirect-Paket kann den Sender dann darauf hinweisen, daß das Ziel über den gleichen Link zu erreichen ist.

2.4.2 ND in ATM

Falls ND in ATM-Netzen ausgeführt werden soll, treten diverse Probleme auf. Üblicherweise werden zur automatischen Adreßkonfiguration die Subnetz-spezifischen Adressen für die Interface-Kennung innerhalb der IP-Adresse verwendet. ATM-Adressen nach E.164 oder gemäß ATM Forum NSAP sind für diese Zwecke jedoch zu lang. Es ist deshalb erforderlich die ATM-Adressen auf eine innerhalb eines logischen Subnetzes eindeutige Kennung abzubilden. Ein Lösungsvorschlag wird in [1] vorgestellt.

Ein weiteres Problem besteht darin, daß die ND-Protokollmechanismen auf IP-Multicast basieren. IP-Multicast [20][22][45] wiederum ist mit seiner Gruppenverwaltung auf “shared medium”-Subnetze ausgelegt. Die dabei verwendeten Mechanismen arbeiten über ATM-Netzen nicht effizient. Mit dem MARS-Ansatz (MARS: multicast address resolution server) befindet sich derzeit eine Lösung zur effizienteren Realisierung von IP-Multicast über ATM innerhalb der IETF in der Diskussion [2]. Ein aufwendigerer Ansatz wird in [40] beschrieben. Dabei wird eine hierarchisch aufgebaute, baumartige Struktur von sogenannten ND-Servern dazu benutzt, die verschiedenen ND-Pakete zwischen Routern und Endsystemen auszutauschen. Ein weiterer Vorschlag [6] basiert darauf, daß verschiedene ND-Funktionen, die in NBMA-Netzen (NBMA: non-broadcast multiple access) wie ATM, Frame Relay oder ISDN, schwer realisierbar sind, auch nicht durch ND erbracht werden. Beispielsweise kann die IP-Adreßauflösung in NBMA-Netzen durch das Protokoll NHRP [28] erfolgen.

2.4.3 Dynamic Host Configuration Protocol

Das Dynamic Host Configuration Protocol (DHCP) ist im IPv6-Umfeld für die zustandsbehaftete (statefull) Adreßkonfiguration zuständig [7]. DHCPv6 stellt Mechanismen für die automatische Konfiguration von IPv6-Adressen und für die automatische, dynamische Registrierung von Host-Namen im Domain Name System (DNS) [43] zur Verfügung. DHCP basiert auf einem Client/Server-Konzept. Ein Client sendet ein Request-Paket an einen Server und erhält ein Response-Paket als Antwort zurück. Die Antwort enthält die vom Client angeforderten Informationen wie die IP-Adresse oder den Host-Namen. Der Client bestätigt die Antwort mit einem Confirmation-Paket. Im Gegensatz zur zustandslosen, automatischen Adreßkonfiguration wird nicht an jedem Link ein Server benötigt. Gegebenenfalls müssen Relay-Agenten die Client-Pakete an entfernte Server weiterleiten (vgl. Abbildung 8).

Der Client sendet das Request-Paket an die reservierte DHCP-Server/Relay-Agenten-Multicast-Adresse, falls ihm der Server nicht bekannt ist. Als Quelladresse verwendet der Client seine IP-Adresse oder falls diese noch nicht bekannt ist, seine Link-lokale Adresse. Der Client kann vom Server eine IP-Adresse oder einen Host-Namen anfordern. Die Adresse und der Host-Namen haben eine begrenzte Gültigkeit, die zwischen Client und Server verhandelt wird. Nach Ablauf der Gültigkeit muß der Client deren Verlängerung beantragen.

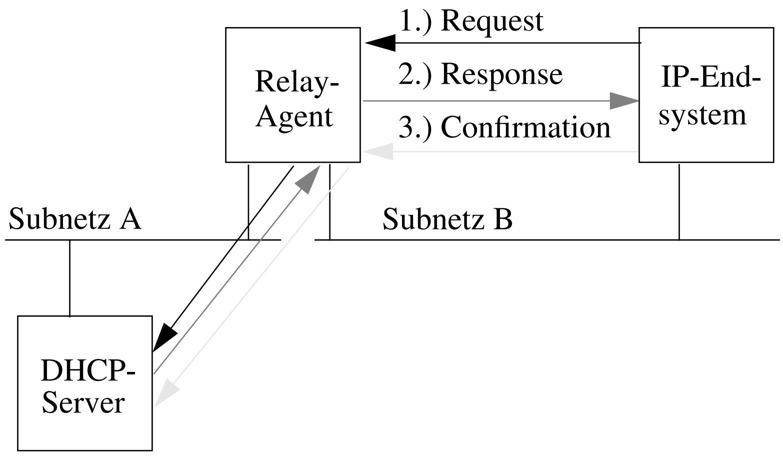

Abbildung 8: DHCP-Server und Relay-Agent

2.5 Sicherheitsaspekte

Die IPv6-Sicherheitsfunktionen umfassen Erweiterungen zur Unterstützung von Authentifizierungen und vertraulicher Kommunikation. Authentifizierungen stellen sicher, daß die Daten vom angegebenen Sender wirklich gesendet wurden und daß sie unverfälscht beim Empfänger ankommen, während Verschlüsselungen der Pakete sicherstellen, daß nur der beabsichtigte Empfänger die Daten in eine lesbare Form verwandeln kann.

