Sebastian Seeber

Vorgehensweise der Migration von IPv4 nach IPv6 in einem Unternehmen der IT Branche unter Berücksichtigung von Microsoft und SAP Anwendungen im städtischen Campus Netz und der Internet Anbindung über Firewalls

MASTERARBEIT

HOCHSCHULE MITTWEIDA (FH)

UNIVERSITY OF APPLIED SCIENCES

Fachbereich Mathematik / Naturwissenschaften / Informatik

Mittweida, September 2011

Sebastian Seeber

Vorgehensweise der Migration von IPv4 nach IPv6 in einem Unternehmen der IT Branche unter Berücksichtigung von Microsoft und SAP Anwendungen im städtischen Campus Netz und der Internet Anbindung über Firewalls

eingereicht als

MASTERARBEIT

an der

HOCHSCHULE MITTWEIDA (FH)

UNIVERSITY OF APPLIED SCIENCES

Fachbereich Mathematik / Naturwissenschaften / Informatik

Mittweida, September 2011

Erstprüfer: Prof. Dr. J. Mario Geißler
Zweitprüfer: Dipl.-Ing. Thomas Barth

Vorgelegte Arbeit wurde verteidigt am: 16. Dezember 2011

Bibliographische Beschreibung:

Sebastian Seeber:
Vorgehensweise der Migration von IPv4 nach IPv6 in einem Unternehmen der IT Branche unter Berücksichtigung von Microsoft und SAP Anwendungen im städtischen Campus Netz und der Internet Anbindung über Firewalls 2011. - 125 S. Mittweida,
Hochschule Mittweida (FH) - University of Applied Sciences,
Fachbereich Mathematik / Naturwissenschaften / Informatik, Masterarbeit, 2011

Referat:

Diese Arbeit beschäftigt sich mit der Migration des Internetprotokolls von der Version 4 zur Version 6. Dazu werden zuerst die Grundlagen des neuen Protokolls und die speziell dafür entwickelten Migrationsmechanismen vorgestellt. Diese werden in einem Testnetzwerk überprüft, um die dabei erlangten Kenntnisse auf das Produktivnetzwerk der Gesellschaft für Informationsverarbeitung perdata mbH zu übertragen. Das letzte Kapitel gibt eine Empfehlung, wie bei der Migration vorzugehen ist und was dabei beachtet werden sollte.

Danksagung

Mit der vorliegenden Arbeit schließe ich mein Masterstudium der Informatik an der Hochschule Mittweida ab.
An dieser Stelle möchte ich mich bei Herrn Prof. Dr.-Ing. J. Mario Geißler für die hervorragende Betreuung sowie das entgegengebrachte Vertrauen während der Bearbeitungszeit bedanken. Außerdem danke ich der perdata Gesellschaft für Informationsverarbeitung mbH, insbesondere Herrn Dipl.-Ing. Thomas Barth für die technische Unterstützung sowie für viele Anregungen und Hinweise, die zum Gelingen dieser Arbeit beigetragen haben.
Mein Dank gilt auch allen Freunden und Bekannten, die mich während der Anfertigung dieser Masterarbeit in irgendeiner Form unterstützt haben.
Des Weiteren bedanke ich mich bei meiner Familie und meiner Freundin, die mich in der gesamten Zeit meines Studiums unterstützt und über kleine Durststrecken hinweg geholfen haben.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Listingsverzeichnis

Abkürzungsverzeichnis

ABR	Area Border Router
ACL	Access Control List
AH	Authentication-Header
ARP	Address Resolution Protocol
AS	Autonomes System
ASBR	Autonomous System Boundary Router
BGP	Border Gateway Protocol
CIDR	Classless Inter-Domain Routing
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
EIGRP	Enhanced Interior Gateway Routing Protocol
ESN	Erweiterte Sequenznummer
ESP	Encapsulating Security Payload
EUI	Extended Unique Identifier
FTP	File Transfer Protocol
HTTP	Hypertext Transfer Protocol
ICMP	Internet Control Message Protocol
ID	Identifikationsnummer
IGMP	Internet Group Management Protocol
IKE	Internet Key Exchange
InterfaceID	Interface Identifier
IPsec	Internet Protocol Security
IS-IS	Intermediate System to Intermediate System Protocol
LSA	Link State Advertisement
MAC-Adresse	Media-Access-Control-Adresse
MTU	Maximum Tranmission Unit
NAT	Network Address Translation
ND	Neighbor Discovery
NLRI	Network Layer Reachability Information
NSSA	Not-So-Stubby Area
OSPF	Open Shortest Path First
QoS	Quality of Service
RARP	Reverse Address Resolution Protocol
RFC	Request for Comments
RIP	Routing Information Protocol
SA	Security Association
SAP	Unternehmen, das betriebswirtschaftliche Software entwickelt

SDM	Switch Database Management
SIP	Session Initiation Protocol
Site	Standort des Netzwerks
SMTP	Simple Mail Transfer Protocol
SSH	Secure Shell
SSM	Source Specific Multicast
TCP	Transmission Control Protocol
UDP	User Datagram Protocol

Kapitel 1

Einleitung

1.1 Aufgabe

„Letzte IPv4-Adressblöcke verteilt“ [Lück, 2011], heißt es in einem Artikel auf der Internetseite der Computer Reseller News vom 28.03.2011. Damit sind die letzten Adressblöcke der rund vier Milliarden IPv4-Adressen verteilt. Eine Umstellung auf das schon lange bekannte und standardisierte Internet Protokoll in der Version 6 scheint dringender denn je. Aus diesem Grund wird im oben genannten Artikel [Lück, 2011] zur Teilnahme an einem Kongress, der sich mit dem Thema der Migration zum neuen Protokoll beschäftigt, aufgerufen. Es ist der dritte Kongress zum Thema IPv6, der sich das Vorantreiben der Umstellung zum Ziel gestellt hat.
Für jedes Unternehmen stellt sich daher nicht mehr die Frage, ob eine Umstellung notwendig ist, sondern wann und wie diese vollzogen wird. Die vorliegende Arbeit soll daher die Fragen klären, wie eine Migration hin zum neuen Internetprotokoll am effizientesten für ein Unternehmen zu bewerkstelligen ist.

1.2 Motivation

Dieser Arbeit ging ein Forschungsprojekt zum Thema der Migration von IPv4 zu IPv6 in einem Unternehmen der IT-Branche voraus [Seeber, 2011]. Darin wurden grundlegende theoretische Sachverhalte zum neuen Internetprotokoll erarbeitet und erste praktische Erfahrungen gesammelt. Jedoch ist der Umfang eines Forschungsprojekts bei Weitem nicht ausreichend, um ein solch komplexes Thema ausreichend zu erörtern.
Zur Erarbeitung einer Vorgehensweise für die Migration, müssen Themen wie Routing, Protokolle höherer Ebenen, Sicherheitstechniken von IPv6 sowie die Unterstützung der

Hersteller von Soft- und Hardware tiefgreifend untersucht werden. Ferner sind praktische Untersuchungen der Geräte und Anwendungen wichtig, um die Implementierung der Standards in den jeweiligen Komponenten zu überprüfen und eventuelle Kompatibilitätsprobleme zwischen diesen zu erkennen.
Ziel ist es, eine sinnvolle und realisierbare Migrationsstrategie zu entwickeln, die eine reibungslose Umstellung des bestehenden Netzwerks ermöglicht.

1.3 Kapitelübersicht

Kapitel 2

Dieses Kapitel beschreibt alle wichtigen grundlegenden Bestandteile von IPv6, die für eine Verwendung dieses Internetprotokolls notwendig sein können. Das Protokoll zum Austausch von Informationen und Fehlermeldungen im Netzwerk ICMP ist dabei sehr detailliert erläutert, da eine Vielzahl von Funktionen für IPv6 darauf aufbauen. Am Ende dieses Kapitels gibt es einen Einblick in die Veränderungen, die an den Routingprotokollen im Zusammenhang mit IPv6 notwendig waren.

Kapitel 3

Dieser Teil der Arbeit beschäftigt sich mit den Mechanismen, die eigens für die Migration des Internetprotokolls entwickelt worden sind. Zudem wird auf Anwendungsszenarien und Konfigurationsbetrachtungen näher eingegangen.

Kapitel 4

Der Aufbau, die Konfiguration und die Überprüfung des Testnetzwerks stehen im Mittelpunkt dieses Kapitels. Hierbei wird außerdem auf die Konfiguration von Tunneln und ACLs eingegangen.

Kapitel 5

Die Sicherheit bei IPv6 ist genauso wie bei IPv4 ein wichtiges Thema für den Netzwerkbetrieb. Dieser Abschnitt beschreibt Sicherheitsrisiken ebenso wie Schutzmechanismen im IPv6-Netzwerk. Weiterhin wird die Konfiguration einer IPsec-Verbindung vorgestellt.

Kapitel 6

Das letzte Kapitel fasst die gewonnenen Ergebnisse zusammen und gibt einen Ausblick auf die Zukunft der Netzwerke mit IPv6.

Kapitel 2

Internetprotokoll Version 6 - Grundlagen

Damit eine Umstellung, egal welcher Art, möglich ist muss, theoretisches Wissen zu beiden Teilen, also dem neuen und dem alten Produkt, vorhanden sein. Für die Migration von einem Netzwerkprotokoll wird davon ausgegangen, dass die Grundlagen des alten Protokolls (IPv4) ausreichend bekannt sind, da mit diesem Protokoll schon lange gearbeitet wird. Aus diesem Grund wird in diesem Kapitel auf das Basiswissen des neuen Internetprotokolls in der Version 6 eingegangen sowie auf Änderungen im Routing und häufig verwendeten Protokollen höherer Schichten. Außerdem wird an markanten Stellen auf die konkreten Unterschiede zwischen dem neuen und dem alten Protokoll hingewiesen.

2.1 Aufbau des IPv6-Headers

Zur Gewährleistung eines optimalen Betriebes eines Netzwerkes ist es notwendig, das genutzte Protokoll zu kennen. Somit können Soft- und Hardware schneller konfiguriert sowie Probleme schneller identifiziert und behoben werden.
Der Header des Internet Protokolls in der Version 6 ist im RFC 2460 [S. Deering, 1998] definiert. Darin ist ein fester Header mit einer Größe von genau 40 Byte vorgesehen. Davon sind jeweils 16 Byte für die Adressen (Sender- und Empfängeradresse) enthalten. Die zusätzlichen 8 Byte enthalten allgemeine Header-Informationen. Welche das im einzelnen sind, wird im Abschnitt 2.1.1 behandelt.
Die in der Version 4 des Internetprotokolls vorhandene *Header-Length* ist nicht mehr notwendig, da der Header eine feste Länge besitzt. Für die Fragmentierung von Paketen bisher notwendige Felder wie *Identification, Flags* und *Fragment Offset* sind ebenso

nicht mehr im Header enthalten, da ein Paket, wenn es fragmentiert werden soll, einfach einen entsprechenden Erweiterungsheader erhält. Nicht mehr Bestandteil des Headers ist die *Header Checksum*, da in der Sicherungsschicht (Link-Layer) sowie auch in der Transportschicht (Transport-Layer) schon eine Checksumme gebildet wird. Durch das Entfallen der Prüfsumme im Header verringert sich überdies die Verarbeitungsgeschwindigkeit eines Paketes im Router. Alle restlichen Felder wurden geändert oder umbenannt.

2.1.1 Der IPv6-Header im Einzelnen

Die Abbildung 2.1 veranschaulicht die Vereinfachung des Headers. Hier wird deutlich, dass nur 8 Byte für allgemeine Header-Informationen genutzt werden.

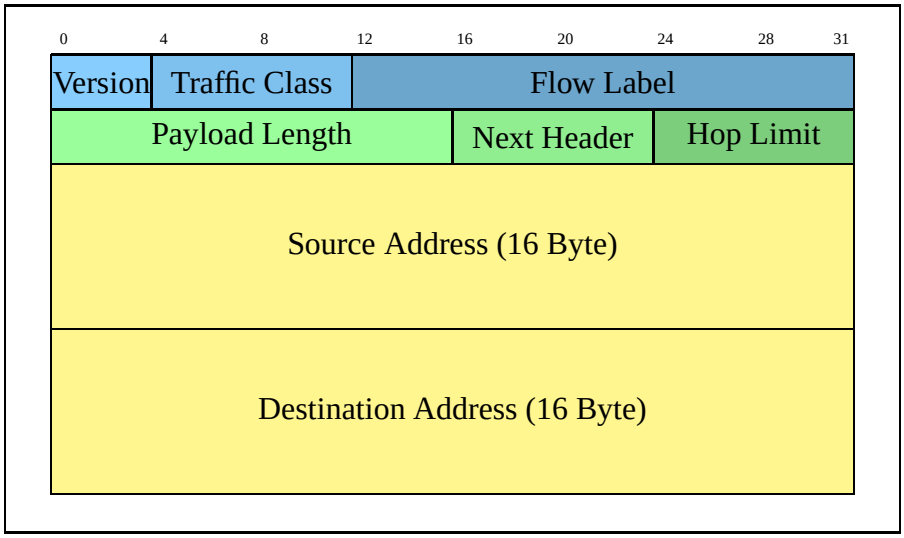


Abbildung 2.1: Aufbau des IPv6-Headers

Die Bestandteile des Headers sind im Folgenden beschrieben.

Version (4 Bits)

In diesem Feld steht die Versionsnummer des Protokolls. Sie ist in unserem Fall sechs. Die Nummer fünf konnte nicht verwendet werden, da sie für ein experimentelles Stream-Protokoll (RFC 1819 [L. Delgrossi, 1995]) vergeben wurde.

Traffic Class (1 Byte)

Type of Service heißt es bei IPv4. Es dient zur besseren Weiterleitung von speziellen Daten (beispielsweise Audio- oder Video-Daten). Diese Daten werden in unterschiedliche Klassen unterteilt, damit Router sowie Knoten diese entsprechend

erkennen und verarbeiten können. Details zu den einzelnen Klassen und deren Nutzung können den RFCs 2474 [K. Nichols, 1998] und 2475 [S. Blake, 1998] entnommen werden.

Flow Label (20 Bits)

Zur Gleichbehandlung von Paketen kann dieses Feld verwendet werden. Dadurch können z.B. Real-Time Pakete mit identischen Optionen schneller verarbeitet bzw. weitergeleitet werden, da die Optionen, die zu einer Folge von Paketen gehören, nur beim ersten Paket ausgelesen und im Router gespeichert werden. Alle Pakete, die zu einem Flow gehören, müssen dasselbe *Flow-Label* sowie dieselben Absender- und Empfängeradressen besitzen.

Wenn ein Router die *Flow-Label* Option nicht unterstützt, muss er diese Pakete unverändert weiterleiten. Keine spezielle Behandlung wird erwartet, wenn der Wert dieses Feldes null beträgt.

Payload Length (2 Bytes)

Dieses Feld gibt die Länge der Daten an, die dem IP-Header folgen. Hierbei ist zu beachten, dass der Header nicht, wie bei IPv4, in den Payload einfließt, da dieser hier eine feste Größe hat. Dem IP-Header folgende Erweiterungsheader werden in die Berechnung mit einbezogen. Durch die Größe des Feldes von 2 Bytes ist der Payload auf 64 kB beschränkt. Damit Pakete mit einer höheren MTU transportiert werden können, muss der Jumbogramm Extension-Header verwendet werden (RFC 2675) [D. Borman, 1999].

Next-Header (1 Byte)

Um die andere Organisation des IP-Headers zu unterstreichen, wurde dieses Feld, welches in IPv4 dem *Protocol Type*-Feld entspricht, umbenannt. Bei IPv6 kann der nachfolgende Header ein Erweiterungs- oder ein Protokoll-Header sein. Wenn es sich dabei um einen Protokoll-Header handelt, wie beispielsweise für TCP oder UDP werden, dieselben wie bei IPv4 verwendet. Bei einem Erweiterungsheader wird hier der Typ des nächsten Headers angegeben. Ein solcher Extension-Header befindet sich immer zwischen dem IPv6-Header und dem Protokoll-Header.

In Tabelle 2.1 befinden sich die wichtigsten Next-Header Werte [1].

[1] Eine aktuelle Liste der Next-Header Werte ist unter `http://www.iana.org/assignments/protocol-numbers/` zu finden.

Wert	Protokoll
0	IPv6: Hop-by-Hop-Options-Header; IPv4: reserviert, nicht genutzt
1	Internet Control Message Protocol (ICMPv4)
2	Internet Group Management Protocol (IGMPv4)
4	IP in IP (Encapsualtion)
6	TCP
8	Exterior Gateway Protocol (EGP)
9	Interior Gateway Protocol (IGP), bei Cisco für IGRP
17	UDP
41	IPv6
43	Routing-Header
44	Fragmentation-Header
45	Interdomain Routing Protocol (IDRP)
46	Resource Reservation Protocol (RSVP)
47	General Routing Encapsulation (GRE)
50	Encrypted-Security-Payload-Header (ESP)
51	Authentication-Header
58	ICMPv6
59	No Next Header for IPv6
60	Destination-Options-Header
88	EIGRPv4 und EIGRPv6
89	OSPF
108	IP Payload Compression Protocol
115	Layer 2 Tunneling Protocol (L2TP)
132	Stream Control Transmission Protocol (SCTP)
135	Mobility Header (Mobile-IPv6)
136 - 252	keine Zuweisung
253 - 254	Testzwecke
255	Reserviert

Tabelle 2.1: Werte des Next-Header Feldes

Hop Limit (1 Byte)

Dieses Feld entspricht dem *TTL (Time to Live)* Feld in IPv4. Gemäß der Definition enthält das Feld bei IPv4 die Anzahl an Sekunden, die ein Paket im Netz verweilen darf, bis es verworfen wird. Die meisten IPv4 Router haben dies so realisiert, dass der Wert bei jedem Router-Durchlauf um eins verringert wird. Da dies eine gängige Realisierung darstellt, wurde dieses Feld in IPv6 einfach entsprechend umbenannt. Es steht nun für die Anzahl der Hops, die das Paket noch weitergeleitet werden kann.

Wenn ein Router ein Paket mit dem Wert eins bekommt, reduziert er diesen Wert auf null. Somit wird das Paket verworfen und der Absender erhält eine ICMPv6-Fehlermeldung („Hop Limit exceeded in transit“)

Source Address (16 Bytes)

In diesem Feld wird die IPv6-Adresse des Absenders eingetragen.

Destination Address (16 Bytes)

Bei IPv4 enthielt dieses Feld die Adresse des endgültigen Empfängers. Bei IPv6 muss dies nicht notwendigerweise zutreffen. Hier kann auch die Adresse des nächsten Hop angegeben sein, falls z.B. ein Routing-Header vorhanden ist.

2.1.2 Extension-Header

Der IPv4-Header der eine variable Länge von 20 bis 60 Byte besitzt, kann zu den notwendigen Angaben auch noch weitere Informationen, wie z.B. zu Security-Optionen oder Source Routing, aufnehmen. Da aber durch die Verwendung unterschiedlich großer Header die Performance beeinträchtigt wird, wurde diese Möglichkeit fast ganz außer Acht gelassen.

Die einfache Struktur des IPv6-Headers ermöglicht eine sehr schnelle Verarbeitung. Damit auch hier zusätzliche Optionen mit angegeben werden können, aber die Verarbeitungsgeschwindigkeit nicht darunter leidet, sind hier zusätzliche Header erforderlich. Dadurch wird der IP-Header nicht vergrößert, sondern es gibt für unterschiedliche Optionen unterschiedliche Header, die nur angefügt werden, wenn dies auch notwendig ist.

In den aktuellen Vereinbarungen für das Internetprotokoll der Version 6 sind sechs Erweiterungsheader (Extension-Header) vorgeschrieben, die eine IPv6-Implementierung enthalten müssen. Die folgende Liste enthält diese Extension-Header:

- Routing-Header
- Fragment-Header

- Destination-Options-Header
- Hop-by-Hop-Options-Header
- Authentication-Header
- Encapsulating-Security-Payload-Header

Durch den Aufbau des IPv6-Headers aus einem festen Header und mehreren Erweiterungsheadern ist es möglich, auch in Zukunft weitere Extension-Header zur Erweiterung des Funktionsumfangs zu erarbeiten und zu implementieren.
Die in den zusätzlichen Headern enthaltenen Informationen werden nur von dem Knoten verarbeitet, der im IP-Header eingetragen ist. So wird bei einem Hop-by-Hop-Header der jeweils als nächste eingetragene Knoten in den IPv6-Header eingefügt. Wenn das Adressfeld dabei eine Multicast-Adresse enthält, wird der Erweiterungsheader von jedem Empfänger abgearbeitet. Die Reihenfolge der Verarbeitung richtet sich nach der Reihenfolge der Header im Paket. Alle Erweiterungsheader sind zwischen dem IP-Header und den Headern höherer Protokollschichten (Upper-Layer Protokolle) platziert. Jeder Extension-Header muss eine Länge besitzen, die einem Vielfachen von 8 Byte entspricht, damit nachfolgende Header auf 8 Bytes ausgerichtet werden können. Wenn ein Knoten bei der Verarbeitung auf ein *Next-Header* Feld stößt, das er nicht kennt, so muss das komplette Paket verworfen und eine ICMPv6 Nachricht („Parameter Problem“) an den Absender geschickt werden. Sobald mehrere Erweiterungsheader in einem Paket enthalten sind, sollte die in RFC 2460 [S. Deering, 1998] definierte Reihenfolge eingehalten werden.

1. IPv6-Header
2. Hop-by-Hop-Options-Header
3. Destination-Options-Header
   Dieser enthält hier die Optionen, die, von den Routern auf dem Weg zum finalen Empfänger, bearbeitet werden müssen.
4. Routing-Header
5. Fragment-Header
6. Authentication-Header
7. Encapsulating-Security-Payload-Header

8. Destination-Options-Header
   In diesem Fall enthält er die Optionen, die vom finalen Paketempfänger zu bearbeiten sind.

9. Upper-Layer-Header

Außerdem sind in diesem RFC die Erweiterungsheader: Hop-by-Hop-Options-Header, Routing-Header, Fragment-Header und Destination-Options-Header definiert. Der Authentication-Header und der Encapsulating-Security-Payload-Header sind in RFC 4302 [Kent, 2005a] bzw. RFC 4303 [Kent, 2005b] festgelegt. In jedem Paket darf der Destination-Options-Header zweimal, alle anderen Extension-Header nur einmal vorkommen. Wird ein IPv6 in IPv6-Tunneling durchgeführt, kann es vorkommen, dass nach dem IPv6-Header wieder ein IPv6-Header folgt. So kann dieser wieder von Erweiterungsheadern gefolgt sein. Auch hier muss sich an die oben genannte Reihenfolge gehalten werden.
In den folgenden Abschnitten werden die ersten vier definierten Extension-Header (RFC 2460 [S. Deering, 1998]) ausführlich betrachtet.

Hop-by-Hop Options Header Der Hop-by-Hop Erweiterungsheader wird verwendet, um Routern entlang des Paketweges Informationen zu vermitteln, die durch ihn zu verarbeiten sind. Er folgt direkt dem IPv6-Header und wird durch eine Null im *Next-Header*-Feld angezeigt. Verwendet wird dieser Header beim Resource Reservation Protocol (RSVP) oder für Nachrichten des Multicast Listener Discovery (MLD).
Bei IPv4 verlangsamen Optionen, die für Router entlang des Paketpfades bestimmt sind, die Verarbeitungszeit, da bei jedem Paket erst die Optionsfelder auf Routinginformationen hin analysiert werden müssen. Im neuen Internetprotokoll ist es nur notwendig zu überprüfen, ob ein Hop-by-Hop Erweiterungsheader vorhanden ist oder nicht. Wenn er enthalten ist, muss er verarbeitet werden, ansonsten wird das Paket einfach weitergeleitet. Der Aufbau des Hop-by-Hop Extension Headers ist in Abbildung 2.2 gezeigt.

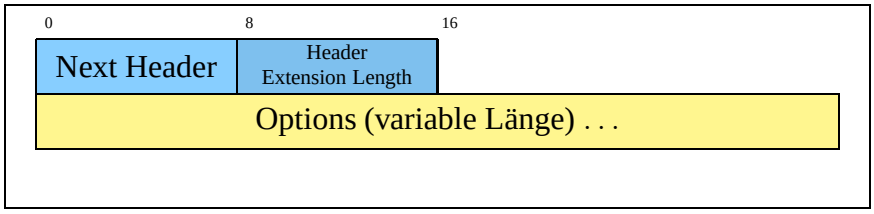


Abbildung 2.2: Aufbau des Hop-by-Hop Options Headers

Im Weiteren werden die einzelnen Felder genauer beschrieben:

Next-Header (1 Byte)
Dieses Feld beschreibt den Typ des Headers, der dem Hop-by-Hop-Options-Header folgt. Die möglichen Werte sind in Tabelle 2.1 beschrieben.

Header Extension Length (1 Byte)
Die Berechnung der Länge dieses Headers, welche in das Feld eingetragen wird, erfolgt in acht Byte Einheiten, wobei die ersten acht Bytes nicht mit eingerechnet werden. Ist er kürzer als acht Bytes, wird hier der Wert null eingetragen.

Options (variable Länge)
Für diesen Header können eine oder auch mehrere Optionen definiert sein. Die daraus resultierende variable Länge des Headers wird über das *Header Extension Length*-Feld angegeben. Falls ein Knoten die angegebenen Optionen nicht kennt, beschreiben die ersten zwei Bits wie er darauf zu reagieren hat.

Wert 0b00
Die Option ist zu überspringen und das Paket weiter zu verarbeiten.

Wert 0b01
Das Paket ist zu verwerfen.

Wert 0b10
Es wird eine ICMPv6 Nachricht mit dem Inhalt „Parameter Problem“ gesendet, wobei der Pointer auf die nicht bekannte Option zeigt. Das Paket ist zu verwerfen.

Wert 0b11
Es kann wie bei Wert 0b10 vorgegangen werden, wenn die Empfängeradresse keiner Multicast-Adresse entspricht.

Das dritte Bit gibt an, ob die Informationen in den Optionen auf dem Weg veränderbar sind (Wert 0b1) oder nicht (Wert 0b0).

Option Jumbogramm Damit IPv6-Jumbogramme versendet werden können, kann diese Option (Jumbo Payload Option RFC 2675 [D. Borman, 1999]) im Hop-by-Hop-Header gesetzt werden. Die maximal mögliche Größe eines IPv6-Paketes ist aufgrund der Länge des *Payload Length* Feldes auf 64 kB beschränkt. Mithilfe eines Jumbogramms können größere Pakete transportiert werden. Im IPv6-Header wird dazu im Feld *Payload Length* der Wert null angegeben sowie im *Next-Header*-Feld der Wert null, der anzeigt, dass ein

Hop-by-Hop-Extension-Header folgt. In diesem Erweiterungsheader wird der *Option Type* Wert 194 eingetragen, der ein Jumbogramm beschreibt. Das *Jumbo Payload Length*-Feld umfasst 32 Bits, womit nun Paketgrößen von 64 kB bis zu 4 GB möglich sind. Damit UDP und TCP mit den Jumbogrammen arbeiten können, gibt es für diese Protokolle Erweiterungen die im RFC 2675 [D. Borman, 1999] definiert sind.

Option Router Alert Das Resource Reservation Protocol (RSVP) und Multicast Listener Discovery (MLD) nutzen diese Option zur Überprüfung der Wegstrecke beim Routing. Sie zeigt einem Router an, dass ein Paket, welches nicht an den Router adressiert ist, Informationen für ihn bereithält.
Die im *Option Type*-Feld enthaltenen ersten drei Bits sind auf null gesetzt, damit das Paket stets weitergeleitet wird, wenn dem Router die Option nicht bekannt ist. In den restlichen fünf Bits ist der *Option Type* fünf definiert. Im darauffolgenden *Data Length*-Feld ist der Wert zwei eingetragen, welcher bedeutet, dass das nachfolgende Datenfeld zwei Byte lang ist. Dieses *Value*-Feld kann folgende Werte (RFC 2711 [C. Partridge, 1999]) [2] enhalten:

0 Das Paket enthält eine MLD-Nachricht.

1 Das Paket enthält eine RSVP-Nachricht.

2 Das Paket enthält eine Active Networks Nachricht.

3-35 Das Paket enthält einen Aggregated Reservation Nesting Level, RFC 3175 [F. Baker, 2001].

36-65535 Diese Werte sind von der IANA für zukünftige Verwendung reserviert.

Routing-Header Der Routing-Header enthält eine Liste von Routern entlang derer das Paket geleitet werden soll. Diese Angabe dient dazu, das Paket über einen bestimmten Pfad zum Ziel zu führen. Vergleichbar ist diese Option mit Loose Source/Record Route in IPv4. Im vorangehenden Header wird der Routing-Header mit dem Wert 43 im *Next-Header*-Feld angekündigt. Damit der definierte Weg eingehalten werden kann, müssen alle Router, die in der Liste eingetragen sind, diesen Header auch verarbeiten. Die Abbildung 2.3 zeigt die Struktur des Routing-Headers.

Wenn der Routing Typ dem verarbeitenden Router nicht bekannt ist, so wird das *Segments Left*-Feld überprüft und danach das weitere Verhalten bestimmt. Hat dieses Feld den Wert

[2] Eine aktuelle Liste der Werte ist unter `www.iana.org/assignments/ipv6-routeralert-values` einzusehen.

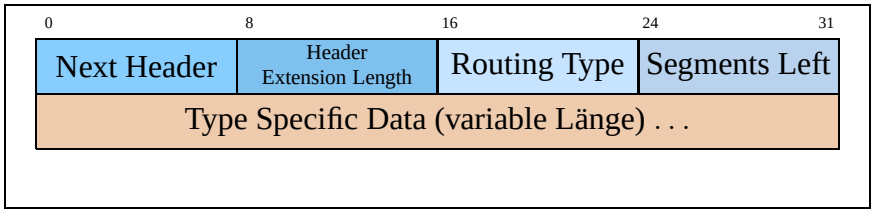


Abbildung 2.3: Struktur des Routing-Header

null, so wird der Routing-Header ignoriert und die Verarbeitung beim nächsten Header fortgesetzt. Sollte das Feld einen Wert verschieden von null besitzen, so wird das Paket verworfen und ein ICMPv6 Paket „Parameter Problem“ an den Absender geschickt, wobei der Pointer auf den *Routing Type* zeigt.

Fragment-Header Um festzustellen, wie groß ein Paket zu einem Empfänger sein darf, nutzt ein IPv6-Knoten Path-MTU-Discovery. Damit wird der komplette Pfad zum Empfänger durchlaufen und die maximale Paketgröße ermittelt. Dies ist notwendig, da IPv6-Router keine Pakete mehr fragmentieren. Nur der Sender fragmentiert ein Paket und der Empfänger baut es wieder zusammen. Sobald dies geschieht, wird ein Fragment-Header eingefügt. Der Wert zur Ankündigung eines Fragment-Header im *Next-Header*-Feld ist 44. In der Abbildung 2.4 ist der Aufbau des Fragment-Headers dargestellt.

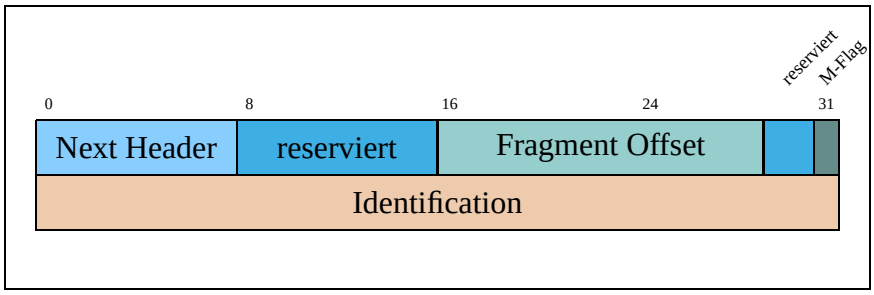


Abbildung 2.4: Gerüst des Fragment-Headers

Ablauf der Fragmentierung Der Fragmentierungsprozess ist in RFC 2460 [S. Deering, 1998] definiert. Jedes unfragmentierte Originalpaket besteht aus einem nicht fragmentierbaren Teil und einem fragmentierbaren Teil.

nicht fragmentierbarer Teil

Zu diesem Teil gehört der IPv6-Header, sowie die Extension-Header, die von jedem Router entlang des Paketpfades verarbeitet werden müssen (Hop-by-Hop-Options-Header, Destination-Options-Header mit Optionen für die Router, Routing-Header).

fragmentierbarer Teil

Zum fragmentierbaren Teil gehören die Erweiterungsheader, die für den endgültigen Empfänger bestimmt sind sowie die Upper-Layer-Header und die Daten.

Der unfragmentierte Teil ist in jedem einzelnen Fragment enthalten. Danach folgt der Fragment-Header und die fragmentierten Daten. Damit der Prozess der Fragmentierung so ablaufen kann, muss der IPv6-Header des Originalpaketes geändert werden. Das *Next-Header*-Feld des letzten unfragmentierbaren Teils des Headers muss auf den Wert 44 gesetzt werden.
Der Empfänger sammelt nun alle Fragmente und setzt diese wieder zusammen. Dazu müssen die Fragmente gleiche Absender- und Empfängeradressen sowie *Identification* Werte besitzen. Außerdem müssen sie innerhalb von 60 Sekunden nach dem Eintreffen des ersten Fragments ankommen, da sie ansonsten verworfen werden.

Destination-Options-Header Dieser Header hält Informationen bereit, die vom Empfänger des Pakets verarbeitet werden müssen. Er wird durch den Wert 60 im *Next-Header*-Feld des vorherigen Headers angekündigt. Wenn Optionen für die Router entlang des Paketpfades enthalten sein sollen, muss der Header vor den Routing-Header eingefügt werden. Sollen Optionen für den endgültigen Empfänger mitgeteilt werden, so muss sich dieser Header vor den Headern höherer Protokollschichten befinden. Da dieser bis zu zweimal in einem Paket vorkommen kann, können Optionen sowohl für Router als auch für den endgültigen Empfänger in einem Paket übermittelt werden. Den Aufbau des Headers veranschaulicht Abbildung 2.5.

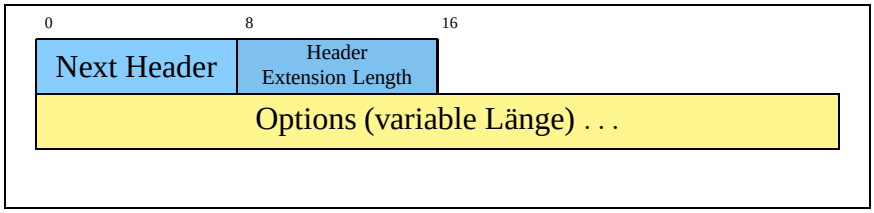


Abbildung 2.5: Aufbau des Destination-Options-Headers

Der Destination-Options-Header findet bei Mobile-IPv6 sowie bei der Tunnel Encapsulation Limit Option (RFC 2473 [A. Conta, 1998]) Verwendung.

2.2 Architektur der Adressierung

Ein bedeutender Punkt für die Entwicklung des neuen Internetprotokolls war die Adressknappheit. IPv4 hat ein Adressformat von 32 Bit, daraus ergeben sich 2^{32} Adressen. Dies

entspricht 4,29 Milliarden. Selbst bei optimaler Ausnutzung stünde nicht einmal für jede Person auf der Erde eine Adresse zur Verfügung. Zu Beginn der Ausgabe von IPv4-Adressen wurden relativ unorganisiert meist große Adressblöcke vergeben. Diese können nun nicht mehr zurückgefordert werden. Aus diesem Grund gibt es viele unbenutzte aber nicht verfügbare IPv4-Adressen.
Da in Zukunft nicht nur jede Person bzw. jeder Computer eine, viele neue Netzwerk-, Kommunikations- und Heimelektronikgeräte meistens sogar mehrere IP-Adressen benötigen, steht außer Frage, dass IPv4 dafür auf keinen Fall genügend Adressen zur Verfügung stellen kann.
Der Adressraum für IPv6-Adressen basiert auf 128 Bits. Dies entspricht 2^{128}, also $3,4 * 10^{38}$ Adressen, die ausreichen würden um jedem Quadratmeter Erdoberfläche $6,65 * 10^{23}$ Adressen zu vergeben.
Auch die Anzahl der Netzwerk-IDs ist bei IPv4 durch die zu Beginn definierten Adressklassen (A,B,C,D,E) auf 2113389 beschränkt gewesen. Mit der Einführung von CIDR wurden diese noch nicht ausreichend erweitert.
Bei IPv6 gibt es ein Präfix für Globale-Unicast-Adressen, so dass 2^{45} Netzwerk-IDs möglich sind. Zusätzlich hat jedes dieser Netzwerke noch 16 Bits für die Unterteilung in weitere (65536) Subnetze zur Verfügung. Weitere grundlegende Informationen zum Aufbau des IPv6-Adressraums befinden sich im RFC 4291 [R. Hinden, 2006].

2.2.1 Adresstypen

Da IPv6 kein Broadcast mehr unterstützt, wird dies zum Teil durch Multicasts ersetzt. Außerdem soll der Anycast mehr Verwendung finden (RFC 1546 [C. Partridge, 1993]). Es bleiben also noch 3 Kategorien von Adresstypen übrig:

Unicast

Mithilfe des Unicasts kann ein Interface eines IPv6-Knotens genau lokalisiert werden. Ein Paket, welches an eine solche Adresse geschickt wird, kommt also genau an dem Interface an, welches an diese Adresse gebunden wurde.

Multicast

Eine Multicast-Adresse identifiziert eine ganze Gruppe von IPv6-Interfaces. Das bedeutet, wenn ein Paket an diese Adresse geschickt wird, empfangen alle Mitglieder dieser Gruppe das Paket.

Anycast

Eine Adresse diesen Typs kann mehreren Interfaces, oftmals auf verschiedenen

Knoten, zugeordnet sein. Ein an eine Anycast-Adresse geschicktes Paket wird nur an eines der Interfaces gesendet, meist an das bezüglich des Routings am nächsten liegende.

Für die Adressierung gilt, wie auch bei IPv4, dass Interfaces adressiert werden und keine Knoten. Außerdem muss jedes Interface mindestens eine Unicast-Adresse besitzen. Bei IPv6 ist es zudem möglich, einem Interface mehrere IPv6-Adressen zuzuweisen. Dabei ist eine Kombination aus Unicast-, Multicast und Anycast-Adressen möglich. Ein Knoten kann ein Paket über jede Unicast-Adresse empfangen, die einem seiner Interfaces zugeordnet ist. Zur Lastverteilung (Load-Sharing) kann eine Unicast-Adresse mehreren Interfaces zugeordnet werden. Dabei ist es wichtig, dass sowohl die Hardware als auch der Treiber Load-Sharing unterstützen.

2.2.2 Globale Routing Präfixe

Die globalen Routing Präfixe sind im RFC 4291 [R. Hinden, 2006] definiert. Der größte Teil des Adressbereiches von IPv6 ist noch nicht vergeben. Es bleibt also genügend Raum für die Zukunft. In Tabelle 2.2 befinden sich die schon zugewiesenen Präfixe. Durch die IANA werden im Moment nur Adressen aus dem Bereich, der mit 0b001 beginnt, vergeben. [3] Dies entspricht in etwa einem Achtel des gesamten Adressraums von IPv6. Weiterhin wurden spezielle Adressen mit dem Präfix 0b0000 0000 vergeben. Dazu gehören die unspezifizierten Adressen, die Loopback-Adresse sowie IPv6-Adressen, in die eine IPv4-Adresse eingebettet werden kann.
Unicast-Adressen besitzen den Präfix 0b001. Dagegen beginnt eine IPv6-Multicast-Adresse mit 0xff. Die schon erwähnten Anycast-Adressen besitzen den selben Präfix wie die Unicast-Adressen. Sobald eine Unicast-Adresse mehr als einem Interface zugeordnet ist, wird sie automatisch zu einer Anycast-Adresse.
Alle IPv6-Adressen, die nicht mit 0b000 beginnen und keine Multicast-Adressen sind, benötigen eine Interface-ID nach dem EUI-64 Format, welches durch das IEEE festgelegt wurde. Erläuterungen zum EUI-64 Format befinden sich in [IEEE].

2.2.3 Globale Unicast-Adressen

Im Moment werden globale Unicast-Adressen aus dem Adressbereich mit dem Präfix 0b001 vergeben. Zukünftig können auch weitere Adressbereiche freigegeben werden,

[3] Eine aktuelle Liste der Reservierungen ist unter `www.iana.org/assignments/ipv6-address-space` zu finden.

Adressen	Präfix Hexadezimal	Präfix Binär
Multicast-Adressen	0xff00::/8	0b1111 1111
Link-lokale Unicast-Adressen	0xfe80::/10	0b1111 1110 10
Unique-Lokale IPv6-Adressen	0xfc00::/7	0b1111 1101
Globale Unicast-Adressen	0x2000::/3	0b001

Tabelle 2.2: Zugewiesene Präfixe

wenn dieser erschöpft ist. Der Aufbau einer globalen Unicast-Adresse ist in Abbildung 2.6 zu sehen. Es ist zu erkennen, dass die Adresse aus den drei Teilen *globales Routing Präfix, Subnetz-ID* und *Interface-ID* besteht. Dieses Format ist in RFC 4291 [R. Hinden, 2006] beschrieben.

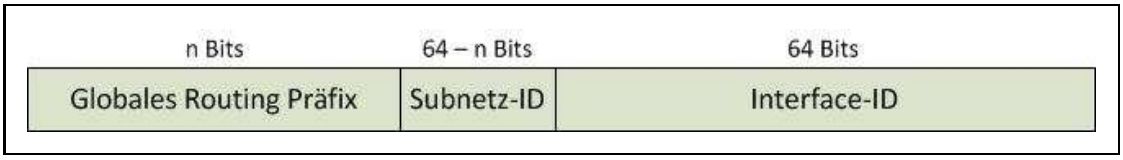


Abbildung 2.6: Format der globalen Unicast-Adresse

Die Elemente der Adresse sind im Einzelnen:

globales Routing Präfix

Es beschreibt den einer Site zugewiesenen Adressbereich. Er wird durch die Provider (ISPs) und die globalen Registrierungsstellen hierarchisch strukturiert. Der Begriff Site beschreibt gemäß RFC 3587 [R. Hinden, 2003] einen sogenannten Cluster von Subnetzen. Dies entspricht dem Standort eines Netzwerks (z.B. Unternehmens- oder Campusnetzwerk).

Subnetz-ID

Diese ID wird auch Subnetzpräfix oder nur Subnetz genannt. Die Strukturierung dieses Bereiches wird durch den Administrator vorgenommen, dabei können einem Link auch mehrere Subnetz-IDs zugewiesen werden.

Interface-ID

Zur Identifizierung eines Interfaces in einem Subnetz dient diese ID. Sie ist innerhalb des jeweiligen Subnetzes einmalig.

2.2.4 Registrierungsstellen-Adresszuteilungen

Die Verteilung von IPv6-Adressen ist durch die IANA an verschiedene internationale Registrierungsstellen weitergegeben worden. Diese verteilen die Adressen regional, um Rou-

tingtabellen der Internet Core Router durch geschickte Präfixvergabe zu optimieren. Die zuständigen Registrierungsstellen sind die folgenden:

- **ARIN** (American Registry for Internet Numbers) für Nordamerika
- **RIPE NCC** (Réseau IP Européens Network Coordination Centre) für Europa
- **APNIC** (Asia Pacific Network Information Centre) für Asien
- **LACNIC** (Latin American and Caribbean Internet Addresses Registry) für Lateinamerika
- **AfriNIC** (African Network Information Centre) für Afrika

Die von diesen Organisationen vergebenen Adressbereiche werden nicht einmalig vergeben, sondern können bei Bedarf wieder zurückgerufen werden, wenn dazu eine technische Notwendigkeit bestehen sollte. Bisher vergebene Adressbereiche sind in der Tabelle 2.3 ersichtlich. [4]

Zuweisung	**Präfix**
Globaler IPv6 Adressraum	2000::/3
Dokumentation, kein Routing	2001:db8::/32
6to4	2002::/16
Teredo	2001::/32

Tabelle 2.3: Zugewiesene Adressbereiche

Die meisten Organisationen und Endanwender erhalten ihre IPv6-Adressen von ihrem ISP. Dieser bekommt die Adressbereiche von der zuständigen regionalen Registrierungsorganisaton. Die Zuteilung von Adressen an Organisationen, Endanwender sowie ISPs ist ein anhaltender Prozess, der nach den Empfehlungen in RFC 3177 [IESG, 2001] vollzogen werden sollte. Die wichtigsten Regeln im Einzelnen:

- Organisationen erhalten /48 Präfixe.
- Große Teilnehmer erhalten in Ausnahmefällen kürzere oder mehrere /48 Präfixe.
- /64 Subnetze werden vergeben, wenn mit großer Sicherheit nur ein Subnetz verbunden werden soll.

[4] Aktuelle Informationen zur Adresszuteilung und IP-Dienste sind unter `www.iana.org/numbers` abrufbar.

- Wenn nur ein Gerät verbunden werden soll, wird ein /128 Präfix vergeben.
- Mobile Netzwerke sollen ein /48 Präfix erhalten, um mehrere Geräte anschließen zu können.

Die Verteilung der Adressbereiche durch die ISPs erfolgt nach den jeweiligen Bedürfnissen, sollte sich aber an den Empfehlungen orientieren. Im neueren RFC 6177 [T. Narten, 2011] wird von den oben genannten Empfehlungen wieder Abstand genommen und es können auch Präfixe mit variabler Länge vergeben werden. Im Großen und Ganzen räumt der RFC 6177 [T. Narten, 2011] den ISPs mehr Flexibilität bei der Vergabe von Adressbereichen und Adressen ein. Diese Aktualisierung zeigt aber auch, dass mit vermehrtem Interesse an IPv6 immer wieder Änderungen und neue Empfehlungen entstehen, die aus praktischer Erfahrung herrühren.
In begründeten Ausnahmefällen kann eine große Organisation, der ein /48 Präfix mit 65536 Subnetzen nicht ausreichend erscheint, auch ein kürzeres Präfix erhalten.

