1 Einleitung

Die Liefereinheit *open*Net Server enthält das Transportsystem von BS2000/OSD. Der BS2000 Communication Manager BCAM unterstützt neben den proprietären NEA-Protokollen ISO- und TCP/IP-Protokolle. Bisher wurden ausschließlich TCP/IP-Protokolle der Version 4 unterstützt, *open*Net Server V2.0 führt in der in diesem Handbuch beschriebenen Stufe 1 TCP/IP-Protokolle der neuen Version 6 (IPv6) ein.

1.1 Zielgruppen des Handbuchs

Das vorliegende Handbuch wendet sich an alle, die

– über die Einführung von IPv6 in BS2000/OSD entscheiden,

– die IPv6-Funktionalität auf BS2000/OSD-Mainframes nutzen oder

– IPv6 in BS2000/OSD installieren wollen.

Kenntnisse des Betriebssystems BS2000/OSD sowie der TCP/IP-Grundbegriffe werden vorausgesetzt.

1.2 Konzept des Handbuchs

Das vorliegende Handbuch beschreibt im Einführungsteil allgemeingültig die von den entsprechenden Gremien verabschiedete und geplante IPv6-Funktionalität und stellt im Umstellungskapitel die BS2000/OSD-spezifische Realisierung der IPv6 Stufe 1 vor.

- Kapitel 2 enthält die wichtigsten Fakten zur bisherigen Entwicklung des Internet.
- Kapitel 3 wendet sich an Entscheider und gibt einen Überblick über die kommerziellen Belange, die Protokollgrundlagen und den aktuellen Stand der IPv6-Entwicklung sowie über die aktuelle Diskussion für und wider IPv6.
- Kapitel 4 beschreibt die funktionalen und technischen Aspekte von IPv6 in einer für Netzwerkspezialisten und Programmierer interessanten Tiefe. Dieses Kapitel empfiehlt sich durch seine detaillierte Beschreibung auch für technisch interessierte Laien.

- Kapitel 5 stellt die Basismechanismen für den Übergang von IPv4 nach IPv6 dar und erläutert den Übergang anhand ausgewählter Anwendungsszenarien.
- Kapitel 6 beschreibt den aktuellen Stand der Implementierung von IPv6 in BS2000/OSD. Diese in IPv6 Stufe 1 realisierte Funktionalität wird hinsichtlich Generierung, Änderungen gegenüber IPv4 und Nutzung vorgestellt und mit Beispielen veranschaulicht.
- Der Anhang liefert vertiefende Informationen zu den Themen IPv6-Adresszuweisung und DNS-Nutzung.

2 Entwicklung des Internet

Das Internet als eine Menge von weltweit zusammengeschlossenen, miteinander kommunizierenden Netzen ist eine einzige Erfolgsgeschichte. Riesige Datenmengen werden von Millionen von Benutzern täglich über das Internet bewegt. Obwohl die Internet Protokoll-Familie vielfältige Protokolle umfasst, wird sie gemeinhin als TCP/IP (Transmission Control Protocol und Internet Protocol) bezeichnet. TCP/IP-Implementierungen sind praktisch auf allen eingesetzten Betriebssystemen und Hardware-Plattformen verfügbar.

Die Anfänge von TCP/IP gehen auf das Jahr 1968 zurück. Die Advanced Research Projects Agency (ARPA) des amerikanischen Verteidigungsministeriums (DoD) entwickelte zu diesem Zeitpunkt das ARPAnet zur der gemeinsamen Nutzung von Ressourcen für verschiedene, an Forschungsprojekten beteiligte Mitglieder. Das ARPAnet nahm 1969 seinen Dienst auf. Im Laufe der folgenden Jahre zeigten sich im praktischen Betrieb diverse Schwächen und Unzulänglichkeiten. Im Jahre 1974 wurden dann basierend auf den gewonnenen Erkenntnissen die Definitionen des Internet Protokolls entwickelt. 1978 wurde nach vielfältigen Erprobungsphasen TCP/IP vom DoD als das Standard-Protokoll für seine Datenkommunikationsnetze festgelegt.

Die Vorteile der IP-Technologie sind hinlänglich bekannt und unbestritten. Mit der Technologie des Internet steht ein offener Standard zur zuverlässigen Verbindung heterogener Umgebungen bei gleichzeitig hoher Wirtschaftlichkeit zur Verfügung. Dazu kommen diverse Dienste, von denen das World Wide Web und E-Mail zweifellos die wichtigsten sind und dem Internet zu einem ungebrochenen Siegeszug verholfen haben.

Das explosionsartige Wachstum des Internet führte zu einem der Problembereiche, der Vergabe von Internet-Adressen. Hervorgerufen durch den steigenden Anbindungswunsch vieler Unternehmen und Service-Provider steigt der Bedarf an IP-Adressen enorm. Alle möglichen Produkte werden in Zukunft internet-tauglich sein. Die Hersteller solcher Geräte sind auf einen ausreichend großen Adressraum angewiesen. Rein theoretisch bietet die 32 bit-Adressierung des Internet Protokolls IPv4 die Möglichkeit, bis zu 4,3 Milliarden Endgeräte zu unterstützen. Jedoch war die bisherige Zuweisung der Adressräume nicht sehr effizient. Bereits Anfang der neunziger Jahre war klar, dass eine Lösung gefunden werden musste. Neben der Verbreiterung des Adressraumes waren Leistung, Vereinfachung von Administration und Routing sowie Sicherheitsaspekte die Schwerpunkte der Forschungsarbeiten.

Das Ergebnis dieser Arbeiten war die Spezifikation für das neue Internet-Protokoll alias IPng (Next Generation) alias IPv6 (so die offizielle Bezeichnung). IPv6 bringt eine Reihe von Fortschritten: Neben der gewaltigen Erweiterung des Adressraums von 32 bit-Adres-

sierung auf 128 bit-Adressierung finden sich eine Vielzahl anderer wichtiger Erweiterungen. Diese reichen von eingebauten Sicherheitsfunktionen über mehr Flexibilität bis hin zu Plug-and-Play-Funktionalitäten und Unterstützung von Echtzeitanwendungen. Alle führenden Hersteller von Netzwerk-Technologie haben dem neuen IPv6-Standard zugestimmt. Damit sind alle Zweifel ausgeräumt: IPv6 ist der Standard der Zukunft.

Bereits Anfang 1998 erreichte das weltweite IPv6-Testnetz, das 6Bone, eine Größe von 400 Rechnern in 40 Ländern. Es gibt über 50 verschiedene IPv6-Implementierungen, die entweder bereits fertig gestellt oder in der Entwicklungsphase sind. Davon befinden sich bereits ca. 25 verschiedene Implementierungen im 6Bone und in ersten Produktivnetzen im Einsatz.

3 Kommerzielle Grundlagen für IPv6

Angesichts des immensen Wachstums des Internet und der sich daraus ergebenden wirtschaftlichen Möglichkeiten zeigt sich die eminente Bedeutung von IPv6 für die kommerzielle Entwicklung, von der sowohl Netzwerkbetreiber als auch Nutzer profitieren werden.

3.1 IPv6-Standardisierungs- und Produktionsstatus

IPv6 ist als Draft Standard verabschiedet, d.h. die Definition ist stabil und somit für den produktiven Einsatz geeignet. Eine große Anzahl von Endanwendern, Standardisierungsgruppen und Netzwerkherstellern hat bei Spezifikation und Test von IPv6 zusammengearbeitet.

Gegenwärtig sind die folgenden Draft Standards verabschiedet:

RFC 2373	IP Version 6 Addressing Architecture
RFC 2374	An IPv6 Aggregatable Global Unicast Address Format
RFC 2460	Internet Protocol, Version 6 (IPv6) Specification
RFC 2461	Neighbor Discovery for IP Version 6 (IPv6)
RFC 2462	IPv6 Stateless Address Autoconfiguration
RFC 2463	Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification

Als Proposed Standard sind die folgenden RFCs verabschiedet:

RFC 1886	DNS Extensions to support IP Version 6
RFC 1887	An Architecture for IPv6 Unicast Address Allocation
RFC 1981	Path MTU Discovery for IP Version 6
RFC 2023	IP Version 6 over PPP
RFC 2080	RIPng for IPv6
RFC 2452	IP Version 6 Management Information Base for the Transmission Control Protocol

RFC 2454	IP Version 6 Management Information Base for the User Datagram Protocol
RFC 2464	Transmission of IPv6 Packets over Ethernet Networks
RFC 2465	Management Information Base for IP Version 6: Textual Conventions and General Group
RFC 2466	Management Information Base for IP Version 6: ICMPv6 Group
RFC 2467	Transmission of IPv6 Packets over FDDI Networks
RFC 2470	Transmission of IPv6 Packets over Token Ring Networks
RFC 2472	IP Version 6 over PPP
RFC 2473	Generic Packet Tunneling in IPv6 Specification
RFC 2507	IP-Header Compression
RFC 2526	Reserved IPv6 Subnet Anycast Addresses
RFC 2529	Transmission of IPv6 over IPv4 Domains without Explicit Tunnels
RFC 2545	Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing
RFC 2590	Transmission of IPv6 Packets over Frame Relay
RFC 2675	IPv6 Jumbograms
RFC 2710	Multicast Listener Discovery (MLD) for IPv6
RFC 2711	IPv6 Router Alert Option

Daneben gibt es noch eine Reihe von weiteren Standards, die durch IPv6-Modifikationen betroffen sind und für die deshalb eine Neufassung erforderlich war.

Beispiele hierfür sind:

RFC 1888	OSI NSAPs and IPv6
RFC 2292	Advanced Sockets API for IPv6
RFC 2375	IPv6 Multicast Address Assignments
RFC 2450	Proposed TLA and NLA Assignment Rules
RFC 2471	IPv6 Testing Address Allocation
RFC 2553	Basic Socket Interface Extensions for IPv6

3.2 IPv6-Design-Ziele

IPv6 bietet gegenüber IPv4 eine Reihe von Verbesserungen, wie z.B. einen erweiterten Adressraum und ein vereinfachtes Layout der Pakete.

Aufbau, Betrieb und Pflege heutiger Netzwerke sind nicht nur arbeitsintensiv, sondern erfordern auch hohen technischen Aufwand. Deshalb wurde der Entwicklung von Autokonfigurationsprotokollen für IPv6 große Aufmerksamkeit gewidmet. Autokonfigurationsprotokolle minimieren den Personalaufwand bei der Zuweisung von IP-Adressen und anderen Netzwerkparametern.

Als weiterer Vorteil ergibt sich daraus, dass mit der Einführung von IPv6 ein Neuanfang besonders in großen, historisch gewachsenen Netzen bei der Vergabe der Adressen möglich ist. So lässt sich beispielsweise die Router-Hierarchie effizienter strukturieren.

Die publicity-trächtigen Übergiffe von Hackern auf Netze selbst renommierter Software-Unternehmen zeigen den Bedarf an verbesserter Sicherheit in IPv4-basierten Netzen. IPv6 enthält verbesserte Authentifizierungs- und Verschlüsselungsmechanismen, die dem gestiegenen Sicherheitsbedürfnis im Internet Rechnung tragen.

Die folgenden Abschnitte geben einen Überblick über die Neuerungen, die IPv6 in Unternehmensnetzwerken und im globalen Internet bringt.

3.2.1 Adressierung und Routing

IPv6 löst etliche Probleme, die derzeit bei der Kommunikation innerhalb und zwischen Unternehmen existieren. So versetzt IPv6 die Designer des Internet Backbone in die Lage, eine flexible und erweiterbare globale Routing-Hierarchie zu definieren. Die Funktionsfähigkeit des Internet Backbones, in dem große Unternehmen und Internet Service-Provider (ISP) zusammen kommen, hängt entscheidend von der Verwaltbarkeit eines hierarchischen Adress-Systems ähnlich dem System der Telefonnumern ab. Zentrale Telefonvermittlungsrechner benötigen z.B. nur die Telefonvorwahl, um ein Ferngespräch gezielt zum Vermittlungsrechner des entsprechenden Ortsnetzes zu leiten.

Ohne eine Adress-Hierarchie sind Backbone-Router gezwungen, in ihren Routing-Tabellen Informationen über die Erreichbarkeit weltweit erreichbarer Netze zu speichern. Angesichts der Anzahl von IP-Subnetzen und des rapiden Wachstums des Internet ist es fast nicht mehr vorstellbar, diese Menge an Routing-Tabellen und deren Änderungen sinnvoll zu verwalten. Mit einer Adress-Hierarchie können Backbone-Router Daten anhand von Adress-Präfixen weiterleiten und benötigen dafür weniger Routing-Informationen.

Routing-Hierarchie in IPv4

In den vergangenen Jahren wurde bei IPv4 die neue Routing-Technik, Classless Inter Domain Routing (CIDR), eingeführt, um eine Routing-Hierarchie zu realisieren. CIDR erlaubt eine Route-Aggregation und unterschiedliche Level der Internet Adress-Hierarchie. Das versetzt Router mit einzelnen Routing-Tabelleneinträgen in die Lage, die Erreichbarkeit verschiedener Subnetze zu gewährleisten.

CIDR garantiert jedoch keine effiziente und skalierbare Routing-Hierarchie. Um die Verwaltung eines eigenen Eintrages für jede Route zu vermeiden, ist es notwendig, dass Router auf einem niedrigen Hierarchie-Level im Backbone-Bereich zu weniger spezifischen Routen auf einem höheren Level der Routing-Hierarchie zusammengefasst werden können. Die IPv4-Adressvergabe vor der Einführung von CIDR und die aktuelle Hierarchie der ISPs verhindern häufig die Zusammenfassung von Routen. Die mangelnde Systematik der Adressen im gegenwärtigen hierarchischen System und die Knappheit der IPv4-Adressen machen die Adressierung und das Routing im Internet unnötig komplex. Zudem ist eine Neuvergabe von IPv4-Adressen bei einem Wechsel des ISP kompliziert und somit teurer als in IPv6.

Problemlösung durch NAT

Viele der im heutigen Internet Backbone vorhandenen Probleme sind sowohl auf der Ebene der angeschlossenen Unternehmen als auch der einzelnen Anwender spürbar. Eine starke Belastung der globalen Routing-Tabellen entsteht beispielsweise durch Unternehmen, die ihre Routen nicht effizient zusammenfassen können. Erhalten Unternehmen nicht genügend global eindeutigen Adressraum, sind sie gezwungen, einen privaten, d.h. vom Internet isolierten Adressraum zu verwenden.

Da dieser private Adressraum global nicht eindeutige Adressen enthält, sind die Anwender normalerweise dazu gezwungen, Gateways und Network Address Translators (NAT) zu verwenden, um die Connectivity in die offene Welt herzustellen. Dadurch sind manche Services gar nicht bzw. nur eingeschränkt verfügbar. Ein NAT erlaubt es einem Unternehmen, eine beliebige interne Adress-Struktur einzuführen, ohne Rücksicht darauf nehmen zu müssen, wie sich die internen Adressen ins globale Internet einfügen. Dies erscheint in der IPv4-Welt mit ihren Adressbeschränkungen als vorteilhaft. Der NAT besetzt die Grenze zwischen dem Internet und dem Unternehmensnetz. Er wandelt die privaten internen Adressen in einen kleineren Bereich von global eindeutigen Adressen des Internet Backbones um bzw. umgekehrt.

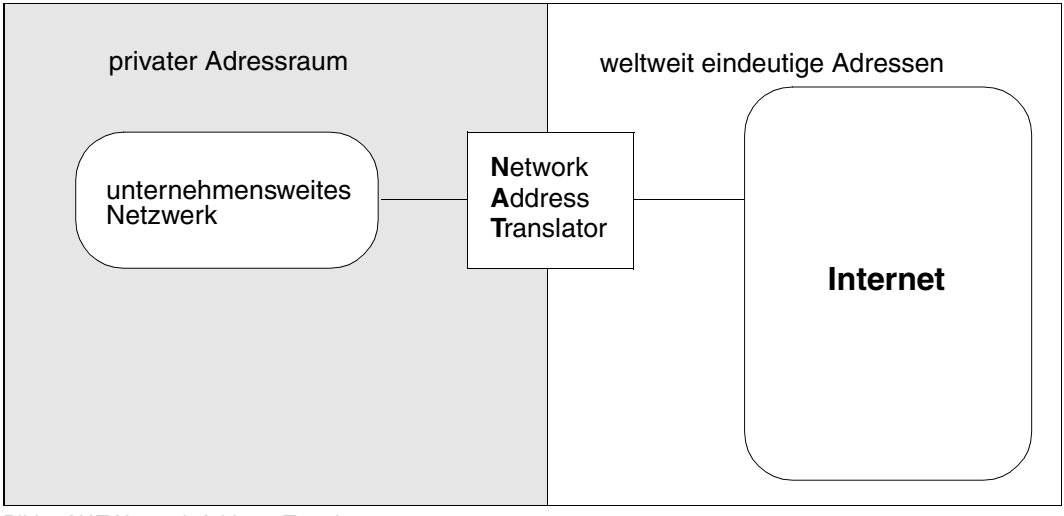

Bild 1: NAT Network Address Translator

Es sollten jedoch nur solche Unternehmen NAT einsetzen, die volle Connectivity zur Außenwelt benötigen, da eine performante und robuste Interaktion mit dem Internet durch einen NAT oft eher behindert wird. Die NAT-Technik der Adressersetzung in jedem Paket, das das Unternehmensnetz verlässt oder von ihm empfangen wird, ist sehr aufwändig. Nimmt die Zugriffshäufigkeit auf das Internet durch das Unternehmen zu, muss die Leistungsfähigkeit des NAT parallel dazu ansteigen. Der Flaschenhalseffekt verstärkt sich in dem Moment, da verschiedene NATs für den Zugang zum Unternehmensnetz synchronisiert werden müssen. Deshalb ist es für Unternehmen mit NAT wesentlich schwieriger, die heute übliche hochperformante Internet Connectivity zu erreichen, die durch mehrere ans Internet Backbone angeschlossene Router gewährleistet wird. Darüber hinaus verhindern NATs die Möglichkeit des symmetrischen Routing, da jeder NAT die Kontrolle über beide Richtungen des Transfers von Daten zwischen dem Unternehmensnetz und dem Internet haben muss.

Zusätzliche Probleme durch den Einsatz von NATs ergeben sich, wenn Anwendungen IP-Adressen in ihren Nutzdaten oberhalb der Netzwerkschicht versenden. Dies ist bei einer Vielzahl von Applikationen wie z.B. File Transfer Protocol (FTP) und mobile IP der Fall. Obwohl NATs jedes Paket bis in die Anwendungsschicht hinein auf eingebettete IP-Adressen analysieren, besteht die Möglichkeit eines Übersetzungsfehlers, der Anwendungsfehler nach sich zieht.

Auch die Verwendung von Sicherheitsmechanismen auf der Ebene des IP-Layers wird von NATs verhindert. Einerseits werden heute NATs als hilfreich für kleinere Netzwerke betrachtet, andererseits von vielen aber als unvorteilhaft für die weitere Entwicklung des Internet angesehen.

3.2.2 Minimierung des Administrationsaufwandes

Den zeit- und kostenträchtigsten Part in der Arbeit heutiger Netzwerkadministratoren beansprucht die Zuordnung von Netzwerkparametern an Computer und andere Netzwerkgeräte, bevor diese ihre Arbeit im Netz aufnehmen können. Informationen wie IP-Adresse, DNS-Server, Default-Router und andere Konfigurationsdetails müssen in jeder aktiven Netzwerkkomponente eingetragen werden. In vielen Fällen erfolgt das manuell, entweder durch die Netzwerkadministration oder noch aufwändiger durch den einzelnen Endanwender selbst. Verschiedene Anstrengungen diese administrativen Aufwände durch zentrale Server zu erledigen, führten zur Entwicklung des „Dynamic Host Configuration Protocols“ (DHCP). Aber die Nutzung von DHCP ergibt neue, daraus resultierende Probleme.

IPv4-spezifische Einschränkungen veranlassen viele Unternehmen, den Netzwerkgeräten immer wieder neue IP-Adressen zuzuordnen. Wechselt beispielsweise ein Unternehmen seinen ISP, muss es entweder alle eigenen Adressen umstellen, damit sie zu den Adress-Präfixen des neuen ISP passen, oder NATs verwenden. Eine Umnummerierung kann ebenfalls bei der Fusion zweier Unternehmen bzw. der Ausgliederung von Unternehmensteilen erforderlich sein.

Routing-Präfixe werden verwendet, um die Routing-Topologie des Unternehmensnetzwerkes und der angeschlossenen Computer zu repräsentieren. Es gibt zwei Möglichkeiten, die zu umständlichen bzw. falschen Routingpräfixen führen:

1. Die Routingpräfixe werden zu lang. Dadurch können die Netzwerkadministratoren keine weiteren Netzwerkgeräte mehr an spezielle Teilnetze anschließen.
2. Die Art und Weise, die einzelnen Netzwerke untereinander und an die Außenwelt anzuschließen, ändert sich.

Beide Fälle bedingen die Umnummerierung eines Teils oder des gesamten Unternehmensnetzwerks. Wünschenswert ist jedoch, dass die Umnummerierung ohne Ausfall des gesamten Netzwerks oder einzelner Netzwerkkomponenten durchgeführt werden kann.

Mangel an Adressen und Probleme der Routing-Hierarchie beeinträchtigen nicht nur den Betrieb großer Unternehmensnetze, sondern auch kleine Netzwerke. Sogar der Teleworker, der sich über das Internet in sein Büro einwählen will, kann davon betroffen sein.

Kleine Netzwerke können komplett aus den Routing-Tabellen des Internet Backbone fallen, falls sie nicht ihrer Adress-Struktur entsprechend angepasst werden. Da bei größeren Netzwerken eine Umnummerierung nicht mehr möglich ist, entstehen für den ISP große Probleme, den Zugang zum Internet zu gewährleisten. Mit der heutigen IPv4-Adressvergabe können ISPs mit einer Vielzahl von individuellen Einwahlbenutzern ihre IP-Adressen nicht so frei vergeben, wie sie es gerne tun würden. Als Konsequenz daraus ergibt sich, dass viele Einwahlbenutzer nur temporär IP-Adressen aus einem beschränkten Pool von IP-Adressen erhalten.

Eine eindeutige IP-Adresse bildet jedoch die Basis für die Anwender, die eine volle Connectivity zu anderen Nutzern des Internet benötigen. Sie vereinfacht auch viele interaktive Produktionsanwendungen. Zwei Beispiele hierzu sind Ferndiagnose-Anwendungen und das Telefonieren über das Internet. Die heutige Hierarchie der begrenzten und schlecht verteilten IPv4-Adressen verursacht bereits Probleme, und diese Probleme werden weiter zunehmen, wenn immer mehr Geräte mit unterschiedlichen Eigenschaften ans Internet angeschlossen werden.

3.2.3 Sicherheit

Verschlüsselung, Authentifizierung und die Gewährleistung der Datenintegrität sind bei einer kommerziellen Nutzung des Internet unbedingt erforderlich. IPv6 bietet entsprechende Zusatzheader, um diese Funktionen zu erbringen.

Authentifizierung

Der IPv6-Authentifizierungsheader erlaubt es dem Empfänger, festzustellen, ob ein IPv6-Paket wirklich von der angegebenen Sourceadresse stammt und auf dem Weg zwischen Absender und Empfänger nicht verändert wurde. Dadurch werden z.B. Hacker daran gehindert, die IP-Adresse ihres Rechners zu verändern, um ihre wahre Identität zu verschleiern. Das Maskieren der eigenen IP-Adresse (Spoofing) kann dazu missbraucht werden, vertrauliche Unternehmensdaten auszuspionieren oder die Kontrolle über Unternehmensserver zu erlangen. Durch Spoofing können Server veranlasst werden, den Zugang zu Passwörtern, wichtigen Dateien und Kontrolleinrichtungen zu ermöglichen. IP-Spoofing ist die bekannteste und am meisten genutzte Art von „Denial of Service“-Angriffen.

Mit IPv4 ist es für einen Server normalerweise unmöglich festzustellen, ob die Pakete, die er empfängt, auch wirklich von einem legitimierten Absender versendet wurden. Einige Unternehmen haben auf diese Bedrohung mit der Einrichtung von Firewalls reagiert. Aber Firewalls verursachen nicht nur Performance-Engpässe, sie stehen außerdem für eine restriktive Netzwerkpolitik, eingeschränkten Zugang zum Internet oder auch eine eingeschränkte Kommunikation zwischen verschiedenen Bereichen bzw. Standorten innerhalb eines Unternehmens.

IPv6 nutzt eine Standardmethode, um die Authentizität der vom Network Layer empfangenen Pakete zu überprüfen. Damit ist gewährleistet, dass Produkte unterschiedlicher Hersteller bei der Authentifizierung zusammen arbeiten können. IPv6-Implementierungen verwenden standardmäßig den MD5 bzw. den SHA-1 Algorithmus für die Authentifizierung, um damit zu garantieren, dass zwei beliebige IPv6-Endsysteme sicher miteinander kommunizieren können. Da die Definition des Headers unabhängig vom verwendeten Authentifizierungsalgorithmus ist, können jedoch auch andere Authentifizierungsalgorithmen verwendet werden.

Encryption

Ein weiteres großes Sicherheitsrisiko im Internet neben dem Spoofing stellt der Missbrauch von Netzwerk-Sniffern und anderen normalerweise nützlichen Diagnosehilfsmitteln dar.

In IPv6 ist die Vertraulichkeit der Daten durch einen zusätzlichen Erweiterungsheader für die End-to-End Verschlüsselung auf Netzwerkebene gewährleistet. Der IPv6-Verschlüsselungsheader enthält nur Indikatoren über die verwendeten Schlüssel. Somit lässt sich die Schlüsselinformation nicht direkt durch eine Analyse der verschlüsselten Pakete ausspionieren.

Für IPv4 wurden inzwischen analoge Sicherheitserweiterungen wie in IPv6 definiert, diese haben jedoch noch keine ausreichend weite Verbreitung bzw. Unterstützung in den IPv4-Protokollmaschinen gefunden.

Beide IPv6-Sicherheitsheader können sowohl direkt bei der Kommunikation zwischen zwei Endsystemen verwendet werden als auch zwischen speziellen Sicherheitsgateways, die den IPv6-Paketen einen zusätzlichen Grad der Sicherheit geben.

3.2.4 Mobilität

IPv4 hat aus verschiedenen Gründen Schwierigkeiten bei der Unterstützung von mobilen Computern wie z.B. Notebooks:

– Ein mobiles Endsystem benötigt eine Forwarding-Adresse, wenn es an einer anderen Stelle im Internet angeschlossen wird. Es ist jedoch nicht immer leicht, eine Forwarding-Adresse im IPv4-Adressraum zu erhalten.

– Die Verteilung der Information über den neuen Standort eines mobilen Endgerätes im Routing-System des Internet erfordert eine zuverlässige Authentifizierung. Diese ist im IPv4-Umfeld jedoch nicht weit verbreitet.

– Mit IPv4 ist es für das mobile Endsystem schwierig festzustellen, ob es am selben Netzwerk angeschlossen ist wie beim letzten Mal oder nicht.

– Mit IPv4 ist es fast unmöglich, alle Kommunikationspartner über den neuen Standort des mobilen Endsystemes zu informieren.

Die Verbesserungen für Mobile Computing sind in einer Reihe von Aspekten beim Protokolldesign für IPv6 berücksichtigt und gehen über eine einfache Verbesserung beim Einwählen ins Internet weit hinaus. Die Verbesserungen bei der Verarbeitung von Zieloptionen, Autokonfiguration, Routing-Headern, Einschalung von Paketen, Sicherheitsfunktionen und Anycast-Adressen dienen einer besseren Integration von Mobile Computing in ein IPv6 basiertes Internet. Die Vorteile von IPv6 Mobile Computing können durch die Kombination mit Flow Labels noch weiter hervorgehoben werden, da damit entsprechende „Quality of Service“-Optionen auch mobilen Endsystemen zur Verfügung gestellt werden können.

3.3 Die IPv6-Lösung

IPv6 bietet gegenüber IPv4 eine Vielzahl entscheidender Vorteile:

- mehrstufige globale und hierarchische Routing-Architektur
- Adress-Autokonfiguration
- vereinfachtes IPv6-Headerformat
- Multicast
- Anycast

3.3.1 Mehrstufige globale und hierarchische Routing-Architektur

IPv6 definiert mit seinem deutlich vergrößerten Adressraum eine mehrstufige, globale, hierarchische Routing-Architektur. Durch die Verwendung von an CIDR angelehnten Adress-Präfixen können IPv6-Adressen so vergeben werden, dass einzelne Routen in den Routern leichter zusammengefasst werden können. Dadurch kann das Wachstum der Routing-Tabellen in den Backbone-Routern eingeschränkt und besser kontrolliert werden. Durch den deutlich vergrößerten Adressraum wird die Verwendung von privaten Adressen überflüssig. ISPs haben genügend Adressen zur Verfügung, um auch kleineren Unternehmen und einzelnen Anwendern, die sich per Telefon ins Internet einwählen, eine global eindeutige IPv6-Adresse zu geben.

3.3.2 Adress-Autokonfiguration

Jedes IPv6-Endsystem erzeugt am Anfang durch „Stateless Autoconfiguration“ für sich eine lokale IPv6-Adresse. Dazu sind keine manuell konfigurierten lokalen Daten bzw. externen Server erforderlich.

Stateless Autoconfiguration

Stateless Autoconfiguration ermöglicht es dem IPv6-Endsystem, im Zusammenspiel mit einem lokalen IPv6-Router für sich eine global eindeutige IPv6-Adresse zu bestimmen. Normalerweise kombiniert das Endsystem seine 48 bit oder 64 bit lange MAC-Adresse (Layer 2-Adresse), die durch den Hardware-Hersteller festgelegt ist, mit einem Netzwerkpräfix, der ihm von einem lokalen Router mitgeteilt wird. Dies reduziert die Kosten des Endanwenders, da keine Netzwerkadministratoren benötigt werden, die jedes Endsystem vor seiner Verwendung entsprechend konfigurieren. Diese Kosten sind bei IPv4 fester Bestandteil der Installationskosten für jedes anzuschließende Gerät. Durch den extrem reduzierten Administrationsaufwand und den Einsatz billiger Netzwerkkomponenten ergeben sich neue Marktmöglichkeiten für die Nutzung von „Embedded Systems“. Dies erweist sich als vorteilhaft, wenn sich häusliche Netzwerke zu einem wichtigen Marktsegment entwickeln.

DHCPv6

In IPv4-Netzwerken wird oft das Dynamic Host Configuration Protocol (DHCP) genutzt, um den Aufwand für die Zuordnung von IP-Adressen und anderen Netzwerkparametern zu reduzieren. DHCP wird als Mittel zur „Stateful“ Address Configuration bezeichnet, da es feste Tabellen verwendet, um IP-Adressen an neu angeschlossene Netzwerkkomponenten zu vergeben. Eine neue Version von DHCP für IPv6 (DHCPv6) wurde entwickelt, um eine ähnliche Art der Adressvergabe zu ermöglichen, da dies von vielen Netzwerkadministratoren gewünscht wird.

DHCPv6 kann neben der ursprünglichen Adressvergabe auch bei der Rekonfiguration eines Netzwerkes sinnvoll eingesetzt werden, da es durch die Verwendung von Multicast-Adressen eine beliebig definierbare Gruppe von Clients ansprechen kann.

Die Möglichkeiten der Autokonfiguration sind für den Anwender auf verschiedenste Art und Weise nützlich. Wenn zum Beispiel ein Unternehmen durch einen Wechsel des ISPs zu einer Umnummerierung gezwungen ist, so erlaubt die IPv6-Autokonfiguration die Zuordnung neuer Adressen an die Endsysteme, ohne dass ein manueller Eingriff zur Rekonfiguration der Workstations und DHCP Clients notwendig ist.

Die Autokonfiguration unterstützt Unternehmen auch bei der Administration sich dynamisch ändernder Endanwender-Konfigurationen. Mobile Computer erhalten immer eine gültige Forwarding-Adresse, unabhängig davon, wo sie ans Netzwerk angeschlossen sind.

3.3.3 IPv6-Headerformat

In IPv6 wurde das Layout des Basisheaders eines IP-Paketes vereinfacht. Einige Optionen des IPv4-Headers wurden gestrichen, andere in eigene Zusatzheader ausgelagert. Durch die einfachere Headerstruktur wird der Mehrbedarf an Netzwerkbandbreite größtenteils wieder ausgeglichen. Die 16 byte langen IPv6-Adressen sind vier mal so lang wie die 4 byte langen IPv4-Adressen, aber durch das Redesign des IP-Headers ist der gesamte IPv6-Header nur doppelt so lang wie der IPv4-Header. Durch das Redesign des IP-Headers sind auch viele Aspekte der Verarbeitung besser lösbar als in IPv4.

Zwar wurden am Anfang der Überlegungen zu IPv6 auch andere Lösungsmöglichkeiten mit variabler Adresslänge betrachtet, schließlich wurde jedoch einer möglichst einfachen Lösung der Vorzug gegeben, zumal 128 bit längerfristig einen ausreichend großen Adressraum bieten. Zusätzliche Arbeiten zur IP-Headerkompression versprechen zudem, den Zusatzaufwand durch den längeren Header bei langsamen Anschlüssen zu reduzieren bzw. ganz zu eliminieren.

IPv6 kodiert IP-Headeroptionen in einer Art und Weise, die das Weiterleiten der Pakete gegenüber IPv4 beschleunigt. Die optionale IPv6-Headerinformation ist in unabhängigen Erweiterungs- bzw. Zusatzheadern verpackt. Diese sind nach dem IPv6-Basisheader und vor den Headern der Transportschicht in jedem IPv6-Paket eingefügt.

Die meisten der IPv6-Erweiterungsheader brauchen, im Gegensatz zu den IPv4-Optionen, von Routern nicht betrachtet und verarbeitetet zu werden. Dies bedeutet eine große Verbesserung in der Nutzbarkeit von IPv6-Optionen gegenüber IPv4, wo die Nutzung von IP-Optionen in jedem Router einen Performance-Verlust bei der Weiterleitung des Paketes verursachte.

Die IPv6-Zusatzheader haben eine variable Länge und können mehr Informationen als die IPv4-Optionen enthalten. Es besteht auch die Möglichkeit, einfach neue Zusatzheader einzuführen.

In Kapitel „Technische Grundlagen für IPv6“ (siehe Seite 33) wird detaillierter auf die Unterschiede zwischen IPv4- und IPv6-Headern eingegangen.

An dieser Stelle sei nur kurz erwähnt, dass Erweiterungsheader sowohl für die explizite Routingsteuerung als auch für die Unterstützung von Mobile Computing, Authentifizierung, Verschlüsselung und Fragmentierung definiert worden sind.

3.3.4 Multicast

In modernen Netzwerken besteht die Notwendigkeit, Ströme von Video- und Audiodaten, animierte Grafiken, Nachrichten und andere zeitabhängige Daten zu Gruppen von funktionell zusammenhängenden, netzmäßig jedoch verstreuten Endstationen zu versenden. Die beste Art dies zu erreichen ist die Verwendung von Multicast auf Netzwerkebene. Dabei sendet ein Server einen Datenstrom mit Multimedia- bzw. zeitkritischen Daten. Dieser Datenstrom wird dann von verschiedenen Empfängern, die sich dafür angemeldet haben, gelesen. Der Sender und die einzelnen Empfänger bilden hierbei eine Multicast-Gruppe.