2.5.1 Authentifizierung

Zur Überprüfung der Echtheit der Daten wird der Authentifizierungs-Header [4] verwendet. Innerhalb des Authentifizierungs-Headers existieren Felder für Authentifizierungsdaten, deren Inhalt vom verwendeten Sicherheitsalgorithmus abhängt. Bei Verwendung des MD5-Algorithmus (MD: message digest) [30][39], wird aus dem zu sendenden Paket und einem zuvor zwischen den Kommunikationspartnern vereinbarten Schlüssel eine 128-bit-Kennung berechnet und in das entsprechende Feld im Authentifizierungs-Header eingetragen. Der Empfänger, der im Besitz des korrekten Schlüssels sein muß, führt dieselben Operationen wie der Sender durch und wird nur dann dieselben Authentifizierungsdaten wie der Sender berechnen, wenn seine Operationen auf dem korrekten Schlüssel basieren und die Daten unververfälscht, d.h. ohne Bitfehler, Manipulationen Dritter usw. übertragen wurden.

2.5.2 Verschlüsselung

Authentifizierungs-Mechanismen verändern die Daten des Pakets nicht und können daher keinen vertraulichen, abhörsicheren Datenaustausch ermöglichen. Für diese Zwecke wird in IPv6 der DES-CBC-Verschlüsselungsalgorithmus vorgeschlagen (DES: Data Encryption Standard, CBC: Cipher Block Chaining). Der IPv6-Verschlüsselung-Header [5][31] enthält Initialisierungsdaten und Informationen über eine vorher zu etablierende Sicherheits-Assoziation. Der Nutzlast, d.h. den verschlüsselten Daten, folgen Padding-Bytes, da die üblichen Verschlüsselungsalgorithmen auf 8 Byte Grenzen arbeiten und die Anzahl der zusätzlich erforderlichen Padding-Bytes.