2.2.5 Interface-ID / EUI-64 Format

Die Generierung der Interface-ID folgt dem EUI-64 Format. Dabei muss bei der Autokonfiguration eines IPv6-Knotens aus der 48-Bit Interface MAC-Adresse ein 64-Bit Identifier generiert werden. Dazu wird 0xfffe zwischen dem dritten und vierten Byte der MAC-Adresse eingefügt. Danach wird das Bit gesetzt, welches sich an zweiter Stelle des ersten Bytes der MAC-Adresse befindet. Eine Link-lokale-Adresse eines bestimmten Interfaces ist die Kombination des Präfixes fe80::/64 und diesem generierten 64-Bit Identifier. Die Vorgehensweise ist zudem in RFC 2464 [Crawford, 1998] beschrieben.

Schutz der Privatsphäre Die Generierung der IPv6-Adresse bei der Autokonfiguration aus der MAC-Adresse des Interfaces sorgte sofort nach Bekanntwerden dieser Vorgehensweise für Diskussion. Die Privatsphäre wäre gefährdet, da sich ein Benutzer mit einer Adresse, die vom MAC-Identifier abgeleitet wird, einfacher nachverfolgen ließe.
Die Verwendung des Interface Identifiers ist keine Anforderung an eine IPv6-Adresse, die ein Knoten besitzt. Ebenso gut kann dieser eine statische, also manuell konfigurierte Adresse oder eine per DHCP konfigurierte Adresse besitzen. Weiterhin hat die IETF ein RFC herausgebracht, welches dieses Thema ausführlich behandelt. In RFC 4941 [T. Narten, 2007a] ist eine neue Vorgehensweise zur Generierung des Interface Identifiers erläutert. Dieser wird mithilfe der *Privacy Extension* durch eine periodisch erzeugte Zufallszahl und nicht über die MAC-Adresse erzeugt. Diese Erweiterung sollte natürlich

nicht für Server verwendet werden, da diese eine eindeutige Adresse benötigen.
Damit ist eine Unterscheidung in zwei Adresstypen möglich:

- **Stabile IP-Adressen**
  Diese Adressen werden über eine manuelle Konfiguration, die Verwendung des DHCP oder über die Autokonfiguration mittels Interface Identifier zugewiesen.
- **Vorübergehende IP-Adressen**
  Temporäre Adressen, die mit einer Zufallszahl nach RFC 4941 [T. Narten, 2007a] zugewiesen werden.

2.2.6 Spezielle Adressen

Spezielle Adressen sind zum einen Adresstypen, die für die Migration zum neuen Internetprotokoll geschaffen wurden, und zum anderen Adressen, die gesonderte Aufgaben erfüllen, wie zum Beispiel die Loopback-Adresse.

IPv6-Adressen die IPv4-Adressen enthalten Für die Umstellungsphase von IPv4 auf IPv6 sind zwei Adresstypen in RFC 4291 [R. Hinden, 2006] definiert, die die Rückwärtskompatibilität ermöglichen.

IPv4 mapped IPv6-Adresse
Zur Darstellung von Adressen von reinen IPv4-Knoten als IPv6-Adresse kann dieser Adresstyp verwendet werden. Dabei enthält die IPv6-Adresse in den letzten 32 Bits die IPv4-Adresse. Dieser IPv4-Adresse wird 0xffff vorangestellt. Die ersten 80 Bits der Adresse enthalten nur Nullen.

IPv4 compatible Ipv4-Adresse
Dieser Adresstyp ist in der Entwicklung des RFC 4291 [R. Hinden, 2006] abgeschafft worden. Er wurde genutzt, um IPv6-Pakete über IPv4 in einer IPv4-Infrastruktur zu transportieren.

Teredo-Adressen Der Teredo-Mechanismus ermöglicht es Hosts, die sich hinter IPv4-NATs befinden, einen Tunnel zu IPv6-Netzwerken zu errichten. Ein genaue Beschreibung der Vorgehensweise ist in Abschnitt 3.2.1 zu finden. Teredo-Adressen besitzen ein 32 Bits langes Präfix mit dem Wert 2001:0::/32. In den darauffolgenden 32 Bits befindet sich die IPv4-Adresse des Teredo-Servers. Daran schließen sich jeweils 16 Bits für den Adress-

und NAT-Typ sowie der verwendete Port an. In den verbleibenden 32 Bits befindet sich die IPv4-Adresse des Teredo-Clients.

ISATAP-Adressen ISATAP beschreibt eine automatische Vorgehensweise für den Aufbau eines Tunnels, der IPv6-Knoten eine IPv6-Kommunikation über eine IPv4-Infrastruktur erlaubt. Eine Beschreibung von ISATAP ist in Abschnitt 3.2.1 und im RFC 5214 [F. Templin, 2008] zu finden. Die ersten 64 Bits enthalten wie andere Unicast-Adressen das offizielle Präfix. Die darauffolgenden 32 Bits enthalten den sogenannten IEEE Organizationally Unique Identifier (OUI) der IANA (0x00 00 5e). Die ersten 16 Bits dieses OUI zeigen an, ob es sich um eine private (0x0000) oder eine öffentliche (0x0200) IPv4-Adresse handelt. Darauf folgen 8 Bits, die den Identifier 0xfe beinhalten. Er zeigt an, dass es sich um eine IPv6-Adresse mit eingebetteter IPv4-Adresse handelt. Die letzten 32 Bits der ISATAP-Adresse enthalten die eingebettete IPv4-Adresse in dezimaler oder hexadezimaler Schreibweise.

6to4-Adressen Damit IPv6-Knoten ohne statisch konfigurierte Tunnel mit Knoten in anderen IPv6-Netzwerken über eine IPv4-Infrastruktur kommunizieren können, wurde der das Migrationsverfahren 6to4 entwickelt. Beschrieben wird dieses Verfahren in Abschnitt 3.2.1 und im RFC 3056 [B. Carpenter, 2001]. Das Präfix 2002::/16 kennzeichnet eine 6to4-Adresse. Des Weiteren gehört zum Gesamtpräfix der 6to4-Adresse, die 48 Bits enthält, die offizielle IPv4-Adresse in hexadezimaler Schreibweise. Darauf folgen 16 Bits für die Subnetzaufteilung. Die verbleibenden 64 Bits repräsentieren wie bei Unicast-Adressen üblich die Interface-ID.

unspezifizierte Adresse Die unspezifizierte oder auch ALL-ZERO Adresse hat den Wert *0:0:0:0:0:0:0:0*. Sie ist im Grunde mit der IPv4-Adresse *0.0.0.0* vergleichbar. Mit ihr wird angezeigt, dass keine gültige Adresse vorhanden ist. Sie wird z.B. beim Booten eines Gerätes als Absenderadresse verwendet, wenn dieser eine Adressanfrage stellt. Nach den Abkürzungsregeln kann sie auch als *: :* geschrieben werden.

Loopback-Adresse Die Loopback-Adresse entspricht der IPv4-Adresse *127.0.0.1* . Bei IPv6 hat sie den Wert *0:0:0:0:0:0:0:1* oder in kurzer Schreibweise *::1*. Eingesetzt wird diese Adresse zum Testen des lokalen IP-Stacks, da dabei kein Paket ins Netz geschickt wird.

Kryptografisch generierte Adresse CGA (Cryptographically Generated Addresses) werden beim Neighbor Discovery (ND) eingesetzt, um die Sicherheit dieses Verfahrens zu verbessern. Dabei wird in der Interface-ID eine kryptografisch generierte Einwege-Hash-Funktion berechnet und integriert. Durch dieses Verfahren ist es möglich ND zu sichern, ohne dass eine Sicherheitsinfrastruktur vorhanden ist.

2.2.7 Site- und Link-lokale-Adressen

In Unternehmensnetzwerken werden meist IP-Adressen aus dem privaten Bereich (192.168.1.1) benutzt, wie sie in RFC 1918 [Y. Rekhter, 1996] spezifiziert sind, da sie von keinem Router ins Internet weitergeleitet, aber im Firmennetzwerk problemlos geroutet werden können. Um eine Verbindung ins Internet zu ermöglichen, wird NAT eingesetzt. Damit ist es möglich, für viele private IP-Adressen einen Zugang zum Internet über eine einzige offizielle IP-Adresse herzustellen.
Private Adressen bei IPv6 sind Link-lokal und lokale Adressen. Sie verwenden die Präfixe 0xfe80 bzw. 0xfd00 (siehe Tabelle 2.2). Da jedes IPv6-Interface eine Link-lokale-Adresse besitzen muss, wird diese automatisch konfiguriert.
Lokale IPv6-Adressen (Unique Local IPv6 Unicast Address (RFC 4193 [R. Hinden, 2005])) sollen, wie private IPv4-Adressen, nicht ins Internet geroutet werden. Ihre Verwendung ist auf Unternehmensnetzwerke oder Teile eines Netzwerks beschränkt. Sie haben die folgenden Eigenschaften:

- Zur Erleichterung der Filterung an Site-Grenzen besitzen sie ein global eindeutiges Präfix.
- Mit ihnen können Sites privat verbunden werden, ohne dass die Gefahr besteht, einen Adresskonflikt herbeizuführen oder ein Netzwerk neuadressieren zu müssen.
- Auch wenn keine Internetverbindung vorhanden ist, kann mit diesen Adressen eine interne Kommunikation unabhängig vom ISP bewerkstelligt werden.
- Die Benutzung erfolgt genauso wie die von globalen Unicast-Adressen.

Die Zuweisung dieser Adressen wird durch jedes Unternehmen selbst durchgeführt. Dazu wird das Präfix *fd00::/8* verwendet. Einzig besteht dadurch die Gefahr, dass bei einer späteren Verbindung zu anderen Unternehmen dieselben IDs verwendet werden (wie auch bei IPv4). Die Wahrscheinlichkeit, dass dieser Fall auftritt, ist aufgrund des großen Adressraums sehr gering. Für private Adressbereiche hat SixXs [5] eine Registrierungsmöglichkeit

[5] SixXs ist ein freier, kostenloser und unabhängiger Internetregistierungsservice für ISPs und Endanwender.

geschaffen. Diese lokalen IPv6-Adressen dürfen natürlich nicht zwischen Sites und dem Internet geroutet werden. Weiterhin muss darauf geachtet werden, dass DNS-Einträge mit lokalen Adressen nicht in globale DNS-Server eingetragen werden.
Link-lokale-Adressen werden im Normalfall per Autokonfiguration vergeben. Lokale IPv6-Adressen werden hingegen durch das Eintragen des lokalen Präfix im Router (Router Advertisement) oder durch DHCPv6 zugewiesen.
Link-lokale-Adressen werden nur am lokalen Link verwendet, um Autokonfiguration und Neighbor Discovery in Netzwerken ohne Router zu realisieren. Verwendung findet dies z.B. bei temporären Netzwerken (Ad-hoc WLAN).

2.2.8 Anycast-Adressen

Anycast-Adressen finden Verwendung bei Load-Balancing und redundanten Geräten, also dort, wo mehrere Server oder auch Router den selben Dienst bereitstellen. Schon für IPv4 in RFC 1546 [C. Partridge, 1993] war dieser Adresstyp vorgesehen. Zur Realisierung soll ein spezielles Präfix für den Anycast verwendet werden, damit er von einer Unicast-Adresse unterschieden werden kann. Zum Einsatz sollte Anycast damals vor allem für DNS und HTTP kommen. Auch wird im RFC erwähnt, wie TCP verändert werden muss, damit es mit diesem Adresstyp umgehen kann.
Umgesetzt wurde dieses Verfahren jedoch nicht. Als Alternative wurde die Shared-Unicast-Adresse verwendet. Dabei wird eine Unicast-Adresse mehreren Interfaces zugewiesen. Dazu sind außerdem mehrere Einträge in den Routingtabellen vorzunehmen. Durch dieses Verfahren wird immer das Interface erreicht, das bezüglich der Metrik am nächsten liegt.
Probleme, die durch die mehrdeutigen Adressen entstehen, müssen durch die verwendenden Anwendungen entsprechend behandelt werden. Schwierigkeiten ergeben sich dadurch, dass sowohl Netzwerk- als auch Transportschicht von global eindeutigen Adressen ausgehen. Sogenannte *stateless*, also einfache, voneinander unabhängige Transaktionen (z.B. DNS über UDP) sind davon nicht beeinträchtigt.
Bei IPv6 besitzen Anycast-Adressen kein spezielles Präfix. Zudem sind sie im selben Adressbereich angesiedelt wie die Unicast-Adressen. In der Konfiguration eines jeden Interfaces muss angegeben werden, dass es sich um eine Anycast-Adresse handelt. Wenn in einem Umfeld mehrere Interfaces mit derselben Anycast-Adresse existieren, muss jeder Host einen eigenen Eintrag in der Routingtabelle bekommen. Falls es zudem keine klar definierte Region zu einem Interface gibt, muss dieses im Internet bekannt gemacht werden. Da dies nicht skaliert, wird es keine oder kaum Unterstützung für globale Anycast-Adressen geben.

Einer Gruppe von Routern kann eine einzige Adresse zugewiesen werden für den Fall, dass diese zu einer gemeinsamen Routing-Domain gehören. Ein Client schickt nun sein Paket an eine Anycast-Adresse, sodass es zum nächsten erreichbaren Router gesendet wird.
Zu beachten ist, dass der Absender bei der Verwendung des Anycast keine Kontrolle darüber hat, an welches Interface ein Paket wirklich ausgeliefert wird. Die Entscheidung dazu wird auf der Routingprotokollebene getroffen. Somit kann es vorkommen, dass mehrere Pakete in Folge, die an eine Anycast-Adresse gesendet wurden, bei verschiedenen Empfängern ankommen. Aus diesem Grund kann es bei einer Serie von Anfragen und Antworten oder bei der Fragmentierung zu Problemen kommen.

Subnet-Router-Anycast-Adresse Diese Adresse zeichnet sich durch einen Aufbau ähnlich einer Unicast-Adresse mit Subnetzpräfix aus, wobei beim Interface Identifier alle Bits den Wert null besitzen. Ein Paket an eine solche Adresse wird an einen beliebigen Router, der sich im Subnetz des Senders befindet, weitergeleitet. Router müssen diese Adresse in allen Subnetzen, in denen sie ein Interface besitzen, unterstützen.

Weitere Informationen zum Anycast-Adresstyp und reservierten Anycast-Adressen befinden sich im RFC 2526 [D. Johnson, 1999]. Eine reservierte Subnetz-Anycast-Adresse kann zwei Formate haben (siehe Abbildung 2.7). Die reservierten Anycast-IDs sind in Tabelle 2.4 zu finden.

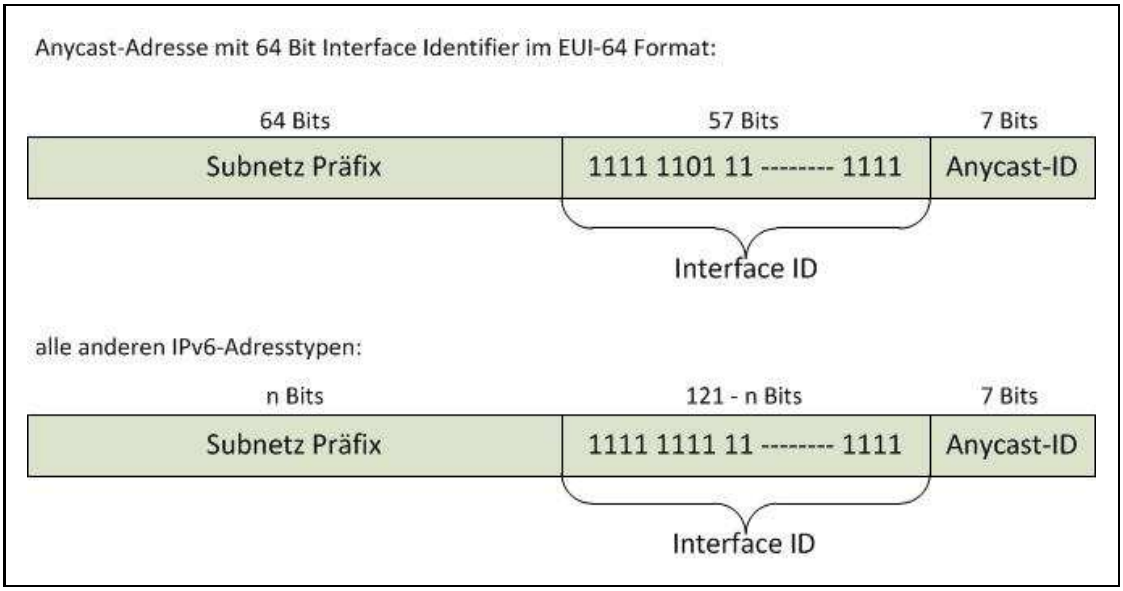


Abbildung 2.7: Struktur der Anycast-Adresse

Der wesentliche Unterschied zwischen Anycast- und Shared-Unicast-Adressen ist der, dass bei Shared-Unicast-Adressen die Anwendung den Anycast unterstützen muss, hingegen bei Anycast-Adressen versucht wird, dies so wenig wie möglich zu fordern. Zur optimalen Verwendung von beiden sind weitere Anforderungen zu definieren, insbesondere Veränderungen, die an *stateful* Protokollen notwendig sind.

Beschreibung	**Hexadezimal**	**Dezimal**
reserviert	0x7f	127
mobile IPv6 Home Agent Anycast	0x7e	126
reserviert	0x00 - 0x7d	0 - 125

Tabelle 2.4: Reservierte Anycast-IDs

2.2.9 Multicast-Adressen

Um eine Gruppe von Interfaces anzusprechen, werden Multicast-Adressen verwendet. Ein an eine Multicast-Adresse gesendetes Paket wird an alle Gruppenmitglieder der Multicastgruppe gesendet. Zu erkennen ist eine solche Adresse am Wert 0xff des hochwertigen Bytes. Mit der Entwicklung von IPv6 wurde der Multicast überarbeitet und erweitert. Die Abbildung 2.8 zeigt den Aufbau einer IPv6-Multicast-Adresse. Das erste Byte mit dem Wert 0xff zeigt die Multicast-Adresse an. Die diesem Byte folgenden 4 Bits dienen als Flags.

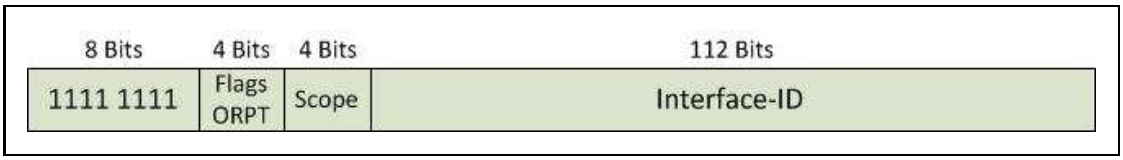


Abbildung 2.8: Format der Multicast-Adresse

Das nach den Flags folgende 4 Bits lange *Scope*-Feld ermöglicht das Begrenzen der Reichweite eines Multicasts. Die Werte, die dieses *Scope*-Feld annehmen kann, sind in Tabelle 2.5 aufgeführt. *Scopes*, die als reserviert gekennzeichnet sind, dürfen nicht verwendet werden. Zur Vereinbarung individueller Reichweiten kann der Administrator die nicht zugewiesenen *Scopes* verwenden.

Reguläre Multicast-Adressen Reguläre Multicast-Adressen werden in RFC 3306 [B. Haberman, 2002] und RFC 3307 [Haberman, 2002] näher definiert. Ebenso sind darin Empfehlungen für die Vergabe von neuen Multicast-Adressen und Gruppen-IDs enthalten. Die Multicast-Gruppen-ID ist in den letzten 112 Bits der Adresse enthalten. Die schon

zugewiesenen, dauerhaften Multicast-Adressen mit einer definierten Reichweite (Scope) sind der Tabelle 2.6 zu entnehmen. Multicast-Adressen, die keinen definierten Scope aufweisen und dauerhaft vergeben sind, gibt es so viele, dass eine Aufzählung hier nicht möglich ist. [6]
Die Subnetz-Broadcast-Adressen aus IPv4 gibt es bei IPv6 nicht mehr. Die Link-lokal-IPv4 Broadcast-Adressen werden bei IPv6 durch die Link-Local-All-Nodes Multicast-Adresse ff02::1 realisiert. Die temporär zugewiesenen Multicast-Adressen sind nur in einer begrenzten Reichweite sinnvoll. Multicast-Adressen dürfen auch nicht als Absenderadressen in Routing-Headern oder normalen IPv6-Headern vorkommen. Die Verwaltung der Multicast-Adressen wird mittels IPv6 ICMPv6 durchgeführt.

Solicited-Node-Multicast-Adresse Bei einer solchen Multicast-Adresse muss sich jeder Knoten mit jeder seiner Adressen (Unicast-/Anycast-Adresse) registrieren. Sie werden verwendet, wenn MAC-Adressen aufzulösen sind und um festzustellen, ob eine IPv6-Adresse schon von einem anderen Knoten genutzt wird. Die Definition dieses Adresstyps befindet sich in RFC 4291 [R. Hinden, 2006].
Bei IPv4 wird der ARP-Request an die Broadcast-Adresse der MAC-Schicht gesendet und somit erhält jedes Interface im Segment eine Anfrage. Dies wird bei IPv6 erleichtert. Hier wird zur Auflösung der MAC-Adresse eine Neighbor Solicitation Nachricht an die Solicited-Node Multicast-Adresse und nicht an die Link-lokal All-Nodes Multicast-Adresse gesendet.
Die Adresse besteht aus den 24 niederwertigen Bits einer IPv6-Adresse und dem Präfix ff02:0:0:0:0:1:ff00::/104. Daraus ergibt sich der Bereich für die Solicited-Node Multicast-Adresse von ff02::1:ff00:0000 bis zu ff02::1:ffff:ffff.

Mapping von Multicast-Adressen auf MAC-Adressen Eine Multicast-Adresse muss auf dem Link-Layer in eine MAC-Adresse umgewandelt werden, wenn ein Paket an eine IPv6 Multicast-Adresse gesendet werden soll. Wie dies abläuft wird in RFC 2464 [Crawford, 1998] beschrieben. Fest stehen die ersten zwei Byte der IPv6-MAC-Multicast-Adresse (0x3333). Die darauf folgenden vier Bytes bestehen aus den letzten vier Bytes der Multicast-Adresse. Aus genau diesem Grund kann eine Einschränkung der Multicast-Gruppen-IDs auf 32 Bits sich als sinnvoll herausstellen. So kann ausgeschlossen werden, dass zwei Multicast-Gruppen dieselbe MAC-Multicast-Adresse besitzen.

[6] Eine aktuelle Liste mit zugewiesenen Multicast-Adressen ist unter `www.iana.org/assignments/ipv6-multicast-addresses` abrufbar.

Scope	Wert
reserviert	0
Interface-local Scope	1
Link-lokaler Scope	2
reserviert	3
Admin-local Scope	4
Site-local Scope	5
ohne Zuweisung	6,7
Organization-local Scope	8
ohne Zuweisung	9, a, b, c, d
Global Scope	e
reserviert	f

Tabelle 2.5: Multicast-Scopes

Adresse	Beschreibung
Interface-local Scope	
ff01:0:0:0:0:0:0:1	All Nodes Adresse
ff01:0:0:0:0:0:0:2	All Router Adresse
Link-lokaler Scope	
ff02:0:0:0:0:0:0:1	All Nodes Adresse
ff02:0:0:0:0:0:0:2	All Router Adresse
ff02:0:0:0:0:0:0:3	ohne Zuweisung
ff02:0:0:0:0:0:0:5	OSPFIGP
ff02:0:0:0:0:0:0:6	OSPFIGP Designated Router
ff02:0:0:0:0:0:0:9	RIP Router
ff02:0:0:0:0:0:0:a	EIGRP Router
ff02:0:0:0:0:0:0:f	Universal Plug and Play
ff02:0:0:0:0:0:0:16	MLDv2-fähige Router
ff02:0:0:0:0:0:1:2	DHCP-Agents
ff02:0:0:0:0:0:1:3	Link-lokale Multicast Namensauflösung
Site-local Scope	
ff05:0:0:0:0:0:0:2	All Router Adresse
ff05:0:0:0:0:0:1:3	All DHCP Servers

Tabelle 2.6: Beispiele einiger Multicast-Adressen

Dynamische Adresszuordnung von Multicast-Adressen Die dynamische Zuordnung von Multicast-Adressen erfordert eine Vielzahl von Protokollen. Mit der neuen Definition durch RFC 3306 [B. Haberman, 2002] wird die Adressarchitektur von Multicast-Adressen in IPv6 so erweitert, dass dies mit wesentlich geringerem Aufwand möglich ist. Außerdem sind nun Source-Specific Multicast-Adressen verwendbar. Dabei basiert die Adresszuordnung auf einer veränderten Multicast-Adresse, welche zusätzliche Präfixinformationen beinhaltet. Eine Übersicht zum Source-Specific-Multicast (SSM) ist im RFC 3569 [S. Bhattacharyya, 2003] zu finden. Damit ist es möglich, ein Interface für eine Multicast-Adresse zu registrieren und zusätzlich noch die Quellen anzugeben, von denen Pakete empfangen bzw. nicht empfangen werden sollen. Dieser Source-Specific Multicast ist erst durch das erweiterte Multicast-Adressformat möglich.

2.2.10 Obligatorische Adressen

In der Definition wird festgelegt, dass jeder Knoten die folgenden Adressen für jedes ihm zur Verfügung stehende Interface kennen muss:

- Link-lokale-Adresse für jedes seiner Interfaces
- alle zusätzlich zugewiesenen Unicast- und Anycast-Adressen
- die All-Nodes Multicast-Adresse
- seine Loopback-Adresse
- alle Multicast-Adressen von den Gruppen zu denen der Knoten gehört
- die Solicited-Node Multicast-Adresse für jede der zugeordneten Unicast- oder Anycast-Adressen

Zusätzlich dazu müssen Router weitere Adressen kennen. Diese sind im Folgenden aufgeführt:

- alle All-Router Multicast-Adressen,
- alle eingerichteten Anycast-Adressen und
- Subnet-Router Anycast-Adresse für alle seine Interfaces, die er als Router bedient.

2.2.11 Adressauswahl

Ein IPv6-Interface hat laut Spezifikation mehrere Adressen. Diese können sich in folgenden Punkten unterscheiden:

- Reichweite (Scope) (Link-lokal oder global)
- Status (preferred oder deprecated)
- virtuelle Tunnel-Interfaces oder offizielle dauerhafte IPv6-Adressen
- bei Verwendung von Dual-Stack eine IPv4 und mehrere IPv6-Adressen

Die jeweiligen Netzwerkanwendungen müssen nun herausfinden, über welche Adresse sie eine Verbindung aufbauen können. Als Beispiel dient ein Client, der eine private IPv4-Adresse und eine globale IPv6-Adresse besitzt. Hier sollte die Verbindung bevorzugt über die IPv6-Adresse erfolgen. Hat der Client aber eine Link-lokale-IPv6-Adresse und eine offizielle IPv4-Adresse, so muss der Verbindungsaufbau über die IPv4-Adresse erfolgen. Im RFC 3484 [Draves, 2003] werden zwei Algorithmen vereinbart, zum einen die Source-Address-Selection und zum anderen die Destination-Address-Selection. Alle IPv6-Geräte müssen diese implementieren. Diese Algorithmen regeln das standardmäßige Verhalten im Umgang mit der Adressauswahl. Die wichtigsten Regeln dazu sind im Folgenden aufgeführt:

- Adresspaare mit der selben Reichweite oder dem selben Typ sind bevorzugt
- wenn eine Adresse als *preferred* gekennzeichnet ist, muss sie auch bevorzugt verwendet werden.
- Adressen, die für Migrationsmechanismen(ISATAP, 6to4) verwendet werden, sollen nicht benutzt werden, wenn native IPv6-Adressen vorhanden sind
- ein kleiner Scope für eine Ziel-Adresse wird einem größeren bevorzugt
- für den Fall, dass alle Kriterien ähnlich sind, sollten die Adresspaare verwendet werden, die den längsten übereinstimmenden Präfix besitzen
- für die Quelladresse sind globale Adressen den temporären vorzuziehen
- bei Mobile-IP sind Home-Adressen den Care-of-Adressen zu bevorzugen.

Diese Regeln werden angewendet, wenn keine anderen Definitionen der jeweiligen Anwendungen vorliegen. In der Definition ist weiterhin vorgesehen, eine spezielle Konfiguration zu erstellen, die eine bevorzugte Kombination von Quell- und Zieladresse bereitstellt und die standardmäßige Konfiguration überschreibt. Trotzdem kommt es in einigen Umgebungen immer wieder zu Problemen. Die weitestgehend bekannten sind in RFC 5220 [A. Matsumoto, 2008a] beschrieben. Eine Vorgehensweise, um diese Probleme zu verhindern, ist in Arbeit. Die Anforderungen an eine solche Lösung sind im RFC 5221 [A. Matsumoto, 2008b] niedergeschrieben.

2.3 ICMP Version 6

ICMP ist das wohl am meisten genutzte Protokoll, wenn es um das Auffinden und Beheben von Fehlern in einem Netzwerk geht. Es gibt wichtige Informationen zum Zustand des Netzwerks und ermöglicht einen Test zur korrekten Weiterleitung von Paketen. Zur Überprüfung der Erreichbarkeit eines Knotens wird der Ping eingesetzt, der auf ICMP basiert. Er nutzt dafür ICMP Echo-Request- und Echo-Reply-Nachrichten. Im Zuge der Entwicklung von IPv6 wurde auch ICMP weiterentwickelt. Dabei wurden neue Nachrichtentypen definiert, die die Funktionalität erweitern. Diese Entwicklung von ICMP für das Internetprotokoll in der Version 6 trägt den Namen ICMPv6.
Der nächste Abschnitt gibt einen Überblick über die wichtigsten Neuerungen.

Router Renumbering

Diese Weiterentwicklung ermöglicht es, durch die Definition von neuen Nachrichtentypen das Netzwerk neu zu nummerieren und Adressinformationen zwischen Routern und Endgeräten auf dem aktuellen Stand zu halten.

Neighbor Discovery

Neighbor Discovery (ND) besteht aus fünf ICMPv6 Nachrichten. Sie dienen der Auflösung von Adressen und konfigurieren die Kommunikation der Knoten am selben Link. Damit ersetzt ND das bekannte ARP und RARP, welches zur Auflösung von IP-Adressen in Layer 2 Adressen dient. Zusätzlich wird es genutzt, um Router zu finden, die Erreichbarkeit von Nachbarn zu überwachen sowie Änderungen der Link-Layer-Adressen zu erkennen.

Multicast Listener Discovery

Zur Verwaltung der Multicastgruppenzugehörigkeit sind drei ICMP Nachrichtentypen entwickelt worden. Damit ersetzt Multicast Listener Discovery (MLD) das in IPv4 verwendete IGMP.

Unterstützung für Mobile-IPv6

Zur Verwendung von Mobile-IPv6 wurden vier neue Nachrichtentypen in ICMPv6 definiert.

ICMPv6 muss in jeder Implementierung von IPv6 vollständig vorhanden sein und ist in RFC 4443 [A. Conta, 2006] definiert.

2.3.1 ICMPv6 Paketaufbau

ICMPv6 besteht im Wesentlichen aus zwei Gruppen von Nachrichten. Das sind zum einen die ICMP-Fehlermeldungen, deren Message-Type-Feld höchstes Bit auf null gesetzt ist und somit Fehlermeldungen mit Werten zwischen null und 127 enthalten und zum anderen ICMP-Informationsmitteilungen, deren Message-Type-Feld höchstes Bit den Wert eins hat und somit Werte für Informationsnachrichten von 128 bis 255 ermöglicht.
Einem Header von ICMPv6 geht ein IPv6-Header voraus. Dieser kann mehrere Extension-Header enthalten. Den Wert 58 im Next-Header-Feld enthält nur der Header unmittelbar vor dem ICMPv6-Header.
Der RFC 4443 [A. Conta, 2006] enthält die folgenden Nachrichtentypen für ICMPv6:

ICMPv6-Informationsnachrichten

- Echo-Request - Message-Type 128
- Echo-Reply - Message Type 129

ICMPv6-Fehlermeldungen

- Packet Too Big - Message-Type 2
- Time Exceeded - Message-Type 3
- Destination Unreachable - Message-Type 1
- Parameter Problem - Message-Type 4

Alle genannten Nachrichtentypen haben den selben Aufbau des Headers (Abbildung 2.9).

ICMPv6-Informationsnachrichten nutzen das *Code*-Feld nicht und setzen es aus diesem Grund auf null. Eine Ausnahme bildet die Router-Renumbering Nachricht.
Die Nachrichtentypen und die entsprechenden Nummern in ICMPv6 sind nicht gleich derer in ICMPv4, da ICMPv6 ein komplett neues, nicht zum alten ICMP kompatibles, Protokoll darstellt. [7]

[7] Eine aktuelle Liste der verwendeten ICMPv6-Nachrichtentypen und Werte des *Code*-Felds sind unter `www.iana.org/assignments/icmpv6-parameters` zu finden.

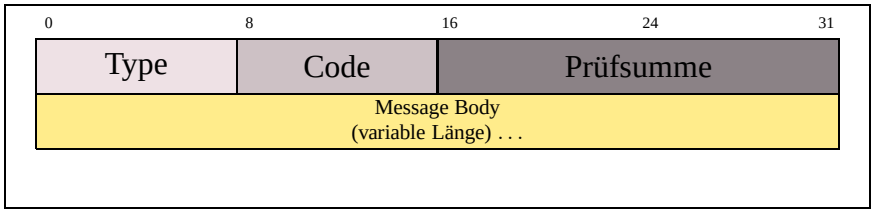


Abbildung 2.9: Gerüst des ICMPv6-Headers

2.3.2 Fehlermeldungen in ICMPv6

Je nach der Art der Fehlermeldung kann diese einen leicht veränderten Header besitzen. Damit die einzelnen Meldungen genau identifiziert werden können, folgen genaue Beschreibungen der einzelnen Nachrichten.

Destination Unreachable

Kann ein Paket nicht zugestellt werden, wird eine Destination Unreachable Nachricht erzeugt und an den Absender des Paketes geschickt. Das im Paket enthaltene *Code*-Feld gibt detaillierte Informationen zum aufgetretenen Fehler (siehe Tabelle 2.7). Am Ende der Nachricht befindet sich der Datenteil. Er enthält einen möglichst großen Teil des Originalpakets, aber nicht mehr als die minimale IPv6-MTU von 1280 Bytes.
Keine ICMPv6-Nachricht wird erzeugt, wenn ein überlasteter Router ein Paket verwirft und dieses die Zieladresse nicht erreicht. Das RFC 4443 [A. Conta, 2006] erlaubt aus Sicherheitsgründen auch Implementierungen, bei denen sich das Senden von Destination Unreachable Nachrichten pro Interface deaktivieren lässt.

Packet Too Big

Eine Packet Too Big Nachricht wird versendet, wenn ein Router ein Paket nicht weiterleiten kann, weil die Größe des Pakets die Größe der MTU des weiteren Links übersteigt. Dieser Nachrichtentyp wird außerdem für Path-MTU-Discovery eingesetzt. Das wichtigste Feld in der Packet Too Big Nachricht ist das *MTU*-Feld. Es gibt die MTU-Größe des nachfolgenden Links an den Absender des ursprüngliches Pakets weiter, damit dieser die neue MTU für die weitere Verbindung nutzen kann.
Laut RFC 4443 [A. Conta, 2006] sollen keine ICMPv6-Nachrichten erzeugt und gesendet werden, wenn die Empfänger eine IPv6-Multicast-Adresse, eine Link-Layer Multicast-Adresse oder eine Link-Layer Broadcast-Adresse sind. Diese Nachrichten bilden eine Ausnahme, da sonst der Path-MTU-Discovery Mechanismus für Multicasts nicht erfolg-

Code	Erläuterung
0	*No route to destination* - Code null wird eingesetzt, wenn der Router ein Paket nicht weiterleiten kann, weil er keine Route zum Zielnetzwerk besitzt. Dies kann beispielsweise der Fall sein, wenn keine Default-Route definiert ist.
1	*Communication with destination administratively prohibited* - Eine Firewall kann diesen Code einsetzen, wenn ein Paket aufgrund eines Paketfilters nicht weitergeleitet werden kann.
2	*Beyond scope of source address* - Dieser Code wird zurückgegeben, wenn der Scope der Quelladresse kleiner ist als der Scope der Zieladresse eines Paketes.
3	*Address unreachable* - Eine Meldung mit diesem Code deutet daraufhin, dass der Fehler keinem anderen Code zugeordnet werden konnte.
4	*Port unreachable* - Der Code vier wird verwendet, wenn das Transportprotokoll auf keinen Listener (port) trifft. Dies kann der Fall sein, wenn eine DNS-Anfrage an einen Host gestellt wird, auf dem kein DNS-Server läuft.
5	*Source address failed ingress/egress policy* - Wenn ein Paket aufgrund einer Ingress- oder Egress-Filter Regel nicht weitergeleitet werden kann, wird dieser Fehlercode erzeugt.
6	*Reject route to destination* - Dieser Code wird eingesetzt, wenn die Route zum Ziel eine sogenannte Reject-Route ist. Reject-Routen werden erzeugt, um Pakete an bestimmte Ziele immer mit einer Fehlermeldung zu beantworten.

Tabelle 2.7: Destination Unreachable Codes

reich ausgeführt werden könnte. Das Ende der Nachricht bildet ein möglichst großer Teil des Originalpakets.

Time Exceeded

Um sicherzustellen, dass Pakete nicht endlos durch ein Netzwerk kreisen, sind diese mit einem Hop Limit versehen. Jeder Router, der ein Paket weiterleitet, verringert dieses Hop Limit bei jedem Paket um eins. Wenn er ein Paket mit dem Hop Limit von eins erhält, so verringert er auch dieses um eins, sodass es den Wert null hat und verworfen wird. Sobald das Paket verworfen wurde, wird eine Time Exceeded Nachricht mit dem Code null an den Absender geschickt. Somit kann der Empfang einer solchen ICMPv6-Meldung entweder bedeuten, dass eine Routing-Schleife vorhanden ist oder der Absender ein zu kleines Hop Limit gewählt hat.
Für die Verwendung von Traceroute wird die ICMPv6-Nachricht Time Exceeded mehrfach erzeugt, um den Weg des Pakets nachvollziehen zu können. Es funktioniert wie folgt: Der Knoten, von dem Traceroute gestartet wird, schickt ein Paket mit dem Hop Limit eins an den Empfänger zu dem der Pfad ermittelt werden soll. Der erste Router, an dem das Paket ankommt, verringert nun das Hop Limit um eins, verwirft das Paket und sendet eine ICMPv6 Time Exceeded Nachricht mit seiner Absenderadresse an den ursprünglichen

Absender. Wenn er dieses Paket erhält, kennt er den ersten Router auf dem Weg zu seinem gewünschten Empfänger. Als nächstes erhöht er das Hop Limit sukzessive um den Wert eins, um so die weiteren Router auf dem Weg zum eigentlichen Empfänger herauszufinden. Wenn bei einem Traceroute vom Sender zum Empfänger redundante Wege möglich sind, bedeutet dies für das Traceroute, dass der angezeigte Pfad nicht der alleinige Pfad zum Empfänger ist. Es ist möglich, dass ein Paket unterschiedliche Wege zum Empfänger einschlägt.

Parameter Problem

Eine ICMPv6-Nachricht Parameter Problem wird erzeugt und versendet, wenn ein Knoten ein Feld im IPv6-Header oder in einem Extension-Header nicht erkennt und deshalb das Paket nicht verarbeiten kann. Außerdem wird dieser Typ von ICMPv6-Nachrichten auch für Fehlermeldungen verwendet, die in keine andere Kategorie einzuordnen sind. Am Ende der Parameter-Problem-Nachricht befindet sich das *Data*-Feld, welches wieder einen möglichst großen Teil des Originalpakets beinhaltet.
Wenn der Fehler in einem Teil des Originalpakets liegt, der nicht mit in den Datenteil des ICMPv6-Pakets passt, so zeigt der Pointer auf ein Byte außerhalb des ICMPv6-Pakets.

2.3.3 Informationsmeldungen in ICMPv6

In RFC 4443 [A. Conta, 2006] werden zwei Typen von Informationsnachrichten definiert. Diese sind zum einen der Echo-Request und zum anderen der Echo-Reply. Diese beiden Nachrichten werden für den Ping verwendet. Mit ihm kann die Erreichbarkeit eines Hosts herausgefunden werden. Dazu schickt der Sender einen Echo-Request. Wenn der Empfänger diese Nachricht erhält, also erreichbar ist, antwortet er mit einem Echo-Reply.

Zum Schutz vor Angriffen durch veränderte ICMPv6 Echo-Request/-Reply Nachrichten können diese mit einem IPv6 Authentication Header oder einem IP Encapsulating Security Payload Header geschützt werden. Ein IPv6-Knoten kann so eingerichtet werden, dass er nicht autorisierte ICMPv6 Pings verwirft.
Für zusätzliche Funktionen bei ICMP gibt es eine erweiterte ICMP-Struktur, die in RFC 4884 [R. Bonica, 2007] definiert ist. Dazu werden an das Ende der bisherigen ICMP-Nachricht ein Extension-Header sowie weitere Extension-Objekte angefügt. Eine Erweiterung ist nur möglich, wenn der ICMP-Header ein Längenfeld besitzt. Dies ist bei den folgenden Nachrichten möglich:

- ICMPv4 Time Exceeded

- ICMPv6 Time Exceeded
- ICMPv4 Destination Unreachable
- ICMPv6 Destination Unreachable
- ICMPv4 Parameter Problem

2.3.4 Verarbeitung von ICMPv6-Nachrichten

RFC 4443 [A. Conta, 2006] enthält zu den bereits genannten Spezifikationen noch weitere Regelungen zur Verarbeitung von ICMPv6-Paketen. Im Folgenden sind die wichtigsten näher erläutert:

1. Sobald ein Knoten eine ICMPv6-Fehlermeldung erhält und der Typ dieser Meldung für ihn unbekannt ist, muss er diese an die höhere Schicht weiterleiten.
2. Wenn ein Knoten eine ICMPv6-Informationsnachricht mit einem Typ erhält, der ihm unbekannt ist, so wird diese Nachricht verworfen.
3. Jede ICMPv6-Fehlermeldung enthält einen möglichst großen Teil des ursprünglichen Paketes, dass die Fehlermeldung auslöste. Die minimale IPv6-MTU (1280 Bytes) darf dabei nicht überschritten werden.
4. Eine ICMPv6-Nachricht, die in eine höhere Schicht weitergegeben werden muss, wird auf den Typ des Protokolls der höheren Schicht hin untersucht, um eine korrekte Weitergabe zu gewährleisten. Dieser Protokolltyp wird aus dem Datenteil der ICMPv6-Nachricht ausgelesen. Wenn dies nicht möglich ist, wird das Paket verworfen.

Weiterhin gibt es Fälle, in denen keine ICMPv6-Fehlermeldungen ausgelöst werden dürfen:

1. Infolge einer ICMPv6-Fehlermeldung oder ICMPv6-Redirect-Nachricht.
2. Als Folge eines Paketes, das an eine IPv6-Multicast-Adresse gesendet wurde. Für diesen Fall gibt es die folgenden zwei Ausnahmen:
   - Packet Too Big Nachrichten, die für Path-MTU-Discovery genutzt werden
   - Parameter Problem Nachrichten mit dem Code zwei und auf den Wert 0b10 gesetzten beiden höchsten Bits des *Option Type*-Felds. Dieser Code meldet eine unbekannte IPv6-Option.

3. Als Folge eines Paketes, das als Link-Layer-Multicast oder Link-Layer-Broadcast verschickt wurde. Für diese Fälle gelten die Ausnahmen wie oben beschrieben.

4. Infolge eines Paketes, dessen Quelladresse nicht exakt ein Interface kennzeichnet. Ein Beispiel dafür wäre eine IPv6-Multicast-Adresse oder die IPv6-Unspecified Adresse.

Zur Verarbeitung von ICMPv6-Nachrichten ist es notwendig, dass jeder Knoten eine Funktion bereitstellen muss, mit der er die Anzahl an ICMPv6-Nachrichten, die er selbst versendet, begrenzen kann. Somit besteht ein einfacher Schutz vor Denial-of-Service Angriffen.

2.4 ICMPv6 Funktionen

Mit der Entwicklung von ICMPv6 und der Erweiterung des Protokolls um zusätzliche Nachrichtentypen, entstanden weitere Funktionen, die in diesem Abschnitt näher erläutert werden sollen.

