

Webcode - Übungsdateien

060A-332D-D5E8

Berufsbildende Schule I Mainz



In Kooperation mit dem HERDT-Verlag stellen wir Ihnen eine PDF inkl. Zusatzmedien für Ihre persönliche Weiterbildung zur Verfügung. In Verbindung mit dem Programm HERDT|Campus ALL YOU CAN READ stehen diese PDFs nur Lehrkräften und Schüler\*innen der oben genannten Lehranstalt zur Verfügung. Eine Nutzung oder Weitergabe für andere Zwecke ist ausdrücklich verboten und unterliegt dem Urheberrecht. Jeglicher Verstoß kann zivil- und strafrechtliche Konsequenzen nach sich ziehen.

## Netzwerke

---

IPv6

**Internet Protocol Version 6**

NWIPV6



Joachim Kohlmorgen, Klaus Fichtner, Daniel  
Hemmling, Andre Liesenfeld, Heinz Erich Lutz,  
Ralf Pohlmann, Mathias Schulze

3. Ausgabe, Oktober 2017

ISBN 978-3-86249-749-2

# Impressum

Matchcode: NWIPV6

Autoren: Joachim Kohlmorgen, Klaus Fichtner, Daniel Hemmling, Andre Liesenfeld, Heinz Erich Lutz,  
Ralf Pohlmann, Mathias Schulze

Produziert im HERDT-Digitaldruck

3. Ausgabe, Oktober 2017

HERDT-Verlag für Bildungsmedien GmbH  
Am Kümmerling 21-25  
55294 Bodenheim  
Internet: [www.herdt.com](http://www.herdt.com)  
E-Mail: [info@herdt.com](mailto:info@herdt.com)

© HERDT-Verlag für Bildungsmedien GmbH, Bodenheim

Alle Rechte vorbehalten. Kein Teil des Werkes darf in irgendeiner Form (Druck, Fotokopie, Mikrofilm oder einem anderen Verfahren) ohne schriftliche Genehmigung des Verlags reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

Dieses Buch wurde mit großer Sorgfalt erstellt und geprüft. Trotzdem können Fehler nicht vollkommen ausgeschlossen werden. Verlag, Herausgeber und Autoren können für fehlerhafte Angaben und deren Folgen weder eine juristische Verantwortung noch irgendeine Haftung übernehmen.

Wenn nicht explizit an anderer Stelle des Werkes aufgeführt, liegen die Copyrights an allen Screenshots beim HERDT-Verlag. Sollte es trotz intensiver Recherche nicht gelungen sein, alle weiteren Rechteinhaber der verwendeten Quellen und Abbildungen zu finden, bitten wir um kurze Nachricht an die Redaktion.

Die in diesem Buch und in den abgebildeten bzw. zum Download angebotenen Dateien genannten Personen und Organisationen, Adress- und Telekommunikationsangaben, Bankverbindungen etc. sind frei erfunden. Eventuelle Übereinstimmungen oder Ähnlichkeiten sind unbeabsichtigt und rein zufällig.

Die Bildungsmedien des HERDT-Verlags enthalten Verweise auf Webseiten Dritter. Diese Webseiten unterliegen der Haftung der jeweiligen Betreiber, wir haben keinerlei Einfluss auf die Gestaltung und die Inhalte dieser Webseiten. Bei der Bucherstellung haben wir die fremden Inhalte daraufhin überprüft, ob etwaige Rechtsverstöße bestehen. Zu diesem Zeitpunkt waren keine Rechtsverstöße ersichtlich. Wir werden bei Kenntnis von Rechtsverstößen jedoch umgehend die entsprechenden Internetadressen aus dem Buch entfernen.

Die in den Bildungsmedien des HERDT-Verlags vorhandenen Internetadressen, Screenshots, Bezeichnungen bzw. Beschreibungen und Funktionen waren zum Zeitpunkt der Erstellung der jeweiligen Produkte aktuell und gültig. Sollten Sie die Webseiten nicht mehr unter den angegebenen Adressen finden, sind diese eventuell inzwischen komplett aus dem Internet genommen worden oder unter einer neuen Adresse zu finden. Sollten im vorliegenden Produkt vorhandene Screenshots, Bezeichnungen bzw. Beschreibungen und Funktionen nicht mehr der beschriebenen Software entsprechen, hat der Hersteller der jeweiligen Software nach Drucklegung Änderungen vorgenommen oder vorhandene Funktionen geändert oder entfernt.

<b>1 Informationen zu diesem Buch</b>	<b>4</b>	<b>7 Routing</b>	<b>82</b>
1.1 Voraussetzungen und Ziele	4	7.1 Grundlagen zu Routing	82
1.2 Aufbau und Konventionen	4	7.2 Netzwerkmodelle und Topologien	83
		7.3 Autonome Systeme	86
		7.4 Routing-Algorithmen	88
<b>2 Das IPv4-Protokoll und seine Grenzen</b>	<b>6</b>	7.5 Protokoll-Algorithmen	89
2.1 Internet und Internet-Protokoll	6	7.6 Statisches und dynamisches Routing	92
2.2 Überblick zur Entwicklung von TCP/IP	7	7.7 Konfiguration statischer Routen – MS Server Manager	94
2.3 IP im Kontext des TCP/IP-Modells	9		
2.4 IPv4-Header	11	7.8 Konfiguration statischer Routen – <code>netsh</code>	97
2.5 ARP und ICMP	13	7.9 Dynamisches Routing	99
2.6 Aufbau der IPv4-Adresse	16	7.10 RIPng	100
2.7 Netzwerkkennung (Subnetz-ID, Netz-ID), Hostadresse und Broadcast-Adresse	17	7.11 OSPFv3	101
2.8 Netzwerkklassen	21	7.12 IDRPv2	103
2.9 IPv4: Adressraumknappheit bei IPv4 und Lösungskonzepte	23		
<b>3 Eigenschaften des IPv6-Protokolls</b>	<b>28</b>	<b>8 Übergangsmechanismen</b>	<b>106</b>
3.1 Neuerungen im IPv6-Protokoll	28	8.1 Voraussetzungen für Migration auf IPv6	106
3.2 Gegenüberstellung von IPv4 und IPv6	31	8.2 Dual-Stack	106
3.3 Vergleich der Header IPv4 und IPv6	31	8.3 Dual-Stack Lite (DS-Lite)	107
		8.4 Tunnelmechanismen	109
		8.5 Übersetzungsverfahren	121
<b>4 Aufbau des Adressraums von IPv6</b>	<b>34</b>	<b>9 Sicherheit und IPv6</b>	<b>122</b>
4.1 Der Adressaufbau von IPv6	34	9.1 Sicherheitsprobleme und Probleme bei IPv6	122
4.2 Adressnotation	34	9.2 Die Privatsphäre	123
4.3 MAC-Adresse, EUI-64 und Interface ID	36	9.3 Eingebaute Sicherheit	126
4.4 IPv6-Adresszuweisung durch Regional Internet Registry	38	9.4 Wegfall von NAT	128
4.5 Addressbereiche	40	9.5 Sicherer DHCPv6	130
4.6 Allgemeine Addressbereiche	41	9.6 Bedrohungsszenarien	133
4.7 Besondere Addressbereiche	46	9.7 Zusammenfassung	135
4.8 Subnetting	48		
<b>5 Konfiguration</b>	<b>52</b>	<b>10 Mobile IPv6 und Migration</b>	<b>136</b>
5.1 Möglichkeiten der Konfiguration	52	10.1 Mobile IPv6	136
5.2 Stateless Address Autoconfiguration (SLAAC)	53	10.2 Anmerkungen zu Migrationsszenarien	141
5.3 Stateful Autoconfiguration	55		
5.4 Grundlagen zum DHCPv6-Protokoll	55	<b>Anhang: Testumgebung</b>	<b>144</b>
5.5 DHCP-Client/Server-Kommunikation	57	A.1 Die Testumgebung	144
5.6 IPv6 und DNS	58	A.2 Hyper-V aktivieren	144
5.7 Praxisbeispiel Stateless Autoconfiguration	61	A.3 Netzwerke einrichten mit Hyper-V-Netzwerkmanager	145
5.8 Praxisbeispiel Stateless Autoconfiguration mit DHCPv6 (Stateless DHCPv6)	67	A.4 Virtuelle Maschinen konfigurieren	146
5.9 Praxisbeispiel Konfiguration des DHCPv6-Servers unter MS Windows Server 2016	69	A.5 Standard-Tools	147
		<b>Stichwortverzeichnis</b>	<b>158</b>
<b>6 Netzwerkkontexte</b>	<b>72</b>		
6.1 ICMPv6	72		
6.2 Maximum Transmission Unit (MTU)	76		
6.3 IP over Everything	78		

# 1 Informationen zu diesem Buch

## In diesem Kapitel erfahren Sie

- ✓ wie Sie dieses Buch einsetzen können
- ✓ welche Vorkenntnisse Sie mitbringen sollten

## 1.1 Voraussetzungen und Ziele

### Zielgruppe

Dieses Buch richtet sich in erster Linie an (zukünftige) Systembetreuer, Administratoren und Netzwerkplaner, die das notwendige Wissen für den Umstieg von IPv4 auf IPv6 erwerben und das neue Protokoll verstehen wollen. Die Leser und Kursteilnehmer können neu in das Internetprotokoll IPv6 einsteigen oder bereits gesammelte Erfahrungen erweitern.

### Empfohlene Vorkenntnisse

Um sich problemlos die Kenntnisse von IPv6 aneignen zu können, sollten Sie bereits über folgende Kenntnisse verfügen:

- ✓ Grundkenntnisse im Umgang mit PC-Systemen
- ✓ Grundkenntnisse in der Systemadministration
- ✓ Grundkenntnisse in der Netzwerkadministration
- ✓ Kenntnisse im Bereich der IP-Adressierung und in der Vergabe von IP-Adressen im Netzwerk

### Lernziele

Nach Durcharbeiten dieses Buches haben Sie grundlegende Kenntnisse zum IPv6-Protokoll, die Sie in die Lage versetzen, das Protokoll zu implementieren und zu konfigurieren. Sie kennen die wesentlichen Unterschiede zwischen IPv4 und IPv6 und können diese beschreiben und bewerten.

### Hinweise zu Soft- und Hardware

Die verwendete Testumgebung arbeitet mit virtuellen Maschinen, sodass ein ausreichend dimensionierter PC zur Verfügung stehen sollte. Anstelle einer Virtualisierung kann auch mit mehreren PCs gearbeitet werden. Eine detailliertere Beschreibung der verwendeten Testumgebung steht im Anhang zur Verfügung.

## 1.2 Aufbau und Konventionen

### Aufbau der Unterlage

- ✓ Am Anfang jedes Kapitels finden Sie die Lernziele und Voraussetzungen.
- ✓ Ein Thema wird jeweils in einem Kapitel theoretisch behandelt. Die passenden Übungsaufgaben können Sie downloaden, siehe Erläuterungen im weiteren Verlauf dieses Abschnitts.

## Inhaltliche Gliederung

Zu Beginn wird das aktuelle Protokoll IPv4 betrachtet. Die dort vermittelten Grundlagen sind für das Verständnis der folgenden Kapitel wichtig. Im nächsten Schritt werden Eigenschaften und Adressaufbau von IPv6 erläutert und im Anschluss mit Praxisbeispielen untermauert. Ein theoretischer Einschub beschäftigt sich mit den IPv6-Fähigkeiten verschiedener Übertragungsmedien. Anschließend werden Grundlagen zum Routing und die Anpassungen der diversen Routing-Protokolle auf IPv6 gezeigt. Im Anschluss werden Übergangstechniken vorgestellt und an Praxisbeispielen erläutert. Zum Thema Sicherheit wird gezeigt, was sich in Bezug auf IPv6 geändert hat. Das Buch schließt mit Überlegungen zur Migration auf IPv6 und stellt einige neue Möglichkeiten des Protokolls vor.

## Typografische Konventionen

Im Text erkennen Sie bestimmte Programmelemente an der Formatierung:

**Kursivschrift** kennzeichnet alle von Programmen vorgegebenen Bezeichnungen für Schaltflächen, Dialogfenster, Symbolleisten, Menüs bzw. Menüpunkte, Datei- und Verzeichnisnamen sowie Internetadressen.

**Courier** wird für Systembefehle verwendet.

In Syntaxangaben werden Parameter kursiv ausgezeichnet (z. B. `cd Verzeichnisname`). Eckige Klammern [ ] kennzeichnen optionale Elemente. Alternative Eingaben sind durch einen senkrechten Strich | getrennt. Benutzereingaben auf der Konsole werden **fett** hervorgehoben.

## Was bedeuten die Symbole im Buch?



Hilfreiche Zusatzinformation



Praxistipp



Warnhinweis

## HERDT BuchPlus - unser Konzept:

### Problemlos einsteigen - Effizient lernen - Zielgerichtet nachschlagen

(weitere Infos unter [www.herdt.com/BuchPlus](http://www.herdt.com/BuchPlus))

Nutzen Sie dabei unsere maßgeschneiderten, im Internet frei verfügbaren Medien:



- Rufen Sie im Browser die Internetadresse [www.herdt.com](http://www.herdt.com) auf.

1 Wählen Sie Codes.

2 Geben Sie den folgenden Matchcode ein: NWIPV6.

## 2 Das IPv4-Protokoll und seine Grenzen

### In diesem Kapitel erfahren Sie

- ✓ wie und wann das Internet-Protokoll (IP) entwickelt wurde
- ✓ welche Eigenschaften IPv4 auszeichnen
- ✓ wie IPv4 mit anderen Protokollen zusammenwirkt
- ✓ wie die IPv4-Adressierung funktioniert
- ✓ welche Maßnahmen bei der Verknappung der IPv4-Adressen getroffen werden

### 2.1 Internet und Internet-Protokoll

#### Adressierung im Internet mit dem Internet-Protokoll

Das Internet ist eine riesige Infrastruktur, deren einzige Aufgabe es ist, Daten von einem Ort zum anderen Ort zu transportieren. Wie bei allen Infrastrukturen, die der Informationsvermittlung dienen, liegt das Hauptaugenmerk der Betreiber und Nutzer darin, dass die Daten zuverlässig, sicher und zielgenau zu dem Adressaten gelangen, für den sie bestimmt sind.

Für die zielgenaue Zustellung der Daten beim Adressaten ist das **Internet-Protokoll (IP; engl. Internet Protocol)** zuständig. Das Internet-Protokoll sorgt durch eine eindeutige Adressierung der Endgeräte dafür, dass die Daten bei genau dem technischen Gerät ankommen, für das sie bestimmt sind. Diese elementare Funktion macht die Bedeutung des Internet-Protokolls und seine Komplexität in der Anwendung aus.

IP folgt bei der Adressierung der Logik seiner kommunikationstechnischen Vorgänger, der postalischen und später fernsprechtechnischen Zuordnung von Zeichenketten („postalische Anschrift“, „Telefonnummer“) zu damit eindeutig gekennzeichneten Empfangsgeräten („Briefkasten“, „Telefon“).

#### Technische Anforderungen an IP

Mit der rasanten Entwicklung des Internets und der konstanten Zunahme von datenempfangenden Geräteklassen, einzelnen Geräten und deren jeweils spezifischen Anforderungen wachsen die technischen Herausforderungen an das Internet-Protokoll. Das Internet-Protokoll muss nicht nur die eindeutige Adressierung einer riesigen Anzahl einzelner Empfänger oder genau definierbarer Gruppen von Empfängern sicherstellen, es muss auch den datentechnischen Anforderungen an schnelle und effiziente Verarbeitung in unterschiedlichsten Konstellationen genügen (vgl. Abschnitt 8.1).

#### IPv4 und IPv6

Mit **IPv4 (Internet Protocol Version 4)** wurde für die Adressierung in Daten-Netzwerken ein Standard geschaffen, der diese Aufgaben mehr als zwei Jahrzehnte lang zufriedenstellend erfüllte. Der Nachfolger **IPv6 (Internet Protocol Version 6)** löst IPv4 ab. Diese Ablösung vollzieht sich kontinuierlich, beide Standards werden über einen heute noch nicht absehbaren Zeitraum parallel existieren. IPv4 und IPv6 sind gleichzeitig nutzbar (Dual-Stack). Deshalb ist für die Administration von IPv6 eine grundlegende Kenntnis der Prinzipien des IPv4-Protokolls notwendig. Jeder, der sich mit dem Internet-Protokoll beschäftigt, sollte Gemeinsamkeiten und Unterschiede der beiden Versionen des Protokolls erkennen und zuordnen können.

Wissen zu IPv4 mit seiner im Vergleich zu IPv6 überschaubaren Grundstruktur erleichtert das Verständnis der Prinzipien des Protokolls. Sie ermöglicht, bestimmte Aspekte des neuen Protokolls als Weiter- bzw. Neuentwicklung im Kontrast zum älteren Protokoll zu begreifen und die Funktion beider im Gesamtkontext von Protokollen und Diensten zu bestimmen.

## 2.2 Überblick zur Entwicklung von TCP/IP

### Entwicklung von IP

IP ist integraler Bestandteil der Protokollfamilie **TCP/IP (Network Transmission Protocol/Internet Protocol)** und arbeitet eng mit anderen Internet-Protokollen zusammen. TCP/IP dient oft als (verkürzendes) Synonym für die datenübermittelnde Komponente des Internets, so wie HTML oft als „Abkürzung“ für die inhaltsdarstellende Komponente verwendet wird.

Als IPv4 im Jahr 1983 das Protokoll NCP und sämtliche Vorgängerversionen ablöste und als einzige Technologie zur logischen Adressierung im Internet eingeführt wurde, geschah dies nach rund 20-jähriger Arbeit zur Entwicklung der Netzwerktechnik. Die Entwicklung des Internet-Protokolls war letztlich eine zwangsläufige Konsequenz aus wirtschaftlichen und militärischen Notwendigkeiten der Zeit.

Folgende Anforderungen bestanden:

- ✓ Die Steuerung der Systeme musste von jedem als notwendig angesehenen Punkt des Netzwerkes aus möglich sein.
- ✓ Prinzipiell sollten eine intelligente Wegewahl und die Nutzung unterschiedlicher Pfade und Medien möglich sein, z. B. beim Ausfall von Routern oder Wegstrecken.

Die Folge war eine intensive Zusammenarbeit zwischen Militär, Universitäten und Industrie zum wechselseitigen Vorteil. Motor der Entwicklung waren dabei insbesondere das amerikanische Verteidigungsministerium (**Department of Defense, DoD**) und die 1958 gegründete **ARPA (American Research Projects Agency)**. Letztere hat insbesondere die Aufgabe, Projekte zu koordinieren und mit den notwendigen Mitteln auszustatten.

### Das Arpanet als Vorläufer des Internets

Das **Arpanet** wurde 1962 auf Initiative und unter Beteiligung des DoD und des Massachusetts Institute of Technology (MIT) gegründet. Der ursprünglich militärische Zweck lag darin, ein Kommunikationsnetz zu schaffen, das in der Lage ist, den Datenfluss auch im Falle eines Ausfalls bzw. der Zerstörung weiter Bereiche seiner Infrastruktur durch selbstständige Neuorientierung der Kommunikationsdaten in den verbleibenden Netzbaukomponenten weitestgehend sicherzustellen. Ziel war es, Techniken für eine dezentrale Netzwerkstruktur zu schaffen, die beim Ausfall einzelner Subnetze immer noch funktionsfähig bleibt. Prinzipiell sollten alle verfügbaren technischen Verfahren integrierbar sein. Eine Festlegung auf spezielle Hard- und Software-Technologien oder gar die Produkte eines bestimmten Herstellers erfolgte nicht.

1965 wurden zunächst vier amerikanische Forschungseinrichtungen zu einem Netzwerk verbunden. Trotz anfänglicher Bedenken des Verteidigungsministeriums hinsichtlich der Umsetzung der dezentralen Aspekte des Projektes begann 1969 der praktische Aufbau und die Inbetriebnahme des Arpanet, des Vorläufers des heutigen Internets.

### Entwicklung von TCP/IP als offener Standard

Das Arpanet benötigte ein eigenes Konzept, welches die unterschiedlichen Aufgaben und Funktionen von Elementen der Netzwerktechnik beschreibt. Die Entscheidung fiel für ein Modell, das Netzwerkvorgängen spezifischen Funktionen in Form von zunächst drei, später vier hierarchisch aufeinander aufbauenden Schichten (engl. Layer) zuordnet. Das TCP/IP-Referenzmodell (früher DoD-Modell) wurde Ende der sechziger Jahre veröffentlicht.

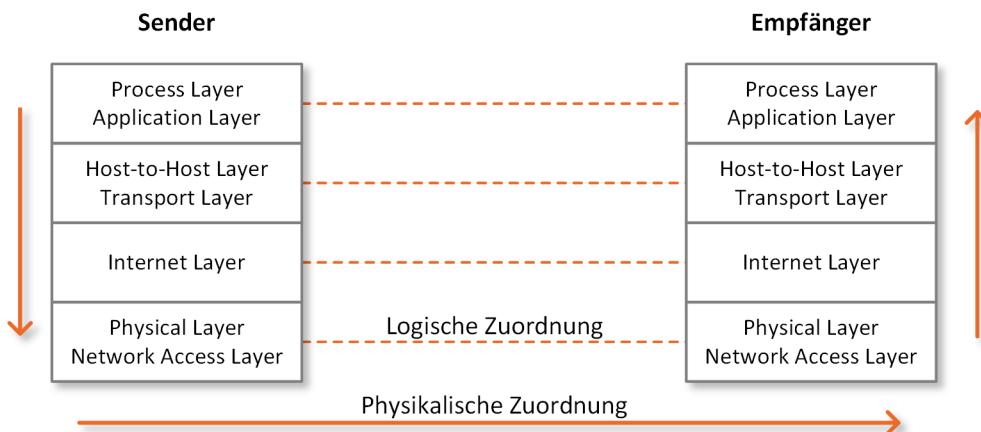
1983 wurde von der International Telecommunications Union (ITU) das OSI-Referenzmodell (Open Systems Interconnection Model) entwickelt, das aus sieben Schichten besteht. Das OSI-Referenzmodell bietet durch eine höhere Anzahl von Schichten, die je nach Bedarf zum Einsatz kommen und ggf. auch weggelassen werden können, eine höhere Flexibilität bei der Gestaltung der Protokolle.

Die Netzwerkschichten beider Modelle können wie folgt verglichen werden:

TCP/IP-Referenzmodell	OSI-Schichtenmodell
Anwendungsschicht (Application Layer)	Anwendungsschicht (Application Layer)
	Darstellungsschicht (Presentation Layer)
	Sitzungsschicht (Session Layer)
Transportschicht (Transport Layer)	Transportschicht (Transport Layer)
Internetschicht (Internet Layer)	Vermittlungsschicht (Network Layer)
Netzzugangsschicht (Network Access Layer)	Sicherungsschicht (Data Link Layer)
	Bitübertragungsschicht (Physical Layer)

Bei der Kommunikation werden die Daten logisch zwischen identischen Schichten ausgetauscht. Die Datenübertragung erfolgt physisch beim Senden an die jeweils untergeordnete Schicht und beim Empfangen an die jeweils übergeordnete Schicht. Die physikalische Übertragung der Daten zum anderen Gerät findet ausschließlich auf der untersten Schicht (Physical Layer) statt.

Im Folgenden wird das einfachere TCP/IP-Referenzmodell verwendet, da dieses zum Verständnis ausreicht.



*Informationsfluss bei der Datenübertragung am Beispiel des TCP/IP-Referenzmodells*

### Die wesentlichen Kennzeichen der Schichtenmodelle

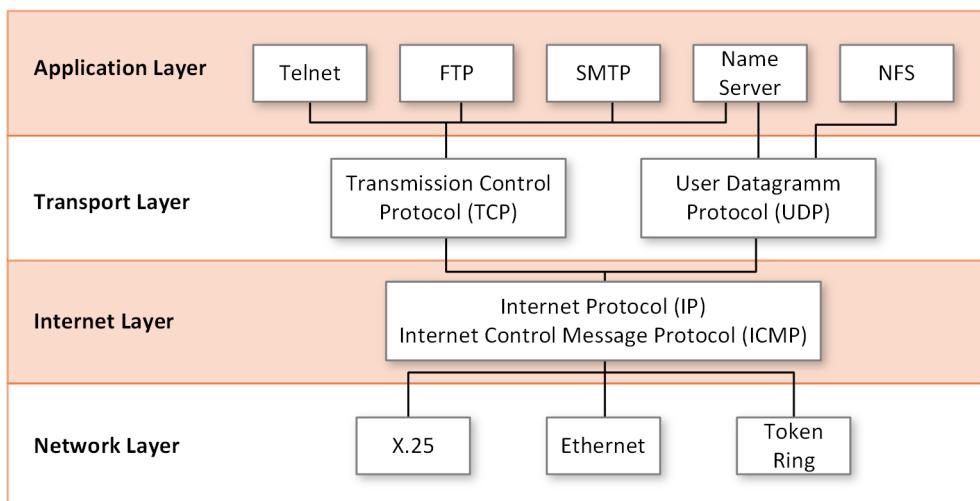
- ✓ Jede Schicht und die ihr zuzuordnenden Protokolle erfüllen spezifische Aufgaben.
- ✓ Alle Schichten verfügen nur über Kenntnis der jeweils eigenen Methoden und Vorgänge.
- ✓ Um die Methoden und Vorgänge bereitzustellen, nutzen sie die Dienste der jeweils unmittelbar darunterliegenden Ebene.

- ✓ Bei der Übertragung werden die Daten senderseitig aufbereitet, indem auf jeder Schicht den Daten protokollspezifische Informationen als Vorspann (Header) vorangestellt werden. Die Aufbereitung erfolgt dabei senderseitig vertikal abwärts. Die jeweils darunterliegende Schicht behandelt die von der jeweils darüberliegenden Schicht übergebenen Daten als Nutzlast (engl. Payload).
- ✓ Für IPv4 und IPv6 ist der Internet Layer (TCP/IP-Modell) bzw. die Vermittlungsschicht (OSI) relevant.

## 2.3 IP im Kontext des TCP/IP-Modells

### Funktion der Schichten und Protokolle

Grundlegend für die Arbeit mit dem Internet-Protokoll ist das Verständnis der dem TCP/IP zugrunde liegenden Dienste und Protokolle. Diese Dienste und Protokolle sind den jeweils unterschiedlichen Netzwerkschichten zugeordnet. IP ist dem Internet Layer zugeordnet.



Funktionen der Schichten und Protokolle

Anwendungsschicht	Dient als Netzwerkschnittstelle für Anwendungsprogramme
Transportschicht	Sorgt für <b>zuverlässige</b> Ende-zu-Ende-Verbindungen zwischen Sender und Empfänger (TCP) bzw. realisiert einen <b>unzuverlässigen</b> (der Empfang eines Datenpakets wird nicht überprüft) Datenübertragungsdienst (UDP), welcher dafür wesentlich schneller arbeitet
Internetschicht	Ist für die logische Adressierung und das Routing verantwortlich. Neben IP (Version 4 und Version 6) werden weitere Protokolle mit teils ergänzender Funktionalität dieser Schicht zugeordnet (bei IPv4 z. B. ARP und ICMP, bei IPv6 das Neighborhood Discovery Protocol und ICMPv6).
Netzzugangsschicht	Dient dem physischen Zugang sowie der Adressierung und Flusssteuerung auf Hardwareebene. Im Unterschied zu den drei höheren Ebenen ist sie nicht Gegenstand von TCP/IP-Standards.

Bereits zu Beginn der 1970er Jahre und damit noch vor Einführung von TCP und IP begannen Entwicklung und Standardisierung von Netzwerkprotokollen. Von primärem Interesse war zunächst die Funktionsfähigkeit der Datenübertragung, auch wenn diese auf Kosten der Sicherheit vorangetrieben werden musste. Das Protokoll FTP (File Transfer Protocol; RFC 114/1971 und RFC 959/1985) ist nur ein Beispiel hierfür.

## Gremien

Das neue Netzwerk benötigte neue Standards und Institutionen. In der Folge wurden wichtige Gremien ins Leben gerufen, die die Standardisierungs- und Normierungsprozesse und die daraus resultierenden Standards entwerfen, diskutieren und verabschieden. Dies bedingt die oben erwähnte Unabhängigkeit von einzelnen Firmen oder Geräteklassen, da jeder Hersteller allein anhand der Standards die entsprechenden Geräte und die dazu passende Software entwickeln kann und so die Interaktion aller Geräte in einem Netzwerk sichergestellt ist.

Ogleich unter maßgeblicher Regie amerikanischer Militäreinrichtungen stehend, vollzog sich die Entwicklung von Beginn an unter Beteiligung von Universitäten, Forschungseinrichtungen und innovativen Firmen. Die wesentlichsten Einrichtungen, die hier eine Rolle spielen, sind:

- ✓ **ICANN** – Internet Corporation for Assigned Names and Numbers  
Non-Profit-Organisation zur Überwachung und Koordinierung der Vergabe von Namen und Adressen im Internet (wird von der Abteilung IANA gewährleistet)
- ✓ **IANA** – Internet Assigned Numbers Authority (vgl. auch Abschnitt 4.4)  
Umfangreiche Zuständigkeiten im Bereich der DNS-Namensaufführung (u. a. Root-Zone, Liste der Top-Level-Domänen)  
Globale Koordination der Zuweisung von IPv4 und IPv6. Damit ist die IANA den kontinentalen Behörden übergeordnet.  
Eine der ältesten und wichtigsten Organisationen im Internet; seit 1988 Abteilung der ICANN, zuvor der DARPA unterstellt
- ✓ **InterNIC** – Internet's Network Information Center  
Stellt Information zu den Registrierungsdiensten für Domänennamen bereit  
Veröffentlicht Informationen zu existierenden Domänen (whois)
- ✓ **IETF** – Internet Engineering Task Force  
Ziel ist die Förderung der Internettechnologie durch Erstellen und Veröffentlichen qualitativ exzellenter Dokumentation und die Koordination von Tagungen und Diskussionsforen. Hierzu zählen:
  - ✓ **RFCs** – Requests for Comment (in Verbindung mit IAB – Internet Activities Board)
  - ✓ **Normen und Thesenpapiere** zu TCP/IP-Protokollen
  - ✓ **Allgemeine Publikationen**

## Merkmale des Internet Protocol

Beim Internet Protocol handelt es sich um ein verbindungsloses Protokoll auf Ebene des Internet Layer im TCP/IP-Referenzmodell bzw. der Vermittlungsschicht (Network Layer) im OSI-Modell. Es ermöglicht die logische, hardware-unabhängige Adressierung und dient dem Routing (der Wegewahl) über die Grenzen lokaler Netzwerkabschnitte hinweg.

Da keine Ende-zu-Ende-Verbindung aufgebaut wird, verfügt IP über keine eigenen Mittel zur Überprüfung des Sendeerfolgs und zur Fehlerkorrektur. Es realisiert damit einen unzuverlässigen Datagrammdienst. Wird jedoch für die Übertragung von Daten eine zuverlässige Verbindung benötigt, muss diese Funktion von Protokollen höherer Schichten bereitgestellt werden. Diesem Zweck dient insbesondere das Transmission Control Protocol (TCP), das von vielen Anwendungsprotokollen bevorzugt oder exklusiv benutzt wird.

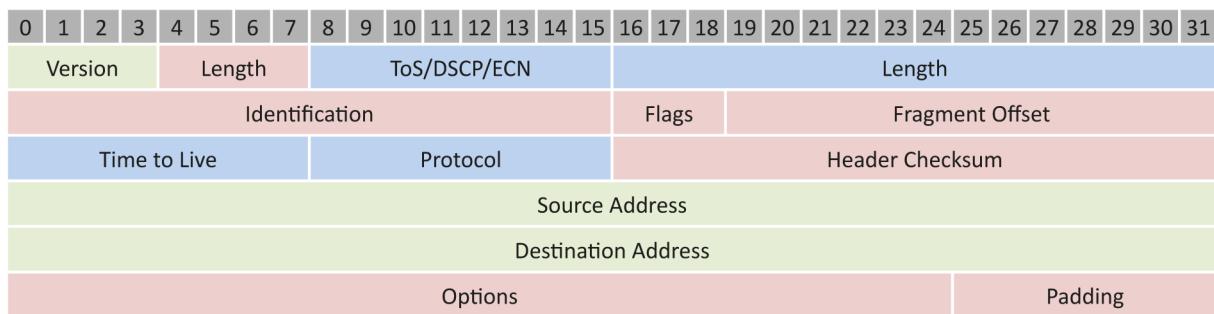
## 2.4 IPv4-Header

### Größe des IPv4-Datagramms

IPv4 muss Pakete von mindestens 576 Byte unterstützen. Ein IPv4-Paket kann minimal 20 Byte groß sein (nur Header), dieses wird aber im Ethernet aufgrund der dortigen minimalen Ethernet-Framegröße per Padding auf mindestens 64 Byte aufgefüllt. Die maximale Größe eines IP-Datagramms beträgt 64 Byte. In der Regel findet eine Anpassung an die von der Hardware unterstützte maximale Framegröße (Rahmengröße) statt. Standardlänge des IP-Datagramms in Ethernet-Netzwerken ist damit 1500 Bytes.

### Aufbau des IPv4-Datagramms

Das Datagramm gliedert sich in den IPv4-Header und den eigentlichen Inhalt (Payload). Der IPv4-Header zu Beginn des Datagramms ist von variabler Größe. Zu dem stets vorhandenen fixen Anteil von 20 Bytes Länge kommen optionale Erweiterungen. Die Darstellung des Headers erfolgt in 32-Bit-Worten, wobei ggf. auf die nächste Ganzzahl aufgerundet wird.



*IPv4-Header Übersicht*

<b>Version</b>	Versionsnummer des Internet Protocol (4 oder 6)
<b>IHL – Internet Header Length</b>	Länge des IPv4-Headers in 32-Bit-Worten Bei Verwendung von Optionen (Options) wird auf die nächste Ganzzahl aufgefüllt (Padding).
<b>ToS – Type of Service</b>	Optionale Einstellungen, die der Prioritätssteuerung dienen sollen; wird in der Regel ignoriert bzw. für Quality of Service (QoS) verwendet (RFC 2474)
<b>Total Length</b>	Gesamtlänge des Datagramms inklusive Header (max. 64 KB)
<b>Identification, Flags und Fragment Offset</b>	Ist die maximal unterstützte Rahmengröße überschritten, werden Datagramme auf dem Weg zum Empfänger weiter unterteilt (fragmentiert). Diese drei Felder dienen gemeinsam dem Zusammenfügen übertragener Fragmente durch den Empfänger.  Die <b>Identification Number</b> dient in Verbindung mit der Senderadresse der Bestimmung der Zusammengehörigkeit von Fragmenten.  Das <b>Flags</b> -Feld trifft Aussagen zum Status der Fragmentierung: <b>DF = Don't fragment</b> Das Datagramm darf nicht fragmentiert werden. <b>(0=nein; 1=ja)</b>  <b>MF = More fragments</b> Dem Fragment folgen noch weitere zur selben Identification Number gehörende Abschnitte. <b>(0=nein; 1=ja)</b>

Identification, Flags und Fragment Offset	<b>Fragment Offset</b> legt fest, an welcher Stelle des kompletten Datagramms das Fragment beginnt. Für das erste Fragment ist Fragment Offset = 0. Fragment Offset erhöht sich mit den nachfolgenden Abschnitten um die jeweilige Länge des Datenfeldes.
Time to Live	Die hier festgelegte maximale Lebenszeit eines Datagramms verhindert ein potenziell endloses Kreisen aufgrund fehlerhaften Routings (Schleifenbildung). In <b>RFC 791</b> wird die Lebenszeit zunächst als Ganzzahl in der Einheit Sekunden definiert. Da jedoch Zeitmessung auf Ebene des Physical Layer nicht möglich ist, reduziert jeder Router auf dem Übertragungsweg die Lebenszeit um 1. Erreicht das Paket den Wert 0, wird es verworfen.
Protocol	Hier steht die Nummer des im Datenabschnitt gekapselten Protokolls einer höheren Schicht. Beispiele hierfür sind: <b>0 IP</b> <b>1 ICMP</b> <b>3 GGP (Gateway-to-Gateway-Protocol)</b> <b>6 TCP</b> <b>17 UDP</b> <b>50 ESP (Encapsulating Security Payload)</b> <b>51 AH (Authentication Header)</b> Die Protokollnummern sind in der Regel der Datei <i>protocols</i> zu entnehmen: Bei Unix- und Linux-Systemen befindet sich diese Datei im Verzeichnis <i>/etc</i> . Unter Windows ist sie im Verzeichnis <i>%systemroot%\system32\drivers\etc</i> zu finden.
Header Checksum	Einfache Prüfsumme des Headers ohne den Nutzinhalt des Datagramms. Da der Header sich ändernde Informationen beinhaltet, ist auf jedem Netzwerknoten eine Neuberechnung erforderlich.
Source Address und Destination Address	Absender- und Empfängeradresse, Länge jeweils 32 Bit
Options und Padding	Optionen dienen der Erweiterung des Headers und bieten die Möglichkeit, neue Entwicklungen zu integrieren. Da die Optionen nicht an einer Wortgrenze enden, wird der restliche Platz auf eine volle Wortlänge aufgefüllt (Padding). Bekannte Optionen beziehen sich etwa auf Routing-Informationen: <b>Strict Source Routing</b> legt die exakte Route des Datagramms fest und erfordert die Aufzeichnung seines Wegs. <b>Loose Source Routing</b> erfordert zwar die Nutzung der angegebenen Router, lässt aber zusätzlich die Wahl weiterer Router zu. <b>Record Route</b> fordert die beteiligten Knoten auf, zur Aufzeichnung des Wegs ihre IP-Adresse an das Datagramm anzuhängen. <b>Time Stamp</b> erfasst die Zeit des Durchlaufs durch den beteiligten Netzwerknoten.