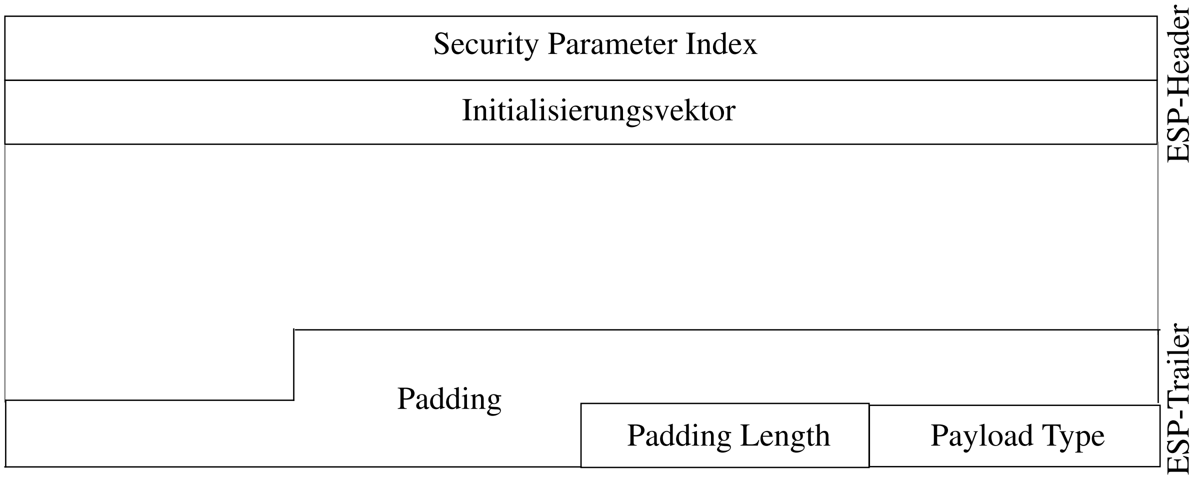

Abbildung 9: Verschlüsselungs-Header

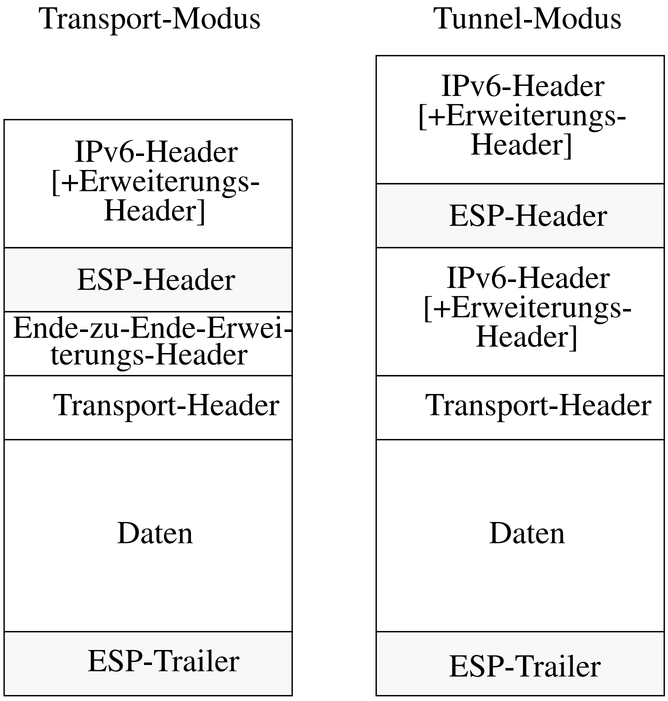

Abbildung 10: Paketformate bei Verschlüsselung

Für die Verschlüsselung werden zwei verschiedene Modi definiert [2]:

- Beim *Tunnel-Modus* wird das gesamte IP-Paket verschlüsselt und ein neuer unverschlüsselter IP-Header erzeugt.
- Beim *Transport-Modus* werden die Nutzlast und die Ende-zu-Ende-Erweiterungs-Header verschlüsselt.

Die Verschlüsselung kann mit Hilfe dieser beiden Modi

- Ende-zu-Ende (Transport-Modus)
- zwischen Endsystemen und Security-Gateways (Tunnel-Modus) oder
- zwischen zwei Security-Gateways (Tunnel-Modus) erfolgen.

Eine Verschlüsselung zwischen Security-Gateways ist zum Beispiel sinnvoll, wenn private virtuelle Netze auf einem unsicheren Backbone-Netz wie z.B. dem Internet aufbauen. In diesem Fall werden die Verschlüsselungsfunktionen nur in den Security-Gateways an den Grenzen zum Backbone implementiert, die Implementierung von Verschlüsselungsfunktionen in den Endsystemen ist dann nicht notwendig. Eine Verschlüsselung zwischen Security-Gateway und Endsystem ist z.B. in einem Szenario sinnvoll, in dem ein Endsystem direkt über einen unsicheren Zugang, z.B. eine Telefonleitung oder Funkstrecke, an ein privates Netz angeschlossen wird. Bei der Verschlüsselung zwischen Security-Gateway und Endsystem bzw. zwischen zwei Security-Gateways muß das zu verschlüsselnde Paket in einen weiteren IPv6-Header einschließlich Verschlüsselungs-Header eingekapselt werden.

2.6 Unterstützung mobiler Endsysteme

Mobile IP-Knoten werden üblicherweise durch eine transiente Adresse, die sogenannte care-of-Adresse, und eine permanent gültige Adresse, die Home-Adresse, identifiziert. Eine care-of-Adresse wird einem mobilen Knoten zugewiesen, wenn er sich nicht in seinem Home-Netz befindet. Die Zuweisung der care-of-Adresse kann über Konfigurationsprotokolle wie z.B. DHCPv6 erfolgen. Nach der Zuweisung ist es erforderlich, daß dem sogenannten Home-Agenten die care-of-Adresse mitgeteilt wird. Die Mitteilungsprozedur der für eine Home-Adresse gültigen care-of-Adresse wird auch als Registrierung bezeichnet. Der Home-Agent ist ein Router im Home-Netz, welcher die Registrierungs-Informationen verwaltet. Nach der Registrierung ist der Home-Agent in der Lage, die an die Home-Adresse des mobilen Knotens gesendeten Pakete zu filtern und an die care-of-Adresse weiterzuleiten. Die care-of-Adresse kann mit dem mobilen Knoten selbst oder mit einem Foreign-Agenten verbunden sein. Ein Foreign-Agent ist ein Router in dem Netz, in dem sich der mobile Knoten aktuell aufhält. Über ihn kann der mobile Knoten Pakete senden bzw. empfangen.