In einem multicastfähigen Netzwerk werden alle Pakete des Servers effizient an die Empfänger der Multicast-Gruppe gesendet. Eine Verdoppelung der Pakete ist hierbei nur notwendig, wenn sich die Empfänger in unterschiedlichen Teilnetzen befinden. Wie in Bild 2 auf der nächsten Seite dargestellt, wird ein einzelnes Paket des Senders von allen Empfängern der Multicast-Gruppe empfangen. Wenn sich die Empfänger der Multicast-Gruppe in verschiedenen Netzwerken befinden, wird ein Verteilungsbaum erzeugt.

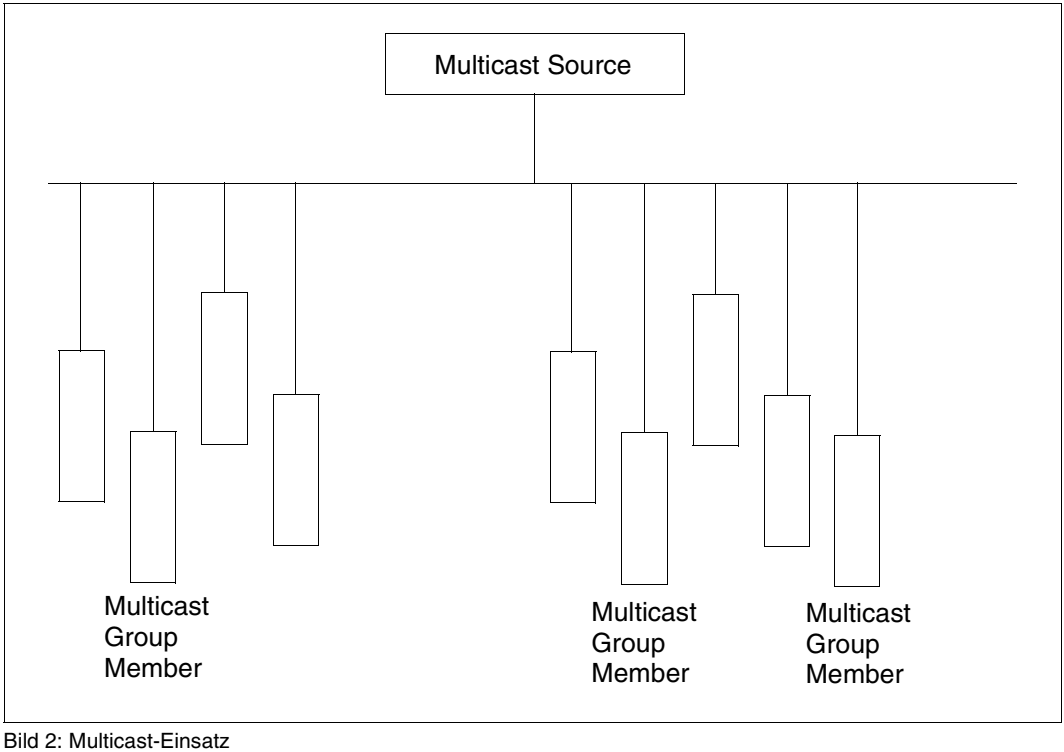

Bild 2: Multicast-Einsatz

Router benutzen Multicast-Protokolle wie Distance Vector Multicast Routing Protocol (DVMRP), Protocol Independent Multicast (PIM) oder Multicast Open Shortest Path First (MOSPF), um dynamisch einen Verteilungsbaum zu erzeugen, der alle Empfänger einer Multicast-Gruppe mit dem Multicast-Sender verbindet. Um Mitglied einer Multicast-Gruppe zu werden, muss der neue Empfänger eine „join“-Nachricht an seinen nächsten Router senden. Daraufhin wird der Verteilungsbaum der Multicast-Gruppe entsprechend erweitert.

Dadurch ist es möglich, dass der Server jedes Paket nur einmal senden muss. Das Paket wird bei Bedarf durch die Router dupliziert und durch das Netzwerk an alle Empfänger der Multicast-Gruppe weitergeleitet. Dies spart im Vergleich zu Unicast und Broadcast sowohl Betriebsmittel bei den Routern als auch Netzwerkbandbreite. Damit ist Multicasting sowohl dem Broadcasting als auch einer entsprechenden Abbildung auf Unicast überlegen.

Multicast-Anwendungen wurden erstmalig für IPv4 entwickelt, aber IPv6 erweitert die Anwendungsmöglichkeiten für Multicasting durch die Definition eines viel größeren Multicast-Adressraumes. Alle IPv6-Endsysteme und Router müssen die Nutzung von Multicast-Adressen ermöglichen. Als Ablösung für das in IPv6 nicht mehr unterstützte Broadcasting gibt es in IPv6 die Möglichkeit, Multicast-Adressen mit unterschiedlichen Gültigkeitsbereichen zu definieren.

3.3.5 Anycast

Die Anycast-Adressierung von IPv6 ist ein neues Feature, das in IPv4 nicht existiert. Vom Konzept her ist es eine Mischung zwischen Unicast und Multicast. Eine Gruppe von Endsystemen bildet eine Anycast-Gruppe. Ein Paket, das an die Anycast-Adresse der Gruppe gesendet wird, wird nur einem Endsystem der Gruppe zugestellt. Dies ist der Unterschied zu Multicasting, bei dem ein Paket an alle Empfänger der Gruppe zugestellt wird. Die Endsysteme in einer Anycast-Gruppe sind speziell konfiguriert, um Anycast-Adressen, die eine Teilmenge der Unicast-Adressen sind, zu erkennen.

Anycast ist ein neuer Dienst und die entsprechenden Anwendungen sind noch nicht vollständig entwickelt. Es gibt jedoch verschiedenste Szenarien, in denen Anycasting nützlich ist.

Wenn ein Unternehmen Anycast-Adressen verwendet, um die Router des ISP Backbones anzusprechen und diese alle dieselbe Anycast-Adresse verwenden, dann hat das Unternehmen verschiedene redundante Zugänge zum Internet. Sollte nun ein Router ausfallen, übernimmt automatisch ein anderer Router der Anycast-Gruppe dessen Funktion.

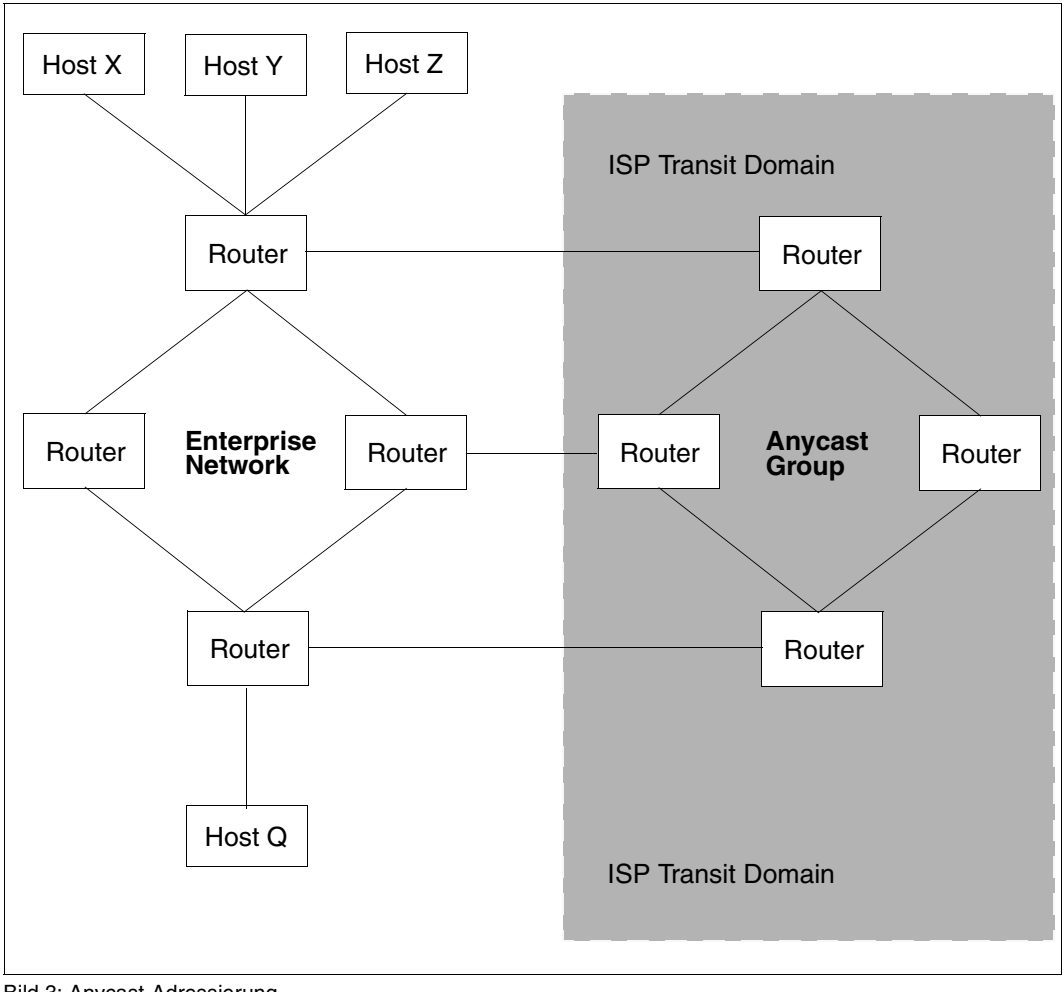

Bild 3: Anycast-Adressierung

In Bild 3 senden die Endsysteme Q, X, Y und Z in einem Unternehmensnetzwerk ihre Pakete an eine Anycast-Adresse, die die Backbone-Router in der ISP-Transitiondomain adressieren. Die Boarder-Router des Unternehmens leiten die Pakete wie Unicast Pakete weiter. Dann werden die Pakete von einem einzigen der Backbone-Router empfangen und weiter geleitet.

Falls für einen mobilen Computer die Home-Agenten ebenfalls über eine Anycast-Adresse adressiert werden, kann auch hier ohne zusätzliche Konfigurationsmaßnahmen ein Redundanzkonzept realisiert werden. In diesem Fall arbeitet jeder mobile Computer nur mit genau einem Home-Agenten zusammen, ohne jedoch wissen zu müssen mit welchem.

Anycast wurde definiert, um Endsystemen einen effizienten, ausfallsicheren Zugriff auf bekannte Dienste, gespiegelte Datenbanken, Web-Server usw. zu ermöglichen. Es bietet einen kostengünstigen Weg, um die Robustheit von verteilten Anwendungen zu erhöhen und eine lastabhängige Verteilung durchzuführen. So kann beispielsweise allen DNS-Servern eines Unternehmens dieselbe Anycast-Adresse zugeordnet werden.

3.3.6 Quality of Service

Der IPv4-Header enthält ein „differentiated services"-Byte, dem im IPv6-Header das „traffic class"-Byte entspricht. Beide Byte sind für eine einfache Unterstützung unterschiedlicher Services gedacht. Sowohl IPv4 als auch IPv6 unterstützen jedoch auch das RSVP Protocol zur Nutzung komplexerer Quality of Service-Definitionen. Zusätzlich enthält der IPv6-Header ein 20 bit langes „flow information" Feld, das für die Verwendung durch zusätzliche Quality of Service-Funktionen zur Verfügung steht.

Anwendungen, die Quality of Service-Funktionen nutzen, sind noch in der Planungsphase, aber IPv6 legt eine Basis fest, durch die Quality of Service-Funktionen wie z.B. Bandbreiten-Reservierung und maximale Laufzeit eines Paketes in einer offenen Netzwerkumgebung verfügbar gemacht werden können.

Ein weiterer Vorteil von IPv6 für Quality of Service-Funktionen ist, dass das Flow Label im IPv6-Basisheader abgelegt ist und damit zur Wegeoptimierung unterschiedlicher Datenströme durch Router verwendet werden kann. Außerdem kann dadurch das Flow Label auch bei der Nutzung von Verschlüsselung verwendet werden.

3.3.7 Der Übergang nach IPv6

Der Übergang von IPv4 nach IPv6 kann auf verschiedene Art und Weise erfolgen. Während sich Einige für einen möglichst schnellen Übergang zu IPv6 aussprechen, wollen Andere diesen Übergang erst dann durchführen, wenn der IPv4-Adressraum endgültig erschöpft ist oder technische Gründe einen Übergang erfordern. Unabhängig davon, welcher Meinung man den Vorzug gibt, steht fest, dass derzeit Millionen von IPv4-Rechnern und Router im Internet installiert sind und dass IPv4 und IPv6 für eine längere Zeit nebeneinander existieren müssen.

Deshalb wurde beim Design von IPv6 großer Wert darauf gelegt, dass für Router und Endsysteme ein gleitender Übergang von IPv4 nach IPv6 möglich ist. Durch diesen gleitenden Übergang wird zum Einen die Entstehung von IPv6-Inseln verhindert, zum Anderen ist es nicht notwendig, einen radikalen Wechsel von IPv4 nach IPv6 durchzuführen. Es wurden verschiedenste Umstellmechanismen realisiert, die es den Netzwerkadministratoren erlauben, selbst zu entscheiden, wann und wie sie ihre Endsysteme und Router umstellen. IPv6 kann wahlweise zuerst in den Endsystemen oder in den Routern eingeführt werden. Es ist jedoch auch möglich, IPv6 zuerst in einem Teilbereich mit einer beschränkten Anzahl von

Endsystemen und Routern einzuführen. Dabei muss dieser Teilbereich weder unbedingt aus einem zusammenhängenden Subnetz bestehen noch muss er sich sich auf einen Ort beschränken.

Viele auf IPv6 hochgerüstete Endsysteme und Router werden auf längere Zeit eine Abwärtskompatibilität zu IPv4-Geräten benötigen. Sie werden deshalb auch auf längere Zeit zusätzlich eine IPv4-Adresse besitzen.

Um den Anwendern einen einfachen Übergang von IPv4 nach IPv6 zu ermöglichen, wurden in einer speziellen IETF-Arbeitsgruppe (NGTRANS) entsprechende Übergangsmechanismen definiert. Beispiele hierfür sind Dual Stack-Endsysteme und Router sowie das Tunneln von IPv6-Paketen über IPv4. Ein Dual Stack-Endsystem ist ein Computer, der sowohl über IPv4 als auch IPv6 Pakete empfangen und versenden kann. Dadurch kann eine darauf ablaufende Anwendung Pakete aus beiden Adressfamilien senden und empfangen, was einen einfachen Übergang von IPv4 nach IPv6 auf Anwendungsebene erlaubt.

3.3.8 DNS für IPv6

Vor dem Einsatz von IPv6-Endsystemen bzw. Dual Stack-Endsystemen müssen die Netzwerkadministratoren Umstellungen bzw. Erweiterungen in ihren Domain Name Service (DNS) durchführen. Um dies zu ermöglichen, wurde ein neuer DNS Resource Record Typ (AAAA) definiert. Während des Testbetriebes von IPv6 im 6Bone wurden jedoch Schwächen bei der Nutzung von AAAA DNS Resource Record erkannt und deshalb ein neuer A6 DNS Resource Record Typ zur Abbildung von DNS-Namen auf IPv6-Adressen definiert. Auch die umgekehrte Abbildung von IPv6-Adressen auf DNS-Namen wurde definiert.

Wenn ein IPv6-fähiger DNS-Server vorhanden ist, kann ein Dual Stack-Endsystem zweigleisig mit IPv6-Endsystemen kommunizieren. Erfragt ein Dual Stack-Endsystem beim DNS-Server eine Partner-IP-Adresse und erhält eine 32 bit lange IPv4-Adresse, kommuniziert er mit dem Partner-Endsystem auf IPv4-Basis. Erhält er auf seine Anfrage eine 128 bit lange IPv6-Adresse, so kommuniziert er mit dem Partner-Endsystem auf IPv6-Basis. Wenn kein IPv6-fähiger DNS-Server vorhanden ist, können die Endsysteme die Abbildung von Namen auf IPv6-Adressen durch lokale Tabellen zur Namensabbildung durchführen.

IPv6-Autokonfiguration und IPv6-DNS können durch dynamische DNS-Updates zusammengeführt werden. Dadurch kann der DNS-Server sicher und automatisch seine Resource Records modifizieren, wann immer ein neues IPv6-Endsystem eine neue IPv6-Adresse belegt. Ebenso kann damit die automatische Umnummerierung von IPv6-Endsystemen unterstützt werden.

3.3.9 Änderungen von Anwendungen für IPv6

Anwendungen, die TCP/IP-Netzwerkfunktionen nicht direkt nutzen (also keine SOCKETS-Funktionen aufrufen), brauchen nicht geändert zu werden, um in einer Dual Stack-Umgebung ablaufen zu können. Speziell in BS2000/OSD bedeutet dies, dass alle DCAM-NEA-, DCAM-ISO- und CMX-Anwendungen unverändert weiterlaufen. Dies gilt auch, wenn auf Netzwerkebene das IPv6-Protokoll verwendet wird. SOCKETS-Anwendungen, die IPv6 nutzen wollen, müssen hingegen geändert werden. IPv6 ist am SOCKETS-Interface durch eine neue Adressfamilie realisiert, die in diesem Fall genutzt werden muss. Jedoch können IPv6-fähige Anwendungen auch mit IPv4-Partnern kommunizieren, so dass nicht zwei Ausprägungen der Anwendung, jeweils eine für IPv4 und eine für IPv6, gleichzeitig ablaufen müssen.

3.3.10 Routing in IPv6-/IPv4-Netzwerken

Router, die sowohl IPv6 als auch IPv4 unterstützen, können ähnlich administriert werden wie die heutigen reinen IPv4-Router. Protokoll-Erweiterungen für BGP4 wurden durch die IETF definiert, um IPv6 zu unterstützen. Diese Erweiterungen werden seit 1997 im 6Bone für das IPv6-Routing genutzt. Die BGP Erweiterungen wurden von allen wichtigen Routerherstellern implementiert und sind in einem RFC definiert. IPv6-Versionen für andere gebräuchliche Routerprotokolle wie Open Shortest Path First (OSPF) und Routing Information Protocol (RIP) sind ebenfalls definiert und implementiert.

Netzwerkadministratoren können die logische Struktur des IPv6-Netzwerkes völlig getrennt von der logischen Struktur ihres IPv4-Netzwerkes halten, obwohl beide dasselbe physikalische Netzwerk nutzen. Damit können beide Netzwerke unabhängig voneinander verwaltet werden. Alternativ dazu können sie die logische Struktur des IPv6-Netzwerkes an die des existierenden IPv4-Netzwerkes mit den entsprechenden Domaingrenzen und Subnetzstrukturen anpassen. Beide Ansätze haben ihre Vor- und Nachteile.

Eine eigenständige IPv6-Netzwerkstruktur kann dazu benutzt werden, die heute in vielen Unternehmen historisch gewachsenen, aber ineffizienten IPv4-Netzwerkstrukturen zu beseitigen und einen neuen, strukturierten, hierarchischen Netzwerkplan zu realisieren, der einen optimalen Anschluss zu einem oder mehreren ISPs gewährleistet. Umnummerierung, Zusammenfassung von Routen und andere Ziele einer hierarchischen Netzwerktopologie werden damit ebenfalls vereinfacht.

Anfangs werden viele IPv6-Endsysteme nur über reine IPv4-Router miteinander kommunizieren können. Es ist also die Situation von einzelnen IPv6-Inseln in einer IPv4-Welt gegeben. Deshalb wurden Mechanismen geschaffen, die es IPv6-Endsystemen erlauben, über dazwischen liegende IPv4-Netzwerke miteinander zu kommunizieren.

Die wichtigste Technik ist das Tunneln von IPv6-Paketen über IPv4-Netzwerke. Dabei werden IPv6-Pakete in IPv4-Pakete verpackt. Das Tunneling erlaubt es den einzelnen IPv6-Endsystemen, die existierende IPv4-Infrastruktur zu nutzen, ohne dass irgend eine der vorhandenen IPv4-Komponenten des Netzwerkes geändert werden muss.

Ein Dual Stack-Router am Rand der IPv6-Insel fügt einfach einen IPv4-Header vor das IPv6-Paket und sendet es als normales IPv4-Paket weiter. Die IPv4-Router auf dem Weg leiten es weiter, ohne es als IPv6-Paket zu erkennen. Am anderen Ende des Tunnels packt ein anderer Dual Stack-Router das Paket wieder aus und sendet es als IPv6-Paket an seine endgültige Zieladresse. Das Verpacken bzw. Entpacken von IPv6-Paketen in IPv4-Paketen kann ebenso auch durch die Endsysteme erfolgen.

Es gibt zwei verschiedene Tunnelarten:

– Automatische Tunnel
– Konfigurierte Tunnel

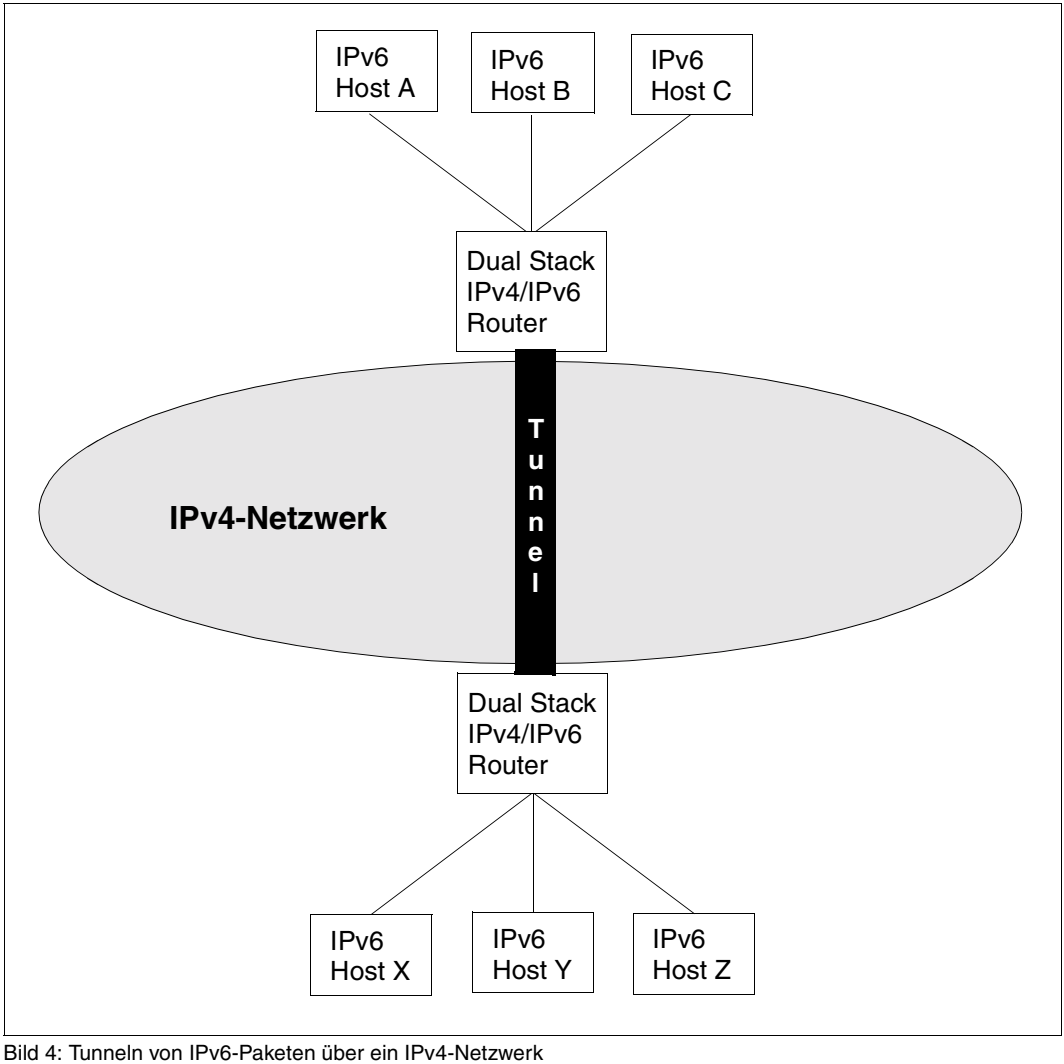

Bild 4: Tunneln von IPv6-Paketen über ein IPv4-Netzwerk

3.3.10.1 Automatische Tunnel

Automatische Tunnel nutzen IPv4-kompatible IPv6-Adressen. Eine IPv4-kompatible IPv6-Adresse besteht aus einer IPv4-Adresse, die durch führende Nullen zu einer 128 bit langen IPv6-Adresse erweitert wurde.

Sofern Pakete mit einer IPv4-kompatiblen IPv6-Adresse versendet werden, kann das Endsystem am einen Ende des Tunnels aus der IPv6-Adresse eine IPv4-Adresse bilden und die IPv6-Pakete in IPv4-Pakete verpacken. Am anderen Ende des Tunnels wird der IPv4-Header wieder entfernt.

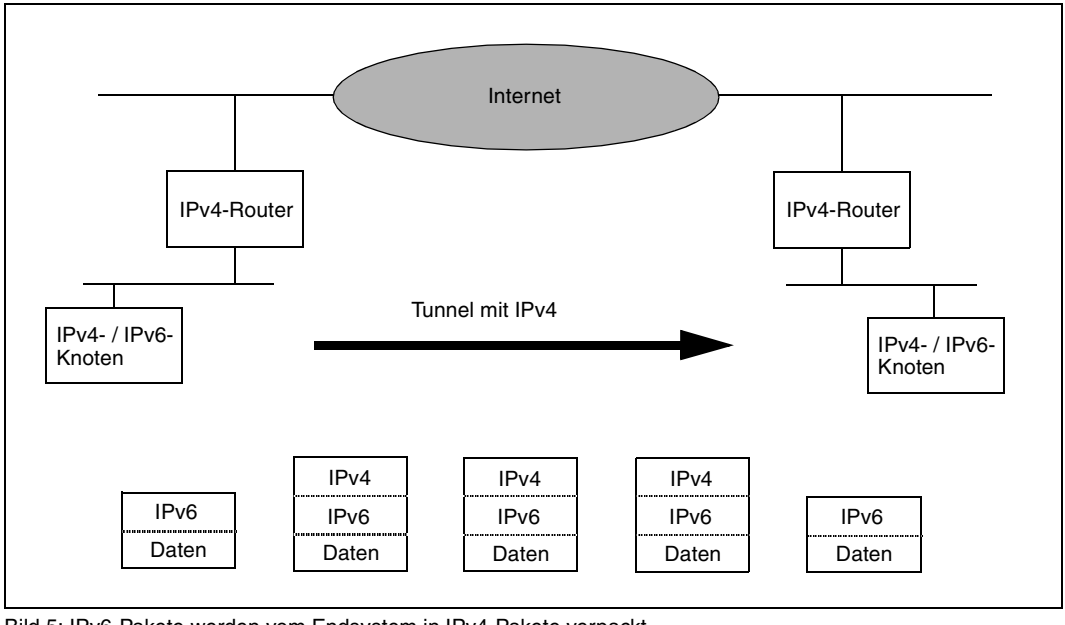

Bild 5: IPv6-Pakete werden vom Endsystem in IPv4-Pakete verpackt

Automatisches Tunneling erlaubt es IPv6-Endsystemen, existierende IPv4-Netzwerke auf einfache Art und Weise zu nutzen. Aber es verlangt die Nutzung von IPv4-kompatiblen IPv6-Adressen, wodurch der Vorteil des erweiterten IPv6-Adressraumes nicht genutzt werden kann.

IPv6-Endsysteme, die IPv4-kompatible IPv6-Adressen nutzen, können zwar die Vorteile des erweiterten IPv6-Adressraumes nicht nutzen, aber sie sind in der Lage, die anderen Erweiterungen von IPv6 wie Flow Label Authentifizierung, Verschlüsselung sowie Multicast- und Anycast-Adressierung zu verwenden.

Sobald ein Endsystem mit IPv4-kompatiblen IPv6-Adressen nach IPv6 migriert ist, ist der Weg für eine einfache Nutzung des vollen IPv6-Adressraumes leichter zu gehen. Die Nutzung von IPv4-kompatiblen IPv6-Adressen bedeutet, dass Netzwerkadministratoren IPv6-Endsysteme unter Beibehaltung der bisherigen Adress- und Netzwerkstruktur in ihr Netzwerk aufnehmen können. Automatische Tunnel sind verfügbar, wenn sie benötigt werden. Sie sind aber nicht mehr notwendig, wenn die Backbone-Router auf IPv6 hochgerüstet worden sind. Die Umstellung von IPv4-kompatiblen IPv6-Adressen auf normale IPv6-Adressen kann einfach und schnell durchgeführt werden, falls die Backbone-Router IPv6 voll unterstützen.

3.3.10.2 Konfigurierte Tunnel

Für den Aufbau konfigurierter Tunnel muss der Netzwerkadministrator eine Abbildung zwischen IPv6-Adressen und IPv4-Adressen für die Tunnel-Endpunkte definieren. Außerhalb des konfigurierten Tunnels erfolgt die Weiterleitung der IPv6-Pakete anhand der IPv6-Adresse. Am Anfang des konfigurierten Tunnels werden die IPv6-Pakete in IPv4-Pakete verpackt und an die konfigurierte IPv4-Zieladresse gesendet.

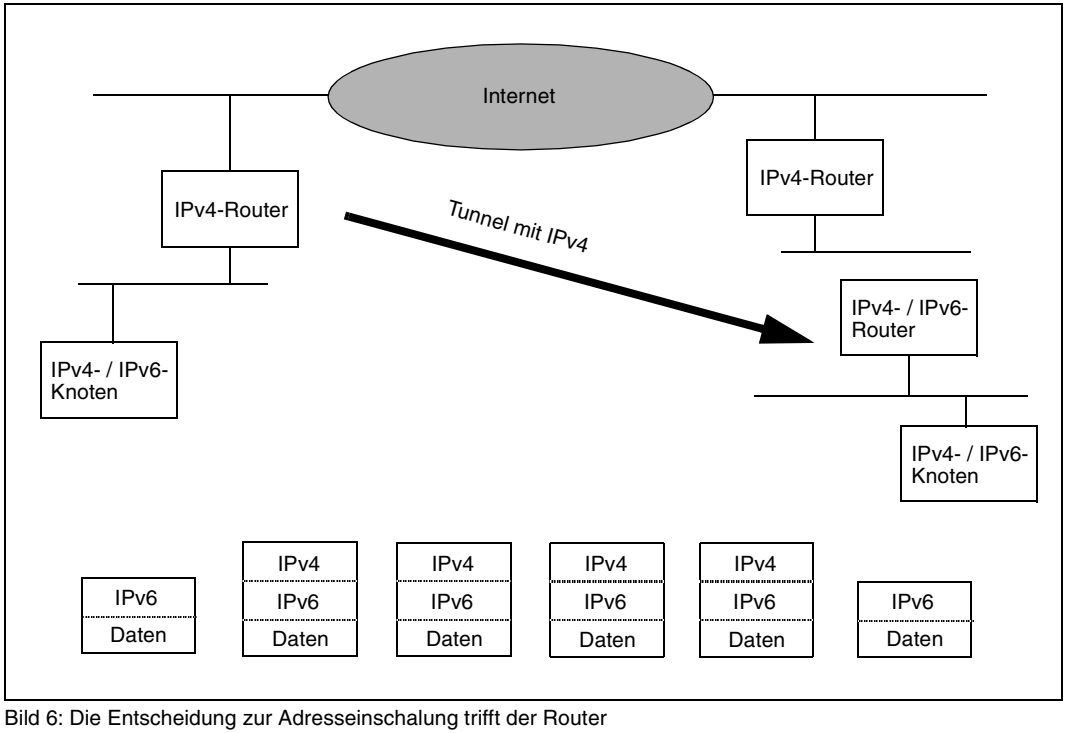

Bild 6: Die Entscheidung zur Adresseinschalung trifft der Router

Am Endpunkt des konfigurierten Tunnels werden die IPv6-Pakete dann wieder ausgepackt und regulär anhand ihrer IPv6-Adresse weitergeleitet. Dies erfordert zwar einigen Verwaltungsaufwand an den Endpunkten des konfigurierten Tunnels, aber die verpackten IPv6-Pakete werden dynamisch durch das IPv4-Netzwerk weitergeleitet, ohne dass die IPv4-Router etwas von IPv6 bemerken.

Bei konfigurierten Tunneln besteht im Gegensatz zu automatischen Tunneln kein direkter Zusammenhang zwischen den verwendeten IPv4- und IPv6-Adressen.

3.3.11 Die Dual Stack-Übergangsmethode

Die ersten Nutzer von IPv6-Endsystemen setzen voraus, dass sie auch weiterhin mit bereits existierenden IPv4-Endsystem kommunizieren können. Dies wird durch die Dual Stack-Übergangsmethode von IPv4/IPv6 erreicht. Viele Endsysteme und Router unterstützen bereits in der heutigen heterogenen Kommunikationswelt verschiedene Netzwerkprotokolle. Die meisten der heute installierten Router sind Multiprotokollrouter, auch viele Endsysteme unterstützen verschiedene Netzwerkprotokolle. Im BS2000/OSD sind dies bisher die NEA-, ISO- und IPv4-Protokolle. Die Unterstützung eines zusätzlichen Netzwerkprotokolls ist deshalb ein wohlbekanntes Problem. Wenn auf einem Endsystem ein dualer IPv4-/IPv6-Stack läuft, dann hat dieses Endsystem sowohl Zugang zu IPv4-Partnern als auch zu IPv6-Partnern. Ein IPv4-/IPv6-Dual Stack-Router kann dann sowohl IPv4-Pakete als auch IPv6-Pakete weiterleiten.

Dual Stack-Maschinen können sowohl völlig unabhängige IPv4- und IPv6-Adressen benutzen als auch IPv4-kompatible IPv6-Adressen.

3.4 Diskussion zu IPv6

Die große Zukunft und das breite technische Spektrum von IPv6 spiegeln sich in der kontroversen Diskussion um IPv6 wider, die in der Internet-Gemeinde geführt wird. Netzwerkbetreiber, die eine zukunftsorientierte Netzwerkstrategie schaffen wollen, könnte diese Diskussion davon abhalten, bereits heute die notwendigen Maßnahmen zur Umstellung auf IPv6 zu ergreifen.

In den folgenden Abschnitten soll versucht werden, diese Diskussion etwas zu entwirren und dem Leser einige Argumente pro IPv6 an die Hand zu geben. Andernfalls besteht die Gefahr, dass das Internet sich lediglich zu einer zusammengeschusterten IPv4-Version weiterentwickelt.

Nur der Mangel an Adressraum ist die treibende Kraft hinter IPv6

Viele der Diskussionen über ein neues Internet-Protokoll gründen auf der Tatsache, dass früher oder später der 32 bit große IPv4-Adressraum keine global eindeutigen Network Layer-Adressen mehr zulässt. Die verschiedenen Adress-Registraturen, die riesigen Netzwerk Service-Providern und Netzwerk-Operatoren Blöcke mit IP-Adressen zuweisen, sind mit der Art, wie diese Adressen vergeben werden, sehr vorsichtig geworden, weil die meisten Vorhersagen bezüglich der Erschöpfung der IPv4-Adressen einen Zeitrahmen aufzeigen, der bereits in diesem Jahrzehnt liegt.