## 2.5 ARP und ICMP

### IP und Protokolle zur Unterstützung der Kommunikation

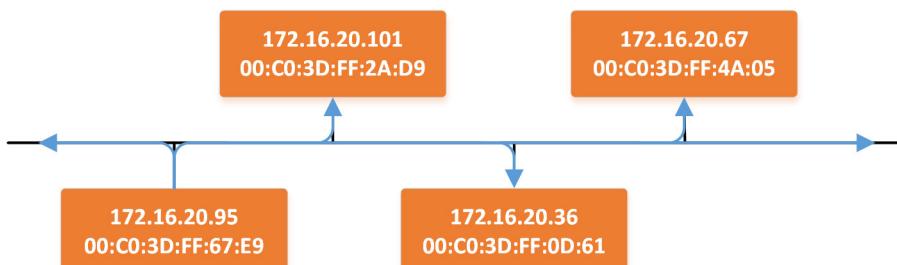
Aufgrund fehlender Korrektur- und Adressauflösungsmechanismen des Internet Protocol gehören zu jeder IP-Implementierung weitere Protokolle der Verbindungsschicht mit ergänzender Funktion. Da die mit IPv4 genutzten Protokolle bereits zu Beginn der 1980er Jahre entstanden sind, existieren Schwächen, die im Kontext von IPv6 Anlass zu tiefgreifender Modifizierung (Beispiel: ICMP) gaben oder sogar zum vollständigen Ersatz (Beispiel: ARP) des jeweiligen Protokolls führten.

### ARP – Address Resolution Protocol

Die Kommunikation in IP-Netzwerken erfordert die korrekte logische (IP) und physikalische (MAC) Adressierung von Sender und Empfänger. Innerhalb lokaler IPv4-Netzwerke dient das Address Resolution Protocol (ARP, RFC 826/ 1982) zur Zuordnung der physischen MAC-Adresse zu einer logischen IP-Adresse.

Will ein Host (Rechnersystem in einem Netzwerk) A einem Host B eine Nachricht schicken, gehen diesem Vorgang die folgenden Überprüfungs- und Auflösungsvorgänge voran:

- ✓ **Host A** prüft seinen **ARP-Cache** (Speicherort für bereits erfolgte und hier dokumentierte Verbindungen) auf die Verfügbarkeit der erforderlichen Kombination von IP- und MAC-Adresse.
- ✓ Ist dies der Fall, sind zur Ermittlung keine weiteren Schritte notwendig. Die Sendung kann direkt erfolgen.
- ✓ Liegt die notwendige Information nicht vor, generiert der Sender eine Anfrage (**ARP Request**) in Form eines **Broadcasts** („Rundruf“ an alle erreichbaren Geräte). Dieser beinhaltet auch die für die Rückantwort erforderliche eigene IP- und MAC-Adresse.
- ✓ **Host B** erhält die Anfrage und erkennt, dass sie an ihn gerichtet ist. Er kopiert die IP- und MAC-Adresse von Host A in seinen **ARP-Cache**.
- ✓ Anschließend antwortet er dem anfragenden Host A mit seiner IP- und MAC-Adresse.
- ✓ **Host A** speichert diese Information temporär in seinem **ARP-Cache** und nutzt sie für die nachfolgende Kommunikation.



**ARP Request: Wem gehört die IP-Adresse 172.16.20.101?**

Beispiel:	Sender-Hardware-Adresse:	00:C0:3D:FF:67:E9
	Sender-IP-Adresse:	172.16.20.95
	Ziel-Hardware-Adresse:	FF:FF:FF:FF:FF:FF
	Ziel-IP-Adresse:	172.16.20.101

**RARP Request: Ich nenne meine physikalische Adresse.**

**Wer kennt meine IP-Adresse?**

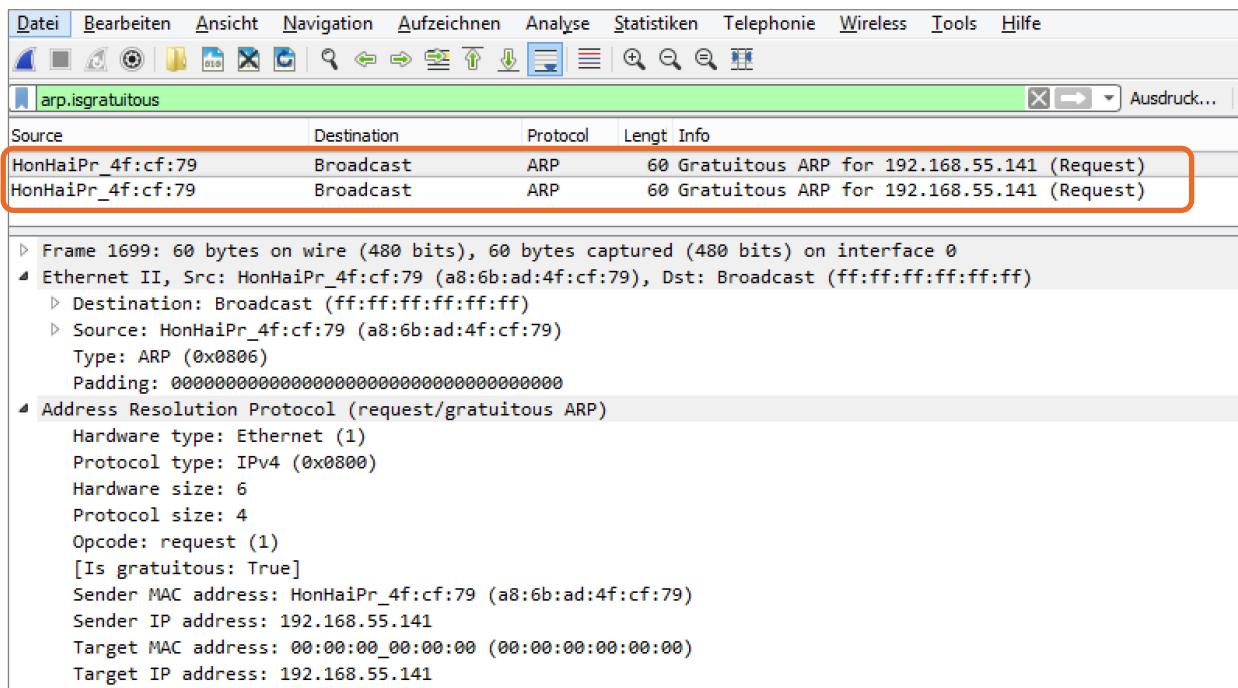
Beispiel:	Sender-Hardware-Adresse:	00:C0:3D:FF:2A:D9
	Sender-IP-Adresse:	0.0.0.0
	Ziel-Hardware-Adresse:	FF:FF:FF:FF:FF:FF
	Ziel-IP-Adresse:	255.255.255.255

## Kommunikation zwischen Netzen/Netzabschnitten

Befindet sich das Ziel außerhalb des eigenen Netzwerkes, erkennt das der Client und schickt seine Anfrage über die MAC-Adresse des Routers (IP-Adresse des Standardgateways). Der Router leitet das Datenpaket nach seinen Regeln weiter zum Empfänger oder zum nächsten Router. Der Client weiß dabei nichts über den Weg zum Zielnetz oder dessen Struktur.

### Gratuitous ARP

Beim **Gratuitous ARP** teilt ein Host seine IP/MAC-Kombination anderen Hosts im Netzwerk als Broadcast an die MAC-Adresse FF:FF:FF:FF:FF mit. Absender- und Empfänger-IP ist die des Hosts selbst. Wenn z. B. in redundanten Systemen eine Server-IP auf einen anderen Knoten wechselt, schickt der neue Server ein Gratuitous ARP, um die ARP-Caches der lokalen Clients/Router zu aktualisieren.



The screenshot shows a Wireshark capture window titled "arp.isgratuitous". The packet list pane displays two ARP frames. Both frames are ARP requests (opcode 1) sent to a broadcast destination (ff:ff:ff:ff:ff). The source MAC address is HonHaiPr\_4f:cf:79 (a8:6b:ad:4f:cf:79), and the source IP address is 192.168.55.141. The target MAC address is 00:00:00:00:00:00, and the target IP address is also 192.168.55.141. The "Info" column for both packets indicates "60 Gratuitous ARP for 192.168.55.141 (Request)". The details pane shows the structure of the ARP frame, including the hardware type (Ethernet), protocol type (IPv4), and the request opcode. The bytes pane shows the raw hex and ASCII data of the captured frames.

Source	Destination	Protocol	Lengt	Info
HonHaiPr_4f:cf:79	Broadcast	ARP	60	Gratuitous ARP for 192.168.55.141 (Request)
HonHaiPr_4f:cf:79	Broadcast	ARP	60	Gratuitous ARP for 192.168.55.141 (Request)

*Gratuitous ARP-Anfrage, mitgeloggt im Tool „Wireshark“*

### Broadcast

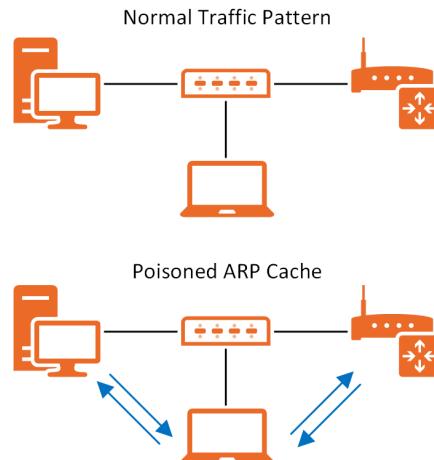
Bei DHCP schickt der Client Ethernet Broadcasts mit seiner eigenen MAC-Adresse als Absender und FF:FF:FF:FF:FF als Ziel-MAC-Adresse an alle lokalen Systeme. Da er selbst am Anfang in der Regel noch keine IP-Adresse hat, setzt er als Absender-IP 0.0.0.0 und als Empfänger-IP 255.255.255.255 ein.

## ARP und Sicherheit

Zu der Zeit, in der ARP entwickelt wurde, legte man Wert auf Einfachheit und Robustheit, nicht jedoch auf Sicherheit. Aufgrund fehlender Überprüfungsmaßnahmen kann ein Angreifer falsche IP-zu-MAC-Adresskombinationen verteilen. Unter Verwendung einfacher Werkzeuge ist es ihm prinzipiell möglich, Gateway-Funktion anzunehmen und sich in den Datenaustausch anderer Stationen einzuklinken, indem er den Beteiligten gegenüber jeweils die MAC-Adresse des anderen Hosts annimmt (**ARP Cache Poisoning**).

Lösungsversuche zur Abwehr solcher Manipulationsversuche umfassen insbesondere die folgenden Maßnahmen:

- ✓ Überwachen der Zuordnung zwischen IP- und MAC-Adressen mithilfe entsprechender Programme (Arpwatch, X-ARP)
- ✓ Statische ARP-Einträge (hoher Administrationsaufwand)
- ✓ Festlegung, dass in der ARP-Tabelle nur Antworten auf eigene Anfragen akzeptiert werden



## ICMP Version 4

**ICMP (Internet Control Message Protocol;** RFC 792/1981) ermöglicht den Austausch von Fehler- und Kontrollmeldungen für IP in der Version 4. ICMP-Typen und -Codes werden von verschiedenen Diagnosetools verwendet, zum Beispiel:

- ✓ ping
- ✓ tracert
- ✓ pathping

Wie bei den Protokollen der Transportschicht TCP oder UDP wird ICMP als Payload in das IP-Datagramm gepackt. Da ICMP fester Bestandteil jeder IP-Implementierung ist und nicht der Transportsteuerung dient, wird es der Vermittlungsschicht zugeordnet.

ICMP ist das Protokoll mit der Protokollnummer 1 und findet ausschließlich bei IPv4 Verwendung. Bei IPv6 und dem Neighborhood Discovery Protocol wird es durch ICMPv6 (Protokollnummer 58) ersetzt. Dieses verfügt über eigene Typ- und Codenummern.

Beispiele für wichtige ICMPv4-Typen und -Codes sind:

Typ 0	Echo Reply	Verwendet von ping und tracert: Antwort des Zielsystems
Typ 3	Destination Unreachable	Ziel nicht erreichbar (Wird per Code genauer bezeichnet)
Typ 3	Code 0	Network Unreachable
Typ 3	Code 1	Host Unreachable
Typ 3	Code 3	Port Unreachable
Typ 3	Code 4	Fragmentation Needed – Paket ist zu groß; zugleich ist die Option Don't Fragment gesetzt
Typ 8	Echo Request	Verwendet von ping und tracert: Anfrage
Typ 12	Parameter Problem	Beschädigter Header oder fehlende Optionen

Zusätzlich zu den genannten existieren zahlreiche weitere Parameter, die z. B. der Zeitsynchronisierung, der IP-Informationsanforderung und -antwort sowie der Subnetzmaskenanfrage und -antwort dienen. Einige von ihnen sind schon seit Langem bedeutungslos, weshalb sie bei ICMPv6 nicht mehr verwendet werden.

## 2.6 Aufbau der IPv4-Adresse

### Adressformat

**RFC 791** (1981) legt für IPv4-Adressen eine fixe Länge von 32 Bit fest. Diese wird in vier Oktette zu je 8 Bit unterteilt, die durch einen Punkt voneinander getrennt dargestellt werden.



Üblicherweise erfolgt die Darstellung zur besseren Lesbarkeit in gepunkteter dezimaler Form. Für die technische Realisierung und das Verständnis der IP-Adresse ist jedoch die Notation im dualen Zahlensystem maßgeblich.

**Beispiel:** 192.168.0.1

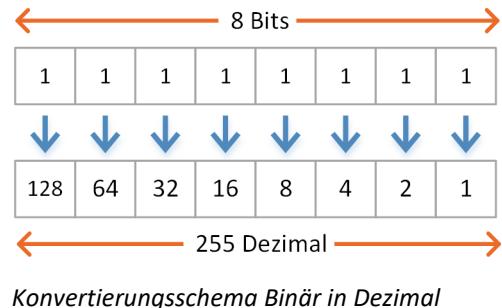
Dezimal	192.	168.	0.	1
Dual	11000000.10101000.00000000.00000001			

Zur Veranschaulichung dient die Zuordnung der (dualen) Bits eines Oktetts zu ihrer dezimalen Wertigkeit; da auch alle Bits die Wertigkeit „0“ annehmen können, kann jedes Oktett die Werte 0 bis 255 annehmen.

### Netzanteil und Hostanteil

Neben der Unterteilung der Adresse in Oktette erfolgt eine weitere Differenzierung, die der Zuordnung des Gerätes zu seinem Netzwerkabschnitt dient. Jede IPv4-Adresse wird in zwei Informationen unterteilt:

- ✓ den Netzwerkanteil
- ✓ den Hostanteil



Der **Netzwerkanteil** legt die in einem Netzwerk zu verwendende gemeinsame Kennung fest. Der **Hostanteil** definiert die individuelle Adresse eines Geräts.

Der Netzwerkanteil ist durch die ersten Stellen der IP-Adresse definiert und kann fast jede Anzahl der führenden Bits der Adresse ausmachen, je nach Komplexität des zugrunde liegenden Netzabschnitts. Alle Bits der IP-Adresse, die nicht dem Netzwerkanteil zugeordnet sind, gehören dementsprechend zum Hostanteil.

### Beispiel

In ihrer Funktion sind diese beiden Größen der Telefonvorwahl (Netzwerkteil) und der Rufnummer (Hostanteil) vergleichbar. Befindet sich das Ziel im gleichen logischen Netzwerk wie der Sender, erfolgt die Zustellung direkt und gänzlich ohne vermittelnde Instanzen. Dies entspricht dem Telefonieren innerhalb des Ortsnetzes. Für Ziele außerhalb des logischen Netzwerks hingegen erfolgt die Zustellung über dedizierte Komponenten der Vermittlungsschicht (Router, Layer 3-Switche).

## 2.7 Netzwerkkennung (Subnetz-ID, Netz-ID), Hostadresse und Broadcast-Adresse

### Adressarten innerhalb eines Netzwerks

Innerhalb eines Netzwerks unterscheidet man die folgenden Adressarten:

<b>Subnetzmaske</b>	<p>Die Subnetzmaske definiert die Größe eines Netzwerkabschnitts eindeutig. Sie wird in Form einer IP-Adresse ausgeführt, dient allerdings ausschließlich als eine Art Schablone, mit deren Hilfe der Netzwerkanteil aus einer gegebenen IP-Adresse herausgerechnet werden kann.</p> <p>Bei der vorgegebenen Länge von 32 Bit besteht die Subnetzmaske zu Beginn der Adresse aus einer durchgehenden Folge binärer Einsen, gefolgt von einer Folge binärer Nullen.</p> <p>Je höher/größer die Subnetzmaske ausfällt (je mehr binäre Einsen vorgegeben sind), umso kleiner ist das betreffende Netzwerk.</p> <p>Die binären Einsen (vgl. „Konvertierungsschema Binär in Dezimal“ in Abschnitt 2.6) legen den Netzwerkanteil einer IP fest, während die binären Nullen dem Hostanteil entsprechen.</p> <p><b>Beispiel</b></p> <p>Die Subnetzmaske 255.255.255.0 (Binär: 11111111.11111111.11111111.00000000)</p> <p>legt die ersten drei Oktette als Netzwerkanteil und das vierte Oktett als Hostanteil fest.</p>
<b>Netzwerkkennung</b>	<p>Die Netzwerkkennung ist die unterste Adresse (alle Hostbits auf 0) und bezeichnet den gesamten Netzwerkabschnitt.</p> <p>Sie findet sich in der Routing-Tabelle des Hosts im Abschnitt Ziel und kann nicht zur individuellen Geräteadressierung verwendet werden.</p> <p><b>Beispiel</b></p> <p>In einem Netzwerk 192.168.1.0 und Subnetzmaske 255.255.255.0 ist die IP-Adresse 192.168.1.0 selbst die Netz-ID. Bei Verwendung der genannten Subnetzmaske kann die IP-Adresse 192.168.1.0 nicht als Adresse eines einzelnen Systems verwendet werden. Vielmehr dient sie zur Identifikation des lokalen Netzwerks.</p>
<b>Broadcast-Adresse</b>	<p>Innerhalb eines Subnetzes dient die Broadcast-Adresse als Rundsendungsadresse an alle Stationen.</p> <p>Dazu wird die oberste Adresse (alle Hostbits auf 1) benutzt, weshalb diese nie für einzelne Rechner eingesetzt werden kann.</p> <p><b>Beispiel</b></p> <p>Im Netzwerk 192.168.1.0 mit Subnetzmaske 255.255.255.0 wäre dies die Adresse 192.168.1.255.</p>
<b>Hostadressen</b>	<p>Die Hostadresse ist die eindeutige Kennung eines Geräts (Netzwerkkarte, Verbindung). Außer der untersten Adresse (Netz-ID) und der obersten Adresse (Broadcast-Adresse) können prinzipiell alle IPs verwendet werden.</p> <p><b>Beispiel</b></p> <p>Im Beispiel stehen die Adressen 192.168.1.1 bis 192.168.1.254 zur Verfügung.</p>

## Notation der Subnetzmaske

Übliche Darstellungsformen der Subnetzmaske sind:

- ✓ Die gepunktete dezimale Notation (vgl. IPv4-Adresse)

**Beispiel:** 255.0.0.0

- ✓ Die **CIDR- (Classless Inter-Domain Routing) Schreibweise** besteht aus einem führenden Slash und der Anzahl binärer Einsen in der Subnetzmaske. Sie wird hinter der IP-Adresse angefügt (vgl. 2.9)

**Beispiel:** 10.3.123.15 /24

Die Schreibweise /24 legt die ersten 24 Bit der IP-Adresse (die ersten drei Oktette) als Netzwerkanteil fest. Damit entspricht die Schreibweise /24 der 255.255.255.0 bei konventioneller Darstellung.

## Subnetzmaske und Netzwerkgröße

Netzwerke identischer Größe verfügen auch über die gleiche Subnetzmaske. Die einem Netzwerk zur Verfügung stehende Anzahl an Adressen kann aus der Differenz von 32 Bit als der Gesamtlänge einer IPv4-Adresse und dem Netzwerkanteil berechnet werden.

### Beispiel

Für eine Subnetzmaske von 255.255.255.0 alias /24 (s. o.) lautet die Berechnung wie folgt:

- ✓ Anzahl Hostbits = 32 – 24 (Netzwerkbits) = 8
- ✓ Gesamtzahl der Adressen =  $2^8$  = 256
- ✓ Verwendbare Hostadressen = 256 – 2 = 254

## Zusammenwirken von IP und Subnetzmaske

IPv4 bestimmt den Netz- und Hostanteil mittels bitweiser logischer Und-Verknüpfung von Adresse und Subnetzmaske. Seiner Position entsprechend wird jedes Bit der IP-Adresse mit dem korrespondierenden Bit aus der Subnetzmaske verknüpft. Nur dann, wenn eine Eins in der IP auf eine Eins in der Subnetzmaske trifft, ergibt sich auch als Resultat eine Eins. In jedem anderen Fall ist das Ergebnis eine Null.

### Beispiel

Ein Host verwendet als IP-Adresse und Subnetzmaske 192.168.0.17/24.

Die CIDR-Notation /24 entspricht 255.255.255.0 in gepunkteter Schreibweise.

Hieraus leitet die IPv4-Komponente wesentliche Informationen ab:

- ✓ Sie bestimmt die Netz-ID des lokalen Netzes.
- ✓ Sie bestimmt die Broadcast-Adresse (meist) als höchste Netzwerkadresse, sofern diese nicht explizit anders definiert wird (jede andere Adresse im Netzwerk bzw. Zero Broadcast als unterste Adresse im Netz).
- ✓ Sie legt einen Eintrag in der Routing-Tabelle des Hosts fest. Über diesen Eintrag „weiß“ das System, welche Adressen zum lokal verbundenen Netzwerk gehören. Diese werden direkt und ohne den Weg über einen Router adressiert.

### Binäre Umrechnung von IP und Subnetzmaske

Die Berechnung der Netzwerkennung erfolgt durch logische Und-Verknüpfung von binärer IP-Adresse und binärer Netzmase:

Oktett A								Oktett B								Oktett C								Oktett D								Typus	
0	128	64	32	16	8	4	2	1	128	64	32	16	8	4	2	1	128	64	32	16	8	4	2	1	128	64	32	16	8	4	2	1	32
1   1   0   0   0   0   0   0	1   0   1   0   1   0   0   0	0   0   0   0   0   0   0   0	0   0   0   1   0   0   0   1	Hostadresse																													
1   1   1   1   1   1   1   1	1   1   1   1   1   1   1   1	1   1   1   1   1   1   1   1	0   0   0   0   0   0   0   0	Subnetzmaske																													
↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓	↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓	↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓	↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓																														
1   1   0   0   0   0   0   0	1   0   1   0   1   0   0   0	0   0   0   0   0   0   0   0	0   0   0   0   0   0   0   0	Netz-ID binär																													
192								168								0								0								Netz-ID dezimal	

#### Bestimmung der Netz-ID

Da die Subnetzmaske immer aus einer Folge von binären Einsen (Netzanteil), gefolgt von einer Folge von binären Nullen (Hostanteil) besteht, ist die Ableitung der Netz-ID aus der IP-Adresse im Grunde recht einfach.

- ✓ Soweit der Block binärer Einsen reicht, kann als Ergebnis der Verknüpfung der entsprechende Wert aus der IP-Adresse übernommen werden (s. Konjunktionstabelle).
- ✓ Ab der ersten Stelle mit einer **binären Null** in der Subnetzmaske (auf die nur noch weitere Nullen folgen können) lautet das Resultat der Verknüpfung immer auf Null. Die binäre Null in der Subnetzmaske lässt als Ergebnis der logischen Und-Verknüpfung (vgl. Anhang) stets nur eine Null zu.

		Netz	Host
IP-Adresse	192.168.0.17	11000000.10101000.00000000.00010001	
Subnetzmaske	255.255.255.0	11111111.11111111.11111111.00000000	
Netz-ID	192.168.0.0	11000000.10101000.00000000.00000000	

Die Netz-ID wird entsprechend der Wertigkeit der entsprechenden Bits oktettweise zusammengesetzt.

Verläuft die Trennung zwischen Netzwerk- und Hostanteil exakt an einer Oktettgrenze (bei 255.0.0.0, 255.255.0.0, 255.255.255.0), ist die Bestimmung des Resultats noch einfacher:

Oktette, die ausschließlich binäre Einsen in der Subnetzmaske beinhalten, übernehmen das Ergebnis aus den entsprechenden Abschnitten der IP-Adresse. In obigem Beispiel ist dies für die ersten drei Oktette der Fall. Diese lauten damit auf 192.168.0.

Oktette, die ausschließlich binäre Nullen in der Subnetzmaske beinhalten, übernehmen das Resultat aus den jeweiligen Abschnitten der Subnetzmaske und besitzen den Wert Null. Im Beispiel ist dies für das letzte Oktett der Fall.

### Bestimmung der Broadcast-Adresse

Während bei der Netz-ID alle Bits im Hostanteil auf eine binäre Null lauten und sie somit die unterste Adresse im Netzwerk ist, werden für die Broadcast-Adresse die entsprechenden Bits standardmäßig auf eine binäre Eins gesetzt. Dazu verwendet man eine Oder-Verknüpfung (vgl. Anhang) mithilfe einer Invertierten Subnetzmaske.

Oktett A								Oktett B								Oktett C								Oktett D								Typus	
0	128	64	32	16	8	4	2	1	128	64	32	16	8	4	2	1	128	64	32	16	8	4	2	1	128	64	32	16	8	4	2	1	32
1	1	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	Hostadresse		
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓				
1	1	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Broadcast binär			
192								168								0								255								Broadcast dezimal	

### Bestimmung der Broadcast-Adresse

Im Beispiel besteht der Hostanteil nur aus dem vierten Oktett. Setzt man hier alle Bits auf eine binäre Eins, ergibt sich eine 255. Die nicht für einzelne Hosts zu verwendende Broadcast-Adresse lautet damit auf 192.168.0.255.

### Berücksichtigung in der Routing-Tabelle

Der Host nutzt die Ergebnisse der Berechnung zur Bestimmung der Adressen, die er ohne Vermittlung durch einen Router erreichen kann. Hierzu generiert er einen oder mehrere entsprechende Einträge in seiner Routing-Tabelle. Diese nutzt er bei jeglicher Netzwerkkommunikation ständig.

Für ein unter Windows 7 bzw. Windows 8 laufendes System werden in unserem Beispiel die folgenden Einträge generiert, die per Befehl `route print` bzw. `route print -4` abgerufen werden können:

Netzwerkziel	Netzwerkmaske	Gateway	Schnittstelle	Metrik	(Kommentar)
0.0.0.0	0.0.0.0	192.168.0.1	192.168.0.17	20	(Default-Gateway)
192.168.0.0	255.255.255.0	Auf Verbindung	192.168.0.17	276	(Netzwerk)
192.168.0.17	255.255.255.255	Auf Verbindung	192.168.0.17	276	(Eigene IP)
192.168.0.255	255.255.255.255	Auf Verbindung	192.168.0.17	276	(Broadcast)

Versucht man z. B., einen anderen Host mit der IP 192.168.0.34 zu erreichen, wird die Route mit Ziel 192.168.0.0 und Subnetzmaske 255.255.255.0 benutzt, da sich das Ziel in diesem Netzwerk befindet und zugleich keine spezifischere Route existiert. Der Gateway-Eintrag „Auf Verbindung“ besagt, dass es sich dabei um ein unmittelbar verbundenes, lokales Netzwerk handelt. Alle nicht bekannten Ziele werden zum Default-Gateway (Router) geschickt.

## 2.8 Netzwerkklassen

### Grundlagen zu Netzwerkklassen(historisch, vergl. 2.9)

Anfang der 1980er-Jahre hatte niemand die digitale Revolution und den daraus resultierenden Bedarf an Netzwerkabschnitten unterschiedlichster Komplexität vorhersehen können. RFC 791 (1981) definiert drei zur Host-adressierung verwendbare und in ihrer Größe abgestufte Netzwerkklassen fixer Größe. Auf diese Netzwerkklassen wurden klassenspezifische und fest definierte Subnetzmasken angewendet, woraus sich zunächst unveränderbare Netzwerkgrößen ergaben:

- ✓ Adressbereich **A**: 1 - 127
- ✓ Adressbereich **B**: 128.0 - 191.255
- ✓ Adressbereich **C**: 192.0.0 - 223.255.255

RFC 1112 (1989) erweitert dieses Schema um zwei zusätzliche Netzwerkklassen mit spezieller Funktion: Während der **Class-D**-Bereich Multicast-Adressen festlegt und so der Übertragung ein Sender an eine Gruppe von Empfängern dient, wurde **Class E** für experimentelle und zukünftige Zwecke reserviert.

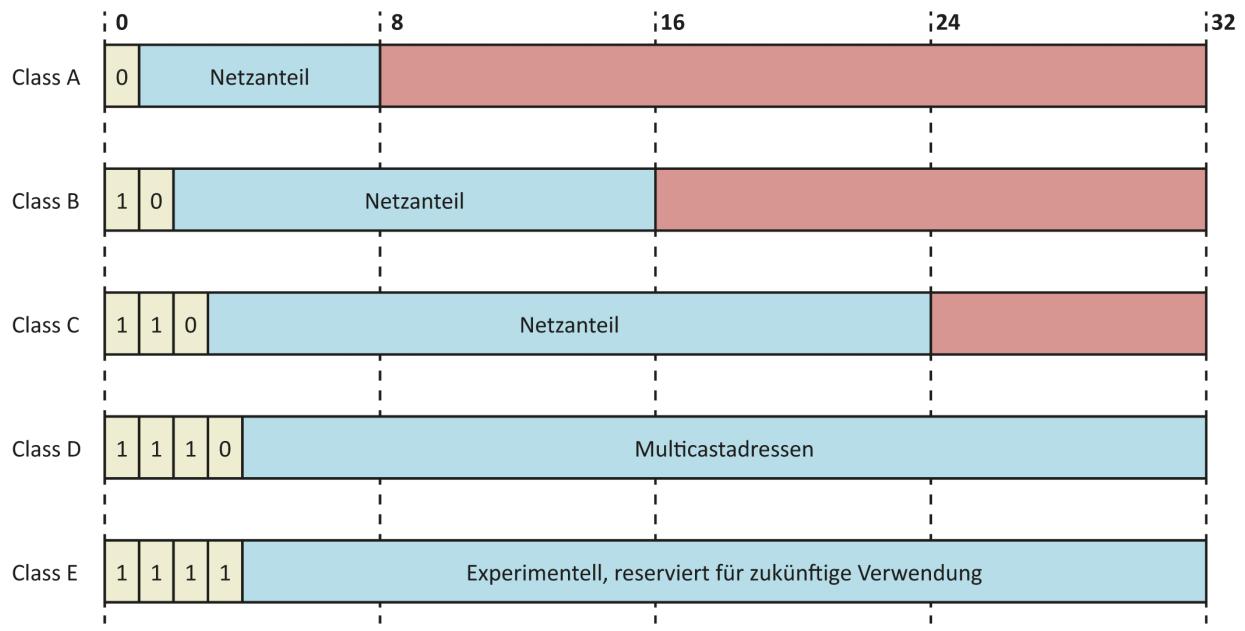
Folgende Netzwerkklassen stehen zur Verfügung:

Netzwerkklasse	Adressbereich	Subnetzmaske	Adressen je Subnetz	Beschreibung
Class A	1 bis 127	255.0.0.0	$256^3 = 16777216$	Sehr große Netzwerke
Class B	128.0 bis 191.255	255.255.0.0	$256^2 = 65536$	Mittelgroße Netzwerke
Class C	192.0.0 bis 223.255.255	255.255.255.0	256	Kleine Netzwerke
Class D	224 bis 239	240.0.0.0	nicht anwendbar	Multicast-Adressen
Class E	240 bis 255	240.0.0.0	nicht anwendbar	Experimentell, reserviert

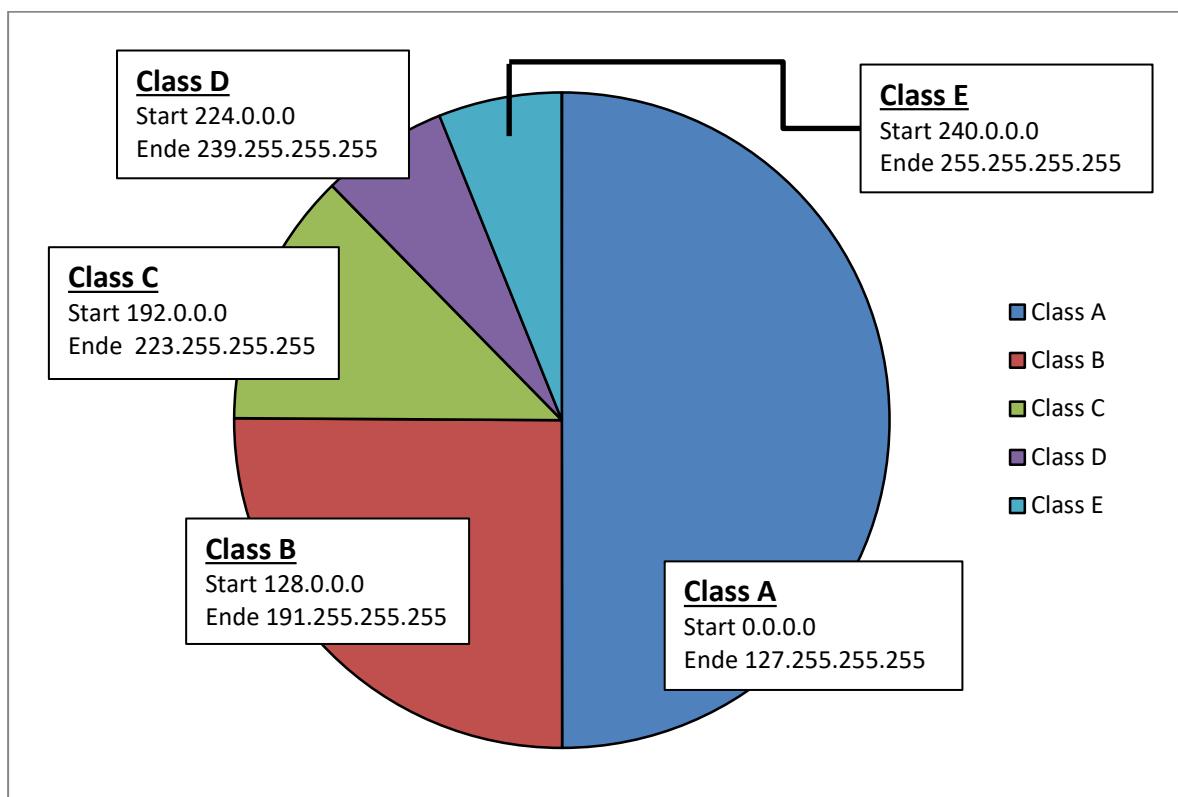
### Netzklassen und ihre binäre Ableitung

Die **Netzwerkklassen** sind aus den ersten bis zu vier Bits des ersten Oktetts einer IPv4-Adresse abzuleiten.

- ✓ Class-A-Netzwerke leiten sich aus der Tatsache ab, dass das erste Bit des ersten Oktetts stets mit einer binären Null beginnt. Damit besteht die Möglichkeit, mit den restlichen sieben Bits 127 Netzwerke einzurichten.
- ✓ Bei jeder weiteren Netzwerkkategorie verschiebt sich die binäre Null, die die Netzwerkkategorie definiert, stets um eine Position nach rechts, wobei das davorliegende Bit, das ja die nächsthöhere Klasse definiert, in eine binäre Eins gewandelt wird. Dadurch ergibt sich der logische Effekt, dass sich aufgrund der exponentiell ansteigenden Wertigkeit der Bits der Gesamtumfang der verfügbaren Adressen von Class A bis Class D mit jedem Schritt halbiert.