In IPv4 sendet der Home-Agent die an einen mobilen Knoten adressierten Pakete über einen sogenannten Tunnel. Die IP-Pakete werden durch den Home-Agenten, welcher den Anfang des Tunnels darstellt, in einen zusätzlichen IP-Rahmen eingekapselt und am Ende des Tunnels wieder in die ursprüngliche Form verwandelt, indem der Einkapselungs-Header wieder entfernt wird [35]. Die Zieladresse im Einkapselungs-Header ist dabei das Ende des Tunnels, d.h. die care-of-Adresse. Falls die care-of-Adresse einen Agenten adressiert, so ist dieser für das endgültige Weiterleiten an den durch die Home-Adresse gekennzeichneten mobilen Knoten verantwortlich. Die Tunnel-Technik, d.h. das Einkapseln von IP-Paketen in IP-Pakete, wird auch beim Übergang von IPv4 nach IPv6 (s. Kap. 4) und im Multicast-Backbone des Internet eingesetzt, wobei Multicast-fähige IPv4-Router über nicht-Multicast-fähige IPv4-Router verbunden werden.

In IPv6 kann bei mobilen Knoten auf die Tunnel-Technik verzichtet werden, indem die beiden Adressen, d.h. die care-of-Adresse und die Home-Adresse, in den Routing-Header aufgenommen werden [36]. Ein an einen mobilen Knoten weiterzuleitendes Paket wird dann vom Home-Agenten über die care-of-Adresse an die Home-Adresse geleitet. Eine weitere Vereinfachung bei IPv6 besteht darin, daß die Informationen zur Registrierung in IPv6-Optionen oder als ICMPv6-Nachrichten ausgetauscht werden. Bei IPv4 werden die Registrierungsnachrichten über UDP ausgetauscht. Das Mitteilen einer neuen care-of-Adresse erfolgt in IPv6 über die Binding Update Option, wodurch in normalen IPv6-Paketen anderen IPv6-Systemen die care-of-Adresse mitgeteilt werden. Pakete an den mobilen Knoten können dann auch ohne den Umweg über den Home-Agenten direkt an den mobilen Knoten, d.h. direkt an die care-of-Adresse, gesendet werden. In Abbildung 11 sendet ein IP-Endsystem zunächst ein Paket an die Home-Adresse des mobilen Endsystems. Der Home-Agent leitet die Nachricht (ggf. über den Foreign-Agenten) an das mobile Endsystem weiter. Das mobile Endsystem teilt dem Sender des ersten Pakets seine care-of-Adresse mit (Binding-Update-Option), die der Sender in den nachfolgenden Paketen in die Adressierungsinformation aufnimmt. Das Paket wird dann direkt an die care-of-Adresse gesendet, wodurch der Weg des Pakets in diesem Fall verkürzt wird. In IPv4 mußten die Binding-Update-Informationen durch spezielle Nachrichten verschickt werden.