2.4.1 Neighbor Discovery

Das in RFC 4861 [T. Narten, 2007b] definierte Neighbor Discovery (ND) kombiniert ARP, ICMP Router-Discovery und ICMP Redirect, die aus IPv4 bekannt sind. Hinzugekommen sind Mechanismen zur Überprüfung der Verfügbarkeit eines Nachbarn im Netzwerk (Neighbor Unreachability Detection) sowie zur Vermeidung doppelter IP-Adressen im Netzwerk (Duplicate IP Address Detection).
Neighbor Discovery wird von IPv6-Knoten für die folgenden Aufgaben verwendet:

- Überwachung der Erreichbarkeit der Nachbarn
- Auffinden von Routern am eigenen Link
- Erkennung doppelter IP-Adressen
- Ermittlung von Netzwerkpräfixen, Routen und anderen Informationen zur Konfiguration des Knotens
- Autokonfiguration von IPv6-Adressen
- Ermittlung der MAC-Adressen von Knoten am eigenen Link

- Prüfung von Änderungen der Adressen auf MAC-Ebene

Im Vergleich zu den dafür notwendigen Funktionsaufrufen und Abläufen in IPv4 haben sich bei IPv6 viele Verbesserungen ergeben.
Das zur Konfiguration der Informationen über die dem Knoten zur Verfügung stehenden Router notwendige Router Discovery ist bei IPv6 nun fester Bestandteil des Standard Protokollpakets. Die dazu verwendeten Router Advertisement Pakete enthalten die MAC-Adresse des Routers, so dass keine zusätzliche Anfrage notwendig ist, um diese aus der IP-Adresse aufzulösen. Außerdem enthalten diese Pakete das für diesen Link gültige Präfix. Damit entfällt die Konfiguration von Subnetzmasken.
Neighbor Discovery ermöglicht zudem eine einfache Neuadressierung des Netzwerks. Dabei können Präfixe und Adressen parallel eingeführt und alte kontinuierlich deaktiviert werden. Mithilfe von Router Advertisements kann eine zustandslose Autokonfiguration (Stateless Address Configuration) erfolgen oder der Host darüber informiert werden, dass er die Konfiguration per DHCP durchzuführen hat. Ebenso über ein Router Advertisement bekommen die jeweiligen Links Präfixe zugewiesen. Im Normalfall bekommt ein Host alle Präfixe für den Link vom Router mitgeteilt. Er kann aber auch so konfiguriert werden, dass er kein oder nur ein Präfix bekannt gibt. Somit kann erreicht werden, dass ein Host ein Paket, dessen Ziel er nicht am selben Link vermutet, erst einmal zum Router gesendet wird. Sobald es eine bessere Route zum gewünschten Ziel gibt, kann diese per ICMPv6 Redirect bekannt gegeben werden. Die Neighbor Unreachability Detection ist ebenso Teil des Standard Protokollpakets. Sie erkennt, ob ein Interface seine MAC-Adresse ändert oder ein Router ausfällt. Ist ein Alternativrouter vorhanden, kann in einem solchen Fall zum ihm gewechselt werden. Damit wird auch das Problem nicht mehr aktueller ARP-Caches gelöst. Sobald ND erkennt, dass eine Kommunikation einen Fehler verursacht, wird den Knoten mitgeteilt, keine weiteren Pakete an den nicht mehr erreichbaren Host zu senden. Neighbor Discovery ist weitestgehend unempfindlich gegenüber entfernten Hosts, die versuchen, in das Netzwerk einzudringen. Es beantwortet nur Anfragen am selben Link, da diese ein Hop Limit von 255 besitzen. Zur weiteren Verbesserung der Sicherheit können ND IP-Authentication und weitere Security Mechanismen verwenden.

Neighbor Advertisement und Neighbor Solicitation

Diese beiden Nachrichten werden zur MAC-Adressauflösung und Neighbor Unreachability Detection genutzt. Ist der Empfänger einer Neighbor Solicitation Nachricht eine Multicast-Adresse, so wird diese Nachricht zur Auflösung einer MAC-Adresse verwendet. Befindet sich jedoch eine Unicast-Adresse im Empfängerfeld, dient die Nachricht einer Erreichbarkeitsabfrage. Außerdem wird dieser Nachrichtentyp für die Duplicate Ad-

dress Detection (DAD) verwendet. Der Absender im IPv6-Header einer solchen Nachricht ist entweder eine Unicast-Adresse des absendenden Interfaces oder die unspezifizierte Adresse (wird bei DAD verwendet).
Ein Neighbor Advertisement wird entweder als Antwort auf eine Neighbor Solicitation oder zur Verbreitung neuer Informationen am selben Link versendet. Für den ersten Fall wird als Ziel-Adresse des Advertisements die Absenderadresse der Solicitation verwendet. Im zweiten Fall bzw. wenn die Nachricht eine Antwort auf eine DAD-Nachricht darstellt, wird die All-Nodes Multicast-Adresse benutzt.
Das Feld mit der Zieladresse im ICMP-Paket enthält bei einem angeforderten Advertisement die Adresse aus der Solicitation. Im Fall eines unaufgeforderten Versendens des Advertisements wird in dieses Feld die IPv6-Adresse des Interfaces, dessen MAC-Adresse geändert wurde, eingetragen.

Router Advertisement und Router Solicitation

Router Advertisements können von Hosts durch das Senden einer Router Solicitation Nachricht angefordert werden. Sobald ein Router eine solche Nachricht erhält, ist er dazu angewiesen, sofort ein Router Advertisement zu schicken und nicht, wie gewöhnlich, das Intervall abzuwarten, in dem er regelmäßig Advertisements verschickt.
Bei einer Autokonfiguration schickt der zu konfigurierende Host eine Solicitation an die All-Routers Multicast-Adresse (ff02::2). Da er selbst noch keine IPv6-Adresse besitzt, nutzt er als Absenderadresse die unspezifizierte Adresse (::). Router Advertisements werden vom Router entweder an die All-Nodes Multicast-Adresse (ff02::1) verschickt, wenn es sich um die periodisch verschickte Nachricht handelt oder als eine Antwort auf eine Solicitation, da der zu konfigurierende Host noch keine eindeutige Adresse besitzt. Ansonsten werden Advertisements nur an die Unicast-Adresse geschickt, von der die Solicitation empfangen wurde.

ICMPv6 Redirect

ICMPv6 Redirect Nachrichten werden durch Router versendet, um einen Host darüber zu informieren, dass ein besserer First-Hop Router zur Verfügung steht oder das Ziel sich am selben Link befindet. Die Absenderadresse einer solchen Nachricht ist die Link-lokale-Adresse des Interfaces des Versenders der Nachricht. Der Empfänger ist der Absender des Paketes, welches den ICMPv6 Redirect ausgelöst hat. Für diesen Nachrichtentyp gibt es zwei Optionen. Bei der ersten Variante enthält das Optionsfeld die Link-Layer-Adresse des Ziels. Diese muss, sofern sie bekannt ist, mit angefügt werden. Bei der zweiten Va-

riante wird der übrige Platz mit dem Originalpaket aufgefüllt. Diese Nachricht darf dann nicht größer als die minimale IPv6-MTU sein.

Neighbor Discovery Optionen

Da alle Neighbor Discovery Nachrichten ein Optionsfeld mit variabler Größe enthalten, gibt es für dieses Feld eine Vielzahl an Möglichkeiten. Eingetragen werden diese Optionen nach dem TLV-Format (Type, Length, Value). Die zur Verfügung stehenden Optionstypen sind in RFC 4861 [T. Narten, 2007b] näher spezifiziert. Darin sind auch alle Felder der zugehörigen Optionen zu finden [8]. Beispielsweise kennzeichnet der Typ 3 die Präfixoptionen. Dabei wird der Typ des Präfixes definiert (lokales oder remote Präfix). Außerdem können weitere Optionen zur Autokonfiguration gesetzt oder auch die komplette IPv6-Adresse des Routers ins Präfixfeld eingetragen werden. Zudem kann die Gültigkeitsdauer der Präfixinformationen gesetzt werden.

Inverse Neighbor Discovery

Ursprünglich wurde Inverse Neighbor Discovery (IND) für Frame Relay Netzwerke entwickelt. Es kann aber auch für andere Netzwerke eingesetzt werden. Spezifiziert ist es im RFC 3122 [Conta, 2001]. IND besteht aus zwei Nachrichten mit denen ein Knoten die IPv6-Adresse, die mit einer bestimmten MAC-Adresse verbunden ist, ankündigen (Inverse Neighbor Discovery Advertisement) oder ermitteln (Inverse Neighbor Discovery Solicitation) kann. Im Vergleich zu IPv4 entspricht IND dem Reverse Address Resolution Protocol (RARP).
Sobald ein Knoten die IPv6-Adresse eines benachbarten Interfaces, dessen MAC-Adresse er kennt, auflösen möchte, schickt er eine Solicitation Nachricht an die All-Nodes Multicast Adresse (ff02::1). Auf der Link-Layer-Ebene wird diese Nachricht direkt an das Interface gesendet. Sie muss die MAC-Adresse des Senders und des Empfängers enthalten. Zusätzlich kann sie noch die MTU für den Link sowie eine Liste von IPv6-Adressen des Interfaces, welches in der Source-Link-Layer-Address-Option angezeigt ist, enthalten.
Als Antwort auf eine solche Solicitation Nachricht sendet der Empfänger das Inverse Neighbor Advertisement. Diese Nachricht muss die MAC-Adressen des Empfängers und des Senders enthalten. Außerdem kann wiederum eine Liste von IPv6-Adressen des Interfaces, welches in der Target-Link-Layer-Address-Option enthalten ist und die MTU des

[8] Die aktuelle Liste der Optionen ist unter `www.iana.org/assignments/icmpv6-parameters` zu finden.

Links mitgeschickt werden.
Die oben genannten IPv6-Adresslisten müssen in der jeweiligen IND-Nachricht untergebracht werden können. Wenn die Liste zu umfangreich ist, bleibt diese bei einer IND Solicitation unvollständig. Bei einem IND Advertisement dagegen werden so viele Nachrichten gesendet, bis die komplette Liste übertragen wurde.

Secure Neighbor Discovery

Sicherheitsmechanismen sind für Neighbor Discovery sehr wichtig, da es für eine Vielzahl von Angriffen missbraucht werden kann. Denkbar wäre ein Angriff, bei dem sich ein Gerät als Default-Router ausgibt und Router-Advertisements verschickt, sodass alle Default-Router und Präfixe ungültig werden.
Ein einfacher Schutzmechanismus ist das vom Protokoll vorgegebene Hop Limit von 255. Damit ist gewährleistet, dass keine Neighbor Discovery Nachrichten von außerhalb des Links angenommen werden.
Eine weitere Möglichkeit, ND Nachrichten zu schützen, besteht in der Verwendung von IPsec. Dazu ist entweder ein Key Management Protokoll oder Security Associations notwendig. Beide Varianten sind jedoch sehr aufwendig und unpraktikabel.
Aus diesem Grund wurden die Angriffspunkte und Anforderungen an ein sicheres Neighbor Discovery untersucht und in RFC 3756 [P. Nikander, 2004] zusammengefasst. Für Links ohne physikalische Sicherheit (z.B. WLAN) definiert RFC 3971 [J. Arkko, 2005] das SEcure Neighbor Discovery (SEND). Dieses besteht aus den folgenden Elementen:

- Autorisieren der Router durch Zertifizierungswege von vertrauenswürdigen Drittparteien.
- Ein Router kann erst als Default-Router von einem Endsystem eingesetzt werden, wenn dieser Teilnehmer ein vertrauenswürdiges Element (trust anchor) besitzt, von dem aus ein Zertifizierungspfad zum Router vorhanden ist.
- Der Zertifizierungspfad zu einem trust anchor wird über Certification-Path Solicitation und Certification-Path Advertisement Nachrichten hergestellt.
- Kryptografisch generierte Adressen werden verwendet, um sicherzustellen, dass der Sender einer ND-Nachricht auch der Besitzer der von ihm beanspruchten Adresse ist.
- Die Adressen können nur erhalten werden, wenn der Knoten ein privates/öffentliches Schlüsselpaar erzeugt hat.

- Die Übermittlung des öffentlichen Schlüssels und der dazugehörigen Parameter erfolgt mit einer eigens dafür entwickelten *CGA*-Option in der ND-Nachricht.
- Neighbor und Router Discovery Nachrichten werden mit der ND-Option *RSA-Signature* geschützt.
  - Die beiden ND-Option *Timestamp* und *Nonce* verhindern einen Angriff durch das Wiedereinspielen von Nachrichten (reply-Attack).
- SEND verwendet kryptografisch generierte Adressen und kann nicht für Knoten mit statischen oder durch stateless Autokonfiguration vergebenen Adressen verwendet werden.

Alle aufgeführten Optionen und zusätzliche Nachrichtentypen sind in RFC 3971 [J. Arkko, 2005] vereinbart.

Neighbor Unreachability Detection

Ein Knoten in einem Netzwerk gilt als erreichbar, wenn vor kurzer Zeit eine Bestätigung erfolgte, dass dieser Pakete, die an ihn gerichtet waren, empfangen und verarbeitet hat. Als Nachbarn werden dabei Knoten bezeichnet, die sich am selben Link befinden. Als Bestätigung gilt:

- die Beantwortung einer Neighbor Solicitation mit einem Neighbor Advertisement
- eine aktive TCP Verbindung
- der Erhalt eines TCP Acknowledgement

Die Überwachung der Bestätigungen erfolgt fortlaufend anhand von Tabellen, die durch die IPv6-Knoten geführt werden. Besonders wichtig sind dabei die Neighbor-Cache und Destination-Cache Tabellen, die im Nachfolgenden erläutert werden.

Neighbor-Cache Tabelle

Die Neighbor-Cache Tabelle enthält eine Liste mit Nachbarn, mit denen kürzlich kommuniziert wurde. Dabei enthält jeder Eintrag die Unicast- und Link-Layer-Adresse des Nachbarn sowie ein Flag, das anzeigt, ob es sich um einen Router oder einen Host handelt. Diese Tabelle ist mit dem ARP-Cache von IPv4 vergleichbar. Zusätzlich enthält sie Informationen über Pakete, die für einen Nachbarn vorgesehen sind, sich aber noch in der Warteschlange befinden, erreichbare Nachbarn und den nächsten Zeitpunkt für eine Neighbor Unreachability Detection.

Destination-Cache Tabelle

Diese Tabelle enthält eine Liste der Empfänger, mit denen kürzlich kommuniziert wurde. Sie enthält lokale wie auch entfernte (remote) Empfänger. Die Einträge entfernter Knoten enthalten dabei die MAC-Adresse des Next-Hop-Routers. Zudem enthält jeder Eintrag auch Informationen zur jeweiligen MTU und dem Roundtrip Timer. Die Aktualisierung erfolgt durch ICMPv6 Redirect Nachrichten.

Neighbor Advertisements enthalten ein sogenanntes Override Flag. Es bestimmt, ob eine Information aus einem Advertisement, welches der Knoten erhält, in den Neighbor-Cache geschrieben wird (durch Überschreiben des alten Eintrags) oder nicht.
Die einzelnen Einträge im Neighbor-Cache besitzen fünf unterschiedliche Zustände. Diese sind in Tabelle 2.8 zusammengefasst und können im RFC 4861 [T. Narten, 2007b] nachgelesen werden.

Zustand	Erläuterung
Incomplete	Mit der Auflösung der Adresse wurde begonnen. Die Antwort steht aus. Eine Link-Layer-Adresse ist somit noch nicht bekannt.
Reachable	Die Erreichbarkeit des Nachbarn ist durch den Erhalt eines Neighbor Advertisements innerhalb einer erwarteten Zeit bestätigt worden.
Stale	Die Erreichbarkeit des Nachbarn ist nicht bekannt. Die erwartete Zeit für den Erhalt eines Neighbor Advertisements ist überschritten.
Delay	Die erwartete Zeit für eine Rückmeldung über die Erreichbarkeit eines Nachbarn ist verstrichen und ein Paket wurde innerhalb der Delay-First-Probe Zeit gesendet. Wenn in dieser Zeit keine Bestätigung über den Erhalt des Pakets eintrifft, wird der Zustand des Nachbarn auf *Probe* gesetzt.
Probe	Die Bestätigung der Erreichbarkeit des Nachbarn soll festgestellt werden, indem Neighbor Solicitations versendet werden. Wird nach einer bestimmten Anzahl von Versuchen keine Bestätigung erhalten, so wird der Nachbar aus dem Cache entfernt.

Tabelle 2.8: Zustände der Neighbor-Cache-Einträge

2.4.2 Autokonfiguration

Die Autokonfiguration für IPv6 wurde entworfen, um die Adressierung einer Vielzahl von zukünftigen IPv6-Geräten zu vereinfachen. Sie kann entweder *stateful*, also mittels eines DHCP-Servers, oder *stateless*, ohne DHCP-Server, realisiert werden. Der Einsatz der Autokonfiguration ermöglicht den Hosts automatisch und selbstständig eine eigene IPv6-Adresse zu generieren. Dabei werden lokale Informationen (MAC-Adresse oder zufällig generierte ID) mit Präfixinformationen von Routern kombiniert. Theoretisch ist auch die Kombination der beiden Verfahren möglich. In einem solchen Fall wird die IP-Adresse

per *stateless* Autokonfiguration und zusätzlicher Parameter per DHCP bezogen.
Alle per Autokonfiguration zugeteilten Adressen sind mit einer Gültigkeitsdauer (Lifetime) versehen. Zur Sicherstellung der Eindeutigkeit der zugeteilten Adresse findet eine Prüfung per DAD (Duplicate Adress Detection) statt.
Während des Prozesses der Autokonfiguration kann eine IPv6-Adresse mehrere Zustände annehmen:

- **Vorläufig (tentative)**
  Befindet sich eine Adresse eines Knoten in diesem Zustand, so kann er nicht über diese kommunizieren. Er empfängt lediglich Neighbor-Discovery Nachrichten, um die Eindeutigkeit der IP-Adresse testen zu können.

- **Bevorzugt (preferred)**
  Eine *preferred*-Adresse kann ohne Einschränkungen genutzt werden, da ihre Eindeutigkeit bestätigt wurde. Sie ist mit einer Gültigkeit versehen, die über ein Router Advertisement mitgeteilt wird.

- **Abgelehnt (deprecated)**
  Eine solche Adresse hat nur noch eine begrenzte Gültigkeit. Dabei können vorhandene Kommunikationen bestehen bleiben, aber keine neuen aufgebaut werden.

- **Gültig (valid)**
  Dieser Begriff bezeichnet die Zustände *preferred* und *deprecated*.

- **Ungültig (invalid)**
  Eine ungültige Adresse ist keinem Interface zugewiesen.

Die Definition der Zustände ist in RFC 4862 [S. Thomson, 2007] beschrieben. Eine *stateless*-Autokonfiguration läuft in folgenden Schritten ab:

1. Generierung einer Link-lokalen-Adresse mit dem Präfix fe80. Diese Adresse befindet sich nun im Zustand *tentative*.

2. Beitritt des Interfaces zu den Gruppen All-Nodes Multicast (ff02::1) und Solicited-Node Multicast-Adresse für die Adresse aus dem ersten Schritt.

3. Versenden einer Neighbor Solicitation mit der vorläufigen Adresse im Zieladressfeld. Als Absenderadresse wird die unspezifizierte Adresse verwendet. Damit ist der DAD-Test vollzogen. Falls diese Adresse schon vergeben sein sollte, sendet der Knoten mit der gleichen Adresse ein Neighbor-Advertisement und der Autokonfigurationsprozess wird nicht weiter fortgesetzt. Andernfalls ist die Adresse eindeutig

und kann genutzt werden. Nun wird sie dem Interface zugewiesen und erhält damit den Zustand *preferred*. Damit ist eine IP-Verbindung am Link hergestellt.

4. Zur Erkennung vorhandener Router am Link sendet der Host eine Router Solicitation an die All-Router Multicast-Adresse ff02::2.

5. Von jedem Router am Link folgt nun ein Router Advertisement. Für jedes Präfix, mit gesetztem A-Flag (Autonomous), aus dem Router Advertisement wird aus dem Präfix und dem Interface Identifier eine IP-Adresse generiert. Diese Adressen werden danach in eine Liste der zugewiesenen Adressen des Interfaces eingetragen.

2.4.3 Änderung des Netzwerkpräfixes

Zur Änderung des Netzwerkpräfixes stellt ICMPv6 einen Mechanismus bereit, der dies wesentlich vereinfacht. Spezifiziert ist diese Vorgehensweise im RFC 4192 [F. Baker, 2005]. Demnach gibt es die Möglichkeit, einem Interface mehrere Präfixe zuzuweisen. Es kann ein neues Präfix eingeführt werden, während das alte noch zur Kommunikation genutzt wird. Sobald die Kommunikation vollständig über das neue Präfix vonstattengehen kann, wird das alte Präfix abgeschaltet. Die Schritte der Vorgehensweise sind im Folgenden dargestellt:

1. Das neue Link-Präfix wird an jedem Link im Netzwerk verteilt.

2. Konfiguration des neuen Präfixes parallel zum alten bei allen relevanten Diensten und Geräten.

3. Während die alten Adressen ihre Gültigkeit behalten, erhalten die Host-Interfaces ihre neuen Adressen. Dies geschieht entweder über *stateless*-Autokonfiguration oder DHCPv6.

4. Im DNS werden neue AAAA-Einträge und PTR-Records definiert. Wenn das neue Präfix aktiviert und getestet ist, können die alten DNS-Records (mit dem alten Präfix) gelöscht werden.

5. Nach abgeschlossener Konfiguration und Überprüfung des neuen Präfixes, kann das alte Präfix in allen Bereichen abgeschaltet werden.

6. Bei allen Geräten und Applikationen, die ihre Adressinformationen nicht per DNS oder DHCP erhalten oder IP-Informationen lokal zwischenspeichern, sollte zur Gewährleistung eines reibungslosen Ablaufes mit besonderer Vorsicht vorgegangen werden.

2.4.4 Path MTU

Anders als bei IPv4 fragmentieren Router im IPv6-Netz keine Pakete. Wenn eine Fragmentierung notwendig ist, muss diese durch den Absender des Pakets realisiert werden. Für diesen Vorgang versucht Path-MTU-Discovery sicherzustellen, dass die größtmögliche MTU genutzt werden kann. Der Bergriff Path-MTU bezeichnet hierbei die größte MTU eines Links zwischen dem Absender und dem Empfänger. Der Vorgang des Ermittelns der größten MTU eines Weges ist in RFC 1981 [J. McCann, 1996] spezifiziert und wird im Folgenden erläutert.

1. Unter Verwendung der MTU am lokalen Link schickt der Absender das erste Paket zum Empfänger.
2. Wenn das Paket zu groß für einen Link ist und ein Router es dementsprechend nicht weiterleiten kann, sendet dieser Router eine ICMPv6 *Packet Too Big* Nachricht zurück und verwirft das Paket. Die zurückgesendete Nachricht enthält als Option die MTU des Next-Hop-Links des Routers.
3. Anschließend benutzt der Host die aus dem ICMPv6 Paket entnommene MTU für diesen Empfänger.

In einigen Fällen ist es möglich, dass der oben genannte Ablauf mehrmals durchgeführt werden muss, bis die kleinste MTU auf dem Weg zum Empfänger gefunden ist. Dabei wird jedoch die minimale IPv6-MTU nie unterschritten.
Da der Weg zu einem Empfänger nicht immer der gleiche sein muss bzw. sich auch ändern kann, ist auch die Path-MTU veränderlich. Sobald die MTU kleiner wird, kann dies anhand einer ICMPv6 *Packet Too Big*-Nachricht erkannt werden. Zur Erhöhung der Leistungsfähigkeit des Netzwerks ist es auch wichtig, wenn eine höhere MTU erkannt wird. Um dies zu realisieren, versendet der Host gelegentlich ein Paket mit einer höheren MTU. Dies geschieht so lange, bis er wieder eine *Packet Too Big*-Nachricht erhält.
Die Path-MTU-Discovery Funktion unterstützt auch den Versand von Paketen an Multicast Empfänger. Auch hierbei erhält der Absender *Packet Too Big*-Nachrichten. Aus all diesen Nachrichten wählt er dann die kleinste MTU aller Empfänger für diese Multicast-Adresse aus.

2.4.5 Multicast Gruppenmanagement

Eine Gruppe von Knoten wird mittels einer Multicast-Gruppenadresse gekennzeichnet. Diese Adressen können am hochwertigen Byte 0xff erkannt werden. Zur Gewährleistung

eines effektiven Routings an eine Multicast-Gruppe ist dafür ein spezielles Protokoll notwendig. Es sorgt dafür, dass Pakete an bestimmte Multicast-Gruppen nur über die Interfaces weitergeleitet werden, in dessen Link sich Mitglieder dieser Gruppe befinden.

Multicast Listener Discovery Version 1

Während das Multicast Gruppenmanagement in IPv4 noch über IGMP in der Version 2 nach RFC 2236 [Fenner, 1997] ablief, ist es bei IPv6 schon in ICMPv6 enthalten. Genau spezifiziert ist das Gruppenmanagement in RFC 2710 [S. Deering, 1999]. Das damit zusammenhängende Protokoll trägt die Bezeichnung Multicast Listener Discovery (MLD) Version 1.
Das Protokoll läuft asymetrisch ab. Dabei verhalten sich Knoten, die Nachrichten für bestimmte Multicast-Gruppen erhalten möchten, sogenannte Listener, anders als Router. Für die Adressen, für die ein Router Listener ist, führt er die beiden Teile des Protokolls aus. Die Member Reports genannten Nachrichten verschicken die Listener für ihre Multicast-Adressen, um sich bei den Routern am Link für diese Adresse zu registrieren. Der jeweilige Router trägt diese Adresse dann, falls sie nicht schon vorhanden ist, in eine Liste für diesen Link ein und leitet so lange Nachrichten für diese Multicast-Adresse über den Link weiter, bis die Adresse wieder ausgetragen wird. Eine Austragung kann mit einer sogenannten Done-Nachricht durchgeführt werden, die von einem Knoten an den Router gesendet wird.
Alle MLD-Nachrichten werden mit einem Hop Limit von eins versendet, so dass sie im lokalen Netz bleiben. Außerdem sind sie mit einem Hop-by-Hop Options-Header, der einen Router Alert enthält, versehen. Damit ist sichergestellt, dass auch jeder lokale Router das Paket verarbeitet, auch wenn er kein Mitglied der Multicast-Gruppe ist.

Multicast Listener Discovery Version 2

Die zweite Version des Multicast Listener Discovery basiert auf der Version 3 des IGMP (RFC 3376 [B. Cain, 2002]) und ist im RFC 3810 [for IPv6, 2004] spezifiziert. Als Neuerung im Vergleich zur Version 1 ist der sogenannte Source-Specific-Multicast (SSM) hinzugekommen. Dadurch ist es möglich, sich als Empfänger einer Multicast-Gruppe einzutragen und zusätzlich zu entscheiden, von welcher Quelle Pakete empfangen werden sollen. Um die Rückwärtskompatibilität zu gewährleisten, unterstützt die neue Version auch die Nachrichtentypen der vorherigen Version.

Multicast Router Discovery

Mulicast Router Discovery beschreibt einen Ablauf zum Auffinden von Multicast Routern. RFC 4286 [B. Haberman, 2005] enthält die Spezifikation, in der drei neue Nachrichtentypen beschrieben werden.

Multicast Router Solicitation
: Um Multicast Router Advertisements zu erhalten, kann eine Nachricht diesen Typs von der Link-lokalen-Adresse an die All-Routers Multicast-Adresse ff02::2 gesendet werden.

Multicast Router Advertisement
: Damit Router ankündigen können, dass sie IP-Multicast-Forwarding unterstützen, können sie diese Nachricht versenden. Diese wird von der Link-lokalen-Adresse an die All-Snoopers Multicast Adresse ff02::6a versendet.

Multicast Router Termination
: Sobald ein Router die Multicast-Routing Funktion an einem bestimmten Interface beendet, sendet er eine solche Nachricht. Diese wird von der Link-lokalen-Adresse an die All-Snoopers Multicast Adresse ff02::6a versendet.

Damit alle Nachrichten von allen Routern verarbeitet werden, enthalten sie die Router Alert Option. Zudem werden sie mit einem Hop Limit von eins versendet.

2.5 Routing im IPv6-Netz

Die Weiterleitung eines IP-Paketes ins Internet bzw. schon über den eigenen lokalen Link hinaus erfordert die Verwendung eines Routers. Ein Router enthält eine Tabelle mit den ihm zur Verfügung stehenden Zielen an seinen Interfaces. Sobald ein Paket am Router ankommt, schaut dieser auf die Zieladresse im Paket und sucht in seiner Routing-Tabelle nach einem übereinstimmenden Eintrag. Ein solcher Eintrag kann entweder manuell eingetragen oder mittels eines Routing-Protokolls realisiert werden. Dabei werden Routing-Informationen automatisch zwischen den Routern ausgetauscht. Eine Menge von Netzwerken, die von einer einzigen sogenannten Autorität verwaltet werden, bezeichnet man als autonomes System (AS). Routing-Protokolle, die Routing-Informationen innerhalb eines solchen AS austauschen, werden Interior-Gateway-Protokolle (IGP) genannt. Zu dieser Gruppe gehören unter anderem RIP, OSPF sowie IS-IS und EIGRP.

Routing-Protokolle, die Informationen zwischen unterschiedlichen AS austauschen, werden Exterior-Gateway-Protokolle genannt. Aktuell wird dafür nur BGP-4 eingesetzt. Für alle gerade genannten Protokolle gibt es Erweiterungen oder Neuentwicklungen für IPv6, die im Folgenden näher erläutert werden.

2.5.1 OSPFv3

OSPF Version 3 bezeichnet ein Link-State-Routing-Protokoll, welches nur für IPv6 verwendet wird. Es basiert auf der Version 2 von OSPF, welches für IPv4 entwickelt wurde. Anpassungen sind vor allem durch den größeren Adressbereich notwendig geworden. Die Definition von OSPFv3 ist im RFC 5340 [R. Coltun, 2008] nachzulesen.
OSPF für IPv4 (OSPFv2) ist im RFC 2328 [Moy, 1998] definiert und in den darauffolgenden Jahren durch eine Vielzahl von RFCs um diverse Funktionen erweitert worden. Dazu zählt auch die Unterstützung des Internet Protokolls in der Version 6. Diese ist später zu OSPFv3 ausgegliedert worden.
Zum Ausgleich der Einschränkungen des Routing-Protokolls RIP wurde OSPF als ein weiteres Interior-Gateway-Protocol entwickelt, welches zudem umfangreichere Routing-Tabellen und dementsprechend auch größere Netzwerke unterstützt.

Die wichtigsten Merkmale von OSPFv3

Zu einem großen Teil wurden Überlegungen und Erweiterungen von OSPFv2 in OSPFv3 übernommen. Im Folgenden sind die wichtigsten Änderungen zusammengefasst. Zum Verständnis der Veränderungen in OSPFv3 sind gute Kenntnisse des Routingprotokolls OSPFv2 notwendig. Des Weiteren werden die Begriffe in der genaueren Erläuterung von OSPFv3 ausführlich erklärt.

Flooding-Scopes

Es wurden sogenannte Flooding-Scopes eingeführt. Dabei erhält jeder LSA-Typ einen Code für die Bestimmung des Verteilungsbereichs. Diese Bereiche können Link-lokal, Area und AS sein.

Entfernung der Adresssemantik

IPv6-Adressen sind im Datenteil enthalten und nicht im OSPFv3-Header. Router- und Network-LSAs enthalten keine IPv6-Adressen, da ein Link durch eine 32 Bit große Nummer bezeichnet wird und nicht, wie in OSPF für IPv4, mit einer IP-Adresse. Da die OSPF-Router-, Area- und Link-State-ID nur 32 Bits groß sind,

können auch diese keine IPv6-Adressen aufnehmen. Die Designated- und Backup-Designated-Router werden hier über die Router-ID und nicht über die IP-Adresse des Interfaces beschrieben.

Mehrere Instanzen pro Link

An einem einzelnen Link können mehrere Protokollinstanzen von OSPFv3 vorhanden sein. Hierbei können eigenständige AS mit eigenem OSPF über einen gemeinsamen Link Informationen austauschen. Zudem kann ein einzelner Link auch zu mehreren OSPF-Areas gehören.

Link-lokale-Adressen

OSPFv3 setzt voraus, dass jedem Interface eine Link-lokale-Adresse zugeordnet ist, da diese als Absenderadresse für OSPFv3-Pakete verwendet wird. Diese Link-lokalen-Adressen lernen die Router von allen Nachbarn um sie als Next-Hop Adresse zu nutzen. Dies gilt nicht für virtuelle Links. Diese müssen eine Unique-Local- oder Global-IPv6-Adresse besitzen.

Authentication

Durch die Benutzung von IPv6 kann bei OSPFv3 die Authentifizierung über IPv6 erfolgen. Es sind keine zusätzlichen Maßnahmen seitens OSPFv3 notwendig. Die Prüfsumme bildet den Integritätscheck, der über das ganze OSPF Paket berechnet wird.

Verarbeitung pro Link

OSPFv3 nutzt als Basis Links und keine Subnetze. Interfaces werden zu sogenannten Links verbunden. Dies ersetzt die Begriffe Netzwerk und Subnetz aus OSPFv2. Einem solchen Link können mehrere Präfixe zugewiesen werden. Außerdem ist es bei OSPFv3 möglich, dass Router über einen direkt verbundenen Link kommunizieren können, auch wenn sie über kein gemeinsames Präfix verfügen.

Link-State-Advertisements

Zwei LSA-Typen wurden umbenannt. Dabei wurde der Typ 4 LSA (AS Summary) in Inter-Area-Router LSA und der Typ 3 LSA (Summary) in Inter-Area-Präfix LSA umbenannt. Außerdem wurden noch zwei neue LSA-Typen eingeführt. Das sind zum einen der Link-LSA, dieser enthält die IPv6-Adressinformationen der lokalen Links und zum anderen der Intra-Area-Präfix LSA, der die IPv6-Adressinformationen der Router und Netzwerk Links enthält. Der Typ 5 LSA (Multicast OSPF) ist nicht mehr vorhanden.

Unbekannte LSA Typen werden je nach gesetztem LSA Handlungsbit entsprechend verteilt und nicht wie bei OSPFv2 verworfen.

Link-State-Protokoll Die einzelnen Status der Links in einem AS werden in einer Datenbank des Routers verwaltet. Der Aufbau der Datenbank erfolgt durch Austausch der Informationen mit benachbarten OSPF-Routern. Diese Informationen werden Link-State-Advertisements (LSA) genannt. Wie weit diese LSAs im Netzwerk verteilt werden, hängt vom gewählten Flooding-Scope ab. Folgende Einstellungen sind dabei möglich:

- AS Flooding-Scope - alle Router im AS
- Area Flooding-Scope - alle Router innerhalb der gleichen OSPF-Area
- Link Flooding-Scope - alle benachbarten Router

Die Verteilung, die auch Flooding (Flutung) genannt wird, verläuft von einem benachbarten Router zum nächsten. Dies ist auch der Grund, warum für OSPF stabile Beziehungen, auch Adjacencies genannt, zwischen den Nachbarn notwendig sind. Alle lokalen Interfaces eines Routers in einer Area werden mit den sogenannten LSA (Link-State-Advertisements) beschrieben. Weiterhin werden diese LSAs für IPv6-Routen von anderen Areas, für externe IPv6-Routen und für die Beschreibung von Links mit mehreren Nachbar Routern generiert. Sobald ein Router ein LSA erhält, schreibt er diese Information in seine Link-State-Datenbank (LSDB). Diese Datenbank bildet die Grundlage für die Berechnung der Routing-Tabelle. Für den Aufbau der Weiterleitungstabelle verwendet jeder Router genau den selben Algorithmus. Dieser führt zu einem Baum der kürzesten Wege zu jeder Route, welcher auch Shortest-Path-First Baum genannt wird. Die Kosten für eine Route werden durch ganzzahlige Metriken beschrieben, die den Interfaces zugewiesen sind. Normalerweise richtet sich die Metrik nach der physikalischen Leitungsgeschwindigkeit. Sobald sich die Bandbreite vergrößert, verkleinert sich die Metrik (bzw. die Kosten) und umgekehrt. Für eine realistische Abbildung der Kosten muss der jeweilige Betreiber des Netzes selbst eine geeignete Methode entwickeln. Auch die Hersteller der Router bieten meist eigene Formeln an. OSPF kann auch sogenannte Equal Cost Paths (gleichwertige Routen) in die Routing-Tabelle aufnehmen. Im Normalfall basiert die Auswahl des geeigneten Wegs auf der Absender- und Zieladresse, so dass Pakete mit gleicher Absender- und Zieladresse den gleichen Weg nehmen.

Areas Damit der Arbeitsspeicher und die CPU-Last des Routers nicht überfordert wird, was bei einer sehr großen LSDB durchaus der Fall sein kann, ist es möglich, das AS in mehrere Areas aufzuteilen. Dadurch kann die Datenbank schneller verarbeitet werden, denn mit der Einrichtung solcher Areas wird diese auch partitioniert.
Zur Identifizierung einer Area wird ihr eine Area ID zugeordnet. Diese als Dezimal-

schreibweise mit Punkt dargestellte 32-Bit-Zahl hat keine Bedeutung für die Adressierung. LSAs die mit dem Scope Area Flooding versehen sind, werden nur innerhalb der Area verteilt und in die zugehörige Area LSDB geschrieben. Jeder Router berechnet dann für jede Area separat den SPF-Baum (Shortest-Path-First-Baum). Die dabei erhaltenen Routen werden auch als Intra-Area-Routen bezeichnet. Routen, die sich nicht innerhalb einer Area befinden, können nur über sogenannte Area-Border-Router (ABR) erreicht werden. Um zu vermeiden, dass jede Area an alle anderen Areas angrenzt, gibt es zwei Hierarchiestufen. Die oberste bildet die Backbone-Area, daran schließen sich alle Nicht-Backbone-Areas an. Daraus folgt, dass jeder Area-Border-Router mindestens ein Interface in der Nicht-Backbone- und ein Interface in der Backbone-Area besitzen muss. Die ABR haben die Aufgabe, alle Routen der beiden Areas weiterzugeben, um so zu garantieren, dass alle Routen im gesamten AS verteilt werden.
Diese Hierarchie ermöglicht zwei Arten von Routing:

Intra-Area-Routing
: Sobald ein Paket mit einer Quell- und Zieladresse aus der gleichen Area verarbeitet wird, kommt die Area-LSDB zum Einsatz.

Inter-Area-Routing
: Befindet sich die Zieladresse eines Pakets außerhalb der Area, wird es über einen Area-Border-Router in die Backbone-Area weitergeleitet, um von dort an das eigentliche Ziel zu gelangen.

Der Vorteil dieser Hierarchie liegt eindeutig in der Verringerung des Verarbeitungsaufwands. Zum einen ist der Aufbau des Netzwerks in einer Area viel kleiner als im gesamten AS und kann demzufolge schneller berechnet werden (SPF-Baum). Zum anderen wird dieser Baum bei einer Änderung der Topologie nur in der zugehörigen Area neu berechnet und nicht im gesamten AS, da sich für die restlichen Areas die Topologie nicht geändert hat.

Backbone-Area Die Area, die alle ABR des autonomen Systems enthält, wird auch als Backbone-Area bezeichnet. Sie trägt die Area-ID 0.0.0.0 (Area 0) und enthält alle Routen der Backbone-Area selbst sowie die Routen aller Nicht-Backbone-Areas, die daran angeschlossen sind. Jeder Router der Backbone-Area muss mindestens einen direkten Link zu einem anderen Router der Backbone-Area besitzen. Durch den Einsatz virtueller Links müssen die Router nicht zwingend physisch zusammenhängen.

Nicht-Backbone-Area Zur Unterscheidung von den Backbone-Areas erhalten Nicht-Backbone-Areas eine eindeutige ID, die ungleich null sein muss. Weiterhin muss jeder Router dieser Area einen direkten physischen Link zu einem anderen Router der Area besitzen. Zum Anschluss an die Backbone-Area muss die Area einen Area-Border-Router besitzen. Dieser nutzt für jede Route, die er zwischen zwei Areas austauscht bzw. weitergibt, ein Inter-Area-Präfix-LSA. Für den Fall, dass die IPv6-Adressbereiche zusammenhängend sind, kann ein ABR die Routen auch zusammenfassen, um die Anzahl der LSAs sowie die Größe der Datenbank zu reduzieren. Eine Zusammenfassung der Routen (Summary Bildung) kann durch eine optimale Planung der IPv6-Adressbereiche stark vereinfacht werden.

Externe Routen IPv6-Routen können auch von nicht OSPF-Routern gelernt werden. Dazu ist ein sogenannter AS Boundary Router notwendig. Dieser besitzt mindestens ein Interface mit einer OSPF-Konfiguration und andere Routen aus anderen Quellen. Diese sogenannten externen Routen können statische Routen, RIP, BGP oder IS-IS Routen sein. Sobald diese importiert sind, können sie in das OSPF AS geflutet werden. Zur Kennzeichnung werden sie über ein AS-External-LSA bekannt gegeben. Bis auf die Stub-Areas erhalten alle Router die externe Route. Sobald sie ein Paket für eine externe Route erhalten, leiten sie dieses an den ASBR oder einen explizit dafür bestimmten Router weiter. Auch hier können die Routen mit zusammenhängenden IPv6-Adressen zusammengefasst und als eine einzige Route importiert werden.
Die Metriken der externen Routen können nicht ohne weiteres importiert werden, da sie mit denen im OSPF AS nicht vergleichbar sind. Um diesem Problem aus dem Weg zu gehen, wurden zwei Metriken für externe Routen entwickelt. Diese werden Extern-1 und Extern-2 genannt. Alle Routen, die sich in der unmittelbaren Nähe des ASBR befinden, bekommen die Metrik Extern-1 zu der jeder OSPF Router die Kosten von sich zum ASBR hinzufügt. Extern-2 bezeichnet die Metrik von Routen, die weiter vom ASBR entfernt sind. OSPF Router die, diese Route nutzen möchten, benutzten als Basis zur Metrikberechnung eine intern festgelegte Metrik, die größer als die größtmögliche OSPF Metrik ist.

Virtuelle Links Ein Tunnel zwischen zwei ABRs einer Nicht-Backbone-Area wird als virtueller bzw. logischer Link zum Austausch des Backbone-Datenverkehrs durch eine Nicht-Backbone-Area bezeichnet. Ein solcher Link ist Bestandteil der Backbone-Area und darf nur eine Nicht-Backbone-Area durchqueren, die keine Stub-Area sein darf. Verwendet werden virtuelle Links, um abgesetzte Areas, die keine physische Verbindung zur

Backbone-Area besitzen, anzubinden. Zudem können sie auch als redundante Links eingesetzt werden.

Stub-Area In einer Stub-Area sind keine externen Routen bekannt, da AS-External-LSAs in dieser Nicht-Backbone-Area blockiert sind, um die Größe der LSDB zu reduzieren. Da die Stub-Area somit keine Kenntnis über externe Routen hat, wird ihr mittels eines Inter-Area-Präfix-LSA die Default Route 0:0:0:0:0:0:0:0 bekannt gegeben. Diese Route hat eine sogenannte Stub-Metrik. Außerdem müssen alle Router, die sich in dieser Area befinden, wissen, dass es sich um eine Stub-Area handelt. Um dies zu realisieren, wird die External Capability deaktiviert. Mittels einer optionalen Konfiguration kann der ABR, der die Default Route für die Stub-Area bekannt gibt, so eingestellt werden, dass er auch keine Routen zu anderen Areas bekannt gibt. Diese Einstellung kann gewählt werden, wenn die Router der Stub-Area die Routen zu den anderen Areas nicht zwingend kennen müssen und die Default Route genügt. Stub-Areas mit dieser Konfiguration werden auch als totale Stub-Areas bezeichnet.
Für den Fall, dass ein Router in eine Stub-Area eingebunden werden muss, der über externe Routen verfügt und auch weiterhin verfügen muss, gibt es einen zusätzlichen LSA-Typ. Dieser NSSA-LSA funktioniert genauso wie ein AS-External-LSA mit der Änderung, dass er auch in Stub-Areas erlaubt ist. Wird dieser verwendet, bezeichnet man die Stub-Area als Not-So-Stubby-Area. Damit diese Funktionalität nutzbar wird, muss bei allen Routern die NSSA-Fähigkeit aktiviert werden, damit die entsprechenden Adjacencies aufgebaut werden können. Im übrigen gelten in Not-So-Stubby-Areas die selben Eigenschaften und Einschränkungen wie bei einer Stub-Area, außer dass diese nie eine totale Stub-Area werden kann.

Adjacencies

Zum Austausch der LSAs müssen die Router über einen stabilen Kommunikationskanal (Adjacency) verbunden sein. Dazu werden zuerst die Nachbarn mittels einer Hello-Nachricht gefunden. Die Interfaces der OSPF-Router haben spezielle Link-Typen: Transit-, Stub-, Virtueller-, Punkt-zu-Punkt-Link. Virtuelle- und Punkt-zu-Punkt-Links besitzen nur genau einen Nachbarn. Wenn mehrere Router am gleichen Netzwerk angeschlossen sind, handelt es sich um Transit-Links. Dabei ist es nicht notwendig, dass jeder Router zu jedem vorhandenen anderen Router eine Adjacency aufbaut. Daher wird auf jedem dieser Links ein Designated Router (DR) gewählt, der mit allen anderen Routern dieses Transit-Links eine Adjacency aufbaut und die Datenbanken synchronisiert. Zur Gewährleistung eines störungsfreien Betriebs wird auf diesen Links noch ein Backup-Designated-Router

(BDR) gewählt, der dieselben Verbindungen zu den anderen Routern aufbaut wie der DR. Der Stub-Link bezeichnet einen Link, auf dem kein Nachbar gefunden wurde. Die Transit- oder Punkt-zu-Punkt-Links können zu Stub-Links werden, wenn keine Nachbarn mehr vorhanden sind.

Hello Das Hello-Paket ermöglicht den Aufbau und das Fortbestehen der Adjacencies über eine bidirektionale Kommunikation sowie die Wahl des DR und BDR. Diese Pakete werden in regelmäßigen Abständen über jedes OSPF-Interface gesendet. Die Zieladresse des Hello-Pakets entspricht dabei der AllSPFRouters Multicast-Adresse, sofern es sich um einen broadcastfähigen Transit- oder einen Punkt-zu-Punkt-Link handelt. Die jeweiligen Nachbarn werden dabei dynamisch gefunden.