Einteilung der Netzwerkklassen



Grafik der maximalen Netzgrößen in den verschiedenen Netzklassen

## 2.9 IPv4: Adressraumknappheit bei IPv4 und Lösungskonzepte

### Subnetting und Classless Inter-Domain Routing (CIDR)

Bereits wenige Jahre nach Einführung von IPv4 wurde deutlich, dass die unflexible Organisation in Form von starren Netzwerkklassen den tatsächlichen Erfordernissen nicht genügen konnte.

Gemessen an der wachsenden Anzahl an Interessenten existierten viel zu wenige Netzwerkbereiche. Zugleich entsprachen die existierenden Netzwerkgrößen nicht dem realen Bedarf. Insbesondere im Class-A-Bereich, der sich ja nur in 127 Netzwerkabschnitte mit jeweils über 16 Millionen Adressen je Netzwerk unterteilen lässt, wurden bei ausschließlich klassenbehafteter Addressierung enorme Kapazitäten verschwendet.

Beginnend mit RFC950 (1985) führte dies zu einer Reihe von Überarbeitungen, die alle das Ziel verfolgten, Netzwerkgrößen und Routeneinträge abseits der starren Netzwerkklassen zu ermöglichen. Die in diesen Kontext fallenden Begriffe lauten:

- ✓ Subnetting (RFC 6918)
- ✓ Variable Length Subnet Mask
- ✓ Classless Inter-Domain Routing (RFCs 1518, 1519, 4632)

Die genannten Verfahren führten zu folgenden Anpassungen:

- ✓ Flexible Skalierung der Subnetzmaske und damit freie Anpassung der Netzwerkgröße
- ✓ Klassenlose Addressierung
- ✓ Aufteilen vorhandener Netze (klassenbehaftet oder klassenlos) in Subnetze
- ✓ Zusammenfassen angrenzender Netzwerke zu einem einzigen Netzwerk und Routenaggregation
- ✓ Strukturierte regionale Zuteilung der restlichen Class-C-Netzwerkbereiche (ab 1993) unter Bildung von vier kontinentalen Zonen

### Klassenlose Addressierung am Beispiel der Subnetzbildung

#### Aufgabenstellung

Die Firma ABC GmbH möchte den Netzwerkbereich 172.16.0.0 mit Subnetzmaske 255.255.0.0 (/16) in vier gleich große Subnetze aufteilen, um sie auf dem Firmengelände in unterschiedlichen Netzwerken einzusetzen.

Folgende Schritte sind zur Ermittlung der neuen Netzwerke nötig:

- Bestimmen Sie die neue Subnetzmaske.
- Ermitteln Sie die Netzwerk-IDs.
- Leiten Sie Broadcast- und Hostadressen ab.

#### Bestimmen der neuen Subnetzmaske

Jede Erhöhung der Anzahl an Netzwerksbits in der Subnetzmaske entspricht einer Halbierung der Netzwerkgröße (Anzahl der möglichen Hosts im Netzwerkbereich). Damit ändert sich die Subnetzmaske von /16 zu /18 bzw. 255.255.192.0.

## Ermitteln der neuen Netzwerk-IDs

Es bestehen mehrere prinzipiell gleichwertige Möglichkeiten zu deren Berechnung:

- ✓ **Binär:**  
Variieren der Subnetzkennung (00, 01, 10, 11). Dies entspricht einem Addieren der Schrittweite, die hier im dritten Oktett 64 beträgt. Sie leitet sich aus der Wertigkeit der in der Subnetzmaske definierten geringwertigsten Eins ab.
- ✓ **Dezimal:**  
Im dritten Oktett kann die Schrittweite von 64 zu jeder vorhergehenden Netz-ID hinzugeaddiert werden, bis die zu erreichende Anzahl an Netzwerken definiert ist. Die Schrittweite selbst ergibt sich dezimal aus 256 Adressen/4 Subnetze = 64 Adressen je Subnetz im dritten Oktett.

## Ermitteln der Broadcast-Adressen

- ✓ **Binär:**  
Innerhalb des Subnetzes werden alle Hostbits auf Eins gesetzt.
- ✓ **Dezimal:**  
Broadcast liegt eine Adresse unterhalb der nächsten Subnetz-ID.  
Beispiel: Lautet Subnetz-ID 2 auf 172.16.64.0, ist die Broadcast-Adresse für Subnetz 1 172.16.63.255.
- ✓ **Schrittweite:**  
Ist die erste Broadcast-Adresse ermittelt, addiert man für die weiteren Netzwerke stets die Schrittweite zwischen den Netzen hinzu.  
Beispiel: Broadcast Subnetz 2 ist 172.16.(63+64).255 und damit 172.16.127.255.

## Ermitteln der Hostadressen

Innerhalb des Subnetzes sind dies alle Adressen, die sich oberhalb der Netz-ID und zugleich unterhalb der Broadcast-Adresse befinden. Hat man dies für das erste Subnetz ermittelt, muss für alle weiteren nur noch die Schrittweite hinzugeaddiert werden.

11111111.11111111.11000000.00000000	255.255.192.0	Subnetzmaske
10101100.00010000.00000000.00000000	172.16.0.0	Netz-ID Subnetz 1
10101100.00010000.00111111.11111111	172.16.63.255	Broadcast Subnetz 1
10101100.00010000.01000000.00000000	172.16.64.0	Netz-ID Subnetz 2
10101100.00010000.01111111.11111111	172.16.127.255	Broadcast Subnetz 2
10101100.00010000.10000000.00000000	172.16.128.0	Netz-ID Subnetz 3
10101100.00010000.10111111.11111111	172.16.191.255	Broadcast Subnetz 3
10101100.00010000.11000000.00000000	172.16.192.0	Netz-ID Subnetz 4
10101100.00010000.11111111.11111111	172.16.255.255	Broadcast Subnetz 4

## Besonderheit RFC 950

RFC 950 ist obsolet und wird hier nur der Vollständigkeit halber erwähnt.

Nach **RFC 950** waren das unterste und das oberste Subnetz jeweils ungültig. Das unterste Subnetz benutzt dieselbe Netz-ID wie das ursprünglich zu teilende Netzwerk (172.16.0.0). Das oberste Subnetz verfügt über dieselbe Broadcast-Adresse (172.16.255.255) wie das reguläre Class-B-Netz.

Diese aus heutiger Sicht ziemlich unverständliche Folgerung erklärt sich aus der Wichtigkeit der Subnetzmaske zur Zeit der Entwicklung des Internet-Protokolls. Zwar existiert die Subnetzmaske bereits von Anfang an (RFC 791), doch zu Zeiten starrer, klassenbasierter Netzwerkklassen wurde sie nicht immer genannt.

**RFC 1878** revidiert diese Aussagen des RFC 950. Da zur IP stets auch die Subnetzmaske genannt werden muss, besteht die in RFC 950 unterstellte Verwechslungsgefahr nicht. Die Verwendung aller Subnetze wird dann gestattet, wenn sich im konkreten Einzelfall, wie heute üblich, niemand an die längst obsolete RFC 950 hält.

### Private IP-Adressbereiche

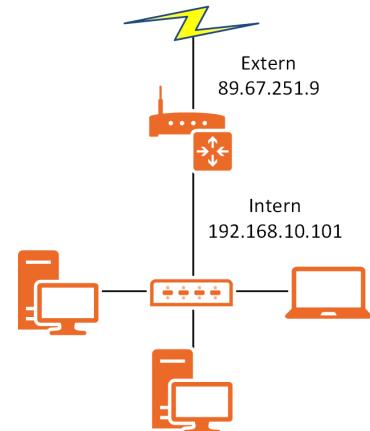
Angesichts der bereits frühzeitig erkennbaren Verknappung offiziell registrierter und damit weltweit eindeutiger IPv4-Adressen legte die IANA sog. Private IP-Netzwerkbereiche fest (**RFC1918**). Private IP-Adressen („privat“ im Sinne von *nicht öffentlich*) ermöglichen es, firmeninterne Netze aufzubauen, ohne auf Public IP-Adressen angewiesen zu sein. An den Netzgrenzen müssen die Private IPs in eine oder mehrere Public IPs umgewandelt werden. Das passiert heute z.B. in jedem DSL-Router mit Hilfe von **NAT** (Network Address Translation).

Im Gegensatz zu den öffentlichen Adressen (Public IPs), die bei einer Registrierungsbehörde oder einem Provider gegen Gebühr registriert werden müssen, steht die Nutzung von Private IPs jedermann frei. Aufgrund fehlender Eindeutigkeit sind sie im Internet nicht zulässig. Deshalb werden sie von Routern grundsätzlich ausgefiltert und verworfen.

Präfix	Ausdehnung von/bis	Größe/Alias	Anzahl Adressen
10/8	10.0.0.0 bis 10.255.255.255	24-Bit-Bereich	16777216
172.16/12	172.16.0.0 bis 172.31.255.255	20-Bit-Bereich	1048576
192.168/16	192.168.0.0 bis 192.168.255.255	16-Bit-Bereich	65536

### NAT

Ist für Private IP-Netzwerke der Zugang zum Internet erforderlich, muss auf der externen Schnittstelle des Routers die Netzwerkadressübersetzung (**NAT; Network Address Translation**) aktiviert werden. Im IP-Header wird dabei die private Senderadresse durch die öffentliche Adresse des Routers ersetzt (Source NAT). Hersteller- und verfahrensabhängig erfolgt dabei oftmals auch eine Manipulation des Senderports, weshalb z. T. auch von **NPAT (Network Port Address Translation)** gesprochen wird. Die Übersetzung zwischen privater IP und Port einerseits und öffentlicher IP und entsprechendem Port andererseits erfolgt dabei in einer Zuordnungstabelle.



Im Unterschied zu Source NAT ersetzt Destination NAT (DNAT) die Empfängeradresse. Damit wird es möglich, Dienstanforderungen an Server in privaten Netzen weiterzuleiten bzw. transparente Proxies einzurichten, von denen der anfordernde Benutzer nichts weiß.

Aufgrund der Möglichkeit zur Anbindung ganzer Netzwerke über eine bzw. wenige öffentliche Adressen entlastet NAT den Druck auf den IPv4-Adressraum erheblich. Wegen der Änderungen am IP-Header verstößt NAT aber gegen das Designziel der Integrität, weshalb es bei IPv6 ursprünglich nicht vorgesehen war.

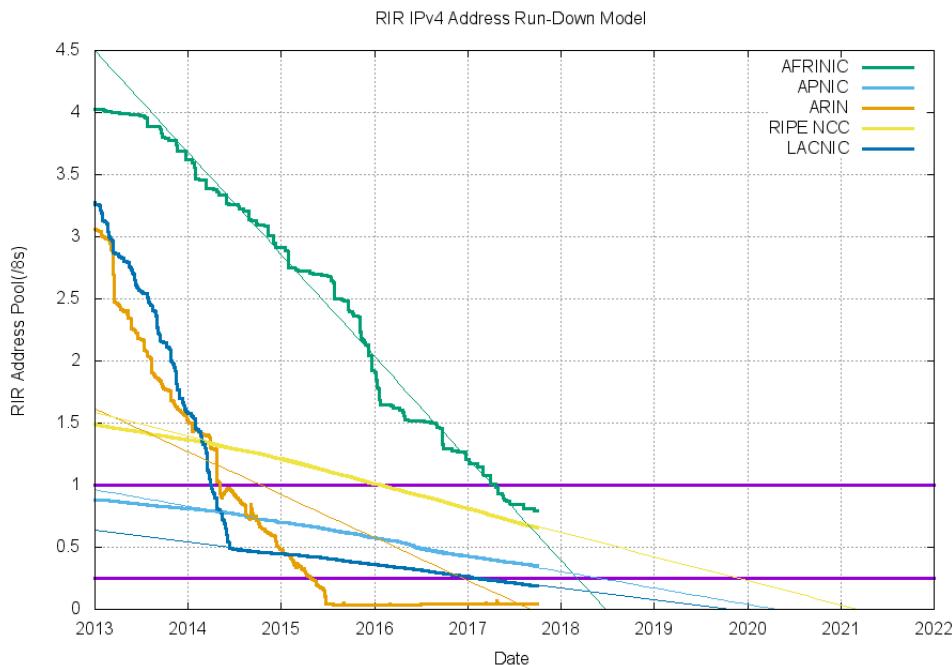
## Weitere spezielle IP-Adressen

Neben den erwähnten Spezifikationen und Differenzierungen sind weitere IP-Adressen bzw. -Adressblöcke besonderen Funktionen zugeordnet:

- ✓ Das Class-A-Netzwerk 127.0.0.0/8 wird als **Loopback** bezeichnet und für hostinterne Kommunikation genutzt. Es kann nicht zur Adressierung von Verbindungen genutzt werden. Datenpakete dürfen den Rechner nicht verlassen.
- ✓ Das Netzwerk 169.254.0.0/16 dient beim Fehlen bzw. Ausfall eines DHCP-Servers zur automatischen Konfiguration von Hosts, sofern nicht manuell eine alternative Konfiguration eingerichtet wurde (APIPA = Automatic Private IP Addressing). Diese linklokalen Adressen werden ausschließlich in privaten Netzwerken verwendet und dürfen den Netzabschnitt nicht verlassen. Sie werden grundsätzlich nicht geroutet.
- ✓ 0.0.0.0 als Hostadresse mit Subnetzmaske 0.0.0.0 steht für „dieser Host“ und wird von DHCP-Clients bis zum Ende des erfolgreichen bzw. fehlgeschlagenen DHCP-Leasevorgangs verwendet.
- ✓ Der Eintrag in einer Routingtabelle mit Ziel 0.0.0.0 und Subnetzmaske 0.0.0.0 verweist auf die Standardroute (Standardgateway-Eintrag). Da die Maske keine Netzwerksbits enthält, gelten alle Adressen als lokal und adressierbar. Aufgrund dieses Fehlens jeglicher Einschränkung und der Priorisierung von Routen über die Länge der Subnetzmaske (die Anzahl binärer Einsen, „Longest Prefix Match“) wird sie nur dann verwendet, wenn keine andere Route existiert.
- ✓ 255.255.255.255 wird als **Restricted Broadcast** bezeichnet. Sie ist zur Adressierung aller Knoten eines lokalen Netzwerks gültig.
- ✓ Der Bereich 100.64.0.0/10 ist laut RFC 6598 „IANA-Reserved IPv4 Prefix for Shared Address Space“ für Carrier Grade NAT reserviert. Er ist vergleichbar mit Private IP-Adressen.

## Wann gibt es keine IPv4 Adressen mehr?

Wie bereits angeführt ist der IPv4 Adressraum verbraucht. Auf der Webseite <https://ipv4.potaroo.net/> finden sich tagesaktuelle Grafiken zur IP-Address Exhaustion der 5 RIRs.



Statistik vom 30.9.2017

Aktuelle Zahlen für das RIPE sind unter <https://www.ripe.net/publications/ipv6-info-centre/about-ipv6/ipv4-exhaustion> abrufbar.



# 3 Eigenschaften des IPv6-Protokolls

## In diesem Kapitel erfahren Sie

- ✓ welche Eigenschaften das IPv6-Protokoll hat
- ✓ die Gründe für ein neues Internetprotokoll

## 3.1 Neuerungen im IPv6-Protokoll

### Was ist neu bei IPv6?

Das IPv6-Protokoll ist der Nachfolger des heutzutage weit verbreiteten IPv4-Protokolls und wurde 1998 als Standard festgelegt. IPv6 arbeitet wie IPv4 auf der OSI-Schicht 3, der Vermittlungsschicht.

IPv6 ist eine konsequente Weiterentwicklung von IPv4, um dem steigenden Bedarf an IP-Adressen gerecht zu werden und einige neue Funktionen sowie eine einfachere Erweiterbarkeit des Protokolls zu ermöglichen.

Die wichtigsten Neuerungen bei IPv6 sind:

- ✓ deutlich erweiterte Anzahl von IP-Adressen
- ✓ modifizierter Header für bessere Performance durch feste Header-Länge
- ✓ vereinfachtes Routing sorgt für kleinere Routing-Tabellen
- ✓ bessere Integration von IPsec (Sicherheit, Datenschutz)
- ✓ Integration neuer Dienstfunktionen (Dienstgüte für Echtzeitanwendungen)
- ✓ Bessere Unterstützung für mobile Geräte
- ✓ Unterstützung von Multicasting
- ✓ größere Flexibilität für zukünftige Anpassungen

### Neue Adresslänge

Die signifikanteste Neuerung ist die Erweiterung des Adressraums von **32 Bit** in IPv4 auf **128 Bit** in IPv6.

Können in IPv4 nur

$$2^{32} = 4.294.967.296$$

Adressen abgebildet werden, die zudem nicht alle frei verfügbar sind, kann IPv6

$$2^{128} = 3,4 \times 10^{38} = 340.282.366.920.938.463.463.374.607.431.768.211.456$$

Adressen erreichen.

Das ermöglicht es theoretisch, jedem im Jahr 2020 lebenden Menschen auf der Erde ca.  $43.861.237.523.414.200.000.000.000.000$  (ca.  $4,3 \times 10^{28}$ ) IP-Adressen zuzuweisen. In der Praxis wird der Adressraum zur effektiven Umsetzung des neuen Protokolls genutzt (nach UN World Population Prospects, the 2010 Revision).

## Vereinfachter IP-Header

Der Header hat bei IPv6 eine feste Länge. Optionen werden durch Erweiterungsheader realisiert. Dadurch können IPv6 Datenpakete effektiver von Routern verarbeitet werden. Der Router muss meist nur den ersten Header auswerten, um eine Routingentscheidung zu treffen.

Der Header kann in seiner **Form** – je nach Paketgröße und -inhalt – variabel gestaltet werden.

Um die steigende Last auf den Routern und Switches zu reduzieren, wurde der IP-Adressheader von 13 auf 7 Felder gekürzt und um Funktionen wie die **Identification** oder die **Header Checksum** reduziert.

## Erweiterungen im Adressbereich

IPv6 verwendet die in IPv4 bekannten **Broadcasts** nicht mehr. Stattdessen unterstützt es von sich aus **Multicast**. Durch die Multicast-Adressen kann eine Gruppe lokaler Geräte angesprochen werden, somit wird die Netzlast reduziert. Um diese Funktion zu gewährleisten, muss jedes Gerät, das IPv6 unterstützt, auch Multicast auf IPv6 unterstützen.

Das Verbinden von mehreren Maschinen zu einem Netzwerk wird durch den neuen Mechanismus **Autokonfiguration** deutlich vereinfacht, da dieser für die Maschine automatisch eine gültige Adresse erzeugt. Hierdurch erübrigtsich in den meisten Fällen das manuelle Einrichten der IPs, oft wird kein DHCP-Server mehr benötigt.

**Anycast** ist direkt in das Protokoll integriert. Anycast erlaubt es, die Routen automatisiert zu erstellen. Es ist nicht mehr notwendig, die Routen von Hand anzulegen, da IPv6 autonom eine Routing-Tabelle konfiguriert und diese aktualisiert.

Neben globalen Unicast-Adressen gibt es bei IPv6 Link-Lokale-Adressen, die nur im Netzabschnitt gültig sind, sowie Unique-Lokale-Adressen, die intern, aber nicht ins Internet geroutet werden. Sie sind mit den privaten IPs bei IPv4 vergleichbar.

## Mobile Datenkommunikation und Multihoming

### Mobile IP

Durch den Einsatz von **Mobile IP** in IPv6 ist es möglich, mit mobilen Geräten ohne aufwendige Routing-Konfigurationen an verschiedenen Standorten im Datennetz erreichbar zu sein (vgl. Abschnitt 10.1). (Das Konzept gibt es in ähnlicher Form auch bei IPv4.)

### Multihoming

Um die Performance und die Ausfallsicherheit von Internetverbindungen aus dem privaten Netzwerk heraus zu steigern, können **Border Gateway Protocol Router** (BGP-Router, z. B. Linux-Server mit Quagga, Cisco-Router) eingesetzt werden, die eine Verbindung mit mehreren Internet-Providern ermöglichen.

Spezielle **providerunabhängige Adressen** (Provider Independent Addresses; PI-Addresses; reservierte IPv6-Adressräume, die dem Endnutzer direkt von der RIR zur Verfügung gestellt werden und somit unabhängig von der Auswahl des ISP sind) ermöglichen unterschiedliche Techniken, um die Anbindung an das Internet zu stabilisieren.

Providerunabhängige Netze waren bei IPv6 zunächst nicht vorgesehen, da sie dem Ziel des vereinfachten Routings entgegenstehen. Im April 2009 wurde ein entsprechender Vorschlag vom RIPE umgesetzt (RIPE-481). Für PI-Netze steht nun der Präfix 2001:678::/29 zur Verfügung. Die kleinstmögliche Netzgröße ist /48.

So ist es z. B. möglich,

- ✓ bei einer erhöhten Netzwerkauslastung ab einer bestimmten Bandbreitenbelastung einen weiteren ISP temporär hinzuzuschalten,
- ✓ unterschiedliche Dienste über verschiedene ISP abzuwickeln,
- ✓ Upload und Download über verschiedene ISP zu realisieren,
- ✓ dass der Ausfall einer ISP-Verbindung für die Nutzer nicht zur Abkopplung vom Internet führt.

### Das Ende-zu-Ende-Konzept

Keine Neuerung, sondern eine Rückbesinnung auf ursprüngliche Prinzipien eines Netzwerkes stellt das sog. **Ende-zu-Ende-Konzept** dar. Dies soll in IPv6 als ein zentrales Designelement des Internets wiederhergestellt werden. Das Ende-zu-Ende-Konzept besagt, dass jedes Gerät direkt mit dem Partner kommuniziert. Dies ist jedoch durch die Adressknappheit bei IPv4 nicht möglich, da nicht mehr für jedes Gerät eine eigene IP-Adresse zur Verfügung steht. Deshalb wird bei IPv4 NAPT (Network Adress and Port Translation) oft nicht ganz korrekt als NAT bezeichnet eingesetzt, das als „öffentliche“ IP-Adresse für mehrere Endgeräte in einem lokalen Netz fungiert. NA(P)T verletzt damit jedoch das Ende-zu-Ende-Konzept, da NA(P)T als Instanz zwischen den anzusprechenden Endgeräten die Pakete verändert.

Netzkomponenten sollen laut dem Ende-zu-Ende-Prinzip Datenpakete jedoch nur weiterleiten, um die Kosten bei den Providern gering zu halten. Gleichzeitig benötigen die Netzgeräte weniger „Intelligenz“, um die Pakete weiterzuleiten (Preis-/Komplexitätsargument). Das sorgt wiederum für geringere **Latenzzeiten** (Antwortzeiten zwischen zwei Geräten).

### Vereinfachtes Routing

Durch den größeren Adressraum ist eine strukturierte Vergabe von IP-Blöcken möglich. Es kann eine Hierarchie abgebildet werden, die das globale Routing auf die ersten Bytes der IPv6-Adresse beschränkt. Das erlaubt die Aggregation (zusammenfassen) von Routen und somit kürzere Routingtabellen. Dadurch ist ein effizienteres Routing (Faktor >10) möglich (vgl. Abschnitt 4.4).

### Integriertes IPsec

**IPsec** ist ein etabliertes Verfahren, um Dokumente und Daten zu authentifizieren und zu verschlüsseln. IPsec war ursprünglich fester Bestandteil von IPv6. Im Laufe der Entwicklung wurde IPsec wie schon bei IPv4 zur Option.

### Verbesserte Unterstützung von Dienstarten

IPv6 bietet eine verbesserte Unterstützung von Video- und Audio-Übertragungen. Dazu ist eine Option zur Echtzeitübertragung vorhanden.

### Exkurs IPv5

Das Protokoll IPv5 (das diesen Namen offiziell niemals getragen hat - er wurde nachträglich durch die IANA zugewiesen, um eine Kontinuität in der Zählung herzustellen) fungierte als experimentelles Protokoll für Echtzeit-Datenströme und trug die offizielle Bezeichnung ST-2 (Internet Stream Protocol Version 2, definiert in RFC 1819). ST-2 wurde vom Protokoll RSVP (Resource Reservation Protocol) zur Bandbreitenanforderung bei Routern abgelöst. Ursprünglich sollte ST-2 zur Übertragung von Audio- und Video-Daten per Multicast genutzt werden. Die Serienreife und ein praktischer Einsatz für das Protokoll wurden jedoch niemals erreicht. Die Entwicklung wurde schließlich aus Kosten-Nutzen-Erwägungen zugunsten der Protokolle IPv6 und RSVP eingestellt.

### 3.2 Gegenüberstellung von IPv4 und IPv6

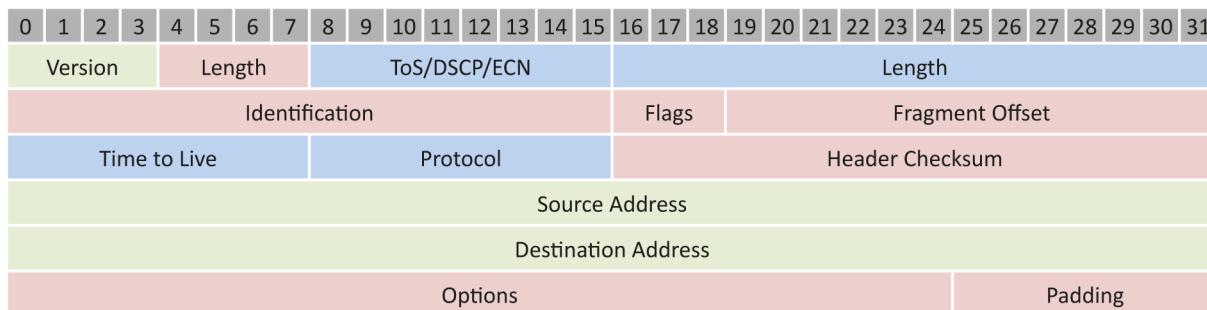
IPv4	IPv6
Adresslänge von 32 Bit	Adresslänge von 128 Bit
Nutzt DNS (A) Einträge, um Namen aufzulösen	Nutzt DNS (AAAA) Einträge, um Namen aufzulösen
IPsec ist optional	IPsec ist im Protokoll integriert
TOS-Feld im Header kann für QoS genutzt werden	Zusätzliches Flow Label
Host und Router können Pakete fragmentieren	Nur der Host kann Pakete fragmentieren, Router leiten nur noch weiter
Checksumme im Header	Keine Checksumme im Header
Header unterstützt Options	Options nur über eine Erweiterung des Headers nutzbar
Beziehung MAC-Adresse zu einer IP-Adresse wird mit ARP ermittelt.	Nutzt eine Multicast-Nachbar-Suche, um die MAC-Adresse zu bekommen
Internet Group Management Protocol (IGMP) zum Verwalten der Subnetz-Mitgliedschaft	Multicast Listener Discovery (MLD)-Nachrichten zum Verwalten des lokalen Subnetzes
Verwendet Broadcast-Adressen, um an alle Mitglieder eines Subnetzes zu senden	Link-Lokale Scope All Multicast-Adressen, um jeden Host im Subnetz zu erreichen
Konfiguration manuell oder per DHCP	Konfiguration über SLAAC, statisch und DHCPv6
Muss mindestens ein 576 Byte großes Paket unterstützen, welches fragmentiert werden darf	Muss mindestens ein 1280 Byte großes Paket unterstützen; keine Fragmentierung

### 3.3 Vergleich der Header IPv4 und IPv6

#### Änderungen am Header

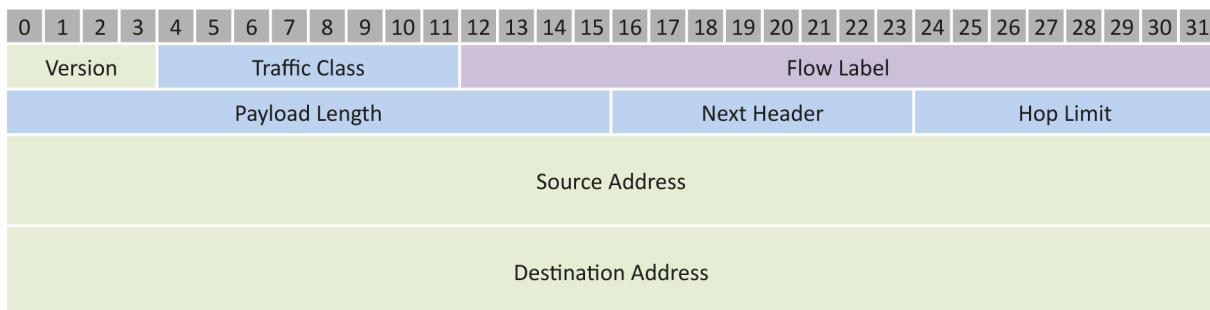
Einige Felder aus dem IPv6-Header wurden entfernt, um ihn kleiner gestalten zu können und somit die Netzwerkhardware zu entlasten. Gleichzeitig gibt es neue Felder, um den erweiterten Ansprüchen im Dienstesektor gerecht zu werden. Diese Modifizierungen bewirken die Verschiebung von Feldern, die zudem noch umbenannt wurden.

#### IPv4 Header



IPv4 Header

## IPv6 Header



IPv6 Header

## Änderungen bei den Feldern

### ✓ Entfernte Felder

Length (Länge)	Gibt die Gesamtlänge des Headers an
Identification (Identifikation)	Wird dazu genutzt, um fragmentierte Pakete wieder zusammenfügen zu können
Flags (Statusindikatoren)	Dient dazu, Fragmente zu kontrollieren oder zu identifizieren
Fragment Offset (Fragmentierungskennzeichnung)	Information darüber, wie viele Fragmente zu einem Paket gehören und in welcher Reihenfolge sie zusammengesetzt werden müssen
Header Checksum (Header Kontrollwert)	Kontrolliert, ob im Header ein Fehler aufgetreten ist

### ✓ Verschobene und/oder umbenannte Felder

Type of Service (Servicetyp)	Wurde in „Traffic Class“ umbenannt und durch den Wegfall des „Header Length“-Feldes um eine Position nach vorne verschoben
Total Length (Gesamte Länge)	Wurde in „Payload Length“ umbenannt und steht an vierter Stelle; gibt die Größe der Nutzdaten wider
Next Header (Nächster Kopfteil)	Steht an fünfter Stelle. Es identifiziert die Art des übergeordneten Protokolls oder die Art des Erweiterungshanders. (Siehe unten)
Time to Live (Lebensdauer)	Wurde in „Hop Limit“ umbenannt. Es sorgt dafür, dass Pakete nach einer bestimmten Anzahl von Weiterleitungen verworfen werden.

### ✓ Hinzugefügtes Feld

Flow Label (Flussnummer)	Dieses Feld wird für QoS (Quality of Service) und Echtzeitanwendungen benötigt. Pakete mit demselben Wert in diesem Feld werden gleich behandelt.
--------------------------	---

Folgende Erweiterungsheader sind nach RFC 8200 definiert (vormals RFC 2460):

Next Header *	Name	Beschreibung
0	Hop-by-Hop Options	Optionen für Router, muss von allen Routern beachtet werden
60	Destination Options	Optionen für den Zielhost
43	Routing	Angabe einer bestimmten Route
44	Fragment	Wenn das Paket fragmentiert ist, werden die Informationen in diesem Header angegeben
51	AH	Authentication Header, für <b>IPsec</b> benötigt
50	ESP	Encrypted Security Payload, für <b>IPsec</b> benötigt
59	Upper-Layer No-Next-Header	Header höherer Protokollschichten (z. B.: Tunnel)

\* Die Dezimalwerte für Next-Header/Protokollnummern sind bei der IANA unter <https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml> aufgelistet.

Alle Systeme müssen in der Lage sein, die oben genannten Header in beliebiger Reihenfolge zu verarbeiten, wobei die obige Reihenfolge in RFC 8200 dringend empfohlen (strongly recommended) wird. Ausnahme ist der Hop-by-Hop Options Header, der immer direkt nach dem IPv6 Header kommen muss. Alle Erweiterungsheader dürfen nur einmal vorkommen, Ausnahme ist der Destination Options Header, der zweimal erscheinen sollte, einmal vor dem Routing Header und einmal vor dem Upper-Layer Header.

Die Kombination aus optimiertem Header, insbesondere Wegfall von Längenfeld und Prüfsumme, sowie fester Länge von Header und Erweiterungsheadern bewirken gegenüber IPv4 eine schnellere Verarbeitung durch die beteiligten Systeme und insgesamt einen Performancegewinn.

# 4 Aufbau des Adressraums von IPv6

## In diesem Kapitel erfahren Sie

- ✓ wie die Adresszuweisung im IPv6-Adressraum funktioniert
- ✓ wie IPv6-Adressbereiche aufgeteilt werden

## Voraussetzungen

- ✓ Überblick über das IPv4-Protokoll
- ✓ Grundlagen des IPv6-Protokolls
- ✓ Adressaufbau von IPv6

## 4.1 Der Adressaufbau von IPv6

Die Adressierung erfolgt bei IPv6 nach dem gleichen Grundschema wie bei IPv4. Die wichtigsten Änderungen beim IPv6 Adressaufbau sind:

- ✓ Feste Länge der Host-ID (wenige Ausnahmen).
- ✓ Adressen sind einmalig und identifizieren einzelne Teilnehmer oder Teilnehmergruppen eindeutig.
- ✓ Analog zur Subnetznummer im IPv4 gibt es das IPv6-Präfix. Anhand des Präfixes kann ein sendewilliger Teilnehmer erkennen, ob sein Ziel direkt erreichbar ist oder die Weiterleitung über einen Router erforderlich ist.
- ✓ Die klassenbezogene oder klassenlose Subnetzmaske einer IPv4-Adresse identifiziert die zur Subnetznummer gehörenden Bits der Adresse. Gleichermaßen zählt die Präfixlänge im IPv6 die zum Präfix gehörenden Bits ab.

Im IPv4 kennzeichnen die verbleibenden Bits, die nicht zur Subnetznummer gehören, den Host. Bei IPv6 bilden die nicht zum Präfix gehörenden Bits den **Interface-Identifier** (Interface-ID).

## 4.2 Adressnotation

### Standard-Notation

Die Adresslänge einer IPv6-Internetadresse ist von 32 auf 128 Bit vervierfacht worden.

```
0010000000000001:0000110010101000:000000111111110:0000000000000000:  
0000000100000000:0000000000000000:0000000000000000:000000000000111
```

Um nicht 128-stellige Dualzahlen handhaben zu müssen, werden für eine bessere und übersichtlichere Lesbarkeit

- ✓ jeweils vier aufeinander folgende Bits zu einem Hexadezimalwert zusammengefasst.  
Daraus ergeben sich 32 hexadezimale Ziffern;

... 1000:0000 0011 1111 1110:0000 ...

2001:0db8:03fe:0000:0100:0000:0000:0007

- ✓ dann werden je 4 hexadezimale Ziffern zu einer Gruppe zusammengefasst. Zwischen den 8 Gruppen werden zur besseren Lesbarkeit Doppelpunkte notiert.

... 1000:0000 0011 1111 1110:0000 ...

2001:0db8:03fe:0000:0100:0000:0000:0007

Das Hexadezimalsystem unterscheidet nicht zwischen Groß- und Kleinschreibung. Im Allgemeinen wird für IPv6-Adressen Kleinschreibung verwendet.

Die Darstellung einer IPv6-Adresse sieht dann folgendermaßen aus (nach RFC 5952):

2001:0db8:03fe:0000:0100:0000:0000:0007

Zur weiteren Vereinfachung dürfen führende Nullen in einer Gruppe entfallen:

2001:0db8:03fe:0000:0100:0001:0001:0007

↓

2001: db8: 3fe: 0: 100: 1: 1: 7 ⇒ 2001:db8:3fe:0:100:1:1:7

Schließlich können zwei oder mehrere aufeinander folgende Gruppen, die alle nur Nullen enthalten, mitsamt den dazwischen stehenden Doppelpunkten entfallen. Diese Regel kann nur einmal angewendet werden, andernfalls wäre die ursprüngliche Ziffernfolge nicht wiederherstellbar. In unserem Beispiel muss man auf

2001:0db8:03fe:0000:0100:0000:0000:0007

↓

2001: db8: 3fe: 0: 100: : 7 ⇒ 2001:db8:3fe:0:100::7

reduzieren.

Sollten sich in einer IPv6-Adresse mehrere Sequenzen von aufeinander folgenden 4-hexadezimal-Feldern befinden, die nur mit Nullen gefüllt sind, sollten,

wenn die Sequenzen gleich lang sind, die **erste**

2001:0db8:0000:0000:0100:0000:0000:0007

↓

2001: db8: : 100: 0: 0: 7 ⇒ 2001:db8::100:0:0:7,

wenn die Sequenzen ungleich lang sind, die **längere**

2001:0000:0000:03fe:0000:0000:0000:0007

↓

2001: 0: 0: 3fe: : 7 ⇒ 2001:0:0:3fe::7

ersetzt werden.