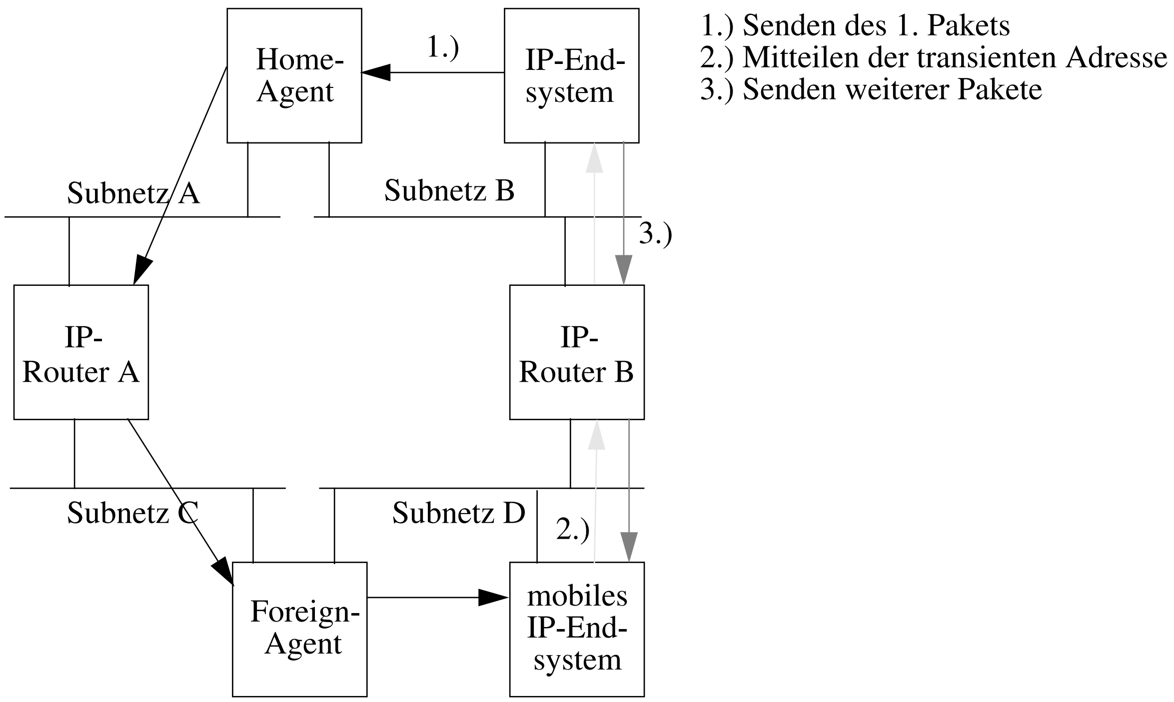

Abbildung 11: Unterstützung mobiler IP-Endsysteme

3 Anforderungen an andere Protokolle

Das modifizierte Adreßformat erfordert zunächst die Anpassung der im IPv6-Umfeld verwendeten Routing-Protokolle RIP [29] und OSPF [17].

Darüber hinaus muß jedes Transportprotokoll oder andere höhere Protokolle, welche IP-Adressen zur Berechnung ihrer Prüfsumme verwenden, aufgrund des veränderten IPv6-Adreßformats modifiziert werden. Für TCP-Implementierungen mußte ein neuer Pseudo-Header definiert werden, der in die TCP-Prüfsummenberechnung mit eingeht und z.B. die IP-Adressen und die Paketlänge enthält [21].

Beim Einsatz von UDP über IPv6, muß nun immer die UDP-Prüfsumme berechnet werden. Bei UDP über IPv4 war die Prüfsummenberechnung nicht zwingend erforderlich. Der IPv6-Header und damit die IP-Adressen werden aber nicht mehr durch eine Prüfsumme abgesichert. Eine Prüfsumme durch das darüberliegende Protokoll soll deshalb eine falsche Auslieferung aufgrund von verfälschten IP-Adressen vermeiden.

Ein weiterer Aspekt betrifft die maximale Länge von Paketen auf der nächst höheren Protokollebene. Durch den auf mindestens 40 Byte angewachsenen IP-Kopf reduziert sich die maximale Datenlänge auf den höheren Protokollebenen. Außerdem erfordern die neuen Adreßformate Änderungen an der Socket-Schnittstelle [24], die sich wiederum in notwendigen Anpassungen der darauf aufsetzenden Anwendungen auswirken.

4 Übergangsstrategien

Die entwickelten Strategien zum Übergang von IPv4 auf IPv6 sollen eine inkrementelle und einfache Migration von IPv4 nach IPv6 ermöglichen [23][25]. IPv4- und IPv6-Knoten sollen in einem gemeinsamen Netz existieren können. Es ist zu erwarten, daß in einer Übergangsphase alle IPv6-Knoten auch einen IPv4-basierten Protokollstack besitzen, um mit der großen Anzahl von reinen IPv4-basierten Knoten weiterhin kommunizieren zu können. Man geht auch davon aus, daß nicht alle IPv4-Knoten auf IPv6 umgestellt werden und daß es zumindest in absehbarer Zeit immer reine IPv4-Knoten wie z.B. Drucker geben wird. Reine IPv6-Knoten sind eventuell in einem späteren Stadium möglich. Damit sie auch mit IPv4-Knoten kommunizieren können, wäre ein Übersetzungsmechanismus zwischen IPv4- und IPv6-Paketen notwendig, was aufgrund der unterschiedlichen Adreßformate nur schwer realisierbar ist.

Für IPv6 soll die bestehende IPv4-Routing-Infrastruktur verwendet werden, solange noch reine IPv4-Router existieren. Dazu wird die bereits in Kap. 2.6 beschriebene Tunnel-Technik [16] angewendet. IPv6-Pakete werden in IPv4-Pakete eingepackt, d.h. mit einem zusätzlichen IPv4-Kopf versehen, und über einen oder mehrere IPv4-Router an den nächsten IPv6-fähigen Knoten übertragen. Letzterer streift den IPv4-Header wieder ab und verarbeitet das IPv6-Paket entsprechend. Dieser Knoten wird auch als Tunnel-Ende bezeichnet. Bei der Tunnel-Technik müssen die IPv4-Zieladressen je nach Situation unterschiedlich bestimmt werden. Ist das ursprüngliche Ziel auch gleichzeitig das Tunnel-Ende, so wird automatisch aus der IPv4-kompatiblen Adresse des IPv6-Zielknotens die IPv4-Adresse extrahiert und als Zieladresse im IPv4-Header eingesetzt. Man spricht wegen der automatischen Abbildung der IPv6-Adresse auf die IPv4-Adresse auch von *automatischem Tunneln* (vgl. Kommunikation zw. H1 und H3 in Abbildung 12). IPv6-Knoten müssen daher mit einer IPv4-kompatiblen Adresse ausgestattet sein, um die Tunnel-Technik unterstützen zu können. Ist das Tunnel-Ende jedoch ein Router muß die IPv4-Zieladresse aus der Routing-Information beim Tunnel-Anfang ermittelt werden, was *konfiguriertes Tunneln* genannt wird (vgl. Kommunikation zw. H1 und H2 in Abbildung 12). Es ist z.B. sinnvoll, den nächsten erreichbaren IPv4/IPv6-Router als Tunnel-Ende zu konfigurieren.