In Hinblick auf Langfristigkeit wurde IPv6 mit einem 128 bit-Adressraum ausgestattet, welcher global eindeutige Adressen für jede denkbare Möglichkeit des Netzwerkausbaus innerhalb der absehbaren Zukunft (d. h. Jahrzehnte) bietet. IPv6 hat 16-byte-Adressen, oder 340.282.366.920.938.463.463.374.607.431.768.211.456 Adressen (tatsächlich mehr als ein Drittel einer Duodecillion). Die Zahl der Adressen verdient Anerkennung, aber sie ist nur ein wichtiges Merkmal des IPv6-Designs. Andere Eigenschaften wurden als direkte Antwort auf gegenwärtige Geschäftsanforderungen entwickelt, wie z.B. festlegbare Netzwerk-Architektur, verbindliche Sicherheit und Datenintegrität, ausgedehnte Servicequalität (QoS), Autokonfiguration und eine effiziente Netzwerk-Routen-Aggregation auf globalem Backbone Level. Diese Features wurden in IPv6 derart effizient umgesetzt, dass es schwierig, wenn nicht unmöglich wäre, sie in vergleichbarer Güte in IPv4 zu realisieren.

IPv4-Erweiterungen können die Funktionalität von IPv6 erreichen

Es gab verschiedenste Anstrengungen, die Lebensdauer von IPv4 durch eine evolutionäre Weiterentwicklung des Protokoll-Standards und verschiedenste proprietäre Techniken zu verlängern. Ein solches Beispiel ist die Entwicklung des Network Address Translators (NAT), welcher IPv4-Adressraum spart, indem er die Datenpakete auffängt und private, unternehmensinterne Adressen in eine oder wenige global eindeutige Internet-Adressen um-

wandelt. Andere Beispiele schließen die verschiedenen QoS und Sicherheitserweiterungen von IPv4 ein, welche generell minderwertiger oder bestenfalls identisch mit in IPv6 spezifizierten Mechanismen sind.

Es ist nicht bekannt, wie lange die Lebensdauer von IPv4 mit Hilfe dieser Techniken ausgedehnt werden kann. Sicher ist jedoch, dass die weitverbreitete Einführung von NAT-Geräten negative Auswirkungen auf die End-to-End Nutzung bekannter Internet-Anwendungen hat. In der Praxis kann nur eine begrenzte Zahl wohlbekannter Anwendungen korrekt mit NAT-Geräten oder mit ihnen verbundenen Application Level Gateways behandelt werden. Mit NAT-Geräten ist zudem die Nutzung von IPv4-End-to-End-Sicherheitsmechanismen eingeschränkt.

Darüber hinaus trägt die Entwicklung einer neuen und innovativen Internet-Applikation die Bürde der durch NATs entstandenen Konstrukte. Da NATs für IPv6 vollkommen überflüssig sind, kann eine Standard End-to-End-IPv6-Sicherheit entwickelt werden. Der NAT-Mechanismus ist ebenso dafür bekannt, dass große Schwierigkeiten bei der Konstruktion von Virtual Private Networks (VPNs) entstehen, weil er die Adressraumverwaltung schwierig macht und Standard-Sicherheitsmechanismen stört.

Ein NAT arbeitet also nur in einem „flachen Universum“ für einen dem globalen Internet zugänglichen Standort. Sogar kleinere Unternehmen haben jedoch intern keine flache Adress-Struktur, sondern vielfach eine interne Hierarchie mit vielfältigen Beziehungen. Realistische NAT-Anwendungslösungen müssten das Routing mit mehreren Eingangs-/Ausgangs-NATs zur Lastverteilung, Multi-NAT-Hop Routen usw. einschließen. All das würde die IPv4-Architektur nur in kleinen Netzwerken schaffen, aber auch hier nur stückchenweise und schlecht.

Es ist schwierig, die Kosten der Umstellung auf IPv6 mit denen zu vergleichen, die das Festhalten an IPv4 und seiner Weiterentwicklung brächten. Jeder Netzwerkmanager wird diesen Vergleich anstellen; aber bei IPv4 zu bleiben, ist keine langfristig befriedigende Lösung.

Die IPv6-Nutzung durch eine große Vielzahl von Netzwerk-Komponenten betrifft weder den Endanwender noch den Geschäftsbetrieb

Innerhalb der nächsten Jahre werden konventionelle Computer über das Internet mit einer Vielzahl neuer Geräte, einschließlich Palmtop Personal Data Assistants (PDA), Hybridmobiltelephone Technologie mit Data Processing-Fähigkeit, Smart Set-Top Boxes mit integriertem Web-Browser und eingebetteten Netzwerk-Komponenten in Geräten vom Kopierer im Büro bis zu Küchenmaschinen verbunden. Einige der neuen Geräte, die IP-Adressen und eine Verbindung fordern, werden konsumorientiert sein, aber viele werden integrierter Bestandteil des Information Management von Unternehmen und Institutionen jeder Größe sein. Diese neuen Geräte verlangen Eigenschaften, die für die meisten Protokollentwickler während der Entstehung des IPv4-Internet noch gar nicht absehbar waren.

Der 128 bit-Adressraum des IPv6 erlaubt Firmen, eine riesige Anzahl neuer Desktops, Mobiles und eingebetteter Netzwerkgeräte kosteneffektiv und verwaltbar einzusetzen. Des Weiteren werden die sebstkonfigurierenden Eigenschaften von IPv6 es einer riesigen Anzahl von Geräten möglich machen, sich dynamisch mit dem Netzwerk zu verbinden, ohne immense Kosten für die Verwaltung einer ständig wachsenden Zahl von Zugängen, Bewegungen und Wechsel zu verursachen.

Die Geschäftsanforderungen an IPv6 werden durch die Endnutzer-Anwendungen angetrieben. Anwendungen für mobile Knoten, E-Commerce usw. sind mit IPv6 leichter zu entwickeln und einzusetzen. Dies gilt vor allem im Vergleich zu IPv4 in Verbindung mit NAT.

IPv6 ist in erster Linie für Backbone-Router relevant, nicht für Enduser-Anwendungen

Es stimmt, dass IPv6 Address Aggregation effiziente mehrstufige Routing-Hierarchien erlaubt, die das unkontrollierte Wachstum von Backbone-Router Tables verhindern. Aber viele der fortgeschrittenen IPv6-Eigenschaften bringen direkte Vorteile bei Enduser-Anwendungen auf der Ebene von Arbeitsgruppen und Abteilungen. Für Anwendungen werden beispielsweise die IPv6-Entschlüsselung und der Authentifikationsservice als ein integraler Bestandteil des IP-Stack verfügbar sein. Mobilen Business Usern und sich schnell ändernden Organisationen erlaubt die IPv6-Autokonfiguration, die effiziente Nutzung von IP-Adressen ohne die Verzögerungen und Kosten einer manuellen Adressverwaltung. Gleiches gilt auch für traditionelles DHCP, welches in vielen gegenwärtigen IP-Netzwerken verwendet wird. IPv6 ist ebenso für Endbenutzer wie für Business von Belang. Dies wird noch wichtiger, wenn QoS-Funktionen und QoS-Routing ein wesentlicher struktureller Bestandteil des Internet sein werden.

Asynchroner Transfer Modus (ATM) Cell Switching macht die Notwendigkeit des IPv6 überflüssig

ATM und andere Switching Methoden bieten eine interessante Technologie für gegenwärtige und zukünftige Netzwerke, aber ATM ist selbst kein Ersatz für die Packet Routing Internet-Architektur. ATM sollte besser als eine Link Layer-Technologie über ein Zugriffsmedium verstanden werden. ATM bietet individuelle Nutzung und verspricht, geprüften Quality of Service (QoS) für Anwendungen bereitzustellen, die diesen beanspruchen. Jedoch sind diese hypothetischen Vorteile noch nicht vollständig für ATM entwickelt, und ebenso ist es möglich, dass diese Vorteile in Zukunft auch für IPv6-Netzwerke verfügbar sind, die nicht über ATM laufen.

Glücklicherweise müssen Netzwerkbetreiber nicht zwischen ATM und IPv6 wählen, weil die beiden Protokolle auch weiterhin unterschiedlich definierte Aufgaben im Unternehmensnetzwerk erfüllen. Für viele Netzwerk-Designer ist ATM ein nützliches Übertragungsmedium für Highspeed IPv6 Backbone-Netzwerke. Standards und Entwicklungsarbeit beschäf-

tigen sich mit der Integration von ATM- und IPv6-Umgebungen. Wie IPv4 bietet auch IPv6 Network Layer-Dienste über alle wichtigen Link Types an, einschließlich ATM, Ethernet, Token Ring, ISDN, Frame Relay und T1.

IPv6 ist etwas, womit sich nur große Telefongesellschaften und die Regierung beschäftigen sollten

Einige Internet-Spezialisten haben IPv6 als etwas charakterisiert, dass außerhalb des Unternehmensnetzwerks und eines gegenwärtig überschaubaren Zeitrahmens steht. Tatsächlich aber ist IPv6 ein Standardweg und eine effiziente Lösung für die Arbeit und die gesteigerte Effektivität der täglichen Geschäftsaktivitäten. Aber der einzige Weg, IPv6 zum Erfolg zu führen, besteht darin, dass Geschäftsleben und Institutionen jeder Art sich über die Unzulänglichkeiten von IPv4 einig werden und Pläne für den Übergang machen. In den letzten Jahren haben Internet-Protokolle einen neuen Typ des Distributed Commerce geschaffen, der die Menschen innerhalb von Unternehmen zusammenbringt und ihnen einen weltweiten Zugriff erlaubt. In der Tat ist das anhaltende und beeindruckende Wachstum des Internet, welches die gegenwärtigen Entwicklungsanstrengungen für IPv6 begründet, weitgehend der Nutzung des World Wide Web durch Geschäftsleben und Endbenutzer zu verdanken. Diesen Endbenutzern Dienste anzubieten, ist für wesentlich mehr Institutionen von Interesse als nur für Regierungen und Telefongesellschaften.

IPv6 erfordert ausgedehnte Modifikationen bei existierenden Betriebssystemen, Anwendungen und Programmtechniken

IPv6 erfordert sicherlich gewisse Modifikationen an den Network Protocol Handling-Modulen, die auf den relevanten Computern installiert sind. Jedoch erfordert dies typischerweise nur geringfügige oder gar keine Änderungen an Betriebssystemen. In BS2000/OSD erfolgte die Unterstützung von IPv6 ohne eine Änderung außerhalb des Transportsystems BCAM. Einfache und natürliche Modifikationen, die sich normalerweise auf wenige Programmzeilen beschränken, erlauben den Anwendungen, IPv6-Adressen direkt zu benutzen. Weil IPv6 einen Teil seines Adressraums für die Kompatibilität mit IPv4-Adressen reserviert, können die für den Umgang mit IPv6-Adressen modifizierten Applikationen weiterhin mit bestehenden IPv4-Clients und -Servern kommunizieren. Darüber hinaus sollten die Übergangsstrategien, die für IPv6 im IPv4-basierten Internet festgelegt wurden, die schrittweise Übernahme von IPv6 zu einem gleitenden Prozess machen, der es existierenden Anwendungen erlaubt, schrittweise und kontrolliert für den Übergang zu IPv6 weiter entwickelt zu werden.

IPv6 - zu klein, zu bald

IPv6 erscheint als eine Erweiterung von IPv4, und es gibt Stimmen, die fragen, wieso man sich all den Ärger machen sollte, nur um ein neues Network Layer-Protokoll zu verwenden, wo man doch besser ein richtig zukunftsweisendes Protokoll mit jeder Menge neuer Eigenschaften entwickeln sollte. Diese Argumentation ignoriert folgende einfachen Tatsachen:

– Der Sinn eines Network Layer-Protokolls ist es, Netzwerke miteinander zu verbinden,

– IPv6 baut auf dem außergewöhnlichen Erfolg von IPv4 auf, indem es die erfolgreichen Teile übernimmt und bekannte Fehler ausmerzt. Das erscheint wesentlich sinnvoller, als wieder bei Null zu beginnen.

Wer argumentiert, es sei für IPv6 noch zu früh, der ignoriert, dass existierende Lösungen zur Verlängerung der Lebensdauer von IPv4 Lückenbüßer sind, und dass stattdessen IPv6 in Betrieb genommen werden kann.

Das Problem der Umnummerierung ist in IPv6 gelöst

Obwohl in IPv6 erhebliche Anstrengungen unternommen wurden, eine einfachere Umnummerierung zu ermöglichen, ist das Problem der Umnummerierung noch nicht komplett gelöst. IPv6-Designer überarbeiten noch immer das Design für die Umnummerierung von Routern. Des Weiteren sind Anwendungen, die von IPv4 nach IPv6 übernommen wurden, nicht automatisch besser geeignet, die Umnummerierung zu unterstützen. Einige Anwendungen erfordern kleine Design-Anpassungen zur Unterstützung der Umnummerierung. Zu guter Letzt ist das größte Hindernis bei der Umnummerierung die Verwaltungspraxis, die die Schlüsselinformationen direkt auf IP-Adressen anwendet, statt passende Indexmethoden zu verwenden. Diese Verwaltungspraxis erfordert viel Aufmerksamkeit und die Entwicklung von moderneren Richtlinien für die Internet-Verwaltung, bevor das Problem der Umnummerierung als gelöst betrachtet werden kann.

Routing ist in IPv6 festgelegt

IPv6 bietet verschiedene Verbesserungen für das Routing. Es erlaubt eine vorteilhaftere und zutreffende Zuweisung von IPv6-Adressen als existierende Zuweisung von IPv4-Adressen. Auch das Weiterleiten von Paketen erfolgt flüssiger als über IPv4-Router, besonders wenn IP-Optionen genutzt werden. Der größere Adressraum von IPv6 bietet die Gelegenheit zu einer optimalen Netzwerkplanung, weil sich der Zwang, Netzwerkverbindungen im Voraus zu planen, erheblich entschäft hat. Weiterhin ist es leichter, die Bewilligung der Sicherheitsmaßnahmen zur Authentifizierung einzusetzen und private Daten privat zu belassen, da jeder IPv6-Router Security-Processing als sicher annehmen kann.

Trotzdem gibt es noch viele operationale Probleme, die Aufmerksamkeit verdienen. Die IPv6-Routing-Protokolle sind an die eng verwandten IPv4-Routing-Protokolle angepasst, und das bedingt einige vergleichbare Probleme. Weiterhin werden Ergänzungen vorgenommen, um die Stabilität, Konvergenz-Zeit und Konfigurierbarkeit der Routing-Protokolle zu verbessern.

Eines der schwierigsten Probleme ist es, Routing-Protokolle bedienerfreundlicher zu gestalten, um zu verhindern, dass für den zuverlässigen Routing-Betrieb Genies nötig sind. Außerdem gibt es noch ungelöste Probleme in Bezug auf Multihoming. All diese Probleme werden die für die Weiterentwicklung von IPv6 Verantwortlichen auch weiterhin beschäftigen.

4 Technische Grundlagen für IPv6

Dieses Kapitel stellt die technischen Grundlagen von IPv6 vor. In vielen Fällen illustrieren die technischen Details die Konzepte des vorherigen Kapitels. Es werden jedoch auch weitere, tiefergehende Eigenschaften des Protokolls vorgestellt, um die Technik von IPv6 besser verstehen zu können.

4.1 Der IPv6-Header im Vergleich zum IPv4-Header

Am Beginn der technischen Betrachtung steht ein Vergleich zwischen dem IPv6- und dem IPv4-Header. Beide Header enthalten eine Versionsnummer und die Absende- bzw. Zieladresse. Wie in Bild 8 ersichtlich, ist der IPv6-Header viel einfacher aufgebaut als der IPv4-Header. Dies vereinfacht die Bearbeitung des Headers durch Router und Endsysteme. Während IPv4-Header eine variable Länge haben, hat der IPv6-Header eine feste Länge von 40 byte. Eine weitere Steigerung der Verarbeitungseffizienz wurde durch die Reduzierung der Felder im IPv6-Header erreicht. Ein IPv4-Header enthält, wie Bild 7 zeigt, mindestens 12 Felder. Zusätzlich kann er optionale Felder enthalten, die im Bild aber nicht detailliert aufgeführt sind. Der IPv6-Header besteht dagegen grundsätzlich aus nur 8 Feldern (siehe Bild 8 auf der nächsten Seite).

<table>
<tr><td>Version
(4 bit)</td><td>IHL
(4 bit)</td><td colspan="2">Type of Service
(8 bit)</td><td colspan="2">Total Length
(16 bit)</td></tr>
<tr><td colspan="3">Identification
(16 bit)</td><td>Flags
(4 bit)</td><td>Fragment Offset
(12 bit)</td></tr>
<tr><td colspan="2">Time to live
(8 bit)</td><td colspan="2">Protocol
(8 bit)</td><td colspan="2">Header Checksum
(16 bit)</td></tr>
<tr><td colspan="6">Source Address
(32 bit)</td></tr>
<tr><td colspan="6">Destination Address
(32 bit)</td></tr>
<tr><td colspan="6">IP Options
(0 - n bit)</td></tr>
</table>

Bild 7: IPv4-Header

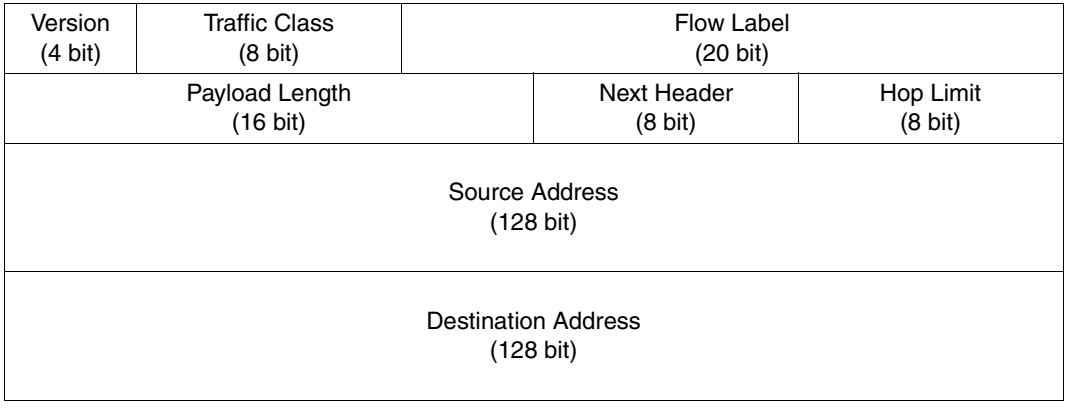

Bild 8: IPv6-Header

Das erste Feld, das im IPv6-Header entfällt, ist das Headerlängenfeld (Header Length), das wegen der festen Headerlänge nicht mehr benötigt wird. Das Gesamtlängenfeld (Total Length) des IPv4-Headers wurde durch das IPv6-Feld „Länge der Nutzdaten“ (Payload Length) ersetzt. Bei der Länge der Nutzdaten werden nur die Nettodaten ohne den immer 40 byte langen IPv6-Header gezählt. Durch das Format des Feldes „Länge der Nutzdaten“ definiert können damit IPv6-Pakete bis zu einer maximalen Nutzdatenmenge von 64 KB übertragen werden. Es können jedoch auch größere IPv6-Pakete, sogenannte „Jumbogramme“, zwischen zwei IPv6-Endsystemen ausgetauscht werden. Dazu wird das Feld „Länge der Nutzdaten“ auf Null gesetzt und die Länge des Jumbogramms in einem speziellen Erweiterungsheader angegeben. Dieser Erweiterungsheader wird im Folgenden noch vorgestellt.

Das IPv4-Feld „Time to Live“ wurde in IPv6 in „Hop Limit“ umbenannt, was seine aktuelle Verwendung besser beschreibt. Das Feld wird zum Aufbrechen von Routingschleifen verwendet. Das „Hop Limit“ wird vom Sender des IPv6-Paketes gesetzt und von jedem Router, der das Paket weiterleitet, um eins reduziert. Sobald der Wert des „Hop Limit“ Null erreicht, wird das IPv6-Paket verworfen. Der maximal mögliche Wert von 255 Hops übersteigt nach Ansicht der IPv6-Protokollentwickler die Anforderungen auch der größten vorstellbaren Netze.

Zusätzlich zum Headerlängenfeld entfallen noch weitere Felder des IPv4-Headers. Die Felder „Fragment Offset“, „Identification“ und „Flags“ wurden gestrichen, da sie in optionale Erweiterungsheader ausgelagert wurden (vgl. Abschnitt "Fragmentierungsheader" auf Seite 38).

Das IPv4-Feld „Header Checksum“ entfiel, da normalerweise andere Layer die Fehlererkennung des Protokoll-Stacks durchführen. Fehlerhafte Pakete werden entweder auf der Ebene des Link Layers oder auf der Ebene des Transport Layers erkannt. Die Überprüfung bzw. Berechnung der Header Checksum führt in IPv4 bei Routern und Endsystemen zu einer deutlichen Performance-Verschlechterung.

Schließlich wird das IPv4-Feld „Type of Service“ in IPv6 durch „Traffic Class“ und „Flow Label“ ersetzt.

4.2 Erweiterungsheader

Der IPv4-Header enthält ein Feld „Option“, mit dem Informationen über Source Routing, Sicherheit und weitere optionale Parameter übertragen werden. Diese Optionen werden jedoch kaum genutzt, da Pakete, die solche Optionen enthalten, von Routern mit einer deutlich schlechteren Performance weitergeleitet werden.

Das Feld „Option“ wird in IPv6 durch so genannte Erweiterungsheader ersetzt, die sich in den IPv6-Paketen hinter dem primären IPv6-Header und vor den Headern des Transport Layers befinden. IPv6-Erweiterungsheader ermöglichen die Nutzung von Sicherheitsfunktionen, Fragmentierung, Source Routing und anderen IPv6-Zusatzfunktionen. Es gibt für die Anzahl der Erweiterungsheader keine protokollbedingte Obergrenze. Durch die Auslagerung der Optionen in eigene Header ist deren Bearbeitung einfacher geworden. Bild 9 zeigt ein Beispiel, in dem ein Fragmentierungsheader und ein Verschlüsselungsheader hinter dem primären IPv6-Header und vor dem Transport-Header eingefügt worden sind.

IPv6 Hdr	Fragmentation Hdr	Encryption Hdr	Transport Hdr, etc.

Bild 9: IPv6-Erweiterungsheader

Das Protokollfeld des IPv4-Headers - normalerweise Transmission Control Protocol (TCP) oder User Datagram Protocol (UDP) - wurde in IPv6 durch das „Next Header“ Feld ersetzt. Dieses zeigt den Typ des nächsten Headers an. Dies kann ein TCP- bzw. UDP-Header oder ein IPv6-Erweiterungsheader sein.

IETF Arbeitsgruppen haben bereits eine Anzahl von IPv6-Erweiterungsheadern sowie eine Reihenfolge dieser Header, falls sie in einem Paket vorhanden sind, definiert.

Die vorgeschlagene Reihenfolge der Erweiterungsheader ist wie folgt:

1. primärer IPv6-Header
2. Hop-by-Hop-Optionsheader
3. Zieloptionsheader 1
4. Routing-Header
5. Fragmentierungsheader
6. Authentifizierungsheader
7. Verschlüsselungsheader
8. Zieloptionsheader 2

Daran schließen sich die Header des Transport Layers und die Nutzdaten an.

Jeder Erweiterungsheader, mit Ausnahme des Zieloptionsheaders, kommt in einem IPv6-Paket normalerweise nur einmal vor.

4.2.1 Hop-by-Hop-Optionsheader

Der Hop-by-Hop-Optionsheader enthält, sofern vorhanden, Informationen für alle Router, die das Paket an die Zieladresse weiterleiten. Er muss der erste Header nach dem primären IPv6-Header sein. Da dieser Header von allen Routern auf seinem Weg durch das Netzwerk gelesen wird, können damit zum Beispiel Diagnose-Informationen an die Router verteilt werden.

Eine bereits definierte Anwendung für den Hop-by-Hop-Optionsheader ist die Router Alarm Option, die die Router dazu veranlasst, das IPv6-Paket vor einer Weiterleitung selbst komplett zu bearbeiten. Ein Beispiel für ein solches Paket ist eine RSVP Resource Reservation Meldung für „Quality of Service“-Dienste.

4.2.2 Zieloptionsheader

Es existieren zwei Ausprägungen des Zieloptionsheaders, jeweils mit einer unterschiedlichen Position innerhalb des IPv6-Paketes. Ein Zieloptionsheader, der sich vor dem Routing Header im IPv6-Paket befindet, wird von jedem Router, der das Paket weiterleitet, verarbeitet. Ein Zieloptionsheader, der sich nach dem Routing-Header im IPv6-Paket befindet, wird nur vom adressierten Zielsystem verarbeitet. Derzeit sind noch keine konkreten Anwendungen für die Nutzung des Zieloptionsheaders realisiert.

4.2.3 Routing-Header

Bisher ist eine Art des IPv6-Routing-Headers, der sogenannte „Type 0“ definiert. Dieser Routing-Header gibt dem sendenden Endsystem die Möglichkeit, den Weg des IPv6-Paketes zum Zielsystem zu steuern. Der IPv6-Routing-Header ersetzt die Loose Source Route Option von IPv4. Dieser optionale Header erlaubt es dem Sender, eine Liste von IPv6-Adressen zu spezifizieren, welche das IPv6-Paket auf seinem Weg zum Zielsystem passieren muss.

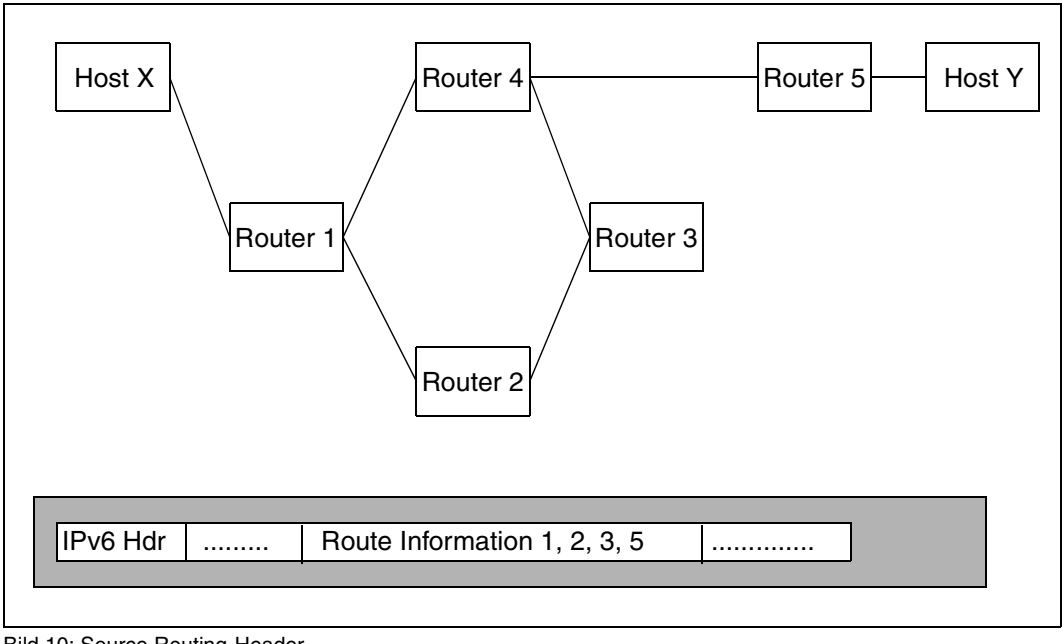

Bild 10: Source Routing-Header

Ein Beispiel für „IPv6 loose source routing“ wird im Bild 10 gezeigt. Im „loose“-Routingmodus können auch nicht im IPv6-Routing Header aufgeführte Router das IPv6-Paket weiterleiten. So kann im Beispiel von Bild 10 das Paket von Router 3 nach Router 4 und dann zum Router 5 weitergeleitet werden, obwohl Router 4 nicht im Routing-Informationsfeld des IPv6-Routing Headers angegeben ist.

Zur Steuerung welcher Router als nächstes vom IPv6-Paket zu adressieren ist, gibt es im IPv6-Routing-Header einen Zähler, der die Anzahl noch nicht besuchter IPv6-Router anzeigt. Jeder in der Routerliste spezifizierte Router reduziert diesen Zähler um 1.

4.2.4 Fragmentierungsheader

Beim IPv4-Netzwerkprotokoll konnte jedes IPv4-Paket an einem beliebigen Punkt im Netz, den Möglichkeiten der Weiterleitung entsprechend, in Teilpakete zerlegt oder fragmentiert werden. Im IPv6-Netzwerkprotokoll gibt es diese Möglichkeit nicht mehr. Es gibt nur noch eine End-to-End Fragmentierung/Reassembly zwischen Sende-Endsystem und Empfänger-Endsystem. Router führen also keine Fragmentierung von zu großen IPv6-Paketen mehr durch.

Die Abschaffung der Fragmentierung durch Router erlaubt ein einfacheres Layout des IPv6-Basisheaders und damit eine bessere Performance der Router beim Weiterleiten der IPv6-Pakete. Besonders, da die heutigen Netzwerke auf Link Layer-Ebene eine Fragmentgröße unterstützen, die für die meisten IPv6-Pakete ausreichend ist. Falls doch eine Fragmentierung erforderlich ist, gibt es einen optionalen IPv6-Erweiterungsheader, der es dem sendenden Endsystem erlaubt, das Ursprungspaket in mehrere kleinere IPv6-Pakete zu zerteilen. Das empfangende Endsystem setzt dann diese Teilpakete wieder zum Ursprungspaket zusammen (Reassembly), sodass dies für die höheren Protokollschichten bzw. die Anwendungen transparent ist.

Der IPv6-Fragmentierungsheader enthält ein Feld, das eine Gruppe von IPv6-Teilpaketen (Fragmenten) identifiziert, sowie Sequenznummern, die die Reihenfolge der Teilpakete festlegen.

Das sendende Endsystem ist für die korrekte Größe der einzelnen IPv6-Pakete verantwortlich. Deshalb muss es die Mindestgröße der Maximum Transmission Unit (MTU) des Weges zum Zielsystem bestimmen. Dies ist die kleinste MTU eines Links auf dem Weg zwischen dem sendenden Endsystem und dem empfangenden Endsystem. Wenn zum Beispiel (siehe Bild 11) zwei FDDI-Netzwerke mit einer MTU von 4500 byte durch ein Ethernet mit einer MTU von 1500 byte verbunden sind, darf das sendende Endsystem nur IPv6-Pakete bis zu einer Maximalgröße von 1500 byte versenden. Wenn also ein 2500 byte großes Paket versendet werden soll, dann wird dieses in zwei Fragmente, eines mit 1500 byte und eines mit 1000 byte Länge, zerlegt.

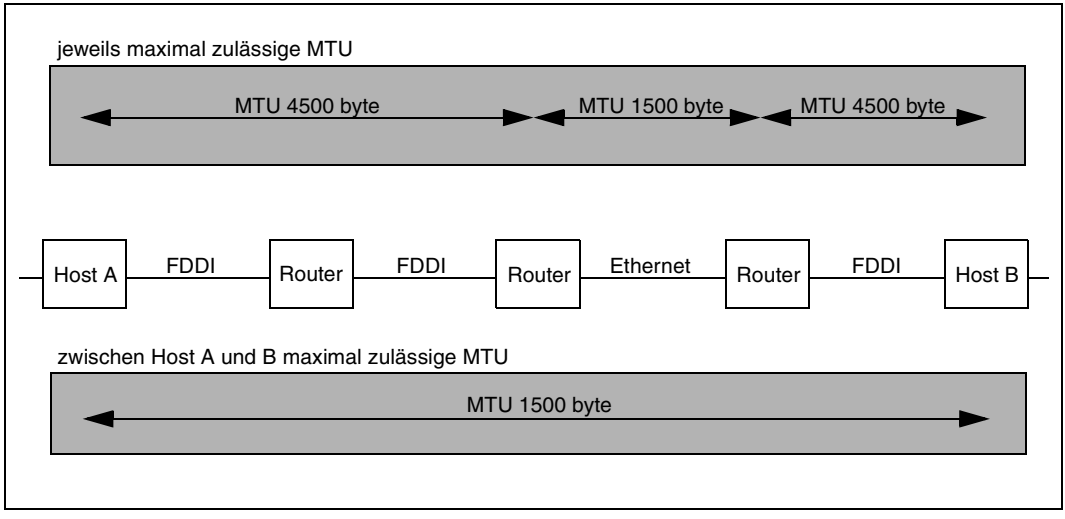

Bild 11: Festlegung der MTU

Die MTU auf einem Weg kann von den Endsystemen durch den MTU Path Discovery Process festgestellt werden. Dabei versucht das sendende Endsystem, die MTU des Weges zum Zielsystem durch Versuch zu ermitteln. Zuerst wird ein Paket mit der link-spezifischen MTU des Sendesystems verschickt. Falls diese MTU für einen Link auf dem Weg des Paketes zu lang ist, schickt der Router, der dies feststellt, eine ICMP-Nachricht „Datagram too big“ an das sendende Endsystem. Diese Nachricht enthält auch die MTU des Links, für den die ICMP-Nachricht gesendet wurde. Das sendende Endsystem kann nun seine Fragmentgröße entsprechend reduzieren und das kleinere Fragment versenden. Dies wird solange wiederholt, bis ein IPv6-Paket beim Ziel-Endsystem ankommt. Die so ermittelte MTU wird dann für die Fragmentierung von IPv6-Paketen zu dem jeweiligen Ziel-Endsystem benutzt.

4.2.5 Erweiterungsheader für sicheren Datentransfer

IPv6 bietet zwei Erweiterungsheader für die Gewährleistung eines sicheren Datentransfers:

– Erweiterungsheader für die Authentifizierung der ankommenden IPv6-Pakete
– Erweiterungsheader für die Verschlüsselung der IPv6-Pakete

4.2.5.1 Authentifizierungsheader

Durch die Verwendung des IPv6-Authentifizierungsheaders können Endsysteme die Authentizität und die Integrität von empfangenen IPv6-Paketen überprüfen. Der IPv6-Authentifizierungsheader nutzt eine aufgebaute Sicherheitszuordnung, die z.B. auf dem Austausch von geheimen Schlüsseln basiert. In einer Client-/Server-Anwendung müssen dann sowohl der Client als auch der Server diesen geheimen Schlüssel kennen. Bevor ein IPv6-Paket gesendet wird, wird ein Message Integrity Code (MIC) des gesamten Paketes unter Verwendung des geheimen Schlüssels erzeugt. Dabei werden auch die Daten des IPv6-Authentifizierungsheaders in die Signatur aufgenommen, um so genannte Replay-Angriffe zu verhindern. Der Message Integrity Code wird vom empfangenden System erneut berechnet und mit dem übertragenen Message Integrity Code verglichen. Bei Nichtübereinstimmung wird das Paket verworfen. Dies gewährleistet die Authentizität des Senders und garantiert, dass die Daten nicht durch Dritte verändert wurden.

Authentifizierung kann zwischen gleichberechtigten Partnern oder zwischen Clients und Servern verwendet werden.