**Von dieser Notation gibt es eine Ausnahme:** Gelegentlich müssen Hosts, die nur IPv4 unterstützen, mithilfe einer **Übersetzung** (vgl. Abschnitt 7.3) angesprochen werden können. Für diese wurde ein eigener Adressbereich mit dem Präfix 0:0:0:0:ffff::/96 reserviert (bis 2006 und nach RFC 4291 veraltet gab es auch den Bereich ::/96), der in den verbleibenden 32 Bit die Notation der IPv4-Adresse in der dezimalen Schreibweise gestattet. Beispielsweise kann ein Teilnehmer mit der IPv4-Adresse 192.168.2.1 mithilfe eines übersetzenden Routers über die zugeordnete IPv6-Adresse erreicht werden.



0:0:0:0:ffff:192.168.2.1

verkürzt dargestellt als

::ffff:192.168.2.1

Die Darstellung in der hexadezimalen Form ist auch zulässig: ::ffff:c0a0:0201

## URL-Notation

Wie bei IPv4 besteht bei IPv6 die Möglichkeit, Internetdienste in entsprechenden Medien (Internet-Browser, FTP-Client, Remote Access) nicht nur über die zugeordnete URL, sondern nach einem definierten Schema direkt über die IP-Adresse, ggf. mit entsprechenden Zusätzen wie z. B. Portnummern oder Zugangsdaten, anzusprechen. In der hier verwendeten **URL-Notation (Uniform Resource Locator)** wird der Doppelpunkt als Trennzeichen verwendet. Man findet ihn bei der Verwendung von IPv4-Adressen u. a.

- ✓ zwischen der IP-Adresse und dem IP-Port, um direkt einen Socket anzugeben  
Beispiel: **http://192.168.2.1:8080**
- ✓ zur Abgrenzung zwischen Benutzername und Kennwort  
Beispiel: **ftp://mueller:12345@192.168.2.1**
- ✓ nach der Schema-Bezeichnung  
Beispiel: **telnet:192.168.2.1**

Die direkte Verwendung von IPv6-Adressen ist wegen der Doppelpunkte in der URL-Notation nicht möglich und würde zu Fehlinterpretationen führen. Daher werden IPv6-Adressen mit eckigen Klammern umgeben und sind dann eindeutig als Adressanteil der URL-Notation gekennzeichnet. Die oben genannten Beispiele sehen mit IPv6 statt IPv4-Adressen so aus:

- ✓ **http://[2001:db8:3fe:0:100::7]:8080**
- ✓ **ftp://mueller:12345@[2001:db8:3fe:0:100::7]**
- ✓ **telnet:[2001:db8:3fe:0:100::7]**

## Netz-Notation

Die Netz-Notation wird verwendet, um zusammenhängende Bereiche von Netzadressen darstellen zu können. Dabei wird die niedrigste Adresse des Netzbereiches verwendet, um das Präfix zu formulieren. Dieser wird die Anzahl der von vorne identischen Bits als sogenannte Präfixlänge angefügt.

**Beispiel:** Der Netzbereich **2001:0db8:03fe:0000:0000:0000:0000:0000** bis **2001:0db8:03fe:0000:ffff:ffff:ffff:ffff** hat das Präfix **2001:db8:3fe:0000**. Dies ist der Bereich, in dem die erste (Startadresse) und die letzte Adresse (Endadresse) des Bereichs (und natürlich auch alle dazwischenliegenden Adressen) übereinstimmen. In diesem Fall sind dies die ersten 64 Bit. Diese Präfixlänge wird mit einem Slash an die IP-Adresse angehängt und zeigt den Anteil des Netzbereiches an der Gesamtadresse an. Die Netz-Notation für diesen Netzbereich lautet somit **2001:db8:3fe::/64**.



Beachten Sie, dass bei der Netz-Notation zwischen der IP-Adresse und der Präfixlänge kein Leerzeichen eingefügt ist.

Diese Notation ist auch als **CIDR-Notation (Classless Inter-Domain Routing)** aus dem IPv4 bekannt. Eine Darstellung mit einer dezimalen Netzmaske, wie im IPv4 üblich, findet keine Anwendung.

## 4.3 MAC-Adresse, EUI-64 und Interface ID

### Forderungen aus RFC 4291 (Update RFC 7136 aus 2014)

Teilnehmer in IPv6-Netzen sollen grundsätzlich eine 64 Bit lange Interface ID erhalten; so fordert RFC 4291: „*64 Bit lange Schnittstellen-IDs sind erforderlich und werden aus einem modifizierten EUI-64-Format hergeleitet.*“ Und weiter: „*Sofern eine IEEE 802 48-Bit MAC-Adresse vorhanden ist, soll diese wegen ihrer Einmaligkeit verwendet werden.*“

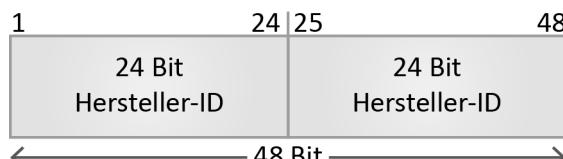
In einem ersten Schritt wird aus der MAC-Adresse das EUI-64-Format abgeleitet.

### MAC-Adresse

Eine **MAC-Adresse** wird weltweit eindeutig für einen Netzwerkadapter vergeben. Sie ist 48 Bit lang.

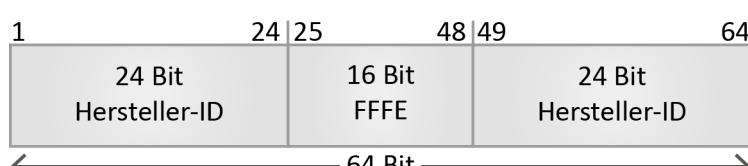
- ✓ In der Hardware fest codiert ist zunächst eine 24-Bit-**Herstellerkennung**, die von der IEEE vergeben wird.
- ✓ Die restlichen 24 Bit werden als **fortlaufende** Gerätenummer direkt vom Hersteller für jeden einzelnen Netzwerkadapter vergeben.

Mit diesem Verfahren wird die weltweite Einmaligkeit der MAC-Adressen gewährleistet.



### EUI-64

Um nun zum EUI-64-Format zu gelangen, wird zwischen diese beiden Blöcke die **hexadezimale Folge FFFE** eingesetzt.

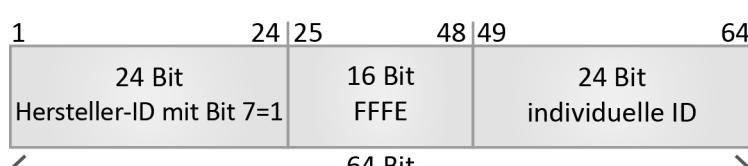


**In einem zweiten Schritt wird dann aus dem EUI-64-Format die Interface-ID abgeleitet.**

### Interface-ID

Sowohl in der MAC-Adresse als auch in der abgeleiteten EUI-64-Formatierung ist laut Definition das 7. Bit der Hersteller-ID, das **U-Bit (universal/local)**, immer auf 0 gesetzt.

In dem Schritt vom EUI-64-Format zur Interface-ID wird dieses 7. Bit nun auf 1 gesetzt und ergibt so die Interface-ID.



#### Beispiel:

- ✓ Zu der MAC Adresse 10-1F-74-B7-49-47
- ✓ gehört die EUI-64-Darstellung 10-1F-74-FF-FE-B7-49-47
- ✓ und die Interface-ID 12-1F-74-FF-FE-B7-49-47

Gemäß RFC 4291 soll diese Interface-ID zusammen mit einem 64 Bit langen Prefix als IP-Adresse verwendet werden. Und so verfahren auch die meisten Betriebssysteme. Besonderheiten bei Microsoft siehe weiter unten.

Diese Interface-ID ist jedoch immer gleich, egal an welchem Ort das Gerät mit diesem Netzwerkadapter jemals betrieben wird. Anders als bei IPv4, bei dem in fast allen Netzen NAT die tatsächliche Identität des Teilnehmers verschleiert, tritt ein IPv6-Teilnehmer immer mit der gleichen „Interface-ID“-Identität auf. Von den verwendeten IPv6-Adressen muss nur das 64-Bit-Präfix entfernt werden, und schon kann mit entsprechenden Analysetools über jeden Teilnehmer ein Nutzungsprofil angefertigt werden.



Um diese Nachverfolgbarkeit zu umgehen, können alternativ zufällig generierte Host-IDs, „Privacy Extensions“ nach **RFC 4941** genutzt werden. In der Tat verwenden z. B. Windows-Betriebssysteme ab den Versionen

Windows 7 und Windows Server 2008 standardmäßig zufällig erzeugte Host-IDs. Mit dem Befehl

```
netsh interface ipv6 set global randomizeidentifiers=disabled
```

kann man die Erzeugung der Interface-ID nach dem modifizierten EUI-64-Format auf Basis der MAC-Adresse erzwingen.

Der Befehl

```
netsh interface ipv6 set privacy state=disabled
```

aktiviert zusätzlich regelmäßig wechselnde Host-IDs.

## 4.4 IPv6-Adresszuweisung durch Regional Internet Registry

Die Adresszuweisung im IPv6-Protokoll ist ein Zusammenspiel zwischen einer Regional Internet Registry (RIR) und einem Internet Service Provider (ISP), der dann an den Endkunden die IPv6-Adresse oder den IPv6-Adressbereich weiter vergibt. In diesem Abschnitt lernen Sie, wie die Adresszuweisung im IPv6-Adressbereich funktioniert.

### Zuweisungspraxis

Die Länge einer IPv6-Adresse beträgt immer 128 Bit. Diese werden in Bereiche mit unterschiedlichen Funktionen unterteilt, wobei sich die Aufteilung aus Sicht des Providers in zweifacher Hinsicht beschreiben lässt:

- ✓ hinsichtlich der Vergabe der IPv6-Adresse an den Kunden
- ✓ hinsichtlich des Erhalts der IPv6-Adresse durch die RIR, für Europa RIPE (siehe S. xx)

### Vergabe der IPv6-Adresse an den Kunden

Von den 128 Bit bilden die ersten 64 Bit das sog. Präfix, das aus zwei Bestandteilen zusammengesetzt ist: dem Global Routing Präfix (Bezeichnung: N-Bit) und der Subnet-ID (Bezeichnung: M-Bit).

N-Bit	M-Bit	64 Bit
Global Routing Präfix	Subnet-ID	Interface-ID

Einen N-Bit langen Global Routing Präfix vergibt ein ISP (Internet Service Provider) an einen Kunden, dem die restlichen (64 - M) Bit der Subnet-ID für die Segmentierung von internen Netzen zur Verfügung stehen. In aller Regel wird der Global Routing Präfix mit 48 Bit oder 56 Bit fest vergeben, sodass dem Kunden 16 Bit bzw. 8 Bit für die Bildung eigener Teilnetze zur Verfügung stehen. Der zweite Teil der IPv6-Adresse besteht aus den letzten 64 Bit und bildet den eindeutigen Interface Identifier für die Netzwerkschnittstelle. Dieser 64 Bit lange Interface Identifier kann wie beschrieben aus der MAC-Adresse der Netzwerkschnittstelle erstellt werden, um die Eindeutigkeit der Adresse zu gewährleisten.

### Erhalt der IPv6-Adresse durch die RIR

Normalerweise bezieht ein Internet Service Provider (ISP) den ersten Teil einer IPv6-Adresse von einem **Regional Internet Registry (RIR)**. „Normalerweise“ deshalb, weil er sie auch von einem übergeordneten ISP erhalten kann. So beziehen beispielsweise in Deutschland einige große Provider ihre Adressbereiche von der Telekom und nicht direkt über die RIR. Bei dem zugewiesenen Adressbereich handelt es sich um die maximal ersten 32 Bit der IPv6-Adresse, die als Netzbereich zugewiesen werden. Dieser Adressbereich wird vom ISP in weitere Subnetze aufgeteilt, wobei die Länge der Zuteilung an Endkunden dem ISP überlassen ist. Der ISP muss nur darauf achten, dass er die minimale Zuweisung eines Netzes einhält. Diese beträgt in diesem Fall 64 Bit. Informationen über die Vergabe von IPv6-Netzen können über die Whois-Dienste der jeweiligen RIRs abgefragt werden.

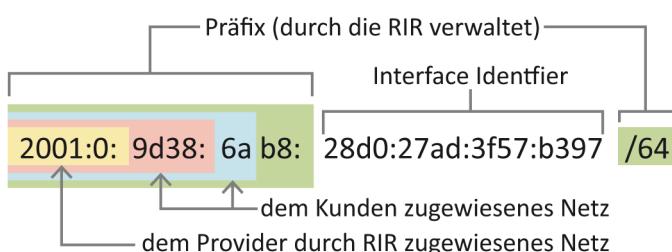
**Beispiel:** Elemente der IP-Adresse 2001:0:9d38:6ab8:28b027ad:3f57:b397

Verbindungsspezifisches DNS-Suffix:

IPv6-Adresse . . . . . : 2001:db8:d700:0:aca4:8a7:f88:f2d9

Verbindungslokale IPv6-Adresse . : fe80::aca4:8a7:f88:f2d9%21

Auszug aus ipconfig



Vom ISP zugewiesene Kundenadresse

Adresse aus dem dem Kunden vom ISP zugewiesenen Netzwerkbereich:	<b>2001:0:9d38:6ab8:28d0:27ad:3f57:b397/64</b>
Die ersten 64 Bit bilden das Präfix:	<b>2001:0:9d38:6ab8::/64</b>
Die letzten 64 Bit bilden den Interface Identifier:	<b>xxxx:xxxx:xxxx:xxxx:28d0:27ad:3f57:b397/64</b>
Dem Provider wurde von der Regional Internet Registry (RIR) folgendes Netz zugewiesen:	<b>2001:0::/32</b>
Der Endkunde erhält vom Provider z. B. das Netz:	<b>2001:0:9d38::/48</b>
... oder auch ein kleineres Netz:	<b>2001:0:9d38:6a00::/56</b>

### Regional Internet Registry (RIR)

Weltweit sind fünf Regional Internet Registries (RIR) für die Vergabe von IP-Adressen zuständig.

Regional Internet Registry	Zuständiger Bereich
AfriNIC	Afrika
APNIC	Asien & Pazifik-Regionen
ARIN	Kanada, USA sowie einzelne Karibik-Inseln
LACNIC	Lateinamerika sowie einzelne Karibik-Inseln
RIPE NCC	Europa, Mittlerer Osten, Zentralasien

Das Réseaux IP Européens Network Coordination Centre (RIPE NCC) ist die RIR, die mit Sitz in Amsterdam für die Vergabe von IP-Adressbereichen in Europa, dem Nahen Osten und Zentralasien zuständig ist.

RIPE NCC ist dem RIPE untergeordnet, das die Handlungsrichtlinien des Gremiums festlegt. ICANN ist für alle Fragen zu IP-Adressbereichen zuständig. Alle vorhandenen Local Internet Registries (LIR) sind nach Ländern geordnet auf der Website der RIPE NCC aufgeführt. Möchte man als Firma, Privatperson oder sonstige Organisation eine IP-Adresse erwerben, muss man sich an eine LIR wenden oder sogar selbst Mitglied bei RIPE werden. Da RIPE eine Non-Profit-Organisation ist, finanziert sich RIPE NCC aus Mitgliedsbeiträgen der angeschlossenen Internet Service Provider, Hochschulen und Großunternehmen der IT-Branche und anderer Interessenten.

RIPE NCC ist darüber hinaus für die Datenbank verantwortlich, die die von RIPE NCC ausgegebenen IP-Adressbereiche enthält. Jeder zugeordnete Adressbereich kann mit dem Whois-Dienst, der über die Website zugänglich ist, ermittelt werden. Hier sind Informationen wie Besitzer, Ansprechpartner und deren Anschriften, E-Mail-Adressen und Telefonnummern aufgeführt.



*Regionale Internet Registrare Weltkarte (Quelle: [www.iana.org/numbers](http://www.iana.org/numbers))*

Im Oktober 2003 wurde seitens der RIRs die Number Resource Organization (NRO) gegründet. Diese ist für die Kommunikation und interne Koordination zuständig und vertritt die Interessen der RIRs gegenüber ICANN.

## 4.5 Adressbereiche

Genau wie bei IPv4 gibt es bei IPv6 Adressbereiche, die besonderen Netzwerkfunktionen zugeordnet sind. Handelte es sich bei IPv4 um lokale und experimentell zu verwendende Adressbereiche (192.168. ...), die später, z. B. über NAT, verwendet wurden, um der Adressknappheit temporär zu entgehen, definiert IPv6 mehrere spezifizierte Adressbereiche, die jeweils unterschiedlichen Zwecken dienen, hauptsächlich der erleichterten Adresszuweisung sowohl in größeren als auch in kleinen, lokalen Netzsegmenten. Folgende Adressbereiche werden unterschieden:

### Allgemeine Adressbereiche

- ✓ Link-Local-Adressen
- ✓ Unique-Local-Adressen
- ✓ Multicast
- ✓ Site-Local-Adressen
- ✓ Global-Unicast-Adressen
- ✓ Anycast

### Besondere Adressbereiche

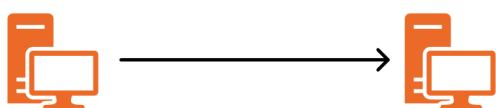
- ✓ Unspecified Address
- ✓ Loopback Address
- ✓ Kompatibilitätsadressen

## 4.6 Allgemeine Adressbereiche

### Link-Local-Adressen

Verbindungslokale Unicast-Adressen (vgl. nachfolgende Erläuterungen), die auf einer physischen Verbindung (Link) eindeutig sind, werden als Link-Local-Adressen bezeichnet. Diese können allerdings global mehrfach vorkommen. Wenn IPv6 auf einem Computer aktiviert ist, verwendet dieser für jede Netzwerkschnittstelle eine automatisch zugewiesene verbindungslokale Adresse, die der Computer aus dem Präfix und der MAC-Adresse der Netzwerkschnittstelle ermittelt. IPv6 nutzt Link-Local-Adressen zur automatischen Einrichtung von Netzwerken. Der Host erhält mittels der verbindungslokalen Adresse Informationen darüber, ob andere IPv6-Hosts bzw. -Router im lokalen Netz vorhanden sind (RFC 4291).

Die Link-Local-Adressen werden innerhalb abgeschlossener Netzwerksegmente verwendet. Man identifiziert die Link-Local-Adressen über den Wert **fe80:0000:0000:0000** oder **fe80:: /64**. Auch hier werden, wie bereits ausgeführt, die ersten 64 Bit des Subnetz-Präfix für die Identifizierung verwendet.



*IPv6-Kommunikation über Link-Local-, Unique-Local-Unicast- oder Global-Unicast-Adresse*

Link-Local-Adressen haben folgendes Format:

10 Bit	54 Bit	64 Bit
1111111010	0	Interface-ID

Link-Local-Adressen werden nicht nur für die Adressierung von Netzwerkschnittstellen in abgeschlossenen Netzwerksegmenten verwendet, sondern auch für die Neighbor Discovery- oder Autokonfiguration. Dank der Einführung von Link-Local-Adressen wird der DHCP-Server in einem Netzwerksegment für die dynamische Adressvergabe überflüssig. Der Link-Local-Adressbereich in IPv6 ist vergleichbar mit APIPA-Adressen unter IPv4, die durch den Wert 169.254.0.0/16 repräsentiert sind.

Wenn ein Netzwerkgerät mit einer Link-Local-Adresse kommunizieren soll, ist es notwendig, dass die Zone-ID immer angegeben sein muss. Die Zone-ID definiert hier den genauen Bereich, die Netzwerkschnittstelle, über die Datenverkehr gesendet bzw. empfangen wird.

Wenn auf einem Host IPv6 aktiviert ist, besitzt das Hostsystem pro Netzwerkschnittstelle eine einzelne Link-Local-Adresse, die aus Präfix und MAC-Adresse der Netzwerkschnittstelle ermittelt werden. Durch die Link-Local-Adresse bezieht der Host die Information, ob andere IPv6-Netzwerkgeräte verfügbar bzw. erreichbar sind. Bei einer einzelnen Netzwerkschnittstelle könnte die Link-Local-Adresse sich z. B. wie folgt darstellen:

Verbindungslokale IPv6-Adresse . : fe80::0029:75ff:fe9d:3f59%12  
*Auszug aus ipconfig*

Bei Link-Local- und Unique-Local-Adressen gibt Windows eine zusätzlich Scope-ID an, im Beispiel die %13. Die 13 ist dabei die intern vergebene Interface-ID.

Wenn mehrere Netzwerke in einem Host eingerichtet sind, bekommt jede Netzwerkschnittstelle eine Link-Local-Adresse, beginnend mit FE80::. Wenn der Host nun eine Verbindung zu einer anderen Link-Local-Adresse aufbauen will, kann er allein aufgrund der Adresse keine Routing-Entscheidung treffen, da der Netzwerkteil der Adressen identisch ist.

Angenommen, in jedem der angeschlossenen Netzwerke gibt es einen Router mit der IP FE80::1. Die Konnektivität soll überprüft werden.

- ✓ ping -6 FE80::1 würde nicht funktionieren, da nicht klar ist, welche Schnittstelle benutzt werden soll.

In diesem Fall muss die Scope-ID an die Anfrage angehängt werden.

- ✓ ping -6 FE80::1%13

Damit wird für die Verbindung auf dem Host die Schnittstelle 13 gewählt, und der Ping wird zum Zielrechner geschickt. Unter Linux wird in solchen Fällen der Schnittstellenname angegeben: ping6 -I eth0 FE80::1

### Site-Local- Adressen (obsolete)

Site-Local- Adressen sind für den IPv6-Adressbereich FEC0::/10 von der IETF (Internet Engineering Task Force) reserviert; RFC 3513 (ersetzt durch RFC 4291) definierte den Bereich FEC0::/10 als site-lokale Adressen, die ähnlich wie die Link-Local- Adressen nur in abgegrenzten Netzwerkbereichen (Sites) verwendet werden sollten. Der Begriff Sites ist dabei jedoch nicht genau definiert.

Site-Local-Adressen haben folgendes Format:

10 Bit	54 Bit	64 Bit
1111111011	Subnet-ID	Interface-ID

FEC0::/10 (FEC0... bis FFFF...), auch standortlokale Adressen (site local addresses), waren die Nachfolger der privaten IPv4-Adressen (z. B. 192.168.x.x). Sie durften nur innerhalb der gleichen Netzwerke geroutet werden. Die Wahl des verwendeten IPv6-Adressraums innerhalb von FEC0::/10 konnte in einem Netzwerk frei getroffen werden. Bei der Zusammenlegung von ehemalig getrennten Netzwerken oder wenn eine VPN-Verbindung zwischen getrennten durch Site-Local-Adressen nummerierten Netzwerken hergestellt wurde, konnte es daher zu Überschneidungen der IPv6-Adressräume an den unterschiedlichen Standorten kommen.



Aus diesem und weiteren Gründen wurden Site-Local-Adressen mit RFC 3879 bereits im September 2004 verworfen.

Nach RFC 4291 wird der Präfix FEC0::/10 in neuen Installationen als Global Unicast behandelt. Altinstallationen dürfen den Präfix allerdings weiterverwenden (...may continue to use this prefix).

### Unique-Local- -Adressen

FC00::/8 und FD00::/8: Für private Adressen gibt es die Unique-Local-Adressen (ULA), beschrieben in RFC 4193. Zur Zeit wird allerdings nur der Präfix FD00::/8 für lokal generierte Unique-Local-Adressen verwendet. Auf dieses Präfix folgen 40 Bit, die als eindeutige Site-ID fungieren. Diese Site-ID ist bei den Unique-Local-Adressen mit dem Präfix FC00/8 oder FD00/8 definiert und sollte nach bestimmten Algorithmen generiert werden, um eine Eindeutigkeit zu gewährleisten. (RFC 4193 gibt jedoch keine konkrete Implementierung der Zuweisung von global eindeutigen Site-IDs an.) Nach der Site-ID folgt eine 16-Bit-Subnet-ID, welche ein Netz innerhalb der Site angibt.

Eine Unique-Local-Unicast-Adresse könnte folgendermaßen aussehen:

fd49:e7f8:d910:a000:1234:5678:abcd:ab12

Sie hat damit folgende Bestandteile:

<b>fd</b>	das Präfix
<b>49:e7f8:d910</b>	ein zufällig erzeugter 40-Bit-Wert
<b>a000</b>	eine willkürlich gewählte Subnet-ID

7 Bit	1	40 Bit	16 Bit	64 Bit
Prefix	L	Global-ID	Subnet-ID	Interface-ID

Unique-Local-Adressen erlauben es innerhalb eines weltweit tätigen Unternehmens, einen einheitlichen, providerunabhängigen Adressraum zu nutzen. Die Adressen müssen allerdings an den Übergängen ins Internet in Global Unicast umgesetzt werden.

ULA-Präfixe aus dem Bereich fd00::/8 werden nach einem bestimmten Algorithmus zufällig erzeugt. Damit wird mit hoher Wahrscheinlichkeit eine Eindeutigkeit hergestellt.

### FC00::/7 Unique Local Unicast



Laut RFC 4193 handelt es sich um eindeutige lokale Unicast-Adressen, die jedoch nur in einem lokalen, abgegrenzten Bereich eingesetzt werden dürfen. Router und Firewalls sollen laut RFC Pakete mit solchen Adressen nicht ins globale Internet weiterreichen.

Der Bereich FD00::/8 ist für nicht öffentliche Netze vorgesehen, vergleichbar mit den Private-IP-Bereichen bei IPv4. Der Bereich FC00::/8 ist zunächst nicht definiert (may be defined in the future).

### Global-Unicast-Adressen

Die Global-Unicast-Adressen sind global gültige sowie eindeutige routingfähige Unicast-Adressen, die im Internet weitergeleitet werden. Die Global-Unicast-Adressen verwenden dabei ein allgemein gültiges Format, das wie folgt lautet:

N Bit	M Bit	64 Bit
Global Routing Präfix	Subnet-ID	Interface-ID

<b>Global Routing Präfix</b>	ein auf eine Site zugeordneter Wert
<b>Subnet-ID</b>	eine Kennung (Identifier) auf einen Link innerhalb der Site
<b>Interface-ID</b>	identifiziert eine Netzwerkschnittstelle innerhalb der Site

Global-Unicast-Adressen haben mit Ausnahme derjenigen, die mit binär 000 beginnen, einen 64-Bit-Interface-ID-Bereich. Das bedeutet:  $n + m = 64$ .

Global-Unicast-Adressen, die mit binär 000 beginnen, sind IPv6-Adressen mit eingebetteten IPv4-Adressen.

## Diverse IPv6-Global-Unicast-Adressen

<b>2000::/3</b>	<b>Global Unicast</b> Global gültige und eindeutige Unicast-Adressen, die im Internet weitergeleitet werden. Bislang erhältliche eindeutige IPv6-Präfixe entstammen dem Bereich von 2000::/16 bis 3FFF::/16.
<b>2001:DB8::/32</b>	Für Dokumentationszwecke reserviert
<b>2001:0000::/32</b>	Teredo-Tunneling gemäß RFC 4380
<b>2001:678::/29</b>	Providerunabhängige Netze (PI), Mindestzuteilung /48
<b>2001:10::/28</b>	Keine IPv6-Adressen, sondern „Overlay Routable Cryptographic Hash IDentifiers“ (ORCHID) gemäß RFC 4843. Diese Pseudo-Adressen sind nicht Routing-fähig und sollten niemals in öffentlichen Netzwerken erscheinen.
<b>2002::/16</b>	<b>6to4-Tunneling</b> Deuten auf Adressen des 6to4-Tunnelmechanismus gemäß RFC 3056 hin
<b>3ffe::/16</b>	Wurden für das Testnetzwerk 6Bone benutzt; dieser Adressbereich wurde gemäß RFC 3701 wieder an die IANA zurückgegeben.
<b>64:ff9b::/96</b>	Kann für den Übersetzungsmechanismus NAT64 gemäß RFC 6146 verwendet werden

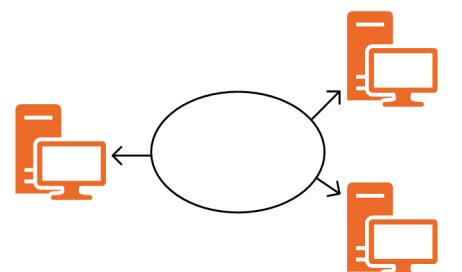
## Durch die IETF reservierte Adressen, die bisher nicht vergeben wurden

<b>4000::/3</b>	6000::/3	8000::/3	A000::/3	<b>C000::/3</b>	E000::/4	F000::/5	F800::/6
-----------------	----------	----------	----------	-----------------	----------	----------	----------

## Multicast

Als **Multicast** bezeichnet man einen Datenverkehr, bei dem eine Sendung von einem Teilnehmer an *mehrere* Empfänger geliefert wird. Multicast ersetzt das aus IPv4 bekannte Broadcast-Verfahren.

FF00::/8 (FF...) stehen für Multicast-Adressen. Eine IPv6-Multicast-Adresse ist eine Kennung für eine Gruppe von Netzwerkschnittstellen, z. B. alle Router. Eine Netzwerkschnittstelle könnte somit zu einer Reihe von Multicast-Gruppen gehören, z. B. ein Router, der zugleich Nameserver und DHCP-Server ist, würde zu drei Multicast-Gruppen gehören.



IPv6-Kommunikation über eine Multicast-Gruppenadresse

Multicast-Adressen haben folgendes Format:

8 Bit	4 Bit	4 Bit	112 Bit
1111111111	flgs	Scope	Gruppen-ID

Das Multicast-Präfix, binär 11111111, am Anfang der IPv6-Adresse identifiziert diese als eine Multicast-Adresse.

Das Feld **flgs** kennzeichnet Flags, die für die Multicast-Adresse gesetzt wurden. Die Größe dieses Feldes beträgt 4 Bit. Gemäß RFC 2373 ist das einzige definierte Flag das T-Flag (Transient Flag). Das T-Flag verwendet das Bit mit niedriger Priorität des Feldes **Flags**. Bei einem Wert von **0** gibt das T-Flag an, dass die Multicast-Adresse eine dauerhaft zugeordnete (bekannte) Multicast-Adresse ist, die von IANA (Internet Assigned Numbers Authority) zugewiesen wurde. Bei einem Wert von **1** gibt das T-Flag an, dass es sich um eine temporäre (nicht dauerhaft zugeordnete) Multicast-Adresse handelt.

In den späteren RFCs 3306 (Unicast-Prefix-based IPv6 Multicast Addresses) und 3956 (Embedding the Rendezvous Point (RP) Address in an IPv6 Multicast Address) finden die P- und R-Bits Verwendung (siehe Tabelle).

Flgs ist ein Satz von 4 Flags:

0	R	P	T
---	---	---	---

Die High-Order-Flag ist reserviert und muss auf 0 initialisiert sein.

T = 0	Kennzeichnet eine permanent zugewiesene Multicast-Adresse, die von der IANA zugewiesen wurde
T = 1	Zeigt eine nicht dauerhaft zugeordnete Multicast-Adresse (well-known)
P-Bit gesetzt	(erzwingt das T-Bit) Unicast-Prefix-based Multicast-Adressen gemäß RFC 3306
R-Bit gesetzt	(erzwingt P- und T-Bit) Multicast-Adressen, welche die Adresse des Rendezvous Point gemäß RFC 3956 enthalten

Das Feld **Scope** (Bereich) bezeichnet den für den Multicastverkehr vorgesehenen Bereich des IPv6-Netzwerks. Die Größe dieses Feldes beträgt 4 Bit. Abgesehen von den von Multicast-Routing-Protokollen bereitgestellten Informationen verwenden Router den Multicast-Bereich, um zu bestimmen, ob Multicast-Daten weitergeleitet werden können.

Die folgenden Gültigkeitsbereiche sind definiert:

interface-lokal	Diese Pakete verlassen die Netzwerkschnittstelle nie. (Loopback)
link-lokal	Werden von Routern grundsätzlich nie weitergeleitet und können deshalb das Teilnetz nicht verlassen
admin-lokal	Der kleinste Bereich, dessen Abgrenzung in den Routern speziell administriert werden muss
site-lokal	Dürfen zwar geroutet werden, jedoch nicht von Border-Routern
organisations-lokal	Die Pakete dürfen auch von Border-Routern weitergeleitet werden, bleiben jedoch „im Unternehmen“ (hierzu müssen seitens des Routing-Protokolls entsprechende Vorkehrungen getroffen werden).
globaler Multicast	Darf frei geroutet werden
0, 3, F	Reservierte Bereiche

Die restlichen Bereiche sind nicht zugewiesen und dürfen von Administratoren benutzt werden, um weitere Multicast-Regionen zu definieren.

Beispiele für festgelegte Multicast-Adressen:

- ✓ FF01::1, FF02::1: All Nodes Adressen. Entspricht dem Broadcast.
- ✓ FF01::2, FF02::2, FF05::2: All Routers Adressen, adressiert alle Router in einem Bereich.

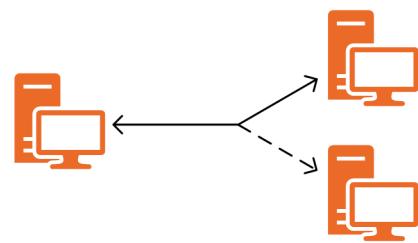
Eine Liste der dauerhaft zugewiesenen Multicast-Adressen findet sich bei der IANA.

Der Link zur IANA-Multicast-Adressen-Liste lautet:

[www.iana.org/assignments/ipv6-multicast-addresses](http://www.iana.org/assignments/ipv6-multicast-addresses)

## Anycast

Ein Anycast ist eine Sendung von einem Teilnehmer an genau einen anderen Teilnehmer. Im Unterschied zum Unicast kann sich die aktuelle Zuständigkeit unter den möglichen Empfängern ändern, so kann z. B. zwecks Ausfallsicherheit bei Nicht-Erreichbarkeit eines Rechners ein anderer Rechner dessen Aufgaben unter Verwendung derselben IP-Adresse übernehmen.



*IPv6-Kommunikation über eine Anycast-Adresse*

Der Sender sendet an eine Anycast-Adresse (**RFC 2526**). Diese setzt sich zusammen aus dem Präfix des verwendeten link-lokalen, unique-lokalen oder globalen Unicast-Adressbereiches, gefolgt von 121 abzüglich Präfixlänge an gesetzten Einser-Bits und in den letzten 7 Bit eine frei wählbare Anycast-ID, wobei die Anycast-ID von hexadezimal 7E für Mobile IP reserviert ist (siehe Mobile IPv6).



Ausnahme: Falls die Präfixlänge 64 ist, muss das 80. Bit der Anycast-Adresse auf Null gesetzt sein, um nicht fälschlich als EUI-64-Interface-ID interpretiert zu werden.

1	n   n+1	80	121   122	128
	n Bits Präfix	121 - n „1“-er Bits Ausnahme Bit 80 = „0“, falls n = 64	7 Bit Anycast-ID	
<b>128 Bit</b>				

*Anycast*

Die zu erreichenden Ziele, in der Regel Server, können sich selbst mit ihrer Anycast-Adresse über Routing-Protokolle im Netz bekannt machen (**Host-Route**). Alternativ, wenn alle Empfänger im gleichen Netzsegment beheimatet sind, kann der davorliegende Router das Subnetz-Präfix ankündigen, dann findet die Auflösung im **NDP** statt.

Anycasts werden genutzt, um Fault-Tolerance und/oder Load-Balancing zu realisieren. Die so zur Verfügung gestellten Dienste müssen dafür geeignet sein; wenn das nicht der Fall ist, ist eine Synchronisation zwischen allen beteiligten Servern erforderlich.

## Vergleich und Zusammenfassung

- ✓ Aus dem IPv4 sind Unicast, Multicast und Broadcast bekannt.
- ✓ Im IPv6 gibt es keine Broadcasts mehr, stattdessen werden gezielt Multicasts eingesetzt.
- ✓ Anycasts dienen der Ausfallsicherheit und Lastverteilung.
- ✓ Die Funktionalität von ARP ist durch NDP im ICMPv6 ersetzt worden.
- ✓ Multicast-Gruppen werden ebenfalls über ICMPv6 gemanagt, IPv6 benötigt kein IGMP.

## 4.7 Besondere Adressbereiche

### Unspecified-Adresse

0:0:0:0:0:0:0:0 oder ::/128: Nicht spezifizierte IPv6-Adresse, die unter IPv4 dem Wert 0.0.0.0 entspricht. Diese Adresse darf nie für eine Netzwerkschnittstelle angegeben werden. Sie zeigt das Fehlen einer Adresse an. Ein Beispiel für die Nutzung der 0:0:0:0:0:0:0:0 oder ::/128 ist im Source-Adressbereich die IPv6-Initialisierung des Hosts, um eine IPv6-Adresse zu erhalten, bevor dieser seine IPv6-Adresse kennengelernt hat.