Wahl des DR/BDR In dem Moment, in dem die Betriebsbereitschaft eines mit OSPF konfigurierten Interfaces hergestellt ist, beginnt auch die Verarbeitung von Hello-Nachrichten. Punkt-zu-Punkt Links werden sofort aktiv.
Transit-Links gehen in einen Wartestatus, um den DR bzw. den BDR zu bestimmen. Dabei lauscht das Interface auf Hello-Nachrichten, um herauszufinden, ob schon ein DR oder BDR vorhanden ist. Das Interface selbst verschickt auch Hello-Pakete, die anzeigen, dass es noch keinen DR oder BDR kennt (Discovery-Modus). Sobald ein Paket mit gesetztem DR-Feld empfangen wurde, wird die Wahl abgebrochen. Wenn dies nicht der Fall ist, wird der Router zum DR ernannt, der die höchste Interface-Priorität besitzt. Sollte es mehrere Interfaces mit gleicher Priorität geben, so wird der Router mit der höchsten Router-ID zum DR. Die Wahl des BDR findet ebenso statt. Router, die weder DR noch BDR sind, werden als DR-other bezeichnet. Router, deren Interface-Priorität auf null gesetzt ist, erklären sich sofort als DR-other, denn sie nehmen nicht an der Wahl teil. Sobald ein DR während des Router-Dead-Intervals keine Hello-Nachrichten mehr versendet, wird er als ausgefallen angesehen und der BDR wird zum DR. Daraufhin wird ein neuer BDR gewählt. Damit ist gesichert, dass die vorhandenen Adjacencies nicht neu aufgebaut werden und die Datenbanken synchron bleiben. Für den Fall, dass der frühere DR wieder aktiv wird, erkennt dieser, dass DR und BDR vorhanden sind und setzt sich selbst auf DR-other. Dem Ausfall eines BDR folgt auch hier eine neue Wahl des BDR. Damit ist der Transit-Link aktiv und hat den Status DR, BDR oder DR-other.

Prüfung des Hello-Paketes Nur IPv6-konforme Hello-Pakete können durch den OSPF-Prozess verarbeitet werden. Für den Fall einer Authorisierung muss zudem eine korrekte Area-ID mitgeliefert werden. Danach wird das Paket intensiver durchsucht. Die Prüfung

des Router-Dead-Intervals sowie des Hello-Intervals stehen allem voran. Danach folgt die Prüfung des E- und N-Bits. Diese stehen für Akzeptanz externer Routen sowie die Fähigkeit NSSA zu unterstützen. Wenn diese Werte mit den voreingestellten Werten des Interfaces konform sind, kann der Nachbar anhand der Router-ID, die sich im OSPF-Header befindet, erkannt werden. Jedes Interface hält eine Liste mit den aktuellen Status der Nachbarn vor. Findet er zum gerade identifizierten Router eine volle Adjacency, wird nur das Hello-Interval zurückgesetzt. Im anderen Fall wird der Status auf Initialisierung gesetzt und der Router durchsucht die Nachbar-ID-Felder der Hello-Nachricht. Sobald er sich selbst in einem dieser Felder wiederfindet, gibt dies einen Hinweis auf eine beidseitige Kommunikation. Daraus ergibt sich, dass der Nachbar ein vom Router selbst versandtes Paket empfangen hat. Es folgt nun die Änderung des Status auf two-way. Im Folgenden kann der Router entscheiden, ob er mit dem Nachbarn eine Adjacency aufbauen möchte. In jedem Fall wird eine Adjacency eingegangen, wenn es sich um ein Punkt-zu-Punkt-Interface handelt. Bei einem Transit-Link wird eine Adjacency nur eingegangen, wenn der Router selbst oder der gefundene Nachbar ein DR oder BDR ist.
Sobald eine Adjacency aufgebaut ist, erfolgt die Synchronisierung der LSDB. Zu Beginn wird die Beschreibung der Inhalte ausgetauscht (Database Description Exchange). Danach sind den Routern die Inhaltsverzeichnisse des jeweiligen anderen Routers bekannt und sie können gezielt Informationen abrufen, die in ihrer eigenen Datenbank fehlen oder nicht mehr aktuell sind (Loading). Nach Abschluss dieser Phase haben beide Router die sogenannte volle Adjacency erreicht und synchrone Datenbanken. Um zu verhindern, dass die Adjacency abläuft, werden fortwährend Hello-Nachrichten gesendet.

Loading Zu Beginn der Loading-Phase vergleichen die Router den Inhalt der eigenen LSDB mit der des Nachbarn. Veraltete LSAs oder fehlende Einträge werden daraufhin angefordert. Zur Anforderung sendet der Router Link-State-Request-Pakete. Mittels dieser Nachrichten ist es zudem möglich, mehrere LSAs anzufordern. Daraufhin sendet der angefragte Router die geforderten LSAs zurück. Auch hier können mehrere LSAs in einem Paket versendet werden. Ein Link-State-Acknowledgement bestätigt den Empfang jeder einzelnen LSA. Durch das Ändern des Nachbarschaftsstatus auf Full wird angezeigt, dass alle Anforderungen erfüllt worden sind.

Link-State-Datenbank

Die Link-State-Datenbank (LSDB) bildet die Basis von OSPF. Diese Datenstruktur enthält Link-State-Advertisements (LSAs), die durch den Austausch dieser Informationen zwischen den Routern entstanden sind. Sie ermöglicht die Bildung des logischen Baums

des Netzwerks, in dem nur die Wege mit den besten (geringsten) Kosten (Shortest Path) zu den jeweiligen Routern eingetragen werden. Dabei erstellt jeder Router einen eigenen Baum, in dem er selbst die Wurzel darstellt. Für die Berechnung des Baumes wird der im nächsten Abschnitt erläuterte Dijkstra Algorithmus 2.5.1 verwendet. Zu Beginn erfolgt die Bildung aller Routen in der eigenen Area (Intra-Area-Baum). Danach werden Inter-Area und externe Routen an den ABR- oder ASBR-Ast des Baums angefügt. Zum Schluss wird der Baum ausgelesen und die gesamten Routen werden in die jeweiligen Sektionen der Routing-Tabelle mitsamt ihren Kosten und der Next-Hop-Adresse (Link-lokale-Adresse des ersten Routers der Route) eingetragen. Die Sektionen enthalten Inter-Area-, Intra-Area-, Extern-1- und Extern-2-Routen.

Dijkstra Algorithmus

OSPF-Router bilden aus der LSDB den für sie gültigen Baum der kürzesten Wege (SPF-Baum) durch Teilung der LSDB in verschiedene Sektionen. Die Sektionen werden je nach Flooding-Scope erstellt, wie im Folgenden zu sehen:

Area-LSDB enthält alle LSAs mit Area-Flooding-Scope,

Link-LSDB enthält alle LSAs mit Link-lokalem Flooding-Scope und

AS-LSDB enthält alle LSAs mit AS-Flooding-Scope.

Ein Area-Border-Router führt für jede an ihn angeschlossene Area eine eigene Area-LSDB. Um einen SPF-Baum zu bilden, sind drei Schritte notwendig. Für die ersten beiden Schritte wird die Area- und die Link-LSDB benötigt. Ein ABR muss diese für jede Area-LSDB ausführen, da er jeweils einen eigenen Baum der kürzesten Wege für jede Area baut.

1. Intra-Area-Routen In der ersten Phase erstellt der Router den Baum für jede an ihn angeschlossene Area unter Zuhilfenahme der Router- und Network-LSAs aus der Area-LSDB. Diese enthalten die eigentlichen Verbindungen für die Festlegung der Area-Topologie.
Der Router setzt sein eigenes Router-LSA als Wurzel des Baumes. Die Link-Einträge im LSA bilden jeweils einen neuen Ast, der entweder auf ein Network-LSA (Transit-Link) oder auf ein anderes Router-LSA (virtueller oder Punkt-zu-Punkt Link) zeigt. Zur Erkennung der LSAs am Ende des Zeigers dient die Nachbar- oder Interface-ID. Diese bilden nun die möglichen Äste, an die die jeweilige Metrik geschrieben wird. Derjenige, der die

kleinste Metrik besitzt, wird zusammen mit seiner LSA fest in den Baum aufgenommen, da er den kürzesten Weg darstellt. Als Nächstes wird die LSA genauer untersucht. Was dabei genau passiert hängt vom Typ der LSA ab.

Network-LSA

Hier bilden alle im Network-LSA enthaltenen Router neue Äste aus. Dabei kennzeichnen die Router-IDs die Router-LSAs am Ende des Zeigers. Sofern ein Router-LSA schon fest im Baum vorhanden ist, wird es ignoriert. Andernfalls werden diese Äste zusammen mit ihren Router-LSAs fest in den Baum aufgenommen. Die zugehörige Metrik wird nicht mit in den Baum aufgenommen, da die Metrik zum Netzwerk schon im Baum vorhanden ist.

Router-LSA

Die Link-Einträge im Router-LSA bilden in diesem Fall die neuen Äste und zeigen dabei auf ein Router- oder Netzwerk-LSA, wobei die Nachbar- oder Interface-ID den LSA am Ende des Zeigers kennzeichnet. Sie werden nur temporär (Kandidat) in den Baum aufgenommen und der Ast wird mit der Metrik versehen. Router- und Network-LSA, die schon fest im Baum vorhanden sind, sind zu ignorieren.

Jedes LSA eines Kandidaten mit den kleinsten Kosten wird fest in den Baum aufgenommen. Sobald der gleiche Kandidat an einer anderen Stelle im Baum mit höheren Kosten vorhanden ist, kann dieser LSA aus dem Baum entfernt werden. Dieser Prozess wird immer weiter fortgeführt, bis keine Network- oder Router-LSAs mehr zur Untersuchung vorhanden sind. Sobald zwei Kandidaten die gleichen Kosten besitzen, müssen beide im Baum bleiben. Danach ist der SPF-Baum zwar fertig, enthält aber noch keine Adressinformationen. Diese sind in den Intra-Area-Prefix-LSAs zu finden. Sie werden daraufhin an die Äste gehangen, an denen sich die Router- und Network-LSAs, die vom Intra-Area-Prefix-LSA referenziert sind, befinden. Ist dieser Vorgang abgeschlossen, sind alle Intra-Area IPv6-Präfixe der einzelnen Routen und die zugehörigen Wege bekannt. Zur Fertigstellung fehlen nun noch die Next-Hop Informationen jeder Route, also die Link-lokalen-Adressen der direkt an den Router angeschlossenen Nachbarn, die in den Link-LSAs der angeschlossenen Links vorhanden sind. Auch sie werden im Baum an den entsprechenden Stellen platziert. Abschließend schreibt der Router alle Routen, die zugehörigen Next-Hop-Adressen sowie die Kosten in die OSPF-Routing-Tabelle. Bei diesen Routen handelt es sich nur um Intra-Area-Routen.

2. Inter-Area-Routen In der zweiten Phase lokalisiert der Router alle Inter-Area-Link-LSAs in der zugehörigen Area, da diese alle Routen der anderen Areas repräsentieren.

Dazu muss der ABR, der diese LSAs erstellt hat, schon im Baum vorhanden sein. Nun werden diese LSAs an das Router-LSA des ABR angehängt. Dazu wird die Metrik aufgeschrieben. Die Kosten ergeben sich aus den Kosten zum ABR sowie der im Inter-Area-Link-LSA notierten Metrik. Für den Fall, dass eine Route nicht nur einmal vorkommt, verbleibt die Route mit den kleinsten Gesamtkosten im Baum. Bei gleichen Kosten bleiben alle Routen mit diesen Eigenschaften erhalten. Nachdem alle Inter-Area-Link-LSAs überprüft sind, werden die Routen in die OSPF Routing-Tabelle eingetragen. Die dafür notwendige Next-Hop-Adresse ergibt sich aus dem angeschlossenen Router, der auf dem Weg zum ABR liegt. Wenn der baumbildende Router selbst ein ABR ist, werden nur Inter-Area-Link-LSA von anderen ABRs in der gleichen Area verwendet.

3. Externe-Routen Um zu erreichen, dass alle ASBRs in der entsprechenden Area auch bekannt sind, identifiziert der Router alle Inter-Area-Router-LSAs in der zugehörigen Area und hängt diese an die ABRs an. Damit ist sichergestellt, dass die internen ASBRs ebenso wie die außerhalb der Area befindlichen ASBRs bekannt sind. Die AS-External-LSAs hängt der Router an die entsprechenden ASBRs im Baum und vermerkt dazu die Metriken. Die Präfixe der neu angehängten LSAs werden von den externen Routen übernommen. Die Gesamtkosten zu den externen Routen werden nun abhängig vom jeweiligen Typ berechnet. Handelt es sich um Extern-1-Routen, wird die Metrik zum ASBR zu den Kosten der Extern-1-Route hinzugezählt. Bei Extern-2-Routen wird ein Wert größer als die größtmögliche OSPF-Metrik zur Extern-2-Route addiert. Sobald eine Route gleichen Typs mehrfach vorkommt, wird nur die mit den kleinsten Kosten behalten. Wenn mehrere gleiche Routen die gleichen Kosten haben, werden alle behalten. Nach Prüfung aller AS-External-LSAs werden die Routen als Extern-1- und Extern-2-Routen in die OSPF-Routing-Tabelle übernommen. Wenn eine gleiche Route vom Typ Extern-2 wie auch vom Typ Extern-1 vorhanden ist, wird die des Typs Extern-1 bevorzugt behandelt. Die zugehörigen Next-Hop-Adressen ergeben sich entweder wieder aus dem direkt angeschlossenen Router auf dem Weg zum ASBR oder zu einer Weiterleitungsadresse, die im AS-External-LSA mit dem F-Bit bekannt gegeben wurde.

Damit ist die OSPF-Routing-Tabelle fertig berechnet und kann an den Routing-Prozess übergeben werden.

Flooding

Änderungen in einem Netzwerk bewirken auch Änderungen bei den Link-Status. Beispiele solcher Änderungen, die auch den Link-Status ändern, können folgende sein:

- Eine Nachbarschaftsbeziehung ändert sich und wird zu einer vollen Adjacency.
- Auf einem Transit-Link wird der Designated-Router geändert.
- IPv6-Präfixe werden hinzugefügt oder entfernt.
- Externe Routen kommen hinzu oder werden entfernt.
- Ein Router Interface Status ändert sich.

Sobald ein Router eine solche oder ähnliche Änderung bemerkt, aktualisiert er die entsprechenden LSAs und verbreitet sie neu, indem er die Sequenznummer erhöht und das LSA dem Flooding-Prozeß übergibt. Je nach gewähltem Flooding-Scope werden danach die neuen LSAs verbreitet.
Router senden die neuen LSAs an alle Nachbarn, mit denen eine volle Adjacency besteht. Je nach Flooding-Scope setzt sich dieser Prozeß beim empfangenden Router weiter fort. Sobald ein Router das LSA erhält, überprüft er, ob dieses schon in der Datenbank existiert. Ist das nicht der Fall, wird es aufgenommen. Ansonsten wird die Sequenznummer für das existierende LSA überprüft. Das LSA wird ersetzt, wenn die Sequenznummer des erhaltenen LSAs größer ist als die in der Datenbank. Im Anschluss daran prüft der Router, an welche Interfaces er das erhaltene LSA weitergeben muss (abhängig vom Flooding-Scope). Im Normalfall wird das LSA nicht über das Eingangsinterface verteilt. Ein DR bildet dabei die Ausnahme. Wenn ihn ein LSA über ein Eingangsinterface erreicht, für das er DR ist und dies nicht von einem BDR kam, wird es auch über das Eingangsinterface geflutet, damit alle DR-Other Router auf diesem Link erreicht werden. Ältere oder gleich alte LSAs werden nie weitergeleitet, um Schleifenbildung sowie ein endloses Flooding eines LSAs zu verhindern.
Ein Link-State-Acknowledgement-Paket bestätigt den Erhalt eines LSAs. Erhält ein Router ein LSA mit einer kleineren Sequenznummer, als das sich in der Datenbank befindliche, so kann er den Erhalt durch das Versenden des LSAs aus der Datenbank bestätigen.

LSA-Aging Jedes LSA führt zusätzlich zur Sequenznummer noch ein Age-Feld, welches das Alter des LSA in Sekunden misst. Jeder Router erhöht es in seiner LSDB für jedes LSA. Nach Weitergabe des LSA wird ein Übertragungsdelay dem Alter hinzugefügt. Das maximale Alter (MaxAge) eines LSAs beträgt 3600 Sekunden. Im Normalfall wird das Alter durch den Herausgeber der LSA meist schon nach dem halben MaxAge (1800 Sekunden) aktualisiert, damit es nicht zum Ablaufen des LSAs kommt. Sobald ein LSA das maximale Alter erreicht hat, wird es nicht mehr für die Berechnung genutzt und aus der Datenbank entfernt.

Damit ein LSA vorzeitig ungültig wird, kann der Herausgeber das LSA vorzeitig altern lassen. Dazu wird das LSA neu herausgegeben und das Alter auf das Maximum gesetzt. So kann ein ASBR beispielsweise eine externe Route, die nicht mehr existiert, zurücknehmen.
Eine weitere Möglichkeit für das Altern eines LSA kann das Erreichen der höchstmöglichen Sequenznummer sein. Auch hier muss das LSA zurückgezogen und mit der initialen Sequenznummer neu herausgegeben werden. Dieser Fall wird aller Wahrscheinlichkeit nach nicht eintreten, selbst wenn die Sequenznummer in jeder Sekunde einmal geändert würde, wäre die höchste Sequenznummer erst nach mehr als 136 Jahren erreicht.

Empfang eigener LSAs In einem redundanten Netzwerk kann es vorkommen, dass ein Router seine eigenen, selbst erstellten LSAs erhält. Normalerweise werden diese verworfen. Eine Ausnahme bildet der Fall, dass die Sequenznummer des empfangenen LSAs höher ist, als die des LSAs in der LSDB. Dies sollte normalerweise nicht passieren, da nur der Herausgeber eines LSAs die Sequenznummer erhöhen darf. Im Falle eines Ausfalls eines Routers kann es sein, dass dieser nach dem Neustart seine LSAs mit der initialen Sequenznummer ausgibt und das frühere LSA noch in den anderen LSDBs der anderen Router existiert. Um diesen Zustand zu beheben, altert der Router, der das selbst erstellte LSA mit höherer Sequenznummer erhält, dieses vorzeitig aus. Dazu erstellt er ein neues LSA mit einer noch höheren Sequenznummer und setzt das MaxAge-Feld auf das maximale Alter. Danach wird das LSA geflutet und aus allen LSDBs entfernt. Nun kann das richtige LSA ausgegeben werden.

Vorgehen bei unbekannten LSAs Sobald ein Router ein unbekanntes LSA bekommt und bei diesem das U-Bit im LSA-Header gesetzt ist, wird es gemäß dem Flooding-Scope geflutet. Ist das U-Bit dagegen nicht gesetzt, wird der Flooding-Scope dafür auf Link-lokal geändert.

2.5.2 BGPv4

Das Border Gateway Protokoll (BGP) in der Version 4 ist kein extra für IPv6 entwickeltes Protokoll. Die IPv6-Funktionalität ergibt sich durch die Multiprotokollerweiterung, die es zulässt, neben IPv4 auch andere Netzwerkprotokolle zu unterstützen. Diese Erweiterung ist in RFC 4760 [T. Bates, 2007] definiert. Darauf aufbauend definiert RFC 2545 [P. Marques, 1999] die Erweiterung für IPv6. Im Folgenden wird die Funktionalität von BGP erläutert, um darauf aufbauend die Erweiterung für IPv6 zu erklären.

BGPv4-Funktionen

Jedes eigene autonome System benutzt ein internes Routing-Protokoll, um Routing-Informationen innerhalb des AS auszutauschen. Dagegen ist BGP ein externes Routing-Protokoll. Es tauscht Routing-Informationen zwischen den einzelnen AS aus. Die Nummerierungsstelle vergibt für jedes AS eine eindeutige AS-Nummer. BGP unterteilt in verschiedene AS-Typen.
Das **Transit-AS** leitet Datenverkehr an andere AS weiter und besitzt demzufolge Verbindungen zu anderen AS. Ein solches Transit-AS kann Routing-Informationen an jedes angeschlossene AS weiterleiten. Im Normalfall sind große Internet Service Provider ein Transit-AS.
Ein weiterer AS-Typ ist das **Stub-AS**. Es hat nur eine Verbindung zu einem Transit-AS. Dabei wird der komplette Datenverkehr über diesen einen Link geleitet. Beispiele dafür sind Campus- oder Firmennetzwerke sowie kleine ISPs.
Den letzten AS-Typ bildet das **Multihomed-Nontransit-AS**. Diese besitzen mehrere Verbindungen zu einem oder mehreren Transit-AS. Trotzdem darf ein solches AS keine Routing-Informationen weiterreichen. Jeglicher Datenverkehr, der nicht in dieses AS gehört, wird nicht weitergeleitet. Die mehrfache Anbindung ist nur aus Redundanz- oder Lastverteilungsgründen erlaubt.
Wie auch bei OSPF müssen, bevor Routing-Informationen zwischen zwei BGP-Routern ausgetauscht werden können, Nachbarschaftsbeziehungen aufgebaut werden. Dazu wird in diesem Fall eine TCP-Verbindung verwendet. Die Partner, die BGP-Informationen austauschen, nennt man BGP-Peers oder auch BGP-Speakers. Nachdem eine TCP-Verbindung erfolgreich aufbaut worden ist, wird die BGP-Verbindung aufgebaut, über die dann BGP-Nachrichten ausgetauscht werden. Die wichtigste Nachricht ist die sogenannte Update-Nachricht, denn sie enthält die Routing-Informationen.
Eine BGP-Route besteht aus einer Erreichbarkeitsinformation des Netzwerks, auch NLRI (Network Layer Reachability Information) genannt und zugehörigen Attributen. Die NLRI enthält die eigentliche Route, die ein Netzwerk oder ein Bereich von Netzwerken repräsentieren kann sowie zusätzliche Pfadattribute. Diese Attribute enthalten Informationen zur Next-Hop-Adresse oder auch die AS-Nummern, die beim Informationsaustausch durchlaufen wurden. Auf der Basis dieser Pfadattribute wird das Routing realisiert. Erhält z.B. ein Router eine BGP-Route mit seiner eigenen AS-Nummer, so wird diese verworfen, da es sich aller Wahrscheinlichkeit nach um eine bekannte Route handelt und es bei der Verwendung dieser zu einer Schleifenbildung kommen kann.
Der Austausch von Routing-Informationen bei BGP erfolgt zwischen zwei Peers unter

Beachtung der jeweiligen Regeln (Policies). Diese bestimmen welche BGP-Routen weitergegeben oder empfangen werden dürfen.

Aufbau einer BGP-Verbindung Um Routing-Informationen austauschen zu können, muss eine BGP-Verbindung etabliert werden. Wenn beide Peers gleichzeitig versuchen, eine BGP-Verbindung aufzubauen, kann es vorkommen, dass zwei parallele Verbindungen aufgebaut werden. Damit dies nicht zu Problemen führt, muss einer der beiden Partner nachgeben. Dafür wird jedem Peer ein eindeutiger BGP-Identifier zugewiesen. Jene Verbindung, die mit dem höheren BGP-Identifier aufgebaut wurde, bleibt bestehen. Dieser Austausch der Identifier und der Aufbau der Verbindung wird durch die BGP-OPEN Nachricht realisiert. Entsprechend ihrer Policies werden dann die BGP-Routen über die erstellte Verbindung ausgetauscht. Danach werden nur noch geänderte BGP-Routen übermittelt. Um zu verhindern, dass eine BGP-Verbindung abläuft, werden kontinuierlich BGP-KEEPALIVE Nachrichten versendet. Fehler beim Verbindungsaufbau oder beim Austausch der Routen werden mittels der BGP-NOTIFICATION Nachricht signalisiert. Für die Verbindungen gibt es die folgenden Typen:

EBGP

Auch externe BGP-Verbindung genannt, verbindet Peers, die sich in unterschiedlichen AS befinden. Dabei dürfen BGP-Routen, die von externen Peers ankommen, an alle anderen Peers weitergeleitet werden.

IBGP

Peers, die sich im gleichen AS befinden, sind durch sogenannte interne BGP-Verbindungen gekoppelt. Sie dürfen keine BGP-Routen weitergeben, die von anderen internen Peers ankommen. Nur Router von externen Peers dürfen weitergegeben werden. Jeder interne Peer baut mit jedem anderen internen Peer eine Verbindung auf, so dass es zu einer Vollvermaschung kommt.

BGP-Routen und Policies Router, die BGP verwenden, legen ihre Routing-Informationen und Policies in verschiedenen Routing-Information-Bases (RIB) ab. So werden eingehende UPDATE-Nachrichten, die mögliche oder zurückgezogene Routen enthalten, in die Adj-RIB abgelegt. Sofern die Route schon vorhanden, ist wird sie in der Adj-RIB ersetzt. Jeder möglichen Route wird eine Präferenz zugewiesen, die von der Eingangspolicy abhängt, wenn sie von einem externen Peer kommt. Erhält der Router die Route von einem internen Peer, so wird die Präferenz auf einen vorher gesetzten Wert festgelegt. Zurückgezogene Routen werden aus der Adj-RIB entfernt.

Jede Route, die sich in der Adj-RIB befindet, muss nun zwei Prüfungen überstehen, damit sie nicht verworfen wird. Als erstes wird getestet, ob die IP-Adresse des Next-Hop Pfadattributes über die lokale Routingtabelle erreichbar ist. Der zweite Test überprüft, ob die lokale AS-Nummer nicht im AS-PATH Pfadattribut enthalten ist. Sind die beiden Prüfungen bestanden, so wird die Route entsprechend der Eingangspolicy akzeptiert oder verworfen und bei ersterem in die Loc-RIB kopiert. Sobald mehrere Routen mit dem gleichen Ziel existieren, wird nur die Route mit der höchsten Präferenz geprüft. Bei gleichen Präferenzen greift eine Entscheidungsregel aus RFC 4271 [Y. Rekhter, 2006].
Routen, die sich in der Loc-RIB befinden, werden nun in die lokale Routingtabelle eingetragen. Die einzutragende Next-Hop Adresse ergibt sich aus der Next-Hop Adresse zur Route, die mit der in dem Pfadattribut NEXT-HOP eingetragenen IP-Adressen übereinstimmt. Alle Routen, die sich in der Loc-RIB und in der lokalen Routingtabelle befinden, können an externe Peers weitergegeben werden. Routen aus der Loc-RIB von externen Peers können an die internen Peers weitergegeben werden. Eine Abgangs-Policy gibt die Routen dann an die jeweilige peer-spezifische Adj-RIB-out. Mithilfe dieser Policy können auch Routen zusammengefasst oder entsprechende Pfadattribute verändert werden.
Sobald eine Änderung in der Adj-RIB-out stattfindet, wird eine BGP-UPDATE Nachricht am entsprechenden Peer verschickt, die nur geänderte Routen enthält.

BGPv4-Erweiterung für IPv6

Eine BGP-Erweiterung die IPv6-konform ist, muss alle Vorkommnisse von IPv4-Adressen im Protokoll berücksichtigen und entsprechend anpassen. Aus diesem Grund sind drei Elemente erweitert worden:

- BGP-Identifier in der OPEN-Nachricht und Pfadattribut AGGREGATOR,
- NLRI in UPDATE-Nachrichten und
- NEXT-HOP Pfadattribut in der UPDATE-Nachricht.

BGP-4 wurde mit der Multiprotokollerweiterung unabhängig vom Protokoll gemacht, indem Multiprotokoll-NLRI und zugehörige Next-Hop Informationen angepasst wurden. Dieses Konzept ist im RFC 2858 [T. Bates, 2000] vereinbart und wurde nun auf IPv6 angewandt, welches im RFC 2545 [P. Marques, 1999] definiert ist.
Zur Verbreitung und zum Zurücknehmen von Routen wurden für IPv6 zwei neue Pfadattribute definiert. Der Identifier für BGP bleibt dabei unverändert. Dies ist auch der Grund, warum jeder Router immer eine lokale IPv4-Adresse konfiguriert haben muss. Mithilfe

einer OPEN-Nachricht tauschen sich die Router aus, ob sie IPv6 unterstützen und verwenden möchten. Der Aufbau der BGP-Verbindung und die Auswahl der Route bleibt ansonsten gleich. Hier müssen lediglich die Routerhersteller die RIBs und die Policies anpassen.
Die UPDATE Nachricht, die nur IPv6-NLRIs bekannt gibt und keine IPv4-NLRIs enthält, setzt die Länge aller zurückgezogenen Routen auf null, da keine IPv4-Routen benutzt werden. Die bekanntgegebenen oder zurückgezogenen IPv6-Routen werden über die Pfadattribute MP-UNREACH-NLRI und MP-REACH-NLRI übermittelt, wobei das NEXT-HOP Pfadattribut ignoriert wird.
So kann eine UPDATE-Nachricht in der Theorie IPv6- und IPv4-NLRIs mit gleichen Pfadattributen und im gleichen Paket transportieren. Dabei können alle Felder und Attribute verwendet werden, sofern der Router die IPv6- und IPv4-NLRIs in verschiedenen RIBs ablegen kann. Die beiden neuen Pfadattribute enthalten folgende Informationen:

MP-REACH-NLRI
: Dieses Attribut enthält die IPv6-NLRIs und die dazugehörigen Next-Hop IPv6-Adressen. Es ist optional und transitiv.

MP-UNRECH-NLRI
: Dieses optionale und transitive Attribut enthält ungültige IPv6-NLRIs, die vom sendenden Peer zurückgezogen wurden. Diese müssen aus der RIB entfernt werden.

2.6 Protokolle höherer Ebenen

Protokolle, die sich in einer höheren OSI-Schicht befinden, bedürfen weniger Anpassungen, da sich die Funktionalität größtenteils nicht geändert hat. Die wichtigste Änderung ist an den Stellen im Protokoll nötig, wo eine IP-Adresse benutzt wird. Ein solches Feld muss für die Verwendung der 128-Bit-Adressen angepasst werden.

2.6.1 DHCP

Das Dynamic Host Configuration Protocol (DHCP) hat eine weite Verbreitung in IPv4-Netzwerken gefunden, um die Geräte mit IPv4-Adressen und zusätzlichen Informationen (DNS-Server,...) zu versorgen. Im Grunde ist dies in einem IPv6-Netzwerk nicht mehr notwendig, da IPv6-Geräte automatisch mit einer Adresse konfiguriert werden können (Stateless Adress Autoconfiguration). Dieser Mechanismus ist in Abschnitt 2.4.2 erläutert.

Trotzdem ist der Einsatz von DHCP in einem IPv6-Netz weiterhin möglich. Dieser wird dann Stateful Adress Autoconfiguration genannt. Diese Variante ist notwendig, wenn ein spezielles IPv6-Adressierungsschema verwendet werden soll oder keine Verwendung der MAC-Adresse in der IPv6-Adresse zulässig ist. Weiterhin wäre das Nichtvorhandensein eines IPv6-Routers im Subnetz ein Einsatzgebiet.
Die beiden Verfahren zur Konfiguration der Geräte mit IPv6-Adressen und weiteren Informationen sind kombinierbar. So kann die IPv6-Adresse durch eine Stateless Address Autoconfiguration erstellt und DHCP dazu genutzt werden, zusätzliche Konfigurationseinstellungen (DNS-Server,...) vorzunehmen.
Die Varianten DHCPv4 für IPv4-Adressen und DHCPv6 für IPv6-Adressen sind nicht zueinander kompatibel. Das bedeutet, dass für die beiden Protokolle jeweils ein separater Server einzurichten ist. Die Spezifikation von DHCPv6 ist im RFC 3315 [R. Droms, 2003] nachzulesen.

2.6.2 DNS

Gerade für IPv6 ist die Verwendung von DNS umso wichtiger, da sich die IPv6-Adressen viel schlechter merken lassen als eine wesentlich kürzere IPv4-Adresse. In gemischten IPv4/IPv6-Umgebungen bekommt ein Host für jede IP-Adresse einen extra DNS-Eintrag, damit er durch die Namensauflösung sowohl über die IPv4 als auch über die IPv6-Adresse erreichbar ist. Für IPv6 DNS-Einträge wurde im RFC 3596 [S. Bhattacharyya, 2003] der AAAA-Record definiert.
Seit der Version 9 des BIND-Servers wird IPv6 vollständig unterstützt. Jede Implementierung, die auf dieser Version aufsetzt, kann IPv6-DNS-Einträge verwenden.
Auch der DNS-Resolver muss in einem IPv6-Netzwerk Anfragen nach AAAA-Records stellen können, um Namen in IPv6-Adressen aufzulösen. Ein Dual-Stack-Rechner wird zur Auflösung immer eine Abfrage auf A und auf AAAA-Records an den DNS-Server stellen. Wenn der gesuchte Host auch Dual-Stack unterstützt und beide Einträge auf dem DNS-Server vorhanden sind, erhält der Anfragende auch beide Einträge zurück.
Ob ein DNS-Server für IPv6 verwendet werden kann, muss genauer untersucht werden. Zum einen kann damit gemeint sein, ob der Server AAAA-Einträge verwalten und herausgeben kann. Dabei ist auch zu beachten, dass auf einem Gerät, welches beide Protokolle zur Verfügung stellt und somit auch mehrere DNS-Einträge besitzt, nicht jeder bereitgestellte Dienst des Geräts auch beide Protokolle unterstützt. Zum anderen kann bei der IPv6-Tauglichkeit die Erreichbarkeit über das Internetprotokoll der Version 6 gemeint sein. Dabei ist zu beachten, dass nicht alle DNS-Resolver, auch wenn diese mit AAAA-

Einträgen umgehen können, ihre Anfragen per IPv6 versenden. So kann ein A-Eintrag ebenso per IPv6 abgerufen werden wie ein AAAA-Eintrag über IPv4.

2.6.3 FTP

Das File-Transfer-Protocol ist dafür ausgelegt, 32-Bit-Adressen zu benutzen und über IPv4 transportiert zu werden. Eine in RFC 2428[M. Allman, 1998] definierte Erweiterung ermöglicht die Verwendung von FTP über IPv6. Zusätzlich dazu ist es notwendig, das der FTP-Server für das Aushandeln des Protokolls bei der Eröffnung der FTP-Session einen Mechanismus vorhalten kann, der dies realisiert. Die Erweiterung ersetzt den PORT-Befehl, der für das Aushandeln eines alternativen Ports für die Datenverbindung Verwendung findet. Ersetzt wurde dieser Befehl durch den EPRT-Befehl. Er übernimmt nun diese Aufgabe und unterstützt auch eine IPv6-Adresse. Außerdem unterstützt der Befehl die Angabe des Netzwerkprotokolls, so dass auch andere Protokolle verwendbar sind. Weiterhin wurde der PASV-Befehl durch den EPSV-Befehl ersetzt, der dazu genutzt wird, den Server in den sogenannten *passive-mode* zu versetzen. In diesem Modus hört er auf einem speziellen Port ohne selbst eine Session initialisieren zu müssen. Der neue Befehl für diesen Modus ermöglicht die Auswahl des Protokolls und die Benutzung von IPv6-Adressen.
Zudem beinhaltet die Erweiterung Optimierungen für das Protokoll und unterstützt die Verwendung von FTP über NAT und Firewalls besser.
Auch das für Dateitransfer immer häufiger verwendete Secure Copy (SCP), welches die Verschlüsselung des Transfers und damit die Sicherheit der Passwörter erhöht, wurde für IPv6 portiert.

2.6.4 Webserver / Clients

Um über IPv6 im Internet surfen zu können, müssen sowohl die Clients, also die Webbrowser, als auch die Webserver IPv6 unterstützen. Viele aktuelle Webserver unterstützen bereits IPv6. Zu beachten ist lediglich, dass der verwendete Server so eingerichtet werden muss, dass er über IPv6 auf dem Port 80 ansprechbar ist.
Der Webbrowser, der eine IPv6-Seite besuchen möchte, muss einen DNS-Request für einen AAAA-Record absetzen können. Dies unterstützen schon die meisten Browser. Bei einigen Browsern ist es möglich, die IPv6-Adresse eines Servers direkt einzugeben. Dazu ist die sogenannte *literal IPv6 Address* zu verwenden. Dies bedeutet lediglich das Umschließen der IPv6-Adresse mit eckigen Klammern, um so eine Abgrenzung zwischen IPv6-Adresse und Port zu ermöglichen.

Wenn der Browser einen Proxy verwendet, so muss dieser selbstverständlich auch über IPv6 erreichbar sein.
Da es in der Übergangszeit zwischen IPv4 und IPv6 viele Webseiten im Internet geben wird, die nur das eine oder nur das andere Protokoll beherrschen, ist der Einsatz eines Proxies eine Möglichkeit, um Webseiten im Internet einer Vielzahl von Nutzern zur Verfügung zu stellen. So könnte beispielsweise ein IPv4-Client, der eine IPv6-Seite besuchen möchte, dies über einen Proxy durchführen, der einen Dual-Stack besitzt. Dieses Szenario kann auch in umgekehrter Form realisiert werden, indem ein IPv6-Client auf eine IPv4-Seite über einen Dual-Stack-Proxy zurückgreift. Auch seitens der Webseitenbetreiber kann ein Proxy installiert werden, so dass ein IPv4-Webserver mithilfe eines Dual-Stack-Proxies von Clients beider Protokolle zu erreichen ist.

2.6.5 E-Mail Protokolle

Die meisten bekannten E-Mail Protokolle unterstützen IPv6. Bei der Verwendung ist darauf zu achten, dass gerade die Namensauflösung bei E-Mail-Verkehr eine wichtige Rolle spielt und deshalb immer gewährleistet sein muss. Auch die aktuellen Versionen der Mailserver, beispielsweise *Postfix*, unterstützen die Kommunikation über IPv6.
Weiterhin ist bei dem Betrieb eines IPv6-Mailservers zu beachten, dass ein Blacklisting mit IPv6-Adressen zur Verhinderung von SPAM kaum Wirkung zeigen wird, da zu erwarten ist, dass für jede SPAM-E-Mail aufgrund der großen Anzahl von IPv6-Adressen einfach eine neue Adresse verwendet wird. [9] Ein content-basierter Filter ist daher sinnvoller.

[9] Wie wirkt sich IPv6 auf E-Mail Verkehr und Anti-SPAM-Filter aus? `http://www.searchsecurity.de/themenbereiche/applikationssicherheit/e-mail-sicherheit/articles/313778/`

Kapitel 3

Realisierung von IPv6 in einem vorhandenen Unternehmensnetzwerk

In diesem Kapitel wird auf die Vorgehensweise der Migration in einem Unternehmensnetzwerk eingegangen. Der erste Teil beschäftigt sich mit den drei häufigsten Migrationsstrategien und wie diese für einen Übergang von IPv4 zu IPv6 mit möglichst wenig Ausfällen der bestehenden Netzwerkstruktur eingesetzt werden können.
Der Aufbau und die Durchführung des Tests für die Migration des Netzwerks der Firma perdata ist Inhalt des zweiten Teils dieses Kapitels.
Zum Schluss dieses Abschnittes wird auf die Kosten der Migration sowie die Unterstützung der Hersteller für einen reibungslosen Ablauf eingegangen.
Da bereits bei der Entwicklung des neuen Internetprotokolls offensichtlich war, dass eine Umstellung der vorhandenen, mit den Jahren gewachsenen Struktur nicht ohne weiteres zu bewältigen ist, wird seitdem an verschiedenen Übergangsmechanismen gearbeitet. Gerade in den letzten Jahren sind spezielle Arbeitsgruppen gegründet worden, die sich intensiv mit verschiedenen Szenarien der Migration beschäftigen. Für eine Umstellung in einem Unternehmen wird gerade die Kombination der unterschiedlichen Techniken zum Erfolg führen. Wichtig für eine Umstellung ist die Tatsache, dass es keine Migration von heute auf morgen geben wird. Vielmehr wird sie in kleinen überschaubaren Schritten erfolgen. Somit ist denkbar, dass nur einzelne Knoten oder Teilnetze umgestellt werden bzw. Dual-Stack nutzen.
Die ersten Übergangsmechanismen sind im RFC 4213 [E. Nordmark, 2005] spezifiziert. Jedoch wird die Entwicklung dieser Mechanismen, genauso wie die Entwicklung des Internetprotokolls der Version 6, immer weiter gehen. Denn auch bei IPv4 wurden viele Funktionen und heute übliche Vorgehensweisen erst mit den Jahren des Bestehens des

Protokolls entwickelt und verbreitet. Einige haben sich durchgesetzt und sind heute kaum noch wegzudenken, von anderen hingegen spricht keiner mehr.

3.1 Dual-Stack

Der wohl einfachste Weg des Umstiegs ist die Verwendung der Dual-Stack-Technik. Dabei haben die Netzwerkgeräte zur gleichen Zeit einen IPv4- und einen IPv6-Stack. Dies bedeutet auch, dass sie sowohl eine IPv4- als auch eine IPv6-Adresse beanspruchen. Je nachdem, mit welchem Partner das Gerät kommuniziert, wird entweder der IPv4- oder der IPv6-Stack verwendet. Damit dies funktioniert, muss die Infrastruktur zwischen den beiden Geräten das jeweilige Protokoll unterstützen. Die Konfiguration der Stacks erfolgt mit den Techniken, die für das entsprechende Protokoll vorgesehen sind. Demnach muss ein Client für die DNS-Auflösung beide DNS-Record-Typen abfragen können. Liefert der DNS-Server sowohl eine IPv4- als auch eine IPv6-Adresse, muss der Client oder die Anwendung entscheiden können, über welches Protokoll die Verbindung aufgebaut wird.

3.1.1 Vorteile

Die Vorteile dieser einfachen Technik liegen im problemlosen Fortbestehen beider Protokolle. So können auch Anwendungen, die wahrscheinlich nie umgestellt werden, bis zur Ablösung von einer neuen oder komplett aktualisierten Anwendung nutzbar bleiben. Daher ist es bei der Verwendung des Dual-Stack Ansatzes sehr einfach möglich, neue bzw. aktualisierte Programme genauso zu nutzen wie noch nicht portierte Anwendungen. Das bedeutet auch, dass beide Protokolle unabhängig voneinander sind. So gefährdet ein Hinzufügen von IPv6-Stacks die IPv4-Umgebung nicht. Wenn es soweit sein sollte, dass IPv4 nicht mehr verwendet wird, kann der IPv4-Stack ohne Beeinflussung des IPv6-Netzes einfach deaktiviert werden.

3.1.2 Nachteile

Ein Nachteil dieser Technik ist sofort ersichtlich. Da der IPv4-Stack weiterhin besteht und damit auch eine Adresse dafür vorgehalten werden muss, ist ersichtlich, dass dieser Ansatz zum Lösen der Adressknappheit nicht verwendbar ist. Weiterhin ist zu beachten, dass alle Geräte, die diese Technik verwenden, sowohl für IPv4 als auch für IPv6 separat konfiguriert werden müssen. So muss überprüft werden, ob Proxies, Loadbalancer, Security

Policies und Firewall-Lösungen beide Protokolle unterstützen. Auch die Router benötigen für beide Protokolle Routing-Tabellen und die entsprechende Unterstützung.

3.2 Tunneltechnik

Die Verwendung der Tunneltechnik ist vor allem dafür vorgesehen, dass eine gut funktionierende IPv4-Infrastruktur nicht verändert werden soll. Dabei wird eine IPv6-Verbindungsstruktur über das IPv4-Netz gelegt, so dass es keiner Konfiguration des IPv4-Netzes bedarf. Beim Tunneln wird ein Protokoll (hier IPv6) in den Header eines anderen Protokolls (hier IPv4) gepackt und über die Infrastruktur des zweiten Protokolls transportiert. Dieser Prozess besteht aus drei Teilen:

- Einpacken des Originalpakets (Encapsulation),
- Auspacken des Pakets (Decapsulation) und
- Verwaltung des Tunnels.

Anwendung findet diese Technik bei IPv6-Netzwerken, in denen die Verbindung über den ISP nur über IPv4 möglich ist. Das interne Netzwerk kann über einen Tunnel, der über das IPv4-Netzwerk des ISP verläuft, Verbindungen zu anderen entfernten IPv6-Netzwerken aufbauen. Auch die Verbindung von sogenannten IPv6-Inseln in einem IPv4-Unternehmensnetzwerk kann über Tunnel erfolgen. Diese werden meist in zwei Kategorien eingeteilt. Zum einen gibt es die *manuell konfigurierten Tunnel*. Hierbei handelt es sich um Punkt-zu-Punkt Tunnel, die von beiden Seiten manuell eingerichtet werden und bidirektional funktionieren. Zum anderen werden sogenannte *automatische Tunnel* verwendet, für die spezielle IPv6-Adressen zu benutzen sind, beispielsweise ISATAP- oder 6to4-Adressen. Dabei ist in der IPv6-Adresse in einem speziellen Feld eine IPv4-Adresse enthalten. Dadurch ist es möglich, dass der eine Tunnelendpunkt die IPv4-Adresse des anderen Endpunkts der IPv6-Adresse entnehmen kann und dynamisch IPv6-Pakete in einem Tunnel über ein IPv4-Netz übertragen werden können.
Tunnel können zwischen unterschiedlichen Geräten aufgebaut werden. Die folgenden Szenarien sind denkbar:

- Router zu Router,
- Router zu Host,
- Host zu Host und

- Host zu Router.

Die Zuordnung zu einem dieser Szenarien wird anhand der Ermittlung der Adresse des Tunnelausgangspunkts durch den Tunneleingangspunkt vorgenommen. Wenn der Tunnelausgangspunkt ein Router ist, muss dieser das Paket auspacken und an den endgültigen Empfänger weiterleiten. Da in diesem Fall die IPv4-Adresse des Tunnelausgangspunkts nicht dem IPv6-Paket entnommen werden kann, muss diese am Tunneleingangspunkt konfiguriert werden. Bei den anderen beiden Szenarien ist der Tunnelausgangspunkt gleich dem Empfänger des ursprünglichen IPv6-Pakets. Der Tunneleingangspunkt kann in diesem Fall bei der Verwendung einer IPv6-Adresse, die eine IPv4-Adresse beinhaltet, die IPv4-Adresse des Tunnelendpunkts feststellen. Auf diese Weise funktioniert ein automatisches Tunneln.
Die Vorgänge des Ein- und Auspackens eines Pakets laufen wie folgt ab:

Einpacken des Pakets (Encapsulation)
: Beim Einpacken verringert der Eingangspunkt des Tunnels das HopLimit des IPv6-Pakets um eins, verpackt es in einen IPv4-Header und schickt dieses Paket dann über den IPv4-Tunnel. Wenn die MTU des Tunnels überschritten wird, liegt es in der Hand des Eingangspunkts, ob das Paket fragmentiert wird oder aber eine ICMPv6-Nachricht *Packet Too Big* an den Absender verschickt wird.