4.2.5.2 Verschlüsselungsheader

Durch den Authentifizierungsheader können Host-Spoofing und Paketmodifikationen durch Dritte verhindert werden. Aber er schützt nicht gegen das Lesen von IPv6-Paketen durch Unberechtigte im Internet bzw. in einem Unternehmensnetzwerk. Dies wird durch den Verschlüsselungsheader (Encapsulating Security Payload (ESP)) von IPv6 verhindert. Pakete, die mit Hilfe des Verschlüsselungsheaders geschützt sind, haben einen höheren Grad an Privatheit und Integrität. Dies ist etwas, das im heutigen Internet nicht weit verbreitet ist, wenn man von speziellen, gesicherten Anwendungen wie privater E-Mail und sicheren HTTP Web-Servern absieht. Der Verschlüsselungsheader bietet eine Verschlüsselung auf Netzwerkebene, die für alle Anwendungen in standardisierter Art und Weise zur Verfügung steht

Der IPv6-Verschlüsselungsheader wird genutzt,

– um die Header des Transport Layers mit den folgenden Nutzdaten oder
– einen IPv6-Header mit den folgenden Nutzdaten zu verschlüsseln.

Beide Methoden werden durch den Verschlüsselungsheader realisiert, der eine End-to-End Verschlüsselung gewährleistet. Wenn nur die Header des Transport Layers und die darauf folgenden Nutzdaten verschlüsselt werden, wird der Verschlüsselungsheader direkt vor den Headern des Transport Layers (UDP, TCP) eingefügt. Dies wird als „Transport Mode“ der IPv6-Verschlüsselung bezeichnet (siehe Bild 12).

unverschlüsselt			verschlüsselt
IPv6 Hdr	Erweiterungsheader	ESP Hdr	Transport Hdr & Payload

Bild 12: Transport Mode der IPv6-Verschlüsselung

Falls die Gewährleistung eines noch höheren Maßes an Sicherheit die Verschlüsselung des gesamten IPv6-Paketes erfordert, wird dieses als Nutzdaten in ein neues IPv6-Paket mit einem Verschlüsselungsheader verpackt. Diese Art der Verschlüsselung eines kompletten IPv6-Paketes inklusive der ursprünglichen Adressinformation wird als „Tunnel Mode“ der IPv6-Verschlüsselung bezeichnet (siehe Bild 13).

unverschlüsselt			verschlüsselt			
IPv6 Hdr	Erw.-Hdr	ESP Hdr	IPv6 Hdr	Erw.-Hdr	ESP Hdr	Transport & Payload
zusätzlicher Verschlüsselungsheader			ursprüngliches Paket			

Bild 13: Tunnel Mode der IPv6-Verschlüsselung

Voll verschlüsselte IPv6-Pakete sind sicherer als IPv6-Pakete, die im Transport Mode verschlüsselt sind, da der ursprüngliche IPv6-Header nicht mehr für Verkehrsanalysen zur Verfügung steht.

Durch die Erzeugung und Bearbeitung eines zusätzlichen IPv6-Headers entsteht zwar eine entsprechende Performance-Verschlechterung, aber durch die Verschlüsselung im Tunnel Mode besteht die Möglichkeit, sichere Tunnel zwischen den Firewalls zweier entfernter Standorte zu erzeugen (siehe Bild 14). Das voll verschlüsselte IPv6-Paket gewährleistet, dass die ursprüngliche Adressinformation und eventuell vorhandene IPv6-Erweiterungsheader nicht öffentlich sichtbar sind. Innerhalb des Tunnels sind nur die Header des temporär benötigten, zusätzlichen IPv6-Headers sichtbar. Nachdem das Paket durch den Tunnel geschleust worden ist, wird der zusätzliche IPv6-Header wieder abgestreift und das Paket auf Grund seiner eigenen Adressinformation an sein Ziel-Endsystem weitergeleitet.

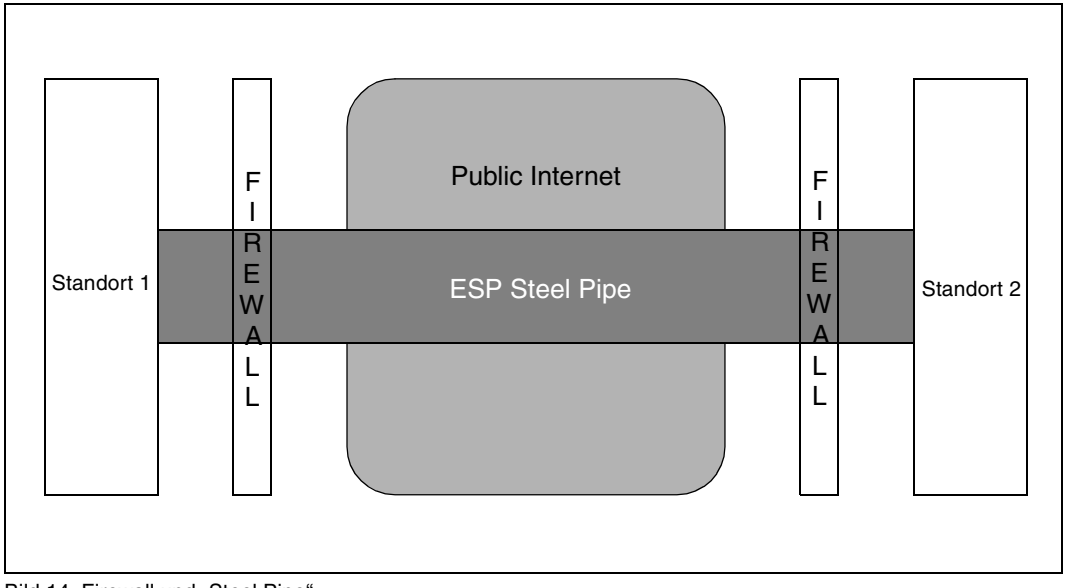

Bild 14: Firewall und „Steel Pipe“

4.2.5.3 Sicherheitslösung

Die Authentifizierungs- und Verschlüsselungsmöglichkeiten von IPv6 bilden zusammen eine Sicherheitslösung, die im heutigen Internet in Zeiten von E-Commerce immer öfter von Anwendungen benötigt wird. Ein Authentifizierungsheader ist oft Bestandteil eines im Tunnel Mode verschlüsselten Datagrams, und garantiert zusätzlich Sicherheit über die Identität des Senders. In anderen Fällen liegt ein Authentifizierungsheader vor einem im Transport Mode verschlüsselten Paket. Dieser Ansatz ist empfehlenswert, wenn eine Authentifizierung des Absenders vor der Entschlüsselung des Paketes durchgeführt werden soll.

Zusammengenommen bilden die Verschlüsselungs- und Authentifizierungsmöglichkeiten von IPv6 einen robusten standardisierten Sicherheitmechanismus, der eine entscheidende Rolle bei der Weiterentwicklung des Internet spielen wird.

4.3 IPv6-Adressarchitektur

Viele Diskussionen über die Vorteile von IPv6 gegenüber IPv4 beschränken sich auf die Größe der Adressfelder (128 bit bei IPv6 und 32 bit bei IPv4) der beiden Protokolle. Mindestens ebenso hoch zu bewerten ist jedoch die Möglichkeit bei IPv6, eine hierarchische Adress-Struktur zu bilden, die Voraussetzung einer effizienten Routing-Architektur ist.

4.3.1 IPv4-Adresshierarchie

IPv4 wurde ursprünglich mit Klasse A, Klasse B und Klasse C Adressen definiert.

<table>
<tr><th colspan="2">Bit 1 8</th><th>9 16</th><th>17 24</th><th>25 32</th><th>Adresstyp</th></tr>
<tr><td>0</td><td>Netzwerk-ID</td><td colspan="3">Host-ID</td><td>Klasse A</td></tr>
<tr><td>10</td><td colspan="2">Netzwerk-ID</td><td colspan="2">Host-ID</td><td>Klasse B</td></tr>
<tr><td>110</td><td colspan="3">Netzwerk-ID</td><td>Host-ID</td><td>Klasse C</td></tr>
</table>

Bild 15: IPv4-Adresstypen

Diese teilten die IPv4-Adresse in einen Netzwerk-Anteil und in einen Endsystem-Anteil. Sie definierten aber keine Hierarchie, die es einer Adresse auf höherem Level erlaubte, mehrere Adressen auf niedrigerem Level darzustellen. Hierarchische Adress-Systeme arbeiten ähnlich wie Ländervorwahlnummern bzw. Ortsnetzvorwahlnummer bei Telefonsystemen. Diese erlauben es bei Ferngesprächen, mit Hilfe eines Teils der Telefonnummer das richtige Land bzw. das richtige Ortsnetz zu adressieren.

Mit dem Anwachsen des Internet wurde die nicht-hierarchische Struktur des ursprünglichen IPv4-Adressraumes immer mehr zum Problem. Dieses Problem wurde durch die Nutzung von CIDR (siehe Seite 8) entschärft. Aber die Adressvergabe in IPv4 erschwerte weiterhin das Routing im Internet. Die Adressvergabe beschränkte sowohl das lokale als auch das globale Interworking.

Um die Schwierigkeiten im Bereich lokaler Netzwerke zu reduzieren, wurde die IPv4-Subnetztechnik entwickelt, die eine leichtere Verwaltung großer Netzwerke gestattete. Durch die Nutzung von IPv4-Subnetzen kann eine Netzwerkadresse bzw. der Netzwerkanteil einer IPv4-Adresse für mehrere physikalische Netzwerke stehen. Dadurch wird eine erhebliche Anzahl von IPv4-Adressen eingespart. So können zum Beispiel durch eine Klasse B Netzwerkadresse hunderte von physikalischen Netzwerken mit jeweils hunderten von einzelnen Endsystemen adressiert werden.

Auf der Ebene von großen Internet Backbones und globalem Routing können IPv4-Adressen durch Supernetting, einer Art der hierarchischen Adressierung, effizienter zusammengefasst werden. Beim Supernetting verwalten Backbone-Router eine einzelne Adresse um

das Weiterleiten von Paketen an mehrere Teilnetze durchzuführen. Dies reduziert die Größe der Routing-Tabellen in den Backbone-Routern. Dadurch wird die Performance des Routings gesteigert und die Router benötigen weniger Speicherplatz für die Verwaltung der Routing-Tabellen.

Subnetting und Supernetting haben sich als effektiv erwiesen, um IPv4-Class C-Adressen besser zu nutzen. Beide Techniken ermöglichen es Routern, eine Adresshierarchie für die Weiterleitung von IPv4-Paketen zu bilden.

Der Prozess, eine IPv4-Routinghierarchie zu bilden, wurde durch den CIDR Ansatz formalisiert. So erlaubt es CIDR zum Beispiel, eine Anzahl von Klasse C-Adressen zu einem Adresspräfix zusammenzufassen, mit dem ähnlich wie mit den schwer zu erhaltenden Klasse A- bzw. Klasse B-Adressen gearbeitet werden kann. CIDR hat die Lebenszeit von IPv4 verlängert und das Wachstum des Internet zu seiner jetzigen Größe erst ermöglicht. Aber es wurde nicht überall im Internet und in allen Unternehmensnetzwerken konsequent eingeführt. Dadurch konnten die Vorteile von CIDR bei der Einsparung von IPv4-Adressen und der Verbesserung des Routings nicht voll genutzt werden. Aufgrund der Struktur der IPv4-Netzwerke und der Schwierigkeiten, die sich bei einer Umstrukturierung der IPv4-Adressen ergeben würden, können die Vorteile von CIDR auch in Zukunft nicht genutzt werden. In IPv4 wird deshalb auch weiterhin ein großer Teil des Adressraumes verschwendet und die Router werden mit unnötig großen und ineffizienten Routing-Tabellen belastet.

Im Unternehmensbereich des Internetworking erfordert IPv4 einen hohen administrativen Aufwand bei der Verwaltung von Subnetzbitmasken und Endsystem-Adressen innerhalb dieser Subnetzstruktur. Dies trifft besonders dann zu, wenn eine große, sich häufig ändernde Anzahl von Endbenutzern verwaltet werden muss. Wenn zum Beispiel ein Endbenutzer innerhalb der Subnetumgebung eines Unternehmens umzieht, muss dieser Umzug sehr sorgfältig geplant und durchgeführt werden, damit der Anwender nicht unnötig lange vom Unternehmensnetzwerk abgekoppelt ist.

4.3.2 IPv6-Adresshierarchie

Durch die mit IPv4 gesammelten Erfahrungen wurde beim Design von IPv6 von Anfang an großer Wert auf die Skalierbarkeit des Adressraumes und die Möglichkeit einer effizienten globalen Routing-Hierarchie gelegt. An der Spitze der Routinghierarchie stehen Adressbereiche, die durch Top Level Aggregator (TLA) beschrieben werden. Innerhalb der durch die Top Level Aggregator definierten Adressbereiche werden weitere Adressbereiche durch Next Level Aggregator (NLA) definiert. Diese Top Level Aggregator bzw. Next Level Aggregator werden an einzelne ISPs oder an global tätige Unternehmen und Organisationen vergeben. Innerhalb des Next Level Aggregator werden dann einzelne Adressen oder auch Adressbereiche an die Endbenutzer bzw. Kunden vergeben. Auf Grund dieser Struktur ist ein effizientes Routing im Backbone-Bereich möglich, da für alle Next Level Aggregator eines Top Level Aggregators nur ein Top Level Aggregator-Eintrag in den Backbone-Routern benötigt wird (siehe Bild 16).

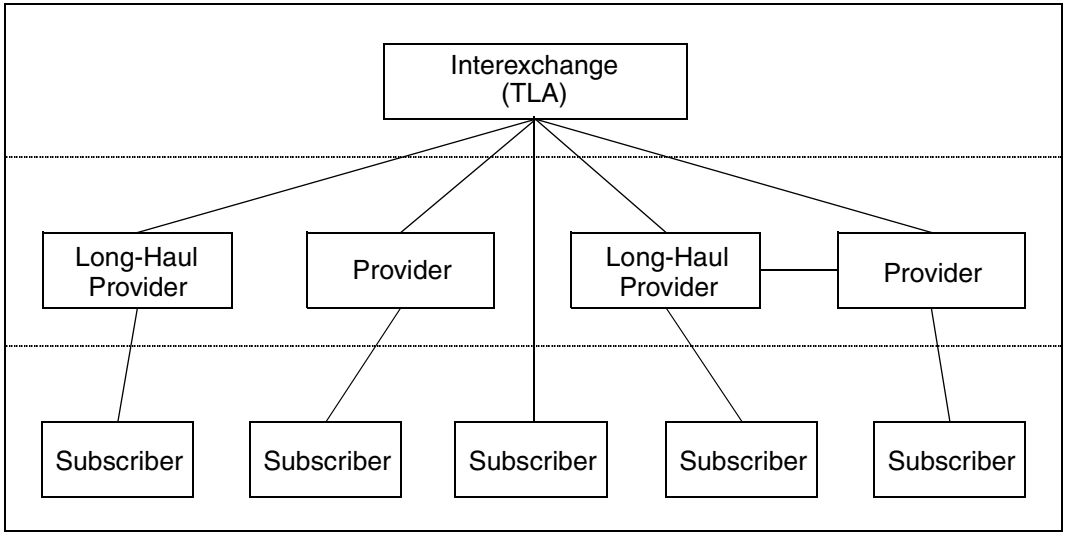

Bild 16: „Aggregator based“-Adressierungschema

Obwohl innerhalb des großen IPv6-Adressraumes eine Vielzahl von Adressierungsschemata möglich wäre, wurde das „Aggregator based“-Schema gewählt, da es die größte Flexibilität bei der Vergabe von Adressen ermöglicht. Eine nur provider-basierte Adressvergabe kann die geographischen Gegebenheiten der Netzwerkinfrastruktur nicht berücksichtigen, während eine Vergabe der Adressen nur nach geografischen Gegebenheiten, wie zum Beispiel im Telefonnetz, oft den Anforderungen von großen Service-Providern widerspricht.

Die „Aggregator based“-Adress-Struktur berücksichtigt die topologische Struktur des Internet mit einer begrenzten Anzahl von Interconnecting-Punkten, an denen Netzübergänge zwischen großen ISPs und Telefonnetzen stattfinden. Die Nutzung dieser Topologie durch IPv6-Adressen beinhaltet damit sowohl eine geografische Komponente als auch eine, die die Struktur der ISPs berücksichtigt.

3 bit	13 bit	8 bit	24 bit	16 bit	64 bit
FP	TLA ID	reserviert	NLA ID	SLA ID	Interface ID
Public Topology				Side Topology	Interface Identifier

Bild 17: „Aggregator-based“-IPv6-Adressen

In Bild 17 wird die Struktur einer IPv6-Adresse gezeigt. Die ersten 3 Bit zeigen an, um welche Art von Adresse es sich handelt (Unicast, Multicast usw.). Die nächsten 13 Bit bilden die Top Level Aggregator ID (TLA ID). Darauf folgen 8 reservierte Bit, die zur Zeit nicht genutzt werden, aber für zukünftige Erweiterungen vorgesehen sind. Daran schließen sich 24 Bit für Next Level Aggregator Identifier (NLA ID) und 16 Bit für Site Level Aggregator Identifier (SLA ID) an.

Das Next Level Aggregator-Adressfeld kann von den ISPs weiter unterteilt werden, um damit eine eigene Adresshierarchie zu definieren (siehe hierzu Bild 18).

24 bit				16 bit	64 bit
NLA 1	Site			SLA	Interface ID
	NLA 2	Site		SLA	Interface ID
		NLA 3	Site	SLA	Interface ID

Bild 18: Unterteilung des NLA-Adressraums

Normalerweise werden ISPs ihren Kunden Bereiche von IPv6-Adressen zur Verfügung stellen. Innerhalb dieser Bereiche können die Kunden der ISPs ihre eigene Adresshierarchie definieren. So lassen sich mit dem Site Level Aggregator-Feld 65535 unterschiedliche Subnetze definieren, in denen jeweils bis zu 2^{64} unterschiedliche Endsysteme enthalten sein können. Diese 64 bit lange Interface ID wird dazu benutzt, die einzelnen IPv6-Interfaces zu identifizieren. Sie wird normalerweise aus der Link Layer Adresse (MAC Adresse) des Interfaces abgeleitet, kann aber auch gesondert generiert werden.

Internet Backbone-Router verwalten derzeit bis zu 40.000 unterschiedliche Routen. Da das Internet weiter wächst, stellt die hierarchische Adress-Struktur von IPv6 die einzige Möglichkeit dar, das Wachstum der Routing-Tabellen in den Backbone-Routern zu kontrollieren. Aufgrund der Hierarchie der zusammengesetzten IPv6-Adressen können alle internen Netzwerke einer beliebigen Hierarchiestufe von den zugehörigen Backbone-Routern durch einen einzigen Routing-Eintrag der übergeordneten Hierarchiestufe adressiert werden. So ist es auf der höchsten Hierarchieebene der Internet Backbone-Router nur erforderlich, für die Weiterleitung der IPv6-Pakete den Top Level Aggregator-Anteil der IPv6-Adresse zu betrachten.

Der große IPv6-Adressraum erlaubt ebenso eine dezentrale Art der Adressvergabe. ISPs können innerhalb der von ihnen verwalteten Adressbereiche Adressen völlig unabhängig von zentralen Verwaltungsinstanzen vergeben. Damit entfallen die bürokratischen Behinderungen bei der Adressvergabe, wie sie sich heute auf Grund der Knappheit von IPv4-Adressen ergeben.

Die zusammengesetzten IPv6-Adressen bilden nur einen Teilbereich des durch IPv6 definierten Adressbereiches. Andere Adressbereiche wurden für Multicast-Adressen definiert. Ebenso wurden Adressbereiche für lokale Adressen eines Standortes bzw. Links definiert.

Standort-lokale bzw. link-lokale Adressen sind private lokale Adressen, die nur innerhalb des Standortes bzw. Links gültig sind und nicht offiziell registriert werden müssen. IPv6-Pakete mit standort-lokalen bzw. link-lokalen Adressen werden nicht außerhalb des Standortes bzw. Links weitergeleitet. Dies ermöglicht es zum Beispiel zwei unterschiedlichen Unternehmen, die selben standort-lokalen bzw. link-lokalen Adressen zu verwenden, ohne dass es dadurch zu einem Konflikt bei der eindeutigen Adressierung der Endsysteme kommt.

Standort-lokale bzw. link-lokale Adressen haben einen großen Vorteil, wenn sie für die interne Kommunikation eines Unternehmens genutzt werden. In diesem Fall ist diese interne Kommunikation bei einem Wechsel des ISPs nicht betroffen, sondern nur die Kommunikation mit externen Partnern.

Link-lokale IPv6-Adressen haben nur innerhalb eines Links Gültigkeit, also werden sie von Routern nicht weitergeleitet. Sie werden beim „bootstrapping“ eines Endsystems benutzt, bevor dieses eine global eindeutige IPv6-Adresse erhält (siehe nächster Abschnitt). Sie können jedoch auch nach Abschluss des „bootstrapping“ noch weiter benutzt werden.

4.4 Adress-Autokonfiguration eines Endsystems

IPv6 stellt einen ausreichend großen Adressraum zur Verfügung, um damit das Wachstum des Internet in den kommenden Jahrzehnten zu ermöglichen. Außerdem können IPv6-Endsysteme ihre Adressen auf eine kostengünstige und leicht überschaubare Art und Weise konfigurieren bzw. rekonfigurieren. Bei hierarchischen Routingstrukturen ist eine automatische Adresskonfiguration notwendig, da nur sie eine skalierbare Adressvergabe bzw. eine Umnummerierung von IPv6-Endsystemen ermöglicht. Selbst wenn der Aufwand für die Vergabe einer Adresse bzw. die Umnummerierung eines Endsystemes gering ist, summiert sich dies bei einem ISP bzw. einem Großunternehmen doch zu einem erheblichen Gesamtaufwand. Wenn die Gesamtkosten für eine Umnummerierung jedoch gering sind, so kann ein Wechsel zu einem anderen ISP für ein Unternehmen durchaus kostensparend sein, während dies derzeit wegen des hohen Umstellungsaufwands kaum möglich ist.

Die Autokonfiguration bietet unabhängig von der Art der Adressvergabe vielfältige Möglichkeiten. Eine Umnummerierung aller Endsysteme in einem Unternehmen kann zum Beispiel notwendig werden, wenn bei einer provider-basierten Adressierung der Internet Service Provider gewechselt wird oder wenn bei einer geographisch orientierten Adressierung das Unternehmen an einen anderen Ort umzieht.

Auf der Ebene von Arbeitsgruppen oder Abteilungen eines vernetzten Unternehmens gehört eine Änderung der IP-Adressen zum alltäglichen Leben. IP-Adressen müssen für neue Endsysteme vergeben werden, für Endsysteme, die ihren Standort wechseln oder für Endsysteme, die von physikalischen Änderungen im Netzwerk betroffen sind. Zusätzlich zu die-

sen mehr konventionellen Anforderungen an die Konfiguration durch die Netzwerkadministration ergeben sich in neuerer Zeit zusätzliche Anforderungen durch die wachsende Zahl von mobilen Endsystemen.

Mit IPv4 lassen sich die Anforderungen an eine einfache dynamische Konfiguration der Endsysteme und Router nicht erfüllen. Auch deshalb wurde in IPv6 das Konzept der Adress-Autokonfiguration entwickelt.

Der Prozess der Adress-Autokonfiguration beginnt mit dem Neighbor Discovery Protocol.

Das Neighbor Discovery Protocol kombiniert und erweitert die Dienste, die in der IPv4-Umgebung durch das Address Resolution Protocol (ARP), das Internet Control Message Protocol (ICMP) und Router Advertising erbracht wurden. Das Neighbor Discovery Protocol basiert auf IPv6-spezifischen Erweiterungen des Internet Control Message Protocols. Die neuen ICMP-Nachrichten ermöglichen es Routern und Endsystemen, am selben Link sowohl Adressen und als auch andere Netzwerkparameter auszutauschen.

Nach dem Anschluss an ein Netzwerk beginnt ein Endsystem normalerweise den Prozess der Adress-Autokonfiguration, indem es für den Netzwerk-Anschluss eine link-lokale Adresse erzeugt. Diese Adresse wird normalerweise aus der IEEE Interface-Adresse abgeleitet. Mit der so gebildeten Adresse sendet das Endsystem eine Neighbor Discovery-Nachricht, um sicherzustellen, dass sie wirklich eindeutig ist:

– Falls darauf keine ICMP Neighbor Solicitation-Nachricht zurückkommt, ist die gebildete Adresse eindeutig.

– Falls eine ICMP Neighbor Solicitation-Nachricht zurückkommt, ist die Adresse nicht eindeutig und das neue angeschlossene Endsystem muss einen erneuten Versuch mit einer anderen Adresse machen. Dies geschieht solange, bis das Endsystem eine eindeutige link-lokale IPv6-Adresse hat.

Mit der link-lokalen IPv6-Adresse als Absenderadresse sendet das Endsystem dann eine ICMP-Nachricht „Neighbor Discovery Router Solicitation“. Als Zieladresse wird eine IPv6-Multicast-Adresse verwendet, mit der alle Router am Link angesprochen werden. Die Router antworten auf die ICMP-Nachricht „Neighbor Discovery Router Solicitation“ mit einer ICMP-Nachricht „Neighbor Discovery Router Advertisement“, die neben anderen Daten auch den Adresspräfix des lokalen Netzwerkes enthält. Das neu angeschlossene Endsystem kann jedoch auch auf das Aussenden einer ICMP-Nachricht „Neighbor Discovery Router Solicitation“ verzichten und auf eine der von den Routern periodisch ausgesendeten ICMP-Nachrichten „Neighbor Discovery Router Advertisement“ warten. Bild 19 zeigt den Austausch der ICMP-Nachrichten zwischen dem neuen Endsystem X und dem Router.

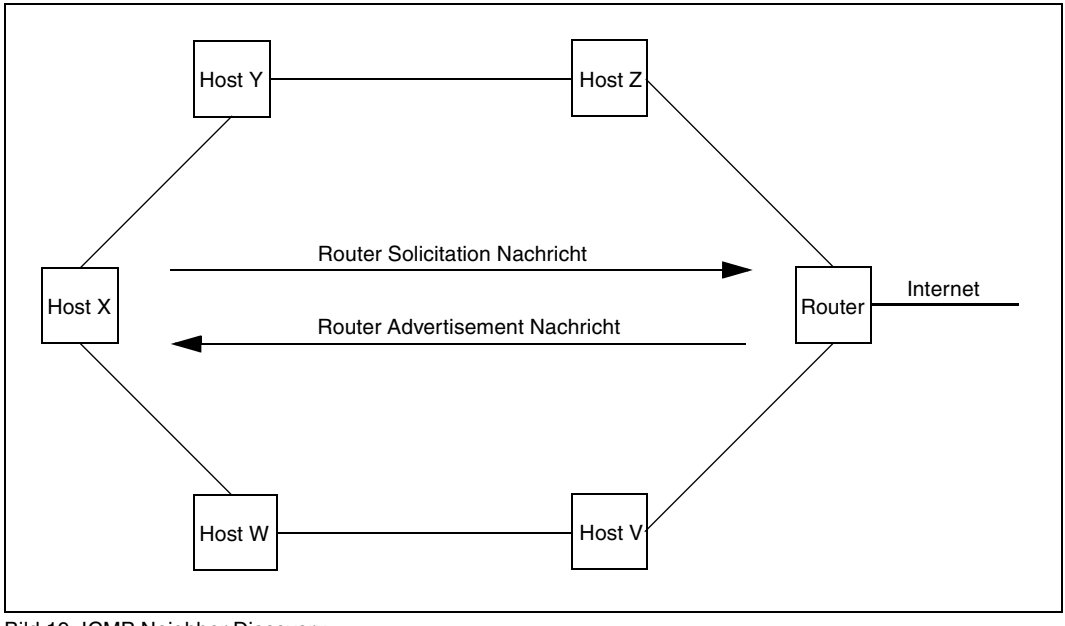

Bild 19: ICMP Neighbor Discovery

Die ICMP-Nachricht „Neighbor Discovery Router Advertisement“ steuert dann, ob das neu angeschlossene Endsystem für den Netzanschluss eine „Stateful“ oder eine „Stateless“ Adress-Autokonfiguration durchführt.

Im Fall der „Statefull“ Adress-Autokonfiguration kontaktiert das Endsystem nun einen Adress-Server, der ihm seine globale Adresse mitteilt. Dies geschieht durch das von IPv4 bereits bekannte und für IPv6 entsprechend erweiterte DHCP-Protokoll.

Im Fall der „Stateless“ Adress-Autokonfiguration kann das neue Endsystem seine globale IPv6-Adresse ohne einen Adress-Server oder einen menschlichen Eingriff nun selbständig generieren. Dazu nutzt das Endsystem die global eindeutige Adresspräfix-Information, die es vom Router in der ICMP-Nachricht „Neighbor Discovery Router Advertisement“ erhalten hat. Dieses Adresspräfix bildet zusammen mit dem Endsystem-Anteil der link-lokalen Adresse eine weltweit eindeutige globale Adresse.

Sobald ein Endsystem mehrere Netzanschlüsse hat, wird dieser Prozess der Adress-Autokonfiguration für jeden Netzanschluss durchgeführt.

Die „Stateless“ Adress-Autokonfiguration hat viele Vorteile. Zum Beispiel kann ein Unternehmen bei einem Wechsel des Internet Service Providers die neuen Adresspräfixe automatisch durch seine Router an alle Endsysteme verteilen. Sofern alle Endsysteme die Stateless Adress-Autokonfiguration nutzen, ist es theoretisch sogar möglich, die Umnummerierung ohne einen einzigen Eingriff bei den Endsystemen durchzuführen. Auf einer niedrigeren Ebene können zum Beispiel Arbeitsgruppen mit oft wechselnden Endsys-

temen ebenfalls von der „Stateless“ Adress-Autokonfiguration profitieren, da diese beim Einschalten jedesmal automatisch konfiguriert und mit gültigen IPv6-Adressen versorgt werden.

Die Adress-Autokonfiguration spielt bei der Unterstützung von mobilen Endsystemen ebenfalls eine wichtige Rolle. Jedes mobile Endsystem kann immer eine gültige IPv6-Adresse erhalten, unabhängig davon, wo es an das Netzwerk angeschlossen wird. Diese Adresse wird als „care-of-Adresse“ bezeichnet und dient als Forwarding-Adresse. Das mobile Endsystem teilt die „care-of-Adresse“ einem Router in seinem ursprünglichen Netz mit. Dieser leitet daraufhin den gesamten Datenverkehr an das mobile Endsystem weiter. Das mobile Endsystem kann aber auch direkte Kommunikationspartner über seine „care-of-Adresse“ informieren, sodass diese ohne den Umweg über das ursprüngliche Netz mit ihm kommunizieren können.

Das mobile Endsystem wird jedoch auch immer über seine ursprüngliche Adresse identifiziert. Dies ist wichtig, um eine eindeutige Identifikation des Endsystems zu gewährleisten und auch einen Ortswechsel während des Betriebs zu ermöglichen, wie dies bei einem drahtlosen Anschluss üblich ist.

Um bei einer Umnummerierung einen gleitenden Übergang von alten zu neuen IPv6-Adressen zu ermöglichen, wurde in IPv6 ein entsprechender Übergangsmechanismus realisiert. Basis dieses Mechanismus ist, dass jedes IPv6-Endsystem bzw. jeder IPv6-Router mehrere IPv6-Adressen je physikalischem Netzanschluss unterstützt. IPv6-Adressen, die einem Netzanschluss zugeordnet sind, können entweder gültig, entwertet oder ungültig sein. Während der Umnummerierung wird die bisherige IPv6-Adresse eines Netzanschlusses entwertet, solange dem Netzanschluss eine neue IPv6-Adresse zugeteilt wird. Die entwertete Adresse kann aber für eine gewisse Zeit noch zum Senden und Empfangen von IPv6-Paketen verwendet werden. Dies erlaubt es Anwendungen, die die alte Adresse noch nutzen, sich geordnet zu beenden. Später wird die entwertete IPv6-Adresse dann ungültig und kann nicht mehr für Kommunikationsbeziehungen genutzt werden. Die neu zugeordnete Adresse ist dann die einzig gültige. Die Nutzung mehrerer IPv6-Adressen für ein Interface erlaubt eine dynamische Umnummerierung, die für Anwendungen und den Endanwender transparent ist. Neben der Umnummerierung von Endsystemen ist für IPv6 auch eine Automatisierung der Umnummerierung von Routern geplant. Die entsprechenden Arbeiten sind jedoch noch nicht abgeschlossen.

Der oben beschriebene Mechanismus der „Stateless“ Adress-Autokonfiguration ist besonders für die konventionelle IP/LAN Welt mit 48 bit bzw. 64 bit Adressierung auf Link Layer-Ebene und hardwaremäßiger Unterstützung des Multicastings geeignet. Netzwerkumgebungen mit einer anderen Charakteristik des Link Layers können eine Modifikation der obigen Verfahren oder sogar gänzlich andere Verfahren erforderlich machen. So unterstützten beispielsweise die gegenwärtigen ATM-Netzwerke kein Multicasting. Es werden jedoch bereits Forschungsaktivitäten unternommen, das IPv6 Neighbor Discovery-Protokoll an die speziellen Gegebenheiten von ATM-Netzwerken sowie anderen Netzwerktechnologien anzupassen.

4.5 Andere Protokolle und Dienste

In den vorhergehenden Betrachtungen wurden die innovativen und teilweise radikalen Änderungen, die IPv6 für das Internetworking bringt, aufgezeigt. In vielen anderen Bereichen des Internetworkings arbeiten jedoch die Protokolle und Dienste bei der Nutzung von IPv6 ähnlich wie dies bisher bei der Nutzung von IPv4 der Fall ist. Wenn sich das Internet auf die Nutzung von IPv6 umstellt, müssen zum Beispiel PPP-, DHCP-, und DNS-Server auf die Nutzung der 128 bit langen IPv6-Adressen umgestellt werden. An der Basisfunktionalität dieser Protokolle ändert sich aber nichts. Dies gilt auch für die verschiedenen Routing-Protokolle.

So ist zum Beispiel ein modifiziertes OSPF-Protokoll mit einer vollen Unterstützung von IPv6 verfügbar, das es erlaubt, die Router mit den 128 bit langen IPv6-Adressen zu adressieren. Außerdem wurden die 32 Bit „linkstate“ Sätze des IPv4 OSPF durch 128 Bit „linkstate“ Sätze für IPv6 ersetzt. Die OSPF-Datenstrukturen für die Unterstützung von IPv6 werden in den Routern parallel zu den OSPF-Datenstrukturen für die Unterstützung von IPv4 gehalten, sodass sich die beiden Datenstrukturen und das Weiterleiten von IPv4- und IPv6-Paketen nicht gegenseitig beeinflussen. Durch die geringfügigen Änderungen, die im OSPF-Protokoll für IPv6 erfolgt sind, sollten Netzadministratoren, die Erfahrung mit OSFP für IPv4 haben, beim Einsatz der OSPF-Version für IPv6 auch keine Probleme haben. Ein entsprechender Update von IPv4 zu IPv6 ist auch für das RIP Routing-Protokoll definiert.

Analog zu den Interior Gateway-Protokollen definiert RFC 2545 eine IPv6-kompatible Version eines Exterior Gateway-Protokolls, das von den Routern benutzt wird, um die Erreichbarkeit von ISPs, großen Unternehmen und anderen Organisationen über den Backbone des Internet sicherzustellen. Die IPv4-basierten Backbone-Router nutzen das Border Gateway Protocol (BGP), um CIDR-basierte Routinginformation über das Internet zu verbreiten. Derzeit werden die Erweiterungen für ein IPv6-basiertes BGP gerade definiert.