Die Unspecified-Adresse darf nicht in IPv6-Paketen oder IPv6-Routing-Headern verwendet werden. Ein IPv6-Paket mit einer Quell-Adresse aus dem IPv6 Unspecified IPv6-Adressbereich wird nicht von einem IPv6-Router weitergeleitet.

### Loopback-Adresse

0:0:0:0:0:0:1 oder ::1/128: Loopback-Adresse. Die Loopback-Adresse ist eine Unicast-Adresse und identifiziert die eigene Netzwerkschnittstelle. Jedes an diese Adresse geschickte IPv6-Paket soll den Host nicht verlassen, sondern intern zurückgeschickt werden. Die Loopback-Adresse muss dabei nicht an eine physikalische Netzwerkschnittstelle gebunden sein. Die Loopback-Adresse wird wie eine IPv6-Adresse aus dem Link-Local-Adressbereich behandelt. Sie können daher diese IPv6-Adressen wie Link-Local-Unicast-Adressen einer virtuellen Netzwerkschnittstelle sehen, die zu einem imaginären Link führt, der nirgendwo hinführt.

Sie dürfen die Loopback-Adresse 0:0:0:0:0:0:1 oder ::1/128 nicht als Quell-IPv6-Adresse der Netzwerkschnittstelle verwenden, um IPv6-Pakete über diese Ihrem internen Netzwerk in ein externes Netzwerk zu routen. Die Loopback-Adresse ist ausschließlich für den eigenen Host zu verwenden.

```
C:\>ping localhost

Ping wird ausgeführt für localhost [::1] mit 32 Bytes Daten:
Antwort von ::1: Zeit<1ms
Antwort von ::1: Zeit<1ms
Antwort von ::1: Zeit<1ms
Antwort von ::1: Zeit<1ms

Ping-Statistik für ::1:
  Pakete: Gesendet = 4, Empfangen = 4, Verloren = 0
            (0% Verlust),
Ca. Zeitangaben in Millisek.:
  Minimum = 0ms, Maximum = 0ms, Mittelwert = 0ms
```

*Ansprechen der Netzwerkkarte: ping localhost*

Ein IPv6-Paket, was an eine Loopback-Adresse gesendet wird, wird umgehend verworfen und daher nicht weitergeleitet. Hinweis: Im Gegensatz zu IPv4 mit dem Adressbereich 127.0.0.1/8 gibt es bei IPv6 nur eine einzige IP für das Loopback-Interface.

### Kompatibilitätsadressen

Um die Migration zwischen IPv4 und IPv6 zu ermöglichen, wurden für die Koexistenz beider Hosttypen die folgenden IP-Adressen definiert:

#### IPv4-kompatible Adresse

Die IPv4-kompatible Adresse 0:0:0:0:0:w.x.y.z oder ::w.x.y.z (wobei w.x.y.z die Dezimalschreibweise mit Punkten einer öffentlichen IPv4-Adresse ist) wird von Dual-Stack Node verwendet, die mit IPv6 über eine IPv4-Infrastruktur kommunizieren. Dual-Stack Nodes sind Knoten sowohl mit IPv4- als auch mit IPv6-Protokollen. Wird die IPv4-kompatible Adresse als IPv6-Ziel verwendet, wird der IPv6-Datenverkehr automatisch mithilfe eines IPv4-Headers eingekapselt und unter Nutzung der IPv4-Infrastruktur an das Ziel gesendet.

### IPv4-zugeordnete Adresse

Die IPv4-zugeordnete Adresse 0:0:0:0:FFFF:w.x.y.z oder ::FFFF:w.x.y.z wird zur Darstellung eines reinen IPv4-Knotens für einen IPv6-Knoten verwendet. Die Adresse wird ausschließlich zur internen Darstellung verwendet. Die IPv4-zugeordnete Adresse wird nie als Quell- oder Zieladresse für ein IPv6-Paket verwendet. Das IPv6-Protokoll unterstützt nicht die Verwendung von IPv4-zugeordneten Adressen.

### IPv6-zu-IPv4-Adresse

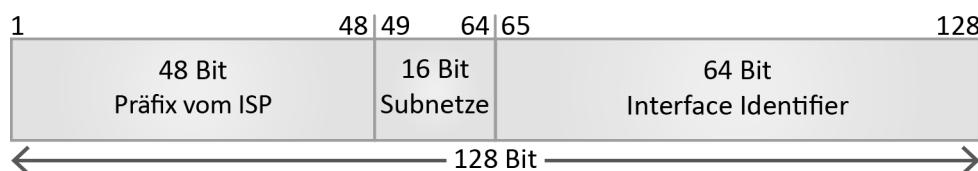
Die IPv6-zu-IPv4-Adresse wird für die Kommunikation zwischen zwei Knoten verwendet, die IPv4 und IPv6 über das Internet ausführen. Die IPv6-zu-IPv4-Adresse wird durch Kombination des Präfixes 2002::/16 mit den 32 Bit der öffentlichen IPv4-Adresse des Knotens gebildet, wobei ein 48-Bit-Präfix entsteht. Für die IPv4-Adresse 131.107.0.1 ist das IPv6-zu-IPv4-Adresspräfix 2002:836B:1::/48. Weitere Informationen zu IPv6-zu-IPv4 finden Sie unter IPv6-Verkehr zwischen Knoten in verschiedenen Sites innerhalb des Internets (6to4).

### Weitere

Eine Übersicht über diese und weitere „Special Purpose Adressen“ findet sich in RFC 6890, die diverse Vorgänger RFC ersetzt, soweit bei der IANA unter <https://www.iana.org/assignments/iana-ipv6-special-registry/iana-ipv6-special-registry.xhtml>

## 4.8 Subnetting

Unter Subnetting wird die Aufteilung eines einzelnen großen Adressbereiches in mehrere kleinere Adressbereiche verstanden.

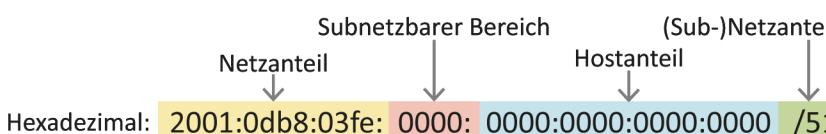


Subnetze werden immer aus dem vierten Block der IPv6-Adresse gebildet. Der ISP vergibt mindestens 48 der 64 zum Präfix gehörenden Bits. Der nicht vergebene Bereich des Präfixes steht für das Subnetting zur Verfügung.

Vergibt der Provider beispielsweise einen /56-Netzbereich, können die verbleibenden (64 - 56) 8 Bit des Präfixes für das Subnetting genutzt werden. Vergibt er einen /48-Netzbereich, können 16 Bit genutzt und damit  $2^{16}$  (65536) Subnetze gebildet werden. Unter Infrastrukturgesichtspunkten werden die Subnetze zuerst auf eventuelle Unternehmensstandorte verteilt. Dort wird dann nach Gebäuden, evtl. Etagen usw. differenziert. Eine Aufteilung des 64 Bit Interface Identifiers ist rechnerisch möglich, widerspricht aber dem Protokolldesign und ist nicht sinnvoll.

### Beispiel 1

Nachfolgend wird das Subnetting bei IPv6 erläutert und exemplarisch an einem einfachen Beispiel in vier Arbeitsschritten durchgeführt.

1	Zunächst wird der Präfix vom ISP zugewiesen. In diesem Fall hat er eine Länge von 48 Bit. Ausgangs-Präfix (Hexadezimal): 2001:db8:3fe::/48
2	Im zweiten Schritt ist zu klären, wie viele Subnetze benötigt werden. Hier: Es sollen sechs Subnetze gebildet werden.
3	Die weitere Berechnung wird im dualen Zahlensystem durchgeführt. Zu der Anzahl der geforderten Subnetze wird die nächsthöhere 2er-Potenz ermittelt. Die nächsthöhere 2er-Potenz zu 6 (Anzahl der benötigten Subnetze) ist 8, da $2^3 = 8$ ; der Exponent ist 3. Der Exponent beschreibt die Anzahl der Bits, um welche die Präfixlänge mindestens verlängert werden muss, um die entsprechend geforderten Subnetze erzeugen zu können. Die Präfixlänge muss um drei Bit verlängert werden, besteht nun aus (48 + 3) 51 Bit. <b>Fazit:</b> Obwohl nur sechs Netze benötigt werden, müssen rechnerisch acht Netze gebildet werden, von denen jedoch nur sechs Verwendung finden.
4	Jetzt können die Präfixe der Subnetze erzeugt werden. Dazu wird das ursprünglich in Arbeitsschritt 1 vorgestellte Präfix (Netzanteil) für alle Subnetze übernommen:  Zur Verdeutlichung wird der Bereich, der zwecks Subnetzbildung zum ursprünglichen Präfix hinzugefügt (hier: die Bits 49 bis 51) wird, zunächst wieder in binärer Schreibweise ausgeführt und dann in die Hexadezimalwerte überführt. Um die sechs Subnetze zu erzeugen, werden drei Bit benötigt. Da der für die Subnetzbildung zur Verfügung stehende Bereich 16 Bit umfasst, von denen die drei „Subnetz-Bits“ abgezogen werden müssen, die dem Netzanteil zugerechnet werden, bleiben noch 77 Bit (13 Bit restlicher Präfixanteil + 64 Bit Hostanteil) übrig, um, wie oben bereits ausgeführt, die Hosts zu adressieren. Die sog. <b>Subnetzadresse</b> ist dabei immer die erste IP-Adresse im Subnetz. Subnetz 1 $\Rightarrow \dots:0000\ 0000\ 0000\ 0000:\dots \Rightarrow \dots:0000:\dots$ (hexadezimal) Subnetz 2 $\Rightarrow \dots:0010\ 0000\ 0000\ 0000:\dots \Rightarrow \dots:2000:\dots$ (hexadezimal) Subnetz 3 $\Rightarrow \dots:0100\ 0000\ 0000\ 0000:\dots \Rightarrow \dots:4000:\dots$ (hexadezimal) Subnetz 4 $\Rightarrow \dots:0110\ 0000\ 0000\ 0000:\dots \Rightarrow \dots:6000:\dots$ (hexadezimal) Subnetz 5 $\Rightarrow \dots:1000\ 0000\ 0000\ 0000:\dots \Rightarrow \dots:8000:\dots$ (hexadezimal) Subnetz 6 $\Rightarrow \dots:1010\ 0000\ 0000\ 0000:\dots \Rightarrow \dots:A000:\dots$ (hexadezimal) Subnetz 7 $\Rightarrow \dots:1100\ 0000\ 0000\ 0000:\dots \Rightarrow \dots:C000:\dots$ (hexadezimal) Subnetz 8 $\Rightarrow \dots:1110\ 0000\ 0000\ 0000:\dots \Rightarrow \dots:E000:\dots$ (hexadezimal) <i>Subnet-Bits 48 - 64 binär (hervorgehoben : 49 - 51) und hexadezimal</i> Jeder Subnetzbereich umfasst $2^{77}$ Adressen: Subnetz 1 $\Rightarrow \dots:0000\ 0000\dots$ bis ...:0011 1111... $\Rightarrow \dots:00\dots$ bis ...:1F... (hexadezimal) Subnetz 2 $\Rightarrow \dots:0010\ 0000\dots$ bis ...:0011 1111... $\Rightarrow \dots:20\dots$ bis ...:3F... (hexadezimal) Subnetz 3 $\Rightarrow \dots:0100\ 0000\dots$ bis ...:0101 1111... $\Rightarrow \dots:40\dots$ bis ...:5F... (hexadezimal) Subnetz 4 $\Rightarrow \dots:0110\ 0000\dots$ bis ...:0111 1111... $\Rightarrow \dots:60\dots$ bis ...:7F... (hexadezimal) Subnetz 5 $\Rightarrow \dots:1000\ 0000\dots$ bis ...:1001 1111... $\Rightarrow \dots:80\dots$ bis ...:9F... (hexadezimal) Subnetz 6 $\Rightarrow \dots:1010\ 0000\dots$ bis ...:1011 1111... $\Rightarrow \dots:A0\dots$ bis ...:BF... (hexadezimal) Subnetz 7 $\Rightarrow \dots:1100\ 0000\dots$ bis ...:1101 1111... $\Rightarrow \dots:C0\dots$ bis ...:DF... (hexadezimal) Subnetz 8 $\Rightarrow \dots:1110\ 0000\dots$ bis ...:1111 1111... $\Rightarrow \dots:E0\dots$ bis ...:FF... (hexadezimal) <i>Die Hostanteile der einzelnen Subnetze (hier: Bit 49 - 56)</i>

Insgesamt könnten, ausgehend von dem (vom ISP zugewiesenen) Adressbereich 2001:db8:3fe::/48 mit dem Präfix 2001:0db8:03fe:0000 und der Präfixlänge /48 bei der Erweiterung auf die Präfixlänge /64 folgende Subnetze erzeugt werden.

1 durchgängiger Adressbereich mit Präfixlänge 48	65536 aufeinander folgende Subnetze mit Präfixlänge 64
2001:db8:3fe::/48 von 2001:0db8:03fe:0000:0000:0000:0000 bis 2001:0db8:03fe:ffff:ffff:ffff:ffff	2001:db8:3fe::/64 von 2001:0db8:03fe: <b>0000</b> :0000:0000:0000 bis 2001:0db8:03fe: <b>0000</b> :ffff:ffff:ffff:ffff
	2001:db8:3fe: <b>1</b> ::/64 von 2001:0db8:03fe: <b>0001</b> :0000:0000:0000 bis 2001:0db8:03fe: <b>0001</b> :ffff:ffff:ffff:ffff
	2001:db8:3fe: <b>2</b> ::/64 von 2001:0db8:03fe: <b>0002</b> :0000:0000:0000 bis 2001:0db8:03fe: <b>0002</b> :ffff:ffff:ffff:ffff
	...
	2001:db8:3fe: <b>ffff</b> ::/64 von 2001:0db8:03fe: <b>ffff</b> :0000:0000:0000 bis 2001:0db8:03fe: <b>ffff</b> :ffff:ffff:ffff

Für eine übersichtliche Darstellung sind die Anfangs- und Endadressen absichtlich ohne mögliche Verkürzungen und mit allen führenden Nullen dargestellt. Auch hier sind die ersten drei Gruppen (48 Bit) durch das vom ISP zugeteilte Präfix fest. In der vierten Gruppe (16 Bit) werden die Subnetze fortlaufend von 0001 bis ffff durchnummert, und die hinteren vier Gruppen (64 Bit) stehen für die Interface-IDs zur Verfügung.

In der Realität kann es zu extrem komplexen Subnetzbildungen kommen. Oft existieren mehrstufige Subnetzstrukturen, und das Präfix endet nicht genau zwischen zwei hexadezimalen Werten in der Adressdarstellung. Mithilfe der Subnetze ergibt sich so die Möglichkeit, die räumliche und funktionale Struktur z. B. eines internationalen Konzerns abzubilden.

## Beispiel 2

Als weiteres Beispiel nehmen wir einen fiktiven weltweit agierenden Konzern „International GmbH“, der im Zuge einer Expansion zu seiner Zentrale zwei neue Standorte „Nord“ und „Süd“ aufbaut und in seinem Netz IPv6 einführen möchte. Zusätzlich sind in jedem Standort Subnetze für die Funktionsbereiche (Geschäftsleitung, Einkauf usw.) erforderlich. Für die Adressierung steht ein unique-lokaler Adressbereich (siehe Unique Local Address) mit dem Präfix fd49:e7f8:d910::/48 zur Verfügung.

Für die Bildung von Subnetzen ist zuerst ein Subnetting für die weltweiten Standorte durchzuführen. Eine Präsenz an maximal 200 Niederlassungen in allen Ländern bedeutet ( $200 \leq 256 = 2^8$ ) für jeden Standort eine Präfixlänge von ( $48 + 8 =$ ) 56 Bit. Damit haben die maximal 256 Standorte die Präfixe:

```
fd49:e7f8:d910:0000::/56
fd49:e7f8:d910:0100::/56
fd49:e7f8:d910:0200::/56
fd49:e7f8:d910:0300::/56
fd49:e7f8:d910:0400::/56
fd49:e7f8:d910:0500::/56
...
fd49:e7f8:d910:fd00::/56
fd49:e7f8:d910:fe00::/56
fd49:e7f8:d910:ff00::/56
```

*Die mit den Bits 49 bis 56 gebildeten Subnetze der Standorte*

Der Zentrale wird nun das Präfix `fd49:e7f8:d910:a000::/64` zugewiesen. Die Subnetze mit den Teilnehmern sollen auch hier eine Präfixlänge von 64 Bit aufweisen. In den Bits 57 bis 64 kann dies abgebildet werden. Die zur Verfügung stehenden 8 Bit erlauben  $2^8 = 256$  Subnetze in jedem Standort. Für die Zentrale bedeutet das:

```
fd49:e7f8:d910:a000::/64  
fd49:e7f8:d910:a001::/64  
fd49:e7f8:d910:a002::/64  
fd49:e7f8:d910:a0ff::/64
```

Diese können nun den Funktionsbereichen wie z. B. Geschäftsleitung, Einkauf, Vertrieb, Marketing und Produktion zugewiesen werden.

# 5 Konfiguration

## In diesem Kapitel erfahren Sie

- ✓ welche Funktionen die Autokonfiguration bietet
- ✓ welche Zusammenhänge zwischen Autokonfiguration und DHCP bestehen
- ✓ was bei DNS zu beachten ist

## Voraussetzungen

- ✓ Kenntnisse über die IPv6-Adressierung
- ✓ Kenntnisse über Adressbereiche und Notationen

## 5.1 Möglichkeiten der Konfiguration

### Statische und automatische Konfiguration

IPv6 ist für den massenhaften, automatischen und unproblematischen Einsatz einer stetig wachsenden Anzahl von netzfähigen Geräten konzipiert.

Eine **statische Konfiguration** ist bei IPv6 nach wie vor für Router, Server oder andere Netzfunktionalitäten erforderlich. Für die Masse der Hosts wie Desktoprechner, mobile Geräte (Handys, Tablets o. a.) oder auch die Kaffeemaschine ist jedoch eine Adresszuweisung per **automatischer Konfiguration** angemessen.

Die automatische Konfiguration gibt es im Zusammenhang mit IPv6 in zwei Varianten:

- ✓ **stateless autoconfiguration** (statuslose Autokonfiguration)
- ✓ **stateful autoconfiguration** (statusbehaftete Autokonfiguration)

### Ziele der automatischen Konfiguration

Folgende Ziele werden mit einer automatischen Konfiguration verfolgt:

- ✓ keine manuelle Konfiguration der Netzeilnehmer nötig
- ✓ jede Schnittstelle bekommt eine eindeutige Adresse (engl. interface identifier), im einfachsten Fall die MAC-Adresse der LAN-Verbindung, in Kombination mit einer Netzwerkidentifikation,
- ✓ DHCP- und routerfreie Kommunikation in kleinen Netzwerken
- ✓ Plug and Play-Kommunikation durch Verwendung link-lokal Adressen
- ✓ DHCP-freie Konfiguration großer, gerouteter Netzwerke
- ✓ automatische Vergabe von globalen Adressen durch Multicast-Ankündigungen des Netzpräfixes durch Router
- ✓ schnelle Umadressierung größerer Netzwerke (z. B. bei einem Providerwechsel) durch eine zeitlich begrenzte Zuweisung mehrerer Adresspräfixe für eine Schnittstelle

## 5.2 Stateless Address Autoconfiguration (SLAAC)

### Merkmale der Stateless Autoconfiguration

Neu an der mit IPv6 eingeführten **Stateless Address Autoconfiguration** (RFC 4862) ist, dass hierzu nicht unbedingt, wie unter IPv4, ein DHCP-Server benötigt wird.

Diese Methode darf nicht mit dem später zu behandelnden **stateless-DHCP** verwechselt werden! Beide Methoden können sich ergänzen, um den Desktoprechnern neben einer Netzwerkangabe auch noch weitere Informationen über eine vorhandene Infrastruktur wie DNS-Server oder DNS-Namen zu übergeben.



### Stateless Autoconfiguration einer Schnittstelle

Die Stateless Autoconfiguration einer Schnittstelle erfolgt bei IPv6 nach folgendem Muster:

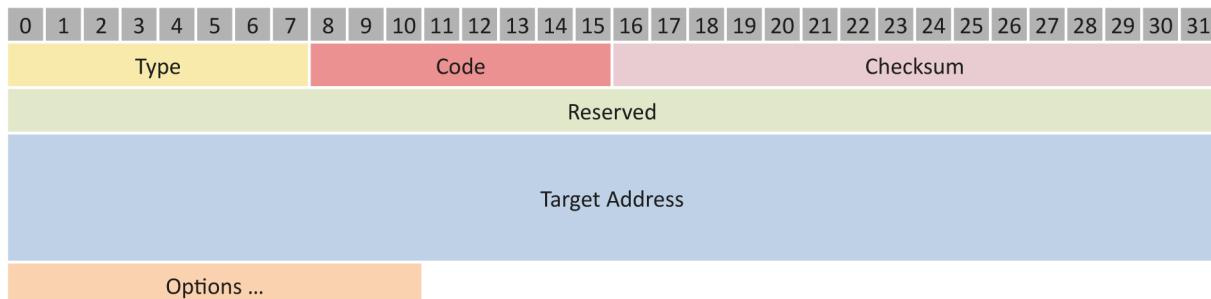
- ✓ Die Autokonfiguration beginnt während des Systemstartes eines Gerätes. Dabei wird automatisch eine link-lokale Adresse generiert (nach RFC 4291 beginnt diese Adresse immer mit dem hexadezimalen Wert **FE80:**). Standardmäßig wird diese Adresse gemäß EUI-64 unter Berücksichtigung der MAC-Adresse und des Präfixes generiert; verschiedene Betriebssysteme bieten auch die Möglichkeit, den Host-Anteil mithilfe eines Zufallsgenerators zu erzeugen (vgl. Abschnitt 4.3).
- ✓ Bevor diese Adresse an die Schnittstelle gebunden werden kann, erfolgt per **NDP (Neighbor Discovery Protocol)** eine **Neighbor Solicitation Message** (RFC 4861). Neighbor Solicitation Message ist eine Nachricht an die soeben erstellte, eigene link-lokale Adresse, um zu überprüfen, ob die Adresse in diesem Netzsegment eindeutig ist. Sollte diese Adresse bereits an ein anderes Gerät vergeben sein, würde dieses sich angesprochen fühlen und eine Antwort senden.
- ✓ Bleibt ein **Neighbor Advertisement** (Antwortnachricht) aus, ist die Adresse eindeutig und kann der Schnittstelle zugewiesen werden.

Bei Verwendung einer **nicht eindeutigen** link-lokalen Adresse, also einer Adresse, die bereits im Netz vergeben ist, stoppt der Autokonfigurationsvorgang, und die weitere Adressvergabe muss manuell erfolgen. Bei Feststellung der Eindeutigkeit der Adresse kann der Host jetzt mit anderen link-lokalen Adressen in seinem lokalen Netzabschnitt kommunizieren.

### ICMPv6-Nachrichten

**Neighbor Solicitation Message** und **Neighbor Advertisement** sind zwei neue **Internet-Control-Message-Protocol (ICMPv6)**-Nachrichten, die im NDP definiert sind.

Die Neighbor Solicitation Message (Nachbarabfrage) hat folgenden Aufbau:



#### ICMPv6-Paket – Wichtige Felder in der Neighbor Solicitation Message

ICMP: Die Signalart in ICMP wird durch den Typ und den Code definiert, z. B. für den „ping“ verwendete Pakete, ICMP-Request und ICMP-Reply werden durch die Typen 8 und 0 angegeben.

### ICMP-Felder

- ✓ Type: 135 (definiert die ICMP-Nachricht als Neighbor Solicitation Message)
- ✓ Code: 0 (für Neighbor Solicitation Message)
- ✓ Reserved: wird vom Sender mit Nullen aufgefüllt und vom Empfänger ignoriert
- ✓ Target Address

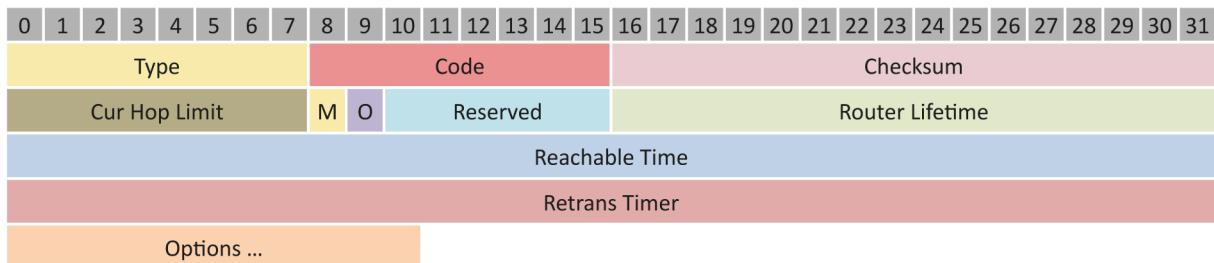
### Router finden

Nach der Konfiguration der link-lokalen Adresse besteht der nächste Schritt darin, einen Router zu finden bzw. Signale eines Routers aufzufangen, um weitere Informationen über die Adressierung im eigenen Netz zu erhalten. Ohne diese zusätzlichen Informationen ist ein Datenaustausch mit anderen Netzen (geroutete lokale Netze, Internet) nicht möglich.

### Router Advertisements (RA)

Deshalb senden Router in regelmäßigen Abständen **Router-Ankündigungen** (engl. **Router Advertisements**), die bekanntgeben, in welchem Netz sich der Host befindet (Netzwerk-Präfix) und welche Art von Autokonfiguration unterstützt wird. Mit diesen Informationen ist der Host in der Lage, seine automatische Netzwerkkonfiguration zu beenden. Erhält der Host das Router Advertisement nicht innerhalb einer durch das Betriebssystem vorgegebenen Zeit, veranlasst der Host mit einem **Router-Solicitation** (Router ersuchen) an die All-Routers-Multicast-Adresse (ff02::2) eine Router-Ankündigung, um unverzüglich an die benötigten Informationen zu gelangen.

#### ICMPv6-Paket – Wichtige Felder im Router Advertisement



### IP-Felder (hier nicht gezeigt)

- ✓ Quell-Adresse: link-lokale Adresse des Absenders (Router)
- ✓ Ziel-Adresse: Router Solicitation oder All-Nodes – Multicast-Adresse (ff02::1)
- ✓ Hop Limit: 255

### ICMP-Felder

- ✓ Type: 134 (definiert die ICMP-Nachricht als Router Advertisement – NDP)
- ✓ Code: 0 (für diesen Fall)
- ✓ Checksum: ICMP-Prüfsumme
- ✓ Cur Hop Limit: Zähler im Header für ausgehende IP-Pakete, bei 0 wird nicht vom Router spezifiziert
- ✓ M: „Managed address configuration“-Flag, wenn gesetzt, erfolgt die IP-Konfiguration per DHCP
- ✓ O: „Other configuration“-Flag, wenn gesetzt, werden weitere Netzinformationen wie z. B. DNS-Server von einem DHCP-Server bereitgestellt. Dazu wird der DHCP-Server im Netzwerk gesucht, dies geschieht nicht, wenn das O-Flag nicht gesetzt wurde.
- ✓ Router Lifetime: Gültigkeit des Default Routers in Sekunden (maximal 9000s möglich), bei 0 ist der ankündigende Router nicht der Default Gateway

Bei Erfolg bekommt der Host nun für seine Schnittstelle das Netzpräfix und ein Standardgateway mitgeteilt. Damit kann der Host im eigenen Subnetz und über Netzgrenzen hinweg arbeiten. Dem Host fehlen noch Informationen z. B. zum verwendeten DNS-Server oder dem verwendeten Domänennamen. DNS-Server und Domainsuchliste können auch über RA publiziert werden, zuletzt beschrieben in RFC 8106 (vormals RFC 5006). Ist in den RA das O-Flag gesetzt, erwartet der Host die Informationen zu DNS und Suchliste von einem stateless DHCPv6-Server.

### Zusammenfassung zu Stateless Autoconfiguration

Bei der Stateless Autoconfiguration wird dem Netzwerkadapter automatisch eine IPv6-Adresse vom Betriebssystem zugewiesen. Die Information über das Netzwerkpräfix kommt von einem Router.

Damit hat der Netzwerkadapter Informationen über sein Netzwerk und eine eindeutige Hostadresse.

Die Routerinformation kann der Host vom Router bekommen, weitere Informationen wie DNS-Server oder DNS-Namen kann er über RA oder von einem DHCP-Server erhalten.

Bleibt die Antwort auf die Router Solicitation aus oder ist das M-Flag im Router Advertisement gesetzt, versucht der Host eine Anfrage an einen DHCPv6-Server an die Multicast-Adresse ff02::1:2.

Dann erfolgt eine Stateful Autoconfiguration.

## 5.3 Stateful Autoconfiguration

Bei der **Stateful Autoconfiguration** wird dem Netzwerkadapter von einem vorhandenen DHCPv6-Server eine Adresse zugewiesen. Diese Adresszuweisung wird vom DHCPv6-Server registriert.

Der Anfragevorgang entspricht der Stateless Autoconfiguration:

- ✓ Autoconfiguration einer link-lokalen Adresse
- ✓ Warten auf ein Router Advertisement oder Senden einer Router Solicitation
- ✓ Wenn die Antwort vom Router ausbleibt oder wenn das M-Flag im Router Advertisement gesetzt ist, erfolgt als Nächstes eine DHCPv6-Anfrage.

## 5.4 Grundlagen zum DHCPv6-Protokoll

DHCP (Dynamic Host Configuration Protocol) ermöglicht die automatische Netzwerk-Konfiguration eines Computers, der in ein bestehendes Datennetz eingebunden werden soll, ohne dass ein manueller Eingriff notwendig wird.

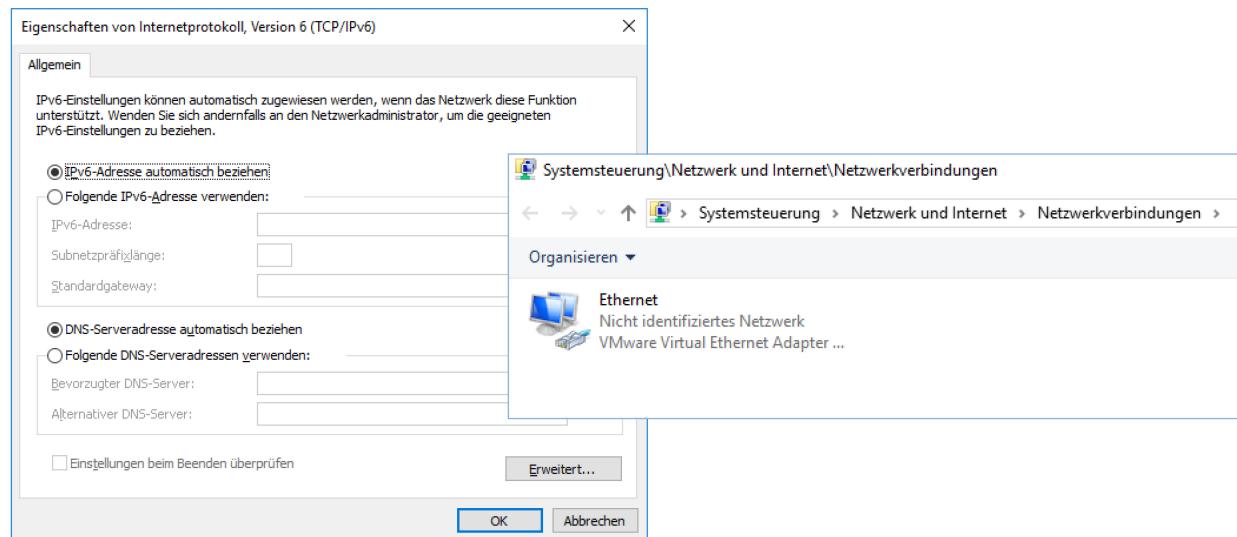
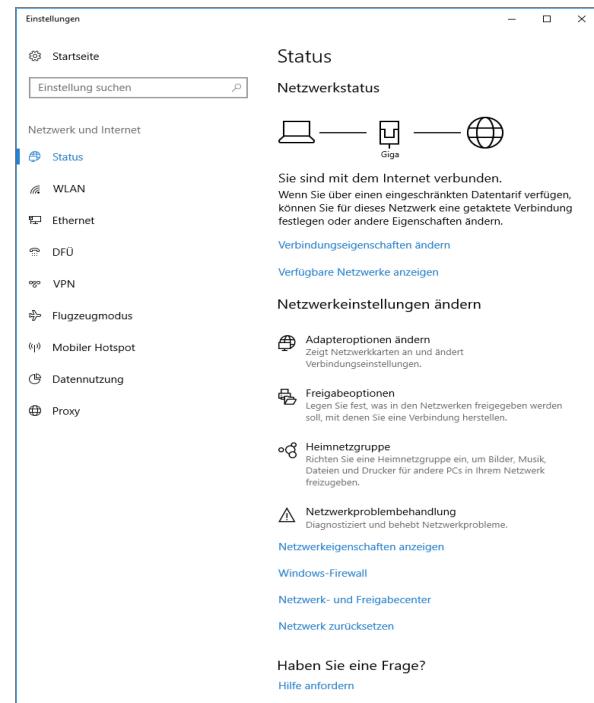
Der Computer erhält von einem DHCPv6-Server beim Start neben der IP-Adresse und der Prefixlänge alle weiteren Informationen, die er zur Netzwerkkommunikation benötigt:

- ✓ seine eigene IP-Adresse
- ✓ die Prefixlänge, normalerweise /64
- ✓ die Adresse des DNS-Servers
- ✓ ggf. weitere Optionen/Informationen
- ✓ das Default Gateway wird bei IPv6 ausschließlich durch RA publiziert, nicht über DHCPv6

Einzig die Angabe, dass der Rechner seine Konfigurationsdaten über DHCP erhalten soll, muss i. d. R. manuell vorgegeben werden.

Die Konfiguration können Sie in Windows ab Vista/ Server 2008 beispielsweise über das Netzwerk- und Freigabe-center der Systemsteuerung links unter **Adapter-einstellungen ändern** und dann in den Eigenschaften der Netzwerkadapter, die DHCP-Einstellungen vornehmen.

Ab Windows 8/Server 2016 sind die Netzwerkeinstellungen über das Adminmenue (Rechtsklick auf Windowsicon) schnell erreichbar. Im Fenster **Adaptoreinstellungen** finden Sie die entsprechenden Einstellungen unter dem Menüpunkt **Eigenschaften** der Internetprotokolle IPv4 und IPv6.



#### *Client für DHCP konfigurieren*

Ist für IPv6 eine Konfiguration über DHCP vorgesehen, müssen das M- und das O-Flag (Managed Address, Other Stateful Configuration) beim Router Advertisement gesetzt sein (vgl. Abschnitt 7.1).

DCHP ist eine Erweiterung des **Bootstrap Protocol (BOOTP)**. Die Grundidee dieses Protokolls ist die Realisierung von **Diskless Workstations**, die beim Boot-Vorgang zunächst automatisch eine IP-Adresse vom BOOTP-Server beziehen und im Anschluss ein startbares Betriebssystem aus dem Netz nachladen können.

## 5.5 DHCP-Client/Server-Kommunikation

### Ablauf der Kommunikation

Die Kommunikation wird vom DHCP-Client mit einem DHCP-Request an den DHCP-Server gestartet.

Die Antworten auf den Request werden, unter Nutzung der MAC-Adresse, vom DHCP-Server oder einem DHCP-Relay-Agent direkt an den Client gesendet.

Um DHCPv6-Messages übermitteln zu können, müssen alle DHCPv6-Server oder Relay-Agents Mitglied einer Multicast-Gruppe mit der Adresse FF02:0:0:0:0:1:0 sein. Das DHCP gehört zu den höheren Schichten und benutzt auf der Transportebene das User Datagramm Protocol (UDP). Da das UDP-Protokoll zu den ungesicherten Protokollen gehört, muss das DHCP-Protokoll die Sicherungsfunktionen auf einer höheren Schicht übernehmen.

Ein DHCP-Client sendet alle Datagramme zum Server immer an den UDP-Port 547 (Zielport). Der DHCP-Client empfängt alle DHCP-Messages über den UDP-Port 546.