Auspacken des Pakets (Decapsulation)
: Fragmentierte IPv4-Pakete werden wieder zusammengesetzt. Anschließend wird der IPv4-Header entfernt und das Paket zum endgültigen Empfänger gesendet.

Die Fragmentierung in einem Tunnel sollte möglichst vermieden werden. Dazu definiert der RFC 4213[E. Nordmark, 2005] die folgenden Regeln:

- Bei einer IPv4-Pfad-MTU von weniger als 1280 Byte wird bei Paketen, die größer als diese MTU sind, eine ICMPv6-Nachricht *Packet Too Big* an den Absender geschickt. Bei kleineren Paketen wird das Paket über den Tunnel geschickt. Dabei darf im IPv4-Header das *Dont Fragment-Bit* nicht gesetzt sein, damit die Fragmentierung im Tunnel erlaubt ist.

- Wenn die IPv4-Pfad-MTU größer als 1280 Byte ist und das Paket größer als die Tunnel-MTU zuzüglich 20 Byte für den IPv4-Header, wird eine ICMPv6-Nachricht *Packet Too Big* an den Absender geschickt. Kleinere Pakete werden über den Tunnel versandt, wobei das *Dont Fragment-Bit* im IPv4-Header gesetzt sein muss.

Durch das Versenden einer ICMPv6-Nachricht *Packet Too Big* gibt der Eingangspunkt des Tunnels die Fragmentierung an den Absender des Pakets zurück. Bei einem Paket kleiner als 1280 Byte ist dies nicht möglich, da ein IPv6-Gerät ein Paket nicht unter die minimale IPv6-MTU (1280 Byte) fragmentieren kann. In diesem Fall wird das Paket im Tunnel fragmentiert.

3.2.1 Automatische Tunnel

RFC 4213 [E. Nordmark, 2005] verweist bei der Definition automatischer Tunnel auf die 6to4-Technik. Diese verwendet ein spezielles Präfix, das die IPv4-Adresse des Tunnelendpunkts beinhaltet. Im Folgenden wird auf die 6to4-Technik sowie weitere Möglichkeiten zur Einrichtung automatischer Tunnel eingegangen:

6to4 Die 6to4-Technik erlaubt es, auf IPv6-Ressourcen über ein IPv4-Netz zuzugreifen, ohne dabei explizit einen Tunnel konfigurieren zu müssen. Das vorhandene IPv4-Netzwerk wird hierbei als Punkt-zu-Punkt Link behandelt. Native IPv6-Geräte bauen Verbindungen über 6to4-Router (auch 6to4-Gateway) auf, die die IPv6-Pakete in IPv4-Pakete verpacken. Das Präfix 2002::/16 ist speziell für den 6to4-Mechanismus vorgesehen. Die Adressen werden hierbei im folgenden Format gebildet: 2002:IPv4-Adresse/48. Daraus folgt, dass 16-Bit für die Bildung von 2^{16} Subnetzen mit jeweils 2^{64} Hosts übrig bleiben. Für die Kommunikation innerhalb eines solchen Netzwerks ist kein Tunnel notwendig. Sobald ein 6to4-Knoten mit einem IPv6-Knoten in einem entfernten IPv6-Netz kommunizieren möchte, ist ein Relay-Router notwendig. Dieser ist für 6to4 wie auch für natives IPv6 eingerichtet. Damit native IPv6-Knoten mit den 6to4-Knoten eine Verbindung aufbauen können, verkündet dieser Relay-Router das 6to4-Präfix ins native IPv6-Netz. Die Definition für 6to4 ist im RFC 3056 [B. Carpenter, 2001] zu finden.

ISATAP RFC 5214 [F. Templin, 2008] definiert dieses Intra-Site-Automatic-Tunnel-Adressing-Protocol (ISATAP). Es wurde zur Verbindung von IPv6-Knoten entwickelt, die sich in einem IPv4-Netzwerk befinden, welches keinen IPv6-fähigen Router bereitstellt. Dieses Protokoll betrachtet IPv4 als eine Art OSI-Schicht 2 für IPv6. Aus diesem Grund ist es möglich, in einem Netzwerk IPv6 einzuführen, auch wenn kein IPv6-Router vorhanden ist. Auch private IPv4-Adressen und NAT beeinträchtigen die Funktionen von ISATAP nicht. Die IPv4-Adresse ist bei ISATAP-Adressen im Interface-ID Teil der IPv6-Adresse enthalten. Die ISATAP-Knoten benötigen keine manuelle Konfiguration für ihre Link-lokale-Adresse. Für eine Verbindung außerhalb eines Unternehmens-

netzwerks braucht es mindestens einen IPv6-fähigen Router, damit dieser dem ISATAP-Knoten ein global gültiges Präfix zuweisen kann. Mit diesem Präfix kann der Knoten eine entsprechende ISATAP-Adresse mit globalem Präfix bilden. Für Verbindungen zu nativen IPv6-Geräten im internen Netz kann ein nativer IPv6-Knoten mit einem als ISATAP konfiguriertem Interface als Router zwischen dem IPv6-Netz und dem ISATAP-Netz eingesetzt werden. Die ISATAP-Knoten benötigen dafür allerdings eine Default-Route zum ISATAP-Router. Ein Border-Router ist notwendig, wenn die ISATAP-Knoten mit nativen IPv6-Knoten anderer IPv6-Netzwerke kommunizieren möchten. Dies kann entweder ein ISATAP-Router oder ein 6to4-Router sein. Trotz privater IPv4-Adressen ist es ISATAP-Knoten möglich, mit der ISATAP-Adresse globale Verbindungen aufzubauen, da die Adressen durch das Präfix global eindeutig sind und geroutet werden können.

Teredo Der Teredo-Service ist im RFC 4380 [Huitema, 2006] definiert. Er beschreibt das Verpacken von IPv6-Paketen in UDP damit auch IPv6-Nutzer, die hinter einem oder mehreren NATs sitzen, Zugang zu IPv6-Netzen haben. Anders als die beiden vorher beschriebenen Verfahren (ISATAP und 6to4) benötigt Teredo keine globale IPv4-Adresse. Teredo sollte nur eingesetzt werden, wenn dies unbedingt notwendig ist, da es einen großen Overhead besitzt. Jeder andere Mechanismus sollte daher Vorrang haben.
Der Teredo-Client muss die IPv4-Adresse seines Teredo-Servers kennen. Sobald der Client gestartet wird, schickt er von seiner Link-lokalen-Adresse eine Router-Solicitation an die All-Router-Multicast-Adresse. Dieser wird über UDP an die IPv4-Adresse des Teredo-Servers weitergeleitet. Das Router-Advertisement, welches der Client vom Server erhält, beinhaltet das Teredo-IPv6-Service-Präfix. Daraus bildet der Client seine Teredo-IPv6-Adresse.
Wenn der Teredo-Server Daten, die von Teredo-Clients stammen, weiterleitet, so werden diese IPv6-Pakete in UDP-Pakete gepackt. Die für die Weiterleitung notwendige IPv4-Adresse sowie den dazugehörigen UDP-Port entnimmt der Teredo-Server der IPv6-Adresse. Als Absenderadresse trägt er die eigene IPv4-Adresse sowie den UDP-Port 3544 ein. Seine Aufgabe ist es, Daten von Teredo-Clients über UDP an die richtige Adresse weiterzuleiten und Daten, die von extern für einen Teredo-Client kommen, intern an den richtigen Teredo-Client weiterzuleiten.

Tunnel Broker Die in RFC 3053 definierten Tunnel Broker stellen eine Art Provider für IPv6 dar. Sie geben Nutzern, die eine IPv4-Verbindung zum Internet haben, die Möglichkeit, sich über IPv6 zu verbinden. Dazu registriert sich der Benutzer beim Tunnel Broker. Dieser kümmert sich um die Einstellungen, den Erhalt und die Löschung des Tunnels für den angemeldeten Benutzer. Außerdem kann er die Last auch auf mehrere Tunnel

aufteilen. Zur Erstellung oder Löschung eines Tunnels sendet der Tunnel Broker die dafür notwendigen Informationen zur Konfiguration an den Tunnelserver. Damit der Tunnel Broker diese Aufgaben ausführen kann, muss er mindestens eine IPv6-Adresse besitzen, um Anfragen bearbeiten zu können.
Der Tunnelserver ist ein Dual-Stack Router, der mit dem Internet verbunden ist. Sobald er Informationen zur Konfiguration eines Tunnels vom Tunnel Broker bekommt, kann er die Serverseite des Tunnels einrichten.
Die Client Seite kann ein Dual-Stack Host oder Router sein, der mit dem IPv4-Internet verbunden ist. Damit er eine IPv6-Verbindung zum Internet bekommen kann, muss er sich erst beim Tunnel Broker registrieren und authentifizieren. Nun kann der Tunnel Broker eine Anfrage an den Tunnel Server zur Einrichtung eines Tunnels senden und die Tunnelinformationen für den Client bereitstellen.
Die Verwendung des Tunnel Broker Mechanismus ist vor allem für kleine, abgeschottete IPv6-Netze bzw. IPv6-Hosts geeignet, die eine offizielle IPv4-Adresse besitzen.

3.2.2 Manuell konfigurierte Tunnel

Diese Art der Tunnel sind in RFC 4213 [E. Nordmark, 2005] definiert. Dabei wird ein Tunneleingangspunkt mit der Adresse eines oder mehrerer Tunnelausgangspunkte konfiguriert. Somit entsteht ein Punkt-zu-Punkt Tunnel. Der Aufwand der Konfiguration von manuellen Tunneln rechtfertigt sich in der erhöhten Sicherheit, da festgelegt ist, wohin die IPv6-Pakete weitergeleitet werden. Ein Knoten, der sich in einem abgeschotteten IPv6- oder IPv4-Netz befindet, kann durch Hinzufügen einer Default-Route zu einem IPv6-Router am Tunnelausgangspunkt an die IPv6-Welt angebunden werden.

3.3 Protokollübersetzer

Mithilfe von Protokollübersetzern ist eine Kommunikation zwischen einem IPv6-Knoten, der sich in einem IPv6-Netzwerk befindet und einem IPv4-Knoten in einem IPv4-Netzwerk möglich. Die erste Spezifikation zu diesen Mechanismen befindet sich in RFC 2766
[G. Tsirtsis, 2000], welcher jedoch durch den RFC 4966 [C. Aoun, 2007] neueren Datums in den Status *historic* verschoben wurde. Die Gründe dafür liegen in einer großen Anzahl von nicht lösbaren Sicherheits- und Konformitätsproblemen mit DNSsec dieser Spezifikation.
Der Einsatz von Protokollübersetzern ist in der Praxis nicht komplett vermeidbar, daher gibt es einen neuen Entwurf mit dem Titel *NAT64: Network Address and Protocol Transla-*

tion from IPv6-Clients to IPv4-Servers. Da diese Technik einige grundsätzliche Nachteile beinhaltet, sollte sie auch nur dann Verwendung finden, wenn kein anderer der vorher genannten Mechanismen einsetzbar ist. Aufgrund der Übersetzungsarbeit des Gateways, die eine gewisse Zeit in Anspruch nimmt, stellt dies in jedem Fall einen Flaschenhals dar, der die Leistungsfähigkeit des Netzwerks beeinflusst. Zusätzlich dazu gehen die Vorteile von IPv6 gänzlich verloren.
Eine Verwendung von Protokollübersetzern macht nur für einen beschränkten Zeitraum Sinn, bis eine Anwendung portiert oder erneuert wurde.

3.4 Weitere Migrationsmechanismen

Zusätzlich zu den drei oben genannten Migrationsmechanismen gibt es noch eine Vielzahl anderer Ansätze, den Übergang von IPv4 zu IPv6 zu vollziehen. Einige davon werden in diesem Abschnitt erläutert.

Dual-Stack Lite Ein Knoten, der Dual-Stack Lite implementiert, besitzt eine IPv4- und eine IPv6-Adresse. Dabei wird der Zugang zu IPv4-Diensten über spezielle Tunnel und NAT über das IPv6-Netz realisiert. Mit diesen Mechanismen soll die Möglichkeit geschaffen werden, globale IPv4-Adressen für mehrere Nutzer zur Verfügung zu stellen. Außerdem wird der Transport von IPv4-Daten über ein reines IPv6-Netz ermöglicht.

6rd - IPv6 Rapid Deployment 6rd stellt einen neuen Mechanismus dar, der auf 6to4 aufbaut und die Nachteile der 6to4-Technik für Provider behebt. Ein Provider für 6to4 kann sicherstellen, dass Pakete, die aus einem 6to4-Netz stammen, auch jedes Ziel im IPv6-Internet erreichen. Auch Pakete aus einem externen 6to4 Netzwerk kommen garantiert an. Aber für Pakete, die aus einem externen IPv6-Netz kommen, kann nicht sichergestellt werden, dass diese ankommen, da sie dafür einen 6to4-Relay-Router passieren müssen. Das kann nicht in jedem Fall garantiert werden. Damit dies trotzdem möglich ist, verwendet 6rd nicht das 6to4-Präfix sondern ein offizielles Präfix des Providers. Weiterhin wird die 6to4-Anycast-Adresse durch eine Anycast-Adresse des Providers ersetzt. Außerdem muss der Provider modifizierte 6to4-Router an der Grenze zwischen IPv4-Infrastruktur und IPv6-Internet unterhalten. Diese werden 6rd-Gateways genannt. Auf der Seite des Kunden stehen zusätzlich IPv6-fähige Router, die 6rd unterstützen.

IPv4/IPv6 in VLANs Auch die Verwendung von VLANs stellt eine Möglichkeit der Migration von IPv6 in Netzwerken dar, in denen kein IPv6-fähiges Kern-Netz vorhanden

ist. Da VLANs reine Layer-2-Netzwerke darstellen, können eigene IPv6-VLANs eingerichtet werden, wenn die Struktur anders sein muss als für IPv4. Dies kann beispielsweise der Fall sein, wenn bestimmte Router noch kein IPv6 unterstützen und demnach für IPv6 andere Routingpfade zu benutzen sind. RFC 4554 [Chown, 2006] betrachtet einige der möglichen Szenarien zur Migration mittels VLANs.

GRE (Generic Routing Encapsulation) GRE stellt einen weiteren Mechanismus für das Tunneln dar, der in RFC 2784 [D. Farinacci, 2000] beschrieben ist. Mithilfe von GRE kann jedes andere Protokoll verpackt und getunnelt werden. Im Fall eines IPv6 über IPv4-Tunnels bezeichnet man IPv6 als das Passenger Protocol und IPv4 als das Carrier Protocol. Die Konfiguration eines solchen Tunnels erfolgt nur manuell. Es muss also an beiden Endpunkten des Tunnels die IPv4-Adresse des jeweiligen anderen Tunnelendpunktes konfiguriert werden, so dass der Tunnel nur zwischen zwei Endpunkten existiert. In einem komplexen Netzwerk muss demnach für jede Route ein eigener Tunnel eingerichtet werden.

3.5 Ablauf der Integration von IPv6

Zur Migration des Internetprotokolls von der Version 4 zur Version 6 gibt es eine große Anzahl an Mechanismen, die bereits in diesem Kapitel erwähnt wurden. Jedoch für kein Netzwerk in einem Unternehmen oder einer Organisation kann nur eine dieser Möglichkeiten Verwendung finden. Die beste Lösung wird erzielt, wenn die verschiedenen Mechanismen gut kombiniert werden. Der einfachste Ansatz zur Unterstützung von IPv6 ist die Nutzung von Dual-Stack. Voraussetzung dafür sind genügend IPv4-Adressen sowie die Unterstützung durch die gesamte Hardware. Unterstützt die Hardware noch nicht vollständig IPv6, so können Tunnellösungen bzw. ISATAP die IPv6-Kommunikation über die vorhandene IPv4-Infrastruktur ermöglichen. Auch VLANs können zur Migration genutzt werden. So kann ein IPv6-Router ein IPv6-Präfix in spezielle VLANs verbreiten. Vorhandene Dual-Stack Knoten beziehen eine IPv6-Adresse und kommunizieren in diesem VLAN miteinander. Auch wenn die Backbone-Geräte noch kein IPv6 unterstützen, ist durch den Einsatz der verschiedenen Tunnelmechanismen eine Einführung von IPv6 durchaus möglich.

3.5.1 Reihenfolge der Umstellung

Für die Umstellung von IPv4 zu IPv6 gibt es keinen Stichtag. So kann jedes Unternehmen für sich selbst entscheiden, wie und wann die Umstellung vollzogen wird. Die bisher vorgestellten Mechanismen erschließen die Möglichkeiten der schrittweisen Umstellung. Für diese Schritte kann man drei mögliche Ansätze definieren. Diese werden im Folgenden erläutert.

Core-to-Edge Der Ansatz der Migration vom Backbone zu den Clients hat sich in der Vergangenheit häufig bewährt. Die Umstellung beginnt im Herzen des Netzwerks und breitet sich im weiteren zu den Clients aus. Die Vorteile dieser Methode liegen auf der Hand. So wird der Core-Router aller Wahrscheinlichkeit nach eine gute IPv6-Unterstützung mitbringen, so dass keine oder nur wenige Software Aktualisierungen notwendig sind. Durch die anfängliche Umstellung im Kern des Netzwerks haben die Betreiber Zeit, sich intensiv mit dem Protokoll vertraut zu machen und erste Erfahrungen zu sammeln. Zum Schluss werden dann die Nutzer zugeschaltet und der wichtige Datenverkehr kann auch über IPv6 erfolgen. Die Probleme bei dieser Art der Migration werden aufkommen, je weiter man sich den Clients nähert. Dort sind Probleme bei der Einbindung von Alt-Systemen, der Interoperabilität von IPv4 und IPv6, des Sicherheitsmanagements sowie der Routing-Policies zu erwarten. Durch den Beginn im Kern des Netzwerks hat man für die äußersten Geräte mehr Zeit zur Planung der Migration und kann die dabei bereits gesammelte Erfahrung mit einbringen. Die Unzufriedenheit der Nutzer sowie eventuelle Risiken können verringert werden, da die Nutzer erst in der Endphase IPv6 zur Verfügung gestellt bekommen. Damit ist sichergestellt, dass alle vorherigen Schritte ausgiebig getestet und in Ruhe überprüft wurden.

Edge-to-Core Aufgrund des Mangels an IP-Adressen ist der Ansatz zu wählen, bei dem IPv6 so schnell wie möglich an Kunden und Mitarbeiter ausgerollt werden kann. Wer zu viel Zeit ins Land gehen lässt bevor er mit der Einführung von IPv6 beginnt, wird früher oder später in die Situation kommen, diesen Ansatz wählen zu müssen. Die damit verbundenen Risiken sind sorgfältig zu erörtern. Jedoch verbleibt keine Zeit für die Planung und ausführliche Tests, die zu großen Risiken und schwer abschätzbaren Kosten führen können. Sehr zeitig wird die Kommunikation der Kunden oder Mitarbeiter auf IPv6 umgestellt, so dass wenig Zeit beleibt, um Erfahrungen zu sammeln. Die Gefahr des Ausfalls von kritischen Diensten ist recht hoch, da die Migration noch in vollem Gange ist, während der Nutzer schon IPv6 nutzt. Software-Fehler oder Interoperabilitätsprobleme treten viel weiträumiger in Erscheinung.

Vereinzelte IPv6-Netze IPv6-Inseln werden geschaffen, wenn nur ein kleiner Kreis von Kunden oder nur einzelne Netzwerkbestandteile IPv6 verlangen. Diese können dann über verschiedenste Tunnel miteinander und mit dem Internet kommunizieren. Ein solcher Ansatz wird meist gewählt, wenn die Kosten zum Anfang möglichst gering zu halten sind. Dabei sollte allerdings beachtet werden, dass ein uneinheitliches Netz viel komplexer und aufwendiger zu betreuen ist. So erhöht die Komplexität auch das Risiko eines Ausfalls.

3.6 Sicherheitsbetrachtungen bei der Migration

Die Möglichkeiten ein IPv6-Netzwerk anzugreifen gleichen denen eines IPv4-Netzwerks. Jedoch ermöglichen die mit der neuen Adressierung aufgekommenen Verfahren der Autokonfiguration neue Formen von Angriffen. Die Verwendung von Extension-Headern birgt neue Angriffsziele, die es zu schützen gilt. Hinzu kommt der Fakt, dass IPv4 und IPv6 eine geraume Zeit parallel in den Netzwerken existieren werden, so dass zwei Angriffswege vorhanden sind, die beide entsprechend gesichert werden müssen. Zu guter Letzt sind die Übergangsmechanismen und vor allem die Tunnel zu betrachten, da sie eine nicht zu unterschätzende Gefahr darstellen können.
Selbstverständlich sollte auch beachtet werden, dass IPv6 bei den meisten Betriebssysteme standardmäßig aktiviert ist. So kann es vorkommen, dass die Autokonfigurationsmechanismen von IPv6 still und leise mit anderen IPv6-Geräten kommunizieren, währenddessen nur Regeln und Maßnahmen zum Schutz der IPv4-Infrastruktur eingerichtet werden.

3.6.1 Neighbor Discovery

Neighbor Discovery wird für die Autokonfiguration, das Erkennen von Nachbarn am Link, das Ermitteln von MAC-Adressen von Nachbarn, das Ermitteln von Routern sowie der Aktualisierung der Erreichbarkeitsinformationen verwendet. ND bietet von sich aus die Schutzmaßnahme, dass Pakete ein Hop Limit von 255 besitzen müssen, da sie ansonsten ignoriert werden. So ist gesichert, dass keine Angriffe von entfernten Knoten möglich sind. Um die Sicherheit zu erhöhen, sieht die Spezifikation in RFC 4861 [T. Narten, 2007b] und 4862 [S. Thomson, 2007] IPsec als zusätzliche Sicherheitsmaßnahme vor. Aufgrund der aufwändigen Schlüsselverwaltung ist dies vor allem in öffentlichen und mobilen Netzen ungeeignet. Mögliche Schwachstellen von Neighbor Discovery beschreibt RFC 3756 [P. Nikander, 2004]. Um die Sicherheit bei der Verarbeitung von ND-Nachrichten zu erhöhen, wurden zusätzliche Optionen vereinbart, die es ermög-

lichen, Signaturen zu verwenden, die öffentliche Schlüssel beinhalten. Der sogenannte *Zero-Configuration* Mechanismus dient der Verteilung von Certificate Chains, der Validierung von Knotenadressen sowie der vertrauenswürdigen Zertifizierung von Routern.
Eine Certificate Chain wird genutzt, um die Autorität eines Routers zu überprüfen. Dazu müssen die Knoten und Router einen gemeinsamen *Trust Anchor* besitzen. Nur so kann ein Host einen Router als Default Router eintragen. Zur Feststellung, ob ein Absender einer ND-Nachricht auch der Inhaber der Adresse ist, werden kryptografisch generierte Adressen verwendet. Dazu muss jeder Knoten ein öffentliches und privates Schlüsselpaar generieren, bevor eine Adresse nutzbar wird. In einer neuen ND-Option werden dann der öffentliche Schlüssel sowie notwendige Parameter übertragen.

3.6.2 Router Advertisements

Sobald ein Knoten gestartet wird, sendet er eine Router Solicitation an die All-Routers Multicast-Adresse. Darauf kann jeder Router antworten und Konfigurationsinformationen zum Client senden. Die Gefahr hierbei liegt in einer absichtlich falschen Konfiguration eines Routers, so dass dieser als Default Router für den Client eingetragen wird.

3.6.3 Tunnel

Bei allen Tunneln ist darauf zu achten, dass Pakete, die durch einen Tunnel ein Netzwerk erreichen, Paketfilter umgehen können. So kann ein Angreifer, der sich im Internet befindet, ein IPv4-Paket an einen Tunnelausgangspunkt schicken, in dem ein IPv6-Paket enthalten ist, das wiederum eine IPv6-Adresse aus dem internen Netz als Absenderadresse trägt. Der Tunnelendpunkt wird das Paket nach Erhalt entpacken und weiterleiten. Der Empfänger glaubt daraufhin ein Paket aus dem internen Netz erhalten zu haben. Bei einem solchen Szenario wäre auch das Hop Limit einer ND-Nachricht auf 255 gesetzt und würde nicht als ein eingeschleustes Paket erkannt. Zur Abwehr eines solchen Angriffs wäre es möglich, den Tunnelausgangspunkt so einzurichten, dass dieser nur Pakete eines vorkonfigurierten Tunneleingangspunktes annimmt. Sobald aber ein Angreifer die Adresse des Tunneleingangspunktes herausfindet und fälscht, ist diese Sicherheitsvorkehrung wieder hinfällig. Bei der Verwendung von automatischen Tunneln ist dies noch problematischer, da hier Pakete von beliebigen Quelladressen akzeptiert werden müssen. Daher ist es zu empfehlen, auf einem Tunnelausgangspunkt, also dem Eingangspunkt zum internen Netz, einen zusätzlichen Paketfilter zu betreiben. Im RFC 4891 [R. Graveman, 2007] sind weitere Angriffsmöglichkeiten und Schutzmechanismen über IPsec erläutert. Neue Tunnelmechanismen wie z.B. Teredo, bei dem IPv6-Knoten hinter NATs versorgt wer-

den können, erschweren den Schutz vor Angriffen, da hier die IPv6-Kommunikation über UDP stattfindet.

3.6.4 Sicherheit im lokalen Netzwerk

Das Ziel der Entwickler des Internet Protokolls, eine Ende-zu-Ende Transparenz herzustellen, ist in vielen Netzwerken durch den Einsatz von NAT verlorengegangen. Mit der Einführung von IPv6 kann diese Transparenz wiederhergestellt und die vorhandenen NATs entfernt werden. Die Kosten für die Bereitstellung eines NAT in einem Unternehmen belaufen sich laut einer Studie aus dem Jahr 2006 [1] auf ca. 40 % der Gesamtkosten für den IT-Betrieb. Diese lassen sich beim Verzicht auf NAT und die Einführung von IPv6 einsparen. Aber auch NAT brachte bei seiner Einführung nicht nur Kosten und Probleme, es verbirgt die Netzwerktopologie nach außen und gibt so ein Gefühl der Sicherheit. Doch sobald dieses Hindernis für einen Angreifer überwunden ist, hat er meist leichtes Spiel, da er zum Großteil ungeschützte Systeme vorfindet.
NAT sollte deshalb soweit es möglich ist nicht weiter eingesetzt werden. Zum Schutz vor spezifischen Angriffen müssen Endknoten besser geschützt werden. Dies kann am einfachsten über host-basierte Firewalls erfolgen. Um ein Netzwerk vor allgemeinen Angriffen zu schützen, müssen die zentralen Sicherheitssysteme weiter ausgebaut werden. Detaillierte Informationen zu den Sicherheitsmechanismen von NAT und neuen Sicherheitsmechanismen zum Schutz des IPv6-Netzwerks sind im RFC 4864 [G. Van de Velde, 2007] zu finden.

3.6.5 Firewallregeln

Bei der Verwendung von IPv4 und IPv6 gibt es für jedes Protokoll ein Sicherheitskonzept. Diese beiden Konzepte sind aber nicht vollkommen gleich. Sie müssen immer an die jeweiligen Ansprüche angepasst werden. Entweder gibt es eine Firewall, die beide Protokolle unterstützt oder zwei unterschiedliche Firewallsysteme, wobei die Verwaltung einer gemeinsamen Lösung weniger aufwändig ist. Die folgenden Punkte sind bei der Firewallkonfiguration zu beachten:

- Intern genutzte Adressen sind über einen Filter für eingehenden Datenverkehr zu schützen.
- Nicht benötigte Dienste zwischen den Netzwerksegmenten sind zu filtern.

[1] `www.ntia.doc.gov/ntiahome/ntiageneral/ipv6`

- Alle Hosts sind mit einer host-basierten Firewall auszustatten.
- Wichtige Systeme sollten kein ND benutzen und nicht einfach zu erratende IPv6-Adressen besitzen.
- Endknoten dürfen keinen Routing-Header verarbeiten oder diese Pakete weiterleiten.
- Die Firewalllösung muss in der Lage sein, Filter auf Basis der Quell- und Zieladresse, den Extension-Headern sowie Protokollinformationen höherer Protokolle zu unterstützen.

Die Filterung von ICMPv6 ist ein Thema für sich, da einige ICMPv6 Pakete zwingend für die Funktionalität des Netzwerks notwendig sind. Im RFC 4890 [E. Davies, 2007] sind Empfehlungen für die Konfiguration einer Firewall angegeben, so dass ein sicherer Betrieb weiterhin möglich ist.

3.7 Kosten

Eine nicht zu unterschätzende Größe einer Migration von IPv4 zu IPv6 sind die Kosten. Die wichtigsten Faktoren, die dabei berücksichtigt werden müssen, sind in den folgenden Abschnitten erläutert.

3.7.1 Planung

Die Planung ist einer der Kostenfaktoren, die am meisten ins Gewicht fallen. Neben der Planung des Fortbestehens der aktuellen Infrastruktur muss die Integration von IPv6 sorgfältig bearbeitet werden. Die Planung für einen möglichen Ablauf einer Integration von IPv6 darf nicht nur den Betrieb der bestehenden Struktur mit IPv6, sondern sollte auch die Nutzung neuer Möglichkeiten im Zusammenhang damit berücksichtigen. So können für Probleme, die während des Betriebs mit IPv4 eingetreten sind, eventuell neue Lösungsmöglichkeiten durch den Einsatz von IPv6 geschaffen werden.
Eine umsichtige und frühzeitige Planung der Umstellung mindert die Problemquellen. So ist auch die schrittweise Einführung, die durch die Vielzahl von Übergangsmechanismen unterstützt wird, einer Umstellung von einem Tag auf den anderen vorzuziehen.

3.7.2 Weiterbildung

Die Kosten für die Schulung der Mitarbeiter sind ein weiterer Faktor. Je besser die Ausbildung der Mitarbeiter geplant ist, umso einfacher gestaltet sich die Migration. Der Aufwand für die Schulung zur IPv6-Migration ist nicht viel umfangreicher als der bei einer Weiterbildung der Mitarbeiter zu einer neuen Technologie, die während des Betriebs von IPv4 notwendig waren.
In den Zeiten der Vorbereitung einer Umstellung auf das neue Internetprotokoll sind Testumgebungen notwendig, um die einzusetzenden Systeme auf ihre Tauglichkeit untersuchen zu können. Diese sollten von den Mitarbeitern betreut werden, die mit der späteren Umsetzung im Unternehmen betraut sind. So können Kosten für eine externe Ausbildung gespart werden. Die Mitarbeiter lernen am eigenen Projekt und können Erfahrungen im Umgang mit IPv6 sammeln.

3.7.3 Software

Damit eine Software über IPv6 kommunizieren kann, ist diese entsprechend anzupassen. Je nachdem, wie die Software aufgebaut ist, kann der Aufwand dafür entweder eher gering oder komplex sein. Sobald eine Anwendung klar zwischen Kommunikations- und Anwendungsebene trennt, genügt der Austausch der Kommunikationsebene.
Eine Vielzahl der Anwendungen, die sich am Markt befinden, wird entweder schon jetzt oder in einer der nächsten Versionen die Kommunikation über IPv6 unterstützen. Auch hier kann bei vielen Produkten abgewartet werden, bis der Produktlebenszyklus eine Aktualisierung der Software vorsieht.
Anwendungen, die Eigenentwicklungen darstellen, müssen konkret analysiert werden, um den besten Weg für die Portierung zu ermitteln. In den meisten Fällen ist eine Unterstützung von IPv6 einfach in die Anwendung zu integrieren. Zudem sollte überlegt werden, ob eine komplexere Portierung lohnenswert ist, um die neuen Funktionen, die IPv6 bietet, durch die Anwendung zu unterstützen und ob damit auch der Funktionsumfang erheblich erweitert werden kann. Die Unterstützung der Betriebssysteme erfordert meist nur das Einspielen eines Updates. Auch dies ist nicht immer notwendig, da ein Großteil der Hersteller die Unterstützung für IPv6 in einer neueren Version ihres Produktes von sich aus anbieten. Auch hierbei ist es nicht notwendig, das Betriebssystem vorher auszutauschen, da es auch hier reicht, den normalen Produktlebenszyklus abzuwarten.

3.7.4 Hardware

Die Kosten für die Hardware sind eher gering, da die IPv6-Unterstützung bei der Erneuerung der Hardware nach Ablauf des Produktlebenszyklus erfolgen kann. Es sollte demzufolge neue Hardware nur mit entsprechender IPv6-Unterstützung gekauft werden. Ein vorzeitiger Austausch dieser Hardware ist nicht notwendig. Meist ist der Support für IPv6 auch durch ein Upgrade der Firmware der Hardware realisierbar.

3.8 Unterstützung der Hersteller

Nur unter der Voraussetzung, dass die Hersteller für ihre Betriebssysteme und Router IPv6 unterstützen, ist eine Einführung überhaupt möglich. Zur Zeit können die meisten neuen Produkte der Hersteller mit dem neuen Internetprotokoll umgehen.

3.8.1 Router

Die neuen Modelle aller gängigen Hersteller sind bereit für die Verwendung von IPv6. Jedoch unterscheiden sie sich oft durch den Grad der Unterstützung. Auch bei Geräten eines Herstellers ist die Funktionalität je nach Modell und Typ unterschiedlich stark vorhanden. Die Anforderungen an einen Router sollten daher genau geprüft werden. Mögliche Fragen könnten dabei sein:

- Kann ein Backbone Router das Forwarding von IPv6-Paketen auf Software- oder auf Hardware-Ebene durchführen?
- Werden die geplanten Übergangsmechanismen ausreichend unterstützt?
- Können die vorgesehenen IPv6-Routingprotokolle verwendet werden?
- Wird Paketfilterung, Multicast und VPN untersützt?

3.8.2 Betriebssysteme

Die meisten aktuellen Versionen kommerzieller Betriebssysteme unterstützen IPv6. Aber auch die meisten auf Linux basierenden Open Source Betriebssysteme können vollständig per IPv6 in ein Netzwerk eingebunden werden. Sogar die Hersteller von Software für PDAs oder andere eingebettete Systeme bieten die Verwendung von IPv6 an.

In einigen Fällen ist allerdings der Funktionsumfang von IPv6 begrenzt. Deshalb sollte man, wenn man spezielle Funktionen nutzen möchte, sich vorher informieren, welche RFCs durch den Hersteller implementiert wurden. Auch das Zusammenspiel zwischen einer IPv6-Applikation und dem darunterliegenden Betriebssystem kann unterschiedlich gut gelingen, so dass auch hier im Einzelfall eine gesonderte Überprüfung notwendig ist. Das IPv6-Forum hat zur Kennzeichnung von Produkten ein sogenanntes IPv6-Ready Logo eingeführt. Produkte, die das Logo der Phase eins (silbernes Logo) tragen, bieten eine minimale IPv6-Unterstützung an. Das goldene Logo hingegen signalisiert, dass auch erweiterte IPv6-Funktionen erfolgreich getestet worden sind. [2]

[2] Das IPv6-Forum ist unter `www.ipv6ready.org` zu finden.

Kapitel 4

Migration in einem Testnetzwerk

4.1 Testaufbau und Vorgehensweise

In der Theorie haben alle vorher genannten Integrationsszenarien und Migrationstechniken ihre Daseinsberechtigungen. Zur Überprüfung der vorher genannten Mechanismen ist es von Vorteil, die Theorie in die Praxis umzusetzen. Damit es dabei nicht zu Problemen im Tagesgeschäft des Unternehmens kommen kann, ist es sinnvoll, dies an einem kleinen Netzwerk mit denselben Komponenten, die im Unternehmensnetzwerk verwendet werden, nur in geringeren Dimensionen, zu überprüfen. Der Test soll dabei möglichst viele Szenarien abdecken, die auch später bei der Migration auftreten. Natürlich ist es unmöglich, auf alle eventuell auftretenden Probleme vorbereitet zu sein, aber zumindest ein Großteil lässt sich verhindern.

4.2 Vorüberlegung

Bevor ein solches Testnetzwerk aufgebaut werden kann, ist zu überlegen, welche Punkte genau untersucht werden sollen und wie der Test abläuft. Im Rahmen dieser Masterarbeit sollen natürlich möglichst viele Themen der Migration nachvollzogen werden. Hierzu ist es notwendig, die aktuell im Unternehmen zur Verfügung gestellten Dienste festzustellen und auf ihre Tauglichkeit hinsichtlich der Unterstützung des Dual-Stack sowie der reinen IPv6-Unterstützung zu überprüfen. Zusätzlich zum Test der einzelnen Dienste soll der Durchsatz der Netzwerkkomponenten in der jeweiligen Migrationsstufe gemessen werden, um eventuelle Beeinträchtigungen dahingehend festzustellen. Zur Simulation der einzelnen Dienste kommen fünf PCs zum Einsatz. Auf einem dieser PCs ist ein Windows 2008 R2 Server Betriebssystem installiert, welches die folgenden Dienste bereitstellt:

- DNS- und DHCP-Server,
- Netzwerkfreigabe,
- FTP- und Web-Server und
- Server für den Remote-Desktopzugriff.

Weiterhin gibt es einen Linux-Server, auf dem ein SuSe Linux Enterprise Server Betriebssystem installiert ist. Auf diesem läuft ein SAP- sowie ein SSH-Server zur Überprüfung des SSH-Zugriffs. Die restlichen PCs dienen als Client-PCs, von denen aus der Zugriff auf die unterschiedlichen Dienste getestet wird. Dazu ist zum einen ein Windows 7 und zum anderen ein OpenSuSe installiert.
Bisher enthält der Test nur Geräte, die einen Dienst bereitstellen und Geräte, die auf einen solchen Dienst zugreifen. Das Zwischenstück, die Infrastruktur, soll aber ebenso auf die Migrationsfähigkeit hin untersucht werden. Hierfür stehen drei Router zur Verfügung. Diese werden so miteinander gekoppelt, dass ein OSPF-Netzwerk aufgebaut wird, in dem trotz eines möglichen Ausfalls einer Verbindung weiterhin Daten zwischen den Clients und dem Server austauschbar sind. Des Weiteren gibt es von einem Router eine Verbindung zum Internet, über diese die Firewall getestet wird. Die Abbildung 4.1 zeigt den Aufbau des Netzwerkes.

4.2.1 Phasen des Tests

Der Test wird in drei Phasen ablaufen. Zunächst wird das Netzwerk komplett aufgebaut sowie alle Server installiert und auf ihre korrekte Funktion hin überprüft. Dies geschieht in einem reinen IPv4-Netzwerk, da dabei auf bereits bekanntes Wissen zurückgegriffen werden kann und eigentlich keine Probleme auftreten sollten. Falls es doch zu Problemen kommt, sind diese nicht auf die Migrationsfähigkeit zu beziehen, sondern stammen von einer falschen Konfiguration oder Installation der Komponenten. Erst wenn alle aufgetretenen Probleme der Anfangsphase gelöst sind, ist eine Fortführung des Tests möglich. Ansonsten kann nur schwer festgestellt werden, woher die Probleme stammen.
In der sich daran anschließenden Phase werden alle Systeme auf Dual-Stack konfiguriert und bekommen demnach eine IPv6-Adresse. Alle bereitgestellten Funktionen werden auf ihre Fehlerfreiheit überprüft sowie die zur Umstellung auf Dual-Stack notwendigen Schritte nachvollziehbar dokumentiert. Für den Fall, dass Probleme auftreten, können entsprechende nachvollziehbare Lösungsvorschläge aus Erfahrungen anderer Anwender,

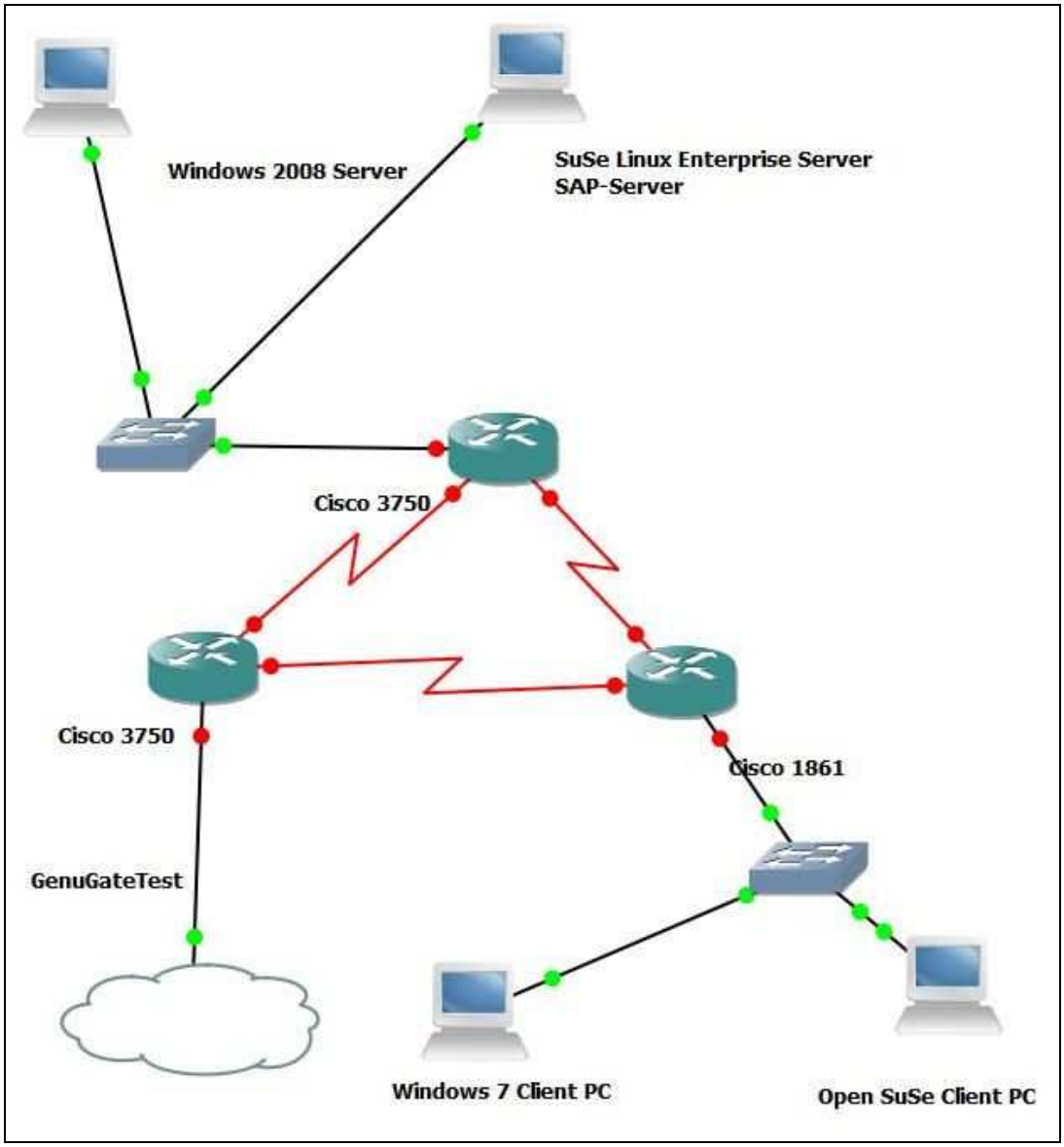


Abbildung 4.1: Aufbau Testnetzwerk

beispielsweise aus dem Internet oder Fachzeitschriften, angewendet und auf ihre Tauglichkeit in der aktuellen Umgebung überprüft werden.
Alle noch vorhandenen IPv4-Funktionen werden in der letzten Umstellungsphase deaktiviert. Im Anschluss daran sind die Funktionen der Server zu überprüfen und eventuelle Fehlerquellen zu analysieren. Sollte es hier zu Fehlern kommen, für die keine Problemlösungen im Internet oder anderer Literatur zu finden sind, ist dies als nicht besonders tragisch anzusehen, da die Verwendung reiner IPv6-Netze in Produktivumgebungen auch in naher Zukunft eher selten vorkommen wird. Der Grund hierfür liegt darin, dass es für die Umstellung von IPv4 zu IPv6 keine Frist gibt, so dass beide Protokolle eine ganze Zeit parallel existieren. Der einzige Grund ein reines IPv6-Netz zu betreiben, wäre ein Mangel an IPv4-Adressen oder die Forderung eines Kunden nach IPv6-Funktionalitäten.