Ein IPv4-Tunnel kann aus IPv6-Sicht als Link mit einer sehr großen MTU-Größe, d.h. 65535-20 byte, angesehen werden. Der Vorteil bei diesem Ansatz ist, daß innerhalb des IPv4-Pfads erzeugte ICMP-Nachrichten, die ein zu großes Paket anzeigen, nur für Pakete welche die o.g MTU-Größe übertreffen, an die IPv6-Quelle weitergeleitet werden müssen. Allerdings ist es bei einem solchen Ansatz wahrscheinlich, daß die IPv4-Knoten zusätzlichen Segmentierungs- und Reassemblierungsaufwand übernehmen müssen, da die wirkliche IPv4-MTU-Größe für die IPv6-Knoten unsichtbar bleibt und daher bei der Wahl der Paketgröße nicht berücksichtigt wird. Um das Segmentieren und Reassemblieren entlang des IPv4-Tunnels zu vermeiden, sollten der Tunnel-Anfang entsprechende ICMPv6-Nachrichten an den Quellknoten senden, falls dessen Pakete die MTU-Größe entlang des IPv4-Tunnels und die minimale IPv6-MTU-Größe übertreffen.

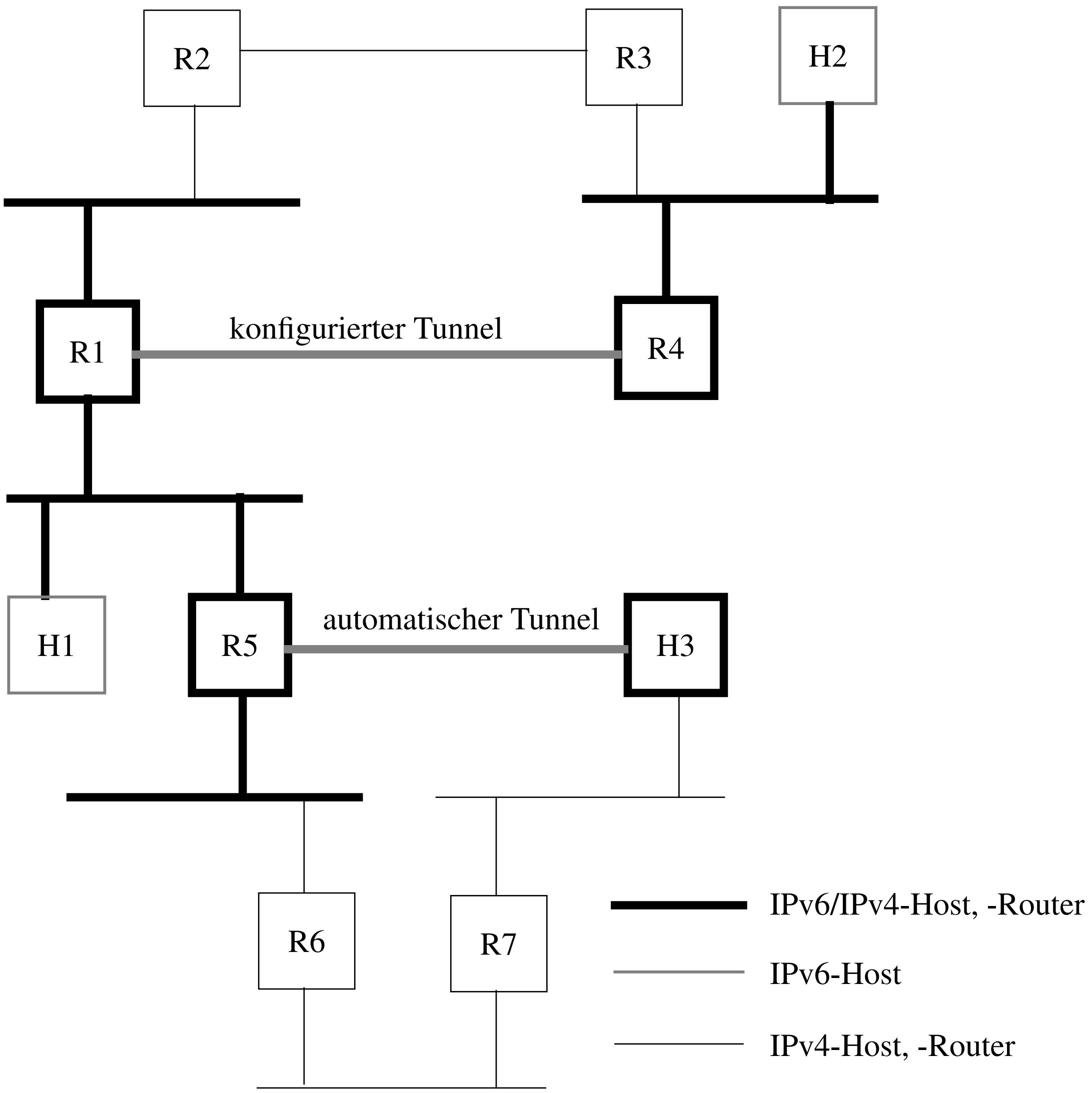

Abbildung 12: Typisches Netzszenario beim Übergang von IPv4 nach IPv6

5 Ausblick

Derzeit werden auf verschiedenen Rechnerplattformen prototypische Implementierungen entwickelt und gegeneinander getestet. Die Tests dienen zum einen dazu, die Implementierungen stabiler zu gestalten, zum anderen sollen aber auch Mängel in den Spezifikationen aufgedeckt und beseitigt werden. Eine breite Einführung und Verwendung von IPv6 wird daher erst für die Jahre 1998-2000 erwartet.

Ein derzeit offener Aspekt ist die weitere Entwicklung von TCP. Es wird zur Zeit diskutiert, ob durch IPv6 erforderlichen Modifikationen (z.B. Definition eines neuen IP-Pseudo-Headers) auf das Nötigste beschränkt werden sollen oder ob wie im IPng-Projekt eine völlig neue Protokollarchitektur entwickelt werden soll. Ein weiterer Ansatzpunkt besteht z.B. darin, daß TCP derzeit keine Änderung der IP-Zieladresse zuläßt, was in einem IPv6-Umfeld jedoch vorkommen kann. Des weiteren wird häufig die bei TCP fehlende Möglichkeit einer selektiven Übertragungswiederholung kritisiert.