### Die Nachrichtentypen des DHCPv6-Protokolls

Das DHCPv6-Protokoll verwendet insgesamt sechs Nachrichtentypen:

- ✓ DHCP-Solicit
- ✓ DHCP-Request
- ✓ DHCP-Release
- ✓ DHCP-Advertise
- ✓ DHCP-Reply
- ✓ DHCP-Reconfigure

DHCP-Solicit-Message	Mithilfe von DHCP-Solicit-Messages ermitteln DHCP-Clients oder DHCP-Relays die Adressen der DHCP-Server am Netz. Verfügt ein DHCP-Client über keine DHCP-Agent-Adresse oder ein Client versucht mit einem neuen Server Kontakt aufzunehmen, muss das Paket an die Multicast-Adresse des Servers übermittelt werden.
DHCP-Advertise-Message	Mit dieser Nachricht gibt ein DHCP-Agent seine Adresse den Clients bekannt. Mithilfe dieser Adresse ist es dem Client möglich, den Request an den DHCP-Server zu übermitteln.
DHCP-Request	Der Client verwendet diese Meldung zur Anforderung von Parametern des DHCP-Servers. Sind dem Client noch keine IP-Adressen der DHCP-Server bekannt, dann wird er diese mittels DHCP-Solicit zu erlangen versuchen.
DHCP-Reply	Serverantwort auf jeden empfangenen DHCP-Request
DHCP-Release	Mit dieser Nachricht erklärt der Client, dass er eine IP-Adresse oder Ressource nicht mehr benötigt und damit freigibt.
DHCP-Reconfigure	Mit dieser Nachricht gibt ein Server dem Client bekannt, dass bestimmte Parameter seiner Konfiguration geändert werden müssen.

## 5.6 IPv6 und DNS

### DNS-Grundlagen

Das Domain Name System dient vor allem dazu, Hostnamen in IP-Adressen aufzulösen. Das System geht auf das Arpanet zurück. Dort wurden zu Beginn alle beteiligten Rechner in eine Textdatei eingetragen. Wollte nun ein Host mit z. B. einem Host „kentucky-sf“ Kontakt aufnehmen, hat das System in der Datei nach dem Namen gesucht und die zugehörige logische Adresse (heute IP) ermittelt. Mit wachsender Rechnerzahl wurde das System unhandlich, da es für jeden neuen Host im Netzwerk nachgepflegt werden musste. Deshalb wurde die Auflösung in ein hierarchisches Datenbanksystem, dem heutigen Domain Name System, kurz DNS überführt. Die Textdatei von damals gibt es immer noch in der Form der Datei *hosts* (Windows %systemroot%\system32\drivers\etc\hosts, Linux /etc/hosts). Sie dient heute nur noch dazu, dem Rechner beim Startvorgang Informationen über sich selbst mitzuteilen, solange kein Nameserver verfügbar ist.

```
127.0.0.1      localhost
198.51.100.100   gawein gawein.example.com

# The following lines are desirable for IPv6 capable hosts
::1      ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
```

*/etc/hosts Datei auf einem Debian 6*

In dem Beispiel sehen Sie die IPv6-Multicastadressen, die für die Stateless Autoconfiguration benötigt werden. Es können auch beliebige andere Adress/host- bzw. Adress/FQDN-Kombinationen eingetragen werden. Damit können Sie dem Client für bestimmte Ziele andere IP-Adressen übergeben als die vom DNS gelieferten, da das System in der Standardkonfiguration zuerst die *hosts*-Datei abarbeitet.



Damit können Sie beispielweise einen Server erreichen, der im DNS nicht erreichbar ist, z. B. bei einem Serverwechsel. Bestimmte FQDNs, z. B. von Webadressen, können Sie unerreichbar machen, indem Sie als Ziel die Localhost-Adressen eintragen.

### Ressource Records (RR) bei DNS

DNS arbeitet mit sogenannten **Ressource Records**, RR. Das sind Bezeichnungen für verschiedene Abfragetypen. Der bekannteste RR ist der A-Record für die Auflösung Hostname zu IP-Adresse.

Weiterhin gibt es RRs, die für die Verwaltung des Systems in sich notwendig sind, wie SOA (Start of Authority) oder NS (Nameserver). Das DNS ist in Zonen aufgeteilt, wobei jede Zone immer nur die Informationen der nächsttieferen Hierarchie kennt.

**Beispiel:** Der Host *iwein* in der Domain *example.net* wird im DNS mit dem Full Qualified Domainname (FQDN) *iwein.example.net* bezeichnet. Darin stecken mehrere Hierarchien, die bei der Abfrage von rechts nach links rekursiv durchlaufen werden. Jeder Rechner fragt zuerst seinen übergeordneten Nameserver nach der IP zu einem Rechner-Namen oder einer Internetadresse wie z. B. *www.google.de*, um die Root-Nameserver zu entlasten. Zudem speichert jeder DNS-Server die Ergebnisse seiner Anfragen zwischen. Für diese Zwischen-speicherung gibt es einen TTL-Eintrag (Time to Live) für jede DNS-Zone. Damit wird die Aktualität der Einträge sichergestellt. Der Punkt „.“ steht für DNS-Root, repräsentiert durch 13 weltweit verteilte Root-Nameserver, die Informationen über die Toplevel-Domains wie z. B. *.de*, *.uk*, oder wie in unserem Beispiel *.net* oder *.arpa* haben.

Die IP-Adressen der Rootserver müssen jedem Nameserver bekannt sein. Sie werden beim Windows DNS-Server als Stammhinweise bezeichnet.

```
# nslookup
> set q=ns
> .
Server:      203.0.113.00
Address:     213.133.100.100#53

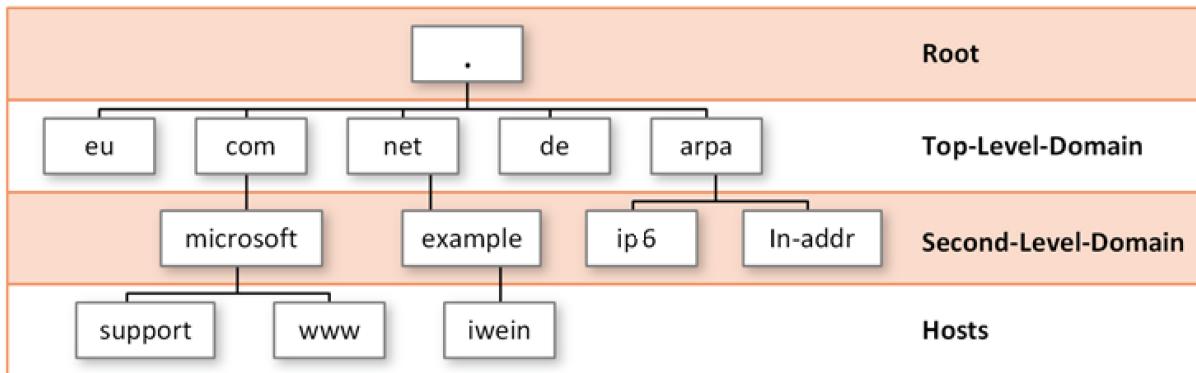
Non-authoritative answer:
.      nameserver = c.root-servers.net.
.      nameserver = m.root-servers.net.
.      nameserver = k.root-servers.net.
.      nameserver = f.root-servers.net.
.      nameserver = b.root-servers.net.
.      nameserver = h.root-servers.net.
.      nameserver = a.root-servers.net.
.      nameserver = e.root-servers.net.
.      nameserver = d.root-servers.net.
.      nameserver = i.root-servers.net.
.      nameserver = l.root-servers.net.
.      nameserver = j.root-servers.net.
.      nameserver = g.root-servers.net.

Authoritative answers can be found from:
a.root-servers.net      internet address = 198.41.0.4
a.root-servers.net      has AAAA address 2001:503:ba3e::2:30
b.root-servers.net      internet address = 192.228.79.201
c.root-servers.net      internet address = 192.33.4.12
d.root-servers.net      internet address = 199.7.91.13
d.root-servers.net      has AAAA address 2001:500:2d::d
e.root-servers.net      internet address = 192.203.230.10
f.root-servers.net      internet address = 192.5.5.241
f.root-servers.net      has AAAA address 2001:500:2f::f
g.root-servers.net      internet address = 192.112.36.4
h.root-servers.net      internet address = 128.63.2.53
h.root-servers.net      has AAAA address 2001:500:1::803f:235
i.root-servers.net      internet address = 192.36.148.17
i.root-servers.net      has AAAA address 2001:7fe::53
```

*Abfrage der Root-Zone“ „ mit nslookup ergibt die Root-Nameserver mit IPv4- und IPv6-Adressen*

### Forward Lookup

Bei der Abfrage nach `iwein.example.net` würde also zunächst der Nameserver des Clients befragt werden, in der Regel ein interner Server oder ein DNS-Server vom ISP (Internet Service Provider). Hat dieser die IP zu dem Namen zwischengespeichert, so antwortet er dem Client direkt. Kennt er die IP-Adresse nicht, so fragt er einen der Root-DNS-Server an. Dieser sendet ihm dann die IP-Adresse des Nameservers für `.net`. Dann wird dieser angefragt, und er meldet den DNS für `.example` usw. Die Antworten "merkt" sich der abfragende Nameserver für eine gewisse Zeit, die TTL (Time to Live), die vom zuständigen Nameserver vorgegeben ist. Erst wenn die TTL abgelaufen ist, wird erneut abgefragt. Für IPv6 funktioniert die Abfrage genauso, allerdings wird anstatt nach A nach AAAA RR gefragt.



Grafische Veranschaulichung der Namensauflösung

### Reverse Lookup

Das Ganze funktioniert auch rückwärts und heißt im Gegensatz zum eben beschriebenen **Forward Lookup**, **Reverse Lookup**. Damit IP-Adressen in die Hierarchie des DNS passen, müssen sie „rückwärts“ geschrieben werden. Aus 172.16.10.15 wird dann 15.10.16.172.in-addr.arpa. Zuständig ist dann die Zone 10.16.172.in-addr.arpa., die alle Hosts aus dem Netz 172.16.10.0/24 kennt. Hier wäre die Reihenfolge der Abfrage .arpa. (TLD für Reverse Lookup), in-addr.arpa. (Second Level Domain für IPv4), 172.in-addr.arpa., 16.172.in-addr.arpa. und schließlich die Zielzone 10.16.172.in-addr.arpa.. Der Punkt am Ende wird normalerweise nicht mitgeschrieben, wurde hier aber mit angehängt, um die Root-Zone zu visualisieren.

In der Praxis würde bei diesem Beispiel die Hierarchie nicht durchlaufen, da es sich um eine IP-Adresse aus dem Private-IP-Bereich handelt. Hier wäre die oberste Hierarchie die Zone selbst.

Bei IPv6 funktioniert es entsprechend: Statt in-addr.arpa ist die SLD (Second Level Domain) ip6.arpa. Die Hierarchie ist entsprechend länger, aus FD00:20::c4d0:7721:48d4:50aa wird a.a.5.0.4.d.8.4.1.2.7.7.0.d.4.c.0.0.0.0.0.0.0.0.2.0.0.d.f.ipv6.arpa.

### Auflösung von IPv6-Adressen und Betrieb mit IPv6

Forward und Reverse Lookup von IPv6-Adressen kann inzwischen jeder aktuelle Nameserver, er muss nur den RR für IPv6, AAAA kennen.

Die Kommunikation der DNS-Abfragen selbst über IPv6 funktioniert ebenfalls mit aktueller Software. Ein mit IPv4 betriebener Nameserver bedient alle RR-Records genauso wie ein mit IPv6 betriebener. Hier kommt wieder das Schichtenmodell zum Zuge. DNS-Anfragen und -Antworten werden vorwiegend über das Transportprotokoll UDP übertragen. Die darunter liegende Vermittlungsschicht hat darauf keinen Einfluss. Der Standardport für DNS ist 53.

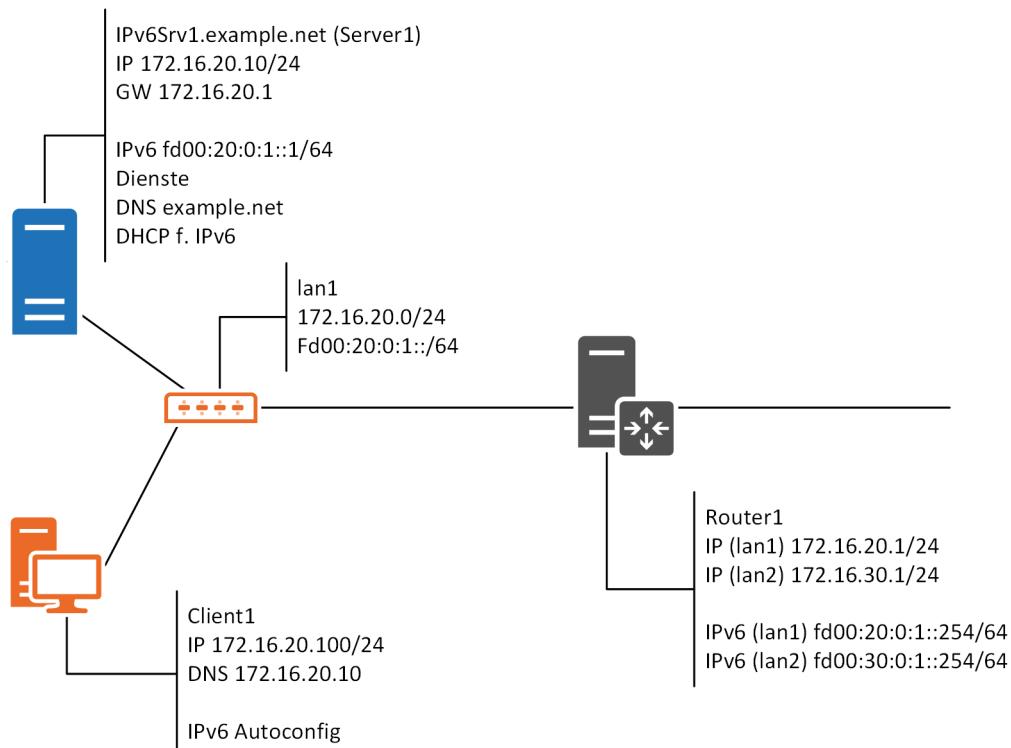
### Wichtige Ressource Records sind

<b>SOA</b>	Start of Authority, Administrative Informationen zur Zone
<b>NS</b>	Hostname eines autoritativen Nameservers
<b>MX</b>	Mail Exchange, Hostname des für diese Domain zuständigen Mailservers
<b>A</b>	IPv4-Adresse
<b>AAAA</b>	IPv6-Adresse
<b>CNAME</b>	Kanonischer Name für einen Host, Alias, verweist auf einen anderen Host bzw. FQDN
<b>SRV</b>	Service, Beantwortet Anfragen nach Diensten in einer Zone
<b>PTR</b>	Domain Name Pointer (für Reverse-Lookup IPv4 und IPv6)

## 5.7 Praxisbeispiel Stateless Autoconfiguration

In diesem Beispiel nehmen Sie die für Stateless Autoconfiguration notwendigen Einstellungen auf dem Router vor.

### Beispielkonfiguration im Testnetz



Aufbau des Testnetzes

### Umgebung

- ✓ **Server1** dient als DNS- und DHCP-Server
- ✓ **Router1** ist für die Autokonfiguration verantwortlich
- ✓ **Client1** ist der Empfänger der IPv6-Adressen

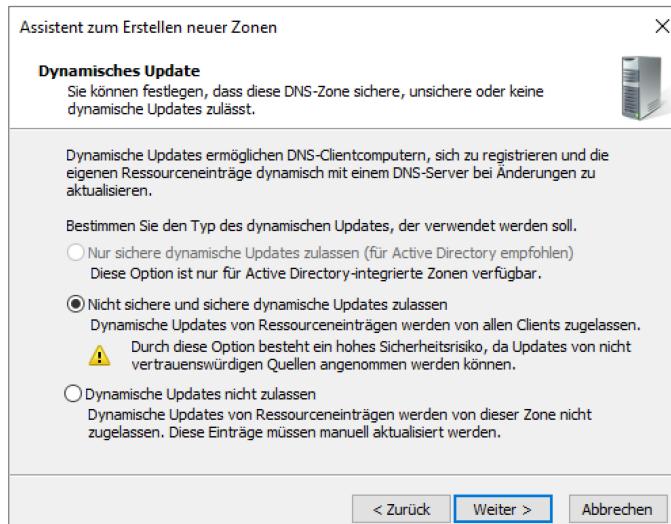
Die Konfiguration erfolgt auf virtuellen Maschinen mit Server 2012 R2 als Server1 und Router1 und Windows 8 für den Client1.

Die Erstkonfiguration ist mit IPv4-Adressen erstellt, die nach den Angaben fest für die Server und Clients vergeben werden, um die allgemeine Konnektivität zu testen. **Lan1** ist ein virtueller Switch.

## Konfiguration Server1

Die Einstellungen für Server1 wurden bereits in der Beispielkonfiguration vorgenommen:

- ✓ Auf dem Server1 ist die Rolle DNS-Server installiert.
- ✓ Die Domäne hat den Namen example.net.
- ✓ Der DNS-Server soll eine automatische Registrierung des Clients zulassen.
- ✓ Der DHCP-Server soll zuerst die Stateless Autoconfiguration unterstützen und dann für die Stateful Autoconfiguration eingerichtet werden.
- Wählen Sie beim Erstellen der DNS-Zonen die Option *Nicht sichere und sichere dynamische Updates zulassen*.
- Zeigen Sie in der DNS-Server-Konfiguration die angelegten Forward- und Reverse-Lookupzonen an.  
Noch sind keine IPv6 Reverse Zonen angelegt.



Name	Typ	Daten
(identisch mit übergeordnete...)	Autoritätsursprung (SOA)	[1] ipv6srv1.example.com...
(identisch mit übergeordnete...)	Namenserver (NS)	ipv6srv1.example.com.
ipv6srv1	Host (A)	172.16.20.1

### Eingerichtete DNS-Zonen

Zur Vereinfachung wurden auf dem Server1 die Firewall für das private und öffentliche Profil ausgeschaltet.

The screenshot shows the 'Windows-Firewall mit erweiterter Sicherheit - Eigenschaften von L...' (Windows Firewall with Advanced Security - Properties of L...) dialog box. The 'Öffentliches Profil' (Public Profile) tab is selected. It says: 'Legen Sie das Verhalten für den Fall fest, wenn ein Computer eine Verbindung mit einem öffentlichen Netzwerk hat.' (Define the behavior for the case when a computer connects to a public network.) Below this, there is a 'Status' section with:
 

- Firewallstatus: Aus
- Eingehende Verbindungen: Blockieren (Standard)
- Ausgehende Verbindungen: Zulassen (Standard)

## Konfiguration Client1

- Konfigurieren Sie Client1 so, dass er seine IPv4-Einstellungen vom DHCP-Server erhält.

Ethernet-Adapter LAN-Verbindung:

```
Verbindungsspezifisches DNS-Suffix:
  Beschreibung . . . . . : Broadcom NetXtreme 57xx-Gigabit
  Physische Adresse . . . . . : 00-1D-09-DE-B8-EC
  DHCP aktiviert . . . . . : Ja
  Autokonfiguration aktiviert . . . . : Ja
  Verbindungslokale IPv6-Adresse . . :
  fe80::0029:757f:769d:3f59%11 (Bevorzugt)
```

### IP- Konfiguration Client1

- ▶ Starten Sie die Netshell und wechseln Sie in den Kontext Interface IPv6:
 

```
C:\>netsh
      netsh> interface
      netsh interface> ipv6
      netsh interface ipv6 Kommando
```
- ▶ Um unter Windows von einer zufällig generierten Host-ID (Standard) zu einer EUI-64-konformen ID zu wechseln, geben Sie folgendes ein:
 

```
netsh interface ipv6> set global randomizeidentifiers=disabled
```

Das Ergebnis ist eine EUI-64-Adresse:

```
Verbindungslokale IPv6-Adresse . . : fe80::21d:9ff:fedc:b8ec%12
IPv4-Adresse . . . . . : 192.168.55.114
Subnetzmaske . . . . . : 255.255.255.0
```

Mit *ipconfig abgefragte IP-Konfiguration*

### Konfiguration Router1

- ▶ Wechseln Sie auf den Router1 und rufen Sie wie oben beschrieben den Netshellkontext interface ipv6 auf.
- ▶ Zeigen Sie die Liste der IPv6-Schnittstellen an:
 

```
show interfaces
```

```
C:\>netsh interface ipv6 show interfaces
Idx      Met          MTU        State           Name
---  -----
  1       50    4294967295  connected   Loopback Pseudo-Interface 1
  12      50         1280  disconnected  isatap.fritz.box
  11      25         1500  connected   LAN-Verbindung
  15      25         1500  connected   LAN-Verbindung 2
```

*show interfaces-Ausgabe Router1*

Der Index (Idx) bezieht sich auf die interne ID der Netzwerkschnittstelle. Sie benötigen Index 11 für Lan1 und Index 15 für Lan2 für die späteren Einstellungen.

- ▶ Zeigen Sie Details zur entsprechenden Schnittstelle an:
 

```
show address 11
```

Sie sehen die automatisch erzeugte Link-Local-Adresse.

```
C:\>netsh interface ipv6 show address 12
Adresse fe80::21d:9ff:fedc:b8ec%12 Parameter
-----
Schnittstellen-LUID      : LAN-Verbindung
Bereichskennung         : 0.11
Gültigkeitsdauer        : infinite
Bevorzugte Gültigkeitsdauer : infinite
DAD-Status              : Bevorzugt
Adresstyp                : Andere
Als Quelle überspringen   : false
```

*Adressinformationen zur Schnittstelle LAN1*

- ▶ Erstellen Sie für Lan1 eine Unique-Local-Adresse:
 

```
netsh int ipv6 add address 11 fd00:20:0:1::254
```
- ▶ Lassen Sie sich die auf dem System vorhandenen IPv6-Adressen anzeigen:
 

```
netsh int ipv6 show addresses 11
```

```
C:\>netsh interface ipv6 show address 11

Adresse fd00:20:0:1::254 Parameter
-----
Schnittstellen-LUID      : LAN-Verbindung
Bereichskennung          : 0.0
Gültigkeitsdauer         : infinite
Bevorzugte Gültigkeitsdauer : infinite
DAD-Status               : Bevorzugt
Adressstyp                : Temporär
Als Quelle überspringen    : false

Adresse fe80::21d:9ff:fede:b8ec%11 Parameter
-----
Schnittstellen-LUID      : LAN-Verbindung
Bereichskennung          : 0.11
Gültigkeitsdauer         : infinite
Bevorzugte Gültigkeitsdauer : infinite
DAD-Status               : Bevorzugt
Adressstyp                : Andere
Als Quelle überspringen    : false
```

#### Zwei IPv6-Adressen auf einem Interface

Im Ergebnis können Sie erkennen, dass die Schnittstelle 11 (Lan1) nun zwei IPv6-Adressen aufweist:

- Um die Schnittstellen-Details einzusehen, geben Sie folgenden Befehl ein:  
netsh int ipv6 show interface 11

```
C:\>netsh interface ipv6 show interface 11

Parameter für die Schnittstelle "LAN-Verbindung"
-----
Schnittstellen-LUID      : ethernet_5
Schnittstellenindex       : 11
Zustand                  : connected
Metrik                   : 10
Link-MTU                 : 1500 Bytes
Erreichbare Zeit          : 43000 ms
Basiswert für erreichbare Zeit : 30000 ms
Intervall für die erneute Übertragung : 1000 ms
DAD-Übertragungen        : 1
Standortpräfixlänge       : 64
Standort-ID                : 1
Weiterleitung              : disabled
Ankündigung             : disabled
Nachbarsuche              : enabled
Nachbar-Nichterreichbarkeitserkennung : enabled
Routersuche                : enabled
Verwaltete Adresskonfiguration : enabled
Andere Zustandsbehaftete Konfiguration : enabled
Schwacher Host sendet     : disabled
Schwacher Host empfängt    : disabled
Automatische Metrik verwenden : enabled
Standardrouten ignorieren   : disabled
Angekündigte Routerlebensdauer : 1800 Sekunden
Standardroute ankündigen     : disabled
Aktuelles Hoplimit           : 255
ARPND-Reaktivierungsmuster erzwingen : disabled
Gerichtete MAC-Reaktivierungsmuster : disabled
```

#### Ausgabe show Interface 11 für LAN-Verbindung

Sie sehen, dass der Parameter Ankündigung ausgeschaltet ist. Das bedeutet, dass die Schnittstelle keine Router Advertisements ins Netz sendet. Der Parameter Standardroute ankündigen ist ebenfalls noch deaktiviert. Dieser dient dazu, einen Default Gateway zu publizieren.

- ▶ Zunächst aktivieren Sie den Parameter Ankündigung:  
set interface 11 advertise=enabled
- ▶ Danach deaktivieren Sie die verwaltete und gemanagte Adresskonfiguration (M-, O-Flags):  
set interface 11 otherstateful=disabled managedaddress=disabled

Weiterleitung	:	disabled
<b>Ankündigung</b>	:	<b>enabled</b>
Nachbarsuche	:	enabled
Nachbar-Nichterreichbarkeitserkennung	:	enabled
Routersuche	:	enabled
<b>Verwaltete Adresskonfiguration</b>	:	<b>enabled</b>
<b>Andere zustandsbehaftete Konfiguration</b>	:	<b>enabled</b>
Schwacher Host sendet	:	disabled

*Ausgabe(Auszug) show interface 11 für LAN1 nach Einschalten der Ankündigung*

Der Parameter Ankündigung ist nun aktiviert; das O-Flag und das M-Flag sind ausgeschaltet.

Im nächsten Schritt muss noch eine Veröffentlichung des Netzpräfixes gestattet werden.

- ▶ Überprüfen Sie zunächst den Status der Veröffentlichung:  
show route

C:\>netsh int ipv6 show route					
Veröff.	Typ	Met	Präfix	Idx	Gateway/Schnittstelle
Nein	System	256	::1/128	1	Loopback Pseudo-Interface
Nein	Manuell	256	fd00:20:0:1::254/128	11	LAN-Verbindung
<b>Nein</b>	<b>Manuell</b>	<b>256</b>	<b>fd00:20:0:1::/64</b>	<b>11</b>	<b>LAN-Verbindung</b>
Nein	System	256	fe80::/64	11	LAN-Verbindung

*Noch ausgeschaltete Veröffentlichung des Netzpräfixes*

Die LAN-Verbindung .../64 weist aus, dass die Veröffentlichung noch nicht stattgefunden hat.

- ▶ Schalten Sie die Veröffentlichung des Netzpräfixes ein:  
netsh interface ipv6 set route fd00:20:0:1::/64 11 publish=yes

Sofern die Route noch nicht vorhanden ist (keine IPv6-adresse vergeben), muss add statt set benutzt werden:

netsh interface ipv6 add route fd00:20:0:1::/64 11 publish=yes

C:\>netsh interface ipv6 show route					
Veröff.	Typ	Met	Präfix	Idx	Gateway/Schnittstelle
Nein	System	256	::1/128	1	Loopback Pseudo-Interface
Nein	Manuell	256	fd00:20:0:1::254/128	11	LAN-Verbindung
<b>Ja</b>	<b>Manuell</b>	<b>256</b>	<b>fd00:20:0:1::/64</b>	<b>11</b>	<b>LAN-Verbindung</b>
Nein	System	256	fe80::/64	11	LAN-Verbindung

*Veröffentlichung eingeschaltet*

Der Veröffentlichungsvorgang benötigt einige Sekunden. Sie überprüfen die erfolgreichen Einstellungen auf dem Client und dem DNS-Server.

- Lassen Sie sich **auf dem Client** die IPv6-Adresse anzeigen:  
ipconfig /all

Ethernet-Adapter LAN-Verbindung:

```
Verbindungsspezifisches DNS-Suffix: example.net
Beschreibung. . . . . : Broadcom NetXtreme 57xx-Gigabit
Physische Adresse . . . . . : 00-1D-09-DE-B8-EC
DHCP aktiviert. . . . . : Ja
Autokonfiguration aktiviert . . . : Ja
IPv6-Adresse. . . . . :
fd00:20:0:1:21d:9ff:fede:b8ec(Bevorzugt)
Verbindungslokale IPv6-Adresse . . . : fe80::21d:9ff:fede:b8ec%11 (Bevorzugt)
IPv4-Adresse . . . . . : 172.16.20.101 (Bevorzugt)
Subnetzmaske . . . . . : 255.255.255.0
```

*IP-Config auf dem Client, nachdem die IPv6-Adresse veröffentlicht wurde*

- Rufen Sie mit nslookup die IP-Adressen des Clients ab.  
Es kann einige Zeit in Anspruch nehmen, bis der Eintrag erscheint.

```
C:\>nslookup IPv6CLIENT1
Server:      ipv6srv1
Address:     fd00:20:0:1::1

Name:         IPv6CLIENT1
Addresses:   fd00:20:0:1:21d:9ff:fede:b8ec
              172.16.20.201
```

*DNS-Einträge für die IPv6 Adresse*

Dem Clientrechner wurde über den Router das Netzpräfix mitgeteilt und daran die Host-ID angehängt. Die Hostadresse ist eine Adresse nach dem EUI-64-Standard, wie am Mittelteil des Hostanteils ff:fe zu erkennen ist.

Als nächstes muss die Standardroute bekannt gegeben werden.

Wechseln Sie zu Router1 und geben Sie folgenden Befehl ein (Sofern der Dienst Routing und RAS nicht installiert wurde, muss der Parameter Weiterleitung (Forwarding) auf der entsprechenden Netzwerkkarte aktiviert werden.):

- Netsh interface ipv6 set interface 11 forwarding=enabled

Wenn noch keine Defaultroute eingetragen ist, muss diese vor der Veröffentlichung hinzugefügt werden:

- netsh interface ipv6 add route ::/0 11 publish=yes

Danach muss der Router sich selbst als Standardgateway bekannt machen:

- netsh interface ipv6 set interface 11 advertisedefaultroute=enabled

Damit erfolgt eine Ankündigung der Defaultroute, und nach wenigen Sekunden ist der Routing-Eintrag beim Client vorhanden.

- Wechseln Sie zu Client1 und geben Sie den folgenden Befehl ein: ipconfig /all

**Ethernet-Adapter LAN-Verbindung:**

```
Verbindungsspezifisches DNS-Suffix: example.net
IPv6-Adresse . . . . . : fd00:20:0:1:21d:9ff:fede:b8ec
Verbindungslokale IPv6-Adresse . . : fe80::21d:9ff:fede:b8ec%11
IPv4-Adresse . . . . . : 172.16.20.101
Subnetzmaske . . . . . : 255.255.255.0
Standardgateway . . . . . : fe80::21d:9ff:fede:b8ec%11
                                         172.16.20.10
```

Sie sehen hier, dass nicht die Unique-Local-Adresse des Routers, sondern dessen Link-Local-Adresse übergeben wird.

Damit sind alle Informationen übergeben, die per Stateless Autoconfiguration vom Client erhalten werden können.

## 5.8 Praxisbeispiel Stateless Autoconfiguration mit DHCPv6 (Stateless DHCPv6)

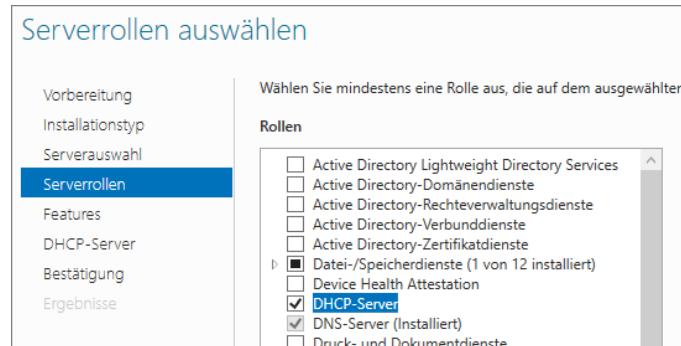
### Vorbereitung

Für diese Art der Konfiguration benötigen Sie einen DHCP-Server, der die entsprechenden Informationen wie z. B. DNS und Domain-Name bereitstellt.

Im statusfreien Modus verteilt der DHCPv6-Server keine IP-Adressen an Clients. Er unterstützt lediglich die Stateless Autoconfiguration mit zusätzlichen Informationen wie z. B. den DNS-Server. Damit die Konfigurationsinformationen vom DHCP-Server kommen können, wird der Router im weiteren Verlauf wieder entsprechend zurückgesetzt.

### DHCP-Serverrolle installieren

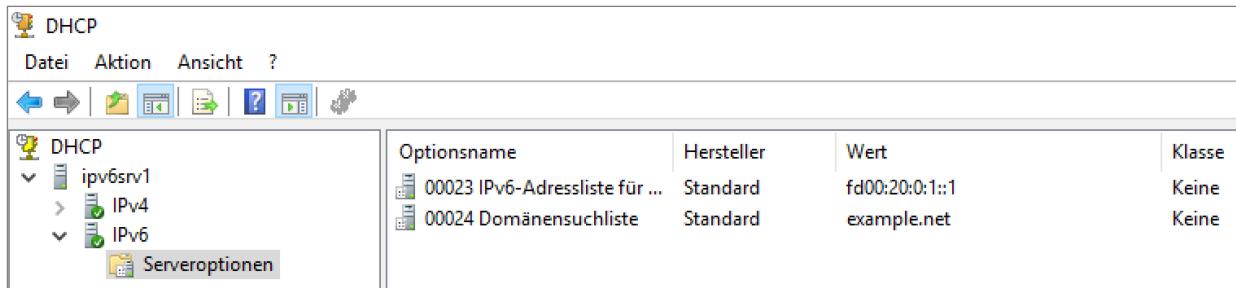
Auf Server1 ist noch kein DHCP-Server vorhanden. Als Erstes wird daher die DHCP-Serverrolle über den Servermanager installiert. Die IPv4-Einstellungen sind im Moment nicht von Bedeutung. In den DHCP-Einstellungen für IPv4 im Assistenten wird einfach das bestehende, zurzeit mit festen IPs konfigurierte IPv4-Netz eingerichtet.



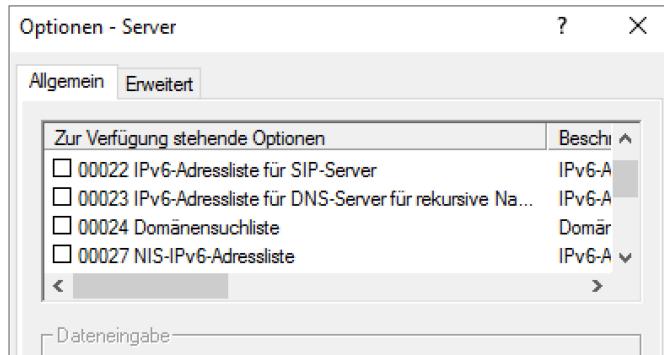
### DHCP-Server installieren

Bei Server 2008 musste bei der Installation noch zwischen dem statusfreien und statusbehaftetem Modus ausgewählt werden. Eine nachträgliche Änderung des Modus war laut Microsoft nicht möglich.

Seit Server 2012 entscheidet ein vorhandener oder nicht vorhandener Bereich zwischen den beiden Möglichkeiten. Zunächst wird der statusfreie Modus vorbereitet. Dazu werden im DHCP-Manager die Serveroptionen konfiguriert.



*IPv6 Stateless DHCP, mit Serveroptionen*



*Auswahl der Serveroptionen*

Abschließend teilen Sie den Hosts mit, dass im Netzwerk eine *Andere statusbehaftete Konfiguration* – nämlich die vom DHCP-Server – vorhanden ist. Dazu wird auf der Netzwerkschnittstelle des Routers das O-Flag für *Other Configuration Information* gesetzt.

- Öffnen Sie hierzu die Konsole des Routers und aktivieren Sie den Netshell-IPv6-Kontext:

```
netsh interface ipv6 set interface 11 otherstateful=enabled
```

- Zeigen Sie die Schnittstelle an:

```
netsh interface ipv6 show interface 11
```

Die entsprechende Einstellung wird als **enabled** ausgewiesen.

Ankündigung	:	enabled
Nachbarsuche	:	enabled
Nachbar-Nichterreichbarkeitserkennung	:	enabled
Routersuche	:	enabled
Verwaltete Adresskonfiguration	:	disabled
<b>Andere zustandsbehaftete Konfiguration</b>	:	<b>enabled</b>

*Ausschnitt show Interface*

## Auswirkungen auf den Client prüfen

Die Auswirkung auf den Client lässt sich dort nach einigen Sekunden abrufen.

- ▶ Geben Sie am Client1 an der Konsole folgenden Befehl ein:  
ipconfig /all

```
Lease erhalten. . . . . : Freitag, 29. September 2017 12:51:39
Lease läuft ab. . . . . : Montag, 9. Oktober 2017 12:51:40
Standardgateway . . . . . :
DHCPv6-IAID . . . . . : 268442889
DHCPv6-Client-DUID. . . . . : 00-01-00-01-1D-3B-C1-2A-00-1D-09-DE-B8-EC
DNS-Server . . . . . . . : fd00:20:0:1::1
Suchliste für verbindungsspezifische DNS-Suffixe: example.net
```

### *Suchdomäne und DNS-Einstellungen auf dem Client*

Im Bildausschnitt sehen Sie die DNS-Serverangabe und das DNS-Suffix (Suchdomäne).