4.3 Erstellung eines Adressierungsplans

Beim Aufbau eines neuen Netzwerks ist zu überlegen, wie man den Komponenten so die Adressen zuweisen kann, dass die jeweiligen Adressen einfach zu merken sind und von diesen auf die Funktion des Gerätes zu schließen ist. Zu Beginn der Planung eines IPv4-Netzwerks wird ein Adressbereich für eine gewisse Unternehmensgröße bzw. eine entsprechende Anzahl an PCs angefordert. Daraufhin wird ein Adressbereich zugeteilt, welcher nach den eigenen Bedürfnissen in Subnetze aufteilbar ist. In den Anfangsjahren von IPv4 wurden zum Teil recht umfangreiche Adressblöcke vergeben, so dass in einigen Unternehmen keine Adressknappheit besteht. Diese haben sich meist ihr Netzwerk nach speziellen Funktionen oder Aufstellungsorten der Geräte aufgeteilt. Jedoch hat ein Großteil der Unternehmen einen zu kleinen Adressbereich bekommen oder ist nach der Anforderung weiter expandiert. So ist es in vielen Firmen der Fall, dass die Adressen für die jeweiligen Subnetze nicht logisch verteilt sind. Der Grund dafür liegt auch darin, dass es zum Zeitpunkt der Netzaufteilung sehr schwer abzuschätzen ist, vor allem wenn wenige Adressen zur Verfügung stehen, für welches Subnetz man wie viele Adressen reserviert. Größtenteils müssen im Nachhinein noch Adressen aus anderen Subnetzen zu einem anderen Teilnetz hinzugenommen werden, wodurch die Struktur und der logische Aufbau sich immer weiter verschlechtert.
Mit der Einführung von IPv6 ergibt sich für viele Unternehmen die Möglichkeit, einen komplett neuen Adressierungsplan zu erarbeiten. Gerade durch die ungeheure Anzahl an Adressen steckt ein großes Potential auch darin, überdurchschnittliche Reserven mit einzuplanen, so dass die Struktur des Netzes auch bei der Erweiterung des Unternehmens nicht verändert werden muss. Neue Techniken von IPv6 erlauben eine einfache Neuadres-

sierung eines gesamten Netzwerks. Dies sollte aber nicht allzu oft durchgeführt werden, da die Änderungen auch in den Köpfen der Administratoren nachvollziehbar sein müssen.

4.3.1 Mögliche Subnetzaufteilungen

Die Aufteilung der Teilnetze erfordert zumindest eine grobe Kenntnis der Aufgabenbereiche und der zur Verfügung gestellten Dienste des Unternehmens. Generell wäre die Partitionierung eines Netzwerks nach Gruppen denkbar, wobei eine Gruppe eine bestimmte Art von Geräten darstellen könnte. Router könnten beispielsweise einer eigenen Gruppen angehören. Auch die Verwendung einer Gruppe für Server und einer weiteren für die Infrastruktur ist denkbar. Ebenso kann eine solche Gruppe auch die Nutzer eines Netzwerksegments repräsentieren. Es können Gruppen für Studenten, Mitarbeiter oder VPN-Benutzer eingerichtet werden. Die bisher im IPv4-Netz verwendeten VLANs können die Grundlage für eine Gruppenaufteilung bilden. Eine gute Übersicht zu den Möglichkeiten bei der Erstellung eines Adressierungsplans ist im Dokument *Preparing an IPv6 Adressing Plan* [Steffann, 2011] der RIPE zu finden.
Für das Unternehmen perdata sind mehrere Möglichkeiten der Subnetzunterteilungen denkbar. Wenn man davon ausgeht, dass ein Unternehmen ein /32 Präfix von der RIPE zugewiesen bekommt und man sich an die von der IANA vorgeschlagene Aufteilung der IPv6-Adresse in 64 Bits Netzanteil und 64 Bits Hostanteil (EUI-64 Format 2.2.5) hält, so bleiben 32 Bits für die Aufteilung in Subnetze. In der hexadezimalen Schreibweise der IPv6-Adresse entspricht dies zwei Vierergruppen, wobei jedes Element einer Vierergruppe 4 Bits repräsentiert, welches Hexadezimal notiert wird.

1. **Vorschlag** Dieser Ansatz sieht vor, die ersten beiden Elemente der ersten Vierergruppe für eine hierarchische Zuteilung in Gruppen zu verwenden. Dabei repräsentiert das erste Element den Typ der Netzwerkkomponenten (Server, Client, Backbone,...). Das folgenden Element steht für den bereitgestellten Dienst (DNS, Web, Router, Firewall, Gateway,...). Die darauf folgenden zwei Vierergruppenelemente bezeichnen den Ort der jeweiligen Netzwerkkomponente (Hauptgebäude, Außenstelle,...). Die übrige Vierergruppe kann für eine weitere Unterteilung der Subnetze verwendet werden.

2. **Vorschlag** Dieses Adressierungsschema sieht im Vergleich zum ersten die Zusammenfassung der beiden Elemente für die Gruppeneinteilung vor, so dass bei diesem Vorschlag 256 Gruppen zur Verteilung zur Verfügung stehen. Der Rest ist identisch zum vorherigen Vorschlag.

Weitere Adressierungsmöglichkeiten können durch die Variation der Anzahl der Elemente der Vierergruppen, die für die Gruppenzuordnung, Ortzuteilung oder Subnetzverteilung zu verwenden sind, erzeugt werden.
Auch andere Ansatzpunkte können zur Erarbeitung eines Adressierungplans hilfreich sein. Für die Sicherung der Zugriffe auf die einzelnen IPv4-Teilnetze sind eine Vielzahl von ACL-Einträgen (Access Control List) nötig. Mit der Erschaffung eines neuen Adressierungsplans könnte die ACL durch eine geschickte Aufteilung des Netzwerks verkleinert werden. Zur Realisierung dieser Vorgabe kommen die Faktoren Funktion, Gruppe, Dienst und Kunde zur Unterteilung in Teilnetze hinzu. Eine Vierergruppe für die Subnetzaufteilung bei perdata könnte die folgenden Elemente enthalten:

- **Funktion** *Infrastruktur, Client,...*
- **Gruppe** *keine, Netz, Server,...*
- **Dienst** *FTP, WWW, Telnet,...*
- **Kunde** *Kunde 1, Kunde 2, ...*

Jedes Element würde in diesem Fall genau 16 verschiedene Werte enthalten. Für eine genaue Angabe der Anzahl der zu verwendenden Elemente einer Vierergruppe ist der voraussichtliche Umfang der Gruppen bzw. Funktionen zu ermitteln.

4.3.2 Vorschlag für einen Adressierungsplan

Aus dem aktuellen für IPv4 verwendeten Adressierungsplan und der Abschätzung der zu erwartenden Erweiterungsbereiche des Netzwerks kann ein Vorschlag für die Adressierung mit IPv6-Adressen vorgelegt werden. Der komplette Adressierungsplan ist im Anhang C zu finden. Für die Subnetzbildung wurde vor allem darauf geachtet, dass für zukünftige Erweiterungen des Netzwerks vorgesorgt ist und die Lesbarkeit erleichtert wird. Außerdem soll es mit diesem Adressierungsschema möglich sein, mit relativ wenigen ACLs den Zugriff von bestimmten Netzbereichen auf bestimmte andere Netzbereiche zu regulieren. Die ersten beiden Elemente der ersten Vierergruppe werden für die Gruppe (Infrastruktur, Extern,...) und das Kundennetz bzw. den Infrastrukturtyp (Server, Netz,...) verwendet. Die dritte größte Gruppe bekommt drei Elemente zugeteilt, also zwei aus der ersten und eins aus der zweiten Vierergruppe. So können für diesen Subnetzbereich 4096 Netze erstellt werden. Wohlgemerkt existiert diese Begrenzung in jedem Subnetz der ersten beiden Elemente der ersten Vierergruppe (Gruppe, Kundennetz/Infrastruktur). Für die verbleibenden zwei Elemente wird die Bezeichnung *weitere Untergliederung* verwendet,

so dass diese für eine zusätzliche Unterteilung in Subnetze verwendbar sind.
Da im Moment relativ wenig Erfahrung bei der Adressierung von IPv6-Netzwerken besteht, kann es durchaus sein, dass der Adressierungsplan sich bei der Umsetzung als ungeeignet erweist und eine weniger strukturierte Unterteilung von Vorteil ist. Für diesen Fall bietet IPv6 die Möglichkeit der einfachen Neuadressierung an. Dieses Verfahren ist in Abschnitt 2.4.3 beschrieben.

Beispiele für Routing-Einträge und ACLs

Um den Aufwand für das Routing und die ACL-Erstellung für den vorgeschlagenen Adressierungsplan abzuschätzen, werden im Folgenden einige Einträge für das Routing und die ACLs vorgestellt. Diese dienen als Beispiel für den notwendigen Konfigurationsaufwand im IPv6-Netzwerk. Dafür wurde das in Abbildung 4.2 gezeigte Netzwerk mit dem Programm GNS3[1] aufgebaut. Hierbei wurde das in Abschnitt 4.3.2 vorgeschlagene

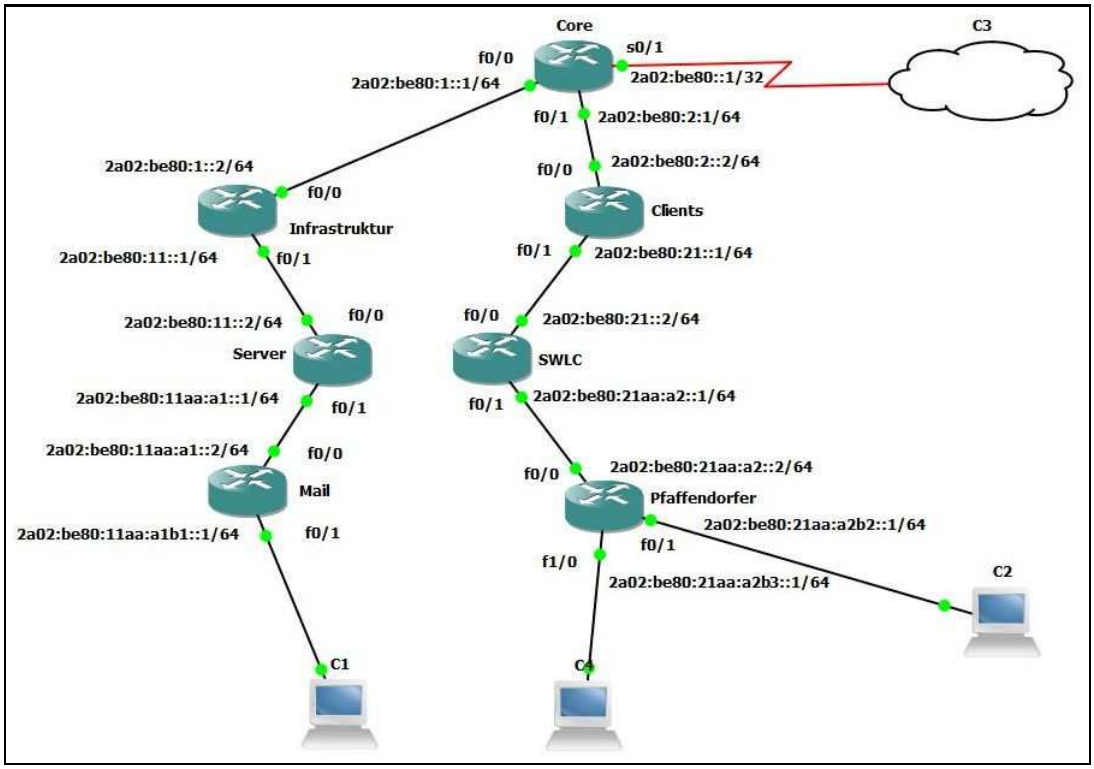


Abbildung 4.2: Testnetz für den Adressierungsplan

Adressierungsschema verwendet. Zur Überprüfung des korrekten Routings und der ACLs dient ein Ping vom Knoten C1 zu C2 bzw. vom C4 zu C2. Dabei sollen die ACLs so gewählt werden, dass ein Ping vom Netzwerk Mail auf C2 möglich ist, jedoch von C4 aus

[1] Dieses grafische Netzwerksimulationsprogramm ist unter `http://www.gns3.net/` zu finden

keine Kommunikation zu C2 stattfinden darf. Der Ping steht dabei stellvertretend für ein beliebiges Netzwerkprotokoll. Der ACL-Eintrag spiegelt in diesem Beispiel das Vorhandensein von unterschiedlichen zu trennenden Systemen unterhalb des Routers Pfaffendorfer wieder, die beide Zugriff auf das Netzwerk unterhalb des Routers Mail benötigen, aber nicht untereinander kommunizieren dürfen.
Das Routing kann wie gehabt auf dem jeweiligen Router durch Aktivieren des Routingprozesses mittels des Kommandos:

```
ipv6 router ospf 1
```

konfiguriert werden. In diesem Beispiel wird der Routingprozess mit der Nummer 1 konfiguriert. Wenn dabei nur IPv6-Adressen Verwendung finden, ist außerdem eine Router-ID zu definieren. Weiterhin muss an jedem Interface, über welches geroutet werden soll, eine IPv6-Adresse konfiguriert sein. Danach erfolgt das Zuordnen zum OSPF-Prozess 1 und zur Area 1 über den Befehl:

```
ipv6 ospf 1 area 1
```

Diese Vorgehensweise ist an allen Interfaces durchzuführen.
Die Konfiguration der ACL erfolgt im simulierten Testnetzwerk am Router Pfaffendorfer. Dieser muss dafür sorgen, dass Daten vom Netz unterhalb des Routers Mail in das Netz zum Knoten C2 durchgelassen werden, Daten vom Netz des Knotens C3 darauf aber keinen Zugriff haben. Damit ergibt sich die folgende ACL:

```
ipv6 access-list permitMail
 permit ipv6 2A02:BE80:11AA:A1B1::/64 2A02:BE80:21AA:A2B2
    ::/64
```

Sie kann dem entsprechenden Interface durch die Eingabe des Befehls

```
ipv6 traffic-filter permitMail out
```

zugewiesen werden.
Ein Test der Einstellungen erfolgte durch einen Ping vom Knoten C1 auf den Knoten C2, der folgendes Ergebnis lieferte:

```
VPCS[1]> ping 2a02:be80:21aa:a2b3::2

2a02:be80:21aa:a2b3::2 icmp6_seq=1 ttl=50 time=204.000 ms
2a02:be80:21aa:a2b3::2 icmp6_seq=2 ttl=50 time=120.000 ms
2a02:be80:21aa:a2b3::2 icmp6_seq=3 ttl=50 time=119.000 ms
```

Dagegen lieferte ein Ping vom Knoten C4 auf den Knoten C2 dieses Ergebnis:

```
VPCS[4]> ping 2a02:be80:21aa:a2b2::2

*2a02:be80:21aa:a2b3::1 icmp6_seq=1 ttl=64 time=26.000 ms (
   ICMP type:1, code:1, Communication with destination
   administratively prohibited)
*2a02:be80:21aa:a2b3::1 icmp6_seq=2 ttl=64 time=11.000 ms (
   ICMP type:1, code:1, Communication with destination
   administratively prohibited)
*2a02:be80:21aa:a2b3::1 icmp6_seq=3 ttl=64 time=8.000 ms (
   ICMP type:1, code:1, Communication with destination
   administratively prohibited)
```

Damit belegt der Test die erforderliche Funktionalität. Während der Konfiguration des Netzwerks im Simulator und für Verwendung dieses Adressierungsplans im Unternehmensnetzwerk ergeben sich damit zwei Aspekte, die es bei der Umsetzung zu beachten gibt. Zum einen sollten für das Routing mehrere Areas eingerichtet werden, damit die Routingtabellen eine überschaubare Größe behalten. Außerdem können Routingtabellen durch Summary Bildung (Abschnitt 2.5.1) oder auch die Einrichtung von Stub-Areas verkleinert werden. Die Funktionalität von OSPF sollte dabei möglichst optimal ausgenutzt werden. Zum anderen werden bei dem vorgeschlagenen Adressierungsplan und den beispielhaft gesetzten ACLs relativ viele Daten tief in die Netzwerkstruktur hineingelassen, bevor sie blockiert werden. Daher ist genau abzuwägen, auf welcher Hierachieebene eine ACL angelegt wird, damit sie ihrer Funktion gerecht werden kann und dabei nicht ungewollt andere Zugriffe blockiert.
Im Vergleich zu IPv4 werden durch den Einsatz des vorgeschlagenen Adressierungsplans die Routingtabellen größer, wenn keine Zusammenfassung der Routen und keine Konfiguration weiterer Areas stattfindet. Das Verwalten und Erstellen von ACLs wird im Vergleich zu IPv4 mit der Verwendung des vorgeschlagenen Adressierungsplans übersichtlicher, da die Funktionalität eines Netzes anhand der Adresse ersichtlich ist. Die Anzahl der ACL-Einträge wird durch die sinnvollere Aufteilung der Netze geringer werden. Jedoch muss genau überlegt werden, an welcher Stelle im Netzwerk eine ACL zu platzieren ist, um die Schutzwirkung maximal zu halten.

4.4 Durchführung

Die Durchführung der Migration im Testnetzwerk erfolgt mit dem Durchlaufen der in Abschnitt 4.2.1 genannten Phasen. Zu jeder Phase wird kurz die Konfiguration erläutert. Im Anschluss daran wird die Funktionalität überprüft.

4.4.1 Test der IPv4-Funktionalität

Die erste Phase des Tests stellt die Ausgangssituation im Unternehmen dar. Hier werden alle in Abschnitt 4.2 genannten Funktionalitäten konfiguriert und überprüft.

Einrichten der Komponenten Die Konfiguration des Windows 2008 Servers erfolgt hauptsächlich über die jeweiligen grafischen Oberflächen. Gleich nach der Installation des Betriebssystem wurde eine statische IPv4-Adresse vergeben, worüber der Server ab diesem Zeitpunkt erreichbar war. Danach erfolgte die Einrichtung der Dienste.
Als erstes wurde ein neuer Ordner angelegt und dieser für einen lokalen Benutzer über den Reiter *Freigabe* in den Eigenschaften des Ordners freigegeben.
Im Anschluss daran wurde über den Unterpunkt *Remoteeinstellungen* der Eigenschaften des Menüpunkts *Computer* der Fernzugriff auf den Server zugelassen.
Die Installation des DNS-Servers fand über den *Servermanager* und die *Rollenverwaltung* durch das Hinzufügen der Rolle *DNS-Server* statt. Nach der erfolgreichen Installation wurden zwei DNS-Einträge (für Windows 2008 Server und für den Windows 7 Client) jeweils für IPv4 und IPv6 vorgenommen.
Weiterhin wurde auf dem Windows 2008 Server noch die Rolle *Webserver (IIS)* installiert. Diese beinhaltet einen Webserver mit einer Testseite und einen FTP-Server.
Über die Befehle

```
/etc/init.d/sshd start
startsap
```

konnten auf dem SuSe Linux Enterprise Server der ssh- und der SAP-Server gestartet werden. Da nun alle Dienste aktiviert sind, kann die Konfiguration der Router bzw. der Test der bereitgestellten Dienste beginnen.

Routerkonfiguration Die zur Verfügung stehenden Router wurden mit den entsprechend notwendigen IP-Adressen konfiguriert und physikalisch verbunden. Für das Routing im Testnetz wird das im Unternehmen eingesetzte Protokoll OSPF verwendet. Mit der Konfiguration von OSPF beschäftigt sich der Abschnitt 4.6.

Überprüfen der Funktionalität Die Prüfung der Funktionalität der eingerichteten Dienste wurde jeweils mit einer Abfrage vom Windows 7 Client-PC und vom OpenSuSe Client-PC durchgeführt. Der Test des ssh-Servers fand mittels des Programms *Putty*, einem kleinen Telnet- und SSH-Client [2], statt.
Für den Test des SAP-Servers konnte der SAPgui-Client Verwendung finden. Beim Test stellte sich heraus, dass alle Dienste reibungslos und ohne Probleme funktionierten. Um festzustellen, ob durch die Verwendung beider Internetprotokolle im Dual-Stack bzw. in der dritten Phase des reinen IPv6-Stacks, Durchsatz- bzw. Geschwindigkeitseinbußen im Netzwerk zu erwarten sind, wird mittels des Programms *JPerf* [3] eine Performancemessung durchgeführt. Um Sicherzustellen, dass nur IPv4 verwendet wird und kein IPv6-Verkehr das Netzwerk stört, musste IPv6 auf allen beteiligten Rechnern und Routern deaktiviert werden. Die einzelnen Werte dazu sind im Anhang A.1 zu finden.

4.4.2 Dual-Stack Testbetrieb

Nachdem durch die Konfiguration der IPv4-Testumgebung ein kleines Abbild des aktuellen Unternehmensnetzwerks aufgebaut ist, kann nun die beispielhafte Migration starten. Dies beginnt mit der Anpassung der Konfiguration der Geräte.

Anpassung der Konfiguration für den Dual-Stack Betrieb Zur Aktivierung des Dual-Stack wurde auf dem Windows 2008 Server sowie auf den Clients-PCs das Internetprotokoll in der Version 6 wieder aktiviert und jedem Gerät eine IPv6-Adresse zugewiesen. Zur Namensauflösung sind zwei DNS Einträge für die IPv6-Adressen vorzunehmen. Damit die jeweiligen Serverdienste auf Anfragen über die IPv6-Adressen antworten, sind sie an die IPv6-Adresse zu binden. Dazu ist meist ein Neustart der zur Verfügung gestellten Dienste notwendig.
Für die Umstellung des SAP-Servers ist zusätzlich zur Konfiguration des Betriebssystems eine Änderung in den SAP-Umgebungsvariablen von Nöten. Dazu sind in der Datei *.sapenv_<Rechnername>.csh (hier: .sapenv_sles11sap.csh)* und in der Datei *.sapenv_<Rechnername>.sh* des Nutzers, der für den Start der SAP-Komponenten zuständig ist, die folgenden Zeilen hinzuzufügen:

```
SAP_IPv6_ACTIVE = 1
export SAP_IPv6_ACTIVE
```

für die Datei *.sh* und

[2] Putty-Client `http://www.chiark.greenend.org.uk/~sgtatham/putty/`
[3] JPerf-grafische Oberfläche zu IPerf `http://sourceforge.net/projects/iperf/`

```
setenv SAP_IPv6_ACTIVE 1
```

für die Datei *.csh*.
Dieselbe Umgebungsvariable ist auf dem Client mit der installierten SAPgui zu setzen. Im Testnetzwerk betraf dies den Windows 7 Client. Im Anschluss daran sind der SAP-Server und die SAPgui neu zu starten. Diese Vorgehensweise ist im SAP-Hinweis 1346768 beschrieben. Um sicherzugehen, dass die Variablen auf dem Server richtig gesetzt sind, bietet es sich an, mit dem Programm *nmap*, einem auf vielen Linux-Systemen vorhandenem Portscanner, den Start des Servers für die IPv4- und die IPv6-Adresse zu überprüfen. Dazu sind die folgenden zwei Konsolenaufrufe notwendig:

```
nmap 192.158.0.3
nmap -6 2a02:238:1:f00b:1001::3
```

Wenn der SAP-Server richtig gestartet ist, müssen dabei für IPv4 und IPv6 dieselben SAP-Ports geöffnet sein.

Routerkonfiguration Auch auf den Routern wurde IPv6 aktiviert und die zugehörigen Interfaces bekamen eine IPv6-Adresse. Das Routing für OSPF in IPv6 ist extra einzurichten. Auf die Routerkonfiguration wird in Abschnitt 4.6.3 eingegangen.

Test der Funktionalitäten für Dual-Stack Auch für diesen Test sind alle angebotenen Dienste des Servers mit einem Zugriff von den Client-PCs aus getestet worden. Der Zugriff auf die Windowsfreigabe vom Windows 7 Client gelang über den Windowsexplorer nur über die Eingabe des Namens des Servers. Ein Zugriff direkt über die IPv6-Adresse war nicht möglich. Auch das Umschließen der Adresse mit eckigen Klammern und die Angabe der Portnummer brachten keine Verbesserung. Der Zugriff vom OpenSuSe-Client auf die Freigabe verlief mit dem *smbclient* ohne Probleme durch Ausführen der Kommandozeile:

```
smbclient -L w2008r2.local -U Administrator
```

Über den Befehl *smbmount* könnte eine solche Freigabe ins Linux Dateisystem eingebunden werden.
Die anderen Funktionen lassen sich wie bei IPv4 gewohnt benutzen. Dabei ist der Zugriff über IPv6-Adresse genauso wie über den DNS-Namen möglich. Auch hier wurde der ssh-Server über das Windows-Programm *Putty* getestet. Die Funktionalität des SAP-Servers erfolgte durch eine Anmeldung über die SAPgui am SAP-Server. Zur Überprüfung des

[3] SAP-Hinweis 1346768 - Aktivierung von IP Version 6 (IPv6) im AS ABAP

Netzwerkdurchsatzes wurde eine Messung mit dem Programm *JPerf* durchgeführt. (siehe Abschnitt A.2)

Erstellung von Tunneln (IPv6 über IPv4) Zur Migration von IPv4 zu IPv6 zählen auch Übergangsmechanismen, die notwendig sind, falls es eine vorhandene Infrastruktur gibt, die kein IPv6 unterstützt. Um ein solches Netz zu überbrücken, kann ein Tunnel aufgebaut werden, der zwei IPv6-Netzwerke über ein IPv4-Netz verbindet. Ein Tunnel kann automatisch über ISATAP oder Teredo aufgebaut werden. Wie diese funktionieren ist in Abschnitt 3.2.1 beschrieben. Zur Überbrückung von IPv4-Netzwerken in einem Unternehmen sollten jedoch automatische Tunnel deaktiviert und nur manuelle Tunnel verwendet werden. Da ein Tunnel auch über Sicherheitsvorkehrungen hinweg, also auch durch Paketfilter, nicht eingeschränkt wird, sind Tunnel nur manuell aufzubauen, da die Kontrolle am Tunnelend- bzw. Tunnelanfangspunkt stattfinden kann.
Tunnel sind keine neue Erfindung für die Einführung von IPv6. In Form von VPN-Tunneln sind sie uns bestens bekannt. Neu hinzugekommen ist die Form, dass IPv6 über IPv4 transportiert wird. Die bei perdata verwendeten Router der 3700er Serie mit
IOS12.2(55)SE3 unterstützen diesen Tunneltyp leider nicht. Cisco bietet für diesen Fall sogenannten GRE-Tunnel (Generic Routing Encapsulation) an, die eine Vielzahl von Protokollen über IPv4 transportieren. Die Konfiguration eines GRE-Tunnels wird im Abschnitt 4.7 genauer erläutert.

Prüfung der Funktionalität des Tunnels Um den Tunnel entsprechend testen zu können, wird ein Router für den IPv4-only Betrieb konfiguriert. Er stellt in diesem Fall den Teil des Netzwerks dar, der über keinerlei IPv6-Funktionalität verfügt. Weiterhin werden die OSPF-Kosteneinstellungen so modifiziert, dass der gesamte IPv6-Verkehr nur noch über den GRE-Tunnel stattfindet bzw. nur bei Ausfall des Tunnels über die IPv6-Infrastruktur erfolgt. Daran ist erkennbar, dass es sich um ein gestelltes Beispiel handelt, da im laufenden Betrieb immer die native IPv6-Lösung zu bevorzugen ist.
Der Test der Dienste über den erstellten Tunnel ergibt keine Veränderungen zum Test beim Dual-Stack Betrieb. Trotzdem sollten bei der Konfiguration eines Tunnels die Hinweise aus dem letzten Teil des Abschnitts 3.2.2 zur Fragmentierung im Tunnel beachtet werden, da es sonst zu schwer nachvollziehbaren Fehlern kommen kann [4].
Die Leistungsfähigkeit des GRE-Tunnels wurde gemessen, um festzustellen, in wie weit ein solcher Tunnel als Backup für eine native IPv6-Route einsetzbar ist. Zu Bedenken ist dabei, dass jedes am Tunneleingang ankommende Paket erst in ein IPv4-Paket eingepackt

[4] Probleme bei GRE Tunneln - `http://www.cisco.com/en/US/tech/tk827/tk369/technologies_tech_note09186a0080093f1f.shtml`

und am Tunnelendpunkt wieder entpackt werden muss. Dieser Vorgang wird in Software realisiert und hängt von der Leistungsfähigkeit des Routers ab. Das Testprotokoll dazu ist im Anhang A.3 zu finden.

4.4.3 IPv6-only Test

Die letzte Phase des Tests soll die Situation nach der Umstellung auf das Internetprotokoll in der Version 6 darstellen. Auch wenn eine komplette Umstellung in weiter Ferne zu liegen scheint, hat diese Phase ihre Daseinsberechtigung. Ein IPv6-only Netzwerk kann schneller als erwartet von Nöten sein, beispielsweise wenn die IPv4-Adressen im Unternehmen aufgebraucht sind und ein weiterer Bedarf an IP-Adressen für neue Geräte besteht oder Kunden an neuen IPv6-Funktionen interessiert sind. Andernfalls wird in der Übergangsphase ein Dual-Stack Netz dominieren.

Umkonfigurieren der eingesetzten Geräte Die als Dual-Stack konfigurierten Geräte bekommen nun keine IPv4-Adressen mehr. Außerdem wird, sofern dies möglich ist, das Internetprotokoll der Version 4 deaktiviert. Leider ist dies nicht bei allen Geräten möglich. Beispielsweise benötigt der SAP-Server für die Kommunikation mit der Datenbank eine IPv4-Verbindung und damit den IPv4-Stack, auch wenn sich Datenbank und SAP-Server auf dem selben physischen Geräte befinden.
Für die OSPF-Konfiguration wird lediglich der Routingprozess für das IPv4-Protokoll gestoppt.

Test des IPv6-only Netzwerks Beim Test im reinen IPv6-Netzwerk ergibt sich für den Zugriff auf die bereitgestellten Dienste keinerlei anderes Verhalten als für den Test im Dual-Stack Betrieb. Auch der Zugriff auf den SAP-Server über die SAPgui gelingt problemlos. Für die Verwendung der Dateifreigabe auf dem Windows 2008 Server vom Windows 7 Client aus ist wiederum hier nur die Verwendung des DNS-Namens möglich. Der Verbindungsversuch mit der IPv6-Adresse schlägt fehl. Alle anderen Dienste sind vom Windows 7 Client wie auch vom OpenSuSe Client problemlos nutzbar.
Abschließend wird wiederum ein Performance-Test mit dem Programm *JPerf* durchgeführt. Zu Bedenken gibt es in diesem Fall zum einen, dass sich nicht auf allen Geräten der IPv4-Stack abschalten ließ und zum anderen, dass kaum eine der getesteten Anwendungen zumindest nach Angaben des jeweiligen Herstellers alle Möglichkeiten des QoS bei IPv6 ausnutzt. Das Messprotokoll ist im Anhang A.4 einsehbar.

4.4.4 Auswertung der Performance Messungen

Die in jeder Testphase sowie nach dem Test des Tunnels ausgeführten Leistungsmessungen des Netzwerks beziehen sich auf die Verwendung von zwei Cisco 3750 Routern, einem Cisco 1860 Router sowie HP-Desktop PCs. Für das Routing kommt das OSPF-Protokoll in der zum jeweiligen Protokoll gehörenden Version zum Einsatz. Die Netzwerkkomponenten waren für die Dauer der Messungen keinen weiteren Aufgaben zugeordnet. Wie aus den Messprotokollen im Anhang A ersichtlich ist, ergibt sich keine Verbesserung oder Verschlechterung bei der Verwendung von nur einem oder beiden Protokollen. Lediglich die Verwendung des Tunnels ist mit Leistungseinbußen verbunden. Dies ist darin begründet, dass jedes Paket vor dem Versand durch den Tunnel eingepackt (Encapsulation) und am Tunnelausgangspunkt wieder entpackt (Decapsulation) werden muss. Dieser Prozess ist entsprechend aufwendig und benötigt Zeit. Das führt dazu, dass der Tunnel mit einer Geschwindigkeit von ca. 8 MBit/s um einiges langsamer als die sonst gemessene Geschwindigkeit von 70 MBit/s ist.

4.5 Erstellung von IPv6-ACLs für Cisco Router

ACLs (Access Control List) dienen dem Schutz eines Netzes vor unerwünschtem Zugriff. Bestimmte Nachrichtentypen lassen sich durch den Einsatz von ACLs blockieren bzw. zulassen. Außerdem können ACLs angelegt werden, um Routing-Updates nur von bestimmten Routern zuzulassen, so dass keine Manipulation des Routings durch Unberechtigte möglich ist. Im Großen und Ganzen haben IPv6-ACLs den selben Funktionsumfang wie IPv4-ACLs. Trotzdem gibt es ein paar kleine Änderungen:

- Für IPv6 sind nur noch benannte ACLs möglich,
- IPv6-ACLs unterstützen Erweiterungsheader und Protokolle höherer Ebenen,
- IPv6-ACLs besitzen implizite Regeln, die jeglichen Neighbor Discovery Verkehr erlauben,
- Unterstützung zeitbasierter ACL-Regeln und
- Unterstützung reflexiver ACLs.

Weiterhin ergeben sich für IPv6-ACLs zusätzliche Paketeigenschaften, die durch eine ACL überprüft werden können.

- TCP/SCTP/UDP - Pakete sowie beliebige Ports können überprüft werden.
- Code- und Type-Felder im ICMPv6-Paket sind überprüfbar.
- Fragmente und Routing-Header können auf unterschiedlichste Parameter hin untersucht werden.

Time-Range ACLs Mithilfe von zeitbasierten ACLs können bestimmte Regeln nur zu bestimmten Tageszeiten aktiviert oder deaktiviert werden. So erlaubt die folgende Konfiguration keinen www-Traffic (Port 80) in der Zeit von 10 - 13 Uhr:

```
time-range zeit1
periodic daily 10:00 to 13:00
!
ipv6 access-list zeitACL
deny tcp any any eq www time-range zeit1
permit ipv6 any any
```

Listing 4.1: Time-Range ACL

Reflexive-ACLs Dieser ACL-Typ wird automatisch erstellt, sobald ein Paket auf eine permit-Regel trifft, die mit dem Schlüsselwort *reflect* gekennzeichnet ist. Die reflexive ACL erlaubt den Zugriff durch die Generierung weiterer permit-Einträge solange es eine weitere gleichwertige Kommunikation gibt oder ein *FIN* in einem Paket erkannt wird. Zum Schutz bei eventuellen Verbindungsabbrüchen ist die Angabe eines Timeouts möglich, nach dessen Ablauf die dynamisch erzeugten Regeln entfernt werden, wenn kein weiterer passender Traffic vorhanden ist. Unterstützte Protokolltypen sind ICMPv6, TCP, UDP, SCTP.
Reflexive-ACLs können dazu genutzt werden, generell ankommende Pakete von außen zu blockieren. Sobald aber ein Host vom Inneren des Netzwerks eine Nachricht bestimmten Typs (die zu einem reflexiven permit passt) nach außen versendet, wird dynamisch für eine bestimmte Zeit oder bis zum Eintreffen eines *FIN* eine ACL angelegt, die Antworten auf die gesendete Nachricht erlauben. Es wird also temporär aufgrund einer Anforderung aus dem Netzwerk eine ACL definiert, die Antworten auf diese Anfrage zulässt.

4.5.1 Beispiel IPv6-ACLs im Testnetzwerk

Der Test des Tunnels wurde im oben genannten Beispiel (Abschnitt 4.4.2) durch die Umleitung des IPv6-Verkehrs aufgrund der OSPF-Kosten durchgeführt. Eine andere Lösung,

ohne die OSPF-Kosten zu verändern, wäre das Anlegen einer IPv6-ACL, die jeglichen IPv6-Verkehr über ein Interface verbietet. Die dazu zu verwendende ACL sieht wie folgt aus:

```
ipv6 access-list DenyIPv6
 deny ipv6 any any log
```

Listing 4.2: IPv6-ACL einem konkreten Interface zuweisen

Diese ACL mit dem Namen *DenyIPv6* verbietet IPv6 von jeder Quelladresse zu jeder Zieladresse. Außerdem wird immer, wenn diese ACL Anwendung findet, eine Meldung ausgegeben (Schlüsselwort *log*). Damit die ACL ihre Funktion erfüllt, muss sie dem Interface zugewiesen werden, an dem sie die Pakete filtern soll. Dies geschieht wie folgt:

```
interface FastEthernet1/0/2
 ipv6 traffic-filter DenyIPv6 in
 ipv6 traffic-filter DenyIPv6 out
```

Listing 4.3: IPv6-ACL, die den kompletten IPv6-Verkehr unterbindet

In diesem Beispiel wird die Access List sowohl für eingehende als auch für ausgehende Kommunikation angewendet.
Alle angelegten und aktiven ACLs können mit dem Kommando

```
show ipv6 access-list
```

angezeigt werden. Zudem sind in der darauffolgenden Ausgabe die aktuellen Treffer zu sehen. Sie zeigen an, wie oft ein Paket durch eine ACL geblockt bzw. durchgelassen wurde. Dieser Wert kann mit dem Befehl

```
clear ipv6 access-list
```

zurückgesetzt werden.

4.5.2 Einschränkungen von IPv6-ACLs beim Cisco 3750

Der Cisco Router 3750 kann laut der Cisco Dokumentation zum aktuellen IOS 12.2(55) nicht mit allen Optionen umgehen. So sind die folgenden Einschränkungen laut der Dokumentation[5] zu beachten:

[5] IOS 12.2(55) Dokumentation `http://www.cisco.com/en/US/docs/switches/lan/catalyst3750/software/release/12.2_55_se/configuration/guide/swv6acl.html`

- Es sind nur Präfixe von /0 bis /64 und Hosts (/128) mit Link-lokalen oder globalen Unicast-Adressen verwendbar.
- Flow-Label und Routing-Header werden in den ACL-Regeln nicht unterstützt.
- MAC-basierte, reflexive, output-port und VLAN-ACLs finden keine Verwendung.

Ergänzend zu den Einschränkungen ist zu erwähnen, dass sich ACLs mit diesen Einschränkungen auf dem Router erstellen lassen, aber danach nicht an ein Interface zugewiesen werden können.
Die Erstellung und Zuweisung von ACLs mit nicht unterstützten Funktionen war im Testnetzwerk trotzdem möglich. Es gab keine Fehlermeldungen beim Zuweisen der ACLs. Auch die Funktionalität der Regeln war gegeben. Dennoch ist von der Verwendung dieser Optionen abzuraten, da für Fehler, die durch die Verwendung dieser Optionen auftreten, der Hersteller keine Unterstützung oder Hilfestellung anbietet.
Durch die Verwendung von IPv6 und IPv4 wird auch der zur Verfügung stehende Speicherplatz für die Ausführung von ACLs unter den beiden Protokollen aufgeteilt und die Abarbeitung der ACLs kann möglicherweise nicht mehr in Hardware ausgeführt werden, was einen erheblich größeren Zeitaufwand bedeutet. Eine genaue Aufschlüsselung der Verwendung des TCAM (Ternary Content Addressable Memory) beim Cisco 3750 sowie eine mögliche Optimierung dessen sind in der Cisco Technote 44921 [6] zu finden.

4.6 OSPF-Konfiguration im Testnetzwerk

Eine OSPF-Konfiguration kann im einfachsten Fall aus einer sogenannten Area bestehen. Dadurch wird ein direkter Weg zu einem Ziel möglich, aber auch die Routingtabellen wachsen sehr schnell an, da jeder Router eine Vielzahl an Netzen kennt, zu denen er die Pakete weiterleiten kann. Eine Aufteilung in mehrere Areas und der Einsatz von Stub- und NSS-Areas sollte, je größer das Netzwerk ist, bei der Konfiguration des Routings berücksichtigt werden. Für das Testnetzwerk hingegen wäre es mit Kanonen auf Spatzen geschossen, wenn eine Vielzahl Areas und Stub-Areas eingerichtet würden. Zudem würde es den Test durch zusätzlich notwendige Konfiguration erschweren.
Die Routerkonfiguration im Test umfasst die Backbone-Area null und eine Area eins, welche die angeschlossenen Geräte, also die Clients und Server, beinhaltet. Die jeweiligen Adressen und Netze sind erst einmal zufällig ausgewählt und können je nach Bedarf oder

[6] Cisco Technote 44921 `http://www.cisco.com/en/US/products/hw/switches/ps5023/products_tech_note09186a00801e7bb9.shtml`

für eventuelle Tests in Richtung Routing und ACLs entsprechend verändert werden. Die Komplette Konfiguration der Router ist im Anhang B nachzulesen.

4.6.1 Aktivierung von IPv6 auf Cisco Routern

Standardmäßig ist die IPv6-Unterstützung bei Cisco Routern deaktiviert. Zur Aktivierung von IPv6 muss ein sogenanntes Switch Database Management (SDM) Template installiert werden. Dieses Template beinhaltet die Konfiguration der Systemressourcen mit gesonderter Gewichtung für spezielle Aufgaben des Routers. So kann bei der Installation des Templates ausgewählt werden, ob der Router für *Routing*, *VLANs* oder *default* zu konfigurieren ist. Das Schlüsselwort *default* bezeichnet hierbei eine ausgewogene Verteilung der Systemressourcen des Routers für IPv4/IPv6 Layer 2 und Layer 3 Funktionalität. Der folgende Befehl aktiviert das zuletzt genannte Template:

```
sdm prefer dual-ipv4-and-ipv6 default
```

Die genaue Verteilung der Systemressourcen für den jeweiligen Router und die IOS Version können in der Cisco Dokumentation nachgelesen werden. Damit die durch das Template neu gesetzten Einstellungen übernommen werden, ist der Router neu zu starten.

4.6.2 OSPF-Routing im IPv4-Netz

Um überhaupt mit der Einrichtung des Routings beginnen zu können, müssen Interfaces mit IP-Adressen und den zugehörigen Subnetzadressen versehen werden. Dies geschieht wie folgt:

```
interface FastEthernet1/0/1
 no switchport
 ip address 192.178.0.1 255.255.255.0

interface FastEthernet1/0/2
 no switchport
 ip address 192.168.0.2 255.255.255.0
```

Listing 4.4: Cisco 3750 - 1 IPv4-Konfiguration

```
interface FastEthernet1/0/1
 no switchport
 ip address 192.158.0.1 255.255.255.0
```

```
interface FastEthernet1/0/2
 no switchport
 ip address 192.188.0.2 255.255.255.0

interface FastEthernet1/0/24
 no switchport
 ip address 192.168.0.1 255.255.255.0
```

Listing 4.5: Cisco 3750 - 2 IPv4-Konfiguration

```
interface Vlan10
 ip address 192.198.0.1 255.255.255.0

interface Vlan20
 ip address 192.188.0.1 255.255.255.0

interface Vlan30
 ip address 192.178.0.2 255.255.255.0

interface FastEthernet0/1/0
 switchport access vlan 20

interface FastEthernet0/1/1
 switchport access vlan 30

interface FastEthernet0/1/5
 switchport access vlan 10

interface FastEthernet0/1/6
 switchport access vlan 10

interface FastEthernet0/1/7
 switchport access vlan 10
```

Listing 4.6: Cisco 1861 IPv4-Konfiguration

Die beiden Cisco 3750 Router besitzen standardmäßig geswitchte Ports. Mit dem ersten Kommando *no switchport* wird diese Funktion deaktiviert. Mithilfe des zweiten Kommandos im Interface Modus wird die IP-Adresse und die Subnetzmaske konfiguriert. Beim Cisco 1861 kann das Switching der Ports nicht deaktiviert werden, so dass dafür VLANs

angelegt und diese dem jeweiligen Port zugewiesen werden müssen.
Im Anschluss daran ist ein Routingprozess einzurichten, der für die Verteilung und das Routing zuständig ist. Dieser bekommt eine eindeutige ID, die meist aus der IPv4-Adresse des Interfaces abgeleitet wird. Beim Anlegen eines Routingprozesses werden diesem noch die Netzwerke bekannt gegeben, für die er das Routing übernehmen soll bzw. welche Netze er zu anderen Routern verbreitet. Die Area-ID wird dabei im selben Kommando mitgegeben.

```
ip routing
router ospf 1
 log-adjacency-changes
 network 192.168.0.0 0.0.0.255 area 0
 network 192.178.0.0 0.0.0.255 area 0
```

Listing 4.7: Cisco 3750 - 1 OSPFv2

```
ip routing
router ospf 1
 log-adjacency-changes
 network 192.158.0.0 0.0.0.255 area 1
 network 192.168.0.0 0.0.0.255 area 0
 network 192.188.0.0 0.0.0.255 area 0
```

Listing 4.8: Cisco 3750 - 2 OSPFv2

```
ip routing
router ospf 1
 log-adjacency-changes
 network 192.178.0.0 0.0.0.255 area 0
 network 192.188.0.0 0.0.0.255 area 0
 network 192.198.0.0 0.0.0.255 area 1
 default-information originate always
```

Listing 4.9: Cisco 1861 OSPFv2

Das letzte Kommando *default-information originate always* bewirkt die Weitergabe der Default-Route über OSPF an die angeschlossenen Router. Sobald diese Einstellungen vorgenommen wurden, beginnen die Router mit dem Austausch der Routing Informationen.