5 Übergang von IPv4 nach IPv6

Die Realisierung eines IPv6- und eines IPv4-Netzwerkes sind vergleichbar. In beiden Fällen muss ein entsprechender Adressraum zur Verfügung gestellt werden, der von den jeweiligen Endsystemen und Routern genutzt wird. Die Router sowie der Domain Name Service (DNS) sind korrekt zu initialisieren.

Der Übergang existierender IPv4-Endsysteme in die IPv6-Welt wird meist mit Hilfe einer Dual Stack-Strategie erfolgen. Alternativ besteht jedoch auch die Möglichkeit, dass neue Endsysteme bereits von Anfang an nur IPv6 unterstützen. Neben der Hochrüstung von Endsystemen und Routern auf IPv6 müssen zusätzlich noch administrative Vorkehrungen für den IPv6-Einsatz getroffen werden. Dies betrifft besonders den Domain Name Service DNS.

Die Adressierung von Anwendungen sollte nicht direkt über IP-Adressen, sondern über logische Namen erfolgen. In IPv4 wurden noch häufig die 32 bit langen Adressen angegeben. Dies ist bei den vierfach längeren IPv6-Adressen zu umständlich und fehleranfällig. Zur Umsetzung von Namen zu Adressen und umgekehrt dient der auf einem entsprechenden Server installierte Domain Name Service.

Zur Verwaltung der IPv6 Resource Records (A6) wird der vom Internet Software Consortium (ISC) bereitgestellte BIND Version 9 Name Server benötigt. Sofern in einem Netzwerk kein IPv6-fähiger DNS-Server vorhanden ist, kann die Information zur Namens- und Adressumsetzung auch lokal in den jeweiligen Endsystemen verwaltet werden. In BS2000/OSD beispielsweise geschieht dies durch die Host-Datei.

5.1 Basis-Übergangsmechanismen

RFC 1933 definiert zwei Basismechanismen für den Übergang von IPv4 nach IPv6.

- Dual Stack-Endsysteme und -Router

  Dieser Mechanismus unterstützt in vollem Umfang sowohl das IPv4- als auch das IPv6-Protokoll.

- IPv6 über IPv4-Tunneling

  IPv6-Pakete werden in IPv4-Pakete verpackt und über ein existierendes IPv4-Netzwerk transportiert.

5.1.1 Dual Stack

Dual Stack-Endsysteme und -Router können mit IPv4- ebenso wie mit IPv6-Partnern kommunizieren. Dazu stellen die Dual Stack-Endsysteme ein User Interface zur Verfügung, das sowohl den Zugang zum IPv4- als auch zum IPv6-Netzwerkprotokoll ermöglicht. Ein wichtiger Bestandteil dieser User Interfaces sind Funktionen zur Namens- und Adressumsetzung, die vom Domain Name Service unterstützt werden.

5.1.2 Tunneling

IPv6-Endsysteme oder Netzwerke, die nur durch IPv4-Netzwerke miteinander gekoppelt sind, können durch Tunnel miteinander verbunden werden. IPv6-Pakete, die einen solchen Tunnel durchqueren, werden in IPv4-Pakete verpackt. Beide Endpunkte des Tunnels haben sowohl eine IPv4- als auch eine IPv6-Adresse. Es existieren zwei Arten von Tunneln, konfiguierte Tunnel und automatische Tunnel. Konfigurierte Tunnel werden durch entsprechende Konfigurationsmaßnahmen erzeugt. Bekanntestes und vom Umfang her größtes Beispiel für die Nutzung von konfigurierten Tunneln ist das 6Bone. Automatische Tunnel benötigen keine manuellen Eingriffe. Die Tunnel-Endpunkte werden automatisch durch IPv4-kompatible IPv6-Adressen erzeugt.

5.2 Hilfsmittel in Systemlösungen

Die Einführung von IPv6 im Internet und in Firmennetzen führt zu zwei problematischen Konstellationen:

– Zum Einen entstehen vereinzelte in eine IPv4-Umgebung eingebettete IPv6-Inseln, die miteinander kommunizieren wollen.

– Zum Anderen muss auch zwischen der existierenden IPv4-Welt und der neu entstehenden IPv6-Welt eine Kommunikation möglich sein.

Das erste Problem kann meist durch den Einsatz von Dual Stack-Routern und Tunneling von IPv6-Paketen gelöst werden. Die Lösung für das zweite Problem basiert auf Dual Stack-Mechanismen, Anwendungsgateways, NAT-Techniken oder der zeitweiligen Nutzung von IPv4-Adressen und ”IPv6 in IPv4”-Tunneling.

Die hier dargestellten Mechanismen basieren auf der Tunnel-Technologie und dienen dazu, einzelne isolierte IPv6-Inseln über die vorhandene IPv4-Infrastruktur miteinander zu verbinden.

5.2.1 Konfigurierte Tunnel

Manuell konfigurierte Tunnel werden benutzt, um einzelne IPv6-Endsysteme oder Netzwerke über die vorhandene IPv4-Infrastruktur miteinander zu verbinden. Typischer Weise werden konfigurierte Tunnel zwischen Standorten eingesetzt, die regelmäßig miteinander kommunizieren.

Anwendungsbereich	**Standorte**
IPv4-Anforderungen	die einzelnen Standorte müssen über IPv4 miteinander verbunden sein
Anforderung an IPv4-Adresse	≥ 1 je Standort
IPv6-Anforderungen	keine
Anforderung an IPv6-Adresse	keine speziellen
Anforderungen an Endsysteme	IPv6 Stack oder Dual Stack
Anforderungen an Router	Dual Stack
andere Anforderungen	keine

5.2.2 Automatische Tunnel

Automatische Tunnel werden wie manuell konfigurierte Tunnel dazu benutzt, um einzelne IPv6-Endsysteme oder Netzwerke über die vorhandene IPv4-Infrastruktur miteinander zu verbinden. Automatische Tunnel werden bei Bedarf dynamisch aufgebaut und, wenn sie nicht mehr benötigt werden, wieder abgebaut. Sie werden bevorzugt zwischen Standorten eingesetzt, die nur selten miteinander kommunizieren. Eine Vorbedingung für die Nutzung von automatischen Tunneln stellt die Verwendung von IPv4-kompatiblen IPv6-Adressen durch die Endsysteme dar, die miteinander kommunizieren. Diese Adressen erlauben es den Endsystemen, die IPv4-Adresse der Tunnel-Endpunkte automatisch zu bestimmen.

Anwendungsbereich	**Standorte**
IPv4-Anforderungen	die einzelnen Standorte müssen über IPv4 miteinander verbunden sein
Anforderung an IPv4-Adresse	≥ 1 je Standort
IPv6-Anforderungen	keine
Anforderung an IPv6-Adresse	IPv4-kompatible IPv6-Adresse
Anforderungen an Endsysteme	Dual Stack
Anforderungen an Router	keine
Andere Anforderungen	keine

5.2.3 Tunnelmakler

Konfigurierte Tunnel verlangen die Kooperation von zwei Partnern, um die beiden Tunnel-Endpunkte korrekt zu definieren. Das Konzept des Tunnelmaklers hilft den am Aufbau von konfigurierten Tunneln beteiligten Endsystemen, die notwendigen Informationen zum Aufbau des Tunnels zu sammeln. Ein Tunnelmakler kann als ein IPv6-ISP gesehen werden, der eine IPv6-Connectivity über IPv4-Tunnel anbietet. Die gegenwärtig dafür verfügbaren Tools sind Web-basierte Anwendungen, die ein interaktives Einrichten der konfigurierten Tunnel erlauben.

Durch die Anforderung des konfigurierten Tunnels wird dem eigenen Endsystem eine IPv6-Adresse aus dem Adressbereich des Tunnelanbieters zugeordnet. Die entsprechenden DNS-Einträge werden automatisch erzeugt.

Der erzeugte konfigurierte Tunnel bietet eine IPv6-Connectivitiy zwischen der IPv6-Umgebung des Tunnelanbieters und dem eigenen Endsystem.

Anwendungsbereich	**Endsystem**
IPv4-Anforderungen	keine speziellen
Anforderung an IPv4-Adresse	1
IPv6-Anforderungen	keine
Anforderung an IPv6-Adresse	keine
Anforderungen an Endsysteme	Dual Stack, Web Browser
Anforderungen an Router	keine
Andere Anforderungen	Tunnel Server

5.2.4 6to4-Konzept

Das 6to4-Konzept ist anwendbar für die Verbindung von isolierten IPv6-Domänen in einer IPv4-Welt. Der Router am Rande einer IPv6-Domäne erzeugt einen Tunnel zu einer anderen IPv6-Domäne. Die IPv4-Endpunkte des Tunnels werden durch einen Teil des Präfixes der IPv6-Domäne identifiziert und adressiert. Dieses Präfix wird aus einem eindeutigen 6to4 Top Level Aggregator und einem Next Level Aggregator gebildet, der aus der IPv4-Adresse des Boarder Routers der IPv6-Domäne besteht. Ein weiterer interessanter Aspekt des 6to4-Konzepts besteht in der automatischen Ableitung eines 48 bit langen IPv6-Adresspräfix aus einer IPv4-Adresse. Durch diesen Mechanismus können Anwender IPv6 nutzen, ohne sich jemals offizielle IPv6-Adressen besorgt zu haben. Das erweist sich dann als wertvoll, wenn der ISP eines Anwenders aus Aufwandsgründen keine IPv6-Unterstützung anbietet.

Anwendungsbereich	**Standorte**
IPv4-Anforderungen	die einzelnen Standorte müssen über IPv4 miteinander verbunden sein
Anforderung an IPv4-Adresse	≥ 1 je Standort
IPv6-Anforderungen	global eindeutiger 6to4-Adress-Präfix
Anforderung an IPv6-Adresse	keine
Anforderungen an Endsysteme	IPv6-Protokollmaschine
Anforderungen an Router	Implementierung der speziellen Regeln zum 6to4-Forwarding
andere Anforderungen	Erzeugung spezieller DNS Resource Records für die 6to4-Adress-Präfixe und die als Next Level Aggregator genutzten IPv4-Adressen

5.2.5 6over4-Konzept

Das 6over4-Konzept verbindet isolierte IPv6-Endsysteme eines Standortes durch Verpacken von IPv6-Paketen in IPv4-Paketen ohne explizite Tunnel. Dazu wird eine virtuelle Verbindung unter Verwendung einer lokalen IPv4-Multicast-Gruppe aufgebaut. Die IPv6-Multicast-Adressen werden auf IPv4-Multicast-Adressen abgebildet, was Voraussetzung dafür ist, dass der Prozess der IPv6 Neighbor Discovery funktioniert. Um das Routing zwischen dem IPv6-Internet und der 6over4-Domäne zu gewährleisten, muss mindestens ein Router des Standortes 6over4-Routing bearbeiten können.

Anwendungsbereich	**Endsysteme**
IPv4-Anforderungen	die einzelnen Endsysteme müssen über IPv4 miteinander verbunden sein
Anforderung an IPv4-Adresse	1 je Endsystem
IPv6-Anforderungen	keine
Anforderung an IPv6-Adresse	keine
Anforderungen an Endsysteme	Dual Stack
Anforderungen an Router	6over4-Routing muss installiert sein
andere Anforderungen	Damit IPv6-Endsysteme, die sich an unterschiedlichen Links befinden, miteinander kommunizieren können, müssen IPv4-Multicast-Pakete von den Routern weitergeleitet werden.

5.2.6 Kommunikation zwischen IPv4- und IPv6-Endsystemen

Wenn die IPv6-Inseln eingerichtet und über einen oder mehrere der oben beschriebenen Mechanismen miteinander verbunden sind, ist die Kommunikation zwischen den einzelnen IPv6-Endsystemen möglich. Die Schaffung einer Kommunikationsmöglichkeit zwischen dem neuen IPv6-Endsystem und den bereits existierenden IPv4-Endsystemen ist ebenfalls wichtig. Sie kann zum Beispiel durch einen Anwendungsgateway, einen Protokollübersetzer auf der Ebene des Netzwerk-Layers oder durch zeitweilige Nutzung von IPv4-Adressen durch das IPv6-Endsystem realisiert werden.

Anmerkung zu Protokollübersetzern:

Normalerweise bildet ein Protokollübersetzer die Felder aus dem Header eines Protokolls auf semantisch ähnliche Felder im Header eines anderen Protokolls ab. Für die Protokollübersetzung zwischen IPv4 und IPv6 wurde ein entsprechendes Regelwerk in RFC 2765 (SIIT) definiert (siehe Seite 60). Es ist jedoch im IPv4-Umfeld üblich, dass Anwendungen genaue Kenntnis über Informationen des Netzwerk-Layers haben. Ein Beispiel dafür ist FTP. In solchen Fällen muss dann nicht nur eine Protokollübersetzung zwischen IPv4 und IPv6 erfolgen, sondern auch noch eine Protokollübersetzung in den Anwenderdaten, was jedoch die Nutzung der IPv6-Sicherheitsfunktionen für diese Anwendungen verhindert.

5.2.6.1 Dual Stack

Im Dual Stack-Modell ist in allen Endsystemen und Routern sowohl IPv4 als auch IPv6 implementiert. Dadurch erfolgt die Kommunikation zu IPv4-adressierten Endsystemen über das IPv4-Protokoll und die Kommunikation zu IPv6-adressierten Endsystemen über das IPv6-Protokoll. Bei diesem Ansatz ergibt sich eine Einschränkung daraus, dass für jedes neue Endsystem neben der IPv6-Adresse auch eine IPv4-Adresse benötigt wird. Die Erweiterung des Adressraumes und die Vereinfachung der Konfiguration von IPv6 lassen sich somit nicht nutzen.

Anwendungsbereich	**Standort**
IPv4-Anforderungen	IPv4-Adressierungsplan und IPv4-Routingplan
Anforderung an IPv4-Adresse	1 je Endsystem. Mehrere je Router
IPv6-Anforderungen	IPv6-Adressierungsplan und IPv6-Routingplan
Anforderung an IPv6-Adresse	1 je Endsystem. Mehrere je Router
Anforderungen an Endsysteme	IPv4- und IPv6-Protokollmaschine
Anforderungen an Router	IPv4- und IPv6-Protokollmaschine
andere Anforderungen	keine

5.2.6.2 Eingeschränkter Dual Stack

Im eingeschränkten Dual Stack-Modell beschränkt sich die IPv4- und IPv6-Implementierung auf ausgesuchte Server. Neue Clients verfügen nur über IPv6-Adressen. Ein Server zeichnet sich dadurch aus, dass die Internet-Services eines Unternehmens wie DNS, Web-Server, Mail-Server, usw. auf ihm ablaufen. Ein Client ist dadurch gekennzeichnet, dass diese Dienste auf ihm nicht ablaufen. Mit diesem Ansatz werden deutlich weniger IPv4-Adressen als in dem oben angeführten Dual Stack-Modell benötigt. Aber die Kommunikation zwischen neuen Clients mit IPv6-Adressen und alten Clients mit IPv4-Adressen ist nicht mehr möglich und muss auf den Servern erst mit speziellen Anwendungsgateways wieder ermöglicht werden.

Anwendungsbereich	**Standort**
IPv4-Anforderungen	Nutzung der vorhandenen IPv4-Infrastruktur
Anforderung an IPv4-Adresse	1 je Server
IPv6-Anforderungen	IPv6-Adressierungsplan und IPv6-Routingplan
Anforderung an IPv6-Adresse	1 je Endsystem. Mehrere je Router
Anforderungen an Endsysteme	IPv4- und IPv6-Protokollmaschine auf den Servern, IPv6-Protokollmaschine auf den Clients
Anforderungen an Router	IPv4- und IPv6-Protokollmaschine
andere Anforderungen	keine

5.2.6.3 SOCKS64-Konzept

Das SOCKS-Gateway ist ein Gateway-System, das in RFC 1928 definiert ist. Es stellt eine Erweiterung des ursprüngliches SOCKS Protokolls dar, das es IPv4-Endsystemen erlaubt, sich an ein SOCKS-Gateway zu wenden, das dann als Relay zum eigentlichen IPv6-Ziel-Endsystem dient. Dasselbe Prinzip kann auch für IPv6-Endsysteme verwendet werden, die eine Verbindung zu einem IPv4-Endsystem aufbauen wollen. Der Einsatz des erweiterten SOCKS-Protokolls ist besonders empfehlenswert, wenn bereits die alten IPv4-Anwendungen SOCKS genutzt haben. Die Nutzung des SOCKS-Protokolls macht keine Änderungen im DNS erforderlich.

Anwendungsbereich	**Standort**
IPv4-Anforderungen	keine
Anforderung an IPv4-Adresse	1 je Endsystem
IPv6-Anforderungen	keine
Anforderung an IPv6-Adresse	eine oder mehrere pro Endsystem
Anforderungen an Endsysteme	die Anwendungen müssen das SOCKS-Protokoll nutzen
Anforderungen an Router	keine
andere Anforderungen	Dual Stack SOCKS-Server

5.2.6.4 SIIT-Protokoll

Das SIIT-Protokoll beschreibt eine Übersetzungsmethode zwischen IPv4 und IPv6. Die Übersetzung ist auf den IPv4- bzw. IPv6-Header beschränkt. Da der Übersetzer ohne einen Zustandsautomaten arbeitet, muss er jedes Paket ohne Kontextwissen über frühere Pakete bearbeiten. Diese Einschränkung bedingt, dass mit diesem Übersetzer keine Sicherheitsmechanismen von IPv6 genutzt werden können. Außerdem können auch keine Netzwerk-Protokollinformationen übersetzt werden, die in höheren Schichten übertragen werden. Die Zuordnung einer temporären IPv4-Adresse an das IPv6-Endsystem ist nicht Bestandteil des SIIT-Protokolls.

Anwendungsbereich	**Standort**
IPv4-Anforderungen	keine
Anforderung an IPv4-Adresse	1 temporäre je Endsystem
IPv6-Anforderungen	keine
Anforderung an IPv6-Adresse	IPv4 mapped IPv6-Adresse
Anforderungen an Endsysteme	IPv6-Protokollmaschine
Anforderungen an Router	keine
andere Anforderungen	keine

5.2.6.5 NAT-PT-Konzept

Das in RFC 2766 definierte NAT-PT-Konzept ermöglicht die direkte Kommunikation zwischen reinen IPv6-Endsystemen und reinen IPv4-Endsystemen. Es basiert auf einem eigenständigen Gerät, das die Übersetzung zwischen IPv4 und IPv6 durchführt. Während der Übersetzung wird ein Zustandsautomat für jede logische Session verwaltet. Auf dem NAT-PT-Gerät befindet sich auch ein Anwendungs-Gateway, der eine Übersetzung von IPv4-DNS-Anfragen bzw. -Antworten und IPv6-DNS-Anfragen bzw. -Antworten durchführt.

Anwendungsbereich	**Standort**
IPv4-Anforderungen	keine
Anforderung an IPv4-Adresse	≥ 1 je Standort
IPv6-Anforderungen	keine
Anforderung an IPv6-Adresse	keine
Anforderungen an Endsysteme	IPv6-Protokollmaschine
Anforderungen an Router	keine
andere Anforderungen	keine

5.2.6.6 Konzept Bump in the Stack (BIS)

Das in RFC 2767 definierte Konzept „Bump in the Stack“ ermöglicht es IPv4-Anwendungen, auf IPv4-Endsystemen mit reinen IPv6-Endsystemen zu kommunizieren. Dabei werden im IPv4-Protokollstack drei zusätzliche Module zwischen der Anwendung und dem Netzwerk Layer eingefügt:

1. eine Erweiterung des Nameresolvers,
2. ein Baustein zur Adressabbildung IPv4/IPv6,
3. ein Protokollübersetzer.

Hinter „Bump in the Stack“ steckt die Idee, dass wann immer eine IPv4-Anwendung mit einem IPv6-Partner kommuniziert, die IPv6-Adresse des Partners auf eine nur im Endsystem lokal gültige IPv4-Adresse abgebildet wird. Die Übersetzung zwischen IPv4-Paketen und IPv6-Paketen erfolgt den SITT-Regeln entsprechend.

Man kann „Bump in the Stack“ als eine spezielle Implementierung eines NAT-PT innerhalb der IP-Protokollmaschine eines Endsystems ansehen. Auf einem Dual Stack-Endsystem kann die Technik „Bump in the Stack“ auch als Bibliotheksfunktion ohne Eingriff in den Systemkern realisiert werden.

Anwendungsbereich	**Endsystem**
IPv4-Anforderungen	keine
Anforderung an IPv4-Adresse	ein Bereich privater Adressen
IPv6-Anforderungen	keine
Anforderung an IPv6-Adresse	keine
Anforderungen an Endsysteme	IPv4- und IPv6-Protokollmaschine mit Erweiterungen
Anforderungen an Router	keine
andere Anforderungen	keine

5.2.6.7 Dual Stack Transition Mechanism (DSTM)

Der Dual Stack Transition Mechanism ist die Kombination zweier Mechanismen.

– Assignment of IPv4 Global Adresses to IPv6 hosts (AIIH).
– Dynamic Tunneling Interface (DTI).

AIIH basiert auf einer Zusammenarbeit zwischen DNS und DHCPv6.

„Dual Stack Transition Mechanism“ geht davon aus, dass ein Dual Stack-Endsystem, das mit einem reinen IPv4-Endsystem kommunizieren will, von einem AIIH Server für die Dauer der Kommunikationsbeziehung eine IPv4-Adresse anfordert. Falls ein reines IPv4-Endsystem die Kommunikationsbeziehung beginnen will, fragt es zuerst einen DNS-Server nach einem A Resource Record mit der IPv4-Adresse des Partner-Endsystems. Wenn dieser DNS-Server keinen Typ A Resource Record hat, beauftragt er einen DHCP-Server, dem Partner-Endsystem eine IPv4-Adresse zuzuweisen. Der DHCP-Server sendet die zugewiesene IPv4-Adresse an den DNS-Server, damit dieser die Anfrage beantworten kann. Gleichzeitig sendet er jedoch auch ein Rekonfigurationskommando an das Partner-Endsystem, damit dieses die zugewiesene IPv4-Adresse auch als eigene IPv4-Adresse verwendet.

Falls das Dual Stack Partner-Endsystem nicht über eine IPv4-Infrastruktur erreichbar ist, verpackt es die IPv4-Pakete in IPv6-Pakete und sendet sie an einen Tunnel-Endpunkt. Dort wird der zusätzliche IPv6-Header wieder abgestreift und das Paket über die IPv4-Infrastruktur weitergeleitet. Dieses Verpacken von IPv4-Paketen in IPv6-Pakete wird durch das Dynamic Tunneling Interface durchgeführt.

Anwendungsbereich	**Standort**
IPv4-Anforderungen	keine
Anforderung an IPv4-Adresse	≥ 1 je Standort
IPv6-Anforderungen	DHCPv6
Anforderung an IPv6-Adresse	keine
Anforderungen an Endsysteme	IPv4- und IPv6-Protokollmaschine
Anforderungen an Router	keine
andere Anforderungen	keine

5.3 Beispiele für typische Umstellszenarien

Die in den nachfolgenden Abschnitten dargestellten Beispiele veranschaulichen die Umstellung von IPv4 auf IPv6. Die ersten, recht allgemein gehaltenen Beispiele zeigen typische Umstellszenarien. Im Abschnitt „Beispiele aus realen Installationen“ auf Seite 68 werden Umstellszenarien anhand realer Installationen eingehend erläutert.

5.3.1 Große Organisationen mit vielen IPv4-Adressen

Im folgenden Beispiel wird ein großes, über viele Standorte verteiltes Unternehmen betrachtet. Dieses Unternehmen verfügt über eine ausreichende Zahl von global eindeutigen IP-Adressen. Aus arbeitstechnischen Gründen ist eine komplette Umstellung des Unternehmens zu einem Zeitpunkt nicht möglich. Die Einführung von IPv6 wird das Entstehen von Inseln in der IPv4-Welt zur Folge haben.

Mögliche Übergangsmechanismen

Für die Realisierung der internen Kommunikation bieten sich folgende Techniken an:

- Dual Stack
- Tunneling
- 6over4, falls ein Netzwerk mit Multicasting vorhanden ist.

Eine Protokollübersetzung ist nur für reine IPv6-Endsysteme notwendig.

Die Connectivity mit der externen IPv4-Welt erfordert keine zusätzlichen Vorkehrungen. Falls der Internet Service Provider keine IPv6-Connectivity anbietet, müssen automatische oder konfigurierte Tunnel für die Kommunikation mit externen IPv6-Partnern eingerichtet werden.

5.3.2 Große Organisationen mit wenigen IPv4-Adressen

Dieses Beispiel zeigt die IPv6-Umstellung anhand eines großen Unternehmens auf, das zwar auch über viele Standorte verteilt ist, aber nicht in ausreichendem Maß über global eindeutige IPv4-Adressen verfügt. Dieser Umstand führt dazu, dass im Intranet private IPv4-Adressen genutzt werden und der Übergang ins Internet von einem NAT dargestellt wird. Auch in diesem Unternehmen muss die IPv6-Umstellung Schritt für Schritt erfolgen.

Mögliche Übergangsmechanismen

Für die interne Kommunikation ergibt sich dieselbe Situation wie im vorhergehenden Beispiel:

– Dual Stack
– Tunneling
– 6over4, falls ein Netzwerk mit Multicasting vorhanden ist.

Eine Protokollübersetzung ist nur für reine IPv6-Endsysteme notwendig.

Durch die Nutzung des privaten IPv4-Adressraums ergeben sich jedoch deutliche Unterschiede bei der externen Kommunikation.

Falls der Internet Service Provider keine IPv6-Connectivity anbietet, müssen für die Kommunikation mit externen IPv6-Partnern automatische oder konfigurierte Tunnel auf Dual Stack Routern mit mindestens einer global eindeutigen IPv4-Adresse eingerichtet werden.

Für die Kommunikation mit externen IPv4-Partnern gibt es die folgenden Möglichkeiten:

– Benutzung des bisherigen NATs

– Verwendung von NAT-PT

– Nutzung von DSTM, um einem Endsystem temporär eine global eindeutige IP-Adresse zuzuordnen. Falls reine IPv6-Endsysteme installiert werden, ist ein spezieller Übersetzungsmechanismus erforderlich.

5.3.3 Büro mit einer IPv4-Adresse

Dieses Beispiel behandelt ein Büro mit einer geringen Anzahl von Endsystemen, die an einem Subnetz hängen. Die Kommunikation mit der Außenwelt erfolgt über einen NAT, dem eine IPv4-Adresse zugeordnet ist.

Mögliche Übergangsmechanismen

Interne Kommunikation	**Externe Kommunikation mit IPv6-Partnern**
IPv6 IPv4 mit privaten Adressraum	1. IPv6-direkt, falls der ISP die IPv6-Funktionalität anbietet 2. Dynamische Tunnel von den Boarder Routern aus zu den Partner-Endsystemen 3. Konfigurierte Tunnel zu Systemen, die von ISPs bedient werden, die IPv6 unterstützen. 4. Für die Kommunikation mit IPv4-Partnern: NAT-PT

5.3.4 Neues Netzwerk

Ein völlig neu aufgebautes Netzwerk mit offiziellen IPv6-Adressen ist Thema dieses Beispiels.

Es sollte ein Internet Service Provider gewählt werden, der einen IPv6-Netzzugang anbietet, weil dann der Zugang zum externen Netz über IPv6 erfolgen kann. Eine Verwendung von IPv4-Adressen ist nicht erforderlich.

Falls der ausgewählte ISP keinen IPv6-Netzzugang anbietet, muss ein als Dual Stack Router ausgelegter Boarder Router installiert werden. Außerdem sind eine IPv4-Adresse und konfigurierte bzw. automatische Tunnel erforderlich.

Mögliche Übergangsmechanismen

Für die interne Kommunikation, die ausschließlich über globale IPv6-Adressen erfolgt, wird ein IPv6-fähiger DNS-Server installiert.

Falls der ISP keine Dienste wie Dual Stack DNS oder NAT-PT für die externe Kommunikation mit reinen IPv6-Endsystemen anbietet, werden neben einer IPv4-Adresse für den Standort die folgenden Mechanismen benötigt:

– Dual Stack DNS
– NAT-PT

Beim Anschluss über ISP, die Dienste für reine IPv6-Endsysteme anbieten, ist für die externe Kommunikation kein weiterer Aufwand nötig.

5.3.5 Internet Service Provider (ISP)

Die primäre Aufgabe eines Internet Service Providers besteht in der Schaffung einer Übertragungsmöglichkeit zwischen seinen Kunden, Intranets, Einzelpersonen und dem Rest des Internet. Zusätzlich bieten ISPs oft auch noch andere Dienste an wie Virtual Privat Networks (VPN), DNS-Server, DHCP-Server usw.

Die Details eines Wechsels zu einem anderen ISP können naturgemäß hier nicht komplett abgehandelt werden, da es eine zu große Bandbreite in den Kundenbeziehungen, Dienstleistungen, Netzwerktopologien usw. der einzelnen ISPs gibt. Es ist jedoch sinnvoll, die ISPs grob in zwei Kategorien einzuordnen.

– Backbone ISPs ISPs, die einen Top Level Aggregator-Präfix haben
– Kleine ISPs ISPs, die keinen Top Level Aggregator-Präfix haben

Bei der obigen Kategorisierung ist jedoch zu beachten, dass auch ein kleiner ISP wiederum Dienstleister für noch kleinere ISPs sein kann und damit auch die Dienste eines Backbone-ISP anbieten muss.

Mögliche Übergangsmechanismen

Als Minimum sollte ein ISP die folgenden beiden Dienste anbieten:

– Gewährleistung der IPv6-Connectivity zum IPv6-Backbone.
Dabei ist unbedingt eine direkte IPv6-Connectivity anzustreben. Für die Anfangszeit kann auch eine durch Tunnel realisierte IPv6-Connectivity durchaus ausreichend sein.

– Gewährleistung der IPv6-Connectivity zu seinen Kunden.
Auch hier ist zu beachten, dass eine direkte IPv6-Connectivity notwendig ist. Auch in diesem Fall kann für eine Übergangszeit eine durch Tunnel realisierte IPv6-Connectivity ausreichend sein.

Für die interne Kommunikation wird der ISP bevorzugt die direkte IPv6-Kommunikation oder Tunnel zu den IPv6-fähigen Routern verwenden.

Die externe Kommunikation weist in diesem Beispiel zwei wichtige Aspekte auf:

1. Connectivity zum IPv6-Backbone und zu anderen ISPs

   Die Art der Connectivity ist hier abhängig von der Größe des ISP. Große oder Backbone-ISPs werden fast automatisch eine Verbindung zum IPv6-Backbone und zu anderen ISPs erhalten. Die Details des Peerings mit anderen ISPs werden in gegenseitigen Verträgen analog dem Peering in der IPv4-Welt vereinbart.

   Kleinere ISPs, von denen es bedeutend mehr gibt als von den Backbone-ISPs, müssen einer anderen Logik folgen. Ein kleiner ISP muss zuerst von einem größeren ISP, der nicht unbedingt ein Backbone ISP sein muss, einen Block von IPv6-Adressen und die Connectivity zu dessen Netzwerk erhalten. Damit haben kleinere ISPs indirekt Zugang zum IPv6-Backbone. Die Details hierzu werden in Verträgen geregelt. Dabei wird offensichtlich, dass eine direkte IPv6-Connectivity anzustreben ist. Für die Anfangszeit kann eine durch Tunnel realisierte IPv6-Connectivity aber durchaus ausreichend sein. Anschließend kann der kleinere ISP auch noch direkte Peering-Verträge mit anderen ISPs abschließen.

2. Connectivity zu den Kunden des ISP

   Um seine IPv6-Connectivity an die Kunden weiter zu geben und damit einen geschäftlichen Nutzen daraus zu ziehen, hat der ISP mehrere Möglichkeiten:

   – Direkte IPv6-Connectivity zum IPv6-Backbone.
   Dabei ist eine direkte IPv6-Connectivity anzustreben. Für eine Übergangszeit kann eine durch Tunnel realisierte IPv6-Connectivity ausreichend sein.

   – Ein 6to4-Gateway zum IPv6-Backbone.

   – IPv6-Connectivity per Übersetzungsdienst wie SIIT oder NAT-PT.

Abhängig von den Wünschen der Kunden und den Präferenzen des ISP wird sich in der Praxis oft eine Kombination der Zugangsmöglichkeiten ergeben. Die endgültige Ablösung von IPv4 durch IPv6 sollte jedoch immer als Ziel im Auge gehalten werden. Der ISP sollte daher besonderes Augenmerk darauf legen, dass er den Weg zu diesem Ziel nicht unnötig erschwert.

5.4 Beispiele aus realen Installationen

In den folgenden Abschnitten werden einige Beispiele aus realen Installationen vorgestellt. Diese Beispiele verstehen sich als Lösungsvorschläge, erheben aber keineswegs den Anspruch, der einzig mögliche oder für jeden Fall beste Lösungsansatz zu sein.

5.4.1 Isoliertes IPv6-Endsystem in einer IPv4-Domäne

Ein Firmenkunde benötigt eine Verbindung zu einem neuen System seiner Bank. Er installiert für die IPv6-Verbindung zu der Bank ein IPv6-Endsystem. Die Bank verfügt über eine reine IPv6-Installation, um die erweiterte Funktionalität von IPv6, besonders hinsichtlich Sicherheit und Authentifizierung, nutzen zu können. Der Boarder Router der Bank (R2) ist zwar als Dual Stack Router ausgelegt, aber das Netzwerk der Bank wird ausschließlich mit IPv6 betrieben.

Migrationsanforderungen

- Das IPv6-Endsystem muss mit allen Endsystemen innerhalb des Kundennetzwerks kommunizieren können.
- Um die Verschlüsselungs- und Authentifizierungsmechanismen mit der von IPv6 gebotenen Funktionalität nutzen zu können, ist der Gebrauch von Übersetzern nicht gestattet.
- Es dürfen keine Änderungen am Netzwerk der Bank vorgenommen werden.

5.4.1.1 Betrachtung möglicher Umstellungsmechanismen

IPv4- bzw. IPv6-Mechanismen erlauben es den Endsystemen innerhalb der Kundeninstallation, mit externen bzw. IPv6-Endsystemen zu kommunizieren. Tunneling wird zur Übertragung der IPv6-Pakete innerhalb von IPv4-Netzwerken benötigt. Protokollübersetzer sind wegen der aus Sicherheitsgründen geforderten End-to-End-Verbindung nicht geeignet. Dual Stack muss im Endsystem des Kunden installiert sein. Der Boarder Router (R1) muss ebenfalls mit IPv4 und IPv6 betrieben werden, um Routing innerhalb des IPv4-Netzwerkes zu erlauben.

Das IPv4-/IPv6-Endsystem soll so konfiguriert sein, dass alle IPv6-Pakete zum Default IPv4-/IPv6-Router geleitet werden. Die IPv6-Pakete werden in Pakete verpackt, sodass sie durch die IPv4-Infrastruktur zum Router gesendet werden können. Als problematisch erweist es sich, die IPv6-Adresse so zu konfigurieren, dass benachbarte Anforderungen durch konfiguriertes Tunneling zum Router übertragen werden. Wenn ein Tunnel zum Router R1 konfiguriert wird, gilt das ebenfalls für den Router R2 (Bank) und der Router R1 bleibt ein normaler IPv4-Router.