Im Testnetz erhalten nun alle Hosts die Informationen zur Suchdomäne und den konfigurierten DNS-Servern vom DHCPv6-Server. Damit ist die Konfiguration des statusfreien Modus abgeschlossen.

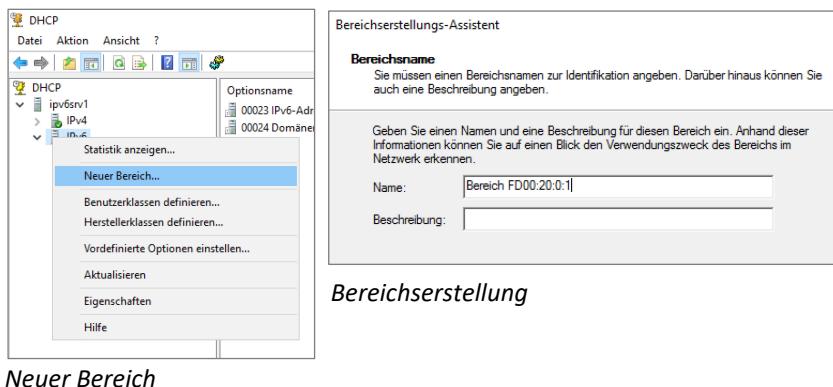
## 5.9 Praxisbeispiel Konfiguration des DHCPv6-Servers unter MS Windows Server 2016

### DHCPv6 im statusbehafteten Modus einrichten

Sofern über die Routeradvertisements das O-Flag gesetzt wurde, werden die Hosts Informationen – wie oben beschrieben – vom DHCPv6-Server anfordern. Sind dagegen im statusbehafteten Modus das M- und das O-Flag gesetzt, erwarten die Hosts auch eine IPv6-Adresse vom DHCPv6-Server.

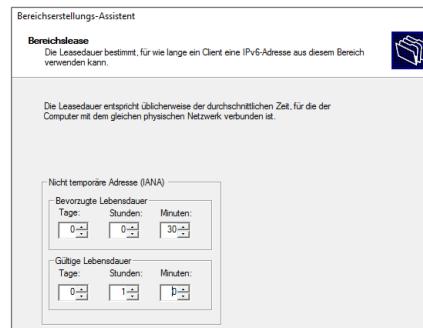
Dazu wird im DHCP-Manager unter dem Knoten IPv6 über das Kontextmenue oder über Aktionen ein neuer Bereich angelegt.

Anders als bei IPv4 wird hier kein IP-Adressbereich, sondern der Netzwerkpräfix eingetragen. Die Präfixlänge ist immer 64 und wird automatisch gesetzt.



Im nächsten Fenster können Bereiche aus der Verteilung ausgeschlossen werden. Da DHCPv6 immer alle 64 Bits des Hostidentifiers setzt, ist bei Vergabe von verkürzten Serveradressen ein Ausschluss in der Regel nicht notwendig.

Wie bei IPv4 wird die Leasedauer gesetzt, hier zu Demozwecken auf 30 Minuten bevorzugt und 1 Stunde gültige Lebensdauer. Im praktischen Betrieb wird man die Zeiten länger setzen, Windowsstandard ist 8/12 Tage.



*Einstellen der Leasedauer*

Wie bei IPv4 können auch Bereichsoptionen vergeben werden. Die Vorgehensweise ist die gleiche wie oben bei den Serveroptionen beschrieben. Sofern auf Bereichsebene keine Optionen definiert sind, gelten die Serveroptionen.

Abschließend teilen Sie den Hosts mit, dass im Netzwerk eine *Andere statusbehaftete Konfiguration* – nämlich die vom DHCP-Server – vorhanden ist. Dazu wird auf der Netzwerkschnittstelle des Routers zusätzlich zum O-Flag noch das M-Flag für Managed Configuration gesetzt.

- ▶ Öffnen Sie hierzu die Konsole des Routers und aktivieren Sie den Netshell-IPv6-Kontext:

```
netsh interface ipv6 set interface 11 managedaddress=enabled
```

- ▶ Zeigen Sie die Schnittstelle an:

```
netsh interface ipv6 show interface 11
```

Verwaltete Adresskonfiguration	:	enabled
Andere zustandsbehaftete Konfiguration	:	enabled

*Show interface, M- und O-Flag gesetzt*

### Mit der abgeschlossenen Konfiguration arbeiten

Damit ist die Konfiguration abgeschlossen. Alle Clients beziehen nun ihre IPv6-Adresse und weitere Informationen über die jeweilige Netzwerkinfrastruktur, wie DNS-Server und Domänen suchliste, vom DHCP-Server. Nur Standard-gateway und evtl. weitere Präfixe kommen vom Router über **Router Advertisement**. Der DHCP-Server verwaltet und dokumentiert die vergebenen IPv6-Adressen. Bei jeder Vergabe wird außerdem der Nameserver benachrichtigt und erstellt die passenden Einträge für Forward und Reverse Lookup.

Die Konfiguration kann nun auf dem Client mit ipconfig /renew6 sofort aktiviert und mit ipconfig /all geprüft werden.

Wichtig: Auch bei Stateful DHCPv6 sind Router Advertisements erforderlich. Der Client erhält vom DHCPv6-Server lediglich eine IPv6-Adresse, keine Route ins eigene Netz.

C:\>netsh interface ipv6 show route	
Veröff. Typ Met Präfix	Idx Gateway/Schnittstelle
Nein System 256 ::1/128	1 Loopback Pseudo-Interface 1
Nein System 256 fd00:20:0:1:21d:9ff:fede:b8ec/128	11 LAN-Verbindung
Nein System 256 fe80::/64	11 LAN-Verbindung
Nein System 256 fe80::21d:9ff:fede:b8ec/128	11 LAN-Verbindung

*show route: Es fehlt die Route ins eigene Netz fd00:20:0:1::/64*

Eine Besonderheit bei IPv6 ist, dass auch Hosts mit statisch konfigurierter IPv6-Adresse eine zusätzliche Adresse vom DHCPv6-Server erhalten, siehe unten.

DHCP	Client-IPv6-Adresse	Name	Leaseablaufdatum
ipv6srv1 IPv4 IPv6 Bereich [fd00:20:0:1::] Bereich Adressleases	fd00:20:0:1:bd62:... IPv6SRV1.example....		14.09.2017 13:59:12

*DHCPv6-Lease des Servers im DHCP-Manager...*

```
Verbindungsspezifisches DNS-Suffix: example.net
IPv6-Adresse. . . . . : fd00:20:0:1::1
IPv6-Adresse. . . . . : fd00:20:0:1::bd62:9c38:130d:c994
Verbindungslokale IPv6-Adresse . : fe80::b09c:b9fe:835b:65de%6
```

*... und im ipconfig des Servers*

## Besonderheiten bei Windows



Windows hält sich leider nicht in allen Punkten an die Vorgaben der RFCs. Alle Windowsversionen sind nicht in der Lage, DNS-Server und DNS-Suchliste aus den Router Advertisements nach RFC 8106 zu konfigurieren. Bei den Client Betriebssystemen gibt es mit dem Windows 10 Creators Update vom Sommer 2017 Besserung.

Ab Windows 8/Server2012 werden die M und O Flags ignoriert. Anders als bei Windows 7 beziehen die Hosts unabhängig von M- und O-Flag eine DHCPv6-Konfiguration, sofern DHCPv6 nicht deaktiviert wurde. Auf Servern sollte es deaktiviert werden, da dort in der Regel die Netzwerkschnittstellen statisch konfiguriert werden.

Das Phänomen der mehrfachen Adressen, DHCP und SLAAC, tritt bei den meisten Betriebssystemen auf, dort allerdings nur wenn M- und O-Flag gesetzt sind.

# 6 Netzwerkkontexte

## In diesem Kapitel erfahren Sie

- ✓ welche Bedeutung ICMPv6 für die IPv6-Kommunikation hat
- ✓ was sich hinter MTU verbirgt
- ✓ wie sich die MTU bei verschiedenen Übertragungsprotokollen verhält

## Voraussetzungen

- ✓ Schichtenmodelle
- ✓ IPv6-Adressaufbau und -Konfiguration

## 6.1 ICMPv6

### ICMP bei IPv6

Das von IPv4 bekannte ICMP gibt es auch in IPv6, allerdings mit einem wesentlich höheren Stellenwert. ICMPv6 ist unverzichtbar für eine funktionierende IPv6-Infrastruktur (vgl. Kapitel 4). ICMPv6 ist in RFC 4443, ergänzt durch RFC 4884 definiert.

Im Kapitel 2 wurde gezeigt, wie die Zuordnung einer physischen Ethernetadresse zu einer logischen IPv4-Adresse mit dem ARP-Protokoll funktioniert. ARP vermittelt dabei zwischen der OSI-Schicht 2 und 3. Bei IPv6 wird diese Aufgabe von ICMPv6, genauer vom Neighbor Discovery Protokoll (NDP, RFC 4861), erledigt. Die Kommunikation findet ausschließlich in der OSI-Schicht 3 statt. Dazu muss sichergestellt werden, dass die relevanten Datenpakete nicht von Firewalls geblockt werden.

### ICMPv6-Meldungen

ICMPv6 ist gemeinsam mit den Netzwerkprotokollen IP, IPv6 und weiteren in der Schicht 3 des OSI-Modells angesiedelt. Die wichtigen Meldungen werden in verschiedenen Kapiteln im jeweiligen Zusammenhang näher erläutert. Deshalb erhalten Sie an dieser Stelle nur eine einführende Übersicht.

### Headeraufbau

ICMPv6-Meldungen werden dem IPv6-Header als Nutzlast mit der Protokollnummer 53 angehängt. Die Meldungen lassen sich grob in die Typen **Fehler** und **Information** aufteilen.

**Fehlermeldungen** sind z. B.:

- |                             |                       |
|-----------------------------|-----------------------|
| ✓ 1 Destination Unreachable | ✓ 3 Time Exceeded     |
| ✓ 2 Packet Too Big          | ✓ 4 Parameter Problem |

Die Meldungen sind in ähnlicher Form auch bei ICMPv4 vorhanden. Eine besondere Bedeutung kommt dem Fehler 2 zu, der im nächsten Abschnitt erläutert wird.

**Informationsmeldungen** werden vom **Neighbor Discovery Protokoll (NDP)** genutzt. NDP ersetzt das aus IPv4 bekannte ARP und sorgt für eine Verteilung von wichtigen Informationen zur Netzwerkinfrastruktur.

- ✓ 128 Echo Request
- ✓ 129 Echo Reply
- ✓ 130 Multicast Listener Query
- ✓ 131 Version 1 Multicast Listener Report
- ✓ 132 Multicast Listener Done
- ✓ 133 Router Solicitation
- ✓ 134 Router Advertisement
- ✓ 135 Neighbor Solicitation
- ✓ 136 Neighbor Advertisement
- ✓ 137 Redirect Message

### Informationstypen für die Adresszuordnung

Neben dem aus ICMP bekannten Echo Request/Reply (ping), finden Sie hier zwei wichtige Informations-typen für die Adresszuordnung:

<b>Neighbor Solicitation</b>	Anfrage an die Nachbarschaft
<b>Neighbor Advertisement</b>	Antwort auf die Anfrage, Nachbarschafts-Anmeldung
<b>Router Solicitation</b>	Anfrage nach Routern im Netzbereich
<b>Router Advertisement</b>	Antwort, Bekanntmachung der Routerinformation

Mit diesen und weiteren Meldungen wird der Neighbor-Cache des Hostrechners gefüllt und kann so ohne erneute Anfrage auf Netzwerkressourcen zugreifen.

Den Cache können Sie unter Windows mit netshell abfragen:

```
netsh interface ipv6 show neighbors 11
```

Schnittstelle 11: LAN-Verbindung	Internetadresse	Physische Adresse	Typ
-			
fd00:ada:cafe:bad:224:feff:fee3:22a2 (Router)		00-24-fe-e3-22-a2	Abgelaufen
fe80::224:feff:fee3:22a2 (Router)		00-24-fe-e3-22-a2	Erreichbar
ff02::1		33-33-00-00-00-01	Permanent
ff02::2		33-33-00-00-00-02	Permanent
ff02::c		33-33-00-00-00-0c	Permanent
ff02::16		33-33-00-00-00-16	Permanent
ff02::1:2		33-33-00-01-00-02	Permanent
ff02::1:3		33-33-00-01-00-03	Permanent
ff02::1:ff00:55		33-33-ff-00-00-55	Permanent
ff02::1:ff75:bea7		33-33-ff-75-be-a7	Permanent
ff02::1:ffc2:8268		33-33-ff-c2-82-68	Permanent
ff02::1:ffe3:22a2		33-33-ff-e3-22-a2	Permanent
ff0e::c		33-33-00-00-00-0c	Permanent

Gut zu erkennen ist die Zuordnung der logischen IP-Adresse (Internetadresse) zur MAC-Hardwareadresse (Physische Adresse). Der Router taucht hier mit seiner Link-Local- (fe80:...) und der Unique-Local- (fd00:...)Adresse auf. Die Kommunikation zum Router findet immer über die Link-Local-Adresse statt (vgl. Kapitel 4). Die mit 33-33 beginnenden MAC-Adressen sind für Multicast reserviert. An die 33-33 schließt sich der Link-Local-Teil der Multicast-Adresse an.

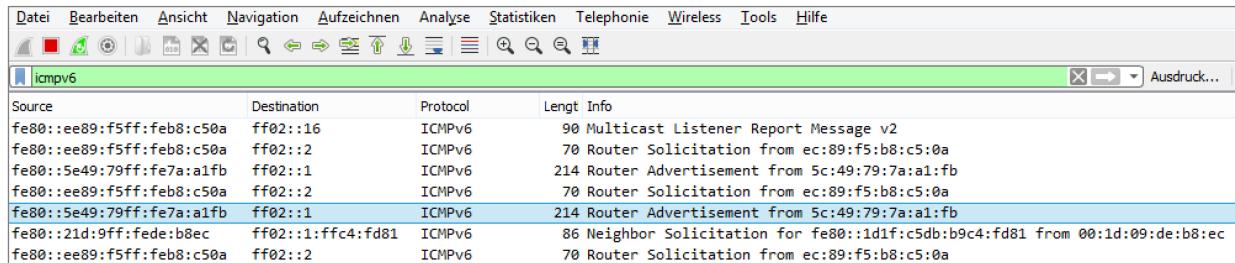
Die Einträge haben eine begrenzte Lebensdauer (TTL – Time to Live) und werden durch wiederholte Solicitation/Advertisement erneuert.

Unter Debian Linux wird der NDP-Cache mit `ip -f inet6 neigh` abgefragt:

```
# ip -f inet6 neigh
2a01:4f8::a:19:16 dev eth0 lladdr 00:26:88:75:cb:a8 PROBE
fe80::1 dev eth0 lladdr 00:26:88:75:cb:a8 router REACHABLE
```

Auch hier sind zwei IP-Adressen des Routers mit einer identischen MAC-Adresse verknüpft, genutzt wird jedoch auch hier nur die Link-Local-Adresse `FE80::1`.

Mit dem Analyse-Tool Wireshark können die NDP-Pakete veranschaulicht werden:



#### Wireshark: NDP-Pakete in einem Netzwerk

In der Detailansicht des Wireshark lassen sich die einzelnen Bestandteile der Netzwerkpakete analysieren, hier ein Router Advertisement (RA):

```
Type: Router Advertisement (134)
Code: 0
Checksum: 0x07d2 [correct]
Cur hop limit: 255
Flags: 0x48
    0... .... = Managed address configuration: Not set
    .1... .... = Other configuration: Set
    ..0.... = Home Agent: Not set
    ...0 1... = Prf (Default Router Preference): High (1)
    .... 0.. = Proxy: Not set
    .... ..0. = Reserved: 0
Router lifetime (s): 1800
Reachable time (ms): 0
Retrans timer (ms): 0
+ ICMPv6 Option (Prefix information : 2001:6f8:1d44::/64)
+ ICMPv6 Option (Prefix information : fd00:ada:cafe:bad::/64)
+ ICMPv6 Option (Recursive DNS Server fd00:ada:cafe:bad:224:feff:fee3:22a2)
+ ICMPv6 Option (MTU : 1470)
+ ICMPv6 Option (source link-layer address : 00:24:fe:e3:22:a2)
```

#### Router Advertisement Detail

Im Screenshot sind die Flags des Routers (vgl. Kapitel 7) zu erkennen. Darunter sind die mitgeteilten Präfixe für Global Unicast und Unique Local Unicast aufgelistet, darauf folgt die IP des DNS-Servers, die eingestellte MTU und die Quell-MAC-Adresse.

#### Auswahl ICMP-Typen

Type	Name	Reference
0	Reserved	
1	Destination Unreachable	[RFC 4443]
2	Packet Too Big	[RFC 4443]
3	Time Exceeded	[RFC 4443]

Type	Name	Reference
4	Parameter Problem	[RFC 4443]
100	Private experimentation	[RFC 4443]
101	Private experimentation	[RFC 4443]
102-126	Unassigned	
127	Reserved for expansion of ICMPv6 error messages	[RFC 4443]
128	Echo Request	[RFC 4443]
129	Echo Reply	[RFC 4443]
130	Multicast Listener Query	[RFC 2710]
131	Multicast Listener Report	[RFC 2710]
132	Multicast Listener Done	[RFC 2710]
133	Router Solicitation	[RFC 4861]
134	Router Advertisement	[RFC 4861]
135	Neighbor Solicitation	[RFC 4861]
136	Neighbor Advertisement	[RFC 4861]
137	Redirect Message	[RFC 4861]
138	Router Renumbering	[Matt_Crawford]
139	ICMP Node Information Query	[RFC 4620]
140	ICMP Node Information Response	[RFC 4620]
141	Inverse Neighbor Discovery Solicitation Message	[RFC 3122]
142	Inverse Neighbor Discovery Advertisement Message	[RFC 3122]
143	Version 2 Multicast Listener Report	[RFC 3810]
144	Home Agent Address Discovery Request Message	[RFC 6275]
145	Home Agent Address Discovery Reply Message	[RFC 6275]
146	Mobile Prefix Solicitation	[RFC 6275]
147	Mobile Prefix Advertisement	[RFC 6275]
148	Certification Path Solicitation Message	[RFC 3971]
149	Certification Path Advertisement Message	[RFC 3971]
150	ICMP messages utilized by experimental mobility protocols such as Seamoby	[RFC 4065]
151	Multicast Router Advertisement	[RFC 4286]
152	Multicast Router Solicitation	[RFC 4286]
153	Multicast Router Termination	[RFC 4286]
154	FMIPv6 Messages	[RFC 5568]
155	RPL Control Message	[RFC 6550]
156	ILNPv6 Locator Update Message	[RFC 6743]
157	Duplicate Address Request	[RFC-ietf-6lowpan-nd-21]

Type	Name	Reference
158	Duplicate Address Confirmation	[RFC-ietf-6lowpan-nd-21]
159-199	Unassigned	
200	Private experimentation	[RFC 4443]
201	Private experimentation	[RFC 4443]
255	Reserved for expansion of ICMPv6 informational messages	[RFC 4443]

Liste der wichtigsten ICMP-Meldungen auf der IANA.ORG-Seite



Eine Übersicht über alle ICMPv6-Meldungen finden Sie unter [www.iana.org/assignments/icmpv6-parameters/icmpv6-parameters.xhtml](http://www.iana.org/assignments/icmpv6-parameters/icmpv6-parameters.xhtml).

## 6.2 Maximum Transmission Unit (MTU)

### MTU berechnen

Es existiert kaum ein Anwenderforum, in dem Sie nicht Tipps finden, wie die richtige MTU-Größe einzustellen ist. Die Beurteilung der Einstellungsvorschläge zur Größe der MTU kann erst aufgrund einer genauen Kenntnis der Berechnung der MTU erfolgen.

Die **Maximum Transmisson Unit** (MTU) ist eine der wichtigsten Größen bei der Übertragung von Daten in Netzwerken. Sie gibt den Umfang der Nutzlast (engl. **Payload**), also der maximalen Größe des zu übertragenden Datenpaketes (Datagramms) in der Netzwerkschicht (OSI 3) an. Als **Datagramm** wird das gesamte Paket der Schicht 3 (OSI) bezeichnet, z. B. ein IPv6-Paket inklusive TCP-Paket und den darin enthaltenen Nutzdaten.

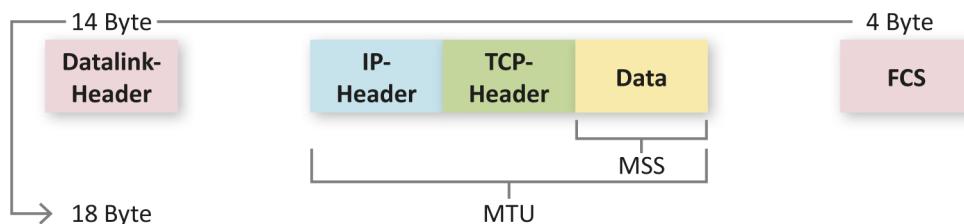
Die Länge der MTU wird in Bytes angegeben, die Einheit wird aber oft weggelassen (z. B. MTU 1500). Am Beispiel eines Ethernet-Frames soll gezeigt werden, wie sich die MTU berechnet:

Die Länge eines Ethernet-Frames setzt sich aus drei Elementen zusammen:

- ✓ einem 14 Bytes großen Header mit Informationen über Absender und Empfänger sowie einigen Kontrollinformationen,
- ✓ dem Paket der Vermittlungs- bzw. Netzwerkschicht (IP, IPv6) mit 1500 Bytes, das wiederum die Daten der höheren OSI-Schichten enthält,
- ✓ einem 4 Byte langen Trailer mit der Checksumme des Ethernet-Frames.

Bei Verwendung von VLAN (Virtual LAN) wird der Ethernet-Header um ein 4 Byte langes VLAN-Tag ergänzt. Die MTU wird davon nicht beeinflusst.

Dieses Diagramm zeigt die Maximum Segment Size (MSS) und die MTU. Unbeachtet der MTU gibt es einen 18 Byte großen Overhead vom Datalink Layer (OSI Schicht 2).



Grafik zur Berechnung der MTU

Die Maximum Segment Size (MSS) ist eine weitere Kenngröße, die sich aus MTU minus IP-Header minus TCP-Header errechnet. Die MTU errechnet sich aus IP-Header plus TCP-Header plus Daten.

Die Herausforderung besteht darin, die Daten so zu verschachteln, dass die MTU nicht überschritten wird. In dem Fall sollten Daten von der Anwendung „kleiner gepackt“ werden.

## Fragmentierung

Das im IPv4-Header vorhandene **Don't Fragment** (DF)-Feld reguliert die Fragmentierung der IP-Pakete. Wenn es gesetzt ist, werden die Pakete nicht automatisch fragmentiert. Stattdessen schickt das verarbeitende Gerät, z. B. ein Router, einen ICMP-Fehler Typ 3, Code 4 (Destination Unreachable Fragmentation Needed, DF Set). Der sendende Rechner reagiert darauf und reduziert die Paketgröße entsprechend.

Die sorglosere Variante ist, das DF-Bit im Header nicht zu setzen. In dem Fall würde die Arbeit auf den Router abgewälzt, der nun die Pakete von sich aus fragmentiert. Sollte ein Fragment eines IP-Paketes verloren gehen, muss die komplette Sequenz neu angefordert werden. Fragmentierung kann einen großen Protokoll-Overhead erzeugen.

## Keine Fragmentierung bei IPv6

Bei IPv6 wurde deshalb bewusst auf das DF-Bit verzichtet. IPv6-Pakete werden auf dem Transportweg nicht fragmentiert. Bei zu großen Paketen erhält der Absender einen ICMPv6-Fehler Typ 2 (Packet Too Big). Damit weiß der sendende Host, dass er die MTU/MSS reduzieren muss.

Packet-Too-Big-Schema				
+	Bits 0–7	Bits 8–15	Bits 16–23	Bits 24–31
0	Type	Code	Prüfsumme	
32	MTU			
...	Fehlerhaftes Protokoll			

## Path MTU Discovery (PMTUD)

Um die MTU von vornherein festlegen zu können, gibt es entsprechende Hilfsprogramme. Im einfachsten Fall können Sie mit ping-Optionen die Paketgröße so lange erhöhen, bis die entsprechenden ICMP/ICMPv6-Meldungen kommen.

Für Windows gibt es das frei verfügbare Befehlszeilenprogramm *mtupath.exe*. Nach Eingabe von *mtupath host* werden Pakete verschiedener Größe gesendet und die Antworten ausgewertet. Mit der Ausgabe erhalten Sie für den Pfad zum Zielrechner Hinweise auf MTU und MSS. Der Durchsatz wird vom schwächsten Glied in der Kette bestimmt, also von der Station, die den kleinsten MTU-Wert verarbeiten kann.

```
PS C:\Windows\system32> mtupath ix.de
MTU path scan to ix.de (2a02:2e0:3fe:100::8), ttl=64, limit=48
# 16 processing - best MSS 1422 (estimated MTU 1470) [uUUUUuUuuUUUUuuUu]

#1 MSS IN RANGE      1 <== 1421 ==> 1422
#2 MSS EXCEEDED    1423 <== 14961 ==> 16384
```

### Ermitteln der MTU mit *mtupath*

Sie sehen, dass standardgemäß ohne Angabe der Option –6 die IPv6-Adresse des Hosts, hier *ix.de*, gewählt wird. In diesem Fall ist für die Schnittstelle und den Router eine MTU von 1470 eingestellt, die auch erreicht wird.

Wichtig ist bei PMTUD, dass ICMP/ICMPv6 nicht irgendwo auf dem Pfad durch Firewall-Regeln geblockt wird (vgl. Kapitel 10). In dem Fall würden die Statusmeldungen den Absender nicht erreichen. Stattdessen würde er vergeblich auf die Quittungen (ACK) für die gesendeten Pakete warten, die irgendwo auf dem Weg verworfen wurden. Hier wird noch einmal deutlich, dass die MTU-Werte in der Regel automatisch von den beteiligten Stationen eingestellt werden. Es sind allenfalls administrative Eingriffe bei den beteiligten Routern notwendig.

Im folgenden Abschnitt werden Besonderheiten bei verschiedenen Trägermedien/-protokollen dargestellt.

## 6.3 IP over Everything

### Unterschied zwischen „IP over Everything“ und „Everything over IP“

„IP over Everything“ wird oft in einem Atemzug mit dem Begriff „Everything over IP“ genannt, der allerdings etwas anderes aussagt.

Bei **Everything over IP** geht es darum, möglichst alle Dienste in einem IP-Netzwerk unterzubringen. Beispiel dafür sind z. B. Video Streaming (IP-TV) und Voice Over IP (VoIP), Provider arbeiten daran, die althergebrachten Telefonnetze durch reine IP-Netzwerke zu ersetzen, dem Next Generation Network (NGN).

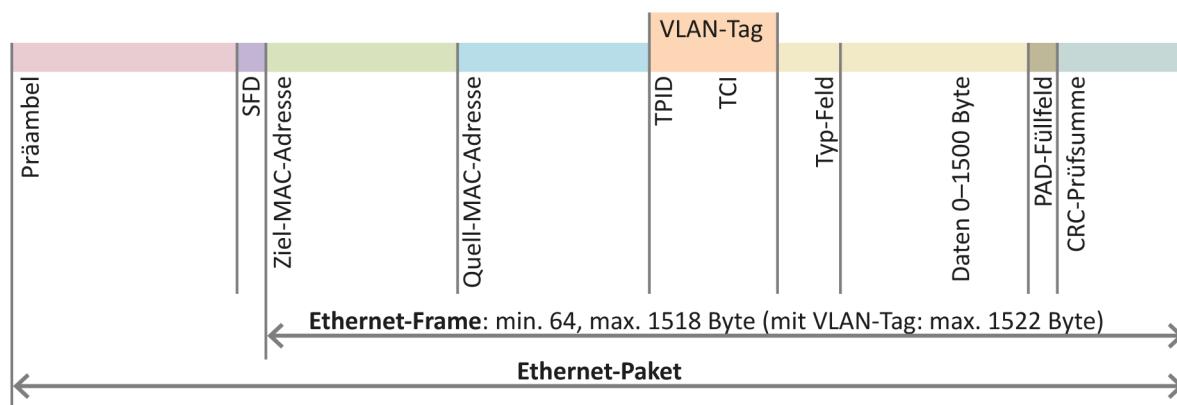
Bei **IP over Everything** geht es darum, Datenpakete über alle gängigen Übertragungsmedien und Protokolle zu übermitteln, also erst einmal die Grundlage für ein allumfassendes IP-Netz zu schaffen.

### Ethernet IEEE 802.3

Das bekannteste Schicht-2-Protokoll ist Ethernet nach IEEE 802.3 (RFC 2464), das mit verschiedensten Übertragungsmedien funktioniert. Es ist eine Weiterentwicklung des für Funkübertragung verwendeten Aloha-Protokolls und lässt sich auf das Jahr 1973 zurückführen, ist also mittlerweile über 50 Jahre im Einsatz (TCP/IP ist 20 Jahre jünger).

Ethernet arbeitet mit sogenannten Frames (Rahmen), die eine Länge von 1518 Bytes (mit VLAN 1522 Bytes) haben. Nach Abzug der für die Organisation benötigten Bytes ergibt sich die bereits oben errechnete MTU von maximal 1500 Bytes.

Zugriffsverfahren, also die Art der Datenübertragung im Ethernet ist CSMA/CD. Ein Problem bei Ethernet in einer Bus-Topologie (vgl. Kapitel 7) ist, dass alle Stationen gleichberechtigt sind (MA – Multiple Access). Obwohl Stationen darauf achten, dass die Leitung frei ist (CS – Carrier Sense) kommt es zu Datenkollisionen, die zwar vom Netz erkannt werden (CD – Collision Detection), aber bei einer großen Anzahl von Stationen in einer Kollisionsdomäne den Durchsatz verringern. In der Spezifikation sind deshalb maximale Segmentlängen und Anzahl von Stationen pro Segment festgelegt. In der Regel wird heute eine Stern-Topologie verwendet, bei der die Kollisionsdomäne auf zwei Stationen beschränkt ist. Durch Duplexbetrieb, senden und empfangen auf verschiedenen Adernpaaren, sollten bei intakten Netzwerkkomponenten keine Kollisionen auftreten.



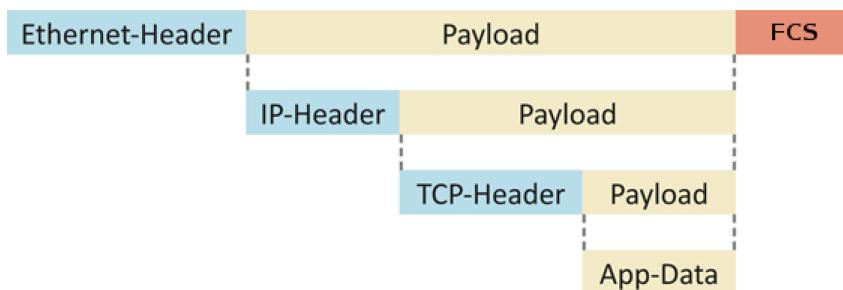
Ethernet-Paket (Layer1) mit Ethernet-Frame mit VLAN-Tag (Layer2) – Aufbau

## Ethernet Frame im Detail

Die Präambel ist eine maximal sieben Bytes lange Bitfolge von Nullen und Einsen (10101010), die zur Synchronisation des Ethernet Frames auf dem Übertragungsmedium dient. Durch die gleichmäßige Bitfolge kann das Signal gut von Störungen unterschieden werden. Beim achten Byte ist die Bitfolge verändert (10101011) und kündigt so den „Start of Frame“ (SOF/SFD) an. In neueren Netzwerkarchitekturen wäre das nicht mehr notwendig, wird aber aus Kompatibilitätsgründen weitergeführt.

Der eigentliche Frame beginnt mit Ziel- und Quell-MAC-Adresse (einheitliche Hardwarekennung), gefolgt vom optionalen VLAN-Tag. Im Feld Ether-Type wird die Art der Nutzlast festgelegt, z. B. 0x800 für IPv4 oder 0x86DD für IPv6. Darauf folgen die Daten der höheren Schicht. Das PAD-Feld wird nur für den Fall benötigt, dass die Nutzdatenmenge nicht ausreicht, um einen Frame mit mindestens 64 Bytes Länge zu erzeugen. In dem eher seltenen Fall springt das PAD-Feld ein und füllt das Datenfeld mit den fehlenden Bytes. Die höheren Schichten müssen das erkennen, damit die Fülldaten nicht als Nutzdaten interpretiert werden. Der Frame endet mit der FCS (Frame Check Sequenz), die eine Prüfsumme über den gesamten Frame beginnend bei der Ziel-MAC bis einschließlich PAD-Feld enthält.

In lokalen Netzen findet die Kommunikation über die Schicht 2 statt (vgl. Kapitel 2, 4). Die IP-Pakete werden also innerhalb des Ethernet Frames übertragen und am Zielrechner ausgepackt. Ist das Zielnetz außerhalb des eigenen Bereiches, wird das IP-Paket vom Router „ausgepackt“ und gemäß dem Medium und Schicht-2-Protokoll des Transfernetzes neu „eingepackt“ und wie oben beschrieben ggf. bei IPv4 fragmentiert.



*Verschachtelung Daten und Header im IP-Paket*

## Token Ring/Token Pass

Parallel zum Ethernet-Standard entstanden viele mehr oder weniger verbreitete Protokolle proprietärer Art, die nur von bestimmten Anbietern genutzt wurden. Einen akzeptablen Marktanteil hatte dabei das seit Mitte der 1980er-Jahre von IBM forcierte Token Ring IEEE 802.5 (RFC 2470). Es setzt im Gegensatz zu Ethernet einen Ring als Topologie voraus. Dabei wird im Ring ein Token von Station zu Station geschickt (Token Pass). Die beteiligten Stationen hängen dann Ihre Nutzdaten an das Token bzw. lesen es aus. Die Methode stellt sicher, dass zur selben Zeit immer nur eine Station Daten senden kann; das Netz ist damit kollisionsfrei. Die MTU beträgt bei Token Ring mit 4 MBit/s 4.464 Bytes, bei 16 Mbit/s 17.914 Bytes.

## ATM – Asynchronous Transfer Mode

ATM wurde entwickelt, um in Weitverkehrsnetzen höhere Übertragungsraten erzielen zu können. Dazu werden Datenpakete verschiedener Größe und Herkunft auf kleine Pakete, sogenannte Zellen, aufgeteilt. Die Zellen haben eine Größe von 53 Byte (5 Byte Header, 48 Byte Daten), was auf den ersten Blick nach wenig aussieht. Vorteil der kleinen Zellen ist, dass sie in einem Zeitmultiplexverfahren sehr schnell übertragen werden können. Aus Sicht des Anwenders entsteht so eine virtuelle Leitung. ATM unterscheidet zwischen Virtual Path und Virtual Channel, die im Header definiert werden. Für die richtige Zusammensetzung ist der ATM-Router am Ende der Verbindung zuständig. Der Router stellt dem Anwender eine MTU von 4500 Bytes zur Verfügung.

ATM wird auch bei ADSL verwendet. Dem Teilnehmer wird darüber eine (oder mehrere) transparente Schicht 2-Verbindung(en) als Ethernet zur Verfügung gestellt. Darüber kann dann mit PPPoE eine oder mehrere Verbindung(en) auf Schicht 3 hergestellt werden.

## PPP – Point to Point

PPP ist ein Einwahlprotokoll der Sicherungsschicht (OSI 2). Das bedeutet, dass wie bei DSL eine Schicht 2-Verbindung zur Gegenstelle aufgebaut wird. Die MTU ist dabei von den Verbindungsparametern abhängig, wird aber aus Kompatibilitätsgründen meist auf 1500 Byte eingestellt.

## PPPoE – Point to Point Protocol over Ethernet

Bei PPPoE-Verbindungen, wie sie z. B. über DSL verwendet werden, entsteht ein Protokoll-Overhead von 8 Bytes, sodass dort statt der MTU von 1500 Bytes nur 1492 Bytes zur Verfügung stehen.

## FDDI – Fibre Distributed Data Interface

FDDI (ISO 9314 1-3 und ANSI ASC X3T9.5) wurde 1989 für schnelle und sichere Datenübertragung auf Glasfaserleitungen entwickelt. Die Topologie ist ein Ring oder Doppelring, auch dort wird das beschriebene Token Pass-Verfahren verwendet. FDDI wurde 1994 für die Übertragung auf Kupferkabel (STP/UTP) als CDDI – Copper Distributed Data Interface erweitert. Durch die Entwicklung von Fast Ethernet haben beide Protokolle heute eher historische Bedeutung. Die MTU ist von verschiedenen Parametern abhängig. In RFC 2019 und RFC 1188 wird sie mit 4352 Bytes angegeben (IPv6/IPv4).