4.6.3 OSPF-Routing im IPv6-Netz

Die bisherige Konfiguration sieht nur das IPv4-Routing vor. Damit auch IPv6-Pakete weitergeleitet werden können, sind zuerst die Interfaces mit IPv6-Adressen zu versorgen. Dazu wird das Kommando *ipv6 address* verwendet. Es verlangt die Angabe der IPv6-Adresse zusammen mit der Präfixlänge. Für das Routing werden ebenfalls im Interface-Konfigurationsmodus die OSPF Prozessnummer und die Area-ID übermittelt. Der Befehl *ipv6 ospf <OSPF-Prozessnummer> area <Area-ID>* erledigt dies. Für die einzelnen Router bedeutet das folgende Einstellungen:

```
interface FastEthernet1/0/1
 no switchport
 ipv6 address 2A02:238:1:F00B:3::1/80
 ipv6 ospf 2 area 0

interface FastEthernet1/0/2
 no switchport
 ipv6 address 2A02:238:1:F00B:4::2/80
 ipv6 ospf 2 area 0
```

Listing 4.10: Cisco 3750 - 1 IPv6-Konfiguration

```
interface FastEthernet1/0/1
 no switchport
 ipv6 address 2A02:238:1:F00B:1001::1/80
 ipv6 ospf 2 area 1
!
interface FastEthernet1/0/2
 no switchport
 ipv6 address 2A02:238:1:F00B:2::2/80
 ipv6 ospf 2 area 0

interface FastEthernet1/0/24
 no switchport
 ipv6 address 2A02:238:1:F00B:4::1/80
 ipv6 ospf 2 area 0
```

Listing 4.11: Cisco 3750 - 2 IPv6-Konfiguration

```
interface Vlan10
 ipv6 address 2A02:238:1:F00B:C001::1/80
```

```
 ipv6 ospf 2 area 1

interface Vlan20
 ipv6 address 2A02:238:1:F00B:2::1/80
 ipv6 ospf 2 area 0

interface Vlan30
 ipv6 address 2A02:238:1:F00B:3::2/80
 ipv6 ospf 2 area 0
```

Listing 4.12: Cisco 1861 IPv6-Konfiguration

Nach erfolgreicher Konfiguration der Interfaces erfolgt nun die Konfiguration des Routingprozesses. Im Falle von IPv6 muss dieser neu angelegt werden. Eine Angabe der angeschlossenen Netzwerke ist nicht notwendig, da diese bereits durch die Zuordnung des Routingprozesses und der Area-ID zum Interface erfolgt ist. Beispielhaft für alle verwendeten Router sind hier die notwendigen Kommandos für den Cisco 1861 angegeben:

```
ipv6 router ospf 2
 router-id 186.0.0.0
 log-adjacency-changes
 default-information originate always
```

Listing 4.13: Cisco 1861 OSPFv3

Die Angabe der Router-ID ist nicht zwingend notwendig. Der Router erstellt diese ID auch selbstständig auf Basis vergebener lokaler IPv4-Adressen, was jedoch zu Verwirrungen beim Lesen der Routingtabelle führen kann. Die letzte Zeile der Konfiguration sorgt wie bei IPv4 für die Verbreitung von Default-Routen über OSPF.

4.7 GRE-Tunnelkonfiguration

Für die Einrichtung eines manuellen Tunnels müssen die Tunnelendpunkte vorher gut geplant sein. Zum einen soll damit eine Verbindung in einem IPv6-Netzwerk über eine IPv4-Infrastruktur hergestellt und zum anderen muss dieser Tunnel entsprechend geschützt werden, da er jegliche Paketfilter umgeht. Für einen Tunnel innerhalb des Unternehmensnetzwerks ist der Sicherheitsaspekt etwas weniger kritisch. Trotzdem ist es wichtig zu wissen, welche Tunnel konfiguriert sind und welche eventuelle Sicherheitslücken sich damit ergeben. Für Tunnel von außen in ein Unternehmensnetzwerk hinein ist

es besonders wichtig, diesen mit einem Paketfilter auszustatten, da sonst über diesen Weg ein uneingeschränkter Zugang möglich ist.
Für die Tunnelkonfiguration im Testnetzwerk wird der Weg von einem Cisco 3750 Router zum anderen Cisco 3750 Router gewählt. Die Verbindung zwischen diesen beiden Routern ist als reine IPv4-Verbindung eingerichtet. Der Tunneleingangspunkt wie auch der Tunnelausgangspunkt bedürfen einer Konfiguration. Für Quell- und Zieladresse der Tunnelpunkte wird jeweils die IPv4-Adresse des Partners verwendet. Zusätzlich dazu erhält jeder Tunnelpunkt eine IPv6-Adresse, damit der Router weiß, an welche er die IPv6-Pakete weiterzuleiten hat. Eine einseitige Konfiguration eines Tunnels wäre auch denkbar, würde aber nur den Transport von IPv6-Paketen in eine Richtung ermöglichen. Die Konfiguration eines solchen Tunnels sieht wie folgt aus:

```
interface Tunnel0
 description 6inGRE Tunnel
 no ip address
 ipv6 address 2A02:238:1:F00B:100F::2/80
 ipv6 enable
 ipv6 ospf 2 area 0
 tunnel source 192.168.0.2
 tunnel destination 192.168.0.1
```

Listing 4.14: Cisco 3570 - 1 Tunnelkonfiguration

```
interface Tunnel0
 description 6inGRE Tunnel
 no ip address
 ipv6 address 2A02:238:1:F00B:100F::1/80
 ipv6 enable
 ipv6 ospf 2 area 0
 tunnel source 192.168.0.1
 tunnel destination 192.168.0.2
```

Listing 4.15: Cisco 3570 - 2 Tunnelkonfiguration

Standardmäßig ist jeder Tunnel bei Cisco ein GRE-Tunnel, so dass die Einstellung des Tunnelmodus nicht notwendig ist. Sobald ein Router, der den Tunneltyp IPv6 über IPv4 unterstützt, verwendet wird, sollte auch der entsprechende Modus verwendet werden, da dadurch die Geschwindigkeit der Verarbeitung höher ist. Damit der Tunnel beim OSPF-Routing berücksichtigt wird, bedarf es einer Konfiguration wie bei jedem anderen IPv6-OSPF-Interface üblich. In diesem Fall wird das Interface durch den OSPF-Prozess zwei

verwaltet und gehört der Area null an.
Damit ist die Konfiguration des GRE-Tunnels abgeschlossen.

4.8 GenuGate Firewall

Für den Test der GenuGate Firewall wurde ein Testsystem installiert. Nach der Installation, die nur für IPv4 und IPv6 gemeinsam durchgeführt werden konnte, war jedoch festzustellen, dass kein IPv6-DNS Server in der Konfigurationsoberfläche eintragbar war. Da sich die Firewall ohne diesen Eintrag nicht ordnungsgemäß testen ließ, wurde dieses Problem an den Hersteller gemeldet. Dieser konnte das Problem beheben, so dass der Fortführung des Tests nichts mehr im Wege stand. Für die Überprüfung der Funktionalitäten soll es zum einen möglich sein, eine Verbindung ins Internet über einen Proxy und zum anderen eine Verbindung ohne einen Proxy aufzubauen. Nach der Konfiguration durch den Hersteller ist es nun möglich, über den Proxy im Internet zu surfen. Dabei wird die Verbindung nach außen allerdings noch über IPv4 hergestellt. Die Verbindung vom Client zum Proxy ist jedoch eine IPv6-Verbindung. Das Besuchen von Internetseiten ohne Proxy funktioniert hingegen ohne Einschränkungen komplett über IPv6. Auch andere Dienste können über die Firewall problemlos per IPv6 kommunizieren. So gelang der Aufbau einer ssh-Verbindung ebenso wie der Aufbau einer ftp-Verbindung über die Firewall und das Internet ohne Probleme über IPv4, wie auch über IPv6. Nach der Erstellung einiger Firewallregeln zur Unterbindung von ssh- und ftp-Verbindungen für IPv4 bzw. für IPv6 konnte erfolgreich überprüft werden, dass diese ihre Wirkung zeigten. Die Funktionalitäten sind, wenn man von der Bedienoberfläche und der Dokumentation ausgeht, für IPv4 gleich denen für IPv6. Lediglich das SIP-Protokoll wird in IPv6 noch nicht unterstützt.
Bei der Konfiguration der Browser für die Überprüfung der Firewall zum Surfen im Internet ist aufgefallen, dass die Eingabe des Proxyservers unterschiedlich erfolgt. Der Internet Explorer erwartet die Angabe der IPv6-Adresse in der sogenannten Literal-IPv6-Schreibweise. Dabei wird die IPv6-Adresse von eckigen Klammern umschlossen. Der Mozilla Firefox hingegen erwartet die Eingabe der IPv6-Adresse ohne eckige Klammern. Diese Einstellung wird hauptsächlich in Testnetzwerken Verwendung finden, da im laufenden Betrieb für die Angabe des Proxyservers meist der DNS-Name verwendet wird und eine Namensauflösung stattfinden kann.

Kapitel 5

Neue Sicherheitstechniken von IPv6

Während der Entwicklung des Internetprotokolls der Version 4 gab es keine Überlegungen zum Thema Sicherheit. Damals war das Netz für einen kleinen Kreis von Forschern ausgelegt, so dass es nicht notwendig erschien, Sicherheitsmechanismen in die IP-Architektur einzubauen. Wenn ein Zugriffsschutz erwünscht war, fand die Realisierung nur auf Programmebene statt. Der Zugang zu FTP oder Telnet wurde mit einer Passwortabfrage gesichert. Die Übertragung lief jedoch ungesichert ab.
Beim Entwurf von IPv6 war von vornherein klar, dass elementare Sicherheitsfunktionen im Protokoll vorhanden sein müssen, die von jeder Internetplattform, die IPv6 anbietet, unterstützt werden. Dafür wurde das IPsec-Paket entworfen, um die Sicherheit der Übertragung auf Protokollebene zu gewährleisten. Das folgende Kapitel beschreibt die Gefahren, die in einem Netzwerk auftreten können und gibt einen Überblick zum Thema IPv6. Am Ende des Kapitels wird auf die Integration von IPsec in das Testnetzwerk aus Abschnitt 4 eingegangen.

5.1 Gefahren im Netzwerk

Zur Evaluierung einer Sicherheitsstrategie für das eigene Netzwerk ist es wichtig, die Gefahren zu kennen, um darauf entsprechend reagieren zu können. In den meisten Fällen wird das Augenmerk nur auf Gefahren aus fremden Netzwerken gelegt. Jedoch sind auch die Gefahren aus dem eigenen Netzwerk nicht zu vernachlässigen. Solche Gefahren können sein:

- Missbrauch von Rechten (Passwortdiebstahl),
- Software-Schwachstellen,

- Routing-Missbrauch,
- Abhören von Leitungen,
- Wiedereinspielen von Nachrichten,
- Menschliches Fehlverhalten,
- Nichtbeachtung von IT-Sicherheitsvorkehrungen,
- Manipulation von IT-Geräten und
- Diebstahl von Software / Daten.

Auch diese Gefahren sind in einem ganzheitlichen Sicherheitskonzept dringend mit zu beachten. Gerade die oben genannten Gefahren können nicht allein durch bessere Sicherheitsmechanismen kontrolliert werden. Auch die Schulung von IT-Mitarbeitern und Anwendern ist sehr wichtig.
Die Angriffsszenarien haben sich mit der Einführung von IPv6 nicht von Grund auf geändert. Hacker und Hersteller von Sicherheitssoftware werden sich auch in Zukunft immer wieder Rennen liefern, um Sicherheitslücken zu finden und diese zu beheben. Zu beachten ist, dass bei der Verwendung beider Protokolle für jedes Protokoll entsprechende Sicherheitskonzepte erarbeitet und verwendet werden müssen. So sind in IPv6-fähigen Firewalls IPv4-Regeln sowie separate IPv6-Regeln zu erstellen. Neue Angriffsmöglichkeiten, die sich mit der Einführung von IPv6 ergeben, sind die Folgenden:

- Aufwändigeres Scannen von Hosts und Ports. Dabei aber auch leichteres Verbergen von böswilligen Benutzern.
- Der Einsatz von Tunnel-Mechanismen ermöglicht es, Pakete in fremde Netze zu schmuggeln, ohne die Absenderadresse preis zu geben. Damit ist auch das Umgehen von Firewalls möglich.
- Der Routing-Header ermöglicht es, den endgültigen Empfänger zu verbergen.
- Der Einsatz der Autokonfiguration von IPv6 schafft die Voraussetzung für eine böswillige Konfiguration der Hosts am Link.
- Die einfache Konfiguration von IPv6 in aktuellen Betriebssystemen aktiviert meist automatisch diverse Tunnelmechanismen, über die Hacker auf Rechner in IPv4-Netzwerke eindringen können, da dieses eigentlich noch nicht für IPv6 konfiguriert ist und somit auch kein IPv6-Sicherheitskonzept existiert.

5.2 IPsec Grundlagen

IPsec bezeichnet eine Sicherheitsarchitektur, die beide Versionen des Internetprotokolls unterstützt. RFC 4301 [S. Kent, 2005] beschreibt den Aufbau im Detail.
Das IPsec-Paket umfasst die folgenden Elemente:

- Protokoll zur Verschlüsselung (Encapsulation Security Payload),
- Schlüsselmanagement,
- Security-Policies und Security-Associations zwischen Kommunikationspartnern,
- Authentisierungsprotokoll (Authentication Header),
- Beschreibung für die Verwendung von Verschlüsselungs- und Authentifizierungsalgorithmen und
- Definition von Anforderungen und Mechanismen zur Gewährleistung der Sicherheit auf Netzwerk-Ebene.

Die folgenden vier Punkte bilden die Grundlage für die Gewährleistung der Sicherheit:

- Unverfälschtheit, Integrität:
  Dieses Ziel garantiert die Entdeckung jeglicher Änderung der übertragenen Daten.
- Vertraulichkeit:
  Die zu übertragenden Daten können von Unberechtigten nicht geändert oder gelesen werden.
- Verpflichtung:
  Das Senden, Löschen oder Erhalten von Daten darf von keinem Kommunikationspartner abgelehnt werden.
- Authentizität:
  Bestätigung der Identität des Absenders oder Empfängers.

Die Einhaltung der Ziele ist nur durch den Einsatz von einer Verschlüsselung, die die Vertraulichkeit gewährt und einer sicheren Prüfsumme, welche die Integrität sichert, möglich. Eine einfache Form der Verschlüsselung, die heute weniger Verbreitung findet, ist die symetrische Verschlüsselung. Dabei verwenden der Absender und der Empfänger einen gemeinsamen Schlüssel, der zur Verschlüsselung und Entschlüsselung dient.

Das aktuell am häufigsten eingesetzte Verschlüsselungsverfahren ist die sogenannte Public-Key-Cryptography, bei der der RSA-Algorithmus zum Einsatz kommt. Dabei wird ein öffentlicher Schlüssel verteilt, der allen Kommunikationspartnern bekannt ist. Der zweite (private) Schlüssel ist geheim und nicht zur Weitergabe bestimmt. Wenn jemand eine Nachricht verschicken möchte, verschlüsselt er diese mit dem öffentlichen Schlüssel des Empfängers. Dieser entschlüsselt nun die empfangene Nachricht mit seinem privaten Schlüssel.

5.2.1 Security Associations

Als Security Associations (SA) werden Festlegungen bezeichnet, die zwischen zwei Kommunikationspartnern auszuhandeln sind. Darin enthalten ist ein Schlüssel (Key), ein Authentifizierungs- oder Verschlüsselungsmechanismus sowie zusätzliche spezielle Parameter für den jeweiligen Algorithmus. Sie werden für jeden Sicherheitsdienst benötigt. Damit ergibt sich für eine gegenseitige Verbindung von zwei Kommunikationspartnern, die sowohl verschlüsseln als auch authentifizieren wollen, eine Anzahl von vier Security Associations.
IPsec bietet dafür zwei Möglichkeiten an. Zum einen gibt es den Transportmodus, bei dem die SA zwischen zwei Endknoten vereinbart wird. Darin enthalten sind die Verschlüsselung oder Authentifizierung für die Daten in allen IP-Paketen. Der IP-Header bleibt hier unverschlüsselt. Zum anderen gibt es den Tunnelmodus, in dem zwischen zwei Sicherheitsgateways die SA definiert wird. Die Pakete zwischen den Gateways werden in diesem Modus komplett in einen neuen Header verpackt und verschlüsselt oder authentifiziert. Somit bildet der Tunnelmodus das Fundament für ein VPN (Virtual Private Network).

5.2.2 IPv6-Sicherheitskomponenten

IPsec ist ein Sicherheitsmechanismus, der für beide Protokolle, IPv4 und IPv6, eingesetzt werden kann. Er stellt für beide Protokolle die gleiche Sicherheit dar. Einziger Unterschied ist, dass die Unterstützung für IPsec bei IPv4 erst zu installieren ist, während diese bei IPv6 jede Implementierung enthält.
Die Spezifikation sieht die Authentication-Header (AH) und die Encapsulating-Security-Payload-Header (ESP) vor, die bei IPv6 die Extension-Header darstellen.

Authentication-Header Der Authentication-Header (AH) ermöglicht die Authentisierung und Integrität für IPv6-Pakete. Für die Authentisierung werden verschiedene Mechanismen unterstützt. Die Definition dieses Headers ist im RFC 4302 [Kent, 2005a] zu finden. Darin wird eine erweiterte Sequenznummer (ESN) beschrieben. Sie dient der Unterstützung der high-speed IPsec-Implementierung. Ob diese ESN benutzt wird, legen die Kommunikationspartner in der SA fest. Beim Einsatz von IKEv2 wird ESN standardmäßig verwendet. Für die Berechnung der Prüfsumme werden die Key Message Authentication Codes, welche auf symetrischen Schlüssel-Algorithmen basieren sowie Einweg-Hash-Funktionen unterstützt.
Der AH kann im Tunnel- und im Transportmodus verwendet werden.

Encapsulating Security Payload Header Der Encapsulating Security Payload Header (ESP) ermöglicht Integrität und Vertraulichkeit für IPv6-Pakete. Die Definition ist in RFC 4303 [Kent, 2005b] zu finden. Der zum Einsatz kommende Verschlüsselungsalgorithmus wird ebenfalls in der SA ausgehandelt. Dies kann vorher durch Konfiguration der Kommunikationspartner beeinflusst werden, so dass auch eine Verhandlung über das Key Exchange Protokoll möglich ist. Im Transportmodus werden beim Einsatz von ESP der IP-Header und die darauffolgenden Erweiterungsheader nicht verschlüsselt, da sie sonst nicht von Routern auf dem weiteren Paketweg zu lesen wären. Im Tunnelmodus wird, wie in Abschnitt 5.2.1 beschrieben, das gesamte Paket verschlüsselt.

5.3 Schlüsselverwaltung

Die Schlüsselverwaltung ist notwendig, damit zwei Kommunikationspartner eine Security Association aushandeln können. Dies erfolgt meist über ein unsicheres Netzwerk.

5.3.1 IKEv1

Internet Key Exchange (IKE) definiert ein Protokoll zum Austausch von Schlüsseln und Verhandeln über die verwendeten Parameter in der SA. Die Version 1 ist im RFC 2409 [D. Harkins, 1998] definiert und wurde mit RFC 4109 [Hoffman, 2005] aktualisiert. Dieses Protokoll besteht genau genommen aus drei Teilen, die im Folgenden erläutert werden.

- ISAKMP (Internet Security Association and Key Management)
  In diesem Teil wird ein Framework definiert, das den Austausch von Schlüsseln

und die Verwaltung von SAs regelt. Verschiedene Mechanismen können verwendet werden, da keine genauen Details im RFC 2408 [D. Maughan, 1998] definiert sind.

- SKEME (Versatile Secure Key Exchange Mechanism for the Internet)
  Diese schnelle Schlüsselaustauschtechnik wird in [Krawczyk, 1996] beschrieben. Nur einige Funktionen dieser Technik werden für IKE verwendet.

- Oakley Key Determination Protocol
  Dieses Protokoll stellt eine Erweiterung für den Diffie/Hellman Algorithmus dar und realisiert den Austausch von Schlüsseln. IKE nutzt nur ausgewählte Funktionen davon.

Der Austausch der Schlüssel über IKE wird in zwei Phasen über den UDP-Port 500 durchgeführt.
In der ersten der beiden Phasen wird ein authentifizierter Kommunikationskanal ausgehandelt (ISAKMP Security Association). Dies geschieht unter Zuhilfenahme des Diffie/Hellman-Verfahrens und eines verschlüsselten Identifikations-Tokens. Die Authentisierung kann entweder über vorher ausgetauschte Schlüssel, einer RSA-Prüfsumme, die mit dem privaten Schlüssel des Absenders verschlüsselt wurden oder über den öffentlichen Schlüssel des Empfängers geschützt werden.
In der zweiten Phase findet ein Austausch der zu verwendenden Algorithmen und der notwendigen Schlüssel über den in der ersten Phase erstellten sicheren Kommunikationskanal statt, die für andere Protokolle (z.B. IPsec) notwendig sind.
Ein häufiger Wechsel der Schlüssel erhöht die Sicherheit. IKE realisiert dies durch mehrere Verhandlungen (Phase zwei), die über den in Phase eins aufgebauten gesicherten Kanal stattfinden.

5.3.2 IKEv2

Die Version 2 von IKE enthält einige Verbesserungen für den Einsatz von IPsec bei der Verwendung von NAT-Traversal sowie für die erweiterte Authentifizierung. IKEv2 ist in RFC 4306 [C. Kaufman, 2005] definiert.
Die Phase eins von IKEv2 besteht aus zwei Nachrichtenpaaren zum Austausch kryptographischer Algorithmen. Zudem werden einmalig generierte Zufallswerte (Nonces) übertragen und ein Diffie-Hellman Austausch durchgeführt. Das zweite Nachrichtenpaar authentisiert die vorher ausgetauschten Nachrichten und übermittelt Identitäten sowie Zertifikate. Hierbei wird die erste sogenannten Child-Security-Association ausgehandelt.
Die von IKEv2 unterstützten Algorithmen sind im RFC 4307 [Schiller, 2005] zu finden.

Eine Interoperabilität verschiedenster Implementierungen ist gewährleistet, wenn die genannten Regeln beachtet werden.
Bei IKEv1 wurde die Gültigkeit der Security Associations ausgehandelt. Bei IKEv2 ist jeder Teilnehmer einer SA dafür verantwortlich und muss bei Bedarf einen neuen Schlüssel verwenden. Wenn die Teilnehmer unterschiedliche Gültigkeitszeiten besitzen, verlangt der Teilnehmer mit der kürzeren Gültigkeit diesen Wechsel. Eine weitere Neuerung von IKEv2 ist die Verwendung paralleler SAs mit gleichen Verkehrskennungen zwischen zwei Endpunkten. So können Daten mit unterschiedlichen QoS Anforderungen zwischen den SAs übertragen werden.

5.4 Zusammenspiel von IPsec und neuen IPv6-Komponenten

Die Integration von IPsec in IPv6 ist ein großer Schritt in Richtung sicherer Kommunikation im Internet. Doch nicht in allen Bereichen kann IPsec ohne Probleme eingesetzt werden. QoS (Quality of Service) erlaubt es einem Router, ein Paket anhand bestimmter Kriterien zu verwerfen. Für IPsec stellt dies jedoch einen Paketverlust und somit eine Verletzung der Sicherheit dar. Eine Folge davon wäre, dass ein bestimmter Dienst nicht verfügbar ist.
Die Verwendung von IPsec und Tunneln stellt ein weiteres Problem dar. Ein IPsec-Tunnel, der eine Ende-zu-Ende Verbindung darstellt, verhindert eine Kontrolle der Daten durch die Firewall, um gefährliche oder unautorisierte Pakete zu erkennen. Möglicherweise könnte ein solches Paket in einem IPsec-Tunnel falsche Routing-Informationen in ein Netzwerk einschleusen. Die Lösung für dieses Problem wäre eine Ende-zu-Sicherheitsgateway Security Association. Diese ist aber in keinem Standard enthalten.
In dieselbe Richtung geht das Problem von IPsec und NAT, da gerade in NATs häufig IPsec für die Verbindung zu einem Unternehmen oder Ähnlichem verwendet wird. Die NAT-Adressübersetzung im IP-Header und in einigen Fällen auch die Port-Übersetzung führen zu Problemen bei IP-Paketen, die durch Authentication oder Veschlüsselung gesichert sind.

5.5 Enterprise Security Mechanismen

In vielen IPv4-Netzwerken ist eine Ende-zu-Ende Sicherheit nicht möglich, da oftmals NAT-Gateways eingerichtet wurden. Der Einsatz von IPv6 würde eine Ende-zu-Ende Si-

cherheit und die dafür notwendige Transparenz wieder ermöglichen. Jedoch haben sich viele Netzbetreiber an NAT und private Adressen als eine Art Sicherheitsmechanismus gewöhnt, da dadurch die interne Topologie nicht für die Außenwelt sichtbar ist.
Diese Transparenz kann mit IPv6 wiederhergestellt werden und Ende-zu-Ende Verbindungen sind möglich. Damit der Schutz des IPv6-Netzwerks nicht zu kurz kommt, muss dafür ein spezielles IPv6-Sicherheitskonzept erstellt werden. Außerdem ist die Konfiguration von NATs nicht zu empfehlen. Der Schutz der internen Topologie kann mit alternativen Mechanismen erfolgen. Beispielsweise sind private Adressen (RFC 4941 [T. Narten, 2007a]) oder Unique Local Adressen (RFC 4193 [R. Hinden, 2005]) einzurichten. Weitere Empfehlungen zur Erhöhung der Sicherheit von lokalen Netzwerken sind im RFC 4864 [G. Van de Velde, 2007] zu finden.
Für IPv4 wurden häufig zur Erhöhung der Sicherheit Perimeter-Firewalls und NATs eingesetzt. Dies ist genauso in IPv6 möglich. Auf lange Sicht gesehen kann dies jedoch einige Einschränkungen bedeuten. Für IPv6-Netzwerke sollte ein Weg gewählt werden, der die Sicherheit des Netzwerks erhöht, aber gleichzeitig auch eine Ende-zu-Ende Kommunikation ermöglicht. IPsec wird bei IPv6 in jedem Knoten angeboten. Sobald es also ein Angreifer schafft, hinter die Firewall zu gelangen, indem er z.B. einen IPsec-Tunnel aufbaut, hat er ein ungesichertes Netzwerk vor sich.
Ein besseres Sicherheitskonzept für IPv6-Netzwerke ist eine Kombination aus zentralen Sicherheitseinrichtungen und gesicherten Hosts. Damit wird hohe Sicherheit an den Endpunkten und eine Steuerung des Verhaltens der Perimeter-Firewall durch die Hosts erreicht. Die Perimeter-Firewall schützt hierbei vor allgemeinen Angriffen auf das Netzwerk und die Knoten schützen sich selbst vor knotenspezifischen Angriffen. Zur Zeit wird an Protokollen gearbeitet, die es Endknoten erlauben, Firewalls zu kontrollieren und zu informieren. Für den Anfang werden bei einem baldigen Umstieg auf IPv6 die Sicherheitsmechanismen aus IPv4 verwendet. NAT sollte jedoch ausgeschlossen werden. Außerdem darf das Ziel der Sicherung von Endknoten und Netzwerken nicht aus den Augen verloren werden. Modelle für ein solches neues Sicherheitsmodell können laut [Hagen, 2009] die Folgenden sein:

- verteiltes Endknoten-Firewallmodell
  Ein sogenannter Sicherheitsserver authentisiert die Endknoten im Netz und verteilt Sicherheitsregeln an die Endknoten-Firewalls. Darin sind Firewallkonfigurationen, IPsec-Schlüssel und ähnliches enthalten. Bei diesem Modell wird auf eine Perimeter-Firewall verzichtet und die Endknoten sind allein für ihren Schutz zuständig.

- hybrides verteiles Firewallmodell
  Bei diesem Modell authentisieren sich die Endknoten und die Perimeter-Firewalls und bekommen Sicherheitsregeln zugeteilt. Die Endknoten können dabei unterschiedliche Level an Privilegien erhalten. Den Zugang zu den jeweiligen Netzen und Knoten sowie den IPsec-Schlüsseln und Protokollen regelt der Sicherheitsserver.

5.6 IPsec-Integration im Testnetzwerk

Mithilfe der IPsec Funktionen ist es möglich, jeglichen Datenverkehr verschlüsselt und authentifiziert zu übertragen. Um eine solche Übertragung einzurichten, sind eine Handvoll Konfigurationsschritte notwendig. Für einen einfachen Test der grundlegenden Funktionalität wird im Testnetzwerk eine Telnet-Verbindung zwischen dem Windows 7 Client-PC und dem Windows 2008 Server aufgebaut. Diese ist im Normalfall nicht verschlüsselt, so dass mit einem einfachen Netzwerkanalysewerkzeug (hier: Wireshark) die Pakete zu untersuchen sind. Im ersten Teil wird dabei eine unsichere Verbindung verwendet, die es ermöglicht, die übertragenen Daten mitzulesen. Im zweiten Teil wird eine verschlüsselte Verbindung über IPsec aufgebaut und darin die Telnet-Sitzung gestartet. In diesem Fall können keine Details der Telnet-Sitzung mitgelesen werden.

5.6.1 Einrichten einer Telnet-Verbindung

Zur Einrichtung einer Telnet-Verbindung sind als erstes Telnet-Client und Telnet-Server zu installieren. Dies geschieht im Windows über den Unterpunkt *Windowskomponenten installieren* unter Software in der Systemsteuerung. Im Testnetzwerk wird auf dem Windows 2008 Server der Telnet-Server und auf dem Windows 7 PC der zugehörige Client installiert. Danach kann eine Telnet-Verbindung über die Kommandozeile mit dem Aufruf

```
telnet 2a02:238:1:f00b:1001::2
```

gestartet werden. Sobald die Verbindung aufgebaut ist, findet eine Aufforderung statt, Benutzername und Passwort zu übergeben. Dafür ist vorher ein Benutzer, der Mitglied in der *TelnetClients* Gruppe ist, anzulegen. Dies ist notwendig, damit keine wichtigen Zugangsdaten unverschlüsselt über das Netz übertragen werden. Während dieses Anmeldevorgangs lief das Netzwerkanalysewerkzeug Wireshark und schnitt den Paketverkehr mit. Der Ausschnitt Listing 5.1 zeigt die Daten, die nach der erfolgreichen Anmeldung am Telnet-Server angezeigt wurden, vollkommen unverschlüsselt.

```
No.     Time        Source                Destination           Protocol Info
     94 16.588270   2a02:238:1:f00b:1001::2 2a02:238:1:f00b:c001::2 TELNET    Telnet Data ...

Frame 94: 259 bytes on wire (2072 bits), 259 bytes captured (2072 bits)
Ethernet II, Src: Cisco_14:f1:a2 (00:1e:13:14:f1:a2), Dst: FujitsuT_80:4d:ea (00:19:99:80:4d:ea)
Internet Protocol Version 6, Src: 2a02:238:1:f00b:1001::2 (2a02:238:1:f00b:1001::2), Dst: 2a02:238:1:f00b:c001::2 (2
    a02:238:1:f00b:c001::2)
Transmission Control Protocol, Src Port: telnet (23), Dst Port: 49195 (49195), Seq: 150, Ack: 96, Len: 185
    Source port: telnet (23)
    Destination port: 49195 (49195)
    [Stream index: 0]
    Sequence number: 150    (relative sequence number)
    [Next sequence number: 335    (relative sequence number)]
    Acknowledgement number: 96    (relative ack number)
    Header length: 20 bytes
    Flags: 0x18 (PSH, ACK)
    Window size: 66048 (scaled)
    Checksum: 0xfca3 [validation disabled]
    [SEQ/ACK analysis]
Telnet
    Data: \r\n
    Data: \r\n
    Data: *===============================================================\r\n
    Data: Microsoft Telnet Server.\r\n
    Data: *===============================================================\r\n
    Data: C:\Users\Administrator>
```

Listing 5.1: Wireshark Ausschnitt Telnet-Verbindung ohne IPsec

Auch der Benutzername und das Passwort werden bei einfachen Telnet-Verbindungen im Klartext über das Netzwerk geschickt. Dies stellt in jedem Fall ein Sicherheitsrisiko dar. Diese Art der Verbindung ist nicht mehr zu verwenden. Ersatz gibt es beispielsweise durch den Einsatz einer SSH-Verbindung. Ein anderer Weg wäre vorher eine verschlüsselte IPsec-Verbindung aufzubauen und darüber eine Telnet-Sitzung zu etablieren.

5.6.2 Erstellen einer IPsec-Verbindung

Zur Einrichtung der sicheren Verbindung über ein Netzwerk mit IPsec sind einige Schritte notwendig, die im Folgenden näher erläutert werden.
Als erstes ist eine Sicherheitsrichtlinie auf dem Server zu erstellen. Dazu wird eine neue IP-Sicherheitsrichtlinie unter *Verwaltung* → *Lokale Sicherheitsrichtline* angelegt. Im dazugehörigen Assistenten kann ein Name und eine Beschreibung der Richtlinie angegeben werden. Am Ende des Assistenten sollte der Punkt *Eigenschaften bearbeiten* aktiviert sein. Unter den Eigenschaften ist nun eine neue IP-Sicherheitsregel durch Anklicken des Buttons *Hinzufügen* anzulegen. Im erscheinenden Sicherheitsregel-Assistenten muss der Punkt *Durch diese Regel wird kein Tunnel festgelegt* aktiviert werden. Für die darauf folgende Auswahl des Netzwerktyps ist *Alle Netzwerkverbindungen* zu wählen. Im Anschluss daran wird eine IP-Filterliste erstellt. Dies geschieht über den Button *Hinzufügen*. Die Filterliste ist nun zu benennen und um Quell- und Zieladressen für diese Filterliste zu erweitern. Im vorliegenden Fall können alle Adressen ausgewählt werden. Dazu müssen für den Filter, der über den Button *Hinzufügen* erzeugt wird, die Quell- und Zieladressen

auf *beliebige IP-Adresse* gesetzt werden. Als Protokolltyp im nächsten Schritt des Assistenten ist im Fall von Telnet *TCP* zu wählen. Darauf folgt die Angabe des Ports, von dem aus eine Verbindung aufbaubar ist und auf welchen Port diese führt. Für Telnet ist *von diesem Port: 23* zu *zu jedem Port* zu wählen. Damit ist die IP-Filterliste erstellt und kann im Sicherheitsregelassistenten ausgewählt werden. Der nächste Schritt sieht die Konfiguration der Filteraktion vor. Diese kann über den Button *Hinzufügen* erstellt werden. Im daraufhin erscheinenden Assistenten muss bei *Allgemeinen Optionen Sicherheit aushandeln* ausgewählt sein. Im Weiteren ist die Option *Keine unsicheren Verbindungen zulassen* auszuwählen, damit keine Verbindung zu Clients ohne IPsec aufbaubar ist. Für die Sicherheitsmethode im darauffolgenden Schritt ist Integrität und Verschlüsselung auszuwählen, damit sowohl die Authentizität als auch die Sicherheit der Daten gewährleistet ist. Auch diese Filteraktion erscheint im Sicherheitsregelassistenten. Anschließend ist eine Authentifizierungsmethode auszuwählen. Hier kann z.B. ein Zertifikat verwendet werden. Für das Testnetzwerk wird im untersten Punkt *Diese Zeichenfolge zum Schutz des Schlüsselaustauschs verwenden* eine Zeichenfolge eingetragen. Danach sind alle Assistenten zu schließen und die Sicherheitsrichtlinie ist fertiggestellt.
Diese Richtlinie ist nun zuzuweisen, damit sie auch aktiv ist. Dies geschieht über einen Rechtsklick auf die Sicherheitsrichtlinie und dann auf den Punkt *Zuweisen*. Damit ist die Konfiguration auf dem Server abgeschlossen.
Die Konfiguration auf dem Client funktioniert in gleicher Weise. Da diese Einrichtung recht umfangreich ist, wenn sie auf jedem PC eingerichtet werden muss, der eine Verbindung zum Telnet-Server aufnehmen soll, kann diese auch auf dem Server exportiert und z.B. über einen USB-Stick zum Client übertragen werden. Der Export der Sicherheitsrichtlinie erfolgt in der Verwaltung der Sicherheitsrichtlinien durch Rechtsklick unter die Sicherheitsrichtlinie und durch Auswählen von *Richtlinien exportieren* unter dem Punkt *Alle Aufgaben*. Der Import auf dem Client-Gerät erfolgt auf ähnliche Weise. In den Sicherheitsrichtlinien ist mit einem Rechtsklick unter die letzte Richtlinie über den Punkt *Alle Aufgaben*, *Richtlinien importieren* auszuwählen. Nachdem die Richtlinien importiert sind, können sie wie gehabt durch Anklicken mit der rechten Maustaste und Auswählen von *Zuweisen* aktiviert werden.
Sobald nun ein Verbindung über den Port 23 aufgebaut wird, erfolgt vorher der Aufbau einer IPsec Verbindung über ISAKMP. Der Wireshark Ausschnitt Listing 5.2 zeigt dieses Verhalten.

```
No.     Time        Source                Destination           Protocol Info
     10 0.974691    2a02:238:1:f00b:c001::2 2a02:238:1:f00b:1001::2 ISAKMP    Identity Protection (Main Mode)

Frame 10: 250 bytes on wire (2000 bits), 250 bytes captured (2000 bits)
Ethernet II, Src: FujitsuT_80:4d:ea (00:19:99:80:4d:ea), Dst: Cisco_14:f1:a2 (00:1e:13:14:f1:a2)
Internet Protocol Version 6, Src: 2a02:238:1:f00b:c001::2 (2a02:238:1:f00b:c001::2), Dst: 2a02:238:1:f00b:1001::2 (2
    a02:238:1:f00b:1001::2)
User Datagram Protocol, Src Port: isakmp (500), Dst Port: isakmp (500)
Internet Security Association and Key Management Protocol
    Initiator cookie: 359c80c5af975d25
    Responder cookie: 0000000000000000
    Next payload: Security Association (1)
    Version: 1.0
    Exchange type: Identity Protection (Main Mode) (2)
    Flags: 0x00
    Message ID: 0x00000000
    Length: 188
    Type Payload: Security Association (1)
    Type Payload: Vendor ID (13) : MS NT5 ISAKMPOAKLEY
    Type Payload: Vendor ID (13) : Microsoft L2TP/IPSec VPN Client
    Type Payload: Vendor ID (13) : Unknown Vendor ID
    Type Payload: Vendor ID (13) : Microsoft Vid-Initial-Contact
    Type Payload: Vendor ID (13) : Unknown Vendor ID

No.     Time        Source                Destination           Protocol Info
     19 1.007403    2a02:238:1:f00b:1001::2 2a02:238:1:f00b:c001::2 ISAKMP    Quick Mode

Frame 19: 138 bytes on wire (1104 bits), 138 bytes captured (1104 bits)
Ethernet II, Src: Cisco_14:f1:a2 (00:1e:13:14:f1:a2), Dst: FujitsuT_80:4d:ea (00:19:99:80:4d:ea)
Internet Protocol Version 6, Src: 2a02:238:1:f00b:1001::2 (2a02:238:1:f00b:1001::2), Dst: 2a02:238:1:f00b:c001::2 (2
    a02:238:1:f00b:c001::2)
User Datagram Protocol, Src Port: isakmp (500), Dst Port: isakmp (500)
Internet Security Association and Key Management Protocol
    Initiator cookie: 359c80c5af975d25
    Responder cookie: d4fabffcb6af293f
    Next payload: Hash (8)
    Version: 1.0
    Exchange type: Quick Mode (32)
    Flags: 0x03
    Message ID: 0x00000001
    Length: 76
    Encrypted Data (48 bytes)
```

Listing 5.2: Wireshark Ausschnitt Verbindungsaufbau mit ISAKMP

Danach ist die IPsec-Verbindung aufgebaut und jeglicher Datenverkehr darüber wird verschlüsselt.

5.6.3 Etablieren einer Telnet-Sitzung über IPsec

Der Aufbau der Telnet-Sitzung über IPsec setzt die Erstellung der IPsec-Verbindung, wie im Abschnitt 5.6.2 erläutert, voraus. Eine IPsec Verbindung wird dann automatisch beim Versuch des Aufbaus der Telnet-Verbindung erstellt. Folglich werden daraufhin alle weiteren Telnet-Pakete verschlüsselt über die IPsec-Verbindung übertragen und sind nicht mehr für andere Mitglieder des Netzwerks mit einem Netzwerkanalysetool lesbar. Eine solche sichere Telnet-Sitzung ist im Ausschnitt Listing 5.3 ersichtlich. In diesem Fall sind keine unverschlüsselt übertragenen Daten zu erkennen, da diese sich in ESP-Paketen (Encapsulation Security Payload) befinden.

```
No.     Time        Source                Destination           Protocol Info
    105 9.495969    2a02:238:1:f00b:1001::2 2a02:238:1:f00b:c001::2 ESP      ESP (SPI=0x35e8cb91)

Frame 105: 290 bytes on wire (2320 bits), 290 bytes captured (2320 bits)
Ethernet II, Src: Cisco_14:f1:a2 (00:1e:13:14:f1:a2), Dst: FujitsuT_80:4d:ea (00:19:99:80:4d:ea)
Internet Protocol Version 6, Src: 2a02:238:1:f00b:1001::2 (2a02:238:1:f00b:1001::2), Dst: 2a02:238:1:f00b:c001::2 (2
    a02:238:1:f00b:c001::2)
    0110 .... = Version: 6
    .... 0000 0000 .... .... .... .... .... = Traffic class: 0x00000000
    .... .... .... 0000 0000 0000 0000 0000 = Flowlabel: 0x00000000
    Payload length: 236
    Next header: ESP (0x32)
    Hop limit: 126
    Source: 2a02:238:1:f00b:1001::2 (2a02:238:1:f00b:1001::2)
    Destination: 2a02:238:1:f00b:c001::2 (2a02:238:1:f00b:c001::2)
Encapsulating Security Payload
    ESP SPI: 0x35e8cb91
    ESP Sequence: 30
```

Listing 5.3: Wireshark Ausschnitt Telnet-Verbindung mit IPsec

Kapitel 6

Fazit

Für die Umstellung auf das neue Internetprotokoll ist ein umfangreiches Grundwissen des alten wie auch des neuen Protokolls notwendig, wie es in Kapitel 2 zu finden ist. Natürlich ist eine Umstellung auch ohne dieses Wissen möglich, jedoch verringert sich der Aufwand für die Fehlersuche erheblich, wenn durch den Einsatz eines Paketanalysators der Verursacher des Problems schnell gefunden werden kann. In den meisten Fällen wird es bei gewissenhaftem Aufbau des Testnetzwerks bzw. der Migration des Unternehmensnetzwerks vor allem Fehler durch die Umsetzung von IPv6 in der Software geben. Diese sind durch ein einfaches Update zu beheben. Jedoch ist dafür zuerst einmal die fehlerhafte Software bzw. das Gerät herauszufinden.
Ebenso sind die Migrationsmechanismen ausführlich zu analysieren, inwieweit sie für die eigenen Bedürfnisse die entsprechenden Lösungen bieten. Auch die Evaluierung neuer Mechanismen, die zum Zeitpunkt dieser Masterarbeit noch nicht verfügbar waren, sind zu betrachten.
Ein weiterer wichtiger Punkt ist die Erstellung eines Adressierungsplans. Er ist notwendig, um eventuelle Änderungen des Netzwerkaufbaus, die durch die Adressknappheit oder andere historische Faktoren entstanden sind, zu verbessern oder komplett neu zu entwickeln. Dabei ist immer im Hinterkopf zu behalten, dass es bei IPv6 keinen Mangel an Adressen geben wird und diese entsprechend den Vorstellungen für ein logisch aufgebautes Netzwerk verteilt werden können.
Das folgenden Kapitel gibt eine Zusammenfassung der Migrationsverfahren und ihrer Eigenschaften.

6.1 Auswertung der Migrationsverfahren

Das wohl einfachste Migrationsverfahren stellt Dual-Stack dar. Durch die Tatsache, dass dafür genügend IPv4-Adressen zur Verfügung stehen müssen, ist diese Technik nicht überall einsetzbar. Eine Kombination mit anderen Mechanismen ermöglicht daher eine individuellere Vorgehensweise, die auf das Unternehmensnetzwerk zugeschnitten ist.
Eine Übersicht der Eigenschaften der Verfahren ist in Tabelle 6.1 zu finden. Diese ist im Zusammenhang mit dem Forschungsprojekt [Seeber, 2011], das dieser Masterarbeit voraus ging, entstanden.

Merkmal	**Proxy**	**Protokollübersetzer**	**Tunnel**	**Dual-Stack**
Migrationsanforderung	−−	+	+ +	+ +
Erweiterbarkeit	−	+	+	+ +
Wartbarkeit	−	○	○	+
Sicherheit	−−	+	○	+
Effizienz	−	−	−	+ +
Fehlertoleranz	−	−	○	○
Transparenz	−−	+ +	+	+ +
Unterstützung	−−	−	+	+

Tabelle 6.1: Bewertung der Migrationsstrategien

Die Tabelle bestätigt, dass nur eine Kombination von Migrationsverfahren zum Erfolg führt. Dabei ist es wichtig, die in Kapitel 3 genannten Techniken und vor allem deren Schwächen zu kennen, um das Netzwerk vor Angriffen optimal zu schützen. Gerade die Verwendung von automatischen Tunneln kann eine unübersichtliche Netzwerktopologie zur Folge haben, die eine Überwachung erschwert.

6.2 Geeignete Vorgehensweise bei der Migration

Dieser Abschnitt erläutert die ideale Vorgehensweise für die Umstellung auf das Internetprotokoll der Version 6 im Unternehmen perdata. Zunächst ist mit dem bei zuständigen ISP angeforderten Präfix ein geeigneter Adressierungsplan zu erstellen. Dabei sind auch die Möglichkeiten einer kompletten Neuordnung der Netzwerkstruktur mit einzubeziehen, da es keinen Adressmangel mehr gibt. Nach der Erstellung ist dieser Plan hinsichtlich des Routings und der Verwendung von ACLs zu verifizieren. Dabei kann, je nach Umfang des Tests, auch eine Software wie der Cisco Packet Tracer oder GNS 3 verwendet werden. Diese Software ermöglicht den Test von Konfigurationseinstellungen mit der aktuellen Firmware der Router.

Im weiteren sind die Sicherheitseinrichtungen hinsichtlich IPv6 zu untersuchen, um für den Fall der Einführung von IPv6 das Netzwerk für beide Protokolle zu sichern. Sind diese Punkte erledigt, kann mit der Einführung von IPv6 begonnen werden. Besteht keine dringende Notwendigkeit, die eine sofortige Aktivierung des neuen Internetprotokolls verlangt, ist es ratsam, die Migration vom Kern zu den Clients durchzuführen. So können die im Abschnitt 3.5.1 genannten Vorteile entsprechend genutzt werden. Da am Anfang hauptsächlich die Netzwerkhardware parallel IPv6 verwendet, ist es nebenbei möglich, durch Aufbau eines entsprechenden Testnetzwerks, die verwendete Software auf die Tauglichkeit für IPv6 und alle später zu verwendenden Funktionen hin zu überprüfen. Diese Vorgehensweise gewährleistet eine weitestgehend reibungslose Umstellung, bei der viele der auftretenden Probleme bereits vor der Einführung beim Anwender gelöst werden können.