6over4 kann genutzt werden, wenn das Netzwerk Multicast Routing unterstützt. Da 6over4 nur innerhalb der IPv4-Domäne funktioniert, muss der Router R1 in der Lage sein, IPv6 zu unterstützen und mit einem 6over4-Interface konfiguriert sein. Das Endsystem kann dann mit dem Router R1 kommunizieren, indem es IPv4-Multicast Pakete benutzt. Der Rest des Übertragungsweges wird durch einen anderen Mechanismus abgedeckt, z.B. durch konfiguriertes Tunneling oder 6to4.

6to4 kann in den Routern R1 und R2 installiert werden, indem ihre eindeutigen IPv4-Adressen wie ein Next Level Aggregator ID verwendet werden, um eine eindeutige IPv6-Adresse zu erzeugen.

5.4.1.2 Lösungsvorschlag 1

Im folgenden Lösungsansatz unterstützt das IPv4-Netzwerk Multicasting. Hierzu werden folgende Mechanismen vorgeschlagen:

– Dual Stack
– 6over4
– 6to4 oder konfiguriertes Tunneling

Router

Der 6over4-Mechanismus verlangt, dass im IPv4-Netzwerk der Boarder Router R1 mit einem IPv6 und einem 6over4-Interface betrieben wird. Es ist nicht notwendig, dass sich der Router und das Endsystem im selben Segment befinden. Wenn dies dennoch der Fall ist, ist 6over4 nicht nötig. Es wird vorausgesetzt, dass zwischen dem Router und dem Endsystem eine IPv4-Infrastruktur existiert. Zwischen den Routern R1 und R2 muss ein konfigurierter Tunnel aufgebaut sein, damit IPv6-Pakete über das IPv4-Internet übertragen wer-

den können. Die Router R1 und R2 könnten 6to4 nutzen, was aber bedeuten würde, dass der Router R2 ebenfalls 6to4 unterstützen müsste. Dies widerspricht jedoch der Forderung, die Bankkonfiguration unverändert zu belassen.

Endsystem

Zuerst wird im IPv4-Endsystem eine Dual Stack-Implementierung installiert, wobei es seine ursprüngliche IPv4-Adresse behält. Die darauf folgende Realisierung von 6over4 erlaubt es dem Endsystem, IPv6-Pakete innerhalb von IPv4-Multicast Paketen zu verpacken. Der Router R1 muss so konfiguriert werden, dass er IPv6-Routing unterstützt. Das Endsystem findet seinen Adress-Präfix dadurch, dass es eine in ein IPv4-Multicast Paket verpackte Router-Solicitation-Nachricht an den Router R1 sendet. Dieser antwortet darauf mit einer ebenfalls in ein IPv4-Multicast-Paket verpackten Router-Advertisement-Nachricht.

5.4.1.3 Lösungsvorschlag 2

Für die folgende Lösung benötigt das IPv4-Netzwerk keine Multicast-Unterstützung. Es sollen Dual Stack oder konfigurierte Tunnel zum Einsatz kommen:

Es wird vorausgesetzt, dass im Endsystem Dual Stack installiert ist. Sollte die automatische Zuweisung einer IPv6-Adresse fehlschlagen, muss eine eindeutige IPv6-Adresse manuell eingegeben werden, wobei das Präfix des Routers R2 benutzt wird. Wenn dies nicht möglich ist, kann die Adresse ermittelt werden, indem ein in ein IPv4-Paket verpacktes Router-Advertisement zum Router 2 geschickt wird. Danach muss ein Tunnel vom Endsystem zum Bank-Router R2 manuell konfiguriert werden. Sobald dies geschehen ist, kann das Endsystem mit dem Partner-Endsystem zu kommunizieren.

Beide Lösungen verlangen die Nutzung von Security Funktionen. Ob MD5 als Authentifizierungsalgorithmus oder DES-CBC als Verschlüsselungsalgorithmus verwendet wird, hängt davon ab, was die Bank für ihr System fordert.

Bei der Implementierung ist zu beachten, dass nicht nur Sicherheitsattacken gegen IPv6, sondern auch gegen IPv4 möglich sind. Der Gebrauch von IP-Security auf IPv4- und IPv6-Ebene sollte aus Effektivitätsgründen trotzdem vermieden werden. Läuft IPv6 beispielsweise verschlüsselt, wäre die Verschlüsselung von IPv4 redundant, es sei denn, auch eine Verkehrsanalyse des IPv4-Verkehrs wird als Sicherheitsbedrohung angesehen. Falls auf IPv6-Ebene eine Authentifizierung genutzt wird, bringt eine weitere Authentifizierung auf IPv4-Ebene keine zusätzliche Sicherheit.

IPv4-Sicherheitsmechanismen sind nutzlos, wenn die Pakete den IPv6-Bereich über den IPv4-Bereich verlassen. Deshalb ist die Nutzung der IPv6-Sicherheitsmechanismen auch dann erforderlich, wenn IPv4-Sicherheitsmechanismen vorhanden sind.

Obwohl die obigen Überlegungen zur Sicherheit der Datenübertragung für den Fall von 6over4 angestellt wurden, gelten sie genauso für andere Sicherheitsmechanismen.

5.4.2 Kleine / mittlere Organisation unter Verwendung von NAT

Es existieren neun Büros, von denen jedes über eine Punkt-zu-Punkt-Verbindung angeschlossen ist. Jeder Standort hat einen DHCP-, DNS- und Mail-Server. Zwischen den Büros werden Router verwendet, um den Verkehr zu verringern. Jeder Router unterstützt das Routing-Protokoll RIP. Zur Zeit wird im Netzwerk ein privater Adressraum mit 192.168/16 Präfix genutzt, wobei jeder Standort einen /24 Präfix verwendet.

Hauptbüro

Das Hauptbüro in London verfügt über einen DNS-Server und ein NAT an der Grenze, um die global nicht eindeutigen IPv4-Adressen in global eindeutige IPv4-Adressen umzuwandeln. Dies geschieht nicht nur aus Sicherheitsgründen, sondern auch um eine eigene internationale Adress-Struktur zu erhalten. Der gesamte externe Verkehr sowie der Verkehr, der für externe Ziele bestimmt ist, wird durch den NAT geschickt. Das externe Verkehrsaufkommen ist niedrig, enthält aber einen hohen Prozentsatz an SMTP-Verkehr. Zusätzlich hat das Hauptbüro einen Firewall, der so konfiguriert ist, dass jeder hereinkommende SMTP-Verkehr von der IP-Adresse des ISP-Servers zum internen Mail-Router, einem LINUX-System mit SENDMAIL, gelangt. Danach überprüft der interne Mail-Router den Domain-Namen in der Kopfzeile der Nachricht und schickt sie direkt zum zuständigen Mail-Server des Büros. Auf dem gleichen System laufen Proxy Dämon Squid und NAT. Ein Intranet-Server läuft auf einem separaten LINUX-System mit Apache.

Der NAT im Hauptbüro wird für die externe Kommunikation verwendet, ihm ist die IPv4-Adresse 194.14.1.1 zugewiesen. Die gesamte externe Kommunikation läuft über dieses Gerät.

Migrationsanforderungen

- Um den laufenden Betrieb uneingeschränkt aufrecht zu erhalten, soll an der Grenze des Netzwerkes ein Protokollübersetzer eingesetzt werden.
- Während der Umstellung muss jederzeit eine Kommunikation mit den reinen IPv4-Endsystemen möglich sein.
- Innerhalb des Kundennetzwerkes sollte möglichst der gesamte IPv4-Datentransfer eliminiert werden.

5.4.2.1 Betrachtung der Umstellungsmechanismen

IPv4- und IPv6-Mechanismen

IPv4- bzw. IPv6-Mechanismen erlauben es den Endsystemen, innerhalb der Kunden-Installation mit externen IPv4- bzw. IPv6-Endsystemen zu kommunizieren.

Tunneling

Tunneling wird zur Übertragung der IPv6-Pakete innerhalb von IPv4-Netzwerken benötigt.

Protokollübersetzer

Ein Protokollübersetzer bietet die Möglichkeit, die NATs an der Grenze des Netzwerkes zu ersetzen. Die Abwicklung der gesamten externen Kommunikation erfolgt dann durch den Protokollübersetzer. Dies bedeutet auch, dass die internen Strukturen des IPv4-Netzwerks unverändert blieben, ohne Abweichung innerhalb des privaten Netzwerks.

Dual Stack mit konfigurierten Tunneln

Konfigurierte Tunnel werden notwendig, wenn nicht die komplette Netzwerk-Infrastruktur zwischen den einzelnen Standorten auf IPv6 umgestellt ist. Zwar wäre eine komplette Umstellung der Netzwerk-Infrastruktur auf IPv6 für den Betrieb einfacher, jedoch ist dies besonders im WAN-Bereich, der außerhalb der Kontrolle des Kunden liegt, nicht immer durchführbar.

Dual Stack mit automatischen Tunneln

Die Nutzung von automatischen Tunneln erlaubt es, die Endsysteme bereits vor den Routern auf IPv6-Betrieb umzustellen. Dies setzt jedoch voraus, dass jedem Endsystem eine IPv4-kompatible IPv6-Adresse zugeteilt wird. Damit würden die Tunnel direkt von Endsystem zu Endsystem führen. Da die einzelnen Standortnetzwerke jedoch klein sind und sich nur wenige Router innerhalb eines Standortes befinden, ist es sinnvoller, auf automatische Tunnel zu verzichten und die Router bereits zu Beginn der Umstellung auf Dual Stack hochzurüsten.

Dual Stack und NAT

Bereits am Eingangsbereich des Netzwerkes befindet sich ein NAT. Alle Endsysteme innerhalb der Organisation können zu Dual Stack hochgerüstet werden. Der NAT kann zur Umwandlung von IPv6- in IPv4-Adressen verwendet werden. Dies befähigt alle Endsysteme im Netzwerk, miteinander zu kommunizieren, indem sie nur IPv6 benutzen. Der Protokoll-

übersetzer wird verwendet, um IPv6-Header in IPv4-Header umzuwandeln. Wegen der Nutzung von privaten IPv4-Adressen ist jedoch wie bisher keine direkte Kommunikation mit externen IPv4-Partner-Endsystemen möglich.

6over4

Das interne Netzwerk verwendet Multicasting nicht, sodass dieses Verfahren in diesem Zusammenhang nicht relevant ist.

Dual Stack, 6to4 und Protokollübersetzer

Diese Mechanismen weisen dem Protokollübersetzer eine eindeutige IPv6-Adresse zu, indem die eindeutige IPv4-Adresse für den Protokollübersetzer und zugleich als ein Next Level Aggregator verwendet wird. Daduch erhält jedes Endsystem innerhalb der Organisation eine eindeutige IPv6-Adresse.

5.4.2.2 Lösungsvorschlag 1

Gegenwärtig wird privater Adressraum verwendet, sodass kein Problem mit limitierten IPv4-Adressen auftritt. Die einfachste Methode des Übergangs besteht in der Verwendung von Dual Stack auf allen Endsystemen. Bei dieser Lösung kann auf die komplexe Methode der Encapsulation verzichtet werden.

Vorgeschlagene Mechanismen bei diesem Lösungsvorschlag:

– Dual Stack
– Protokollübersetzer
– 6to4 (optional)

Stufe 1: Hauptbüro

Begonnen wird mit der Umstellung im Londoner Hauptbüro, da hier das Verkehrsaufkommen am höchsten ist. Es ist wesentlich, hier zuerst auf IPv6 aufzurüsten, damit die Kommunikation zu den regionalen Standorten funktioniert. Die Reihenfolge der Durchführung innerhalb des Hauptbüros wird detailliert in den folgenden Abschnitten gezeigt.

Default-Router R1

Der Default-Router verbindet das Hauptbüro mit allen regionalen Büros. Ein Software-Upgrade wird benötigt, um den Default-Router in die Lage zu versetzen, als Dual Stack-Router zu operieren. Der Router wird IPv6 als unabhängiges Protokoll behandeln, so dass RIPv6 aktiviert und für IPv6 konfiguriert werden muss.

DHCP-Server

Die Installation eines DHCP-Servers hängt davon ab, ob „Statefull“ Autokonfiguration zum Einsatz kommen soll. In diesem Fall muss der Server auf Dual Stack hochgerüstet werden:

– um die Zuweisung von IPv4-Adressen aus dem privaten Adressraum zu erlauben und
– um Statefull IPv6-Adressen zuzulassen.

DNS-Server

Der DNS-Server muss auf IPv6 hochgerüstet werden.

Proxy Server / Protokollübersetzer

Der Proxy Server bzw. Protokollübersetzer muss bilingual ausgelegt sein. Er markiert den Punkt, an dem im Netzwerk die Umsetzung zwischen IPv6 und IPv4 stattfindet. Der Firewall benötigt keine neue Konfiguration, solange die gesendeten und empfangenen Daten IPv4-Format haben und der ISP nicht zu IPv6 übergeht.

Mail- und File-Server

Der Mail-Server und alle File-Server müssen auf IPv6 hochgerüstet werden, wenn die obengenannten Umstellungen durchgeführt worden sind. Der Mail-Server erfordert unter Umständen zusätzlichen Konfigurationsaufwand.

Workstation

Sobald alle oben beschriebenen Umstellungen durchgeführt worden sind, können die Workstations auf IPv6 hochgerüstet werden. Diese Hochrüstung kann in beliebiger Reihenfolge und ohne Zeitlimit erfolgen.

Stufe 2: Londoner Regionalbüros

Nachdem das Hauptbüro hochgerüstet worden ist, können die Regionalbüros in London in Angriff genommen werden. Die Reihenfolge, in der die Regionalbüros umgestellt werden, ist unerheblich. Wichtig ist es, die in Stufe 1 aufgezeigte Vorgehensweise stets einzuhalten:

1. Default-Router
2. DHCP-Server
3. DNS-Server
4. weitere Server, z.B. Mail-Server
5. Workstation

Stufe 3: Weitere Regionalbüros

Nach den Londoner Regionalbüros können die regionalen Büros in den anderen Standorten hochgerüstet werden und zwar in der Reihenfolge: Birmingham - Manchester - Glasgow - Edinburgh. Diese Reihenfolge muss nicht zwingend eingehalten werden, es ist jedoch zu beachten, dass, falls Glasgow vor Machester und Birmingham hochgerüstet würde, Tunnels von Glasgows Router zum Router im Hauptbüro konfiguriert werden müssen. Die Vorgehensweise in jedem Büro hat in Übereinstimmung mit Stufe 1 und 2 zu erfolgen.

Stufe 4: Abschließende Stufe

Nachdem alle Endsysteme innerhalb der Organisation nach IPv6 hochgerüstet sind, kann die IPv4-Komponente in jedem Endsystem deaktiviert werden, sodass nur noch IPv6-Verkehr im Netzwerk erlaubt ist. Die Deaktivierung muss in umgekehrter Reihenfolge zur bisherigen Reihenfolge ablaufen, d.h. IPv4 muss zuerst an den Workstations und zuletzt am Router deaktiviert werden. Der Protokollübersetzer an der Grenze wird alle IPv6-Header in IPv4-Header umwandeln und umgekehrt. Für den Protokollübersetzer ist die Umwandlung von IPv4 nach IPv6 im Allgemeinen schwierig, besonders wenn eine Kommunikationsbeziehung von Extern eingeleitet wird. Im vorliegenden Fall wird nur der ISP-Server die externe Kommunikation anregen, sobald SMTP-Verkehr gesendet wird. Der Protokollübersetzer, der ein Anwendungsprotokoll-Übersetzer sein muss, kann an der Grenze des Netzwerks SMTP-Verkehr entdecken und zum korrekten Endsystem weiterleiten.

6to4 Option

Der 6to4-Mechanismus kann in Verbindung mit dem Protokollübersetzer verwendet werden. Die dem Protokollübersetzer eindeutig zugeordnete IPv4-Adresse wird dem Next Level Aggregator-Feld zugewiesen und so ein globales eindeutiges IPv6-Präfix schaffen. Dies ermöglicht allen Endsystemen innerhalb der Organisation eine global eindeutige IPv6-Adresse, und der NAT kann entweder IPv4- oder IPv6-Pakete empfangen.

5.4.2.3 Lösungsvorschlag 2

Wenn die Einführung von IPv6 innerhalb der Organisation oder die Anpassung von IPv4-Anwendungen für die IPv6-Unterstützung zu aufwändig ist, gibt es einen anderen Lösungsvorschlag. In diesem Fall kann der NAT zu einem Protokollübersetzer hochgerüstet werden, der den externen IPv6-Verkehr unterstützt und die gegenwärtige interne Infrastruktur unverändert belässt. Dadurch ergibt sich für das IPv4-Endsystem das Problem, zu ermitteln, wie etwas an eine IPv6-Adresse außerhalb der Organisation geschickt werden kann. Wenn der gesamte externe Verkehr zu der ISP-Adresse geschickt werden soll, muss der Protokollübersetzer so konfiguriert werden, dass der externe Verkehr über diese IPv6-Adresse abgewickelt wird. Die Endsysteme müssen manuell so konfiguriert werden, dass alle externen Daten zu einer bestimmten IPv4-Adresse geschickt werden. Dazu müssen Router kon-

figuriert werden, die die externen Daten zum Protokollübersetzer schicken. Dieser Lösungsansatz ist in Bezug auf langfristige Verwaltung und Instandhaltung so komplex, dass es wesentlich einfacher ist, das Netzwerk auf IPv6 umzustellen.

5.4.3 Die Einführung von IPv6 in eine ISP-Umgebung

Das Netzwerk eines Internet Service Providers besteht mindestens aus folgenden drei Hauptbereichen:

- Kernnetzwerk,
- Verbindungen zu anderen ISPs und
- kundenspezifisches Zugangsnetzwerk

Die nächsten beiden Abschnitte erklären, wie ein ISP IPv6 in diese Bereiche einführen kann.

Für jeden Bereich müssen zunächst einige vorbereitende Schritte vorgenommen werden:

- Anforderung von IPv6-Adressraums
- Registrierung der IPv6-Standorte und des Routings
- Einrichten des DNS

5.4.3.1 Einführung von IPv6 im Kernnetzwerk

Die Einführung von IPv6 im Kernnetzwerk ist nicht zwingend erforderlich. Ein ISP kann IPv6 auch über eine existierende IPv4-Infrastruktur betreiben. Entscheidet der ISP aber, IPv6 in seinem Kernnetzwerk einzuführen, so kann dies auf verschiedene Arten geschehen.

Der ISP hat die Möglichkeit, reine IPv6-Router oder getrennte Dual Stack-Router im Kernnetzwerk zu installieren. Diese werden durch vorbestimmte Leitungen zusammengeschlossen (ATM, PVCs, Leased Lines etc.) oder, wenn es Dual Stack-Router sind, durch IPv6 in IPv4-Tunnels über die existierende IPv4-Kern-Infrastruktur. Das Routing kann so eingerichtet werden, dass IPv4-Pakete über die alte IPv4-Infrastruktur und IPv6-Pakete über die neue IPv6-Infrastruktur geleitet werden. Wenn die Dual Stack-Router sich für die Verwendung im Kern-Netzwerk als stabil genug erweisen, liegen die Dinge einfacher. Der ISP hat dann die Möglichkeit, die Kernrouter als Dual Stack-Router zu konfigurieren, die sowohl IPv4- als auch IPv6-Pakete transportieren.

Als Nächstes sollte eine Verbindung zum globalen IPv6-Netzwerk erstellt werden, was durch einen direkten IPv6-Anschluss oder durch einige Tunneling-Mechanismen geschehen kann. Wenn sowohl der Kern des Netzwerks IPv6 als auch der andere ISP IPv6 unterstützen, kann eine direkte Verbindung verwendet werden, um IPv6-Pakete zu transportieren. Gibt es keine direkte IPv6-Verbindung, so muss ein Tunneling-Mechanismus verwendet werden, um das globale IPv6-Netzwerk zu erreichen. Es kann automatisches Tunneling verwendet werden (6to4). Ein ISP kann entscheiden, einen oder mehrere Router am

Rand seines Netzwerks aufzustellen, die als 6to4-Gateways fungieren. Das versetzt andere IPv6-Inseln in die Lage, den ISP durch 6to4-Tunneling zu erreichen. Eine Alternative zu dynamischen Tunneln stellen die auch im 6Bone verwendeten statischen Tunnel dar.

5.4.3.2 Einführung von IPv6 im Kundenzugangsnetzwerk

Das Kundennetzwerk besteht aus Anruf- bzw. Wählverbindungen und festen Verbindungen, die mit einem Zugangsrouter verbunden sind. Es gibt zwei Möglichkeiten, IPv6 einzuführen:

1. die Hochrüstung der Zugangsrouter zu Dual Stack-Routern. Die Dual Stack-Router verbinden sowohl IPv4- als auch IPv6-Router.
2. Es werden von den IPv4-Routern getrennte IPv6- oder Dual Stack-Router installiert. IPv4-Kunden nehmen über die alten IPv4-Zugangsrouter Verbindung auf, während IPv6-Kunden über die neuen Zugangsrouter Zugang finden.

Diese IPv6-Zugangsrouter müssen mit dem globalen IPv6-Netzwerk verbunden werden. Sollte der Kern IPv6 nicht unterstützen, so ist einer der im Abschnitt „Hilfsmittel in Systemlösungen“ auf Seite 54 ff beschriebenen Übergangsmechanismen zu nutzen. Es kann beispielsweise automatisches Tunneling (6to4) erfolgen. Eine Alternative zum Gebrauch dynamischer Tunnel stellt der Gebrauch statisch generierter Tunnel dar. Unterstützt das Kernnetzwerk IPv6, so können die Zugangsrouter mit dem nächstliegenden IPv6-Kernrouter durch IPv4- / IPv6-Verbindungen oder durch Tunneling über IPv4 verbunden werden.

Unter der Voraussetzung, dass der Kunde eine reine IPv6-Umgebung besitzt, kann der ISP auch Übergangsmechanismen zur Verfügung zu stellen, die den Kunden in die Lage versetzen, reine IPv4-Endsysteme zu erreichen. Dies kann der ISP z.B. durch eine NAT-PT Installation realisieren.

5.4.4 Internet-Datenaustauschpunkte

Basierend auf der Adressraumverteilung können zwei Modelle zum Einrichten eines IPv6-Internet-Datenaustauschpunktes unterschieden werden.

1. Das erste, eher traditionelle Modell ist in der IPv4-Welt gebräuchlich. In diesem Modell organisiert jeder ISP, der sich an einen Internet-Datenaustauschpunkt anschließt, seinen eigenen IPv6-Adressraum unabhängig von den anderen ISPs. Das Präfix für diesen Adressraum wird zwischen den ISPs ausgetauscht.
2. Beim zweiten Modell handelt es sich um ein Adressmodell, bei dem der Internet-Datenaustauschpunkt als Adressraum-Provider fungiert. Bei diesem Modell erhält der Internet-Datenaustauschpunkt einen Top Level Aggregator und kann von diesem Top Level Aggregator-Adressraum den verbundenen ISPs Next Level Aggregators zuweisen. Um eine Verbindung zum globalen Internet zu erhalten, muss sich der Betreiber des Internet-Datenaustauschpunktes sich mit einem oder mehreren globalen Transit Providern (Top Level Aggregator ISPs), die mit dem Internet-Datenaustauschpunkt verbunden sind, über den globalen Durchgang arrangieren. Dies impliziert, dass der Internet-Datenaustauschpunkt den Übergang für alle angeschlossenen ISPs durchführt, die den dem Internet-Datenaustauschpunkt zugewiesenen Adressraum nutzen. Dieses Szenario erfordert für den Betreiber des Internet-Datenaustauschpunktes ein vollkommen anderes Geschäftsmodell als in Modell 1.

Die beiden oben vorgestellten Modelle erfordern die im Folgenden beschriebenen Eingriffe durch den Internet-Datenaustauschpunkt Operator und/oder die angeschlossenen ISPs.

5.4.4.1 Modell 1

Der Betreiber des Internet-Datenaustauschpunktes fordert einen Next Level Aggregator an und erhält einen weltweit eindeutigen Adressraum. Das kann ein Next Level Aggregator von einem Transit Provider (Top Level Aggregator Provider) sein, der die Verbindung für die Internet-Datenaustauschpunkt-Infrastruktur anbietet. Ein globaler Präfix wird als Next Hop Attribut in BGP4 (BGP4-IPv6) bevorzugt.

- Infrastruktur der Adressierung auf dem Internet-Datenaustauschpunkt

  Aus dem Adressraum werden die Router-Interfaces, die mit der Internet-Datenaustauschpunkt Infrastruktur verbunden sind, mit Adressen versorgt.

- Update der IPv6-Registrierung

  Die Standorte, Zuweisungen und Route-Objekte werden registriert.

- BGP Ankündigungen

  ISPs, die mit dem Internet-Datenaustauschpunkt verbunden sind, geben ihren Adressraum bekannt, der unabhängig von dem des Internet-Datenaustauschpunktes ist.

5.4.4.2 Modell 2

Der Betreiber des Internet-Datenaustauschpunktes fordert von seinen regionalen Registrierungsorganisation einen (Sub-)Top Level Aggregator an. Kunden des Internet-Datenaustauschpunktes bekommen den höchsten Next Level Aggregator aus diesem (Sub-)Top Level Aggregator.

- Infrastruktur der Adressierung auf dem Internet-Datenaustauschpunkt.

  Aus dem Adressraum werden die Router-Interfaces, die mit der Infrastruktur des Internet-Datenaustauschpunktes verbunden sind, mit Adressen versorgt.

- Update der IPv6-Registrierung

  Die Standorte, Zuweisungen und Route-Objekte werden registriert.

- Der Betreiber des Internet-Datenaustauschpunktes schließt globale Transit ISPs (Top Level Aggregator ISPs) zusammen.

  Der Internet-Datenaustauschpunkt sollte verschiedene Top Level Aggregator-ISPs zusammenschließen, die Verbindungen zum globalen IPv6-Netzwerk anbieten werden. Solche Top Level Aggregator ISPs müssen den Verkehr aller am Internet-Datenaustauschpunkt angeschlossenen Kunden weiterleiten.

- BGP Ankündigungen

  Die Transit Provider des Internet-Datenaustauschpunkt-Adressraums geben die (Sub-)Top Level Aggregators vom Internet-Datenaustauschpunkt zum globalen IPv6-Netzwerk bekannt. Den Kunden des Internet-Datenaustauschpunktes teilen sie alle Präfixe mit, die von ihnen aus erreicht werden können und für die sie einen Vertrag mit dem Betreiber des Internet-Datenaustauschpunktes haben. Die Kunden (Next Level Aggregator ISPs) bekommen den nächst höheren Next Level Aggregator vom Betreiber des Internet-Datenaustauschpunktes. Next Level Aggregator-ISPs geben ihre Next Level Aggregators den Top Level Aggregator-ISPs bekannt. Ebenso teilen die Next Level Aggregator-ISPs ihre Next Level Aggregators anderen Next Level Aggregator-ISPs des Internet-Datenaustauschpunkt mit, sofern eine gleichberechtigte Übereinkunft zwischen ihnen besteht.

5.4.5 Vermeidung eines NAT-Einsatzes

Man nimmt zum Beispiel den Fall zweier großer netzwerkabhängiger Organisationen, welcher auf eine Verschmelzung oder Akquisition zurückzuführen ist, oder eine neue Geschäftspartnerschaft. Außerdem unterstellt man, dass beide Unternehmen große, auf IPv4-basierende Netzwerke haben, die aus kleinen Anfängen erwachsen sind. Beide ursprünglichen Unternehmen haben eine große Zahl privater IPv4-Adressen, die nicht notwendigerweise eindeutig im gegenwärtigen globalen IPv4-Adressraum sind. Diese beiden nicht eindeutigen Adressräume zu vereinheitlichen, würde eine kostspielige Umnummerierung und

Restrukturierung von Routern, Endsystemadressen, Domänen, Bereichen, externen Routing-Protokollen usw. bedeuten. Dieses Szenario ist in der gegenwärtigen wirtschaftlichen Situation häufig, nicht nur bei Fusionen und Unternehmensübernahmen, sondern ebenso bei großen Outsourcing-Projekten, wo viele Endsysteme des Auftraggebers vom Outsourcing-Partner in eine bestehende Struktur der Unternehmensadressen integriert werden müssen. Für solche Situationen bietet IPv6 eine passende Lösung.

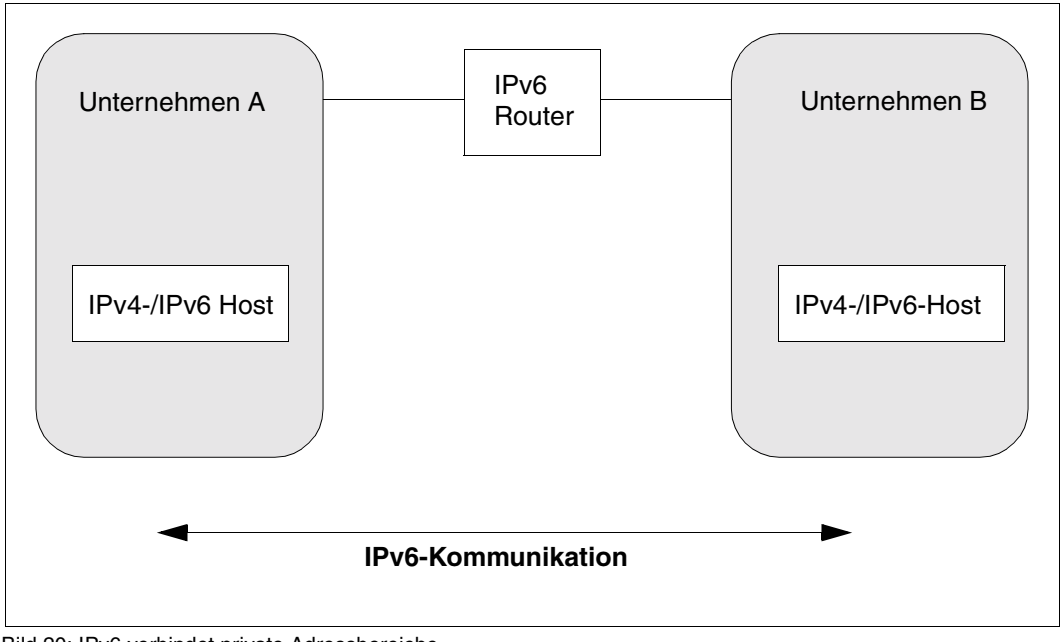

Bild 20: IPv6 verbindet private Adressbereiche

Die Aufgabe, zwei Unternehmensnetzwerke logisch in eine einzige autonome Domäne zu verschmelzen, kann teuer und störend für den geordneten Betrieb der Unternehmensnetzwerke sein. Um die Kosten und die Störung einer umfassenden Umnummerierung zu vermeiden, könnten die Unternehmen versucht sein, sich für die Platzhalterlösung eines Network Address Translators (NAT) zu entscheiden. Im Fusionen- und Unternehmensübernahme-Szenario könnte es ein NAT den beiden Unternehmen erlauben, ihre privaten Adressen mehr oder weniger unverändert zu belassen. Um dies zu erreichen, muss ein NAT die Adressenübersetzung in Real Time für alle Pakete durchführen, die zwischen den beiden Organisationen ausgetauscht werden. Leider beinhaltet diese Lösung alle im Zusammenhang mit den NATs schon angesprochen Probleme, einschließlich Performanceproblemen, Mangel an Skalierbarkeit und Standards sowie unzureichende Connectivity zwischen den Endsystemen im neuen Unternehmen und dem Internet.

Im Gegensatz zum NAT fügt IPv6 die beiden physikalischen Netzwerke (siehe Bild 18) nahtlos aneinander. Angenommen, die beiden ursprünglich unabhängigen Unternehmen sind als Unternehmen A und B bekannt, besteht der erste Schritt darin, zu bestimmen, welche Endsysteme Zugriff auf beiden Seiten der neuen Organisation brauchen. Diese Endsysteme werden mit Dual IPv4-/IPv6-Stacks ausgestattet, die ihnen die Verbindung zu ihrem originalen IPv4-Netzwerk erlauben, während sie ebenso an einem neuen IPv6-Netzwerk teilnehmen, das mit Hilfe der existierenden physikalischen IPv4-Infrastruktur geschaffen wird.

Die Buchhaltungsabteilung des fusionierten Unternehmens wird oftmals finanzielle Anfragen an die Server haben, die den Buchhaltungsangestellten sowohl im Unternehmen A als auch B zugänglich sein müssen. Sowohl Server als auch die Client-Systeme der Buchhaltungsangestellten werden IPv6 benutzen, aber sie werden weiterhin ihre IPv4-Stacks behalten. Die IPv6-Sitzungen des Rechnungsbüros werden die existierenden lokalen und remote Links überqueren wie „just another protocol“. Dies erfordert keinen Wechsel zum physikalischen Netzwerk. Die einzige Anforderung an die IPv6-Verbindung besteht darin, dass Router, die an Rechnungsbüro-Nutzer angrenzen, aufgerüstet werden müssen, damit IPv6 läuft. Wo keine end-to-end IPv6-Verbindungen betrieben werden können, kann eine der IPv4-/IPv6-Tunneling-Techniken verwendet werden.

Im Zuge der weiteren Integration werden andere Bereiche der neu verschmolzenen Unternehmen ebenso IPv4-/IPv6-Hosts erhalten. Wenn neue Geschäftsbereiche und Workgroups hinzugefügt werden, können auch sie Dual Stack-Hosts erhalten oder in bestimmten Fällen reine IPv6-Hosts. Hosts, die via Internet mit der Außenwelt kommunizieren, erhalten am besten Dual Stack, um die Kompatibilität mit IPv4-Knoten außerhalb des Unternehmens zu gewährleisten. Für einige Hosts ist eine reine IPv6-Anbindung ausreichend, sofern sie nur den Zugang zu internen Servern und spezifischen Partnern außerhalb benötigen. Eine Migration zu IPv6 eröffnet die Gelegenheit zu einem Neustart, was Adresszuweisung und Routing-Protokollstrukturen betrifft. IPv6-Hosts und Router profitieren sofort von den IPv6-Vorteilen, wie z.B. Stateless Autoconfiguration, Verschlüsselung, Authentifizierung.

5.4.6 IPv6 von außen nach innen

Die wichtigsten Forderungen der angeschlossenen Nutzer an den Netzverbund konzentrieren sich auf Zugänge zu E-mail, WWW, Datenbanken und Anwendungsservern. In diesem Fall empfiehlt es sich, die IPv6-Hochrüstung in einzelnen, abgeschlossenen Workgroups und Abteilungen zu beginnen und die Hochrüstung der Backbone-Router danach allmählich einzuleiten. Die Entwicklung des IPv6-Protokolls ist stärker auf das interne Routing als auf hochgradiges Backbone Routing ausgelegt. Daher ist dies ein ausgezeichneter Weg für Unternehmen, einen effektiven Übergang zu IPv6 zu realisieren. Wie in Bild 19 dargestellt können unabhängige Workgroups ihre Clients und Server auf Dual Stack-Standard oder auf IPv6 bringen, wodurch Inseln mit IPv6-Funktionalität entstehen.