Referenzen

[1] G. Armitage: IPv6 and Neighbor Discovery over ATM, Internet Draft, draft-ietf-ipatm-ipv6nd-01.txt, Februar 1996.

[2] G. Armitage: Support for Multicast over UNI 3.0/3.1 based ATM Networks, Internet Draft, draft-ietf-ipatm-ipmc-12.txt, Februar 1996.

[3] R. Atkinson: Security Architecture for the Internet Protocol, RFC 1825, August 1995.

[4] R. Atkinson: IP Authentication Header, RFC 1826, August 1995.

[5] R. Atkinson: IP Encapsulating Security Payload (ESP), RFC 1827, August 1995.

[6] R. Atkinson, D. Haskin, J. Luciani: IPv6 over NBMA Networks, Internet Draft, draft-ahl-ipv6-nbma-00.txt, Februar 1996.

[7] J. Bound: Dynamic Host Configuration Protocol for IPv6, Internet Draft, draft-ietf-dhc-dhcpv6-04.txt, Februar 1996.

[8] J. Bound, B. Carpenter, D. Harrington, J. Houldsworth, A. Lloyd: OSI NSAPs and IPv6, Internet Draft, draft-ietf-ipngwg-nsap-ipv6-00.txt, Juli 1995.

[9] R. Braden, L. Zhang, D. Estrin, S. Herzog, S. Jamin: Resource Reservation Protocol (RSVP) - Version 1 Functional Specification, Internet Draft, draft-ietf-rsvp-spec-10.ps, Februar 1996.

[10] S. Bradner, A. Mankin: The Recommendation for the IP Next Generation Protocol, RFC 1752, Januar 1995.

[11] T. Braun: IPng: Das Internet-Protokoll der nächsten Generation, in F. Kaderali (Hrsg.): Corporate Networks und neue Carrier-Netze: Architekturen, Strategien und Entwicklungstrends, Verlag Online-GmbH, Februar 1996, S. C331.1-21.

[12] T. Braun, M. Zitterbart: Hochleistungskommunikation: Transportdienste und -protokolle, Oldenbourg-Verlag, 1996.

[13] D. Clark, D. Tenenhouse: Architectural Considerations for a New Generation of Protocols, ACM SIGCOMM '90, S. 200-208.

[14] D. Comer: Internetworking with TCP/IP, Vol. I: Principles, Protocols, and Architecture, Prentice Hall, 1991.

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

[16] A. Conta, S. Deering: Generic Packet Tunneling in IPv6 Specification, Internet Draft, draft-ietf-ipngwg-ipv6-tunnel-01.txt, Februar 1996.