6.3 Zusammenfassung

Im Großen und Ganzen ist ein umfangreiches Wissen notwendig, um eine Umstellung des Internetprotokolls erfolgreich vollziehen zu können. Zum einen sind Kenntnisse über IPv6 im Allgemeinen sowie zu Techniken der Migration sehr wichtig. Zum anderen ist das umzustellende Netzwerk mit all seinen Feinheiten und historischen Entwicklungen zu kennen, damit der richtige Weg der Migration gefunden werden kann.
Die Migration sollte daher nicht durch einen externen Dienstleister geplant und durchgeführt werden, sondern durch das Unternehmen bzw. die Mitarbeiter der IT-Abteilung selbst. Sie ist wie eine Betriebssystemumstellung nur in einem größeren Umfang einzuordnen. Es ist als ein Projekt über viele Monate hinweg zu sehen, welches nebenbei wachsen muss und nicht aus den Augen verloren werden darf, damit kein überstürztes Handeln notwendig ist. So kann parallel zum normalen Betrieb in kleinen Schritten IPv6 eingeführt werden. Der zusätzliche Arbeitsaufwand der Pflege zweier Internetprotokolle kommt in jedem Fall auf die Mitarbeiter zu. Bei früherer Planung eines solchen Projekts sind die Auswirkungen von Fehlern durch die Software oder in der Planung eher gering für den Rest des Netzwerks. Zudem lernen die Mitarbeiter in kleinen Schritten den Umgang mit IPv6. Demnach kann ein Testnetzwerk am Anfang eine gute Übungsumgebung darstellen, die später, bei entsprechenden IPv6-Kenntnissen, auch ruhigen Gewissens ins Produktivnetz verlagert werden kann. Das Testen einzelner IPv6 Software stellt dann nichts weiter dar, als die Überprüfung einer neuen Software, die IPv4 benutzt. Dies ist im laufenden Betrieb möglich.
Vorhandene Lösungen für Tunnel ins Unternehmensnetzwerk oder QoS sind bei der Mi-

gration ebenfalls zu betrachten, da diese durch in das Internetprotokoll 6 eingebaute Basisfunktionen übernommen werden können. So kann schon jetzt ein VPN-Zugang über IPv6 realisiert werden, ohne dass es einer zusätzlichen Software bedarf. Bedauerlicherweise ist die Konfiguration eines solchen Tunnels recht aufwändig und nicht sehr praktikabel, so dass eine VPN-Verbindung über einen Client, der mit einer Konfigurationsdatei versehen wird, aktuell einfacher einzusetzen ist. Mit einer größeren Zahl an Anwendern werden sich jedoch bald einfachere Konfigurationsmöglichkeiten ergeben, um einen VPN-Tunnel mit IPv6-Basisfunktionen zu realisieren. Auch bei QoS, dass bei IPv4 eher spärlich Verwendung findet, wird sich im Laufe der weiteren Verbreitung von IPv6 noch einiges verändern, so dass diese Funktionalität mehr genutzt werden kann.

6.4 Ausblick

Die letzten IPv4 Adressen sind aufgebraucht [Lück, 2011] und die Einführung von IPv6 geht schleppend voran. Die wenigsten Unternehmen im europäischen Raum bieten ihren Kunden einen IPv6-Internetzugang an. Bis zum Ende des Jahres 2011 möchte zumindest die Telekom ihren Kunden IPv6 anbieten [Hochstätter, 2010]. Damit wäre ein Anfang gemacht und andere ISPs werden diesem Vorbild folgen, sobald die Nachfrage besteht. Unternehmen, die Kunden einen Internetzugang bereitstellen und sich noch gar keine Gedanken zum Thema IPv6 gemacht haben, sollten dies so schnell wie möglich nachholen. Es führt kein Weg an IPv6 vorbei. Alle Unternehmen müssen sich damit befassen, egal ob Software oder Hardwarehersteller, ISP oder mittelständische Unternehmen. Sobald es eine sogenannte „Killer-Applikation“ gibt, die sehr viele Unternehmen verwenden, wird auch dem letzten Unternehmen klar, dass ein Umstieg unausweichlich ist. Dann wird es aber aufgrund des Zeitmangels vermehrt zu Problemen kommen und der Umstieg wird trotz der vielen Migrationsmechanismen mehr schlecht als recht voran gehen.

Anhang A

JPerf Messprotokolle

A.1 IPv4-only Messung

```
bin/iperf.exe -s -P 0 -i 1 -p 5001 -B w2008r2.local -V -f m
------------------------------------------------------------

Server listening on TCP port 5001
Binding to local address 192.158.0.2
TCP window size: 0.01 MByte (default)
------------------------------------------------------------

[288] local 192.158.0.2 port 5001 connected with
   192.198.0.2 port 49166
[ ID] Interval       Transfer     Bandwidth
[288]  0.0- 1.0 sec  8.35 MBytes  70.1 Mbits/sec
[288]  1.0- 2.0 sec  7.89 MBytes  66.2 Mbits/sec
[288]  2.0- 3.0 sec  8.25 MBytes  69.2 Mbits/sec
[288]  3.0- 4.0 sec  7.81 MBytes  65.5 Mbits/sec
[288]  4.0- 5.0 sec  7.98 MBytes  67.0 Mbits/sec
[288]  5.0- 6.0 sec  8.02 MBytes  67.3 Mbits/sec
[288]  6.0- 7.0 sec  8.09 MBytes  67.9 Mbits/sec
[288]  7.0- 8.0 sec  8.09 MBytes  67.8 Mbits/sec
[288]  8.0- 9.0 sec  8.12 MBytes  68.1 Mbits/sec
```

A.2 Dual-Stack Messung

```
bin/iperf.exe -s -P 0 -i 1 -p 5001 -B w2008r2.local -V -f M
------------------------------------------------------------

Server listening on TCP port 5001
Binding to local address 2a02:238:1:f00b:1001::2
```

```
TCP window size: 0.01 MByte (default)
------------------------------------------------------------

[288] local 2a02:238:1:f00b:1001::2 port 5001 connected
   with 2a02:238:1:f00b:c001::4 port 49159
[ ID] Interval       Transfer     Bandwidth
[288]  0.0- 1.0 sec  8.16 MBytes  68.4 Mbits/sec
[288]  1.0- 2.0 sec  8.09 MBytes  67.9 Mbits/sec
[288]  2.0- 3.0 sec  8.04 MBytes  67.4 Mbits/sec
[288]  3.0- 4.0 sec  8.08 MBytes  67.8 Mbits/sec
[288]  4.0- 5.0 sec  8.15 MBytes  68.4 Mbits/sec
[288]  5.0- 6.0 sec  8.16 MBytes  68.5 Mbits/sec
[288]  6.0- 7.0 sec  8.05 MBytes  67.5 Mbits/sec
[288]  7.0- 8.0 sec  8.34 MBytes  69.9 Mbits/sec
[288]  8.0- 9.0 sec  8.23 MBytes  69.1 Mbits/sec

bin/iperf.exe -s -P 0 -i 1 -p 5001 -B w2008r2.local -V -f m
------------------------------------------------------------

Server listening on TCP port 5001
Binding to local address 192.158.0.2
TCP window size: 0.01 MByte (default)
------------------------------------------------------------

[288] local 192.158.0.2 port 5001 connected with
   192.198.0.4 port 49159
[ ID] Interval       Transfer     Bandwidth
[288]  0.0- 1.0 sec  8.24 MBytes  69.2 Mbits/sec
[288]  1.0- 2.0 sec  8.29 MBytes  69.5 Mbits/sec
[288]  2.0- 3.0 sec  8.05 MBytes  67.5 Mbits/sec
[288]  3.0- 4.0 sec  8.18 MBytes  68.6 Mbits/sec
[288]  4.0- 5.0 sec  8.05 MBytes  67.5 Mbits/sec
[288]  5.0- 6.0 sec  8.26 MBytes  69.3 Mbits/sec
[288]  6.0- 7.0 sec  8.16 MBytes  68.4 Mbits/sec
[288]  7.0- 8.0 sec  8.23 MBytes  69.1 Mbits/sec
[288]  8.0- 9.0 sec  8.23 MBytes  69.1 Mbits/sec
```

A.3 GRE-Tunnel Messung

```
bin/iperf.exe -s -P 0 -i 1 -p 5001 -B w2008r2.local -V -f M
------------------------------------------------------------

Server listening on TCP port 5001
Binding to local address 2a02:238:1:f00b:1001::2
```

```
TCP window size: 0.01 MByte (default)
------------------------------------------------------------

[288] local 2a02:238:1:f00b:1001::2 port 5001 connected
   with 2a02:238:1:f00b:c001::2 port 49159
[ ID] Interval       Transfer     Bandwidth
[288]  0.0- 1.0 sec  0.99 MBytes  8.32 Mbits/sec
[288]  1.0- 2.0 sec  0.96 MBytes  8.06 Mbits/sec
[288]  2.0- 3.0 sec  0.96 MBytes  8.06 Mbits/sec
[288]  3.0- 4.0 sec  0.96 MBytes  8.06 Mbits/sec
[288]  4.0- 5.0 sec  0.97 MBytes  8.13 Mbits/sec
[288]  5.0- 6.0 sec  0.96 MBytes  8.06 Mbits/sec
[288]  6.0- 7.0 sec  0.98 MBytes  8.19 Mbits/sec
[288]  7.0- 8.0 sec  0.88 MBytes  7.34 Mbits/sec
[288]  8.0- 9.0 sec  0.95 MBytes  8.00 Mbits/sec
```

A.4 IPv6-only Messung

```
bin/iperf.exe -s -P 0 -i 1 -p 5001 -B w2008r2.local -V -f m
------------------------------------------------------------

Server listening on TCP port 5001
Binding to local address 2a02:238:1:f00b:1001::2
TCP window size: 0.01 MByte (default)
------------------------------------------------------------

[288] local 2a02:238:1:f00b:1001::2 port 5001 connected
   with 2a02:238:1:f00b:c001::3 port 49166
[ ID] Interval       Transfer     Bandwidth
[288]  0.0- 1.0 sec  7.47 MBytes  62.7 Mbits/sec
[288]  1.0- 2.0 sec  7.44 MBytes  62.4 Mbits/sec
[288]  2.0- 3.0 sec  7.44 MBytes  62.4 Mbits/sec
[288]  3.0- 4.0 sec  7.46 MBytes  62.6 Mbits/sec
[288]  4.0- 5.0 sec  7.69 MBytes  64.5 Mbits/sec
[288]  5.0- 6.0 sec  8.18 MBytes  68.6 Mbits/sec
[288]  6.0- 7.0 sec  8.15 MBytes  68.4 Mbits/sec
[288]  7.0- 8.0 sec  8.16 MBytes  68.4 Mbits/sec
[288]  8.0- 9.0 sec  8.29 MBytes  69.5 Mbits/sec
```

Anhang B

Routerkonfigurationen

B.1 Konfiguration Cisco 3750 - 1

```
!
version 12.2
no service pad
service timestamps debug datetime localtime
service timestamps log datetime localtime
service password-encryption
!
hostname CIS37_EU_H5
!
boot-start-marker
boot-end-marker
!
logging buffered 100000 informational
enable secret 5 *********.
!
username ******* privilege 15 secret 5 **********
username ******* privilege 15 password 7 **********
!
!
clock timezone MET 1
clock summer-time metdst recurring last Sun Mar 2:00 last
   Sun Oct 3:00
switch 1 provision ws-c3750-24ts
system mtu routing 1500
ip routing
!
!
ipv6 unicast-routing
!
```

```
!
spanning-tree mode pvst
no spanning-tree optimize bpdu transmission
spanning-tree extend system-id
!
vlan internal allocation policy ascending
!
ip ssh version 2
!
!
interface Tunnel0
 description 6inGRE Tunnel
 no ip address
 ipv6 address 2A02:238:1:F00B:100F::2/80
 ipv6 enable
 ipv6 ospf 2 area 0
 tunnel source 192.168.0.2
 tunnel destination 192.168.0.1
!
interface FastEthernet1/0/1
 no switchport
 ip address 192.178.0.1 255.255.255.0
 ipv6 address 2A02:238:1:F00B:3::1/80
 ipv6 ospf 2 area 0
 ipv6 traffic-filter Port5000 in
 ipv6 traffic-filter Port5000 out
!
interface FastEthernet1/0/2
 no switchport
 ip address 192.168.0.2 255.255.255.0
 ipv6 address 2A02:238:1:F00B:4::2/80
 ipv6 ospf 2 area 0
 ipv6 traffic-filter DenyIPv6 in
 ipv6 traffic-filter DenyIPv6 out
!
interface FastEthernet1/0/3
!
interface FastEthernet1/0/4
!
interface FastEthernet1/0/5
!
interface FastEthernet1/0/6
!
interface FastEthernet1/0/7
!
interface FastEthernet1/0/8
```

```
!
interface FastEthernet1/0/9
!
interface FastEthernet1/0/10
!
interface FastEthernet1/0/11
!
interface FastEthernet1/0/12
!
interface FastEthernet1/0/13
!
interface FastEthernet1/0/14
!
interface FastEthernet1/0/15
!
interface FastEthernet1/0/16
!
interface FastEthernet1/0/17
!
interface FastEthernet1/0/18
!
interface FastEthernet1/0/19
!
interface FastEthernet1/0/20
!
interface FastEthernet1/0/21
!
interface FastEthernet1/0/22
!
interface FastEthernet1/0/23
!
interface FastEthernet1/0/24
!
interface GigabitEthernet1/0/1
!
interface GigabitEthernet1/0/2
!
router ospf 1
 log-adjacency-changes
 network 192.168.0.0 0.0.0.255 area 0
 network 192.178.0.0 0.0.0.255 area 0
!
ip classless
no ip http server
ip http authentication local
ip http secure-server
```

```
!
logging 192.168.212.11
ipv6 router ospf 2
 log-adjacency-changes
!
!
ipv6 access-list Port5000
 deny tcp any any range 5000 6000 log
 permit icmp any any
 permit ipv6 any any
!
ipv6 access-list DenyIPv6
 sequence 20 deny ipv6 any any log
!
!
line con 0
 privilege level 15
 password 7 06031728
line vty 0 4
 privilege level 15
 password 7 03055F060F01
 transport input ssh
line vty 5 15
 privilege level 15
 transport input ssh
!
ntp clock-period 36029116
ntp server 172.17.56.95
end
```

B.2 Konfiguration Cisco 3750 - 2

```
!
version 12.2
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
!
hostname cis3750-mlr-t
!
boot-start-marker
boot-end-marker
!
```

```
username ****** privilege 15 secret 5 *******
username ****** privilege 15 secret 5 *******
!
!
aaa session-id common
switch 1 provision ws-c3750-24ts
system mtu routing 1500
vtp mode transparent
ip routing
ip name-server 8.8.8.8
!
!
ipv6 unicast-routing
!
!
spanning-tree mode pvst
spanning-tree extend system-id
!
vlan internal allocation policy ascending
!
!
ip ssh version 2
!
!
interface Tunnel0
 description 6inGRE Tunnel
 no ip address
 ipv6 address 2A02:238:1:F00B:100F::1/80
 ipv6 enable
 ipv6 ospf 2 area 0
 tunnel source 192.168.0.1
 tunnel destination 192.168.0.2
!
interface FastEthernet1/0/1
 no switchport
 ip address 192.158.0.1 255.255.255.0
 ipv6 address 2A02:238:1:F00B:1001::1/80
 ipv6 ospf 2 area 1
!
interface FastEthernet1/0/2
 no switchport
 ip address 192.188.0.2 255.255.255.0
 ipv6 address 2A02:238:1:F00B:2::2/80
 ipv6 ospf 2 area 0
!
interface FastEthernet1/0/3
```

```
!
interface FastEthernet1/0/4
!
interface FastEthernet1/0/5
!
interface FastEthernet1/0/6
!
interface FastEthernet1/0/7
!
interface FastEthernet1/0/8
!
interface FastEthernet1/0/9
!
interface FastEthernet1/0/10
!
interface FastEthernet1/0/11
!
interface FastEthernet1/0/12
!
interface FastEthernet1/0/13
!
interface FastEthernet1/0/14
!
interface FastEthernet1/0/15
!
interface FastEthernet1/0/16
!
interface FastEthernet1/0/17
!
interface FastEthernet1/0/18
!
interface FastEthernet1/0/19
!
interface FastEthernet1/0/20
!
interface FastEthernet1/0/21
!
interface FastEthernet1/0/22
!
interface FastEthernet1/0/23
!
interface FastEthernet1/0/24
 no switchport
 ip address 192.168.0.1 255.255.255.0
 ipv6 address 2A02:238:1:F00B:4::1/80
 ipv6 ospf 2 area 0
```

```
!
interface GigabitEthernet1/0/1
!
interface GigabitEthernet1/0/2
!
router ospf 1
 log-adjacency-changes
 network 192.158.0.0 0.0.0.255 area 1
 network 192.168.0.0 0.0.0.255 area 0
 network 192.188.0.0 0.0.0.255 area 0
!
ip classless
no ip http server
ip http secure-server
!
ip sla enable reaction-alerts
no cdp run
ipv6 router ospf 2
 log-adjacency-changes
!
!
line con 0
line vty 0 4
 privilege level 15
 transport input ssh
line vty 5 15
 privilege level 15
 transport input ssh
!
end
```

B.3 Konfiguration Cisco 1861

```
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname cis1860-mlr
!
boot-start-marker
boot-end-marker
!
```

```
logging message-counter syslog
!
...
!
aaa session-id common
network-clock-participate wic 1
mmi polling-interval 60
no mmi auto-configure
no mmi pvc
mmi snmp-timeout 180
!
!
dot11 syslog
ip source-route
ip cef
!
...
!
ip name-server 8.8.8.8
ipv6 unicast-routing
ipv6 cef
ntp server 172.17.56.95
!
!
voice-card 0
!
!
!
username ******* privilege 15 secret 5 ********
username ******* privilege 15 secret 5 ******
!
!
!
archive
 log config
  hidekeys
!
!
ip ssh version 2
!
!
interface Tunnel1
 no ip address
!
interface FastEthernet0/0
 ip address 212.122.42.133 255.255.255.248
```

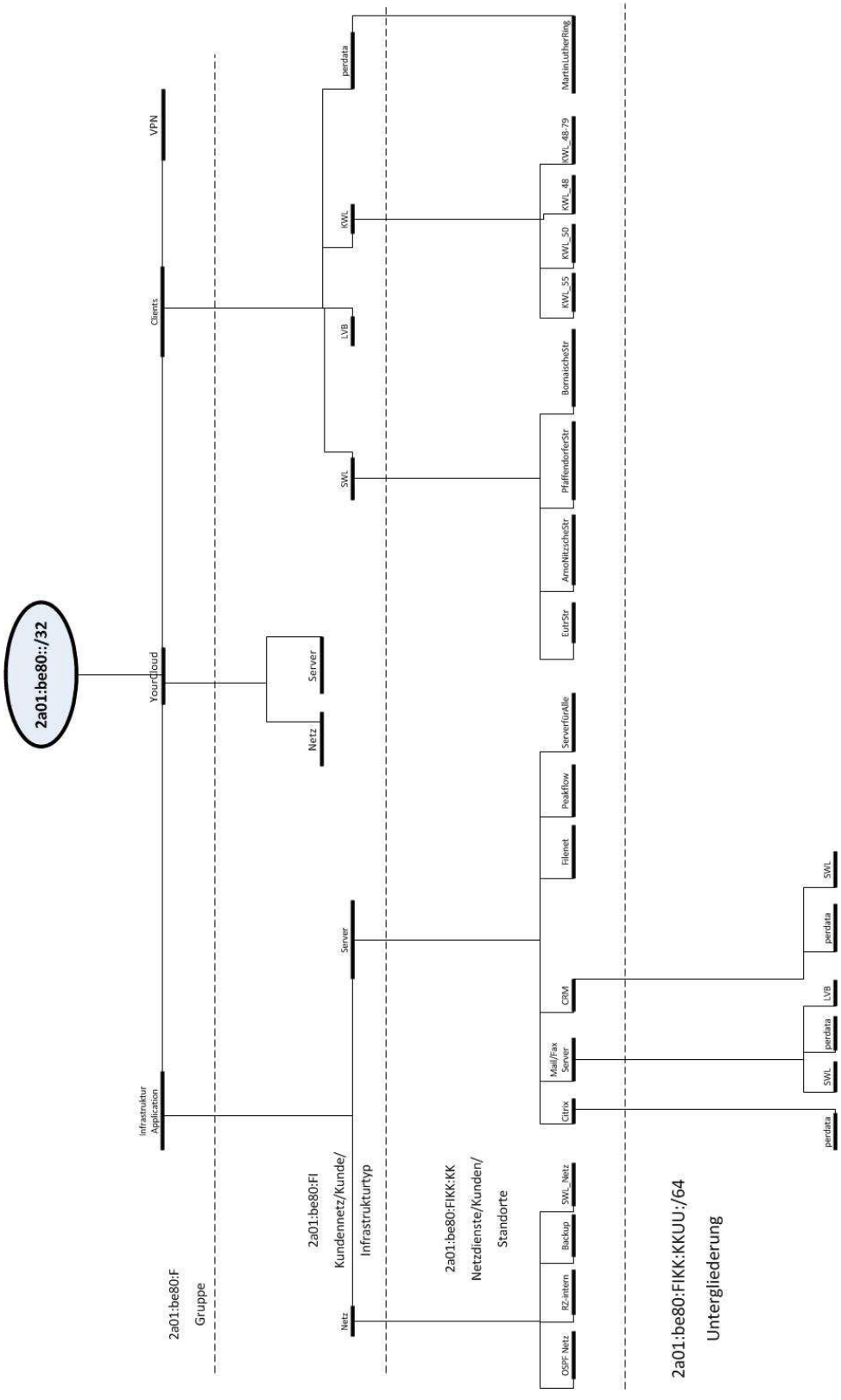
```
 duplex auto
 speed auto
 ipv6 address 2A02:238:1:F00B:1::1/80
!
interface FastEthernet0/1/0
 switchport access vlan 20
!
interface FastEthernet0/1/1
 switchport access vlan 30
!
interface FastEthernet0/1/2
 shutdown
!
interface FastEthernet0/1/3
 shutdown
!
interface FastEthernet0/1/4
 shutdown
!
interface FastEthernet0/1/5
 switchport access vlan 10
!
interface FastEthernet0/1/6
 switchport access vlan 10
!
interface FastEthernet0/1/7
 switchport access vlan 10
!
interface FastEthernet0/1/8
 shutdown
!
interface BRI0/1/0
 no ip address
!
interface BRI0/1/1
 no ip address
!
interface Vlan10
 ip address 192.198.0.1 255.255.255.0
 ipv6 address 2A02:238:1:F00B:C001::1/80
 ipv6 ospf 2 area 1
!
interface Vlan20
 ip address 192.188.0.1 255.255.255.0
 ipv6 address 2A02:238:1:F00B:2::1/80
 ipv6 ospf 2 area 0
```

```
!
interface Vlan30
 ip address 192.178.0.2 255.255.255.0
 ipv6 address 2A02:238:1:F00B:3::2/80
 ipv6 ospf 2 area 0
!
router ospf 1
 log-adjacency-changes
 network 192.178.0.0 0.0.0.255 area 0
 network 192.188.0.0 0.0.0.255 area 0
 network 192.198.0.0 0.0.0.255 area 1
 default-information originate always
!
ip forward-protocol nd
ip route 0.0.0.0 0.0.0.0 212.122.42.134
no ip http server
no ip http secure-server
!
!
!
ipv6 route ::/0 FastEthernet0/0 2A02:238:1:F00B::1
ipv6 router ospf 2
 router-id 186.0.0.0
 log-adjacency-changes
 default-information originate always
!
control-plane
!
!
!
voice-port 0/0/0
!
voice-port 0/0/1
!
voice-port 0/0/2
!
voice-port 0/0/3
!
voice-port 0/1/0
!
voice-port 0/1/1
!
voice-port 0/4/0
 auto-cut-through
 signal immediate
 input gain auto-control
```

```
 description Music On Hold Port
!
!
mgcp fax t38 ecm
mgcp behavior g729-variants static-pt
!
!
!
!
!
!
line con 0
 privilege level 15
 no modem enable
line aux 0
line vty 0 4
 privilege level 15
 password cisco
 transport input ssh
line vty 5 15
 privilege level 15
 transport input ssh
!
end
```

Anhang C

Adressierungsplan


2a01:be80::/32
YourCloud
Server
Netz
VPN
Clients
perdata
KWL
LVB
SWL
MartinLutherRing
KWL_48-79
KWL_48
KWL_50
KWL_55
BornaischeStr
PfaffendorferStr
AmoNitzschkeStr
EutrStr
ServerfürAlle
PeakFlow
Filenet
CRM
Mail/Fax Server
Citrix
Server
Infrastruktur Application
Netz
OSPF Netz
RZ-intern
Backup
SWL_Netz
2a01:be80:F
Gruppe
2a01:be80:FI
Kundennetz/Kunde/
Infrastrukturtyp
2a01:be80:FIKK:KK
Netzdienste/Kunden/
Standorte
SWL
perdata
LVB
perdata
SWL
perdata
2a01:be80:FIKK:KKUU:/64
Unterglliederung

Literaturverzeichnis

[A. Conta 2006] A. CONTA, M. Gupta-Ed. S. D. S. Deering: Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification / Internet Engineering Task Force. http://www.rfc-editor.org/rfc/rfc4443.txt, 03 2006 (4443). – RFC

[A. Conta 1998] A. CONTA, S. D.: Generic Packet Tunneling in IPv6 / Internet Engineering Task Force. Version: 12 1998. `http://www.rfc-editor.org/rfc/rfc2473.txt`. http://www.rfc-editor.org/rfc/rfc2473.txt, 12 1998 (2473). – RFC

[A. Matsumoto 2008a] A. MATSUMOTO, R. Hiromi-K. K. T. Fujisaki F. T. Fujisaki: Problem Statement for Default Address Selection in Multi-Prefix Environments: Operational Issues of RFC 3484 Default Rules / Internet Engineering Task Force. http://www.rfc-editor.org/rfc/rfc5220.txt, 07 2008 (5220). – RFC

[A. Matsumoto 2008b] A. MATSUMOTO, R. Hiromi-K. K. T. Fujisaki F. T. Fujisaki: Requirements for Address Selection Mechanisms / Internet Engineering Task Force. http://www.rfc-editor.org/rfc/rfc5221.txt, 07 2008 (5221). – RFC

[B. Cain 2002] B. CAIN, I. Kouvelas-B. Fenner A. T. S. Deering D. S. Deering: Internet Group Management Protocol, Version 3 / Internet Engineering Task Force. http://www.rfc-editor.org/rfc/rfc3376.txt, 10 2002 (3376). – RFC

[B. Carpenter 2001] B. CARPENTER, K. M.: Connection of IPv6 Domains via IPv4 Clouds / Internet Engineering Task Force. http://www.rfc-editor.org/rfc/rfc3056.txt, 02 2001 (3056). – RFC

[B. Haberman 2002] B. HABERMAN, D. T.: Unicast-Prefix-based IPv6 Multicast Addresses / Internet Engineering Task Force. http://www.rfc-editor.org/rfc/rfc3306.txt, 08 2002 (3303). – RFC

[B. Haberman 2005] B. HABERMAN, J. M.: Multicast Router Discovery / Internet Engineering Task Force. http://www.rfc-editor.org/rfc/rfc4286.txt, 12 2005 (4286). – RFC

[C. Aoun 2007] C. AOUN, E. D.: Reasons to Move the Network Address Translator - Protocol Translator (NAT-PT) to Historic Status / Internet Engineering Task Force. http://www.rfc-editor.org/rfc/rfc4966.txt, 07 2007 (4966). – RFC

[C. Kaufman 2005] C. KAUFMAN, Ed.: Internet Key Exchange (IKEv2) Protocol / Internet Engineering Task Force. http://www.rfc-editor.org/rfc/rfc4306.txt, 12 2005 (4306). – RFC

[C. Partridge 1999] C. PARTRIDGE, A. J.: IPv6 Router Alert Option / Internet Engineering Task Force. Version: 10 1999. `http://www.rfc-editor.org/rfc/rfc2711.txt`. http://www.rfc-editor.org/rfc/rfc2711.txt, 10 1999 (2711). – RFC

[C. Partridge 1993] C. PARTRIDGE, W. M. T. Mendez M. T. Mendez: Host Anycasting Service / Internet Engineering Task Force. Version: 11 1993. `http://www.rfc-editor.org/rfc/rfc1546.txt`. http://www.rfc-editor.org/rfc/rfc1546.txt, 11 1993 (1546). – RFC

[Chown 2006] CHOWN, T.: Use of VLANs for IPv4-IPv6 Coexistence in Enterprise Networks / Internet Engineering Task Force. http://www.rfc-editor.org/rfc/rfc4554.txt, 06 2006 (4554). – RFC

[Conta 2001] CONTA, A.: Extensions to IPv6 Neighbor Discovery for Inverse Discovery Specification / Internet Engineering Task Force. http://www.rfc-editor.org/rfc/rfc3122.txt, 06 2001 (3122). – RFC

[Crawford 1998] CRAWFORD, M.: Transmission of IPv6 Packets over Ethernet Networks / Internet Engineering Task Force. http://www.rfc-editor.org/rfc/rfc2464.txt, 12 1998 (2464). – RFC

[D. Borman 1999] D. BORMAN, R. H. S. Deering D. S. Deering: IPv6 Jumbograms / Internet Engineering Task Force. Version: 8 1999. `http://www.rfc-editor.org/rfc/rfc2675.txt`. http://www.rfc-editor.org/rfc/rfc2675txt, 8 1999 (2675). – RFC

[D. Farinacci 2000] D. FARINACCI, S. Hanks-D. Meyer P. T. T. Li L. T. Li: Generic Routing Encapsulation (GRE) / Internet Engineering Task Force. http://www.rfc-editor.org/rfc/rfc2784.txt, 03 2000 (2784). – RFC

[D. Harkins 1998] D. HARKINS, D. C.: The Internet Key Exchange (IKE) / Internet Engineering Task Force. http://www.rfc-editor.org/rfc/rfc2409.txt, 11 1998 (2409). – RFC

[D. Johnson 1999] D. JOHNSON, S. D.: Reserved IPv6 Subnet Anycast Addresses / Internet Engineering Task Force. http://www.rfc-editor.org/rfc/rfc2526.txt, 03 1999 (2526). – RFC

[D. Maughan 1998] D. MAUGHAN, M. Schneider-J. T. M. Schertler S. M. Schertler: Internet Security Association and Key Management Protocol (ISAKMP) / Internet Engineering Task Force. http://www.rfc-editor.org/rfc/rfc2408.txt, 11 1998 (2408). – RFC

[Draves 2003] DRAVES, R.: Default Address Selection for Internet Protocol version 6 (IPv6) / Internet Engineering Task Force. http://www.rfc-editor.org/rfc/rfc3484.txt, 02 2003 (3484). – RFC

[E. Davies 2007] E. DAVIES, J. M.: Recommendations for Filtering ICMPv6 Messages in Firewalls / Internet Engineering Task Force. http://www.rfc-editor.org/rfc/rfc4890.txt, 05 2007 (4890). – RFC

[E. Nordmark 2005] E. NORDMARK, R. G.: Basic Transition Mechanisms for IPv6 Hosts and Routers / Internet Engineering Task Force. http://www.rfc-editor.org/rfc/rfc4213.txt, 10 2005 (4213). – RFC

[F. Baker 2001] F. BAKER, F. Le Faucheur B. D. C. Iturralde I. C. Iturralde: Aggregation of RSVP for IPv4 and IPv6 Reservations / Internet Engineering Task Force. Version: 09 2001. `http://www.rfc-editor.org/rfc/rfc3175.txt`. http://www.rfc-editor.org/rfc/rfc3175.txt, 09 2001 (3175). – RFC

[F. Baker 2005] F. BAKER, R. D. E. Lear L. E. Lear: Procedures for Renumbering an IPv6 Network without a Flag Day / Internet Engineering Task Force. http://www.rfc-editor.org/rfc/rfc4192.txt, 09 2005 (4192). – RFC

[F. Templin 2008] F. TEMPLIN, D. T. T. Gleeson G. T. Gleeson: Intra-Site Automatic Tunnel Addressing Protocol (ISATAP) / Internet Engineering Task Force. http://www.rfc-editor.org/rfc/rfc5214.txt, 03 2008 (5214). – RFC

[Fenner 1997] FENNER, W.: Internet Group Management Protocol, Version 2 / Internet Engineering Task Force. http://www.rfc-editor.org/rfc/rfc2236.txt, 11 1997 (2236). – RFC

[G. Tsirtsis 2000] G. TSIRTSIS, P. S.: Network Address Translation - Protocol Translation (NAT-PT) / Internet Engineering Task Force. http://www.rfc-editor.org/rfc/rfc2766.txt, 02 2000 (2766). – RFC

[Haberman 2002] HABERMAN, B.: Allocation Guidelines for IPv6 Multicast Addresses / Internet Engineering Task Force. http://www.rfc-editor.org/rfc/rfc3307.txt, 08 2002 (3307). – RFC

[Hagen 2009] HAGEN, Silvia: *IPv6: Grundlagen - Funktionalität - Integration*. Sunny Connection, 2009

[Hochstätter 2010] HOCHSTÄTTER, Christoph H.: *Deutsche Telekom bietet IPv6 für Privatkunden ab Ende 2011*. `http://www.zdnet.de/news/41538861/deutsche-telekom-bietet-ipv6-fuer-privatkunden-ab-ende-2011.htm`. Version: 2010, Abruf: 03.09.2011. – ZD Net-News

[Hoffman 2005] HOFFMAN, P.: Algorithms for Internet Key Exchange version 1 (IKEv1) / Internet Engineering Task Force. http://www.rfc-editor.org/rfc/rfc4109.txt, 05 2005 (4109). – RFC

[Huitema 2006] HUITEMA, C.: Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs) / Internet Engineering Task Force. http://www.rfc-editor.org/rfc/rfc4380.txt, 02 2006 (4380). – RFC

[IEEE] IEEE: Guidelines for 64-bit Global Identifier (EUI-64) Registration Authority / Institute of Electrical and Electronics Engineers. http://standards.ieee.org/develop/regauth/tut/eui64.pdf, . – Guideline

[IESG 2001] IESG, IAB: IAB/IESG Recommendations on IPv6 Address Allocations to Sites / Internet Engineering Task Force. http://www.rfc-editor.org/rfc/rfc3177.txt, 09 2001 (3177). – RFC

[for IPv6 2004] IPV6, Multicast Listener Discovery Version 2 (.: R. Vida, Ed., L. Costa, Ed. / Internet Engineering Task Force. http://www.rfc-editor.org/rfc/rfc3810.txt, 06 2004 (3810). – RFC

[J. Abley 2007] J. ABLEY, G. Neville-Neil P. S. P. Savola: Deprecation of Type 0 Routing Headers in IPv6 / Internet Engineering Task Force. Version: 12 2007. `http://www.rfc-editor.org/rfc/rfc5095.txt`. http://www.rfc-editor.org/rfc/rfc5095.txt, 12 2007 (5095). – RFC

[J. Arkko 2005] J. ARKKO, J. Kempf B. Zill P. N. Ed.: SEcure Neighbor Discovery (SEND) / Internet Engineering Task Force. http://www.rfc-editor.org/rfc/rfc3971.txt, 03 2005 (3971). – RFC

[J. McCann 1996] J. MCCANN, J. M. S. Deering D. S. Deering: Path MTU Discovery for IP version 6 / Internet Engineering Task Force. http://www.rfc-editor.org/rfc/rfc1981.txt, 08 1996 (1981). – RFC

[K. Nichols 1998] K. NICHOLS, F. Baker D. B. S. Blake B. S. Blake: Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers / Internet Engineering Task Force. Version: 12 1998. `http://www.rfc-editor.org/rfc/rfc2474.txt`. http://www.rfc-editor.org/rfc/rfc2474.txt, 12 1998 (2474). – RFC

[Kent 2005a] KENT, S.: IP Authentication Header / Internet Engineering Task Force. Version: 12 2005. `http://www.rfc-editor.org/rfc/rfc4302.txt`. http://www.rfc-editor.org/rfc/rfc4302.txt, 12 2005 (4302). – RFC

[Kent 2005b] KENT, S.: IP Encapsulating Security Payload (ESP) / Internet Engineering Task Force. Version: 12 2005. `http://www.rfc-editor.org/rfc/rfc4303.txt`. http://www.rfc-editor.org/rfc/rfc4303.txt, 12 2005 (4303). – RFC

[Krawczyk 1996] KRAWCZYK, Hugo: SKEME: A Versatile Secure Key Exchange Mechanism for Internet / IEEE. 1996. – Forschungsbericht

[L. Delgrossi 1995] L. DELGROSSI, L. B.: Internet Stream Protocol Version 2 (ST2) / Internet Engineering Task Force. Version: 10 1995. `http://www.rfc-editor.org/rfc/rfc1819.txt`. http://www.rfc-editor.org/rfc/rfc1819.txt, 10 1995 (1819). – RFC

[Lück 2011] LÜCK, Folker: *Letzte IPv4-Adressblöcke verteilt - IPv6-Kongress 2011 in Frankfurt*. `http://www.crn.de/netzwerke-tk/artikel-89649.html`. Version: 2011, Abruf: 06.04.2011. – Computer Reseller News

[M. Allman 1998] M. ALLMAN, C. M. S. Ostermann O. S. Ostermann: FTP Extensions for IPv6 and NATs / Internet Engineering Task Force. http://www.rfc-editor.org/rfc/rfc2428.txt, 09 1998 (2428). – RFC

[Moy 1998] MOY, J.: OSPF Version 2 / Internet Engineering Task Force. http://www.rfc-editor.org/rfc/rfc2328.txt, 04 1998 (2328). – RFC

[P. Marques 1999] P. MARQUES, F. D.: Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing / Internet Engineering Task Force. http://www.rfc-editor.org/rfc/rfc2545.txt, 03 1999 (2545). – RFC

[P. Nikander 2004] P. NIKANDER, J. Kempf E. N. Ed.: IPv6 Neighbor Discovery (ND) Trust Models and Threats / Internet Engineering Task Force. http://www.rfc-editor.org/rfc/rfc3756.txt, 05 2004 (3756). – RFC

[P. Savola 2004] P. SAVOLA, B. H.: Embedding the Rendezvous Point (RP) Address in an IPv6 Multicast Address / Internet Engineering Task Force. http://www.rfc-editor.org/rfc/rfc3956.txt, 11 2004 (3956). – RFC

[R. Bonica 2007] R. BONICA, D. Tappan C. P. D. Gan G. D. Gan: Extended ICMP to Support Multi-Part Messages / Internet Engineering Task Force. http://www.rfc-editor.org/rfc/rfc4884.txt, 04 2007 (4884). – RFC

[R. Coltun 2008] R. COLTUN, J. Moy A. Lindem E. D. Ferguson F. D. Ferguson: OSPF for IPv6 / Internet Engineering Task Force. http://www.rfc-editor.org/rfc/rfc5340.txt, 07 2008 (5340). – RFC

[R. Droms 2003] R. DROMS, J. Bound B. Volz T. Lemon C. Perkins M. C. Ed.: Dynamic Host Configuration Protocol for IPv6 (DHCPv6) / Internet Engineering Task Force. http://www.rfc-editor.org/rfc/rfc3315.txt, 07 2003 (3315). – RFC

[R. Graveman 2007] R. GRAVEMAN, P. Savola H. T. M. Parthasarathy P. M. Parthasarathy: Using IPsec to Secure IPv6-in-IPv4 Tunnels / Internet Engineering Task Force. http://www.rfc-editor.org/rfc/rfc4891.txt, 05 2007 (4891). – RFC

[R. Hinden 2005] R. HINDEN, B. H.: Unique Local IPv6 Unicast Addresses / Internet Engineering Task Force. http://www.rfc-editor.org/rfc/rfc4193.txt, 10 2005 (4193). – RFC

[R. Hinden 2003] R. HINDEN, E. N. S. Deering D. S. Deering: IPv6 Global Unicast Address Format / Internet Engineering Task Force. http://www.rfc-editor.org/rfc/rfc3587.txt, 08 2003 (3587). – RFC

[R. Hinden 2006] R. HINDEN, S. D.: IP Version 6 Addressing Architecture / Internet Engineering Task Force. Version: 02 2006. `http://www.rfc-editor.org/rfc/rfc4291.txt`. http://www.rfc-editor.org/rfc/rfc4291.txt, 02 2006 (4291). – RFC

[S. Bhattacharyya 2003] S. BHATTACHARYYA, Ed.: An Overview of Source-Specific Multicast (SSM) / Internet Engineering Task Force. http://www.rfc-editor.org/rfc/rfc3569.txt, 07 2003 (3569). – RFC

[S. Blake 1998] S. BLAKE, M. Carlson E. Davies Z. Wang W. W. D. Black B. D. Black: An Architecture for Differentiated Services / Internet Engineering Task Force. Version: 12 1998. `http://www.rfc-editor.org/rfc/rfc2475.txt`. http://www.rfc-editor.org/rfc/rfc2475.txt, 12 1998 (2475). – RFC

[S. Deering 1999] S. DEERING, B. H. W. Fenner F. W. Fenner: Multicast Listener Discovery (MLD) for IPv6 / Internet Engineering Task Force. http://www.rfc-editor.org/rfc/rfc2710.txt, 10 1999 (2710). – RFC

[S. Deering 1998] S. DEERING, R. H.: Internet Protocol, Version 6 (IPv6) / Internet Engineering Task Force. Version: 12 1998. `http://www.rfc-editor.org/rfc/rfc2460.txt`. http://www.rfc-editor.org/rfc/rfc2460.txt, 12 1998 (2460). – RFC

[S. Kent 2005] S. KENT, K. S.: Security Architecture for the Internet Protocol / Internet Engineering Task Force. http://www.rfc-editor.org/rfc/rfc4301.txt, 12 2005 (4301). – RFC

[S. Thomson 2007] S. THOMSON, T. J. T. Narten N. T. Narten: IPv6 Stateless Address Autoconfiguration / Internet Engineering Task Force. http://www.rfc-editor.org/rfc/rfc4862.txt, 09 2007 (4862). – RFC

[Sack 2011] SACK, Harald: *IPv6 jetzt*. `http://www.ipv6council.de/documents/presentations/ipv6_jetzt_bmwi_workshop_08_feb_2011.html?L=1`. Version: 2011, Abruf: 04.04.2011

[Schaub 2011] SCHAUB, Markus: IPv6 - Wenn der Dual-Stack nicht weiter hilft. In: *Der Netzwerk Insider* (2011)

[Schiller 2005] SCHILLER, J.: Cryptographic Algorithms for Use in the Internet Key Exchange Version 2 (IKEv2) / Internet Engineering Task Force. http://www.rfc-editor.org/rfc/rfc4307.txt, 12 2005 (4307). – RFC

[Seeber 2011] SEEBER, Sebastian: Migration von IPv4 zu IPv6 in einem Unternehmen der IT-Branche / Hochschule Mittweida - University of Applied Siences. 2011. – Forschungsbericht

[Steffann 2011] STEFFANN, Sander: *Preparing an IPv6 Adressing Plan - Manual*. http://labs.ripe.net/Members/steffann/preparing-an-ipv6-addressing-plan: SURFnet, 03 2011

[T. Bates 2007] T. BATES, D. Katz Y. R. R. Chandra C. R. Chandra: Multiprotocol Extensions for BGP-4 / Internet Engineering Task Force. http://www.rfc-editor.org/rfc/rfc4760.txt, 01 2007 (4760). – RFC

[T. Bates 2000] T. BATES, R. Chandra D. K. Y. Rekhter R. Y. Rekhter: Multiprotocol Extensions for BGP-4 / Internet Engineering Task Force. http://www.rfc-editor.org/rfc/rfc2858.txt, 06 2000 (2858). – RFC

[T. Narten 2011] T. NARTEN, L. R. G. Huston H. G. Huston: IPv6 Address Assignment to End Sites / Internet Engineering Task Force. http://www.rfc-editor.org/rfc/rfc6177.txt, 03 2011 (6177). – RFC

[T. Narten 2007a] T. NARTEN, S. K. R. Draves D. R. Draves: Privacy Extensions for Stateless Address Autoconfiguration in IPv6 / Internet Engineering Task Force. http://www.rfc-editor.org/rfc/rfc4941.txt, 09 2007 (4941). – RFC

[T. Narten 2007b] T. NARTEN, W. Simpson H. S. E. Nordmark N. E. Nordmark: Neighbor Discovery for IP version 6 (IPv6) / Internet Engineering Task Force. http://www.rfc-editor.org/rfc/rfc4861.txt, 09 2007 (4861). – RFC

[G. Van de Velde 2007] VELDE, R. Droms B. Carpenter E. K. T. Hain d. T. Hain H. T. Hain: Local Network Protection for IPv6 / Internet Engineering Task Force. http://www.rfc-editor.org/rfc/rfc4864.txt, 05 2007 (4864). – RFC

[Y. Rekhter 1996] Y. REKHTER, D. Karrenberg G. J. de Groot E. L. B. Moskowitz M. B. Moskowitz: Address Allocation for Private Internets / Internet Engineering Task Force. http://www.rfc-editor.org/rfc/rfc1918.txt, 02 1996 (1918). – RFC

[Y. Rekhter 2006] Y. REKHTER, T. Li Ed. S. Hares E. Ed.: A Border Gateway Protocol 4 (BGP-4) / Internet Engineering Task Force. http://www.rfc-editor.org/rfc/rfc4271.txt, 01 2006 (4271). – RFC

Selbstständigkeitserklärung

Ich erkläre, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Mittweida, den 27. Dezember 2011

Sebastian Seeber