Wenn Routing-Protokolle wie OSPF und BGP für IPv6 erweitert sind, kann der Kern Backbone IPv6-Verbindungen unterstützen. Nachdem die ersten IPv6-Router an den richtigen Stellen installiert sind, kann es wünschenswert sein, IPv6-Inseln durch Router-zu-Router Tunnels zu verbinden. In diesem Fall werden ein oder mehrere Router in jeder Insel als Tunnel-Endpunkte konfiguriert. Wie in Bild 4 auf Seite 23 dargestellt, werden die Tunnel bei voller Nutzung der IPv6 128 bit Adressierung durch die Endsysteme so konfiguriert, dass die am Tunneling beteiligten Router die Adresse des Tunnel-Endpunktes kennen. Mit IPv4-kompatiblen IPv6-Adressen ist automatisches, nichtkonfiguriertes Tunneling möglich.

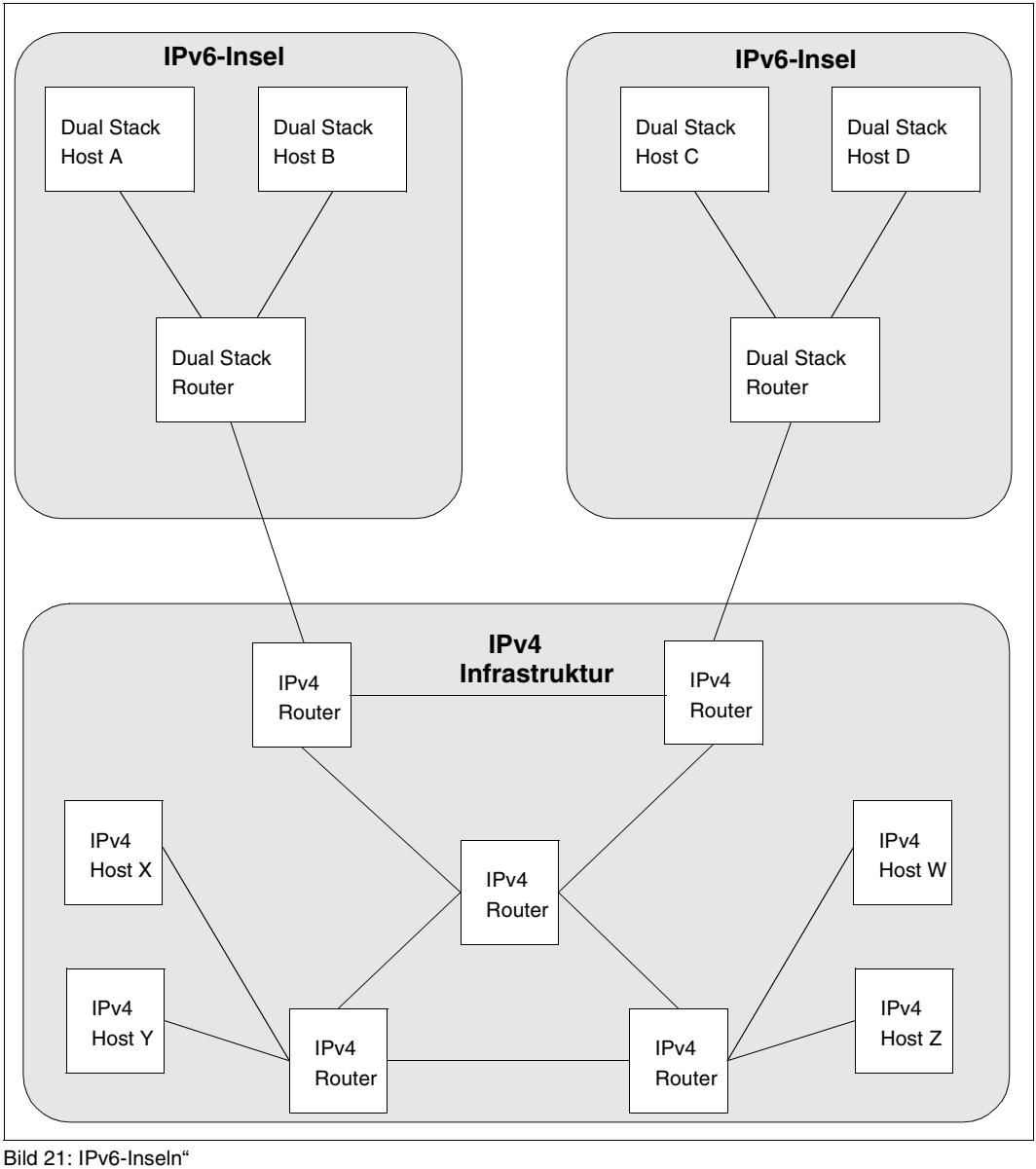

Bild 21: IPv6-Inseln“

Routing-Protokolle behandeln Tunnels als einen einzigen IPv6-Hop, auch wenn der Tunnel viele IPv4-Hops über eine Anzahl verschiedener Medien enthält. IPv6-Router, die mit OSPF laufen, können Link-state Reachability Advertisements durch Tunnels verbreiten, genauso wie sie das über konventionelle point-to-point Links machen würden: In der IPv6-

Umgebung kann OSPF bewirken, dass jeder Tunnel säuberlich innerhalb der Topologie verwendet wird. Generell treffen Router Entscheidungen zur Weiterleitung von Paketen, in der Tunnelumgebung genauso wie in reinen IPv6-Netzwerken. Die zugrundeliegenden IPv4-Verbindungen sind für die IPv6-Routing-Protokolle nicht sichtbar.

5.4.7 Andere Mechanismen

Zusätzliche Mechanismen für den Übergang oder für IPv4-/IPv6-Koexistenz werden weiterhin diskutiert. IPv4-Multicast kann beispielsweise benutzt werden, um Neighbor Discovery durch isolierte IPv6-Knoten zu unterstützen. Es gibt verschiedene Ansätze zur Unterstützung von Transaktionen zwischen reinen IPv4- und IPv6-Knoten, die keine IPv4-kompatiblen Adressen haben.

IETF Mitglieder verwenden intensive Anstrengungen sowohl auf den Übergang als auch auf die grundlegende IPv6-Protokollspezifikation. Die Kombination von Tunnels, kompatiblen Adressen, Dual Stack Endsystemen und Routern gibt den Administratoren Spielraum für Flexibilität und Interoperabilität bei der Einführung von IPv6. Übergangsmechanismen erlauben den Unternehmen, auf gegenwärtigen IPv4-Netzwerken die Vorteile der höher entwickelten IPv6-Features zu nutzen.

6 Einsatz von IPv6 in BS2000/OSD

In diesem Kapitel wird auf BS2000/OSD-Spezifika bei der Einführung von IPv6 in einem Netzwerk eingegangen, und es werden konkrete Hilfestellungen beim Einsatz der in IPv6 Stufe 1 realisierten Funktionalität gegeben.

6.1 SOCKETS-Anwendungen

Die folgenden Abschnitte stellen die für den IPv6-Betrieb notwendigen Änderungen an der SOCKETS-Schnittstelle vor. Beachten Sie in diesem Zusammenhang auch bitte das Handbuch zu „SOCKETS(BS2000) V2.0“.

6.1.1 Voraussetzungen

Für die Nutzung von IPv6 als Netzwerkprotokoll wird vorausgesetzt, dass *open*Net Server Version 2.0 eingesetzt ist und die SOCKETS-Anwendung mit SOCKETS(BS2000) V2.0 produziert wird. Außerdem muss das Subsystem SOC6 gestartet sein.

6.1.2 Umstellen von Funktionen

socket()

Beim Aufruf der Funktion *socket()* muss als *domain*-Parameter AF_INET6 statt AF_INET angegeben werden.

bind()

Beim Aufruf der Funktion *bind()* muss als *name*-Parameter eine Struktur vom Typ *sockaddr_in6* statt einer Struktur vom Typ *sockaddr_in* angegeben werden. Der Parameter *namelen* muss auf die Länge der Struktur *sockaddr_in6* gesetzt werden.

connect()

Beim Aufruf der Funktion *connect()* muss als *name*-Parameter eine Struktur vom Typ *sockaddr_in6* statt einer Struktur vom Typ *sockaddr_in* angegeben werden. Der Parameter *namelen* muss auf die Länge der Struktur *sockaddr_in6* gesetzt werden.

accept()

Die Funktion *accept()* liefert im Parameter *addr* eine Struktur vom Typ *sockaddr_in6* statt einer Struktur vom Typ *sockaddr_in* zurück. Der Parameter *addrlen* wird auf die Länge der Struktur *sockaddr_in6* gesetzt.

sendto()

Beim Aufruf der Funktion *sendto()* muss als *to*-Parameter eine Struktur vom Typ *sockaddr_in6* statt einer Struktur vom Typ *sockaddr_in* angegeben werden. Der Parameter *tolen* muss auf die Länge der Struktur *sockaddr_in6* gesetzt werden.

recvfrom()

Beim Aufruf der Funktion *recvfrom()* muss als *from*-Parameter eine Struktur vom Typ *sockaddr_in6* statt einer Struktur vom Typ *sockaddr_in* angegeben werden. Der Parameter *fromlen* muss auf die Länge der Struktur *sockaddr_in6* gesetzt werden.

gethostbyname()

Die Funktion *gethostbyname()* muss durch die Funktion *getipnodebyname()* ersetzt werden. Der von der Funktion *getipnodebyname()* angeforderte Speicher muss mit der Funktion *freehostent()* wieder frei gegeben werden.

gethostbyaddr()

Die Funktion *gethostbyaddr()* muss durch die Funktion *getipnodebyaddr()* ersetzt werden. Der von der Funktion *getipnodebyaddr()* angeforderte Speicher muss mit der Funktion *freehostent()* wieder frei gegeben werden.

getsockname()

Die Funktion *getsockname()* liefert im Parameter *name* eine Struktur vom Typ *sockaddr_in6* statt einer Struktur vom Typ *sockaddr_in* zurück.

getpeername()

Die Funktion *getpeername()* liefert im Parameter *name* eine Struktur vom Typ *sockaddr_in6* statt einer Struktur vom Typ *sockaddr_in* zurück.

inet_addr()

Die Funktion *inet_addr()* muss durch die Funktion *inet_pton()* ersetzt werden.

sinet_ntoa()

Die Funktion *inet_ntoa()* muss durch die Funktion *inet_ntop()* ersetzt werden.

htonl()

Der Makro *htonl()* zur Umsetzung einer IPv4-Adresse vom Hostformat ins Netzwerkformat entfällt bei der Nutzung von IPv6, da IPv6-Adressen immer im Netzwerkformat dargestellt werden.

ntohl()

Der Makro *ntohl()* zur Umsetzung einer IPv4-Adresse vom Netzwerkformat ins Hostformat entfällt bei der Nutzung von IPv6, da IPv6-Adressen immer im Netzwerkformat dargestellt werden.

Umstellung der Adress-Struktur

Die Struktur *sockaddr_in* muss durch die Struktur *sockaddr_in6* ersetzt werden.

Bei den einzelnen Strukturelementen gelten die folgenden Umsetzregeln:

`sin_family`	wird ersetzt durch	`sin6_familiy`	:= AF_INET6 statt AF_INET
`sin_port`	wird ersetzt durch	`sin6_port`	:= Portnummer wie bisher
`sin_addr`	wird ersetzt durch	`sin6_addr`	:= IPv6-Adresse statt IPv4-Adresse

Die IPv6-Adresse muss mit der Funktion *memcpy()* zugewiesen werden. Statt der INADDR_ANY-Adresse muss die INADDR6_ANY-Adresse verwendet werden.

6.1.3 Connectivity mit IPv4-only-Partnern

Die Kommunikation mit reinen IPv4-Partnern ist für IPv6 SOCKETS Anwendungen problemlos möglich. Durch sogenannte „mapped Adressen“ werden IPv4-Adressen syntaktisch wie IPv6-Adressen behandelt. Das SOC6 Subsystem erkennt bei den jeweiligen Funktionen ob es sich um mapped Adressen handelt und führt dann die jeweilige Funktion so aus als ob sie in der Adressfamilie AF_INET aufgerufen worden wäre.

6.1.4 Beispiele

Die beiden folgenden Beispiele zeigen wie SOCKETS-Anwendungen auf IPv6 umgestellt werden können.

Die geänderten bzw. neuen Codeteile sind **fett** dargestellt. Bei geänderten Codeteilen befindet sich darunter als Kommentar der ursprüngliche IPv4-spezifische Codeteil.

Beispiel 1: Verbindungsorientierter Server bei AF_INET6

```
#include <stdio.h>
#include <sys.types.h>
#include <sys.socket.h>
#include <netinet.in.h>
#include <netdb.h>

main(argc, argv)
int argc;
char *argv[];
{
#define TESTPORT 2222
int sock, length;
```

struct sockaddr_in6 server;

```
/* struct sockaddr_in server; */
```

struct in6_addr in6addr_any = IN6ADDR_ANY_INIT;

```
int msgsock;
char buf[1024];
int rval;
/* Socket erzeugen /
```

sock = socket(AF_INET6, SOCK_STREAM, 0);

```
/* sock = socket(AF_INET, SOCK_STREAM, 0); */
```

```
if (sock < 0)
{ perror("Create stream socket");
exit(1);
}

/* Dem Socket einen Namen zuordnen */

server.sin_family = AF_INET6;
/* server.sin_family = AF_INET; */

memcpy(server.sin6_addr.s6_addr, in6addr_any,16) ;
/* server.sin_addr.s_addr = htonl(INDDR_ANY) ; */

server.sin6_port = htons(TESTPORT);
/* server.sin_port = htons(TESTPORT); */

if (bind(sock, &server, sizeof (server) ) < 0)
{ perror("Bind stream socket");
exit(1);
}
/* Beginn mit der Annahme von Verbindungsanforderungen */
listen(sock, 5);
msgsock = accept(sock, 0, (int *)0);
if (msgsock == -1)
{ perror("Accept connection");
exit(1);
}
else do {
memset(buf, 0, sizeof buf);
if ((rval = recv(msgsock, buf, 1024, 0)) < 0)
{ perror("Reading stream message");
exit(1);
}
else if (rval == 0 )
fprintf(stderr, "Ending connection\n");
else
fprintf(stdout, "->%s\n", buf);
} while (rval != 0);
soc_close(msgsock);
soc_close(sock);
}
```

Beispiel 2: Verbindungsorientierter Client bei AF_INET6

```
#include <stdio.h>
#include <sys.types.h>
#include <sys.socket.h>
#include <netinet.in.h>
#include <netdb.h>
#include <sys.uio.h>
main(argc, argv)
int argc;
char *argv[];
{
#define TESTPORT 2222
#define DATA "Here's the message ..."
int sock, length;
int error_num;

struct sockaddr_in6 client;
/* struct sockaddr_in6 client; */

struct hostent *hp;

struct hostent *getipnodebyname();


char buf[1024];

/* Socket erzeugen */

sock = socket(AF_INET6, SOCK_STREAM, 0);
/* sock = socket(AF_INET6, SOCK_STREAM, 0); */

if (sock < 0)
{ perror("Create stream socket");
exit(1);
}
/* Ausfüllen der Adreß-Struktur */

client.sin6_family = AF_INET6;
/* client.sin_family = AF_INET; */

client.sin6_port = htons(TESTPORT);
/* client.sin_port = htons(TESTPORT); */
```

```
hp = getipnodebyname(argv[1], AF_INET6, 0, &error_num);
if ((hp = = 0) || (error_num != NETDB_SUCCESS))
/* hp = getipnodebyname(argv[1], AF_INET6, 0, &error_num); */
/* if ((hp = = 0) || (error_num != NETDB_SUCCESS)) */

{ fprintf(stderr,"%s: unknown host\n", argv[1]);
exit(1);
}

memcpy((char *) &client.sin6_addr, (char *)hp->h_addr, hp->h_length);
/* memcpy((char *) &client.sin_addr, (char *)hp->h_addr, hp->h_length); */

/* Verbindung starten */
if ( connect(sock, &client, sizeof(client) ) < 0 )
{ perror("Connect stream socket");
exit(1);
}

/* Auf den Socket schreiben */
if ( send(sock, DATA, sizeof DATA, 0) < 0)
{ perror("Write on stream socket");
exit(1);
}
soc_close(sock);
}
```

6.2 DCAM- und CMX-Anwendungen

Für DCAM- und CMX-Anwendungen ergeben sich keine Änderungen, wenn für die Kommunikation mit Partnersystemen IPv6 als Netzwerkprotokoll genutzt wird. Es ist jedoch erforderlich, dass auch das Partnersystem den NEA- bzw. ISO-Transportservice mit IPv6 als Netzwerkprotokoll unterstützt.

6.3 Administration

6.3.1 Konfigurationsmaßnahmen

Die Unterstützung von IPv6 und ICMPv6 ist eine in BCAM per BCOPTION IPV6=ON/OFF ein-/ausschaltbare Option.

Es werden ausschließlich Adressen gemäß RFC 2373 unterstützt.
Da in diese Adressen über den Interface-Identifier auch LAN-Adressen Eingang finden, wird empfohlen bei den eigenen LAN-/FDDI-Anschlüssen stets die eigenen LAN-Adressen zu definieren, um Mehrdeutigkeiten zu vermeiden.

Im Übrigen bestehen für die Definition von IPv6-Partnersystemen die gleichen Möglichkeiten, wie bei IPv4.

6.3.2 Autokonfiguration

Die automatische Generierung eigener IPv6-Adressen ist eine in BCAM per BCOPTION IPV6-AUTO-CONFIG=ON/OFF ein-/aus-schaltbare Option.
Falls diese Option eingeschaltet ist,

– generiert *open*Net Server eigene Link-lokale IPv6-Adressen

– generiert *open*Net Server eigene IPv4-kompatible Adressen

– wertet *open*Net Server von Routern versendete ICMPv6 Präfix-Optionen aus und erzeugt entsprechende Adressen.

Außerdem wird stets die Eindeutigkeit eigener IPv6-Adressen überprüft.

6.3.3 Tunneling

Für IPv4-kompatible IPv6-Adressen wirkt ein automatisches Tunneling.

Statische IPv4- bzw. IPv6-Tunnel müssen per MODIFIY-ROUTE ADD-IP-NET bzw. ADD-IPV6-NET eingetragen werden (siehe Handbuch „BCAM“ Band 1).

6.3.4 Konfigurationsbeispiel

Das folgende Konfigurationsbeispiel zeigt lokal und remote sowie über IPv4-Tunnel erreichbare IPv6-Systeme.

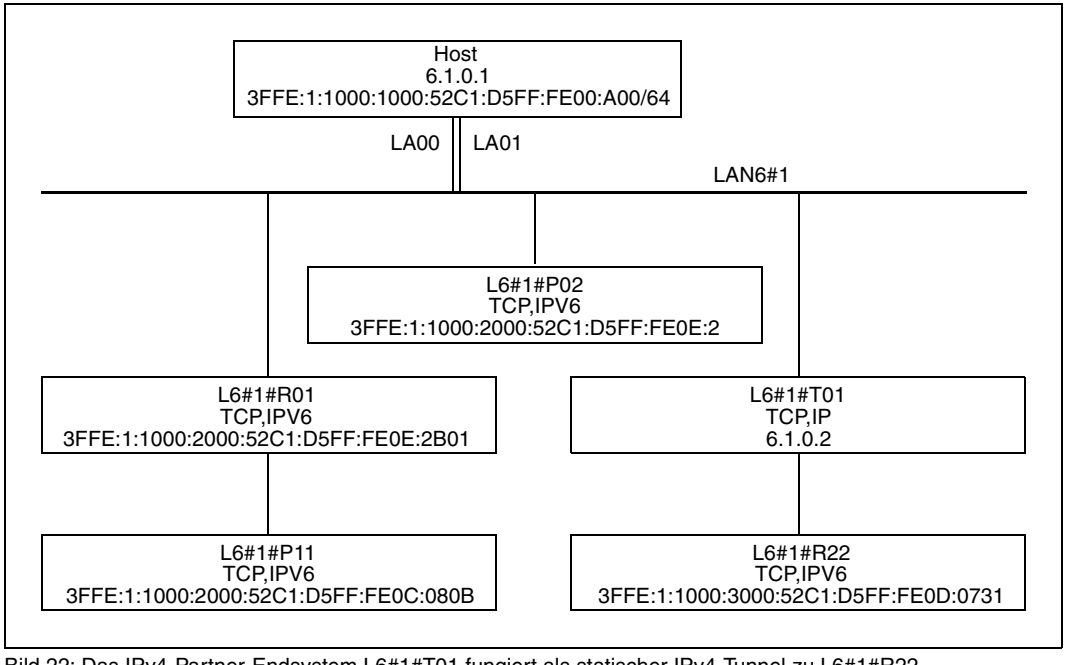

Bild 22: Das IPv4-Partner-Endsystem L6#1#T01 fungiert als statischer IPv4-Tunnel zu L6#1#R22

Statische Generierung mit KOGS-Makros:

```
*
*********************************************************************
* LAN **    LAN6#1                                                  *
*********************************************************************
*
  XLTNG UEPROZ=CSMACD,                                              *
        LTGNAM=LAN6#1,                                              *
        UEWEG=LAN,                                                  *
        DEVTYP=TRANSO,                                              *
        DEVMN=(LAOO,LAO1),                                          *
        LANADR=5OC1D5OOOAOO,                                        *
        IPV6ADR=3FFE:1:1OOO:1OOO:52C1:D5FF:FEOO:AOO/64,             *
        IPADR=OO6.OO1.OOO.OO1
*
  XKNOT KNOTNAM=LAN6#1
*
  XPRO  PROTYP=HOST/BCAM,                                           *
        PRONAM=L6#1#RO1,                                            *
        PROFIL=(TCP,IPV6),                                          *
        IPV6ADR=3FFE:1:1OOO:1OOO:52C1:D5FF:FEOE:2BO1,               *
        NAKNO=JA
*
  XPRO  PROTYP=HOST/BCAM,                                           *
        PRONAM=L6#1#P11,                                            *
        PROFIL=(TCP,IPV6),                                          *
        IPV6ADR=3FFE:1:1OOO:2OOO:52C1:D5FF:FEOC:O8OB,               *
        NAKNO=NEIN
*
  XPRO  PROTYP=HOST/BCAM,                                           *
        PRONAM=L6#1#PO2,                                            *
        PROFIL=(TCP,IPV6),                                          *
        IPV6ADR=3FFE:1:1OOO:1OOO:52C1:D5FF:FEOE:2,                  *
        NAKNO=JA
*
```

```
  XPRO  PROTYP=HOST/BCAM,                                          *
        PRONAM=L6#1#T01,                                           *
        PROFIL=(TCP,IP),                                           *
        IPADR=006.001.000.002,                                     *
        NAKNO=JA
*
  XPRO  PROTYP=HOST/BCAM,                                          *
        PRONAM=L6#1#R22,                                           *
        PROFIL=(TCP,IPV6),                                         *
        IPV6ADR=3FFE:1:1000:3000:52C1:D5FF:FE0D:0731,              *
        NAKNO=NEIN
*
  XEND
```

Dynamische Generierung mit BCIN-Kommandos:

```
/BCIN LAN6#1,GEN=LOCAL,DEV=(LA00,LA01),                            -
/            PROFIL=(,(IPV6,IP),CSMACD),IPADR=(6,1,0,1),           -
/            I6-ADDRESS='3FFE:1:1000:1000:52C1:D5FF:FE00:A00/64', -
/            LANADR=X‘50C1D5000A00’

/BCIN L6#1#R01,GEN=NODE,ROUTE=LAN6#1,                              -
/              PROFIL=(TCP,IPV6),                                  -
/              I6-ADDRESS=‘3FFE:1:1000:1000:52C1:D5FF:FE0E:2B01’

/BCIN L6#1#P11,GEN=REMOTE,ROUTE=L6#1#R01,PROFIL=(TCP,IPV6),        -
/              I6-ADDRESS=‘3FFE:1:1000:2000:52C1:D5FF:FE0C:080B’

/BCIN L6#1#P02,GEN=NODE,ROUTE=LAN6#1,                              -
/              PROFIL=(TCP,IPV6),                                  -
/              I6-ADDRESS=‘3FFE:1:1000:1000:52C1:D5FF:FE0E:2’

/BCIN L6#1#T01,GEN=NODE,ROUTE=LAN6#1,PROFIL=(TCP,IP)               -
/              IPADR=(6,1,0,2)

/BCIN L6#1#R22,GEN=REMOTE,ROUTE=L6#1#T01,PROFIL=(TCP,IPV6),        -
/              I6-ADDRESS=‘3FFE:1:1000:3000:52C1:D5FF:FE0D:0731’
```

Dynamische Generierung mit SDF-Kommandos:

```
/CREATE-NODE NODE-NAME=LAN6#1

/CREATE-LINE LINE-NAME=LAN6#1,-
/            IP-ADDRESS=6.1.0.1,-
/            IPV6-ADDRESS='3FFE:1:1000:1000:52C1:D5FF:FE00:A00/64', -
/            L2-PROTOCOL=*CSMACD(NODE-NAME=LAN6#1,                  -
/            WRITE-DEVICE=LA00,READ-DEVICE=LA01,                    -
/            LAN-ADDRESS=X'50C1D5000A00')


/CREATE-PROCESSOR PROCESSOR-NAME=L6#1#R01

/CREATE-ROUTE ROUTE-NAME=L6#1#R01,                                  -
/       PATH=*NODE(NODE-NAME=LAN6#1,                                -
/       L3-PROTOCOL=*IPV6(                                          -
/       IPV6-ADDRESS='3FFE:1:1000:1000:52C1:D5FF:FE0E:2B01'))

/CREATE-PROCESSOR PROCESSOR-NAME=L6#1#P11

/CREATE-ROUTE ROUTE-NAME=L6#1#P11,                                  -
/             PATH=*VIA-ROUTER(ROUTER-ROUTE-NAME=L6#1#R01,          -
/             L3-PROTOCOL=*IPV6(                                    -
/             IPV6-ADDRESS='3FFE:1:1000:2000:52C1:D5FF:FE0C:080B'))

/CREATE-PROCESSOR PROCESSOR-NAME=L6#1#P02

/CREATE-ROUTE ROUTE-NAME=L6#1#P02,                                  -
/             PATH=*NODE(NODE-NAME=LAN6#1,                          -
/             L3-PROTOCOL=*IPV6(                                    -
/             IPV6-ADDRESS='3FFE:1:1000:1000:52C1:D5FF:FE0E:2'))

/CREATE-PROCESSOR PROCESSOR-NAME=L6#1#T01

/CREATE-ROUTE ROUTE-NAME=L6#1#T01,                                  -
/             PATH=*NODE(NODE-NAME=LAN6#1,                          -
/             L3-PROTOCOL=*IP(IP-ADDRESS=6.1.0.2))

/CREATE-PROCESSOR PROCESSOR-NAME=L6#1#R22

/CREATE-ROUTE ROUTE-NAME=L6#1#R22,                                  -
/PATH=*VIA-TUNNEL(TUNNEL-ROUTE-NAME=L6#1#T01,                       -
/L3-PROTOCOL=*IPV6(                                                 -
/IPV6-ADDRESS='3FFE:1:1000:3000:52C1:D5FF:FE0D:0731'))
```

Automatische Endsystemaufnahme:

```
/BCOPTION AUTO-ES-CREATE=ON

/CREATE-NODE NODE-NAME=LAN6#1

/CREATE-LINE LINE-NAME=LAN6#1,                                          -
/            IP-ADDRESS=6.1.0.1,                                        -
/            IPV6-ADDRESS='3FFE:1:1000:1000:52C1:D5FF:FE00:A00/64', -
/            L2-PROTOCOL=*CSMACD(NODE-NAME=LAN6#1,                      -
/            WRITE-DEVICE=LA00,READ-DEVICE=LA01,                        -
/            LAN-ADDRESS=X'50C1D5000A00')

/CREATE-PROCESSOR PROCESSOR-NAME=L6#1#T01

/CREATE-ROUTE ROUTE-NAME=L6#1#T01,                                      -
/             PATH=*NODE(NODE-NAME=LAN6#1,                              -
/             L3-PROTOCOL=*IP(IP-ADDRESS=6.1.0.2))

/MODIFY-ADDRESS-ASSIGNMENT ROUTE-NAME=L6#1#T01,                         -
/               ADD-IPV6-NET='3FFE:1:1000:3000:/64'
```

Eintrag in die Processor-Datei:

```
L6#1#R01 IPV6 3FFE:1:1000:1000:52C1:D5FF:FE0E:2B01
L6#1#P11 IPV6 3FFE:1:1000:2000:52C1:D5FF:FE0C:080B
L6#1#P02 IPV6 3FFE:1:1000:1000:52C1:D5FF:FE0E:2
L6#1#R22 IPV6 3FFE:1:1000:3000:52C1:D5FF:FE0D:0731
```

Definitionsmaßnahmen bei Routern

In der Regel müssen die eigenen IPv6-Adressen eingetragen werden, z.B. zu versendende Präfix-Informationen, Router-Lifetime, usw.

Definitionsmaßnahmen bei Partner-Prozessoren

In der Regel müssen nur die eigenen IPv6-Adressen eingetragen werden.

6.4 Einschränkungen

Die folgenden Abschnitte zeigen auf, welche der in Kapitel 3 und 4 beschriebenen IPv6-Features in IPv6 Stufe 1 nicht bzw. eingeschränkt realisiert sind.

6.4.1 Path MTU Discovery

Die Funktion der Path MTU Discovery ist in der vorliegenden Version nicht realisiert. Dies bedeutet jedoch keine funktionelle Einschränkung im Betrieb von IPv6.

6.4.2 IPSEC

Da der Normungsprozess von IPSEC noch nicht abgeschlossen ist, wird die IPSEC-Funktionalität in dieser Version - IPv6 Stufe 1- nicht unterstützt.

6.4.3 Hardware

Bei der Verwendung des HNC-91849 kann es auf Grund der beschränkten Anzahl von Multicast-Adressen, die dieses Gerät unterstützt, zu Problemen bei der parallelen Nutzung von IPv6- und IPv4-Multicasting kommen.

6.4.4 Quality of Service

Der Normungsprozess zu Quality of Service ist noch nicht abgeschlossen, daher wird die Quality of Service Funktionalität in der hier beschriebenen Version noch nicht unterstützt.

7 Anhang

Dieser Anhang informiert über folgende Themen:

– IPv6-Adressregeln
– IPv6 und DNS

7.1 Anhang 1: IPv6-Adressregeln

7.1.1 IPv6-Adresszuweisung

Die meisten Übertragungsmechanismen erfordern Dual Stack Systeme und global routbare IPv6-Adressen ebenso wie gobal routebare IPv4-Adressen, obgleich manchmal private IPv4-Adressen [RFC 1918] ausreichen würden. Aber um die Kommunikation zwischen IPv4- und IPv6-Endsystemen über das Internet zu erlauben, wird mindestens eine global eindeutige IPv4-Adresse gebraucht. Global eindeutige IPv4-Adressen werden von einer der regionalen Internet Registries (IR), lokalen Internet Registries (LIR) oder einem Internet Service Provider (ISP) vergeben. Ohne spezielle Registrierung kann eine IPv6-Website lokale Website-Adressen verwenden, welche der IPv4 Privatadresse [RFC 1918] ähnlich sind. Aber lokale Websites erlauben keine Kommunikation über das Internet. Dazu braucht man eine global routebare IPv6-Adresse. Die meisten Websites bekommen ein /48-Präfix mit 16 Bit zum Subnetting und 64 Bit zum Interface ID Addressing. Das bedeutet, dass 65536 Subnets definiert werden und in jedem Subnet beinahe 20 Trillionen Endsysteme adressiert werden können.

0	**48**	**64**	**127**
prefix	subnet	Interface ID	

Zur Zeit exisiert das 6Bone, ein experimentelles Netzwerk, welches auf Basis von IPv6 läuft. Für dieses Netzwerk ist ein Teil des Adressraums zugewiesen, der sogenannte Pseudo TLA (pTLA) 3ffe::/16. Provider pTLAs werden durch den 6Bone zugewiesen. NLAs werden in der Reihenfolge durch diese Organisationen zugewiesen, die die pTLA-Zuweisung durch den 6Bone erhalten haben.

Wie man IPv6-Adressraum erhält

Man kann IPv6-Adressen von denselben Organisationen erhalten, die auch IPv4-Adressen vergeben. Internet Registries vergeben einen Teil ihres IPv6-Adressraums an lokale Internet Registries, die daraufhin Teile des Adressraums an ihre Kunden weitergeben. Die kleinste Zuweisung, die ein Kunde erhalten kann, ist ein /48 Präfix. Ein Unterschied zwischen IPv4 und IPv6 besteht darin, dass einer der Hauptvorteile der IPv6 Zuweisung die Route Aggregation ist, d.h. die Anzahl der Präfixe zu minimieren, die im default-freien Kern des Internet angezeigt werden müssen.

Die regionalen Internet Registries benutzen einen Mechanismus [IRALLOC] , um TLAs an ISPs zuzuweisen. Von ISPs, die am 6Bone teilnehmen, kann eine spezielle Vorqualifikationsprozedur [6PAPA] benutzt werden.

Die ISPs können in zwei Kategorien unterteilt werden: diejenigen ISPs, die eine (sub-)TLA von ihrer regionalen Internet Registry erhalten können und die ISPs, die keine (sub-)TLA bekommen. Im Folgenden wird die erste Kategorie als TLA-ISPs und die zweite als NLA-ISPs bezeichnet, weil sie von ihrem Upstream-Provider ein NLA bekommen.

TLA-ISPs bekommen zunächst ein sub-TLA und können später volle /(29+n) Präfixe verwenden, wobei n ($0 \leq n \leq 19$) die Zahl der Bit ist, die die NLA-ISPs [RFC 2374] identifizieren.

3	**13**	**13**	**19**		**16**	**64 bit**
FP	TLA ID	sub-TLA	NLA ID		SLA ID	Interface ID
TLA ISP Präfix			NLA ISP Identifier (n bit)	Bit für NLA ISP		

Ein NLA-ISP enthält ein Präfix zwischen /29 und /48. Es nutzt die verbleibenden Bit in der NLA ID, um seine Kunden zu identifizieren. Diese Kunden erhalten ein /48 Präfix.

3	**13**	**13**	**19**		**16**	**64 bit**
FP	TLA ID	sub-TLA	NLA ID		SLA ID	Interface ID
TLA ISP Präfix			end customer site identifier (19 - n bit)	Bit für NLA ISP		

7.1.2 IPv6-Registrierungsprobleme

In der gegenwärtigen IPv4-Welt werden Adressraum-Zuweisungen in verschiedenen Datenbanken verzeichnet, die von regionalen IRs verwaltet werden. Autonome System (AS) Information und Routingverhalten werden in der verteilten Internet Routing Registry-Datenbank (IRR) aufgezeichnet. Die IRs, LIRs und ISPs sollen Adressraum-Anweisungen und Zuweisungen, Kontaktpersonen, AS Zahlen, Routingverhalten und andere nützliche Daten zur Netzwerkverwaltung in den verschiedenen Datenbanken festhalten.