[17] R. Coltun, D. Ferguson: OSPF Version 2 For IP Version 6, Internet Draft, draft-ietf-ospf-ospfv6-01.txt, Februar 1996.

[18] M. Crawford: A Method for the Transmission of IPv6 Packets over Ethernet Networks, Internet Draft, draft-ietf-ipngwg-ethernet-ntwrks-02.txt, März 1996.

[19] M. Crawford: A Method for the Transmission of IPv6 Packets over FDDI Networks, Internet Draft, draft-ietf-ipngwg-fddi-ntwrks-03.txt, März 1996.

[20] S. Deering: Host Extensions for IP Multicasting, RFC 1112, August 1989.

[21] S. Deering, R. Hinden: Internet Protocol, Version 6 (IPv6) Specification, RFC 1883, Dezember 1995.

[22] W. Fenner: Internet Group Management Protocol, Version 2, Internet Draft, draft-ietf-idmr-igmp-v3-00.txt, August 1995.

[23] R. Gilligan, E. Nordmark: Transition Mechanisms for IPv6 Hosts and Routers, Internet-Draft, draft-ngtrans-trans-mech-02.txt, Dezember 1995.

[24] R. Gilligan, S. Thomson, J. Bound: IPv6 Program Interfaces for BSD Systems, Internet-Draft, draft-ietf-ipngwg-bsd-api-04.txt, Januar 1996.

[25] D. Haskin, R. Callon: Routing Aspects Of IPv6 Transition, Internet Draft, draft-haskin-ipv6-routing-aspects-01.txt, Juni 1995.

[26] R. Hinden: Simple Internet Protocol Plus White Paper, RFC 1710, Oktober 1994.

[27] R. Hinden, S. Deering: IP Version 6 Addressing Architecture, RFC 1884, Dezember 1995.

[28] D. Katz, D. Piscitello, B. Cole, J. Luciani: NBMA Next Hop Resolution Protocol (NHRP), Internet Draft, draft-ietf-rolc-nhrp-07.txt, Dezember 1995.

[29] G. Malkin: RIPng for IPv6, Internet Draft, draft-ietf-rip-riping-01.txt, April 1995.

[30] P. Metzger, W. Simpson: IP Authentication using Keyed MD5, RFC 1828, August 1995.

[31] P. Metzger, P. Karn, W. Simpson: The ESP DES-CBC Transform, RFC 1829, August 1995.

[32] J. McCann, S. Deering: Path MTU Discovery for IP version 6, Internet Draft, draft-ietf-ipngwg-pmtuv6-01.txt, Februar 1996.

[33] T. Narten, E. Nordmark, W. Simpson: Neighbor Discovery for IP Version 6 (IPv6), Internet Draft, draft-ietf-ipngwg-discovery-06.txt, März 1995.

[34] C. Partridge: Using the Flow Label Field in IPv6, RFC 1809, Juni 1995.

[35] C. Perkins: IP Mobility Support, Internet Draft, draft-ietf-mobileip-protocol-15.txt, Februar 1995.

[36] C. Perkins, D. Johnson, Mobility Support in IPv6, Internet Draft, draft-perkins-mobility-ipv6-02.txt, Juli1995.

[37] Y. Rekhter, T. Li: An Architecture for IPv6 Unicast Address Allocation, RFC 1887, Dezember 1995.

[38] Y. Rekhter, P. Lothberg, R. Hinden, S. Deering, J. Postel: An IPv6 Provider-based Unicast Address Format, Internet Draft, draft-ietf-ipngwg-unicst-addr-fmt-03.txt, Februar 1996.

[39] R. Rivest: The MD5 Message-Digest Algorithm, RFC 1321, April 1992.

[40] P. Schulter: A Framework for IPv6 over ATM, Internet Draft, draft-schulter-ipv6atm-framework-01.txt, Februar 1996.

[41] H. Schulzrinne: IPv6 - The Next Generation Internet Protocol, Praxis der Information und Kommunikation (PIK), Heft 3, 18. Jahrgang, Juli-September 1995, S. 165-167

[42] S. Thomson, Th. Narten: IPv6 Stateless Address Autoconfiguration, Internet Draft, draft-ietf-addrconf-ipv6-auto-07.txt, Dezember 1995.

[43] S. Thomson, C. Huitema: DNS Extensions to support IP version 6, RFC 1886, Dezember 1995.

[44] S. Thomas: A Method for the Transmission of IPv6 Packets over Token Ring Networks, Internet Draft, draft-ietf-ipngwg-token-ring-01.txt, Januar 1996.

[45] D. Waitzman, C. Partridge, S. Deering: Distance Vector Multicast Routing Protocol, RFC 1075, November 1988.