Für die 6Bone-Gemeinde wurde eine spezielle IPv6 Registrierungsdatenbank auf dem Whois-Server „whois.6bone.net“ geschaffen. Dies ist eine besondere Version der RIPE Database Software und wird als „6bone database“ bezeichnet. Diese Datenbank hat spezielle Ziele, die „inet6num:“ für zugewiesene IPv6-Präfixe und „ipv6-site:“, um spezifische Informationen über mit dem 6Bone verbundenen Sites festzuhalten, wie z.B. konfigurierte Tunnels und Origin AS. Auf der IPv6-Site können Objekte gefunden werden, nämlich IPv6-Anwendungen, die auf dieser spezifischen Seite unterstützt werden.

Die Datenbank kann durch die Anwendung eines modifizierten Whois Clients oder des Web-basierten Whois-Service unter http://www.6bone.net/whois.html befragt werden. Zur Zeit unterstützt nur die 6Bone-Datenbank diese speziellen IPv6-Objekte. Gegenwärtig gibt es keine Datenbank-Objekte, um IPv6-Routingverhalten zu registrieren.

Wenn die regionalen IRs damit anfangen, (sub-)TLAs die zugewiesenen und bestimmten IPv6 Präfixe zuzuweisen, müssen Routingverhalten etc. registriert werden. Derzeit ist noch unklar, wie die IPv6-Registrierung endgültig vor sich gehen wird.

Beispiel für den IPv6-Gebrauch

Die Standorte erhalten ein /48-Präfix. Ein Beispiel, wie solch ein /48 genutzt wird, verdeutlicht folgendes Bild. In diesem Beispiel ist dem Standort 3FFE:1234:5678::/48 zugewiesen.

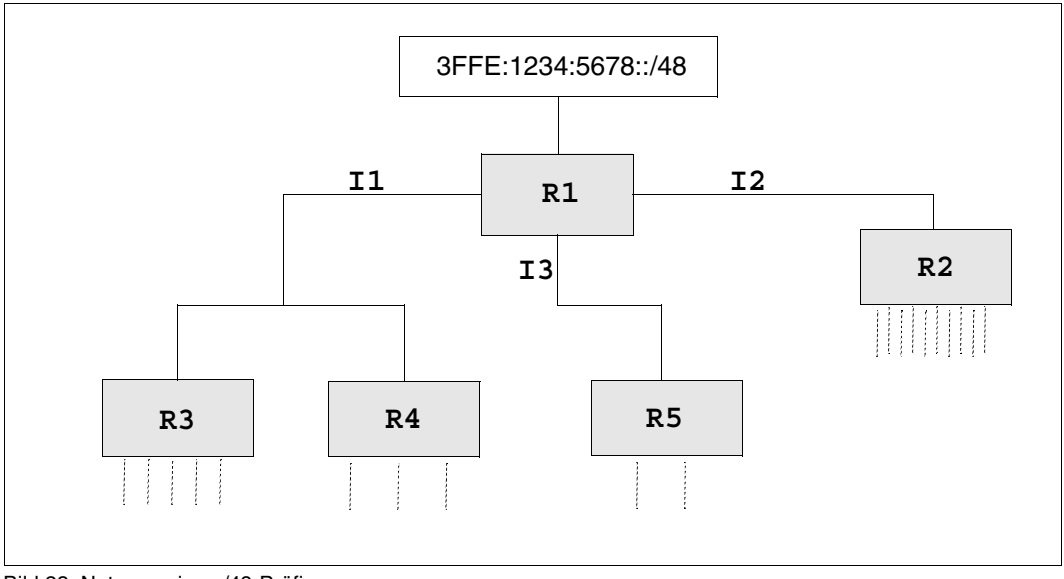

Bild 23: Nutzung eines /48-Präfix

R[1-5] bezeichnen Router, I[1-3] stellen Interfaces von R1 dar. Angenommen die erwartete Zahl von Endsystemen an den Links beträgt:

Router	Immediate	1. Jahr	2. Jahr
R2	34	50	70
R3	19	20	25
R4	9	10	15
R5	3	5	10

Dann könnte ein Zahlenplan wie folgt aussehen. Auf R1 werden die folgenden Präfixe auf den Interfaces genutzt:

```
I1        3FFE:1234:5678:2000::/50
I2        3FFE:1234:5678:0000::/49
I3        3FFE:1234:5678:2300::/50
```

Anfänglich bekommt R2 256 /64s, R3 bekommt 48 /64s, R4 32 /64s und R5 16 /64s.

```
3FFE:1234:5678:0000::/50
-------------------------
3FFE:1234:5678:0000::/49        I2
3FFE:1234:5678:1000::/49        frei
3FFE:1234:5678:2000::/49        I1 + I3
3FFE:1234:5678:3000::/49        frei
.....                           ...

3FFE:1234:5678:F000::/49        frei
3FFE:1234:5678:0000::/49
-------------------------

3FFE:1234:5678:0000::/64        Interfaces von R2
.....                           ...
3FFE:1234:5678:00FF::/64        Interfaces von R2
3FFE:1234:5678:0100::/64        reserviert für R2
.....                           ...
3FFE:1234:5678:02FF::/64        reserviert für R2
3FFE:1234:5678:0300::/64        frei
.....                           ...

3FFE:1234:5678:2000::/49
-------------------------

3FFE:1234:5678:2000::/50        I1
3FFE:1234:5678:2100::/50        reserviert für I1
3FFE:1234:5678:2200::/50        reserviert für I1
3FFE:1234:5678:2300::/50        I3
3FFE:1234:5678:2400::/50        reserviert für I3
3FFE:1234:5678:2500::/50        reserviert für I3
3FFE:1234:5678:2600::/50        frei
.....                           ...
3FFE:1234:5678:2F00::/50        frei

3FFE:1234:5678:2000::/50
-------------------------

3FFE:1234:5678:2000::/64        Interfaces von R3
.....                           ...
3FFE:1234:5678:202F::/64        Interfaces von R3
3FFE:1234:5678:2030::/64        reserviert für R3
.....                           ...
3FFE:1234:5678:204F::/64        reserviert für R3

3FFE:1234:5678:2050::/64        Interfaces von R4
.....                           ...
3FFE:1234:5678:206F::/64        Interfaces von R4
```

```
3FFE:1234:5678:207O::/64        reserviert für R4
.....                           ...
3FFE:1234:5678:209F::/64        reserviert für R4
3FFE:1234:5678:20A0::/64        frei
.....                           ...
3FFE:1234:5678:20FF::/64        frei


3FFE:1234:5678:230O::/50
------------------------

3FFE:1234:5678:230O::/64        Interfaces von R5

.....                           ...
3FFE:1234:5678:23OF::/64        Interfaces von R5
3FFE:1234:5678:2310::/64        reserviert für R5
.....                           ...
3FFE:1234:5678:231F::/64        reserviert für R5
3FFE:1234:5678:232O::/64        frei
.....                           ...
3FFE:1234:5678:23FF::/64        frei
```

7.2 Anhang 2: IPv6 und DNS

7.2.1 Forward Mapping

Die 128 bit lange IPv6-Adresse eines Endsystems kann in einem Typ AAAA-Record gespeichert werden:

```
$ORIGIN ipv6.surfnet.nl.
...
zesbot          IN   AAAA   3FFE:0604:0000:0001:02C0:4FFF:FEC6:9CC7
```

Dies ist vergleichbar mit dem Gebrauch des Typ A-Record in IPv4:

```
$ORIGIN ipv6.surfnet.nl.
...
zesbot          IN   A      192.87.110.60
```

Typ A und Typ AAAA Records eines angegebenen Bereichs werden in derselben DNS-Datei gespeichert. Ein Knoten mit mehr als einer IPv6-Adresse muss mehr als einen Typ AAAA-Record haben:

```
$ORIGIN ipv6.surfnet.nl.
 ...
amsterdam9      IN   AAAA   3FFE:0600:8000:0000::0001
                IN   AAAA   3FFE:0600:8000:0000::0005
                IN   AAAA   3FFE:0600:8000:0000::0009
                IN   AAAA   3FFE:0600:8000:0000::000D
```

Zur Zeit wird ein neuer Record-Typ definiert, um einen Domain Name auf eine IPv6-Adresse abzubilden. Dieser als Typ A6 bezeichnete Record enthält einen Verweis auf einen Präfix [DNSLOOKUP]. Der Typ A6-Record dient der Erleichterung von Network Renumbering und Multihoming. Domänen, die Typ A6-Records für IPv6-Adressen verwenden, können übergangsweise automatisch generierte Typ AAAA-Records nutzen. Sobald Typ A6-Records genügend weit verbreitet sind, ist zu erwarten, dass Typ AAAA-Records nicht länger genutzt werden.

7.2.2 Reverse Mapping

IPv4 nutzt die Domäne *in-addr.arpa* zum Reverse Mapping. Eine IPv4-Adresse wird als Name in der Domäne *in-addr.arpa* dargestellt. Der Name wird in Dezimalziffern durch Kommata getrennt mit dem Suffix *.in-addr.arpa* angeben. Die Angabe der Byte erfolgt in umgekehrter Reihenfolge, d.h. die niederwertigsten Byte werden zuerst angegeben, gefolgt von den nächst niederwertigen und so weiter. Der Eintrag der IPv4-Adresse 192.87.110.60 in der Domäne *in-addr.arpa* hat folgenden Aufbau:

```
60.110.87.192.in-addr.arpa.
```

In der DNS-Datei wird der Name in dieser Form gespeichert:

```
$ORIGIN 110.87.192.in-addr.arpa.
......
60              IN   PTR   zesbot.ipv6.surfnet.nl.
```

Für IPv6-Adressen existiert die Domäne *ip6.int*. Das weitere Vorgehen entspricht genau dem von IPv4 bekannten Verfahren, mit der Ausnahme, dass IPv6-Adressen dargestellt werden durch Hexadezimalziffern, die voneinander durch Punkte getrennte sind. Die IPv6-Adresse 3FFE:0604:0000:0001:02C0:4FFF:FEC6:9CC7 wird den Regeln entsprechend in die Domäne *ip6.int* eingetragen:

```
7.c.c.9.6.c.e.f.f.f.f.4.0.c.2.0.1.0.0.0.0.0.0.0.4.0.6.0.e.f.f.3.ip6.int.
```

Dieser Name wird folgendermaßen in einer DNS-Datei gespeichert (unter Verwendung eines /64 Präfix):

```
$ORIGIN 1.0.0.0.0.0.0.4.0.6.0.e.f.f.3.ip6.int.
......
7.c.c.9.6.c.e.f.f.f.f.4.0.c.2.0   IN   PTR   zesbot.ipv6.surfnet.nl.
```

IPv4- und IPv6-Reverse Mappings werden in verschiedenen DNS-Dateien gespeichert.

Fachwörter

6Bone
: Weltweites IPv6-Testnetz

Address Resolution Protocol
: Protokoll aus dem IPv4-Umfeld zur Zuordnung von Network Layer- und Link Layer-Adressen

Adress-Autokonfiguration
: Mechanismus von IPv6 damit ein Endsystem automatisch seine Netzwerkadresse und weitere für die Kommunikation erforderliche Daten ermitteln kann

Anycast-Adresse
: Spezielle IPv6-Adresse, die von mehreren Endsystemen gleichzeitig genutzt werden kann. Das IPv6-Netzwerk sorgt aber dafür, dass ein IPv6-Datagramm nur an jeweils eines der Endsysteme zugestellt wird.

Anycasting
: Kommunikationsmodell, das auf der Verwendung von Anycast-Adressen beruht. Speziell für Redundanz und Ausfallkonzepte geeignet.

ARP
: Abkürzung für Address Resolution Protocol. Siehe dort.

Authentifizierungsheader
: IPv6-Zusatzheader für die Authentifizierung des jeweiligen IPv6 Paketes

Automatische Tunnel
: Durch die Nutzung von IPv4-kompatiblen IPv6-Adressen automatisch erzeugter Tunnel

Backbone
: Ein Netz mit der Aufgabe, mehrere andere angeschlossene Netze, die sogenannten Frontend-Netze, an denen die Systeme und Anwendungen hängen, zu verbinden

BGP

Abkürzung für Border Gateway Protokoll. Siehe dort

BIND

Implementierung des Domain Name Service durch das Internet Software Consortium

Border Gateway Protokoll

Exterior Gateway Protokoll

Bridge

Gerät oder System, das zwei LANs koppelt und dabei nur auf Daten - im Wesentlichen auf Adressen - des Data Link Layers operiert. Eine „normale“ Bridge, auch lokale Bridge genannt, verknüpft zwei LANs direkt, eine Remote Bridge verknüpft zwei LANs gleicher Technologie über ein WAN.

care-of Adresse

Forwarding-Adresse für mobile Endsysteme

Client

Begriff aus der Client-/Server-Architektur: derjenige Partner, der die Dienste eines Servers in Anspruch nimmt

CSMA/CD

Carrier Sense Multiple Access/Collision Detection. Ein in IEEE 802.3 bzw. ISO 8892-3 definiertes Verfahren für LANs. Ethernet und 802.3 sind ähnlich, beide funktionieren nach dem CSMA/CD-Verfahren. Die beiden Begriffe werden daher oft synonym verwendet.

Datagram

Bezeichnung für Nachrichten, die bei verbindungsloser Kommunikation verschickt werden. Es wird nicht garantiert, dass Datagramme überhaupt, in der korrekten Reihenfolge oder nicht dupliziert beim Empfänger ankommen.

DHCP

Abkürzung für Dynamic Host Configuration Protocol. Siehe dort.

Distance Vector Multicast Routing Protocol

Routerprotokoll zum Aufbau eines Verteilbaumes für Multicast-Nachrichten

DNS

Abkürzung für Domain Name Service. Siehe dort.

Domain Name Service
System zur Verwaltung von Namen und Adressen im Internet

Draft
Vorstufe eines RFCs

DVMRP
Abkürzung für Distance Vector Multicast Routing Protocol. Siehe dort.

Dynamic Host Configuration Protocol
Protokoll zur automatischen Konfiguration von Endsystemen im Internet

Endsystem
System, in dem im Gegensatz zu einem Intermediate System eine Transport-Entity (und damit auch Applikationen) residieren. Das Endsystem kann physikalisch aus mehreren HW-Teilen bestehen.

Erweiterungsheader
IPv6-Header, die optionale Informationen enthalten, die nicht Bestandteil des IPv6-Basisheaders sind

Ethernet
von XEROX eingeführtes LAN, das auf dem Yellow Cable basiert und CSMA/CD als Übertragungsverfahren benutzt. Ähnlich zu einem IEEE 802.3-LAN

FDDI
Fiber Distributed Data Interface: in ISO 9314 definiertes Verfahren für LANs, ähnlich dem Token Ring mit höherer Geschwindigkeit

File Transfer Protocol
Protokoll für die Übertragung von Dateien im Internet

Firewall
Rechner, der ein Netzwerk gegen Angriffe und Eindringversuche von außen schützt

Forwardingadresse
Adresse, unter der ein mobiles Endsystem aktuell erreichbar ist

Fragmentierungsheader
IPv6-Zusatzheader für die Fragmentierung von IPv6-Paketen in kleinere Teilpakete

FTP
Abkürzung für File Transfer Protocol. Siehe dort.

Gateway
Im allgemeinen Sprachgebrauch ein System, das zwei oder mehrere Netze verknüpft und nicht als Bridge arbeitet. Varianten: Gateway auf der Netzebene (=Router), Transportgateway, Anwendungsgateway

Hop by Hop-Optionsheader
IPv6-Zusatzheader für die Aufnahme von Informationen, die jeder Router, der das jeweilige IPv6-Paket weiterleitet, auswerten muss

IAB
Abkürzung für Internet Architecture Board. Siehe dort.

ICMP
Abkürzung für Internet Control Message Protocol. Siehe dort.

IETF
Abkürzung für Internet Engineering Task Force. Siehe dort.

Internet
Kommunikationsarchitektur, gekennzeichnet durch die Verwendung von TCP und IP, entstanden aus dem ARPA-Netz in den USA. Erweiterungen werden durch den IAB über den RFC-Prozess kontrolliert

Internet Architecture Board
kontrolliert die Neuentwicklungen im Internet über den RFC-Mechanismus

Internet Control Message Protocol
Internet-Protokoll, mit dessen Hilfe Kontroll- und Fehlerinformationen übertragen werden

Internet Engineering Task Force
kontrolliert die Neuentwicklungen im Internet Netz über den RFC-Mechanismus

Internet Service Provider
Dienstleister, der für seine Kunden den Zugang zum Internet bereitstellt

Internet Software Consortium
Eine nichtkommerzielle Organisation, die Standardimplementierungen wichtiger Internetdienste realisiert. Lieferant des Sourcecodes des DNS-Servers BIND

Interworking
Oberbegriff für das Verknüpfen von Kommunikationsnetzwerken aller Art, wobei verschiedene Methoden zum Einsatz kommen

IPng
IP next generation, synonym für IPv6

IPv4
Internet Protocol: verbindungsloses Netzwerkprotokoll der Internet-Architektur mit 4 byte langen Adressen

IPv6
Internet Protocol: verbindungsloses Netzwerkprotokoll der Internet-Architektur mit 16 byte langen Adressen als Nachfolger von IPv4

IPv6-Header
Header des neuen Internetprotokolls IPv6

ISC
Abkürzung von Internet Software Consortium. Siehe dort

ISP
Abkürzung für Internet Service Provider. Siehe dort

Konfigurierter Tunnel
Durch manuelle Konfigurationsmaßnahmen eingerichteter Tunnel

LAN
Local Area Network: ursprünglich ein mit hoher Geschwindigkeit arbeitendes Netzwerk mit geringer Reichweite. Heute jedes Netz auch großer Reichweite, das gemäß CSMA/CD, Token Ring oder FDDI arbeitet

Link
Direkte Verbindung zwischen zwei Systemen (Endsystem/Router)

Link Layer
Linkebene der Kommunikation, gewährleistet die direkte Kommunikation zwischen zwei Systemen (Endsystem/Router) in einem Netzwerk

Message Integrity Code (MIC)
Ein durch ein Authentifizierungsverfahren gebildeter Code, der die Integrität einer Nachricht gewährleistet

MOSPF
Abkürzung für Multicast Open Shortest Path First. Siehe dort.

MTU
Maximale Größe eines Paketes, das über ein Netzwerk übertragen werden kann

Multicast-Adresse
Adresse, mit der eine Nachricht an mehrere Empfänger in einem Netz versendet werden kann

Multicast Open Shortest Path First
Routerprotokoll zum Aufbau eines Verteilbaumes für Multicast-Nachrichten

Multicasting
Verschicken einer Nachricht an mehrere Empfänger in einem Netz, die durch eine Gruppenadresse adressiert werden

NAT
Abkürzung für Network Translators. Siehe dort.

Network Layer
Netzwerkebene der Kommunikation; gewährleistet die Kommunikation zwischen verschiedenen Endsystemen in einem Netzwerk

Network Translators
Protokollübersetzer auf Netzwerkebene, der von einem Netzwerkprotokoll A in ein Netzwerkprotokoll B und umgekehrt übersetzt

Next Level Aggregator
Zweite Hierarchiestufe einer IPv6-Adresse

Next Level Aggregator Identifier
Identifier für die zweite Hierarchiestufe einer IPv6-Adresse

NGTRANS
IETF-Arbeitsgruppe, die Mechanismen für den Übergang von IPv4 nach IPv6 definiert

NLA
Abkürzung für Next Level Aggregator. Siehe dort.

NLA ID
Abkürzung für Next Level Aggregator Identifier. Siehe dort.

Open Shortest Path First
Interior Gateway Protokoll

OSPF
Abkürzung für Open Shortest Path First. Siehe dort.

Point to Point Protocol
Definiert in RFC1171, dient als Protokoll zwischen Routern über serielle Leitungen, um verschiedene Netzprotokolle darüber zu transportieren bzw. zu multiplexen

PPP
Abkürzung für Point to Point Protocol. Siehe dort.

Primärer IPv6-Header
Erster IPv6-Header. Enthält unter anderen die Adressinformation des IPv6-Paketes. Ist als einziger IPv6-Header immer in einem IPv6-Paket vorhanden

Renumbering
Synonym für Umnummerierung. Siehe dort.

RFC
Request for Comment, Verfahren im Internet zur Kommentierung von vorgeschlagenen Normen, Festlegungen oder auch Berichten, auch Bezeichnung für ein auf diese Weise verabschiedetes Dokument

RIP
Abkürzung für Routing Information Protocol. Siehe dort.

Routeaggregation
Zusammenfassung mehrerer Routen in Routern, um Speicherplatz für Routing-Tabellen zu sparen und damit ein schnelleres Routing zu erreichen

Router
Element in einem Netz, das zwischen Netzen residiert und Nachrichtenströme durch die Netze lenkt und dazu Wegewahl, Flusskontrolle, Adressierung und andere Funktionen behandelt.

Routing
Weiterleiten von Nachrichten durch verschiedene Netzwerke

Routing Header
IPv6-Zusatzheader für die Steuerung des Weges des jeweiligen IPv6 Paketes

Routing Information Protocol
Interior Gateway Protokoll

Routing-Protokoll
Protokoll, mit dem sich Router untereinander bzw. mit den angeschlossenen Endsystemen über Topologie, Änderungen und Kosten von Routen informieren

Routing-Tabellen
Tabellen, mit deren Hilfe ein Router ankommende Datenpakete weiterleitet

Server
Logische Instanz bzw. Anwendungskomponente, welche Aufträge eines Clients ausführt und die (koordinierte) Nutzung allgemein verfügbarer Dienste (File, Print, DB, Kommunikation,...) bereitstellt, kann selbst bzgl. eines anderen Service Client sein

Signatur
Ein durch ein Authentifizierungsverfahren gebildeter Code, der die Integrität einer Nachricht gewährleistet

Site Level Aggregator
Dritte Hierarchiestufe einer IPv6-Adresse

Site Level Aggregator Identifier
Identifier für die dritte Hierarchiestufe einer IPv6-Adresse

SLA
Abkürzung für Site Level Aggregator. Siehe dort.

SLA ID
Abkürzung für Site Level Aggregator Identifier. Siehe dort.

TLA
Abkürzung für Top Level Aggregator. Siehe dort.

TLA ID
Abkürzung für Top Level Aggregator Identifier. Siehe dort.

Token Ring
Technik der Token Ring LANs, dabei läuft ein Token im ringförmigen LAN herum, das zur Regelung der Sendeberechtigung der verschiedenen Stationen dient

Top Level Aggregator
Erste Hierarchiestufe einer IPv6-Adresse

Top Level Aggregator Identifier
Identifier für die erste Hierarchiestufe einer IPv6-Adresse

Transport Layer
Transportebene der Kommunikation, gewährleistet die Kommunikation zwischen verschiedenen auf Endsystemen residierenden Anwendungen in einem Netzwerk

Tunnel
Mechanismus, um IPv6-Pakete in IPv4-Pakete verpackt über eine existierende IPv4-Infrastruktur zu übertragen

Umnummerierung
Umnummerierung der IP-Adressen von Endsystemen und Routern eines Netzwerkes bzw. Teilnetzwerkes

Unicast-Adresse
Adresse, mit der eine Nachricht an genau einen Empfänger in einem Netz versendet werden kann

Verschlüsselungsheader
IPv6-Zusatzheader für die Verschlüsselung des jeweiligen IPv6-Paketes

WAN
Wide Area Network, öffentliches oder privates Netz, das große Entfernungen überbrückt und dabei - im Gegensatz zu LANs - relativ langsam mit höherer Fehlerrate arbeitet. Bei ATM Netzen z.B. gelten diese beiden Charakterisierungen nicht mehr.

Zieloptionsheader
IPv6 -Zusatzheader für die zusätzliche Optionen des jeweiligen IPv6-Paketes

Zusatzheader
Synonym für Erweiterungsheader des IPv6-Protokolls

Literatur

***open*Net Server V2.0** (BS2000/OSD)
BCAM V16.0A Band 1
Benutzerhandbuch

Zielgruppe
Das Handbuch richtet sich an Netzplaner, -generierer und -verwalter, die in BS2000-Systemen BCAM betreiben.
Inhalt
BCAM Band 1 beschreibt BCAM selbst, seine Einbettung in TRANSDATA und TCP/IP- und ISO-Netze, sowie Generierungs- und Administrationstätigkeiten. Generierungsbeispiele verdeutlichen die Beschreibung. Es werden BCAM-Tools zur Generierung und Diagnose beschrieben.

***open*Net Server V2.0** (BS2000/OSD)
BCAM V16.0A Band 2
Referenzhandbuch

Zielgruppe
Das Handbuch richtet sich an Netzoperateure, -generierer und -verwalter, die in BS2000-Systemen BCAM betreiben.
Inhalt
BCAM Band 2 baut auf Band 1 auf und beschreibt ausführlich die zur Generierung und zum Betrieb nötigen BCAM-Kommandos. Es werden die zur statischen Generierung nötigen KOGS-Makros vorgestellt und die BCAM-Fehlermeldungen aufgelistet.

***open*Net Server V2.0, *inter*Net Services V2.0** (BS2000/OSD)
SNMP-Management für *open*Net Server und *inter*Net Services
Benutzerhandbuch

Zielgruppe
Das Handbuch richtet sich an Netz- und Systemverantwortliche, die ein SNMP-basiertes Netz- und Systemmanagement nutzen möchten.
Inhalt
Das Handbuch beschreibt detailliert die mit *open*Net Server ausgelieferten MIBs, die mit *inter*Net Services ausgelieferte FTP-MIB, die Installation und den Betrieb der Subagenten. Ein eigenes Kapitel behandelt ausführlich die Bedienung des BCAM Managers.

***inter*Net Services V2.0** (BS2000/OSD)
Administratorhandbuch

Zielgruppe
Das Handbuch richtet sich an Netzplaner, -generierer und -verwalter, die in BS2000/OSD Internet Services betreiben wollen.
Inhalt
Das Handbuch beschreibt die Funktionalität der Internet Services
BOOTP/DHCP, TFTP, DNS, FTP, LDAP und NTP in BS2000/OSD. Installation, Administration, Betrieb, Logging- und Diagnose-Möglichkeiten der einzelnen Komponenten sowie FTP-Exit und TELNET-Exits sind weitere Themen dieses Handbuchs.

***inter*Net Services V2.0** (BS2000/OSD)
Benutzerhandbuch

Zielgruppe
Das Handbuch richtet sich an Netzplaner, -generierer und -verwalter sowie Nutzer, die die Internet Services in Verbindung mit BS2000/OSD nutzen wollen.
Inhalt
Das Handbuch stellt die Komponenten von *inter*Net Services vor. Ausführlich werden die Nutzung von FTP, der FTAC-Schnittstelle für FTP und TELNET beschrieben. Netzverwalter benötigen dieses Handbuch zusätzlich zum Administratorhandbuch.

***inter*Net Value Edition V1.0B** (BS2000/OSD)
Benutzerhandbuch

Zielgruppe
Das Handbuch richtet sich an Netzplaner, -generierer und -verwalter, die in BS2000/OSD Mail Service betreiben wollen.
Inhalt
*inter*Net Value Edition ist eine kostenfreie Ergänzung der *inter*Net Services. Das Handbuch stellt die Komponenten der *inter*Net Value Edition vor und gibt Hinweise zu Installation, Administration und Betrieb des Mail Service in BS2000/OSD.

SOCKETS(BS2000) V2.0
SOCKETS für BS2000/OSD
Benutzerhandbuch

Zielgruppe
C-Programmierer, die mit den Funktionen der SOCKETS(BS2000)-Schnittstelle Kommunikationsanwendungen im BS2000/OSD entwickeln wollen.
Inhalt
– Einführung in SOCKETS(BS2000)
– Benutzerfunktionen von SOCKETS(BS2000)
– Installation und Programmerstellung

CMX (BS2000)
Kommunikationsmethode im BS2000
Benutzerhandbuch

Zielgruppe
Programmierer von Transport-Service-Anwendungen (TS-Anwendungen)
Inhalt
CMX (BS2000) bietet Anwendungsprogrammen eine einheitliche Schnittstelle zu den Transportdiensten. Mit CMX (BS2000) können Sie Anwendungsprogramme erstellen, die unabhängig vom Transportsystem mit anderen Anwendungen kommunizieren können.

SNMP Management V5.0
SNMP Management für BS2000/OSD
Benutzerhandbuch

Zielgruppe
Das Handbuch wendet sich an Netzverwalter, -operateure und Systemverwalter, die BS2000-Systeme in ein SNMP-basiertes Management integrieren bzw. ein solches System bedienen wollen.
Inhalt

Dieses Handbuch beschreibt einerseits die Einbettung von SBA-BS2, SSC-BS2, SSA-SM2-BS2 und SSA-OUTM-BS2 in BS2000/OSD und die zum Betrieb notwendigen Installations- und Konfigurationsschritte sowie den Betrieb selbst. Die zur Überwachung notwendigen Agenten und ihre MIBs werden detailliert vorgestellt. Andererseits wird die Installation und Konfiguration der entsprechenden Management-Anwendungen auf den Management-Plattformen Unicenter TNG, TransView SNMP und HP OpenView beschrieben.
Weitere zentrale Themen des Handbuchs sind der Zugriff auf Management-Informationen über das World Wide Web sowie der Trap-Server für Solaris und Reliant UNIX.

IMON (BS2000/OSD)
Installationsmonitor
Benutzerhandbuch

Zielgruppe
Das Handbuch wendet sich an die Systembetreuung des Betriebssystems BS2000/OSD.
Inhalt
Das Handbuch beschreibt die Installation und Verwaltung von BS2000-Software mit dem Installationsmonitor IMON und seinen drei Komponenten IMON-BAS, IMON-GPN und IMON-SIC. In zwei Beispielkapiteln wird die Installation (standard und kundenspezifisch) mit der Komponente IMON-BAS für Systeme mit BS2000-OSD V2.0 und ab BS2000-OSD V3.0 ausführlich dargestellt.

BS2000/OSD-BC
Einführung in die Systembetreuung
Benutzerhandbuch

Zielgruppe
Das Handbuch wendet sich an die Systembetreuung und das Operating des Betriebssystems BS2000/OSD.
Inhalt
Es sind u.a. folgende Themen zur Verwaltung und Überwachung des BS2000/OSD-Grundausbaus enthalten: Systemeinleitung, Parameterservice, Job- und Tasksteuerung, Speicher-, Geräte-, Benutzer-, Datei- und Pubset-Verwaltung, Privilegienvergabe, Accounting und Operatorfunktionen.

BS2000/OSD-BC
Einführung in die Systembetreuung
Benutzerhandbuch

Zielgruppe
Das Handbuch wendet sich an die Systembetreuung und das Operating des Betriebssystems BS2000/OSD.
Inhalt
Es sind u.a. folgende Themen zur Verwaltung und Überwachung des BS2000/OSD-Grundausbaus enthalten: Systemeinleitung, Parameterservice, Job- und Tasksteuerung, Speicher-, Geräte-, Benutzer-, Datei- und Pubset-Verwaltung, Privilegienvergabe, Accounting und Operatorfunktionen.

Bestellen von RFCs

Die im Text zitierten Request for Comments (RFCs) sind, soweit sie nicht mitausgeliefert wurden, als gedruckte Ausgaben gegen eine Kopiergebühr oder als Datei über „anonymous Internet FTP“ bzw. E-Mail erhältlich.

Anonymous Internet FTP: Um einen RFC über Internet vom System *nic.ddn.mil* (IP-Adresse 192.67.67.20) zu erhalten, gehen Sie wie folgt vor:
– Erzeugen Sie eine FTP-Verbindung zum System: *ftp nic.ddn.mil*.
– Sie können nun aus dem Verzeichnis *rfc* die gewünschten Dokumente laden. Eine Liste aller verfügbaren Dokumente finden Sie in der Datei *rfc-index.txt*.

E-Mail:

Wenn Sie keinen Internet-Anschluß haben, aber Zugang zu Electronic Mail, können Sie einen RFC auch über E-Mail anfordern. Das Dokument wird Ihnen als Antwort auf Ihre Anfrage *Mail* zurückgesandt.

Senden Sie hierzu eine Mail an den Benutzer *service* auf dem System *nic.ddn.mil*: `mail service@nic.ddn.mil`
Geben Sie im Feld *Subject* die Nummer des gewünschten RFCs ein, z.B.:
`Subject: RFC 1155`

Schriftliche Anfragen zu RFCs richten Sie an:

DDN Network Information Center
SRI International
333 Ravenswood Ave.
Menlo Park, CA 94025, U.S.A.
Telefon: 415-859-3695

E-Mail: nic@nic.ddn.mil

Wenden Sie sich zum Bestellen von Handbüchern bitte an Ihre zuständige Geschäftsstelle.

Stichwörter

V

Z

Inhalt

*open*Net Server V2.0

IPv6 Einführung und Umstellhandbuch Stufe 1

Zielgruppe

Das Handbuch wendet sich an alle, die über die Einführung von IPv6 in BS2000/OSD entscheiden sowie an alle, die die IPv6-Funktionalität auf BS2000/OSD-Mainframes nutzen oder IPv6 in BS2000/OSD installieren wollen.

Inhalt

Das Handbuch informiert über die kommerziellen und technischen Grundlagen von IPv6. Darüber hinaus wird der Übergang von IPv4 nach IPv6 anhand von Beispielen erläutert und der aktuelle Stand der Implementierung von IPv6 in BS2000/OSD dargestellt. Detaillierte Informationen zu den Themen „IPv6-Adressierung“ und „DNS-Nutzung“ werden im Anhang des Handbuchs geliefert.

Ausgabe: Februar 2001

Datei: ipv6_ums.pdf

Fujitsu Siemens Computers GmbH
Handbuchredaktion
81730 München

Kritik
Anregungen
Korrekturen

Fax: 0 700 / 372 00000

e-mail: manuals@fujitsu-siemens.com
http://manuals.fujitsu-siemens.com

Absender

Kommentar zu openNetServer V2.0
IPv6 Einführung und Umstellhandbuch Stufe 1

Information on this document

On April 1, 2009, Fujitsu became the sole owner of Fujitsu Siemens Computers. This new subsidiary of Fujitsu has been renamed Fujitsu Technology Solutions.

This document from the document archive refers to a product version which was released a considerable time ago or which is no longer marketed.

Please note that all company references and copyrights in this document have been legally transferred to Fujitsu Technology Solutions.

Contact and support addresses will now be offered by Fujitsu Technology Solutions and have the format *...@ts.fujitsu.com*.

The Internet pages of Fujitsu Technology Solutions are available at *http://ts.fujitsu.com/*...
and the user documentation at *http://manuals.ts.fujitsu.com*.

Hinweise zum vorliegenden Dokument

Zum 1. April 2009 ist Fujitsu Siemens Computers in den alleinigen Besitz von Fujitsu übergegangen. Diese neue Tochtergesellschaft von Fujitsu trägt seitdem den Namen Fujitsu Technology Solutions.

Das vorliegende Dokument aus dem Dokumentenarchiv bezieht sich auf eine bereits vor längerer Zeit freigegebene oder nicht mehr im Vertrieb befindliche Produktversion.

Bitte beachten Sie, dass alle Firmenbezüge und Copyrights im vorliegenden Dokument rechtlich auf Fujitsu Technology Solutions übergegangen sind.

Kontakt- und Supportadressen werden nun von Fujitsu Technology Solutions angeboten und haben die Form *...@ts.fujitsu.com*.

Die Internetseiten von Fujitsu Technology Solutions finden Sie unter *http://de.ts.fujitsu.com/*..., und unter *http://manuals.ts.fujitsu.com* finden Sie die Benutzerdokumentation.