## Frame Relay

Frame Relay ist eine Weiterentwicklung des leitungsvermittelten Protokolls X.25 (Datex-P). Es ist ein paketvermittelndes Netz, stellt dem Nutzer aber eine virtuelle transparente Verbindung auf Schicht 2 (OSI) zur Verfügung. Die Paketübertragung erlaubt dem Anbieter die Nutzung eines Mediums durch mehrere Kunden. Neben der garantierten Bandbreite kann der Kunde auch mehr Bandbreite nutzen, sofern sie verfügbar ist. Dabei werden Bandbreiten bis 2 Mbit/s angeboten. Frame Relay eignet sich als preiswerte LAN-zu-LAN-Verbindung. Es wird heute noch angeboten, wird aber mehr und mehr durch ebenso preiswerte VPN-Verbindungen verdrängt. Da es eine transparente Schicht 2-Verbindung zur Verfügung stellt, ist IPv6 problemlos möglich. Die MTU beträgt 4482 Bytes.



Anstelle der betagten Weitverkehrsprotokolle wie ATM, FDDI sowie die hier nicht behandelten SDH/PDH aus der Telekommunikationswelt treten heute vermehrt Netze auf Ethernetbasis wie Metro-Ethernet oder MPLS auf.

## MPLS – Multi Protocol Label Switching

IP-Router waren lange ein Engpass bei der Erhöhung der Geschwindigkeit im Backbone. ATM brachte hier einen deutlichen Fortschritt und war lange Zeit das Maß aller Dinge bei Internetbackbones. In modernen Routern wird das IP-Forwarding von der Hardware erledigt, lediglich das Abarbeiten der Routing-Tabellen erweist sich noch als „Bremse“.

Bei IP-Routing trifft jeder Router aufgrund eines Vergleichs der entsprechenden Paketdaten mit der u. U. sehr ausführlichen Routing-Tabelle erneut eine Entscheidung, wohin ein Paket geschickt wird. MPLS schließt diese Lücke. Dabei wird zwischen Schicht 2 und Schicht 3 ein zusätzliches MPLS-Label (4 Byte) eingesetzt, das von den entsprechenden Routern wesentlich schneller ausgewertet und bearbeitet werden kann als das folgende IP-Paket. Im Label wird neben dem Ziel auch die Route vorher festgelegt, sodass ähnlich wie bei ATM ein virtueller Pfad entsteht. Da das MPLS-Label dem Schicht 3-Datenpaket vorgelagert ist, ist auch IPv6 problemlos möglich. Die Frame Size wird hier an den Routern entsprechend vergrößert, sodass sich für die MTU im LAN nichts ändert.

## MTU bei IPv4 und IPv6

Grundsätzlich gelten die genannten Werte für beide Protokolle, sofern sie nativ übertragen werden. Zu beachten ist, dass, wie bereits beschrieben, IPv6-Pakete nicht mehr fragmentiert werden. Konnte man sich bei IPv4 darauf verlassen, dass der Router die Pakete „zurecht“ fragmentiert, ist das bei IPv6 nicht mehr der Fall.

Ein weiterer Unterschied besteht in der minimalen Länge, die bei IPv4 576 Bytes und bei IPv6 1280 Bytes beträgt.

Wird IPv6 über IPv4 getunnelt (vgl. Kapitel 8), wird die Nutzdatenmenge entsprechend kleiner. Bei getunnelten Protokollen wie z. B. ISATAP (vgl. Kapitel 8) wird meist die kleinstmögliche MTU von 1280 Bytes als Standard gesetzt. Größere MTUs sind experimentell zu ermitteln. In der Konsolenausgabe ist die MTU der LAN-Verbindung mit 1492 Bytes angegeben. Die Verbindung ist über PPoE aufgebaut, siehe oben. Die Loopback Schnittstelle unterliegt nicht den Einschränkungen des Ethernetprotokolls und ist mit 4.294.967.295 Bytes (4GByte) angegeben.

C:\>netsh interface ipv6 show interfaces					
Idx	Met	MTU	State	Name	
4	25	1500	disconnected	WiFi	
1	75	4294967295	connected	Loopback Pseudo-Interface 1	
49	25	1280	connected	isatap.example.net	
12	25	1492	connected	LAN-Verbindung	
26	55	1500	disconnected	LAN-Verbindung 2	

#### Verschiedene MTU-Größen unter Windows

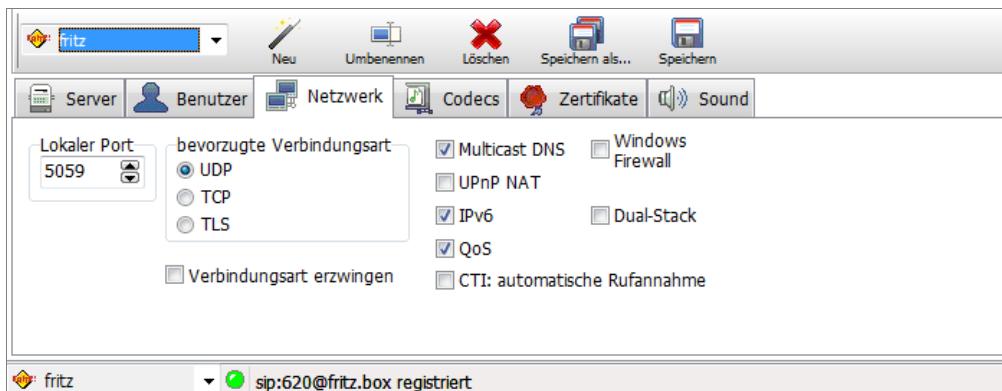
```
root@mail:~# ifconfig
eth0    Link encap:Ethernet HWaddr 00:50:56:00:6f:09
        inet addr:198.51.100.89 Bcast:198.51.100.95 Mask:255.255.255.224
        inet6 addr: 2001:db8:200:5348::89/64 Scope:Global
        inet6 addr: 2001:db8:200:5348::bee/64 Scope:Global
        inet6 addr: fe80::250:56ff:fe00:6f09/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
```

#### MTU unter Linux

### Zusammenfassung

Der kurze Blick auf gängige Protokolle der Sicherungs-/Zugangsschicht hat die Tauglichkeit der in Kapitel 2 vorgestellten Schichtenmodelle erneut bestätigt. Auf der OSI-Schicht 2 können beliebige Schicht 3-Protokolle übertragen werden, solange zwischen beiden ein Bezug hergestellt werden kann. Dieser Bezug wird, wie zuvor gezeigt, mit ARP (IPv4) oder NDP (IPv6) hergestellt.

Bei den Anwendungen ist es nicht ganz so einfach. Anwendungen, die in den höheren Schichten IP-Adressen verarbeiten, müssen für IPv6 angepasst werden. Umgekehrt ist es einfacher. Die meisten IPv6-fähigen Anwendungen sind zugleich IPv4-fähig.



Anwendung (Softphone), bei der zwischen IPv4, IPv6 und Dual-Stack ausgewählt werden kann

# 7 Routing

## In diesem Kapitel erfahren Sie

- ✓ was Routing ist und wie Routing funktioniert
- ✓ welche Netzwerkmodelle und Topologien es gibt
- ✓ welche Arten von statischen Routen unter IPv6 unterschieden werden
- ✓ wie statische Routen für IPv6 unter Windows Server 2008 R2 konfiguriert werden
- ✓ welche dynamischen Routing-Protokolle für IPv6 verwendet werden können
- ✓ welche Eigenschaften RIPng aufweist und wie es sich von RIPv2 unterscheidet
- ✓ welche Eigenschaften OSPFv3 besitzt und wie es sich von OSPFv2 unterscheidet
- ✓ welches externe Routing-Protokoll für das Internet bei IPv6 verwendet wird

## Voraussetzungen

- ✓ Verständnis von IP-Adressen und IP-Netzen

## 7.1 Grundlagen zu Routing

### Was ist Routing?

Routing beschreibt, wie Daten zwischen verschiedenen Endgeräten über eine Netzwerkverbindung ausgetauscht werden können, ohne dass die betreffenden Geräte alle Gegenstellen **direkt** kennen müssen.

### Allgemeine Funktionsweise beim Routing

#### Das logische IP-Netzwerk

IP-basierte Netzwerke werden in logische Topologien strukturiert. Durch die Strukturen der Topologien werden die Netzwerke in logische Segmente (logische IP-Netzwerke) eingeteilt und hierarchisch gegliedert. Diese logischen Segmente werden durch Router miteinander verbunden.

Ein logisches IP-Netzwerk ist immer eine Broadcast-Domain, d. h., Broadcast-Datenverkehr (Datenverkehr an alle Geräte) wird **nur** innerhalb des logischen IP-Netzwerks gesendet und empfangen. Router leiten im Normalfall keinen Broadcast-Traffic weiter. Es ist nur bei der Nutzung spezieller Kommandos möglich – auch Multicast-Datenverkehr (Datenverkehr an eine Gruppe von Geräten) wird standardmäßig **nicht** weitergeleitet.

#### Router verbinden logische IP-Netzwerke

Ein Router ist immer in mindestens zwei logische IP-Netzwerke eingebunden, die er alle direkt erreichen kann. Damit ein Router entfernte logische Netzwerke erreichen kann, benötigt er immer zusätzliche Informationen, sogenannte **Routen**, die ihm Wege in die entfernten Netzwerke aufzeigen.

#### Routen kennen den Weg

Routen enthalten Information, wie entfernte Netzwerke erreicht werden können. Dabei ist diese Information immer: „An welchen direkt benachbarten Router muss das Paket gesendet werden?“ – IP ist ein verbindungsloses Protokoll.

### Statische und dynamische Routen

Router lernen Routen entweder durch **statische Konfiguration** (statische Routen) durch den Administrator oder durch **Kommunikation untereinander** (dynamische Routen). Routen werden in einer Routing-Tabelle verwaltet.

### Routen in Routing-Tabellen verwalten

Die Übergabepunkte der Netzsegmente – die Router – wissen aufgrund von Routen, die innerhalb von Routing-Tabellen verwaltet werden, an welchen direkten Nachbar-Router das Paket weitergeleitet werden muss, um das eigentliche Zielnetzwerk zu erreichen.

Bei der Weiterleitung eines IP-Pakets vergleicht ein Router die Ziel-IP-Adresse im IP-Header des gesendeten Datenpaketes mit den Routen innerhalb seiner Routing-Tabelle. Wird ein übereinstimmender Eintrag gefunden, wird das Paket gemäß der Anweisung innerhalb der Routen an den entsprechenden, direkten Nachbar-Router (Next-Hop) weitergeleitet, der es dann wieder an seinen direkten Nachbarn weiterleitet. Wird kein passender Eintrag gefunden, wird der Frame verworfen.

Es gibt immer einen durch die logische Struktur bestimmten (Übergabe)-Punkt, an den die Datenpakete gesendet werden – den **Next-Hop-Router**.

### Die Rolle des Default Gateway

Dieser Übergabe-Punkt ist für das adressierende Gerät in der Regel sein **Default Gateway**, der Router, der das eigene Netzsegment mit anderen Segmenten verbindet.

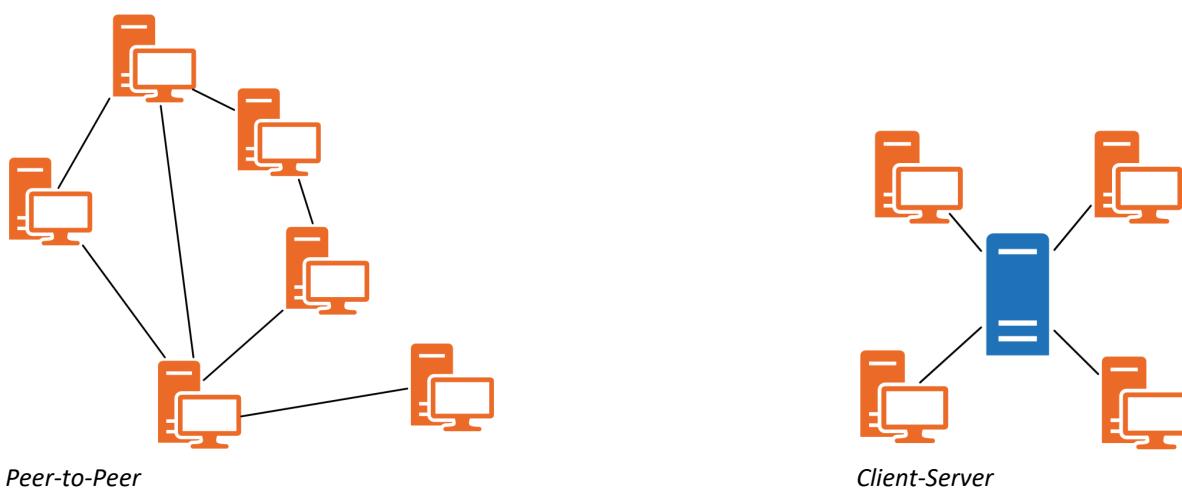
Um ein Paket zwischen zwei Endgeräten über paketorientierte IP-Netzwerke auszutauschen, müssen die Endgeräte die jeweils zu adressierenden Gegenstellen erreichen können, auch wenn sich der Adressat nicht im direkten Zugriff des Adressierenden, also im selben logischen Netzsegment, befindet. Daher ist die Kenntnis des Default Gateway für ein Endgerät notwendige Voraussetzung um ein entferntes Netzwerk zu erreichen.

Wenn ein Router ein Paket aufgrund fehlender Routen nicht weiterleiten kann, verwirft er das Paket und informiert den Sender durch Senden einer ICMP-unreachable-Nachricht.

## 7.2 Netzwerkmodelle und Topologien

### Netzwerkmodelle

Es gibt bei IPv6 genau wie bei IPv4 allgemein zwei Netzwerkmodelle.



- ✓ Zum einen das **Peer-to-Peer**-Netzwerk, in dem alle Maschinen gleichberechtigt sind. Jede Maschine kann also Dienste zur Verfügung stellen, aber auch Dienste in Anspruch nehmen.
- ✓ Zum anderen gibt es das **Client-Server**-Modell. Dort bietet nur der Server Dienste an, die von den Clients genutzt werden können. Vorteil dieses Modells ist, dass die Daten zentral gespeichert und verwaltet werden.

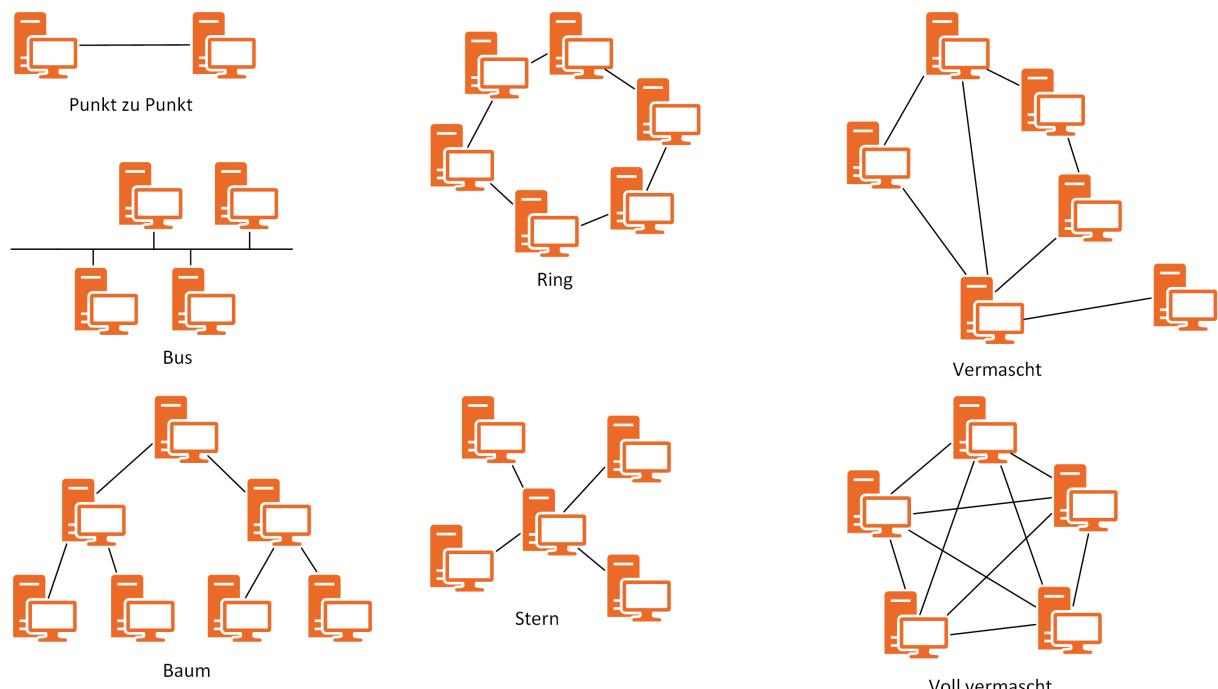
## Topologien

Router können über unterschiedliche Netzwerktopologien miteinander verbunden sein.

Netzwerktopologien sind abhängig von der verwendeten Netzwerkart und der verwendeten Netzwerktechnologie bzw. -Protokoll, d. h., sie sind abhängig von OSI Schicht 1- und 2-Funktionalitäten.

### Übersicht Topologien

- |   |   |
|---|---|
| <ul style="list-style-type: none"> <li>✓ Punkt-zu-Punkt-Topologie</li> <li>✓ Bus-Topologie</li> <li>✓ Ring-Topologie</li> <li>✓ Zell-Topologie</li> </ul> | <ul style="list-style-type: none"> <li>✓ Baum-Topologie</li> <li>✓ Stern-Topologie</li> <li>✓ (Voll-)Vermascht-Topologie</li> </ul> |
|---|---|



### Topologiemodelle

#### Punkt-zu-Punkt-Topologie

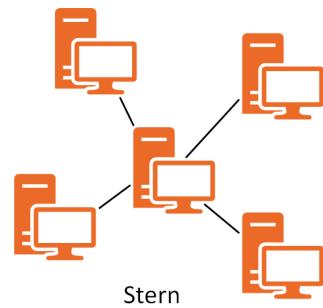
Die Punkt-zu-Punkt-Topologie ist die Grundlage jeder anderen Topologie. Jede andere Topologie besteht aus einer Variation oder aus aneinandergeketteten Punkt-zu-Punkt-Verbindungen, die durch ein Routing-Protokoll oder statisches Routing eine andere, neue Topologie bilden. Eine Punkt-zu-Punkt-Topologie hat einen sehr einfachen Aufbau. Es werden einfach zwei Knoten direkt miteinander verbunden. Bei Fibre-Channel-Netzwerken (Glasfaser) wird noch immer eine direkte Punkt-zu-Punkt-Verbindung verwendet, um eine optimale Bandbreite zu erreichen. Eine Vollvermaschung besteht im Grunde nur aus vielen Punkt-zu-Punkt-Verbindungen.



Vorteile	Nachteile
<ul style="list-style-type: none"> <li>✓ Höchste Übertragungsrate</li> <li>✓ Störungssicher</li> <li>✓ Abhörsicher</li> <li>✓ Vorhersehbare, nutzbare Übertragungskapazität</li> <li>✓ Leicht erweiterbar</li> <li>✓ Leichte Fehlersuche</li> <li>✓ Kein Routing benötigt</li> </ul>	<ul style="list-style-type: none"> <li>✓ Jeder Nutzer muss auf jedem Rechner eingetragen sein</li> <li>✓ Keine zentrale Verwaltung möglich</li> <li>✓ Freigaben auf Benutzerebene sind nicht möglich</li> </ul>

### Stern-Topologie

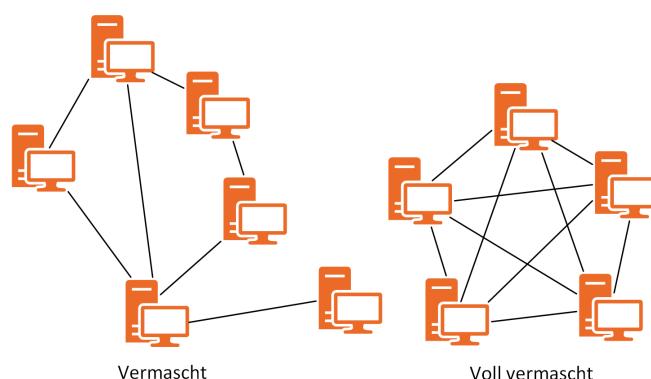
Bei der Stern-Topologie werden alle Teilnehmer an einen zentralen Knoten mit einer Punkt-zu-Punkt-Verbindung angebunden. Dadurch entsteht eine erhöhte Ausfallwahrscheinlichkeit für einzelne Knoten, jedoch hat der Ausfall einer Maschine keine Auswirkungen auf den Rest des Netzes. Fällt jedoch die zentrale Einheit aus, so fällt das gesamte Netz aus. Dem kann jedoch durch Redundanz entgegengewirkt werden.



Vorteile	Nachteile
<ul style="list-style-type: none"> <li>✓ Endgeräte können ausfallen; Netz funktioniert weiterhin</li> <li>✓ Hohe Übertragungsraten, falls ein Switch eingesetzt wird</li> <li>✓ Leicht erweiterbar und zu verstehen</li> <li>✓ Leichte Wartbarkeit und Fehlersuche</li> <li>✓ Kein Routing benötigt</li> <li>✓ Sehr gut für Multicast- und Broadcast-Anwendungen</li> <li>✓ Telefon- und Rechnerkombinationen möglich</li> </ul>	<ul style="list-style-type: none"> <li>✓ Fällt der Verteiler aus, fällt das gesamte (Teil-) Netz aus</li> <li>✓ Wird ein Hub benutzt, sind die Übertragungsraten sehr niedrig</li> <li>✓ Relativ hoher Kabelaufwand</li> </ul>

### Maschen-Topologie

Bei einem vermaschten Netz ist jede Maschine mit einer oder mehreren anderen Maschinen im Netz verbunden. Fällt eine Verbindung aus, so ist es in der Regel möglich, durch entsprechendes Routing weiterhin mit dem Partner zu kommunizieren. Ist jede Maschine mit jeder anderen Maschine in einem Netz direkt verbunden, spricht man von einem vollvermaschten Netz.



Vorteile	Nachteile
<ul style="list-style-type: none"> <li>✓ Sicherste Variante eines Netzwerkes</li> <li>✓ Bei Ausfall eines Endgerätes ist durch Routing weiterhin eine Kommunikation möglich</li> <li>✓ Sehr leistungsfähig dank vieler Verbindungen</li> <li>✓ Vollvermaschte Netze benötigen kein Routing; nur Direktverbindungen vorhanden</li> </ul>	<ul style="list-style-type: none"> <li>✓ Sehr hoher Aufwendungsgrad auch bei nicht vollständig vermaschten Netzen</li> <li>✓ Sehr viele Kabel und Karten benötigt</li> <li>✓ Sehr hoher Energieverbrauch</li> <li>✓ Komplexes Routing bei nicht vollvermaschten Netzen nötig</li> </ul>

## 7.3 Autonome Systeme

Eine Sammlung von IP-Netzen, welche gemeinsam verwaltet werden, werden auch **Autonome Systeme (AS)** genannt. Meist werden Autonome Systeme von einem Provider (z. B. Deutsche Telekom, Vodafone), einem internationalen Unternehmen (z. B. Google, Microsoft) oder einer Universität verwaltet. Autonome Systeme können aus vielen Teilnetzen bestehen und sind über ein internes Routing-Protokoll miteinander verbunden. Viele, untereinander verbundene Autonome Systeme bilden das Internet.

### Verwaltung

Zur Verwaltung wird jedem Autonomen System durch die **IANA (Internet Assigned Numbers Authority)** eine Nummer zugewiesen. Die IANA delegiert die Verteilung dieser **ASN (Autonomous System Number)** jedoch weiter an die **RIR (Regional Internet Registries)**. Dies sind ARIN, RIPE NCC, APNIC, LACNIC und AfriNIC (vgl. Kapitel 4).

ASN sind Routing-Informationen für das externe Routing-Protokoll **BGP**. Eine ASN wird erst vergeben, wenn das neue Autonome System mit mindestens **zwei** anderen Autonomen Systemen über ein Routing-Protokoll kommuniziert. Ist dies nicht der Fall, kann auch eine private ASN, statisches Routing oder eine andere Lösung verwendet werden. Private ASN müssen nicht registriert werden, können aber nur für interne Zwecke verwendet werden.

Durch die Aufteilung in Autonome Systeme wird eine bessere Skalierbarkeit des Internets erreicht. Die verbesserte Skalierbarkeit basiert darauf, dass weniger Speicherplatz sowie Bandbreite benötigt werden, um die Routing-Informationen zu übertragen und zu speichern. Diese Routing-Informationen beinhalten bei externen Routing-Protokollen (Informationen werden im Internet ausschließlich über das Protokoll BGP übertragen; vgl. folgende Erläuterungen), die zwischen autonomen Systemen routen, die ASN.

### Routing

Für das Routing innerhalb eines Autonomen Systems ist der Betreiber des Systems verantwortlich. Es ist dem Betreiber prinzipiell freigestellt, ob er statisches Routing betreibt oder ein dynamisches Routing-Protokoll verwendet.

Für das Routing zwischen den Autonomen Systemen gibt es einheitliche Standards. Routing-Protokolle zwischen zwei Autonomen Systemen werden auch **EGP (Exterior Gateway-Protokolle)** genannt. Das einzige weltweit eingesetzte EGP-Protokoll ist das **BGP (Border Gateway Protocol)**.

Routing-Protokolle für das Routing innerhalb eines Autonomen Systems heißen **IGP (Interior Gateway-Protokolle)**. Beispiele dafür sind das **RIP (Routing Information Protocol)** oder das **OSPF (Open Shortest Path First Protocol)**.

## Kunden, Peers, Provider

Beim Routing zwischen mehreren Autonomen Systemen (**Inter-AS-Routing**) unterscheidet man zwischen **Kunden**, **Peers** und **Providern**.

Tier-1 Provider	Internetprovider, die nur Kunden und Peers haben, aber nirgends Kunde sind
Provider	Erhält Geld dafür, dass er dem Kunden einen Zugang zum Internet zur Verfügung stellt
Peers	Zwei in etwa gleich große, autonome Systeme, deren Betreiber sich darauf geeinigt haben, die Kosten für eine Leitung untereinander aufzuteilen. Dadurch gibt es weder Kunden noch Provider.
Kunde	Zahlt dafür, über ein Autonomes System Daten zu beziehen und mit dem Internet Daten austauschen zu können

## Stub-AS, Multihoming, Transit-AS

Man unterscheidet Autonome Systeme danach, ob diese in ihrem übergeordneten Netz einen End- oder Zwischenknoten bilden.

- ✓ **Stub-AS:** Sind über genau **einen** Link an einen **Provider** angeschlossen. Dürfen theoretisch nicht existieren, da festgeschrieben ist, dass ein Autonomes System an mindestens zwei Provider angeschlossen sein muss.
- ✓ **Multihomed Stub-AS:** Genau wie Stub-AS eigentlich nicht erlaubt, da sie nur an **einen Provider** angeschlossen sind. Sie sind jedoch aufgrund der Ausfallsicherheit mit **mehr als einem** Link an einen Provider angeschlossen.
- ✓ **Multihomed AS:** Ein Autonomes System, das an mehr als einen Provider angeschlossen ist.
- ✓ **Transit-AS:** Sind immer mit anderen Transit-AS verbunden und bilden die Serviceprovider für alle anderen Autonomen Systeme. Die Transit-AS sind das Backbone des Internet. Ein Transit-AS ist immer ein Provider für mindestens ein anderes Autonomes System.

## Policybasiertes Interdomain-Routing

Im Allgemeinen lassen sich die Regeln für das Weiterleiten der Routing-Informationen zwischen Autonomen Systemen in drei Regeln zusammen. Diese Regeln basieren auf rein wirtschaftlichen Aspekten, nicht auf technisch-optimalen. Dies ist jedoch notwendig, da es sonst bei den Anbietern zu massiven, finanziellen Verlusten kommen kann. Aus der Sicht eines Providers:

- ✓ Ist ein Autonomes System mein Kunde, teile ich diesem alle Routen mit, die ich kenne. Dadurch wird so viel Verkehr wie möglich über mich abgewickelt. Da hier meist in Volumen abgerechnet wird, wird so am meisten verdient.
- ✓ Ist ein Autonomes System mein Provider, teile ich diesem die Routen zu meinen Kunden mit. Damit sind diese erreichbar, und ich kann an diesen Geld verdienen. Routen zu meinen anderen Providern oder Peers teile ich jedoch nicht mit, da es ansonsten passieren könnte, dass ich im schlimmsten Fall den Traffic bei beiden Providern zahlen muss.
- ✓ Mit meinen Peers teile ich auch nur die Routen zu meinen Kunden. Damit kann der Kunde auch die Peers erreichen, und ich muss nicht noch extra Traffic zahlen, der über den Provider läuft. Der Peer darf auch keine Routen zu meinem Provider erhalten, da er ansonsten auf meine Kosten Daten mit diesem austauschen kann. Routen zu anderen Peers teile ich normalerweise auch nicht, da ein Peer sonst unnötig mein Netzwerk belastet, ich jedoch nichts daran verdiene.

## Beispiele für AS-Nummern

Normalerweise haben ISPs eigene AS-Nummern. Jedoch haben auch große internationale Unternehmen oder Universitäten eigene AS-Nummern. Einige Beispiele:

Unternehmen	AS-Nummer
BelWü	AS553
DFN	AS680
Vodafone	AS3209
freenet.de	AS5430
Wikimedia Foundation	AS14907
Universität Frankfurt	AS20633

## 7.4 Routing-Algorithmen

### Funktionsweise des Routing

Grundsätzlich basiert das Routing darauf, eine Tabelle zu erstellen, die die Informationen darüber enthält, wohin ein Paket geleitet werden soll.

Empfängt der Router ein Paket, so entfernt er den Header dieses Pakets und liest die für ihn wichtigen Informationen aus. Die wichtigste Information ist die Ziel-Adresse (**Destination IP**). Hat er die Adresse ausgelesen, vergleicht er diese mit den Informationen in seiner Routing-Tabelle. Anhand der Routing-Tabelle weiß der Router, über welche Schnittstelle (**Next Hop**) er das Paket weiterleiten muss, damit es beim Empfänger ankommt. Wurde der Next-Hop-Router ermittelt, baut der Router ein neues Paket mit einem angepassten Header und sendet das Paket über die ermittelte Schnittstelle nach draußen.

Um seine Routing-Tabelle aufzubauen, kann sich ein Router verschiedener Methoden bedienen. Jedoch unterstützt nicht jeder Router auch jede Methode.

- ✓ Netzwerke, die direkt an einen Router angebunden sind, werden automatisch vom Router eingefügt.
- ✓ statische Routen, die von einem Administrator eingepflegt werden (sehr hoher administrativer Aufwand)
- ✓ dynamisches Erlernen von Routen durch Nutzung von Routing-Protokollen

### Übersicht der Routing-Protokolle

Im Allgemeinen wird zwischen zwei Arten von Routing-Protokollen unterschieden.

Zum einen gibt es die **internen** Routing-Protokolle für Routen innerhalb eines Autonomen Systems (**Interior Gateway Protocols**):

- ✓ **IGRP/EIGRP** (Interior Gateway Routing Protocol/Enhanced IGRP)
- ✓ **OSPF** (Open Shortest Path First)
- ✓ **IS-IS** (Intermediate System to Intermediate System)
- ✓ **RIP** (Routing Information Protocol)

Dann gibt es noch die Routing-Protokolle zum Routing zwischen mehreren Autonomen Systemen (**Exterior Gateway Protocols**):

- ✓ **BGP** (Border Gateway Protocol): (seit 2002 in der Version BGP4) ist heute weltweit der De-facto-Standard
- ✓ **EGP**: Mit dem Exterior Gateway Protocol wurden früher die Internet-Backbones verbunden. Es ist inzwischen veraltet.

## 7.5 Protokoll-Algorithmen

Alle Routing-Protokolle verwalten die Routing-Tabelle dynamisch durch Austausch von Routing-Informationen zwischen den Routern. Jedes Routing-Protokoll benutzt dazu eine von drei Verfahrensweisen (Algorithmen):

<b>Link State Routing</b>	Beim Link State Routing teilt ein Router jedem erreichbaren anderen Router alle Netze mit, die er selber kennt. Dadurch kennt nach einiger Zeit jeder Router das gesamte Netz und kann sich selbst den optimalsten Weg suchen. OSPF nutzt diese Technik.
<b>Distance Vector Routing</b>	Bei Distanzvektorprotokollen teilt ein Router jedem erreichbaren Router mit, wie für ihn das Netz aussieht. Dadurch verwendet ein Router immer die für ihn optimale Lösung. Das Problem mit der kürzesten Route wird somit auf mehrere Router verteilt. RIP nutzt diese Technik.
<b>Path Vector Routing</b>	Beim Pfad-Vektor-Routing wird ähnlich gearbeitet wie bei Distanzvektorprotokollen. Jedoch gibt es hier noch einige Optimierungen, um eine Schleifenbildung zu verhindern. BGP nutzt diese Technik.

### Link State Routing

Ein Link ist die physikalische Verbindung (Schnittstelle) eines Routers in ein Netzwerk.

Ein Link State Routing Protocol ist ein Protokoll, das Informationen über Links (Verbindungen) versendet.

Diese Link-Informationen sind immer lokale Informationen, d. h., sie beziehen sich für jeden Router nur auf dessen lokale Verbindungen, wie z. B.:

- ✓ Status des Links (up/down)
- ✓ Logische Adresse (IP-Adresse, Net-ID) auf dem Link
- ✓ Bandbreite auf dem Link (die Kosten, die als Metrik verwendet werden)
- ✓ Art des Link (z. B. Punkt-zu-Punkt-Verbindung, Ethernet-Multiaccess-Verbindung etc.)
- ✓ Benachbarte Link-State-Router über den Link

### Arbeitsweise

- ✓ Link-Informationen werden unverändert an **alle** anderen Link-State-Router im Netzwerk gesendet, sodass alle Router im Netzwerk über die gleichen Informationen des Netzwerks (die Link-Informationen aller Router) verfügen.
- ✓ Alle empfangenen Link-Informationen **aller** Router werden vom Router in einer Tabelle oder Datenbank (OSPF: LSDB - Link State Database) verwaltet.

Die gesammelten Link-Informationen innerhalb der Datenbank bilden die gesamte Topologie des Netzwerks ab.

Jeder Link-State-Router kennt die gesamte Topologie des Netzwerks („kennt den Netzplan“) – Link-State-Protokolle sind daher schleifenfrei.

- ✓ Aus diesen topologischen Informationen errechnet jeder Router mithilfe eines mathematischen Algorithmus die besten Pfade zu den Zielnetzwerken, wobei meist die Bandbreite auf den Verbindungen als Metrik verwendet wird (die Metrik von Link-State-Protokollen wird als **Kosten** bezeichnet und ist i. d. R. einstellbar). OSPF verwendet dazu den Dijkstra-Algorithmus, auch SPF – Shortest Path First-Algorithmus genannt.
- ✓ Die errechneten besten Pfade werden dann als Routen in die Routing-Tabelle geschrieben.

- ✓ Routing-Informationen (OSPF: LSAs – Link State Advertisements) werden **sofort** und **nur** bei Änderungen des Netzwerks an **alle** anderen Router gesendet bzw. verteilt, die daraufhin die Informationen innerhalb ihrer Topologie-Datenbank anpassen und die Routing-Tabelle neu berechnen.

Dieser Informationsaustausch findet sehr schnell statt, d. h., die Zeit, die benötigt wird, um alle Router auf den aktuellen Stand des Netzwerks zu bringen – die Konvergenzzeit – ist sehr gering. Sobald alle Router über alle aktuellen Netzwerkinformationen verfügen, spricht man von **Konvergenz** des Netzwerks.

Die Routing-Informationen enthalten **nur** die Änderungen, d. h., Link-State-Routing-Protokolle benötigen weniger Bandbreite als ein Distance-Vector-Protokoll, das immer alle Routen in Updates sendet.

- ✓ Direkt benachbarte Router tauschen in kurzen, zeitlichen Intervallen (OSPF: „Hello Interval Time“) kleine „Hello“-Pakete untereinander aus.

Damit wird die Erreichbarkeit eines benachbarten Routers geprüft. Werden mehrere „Hellos“ von einem bis dahin bekannten Nachbarn nicht mehr empfangen, wird der Nachbar für tot erklärt (OSPF: „Dead Interval Timer“) und entsprechende Informationen über den Ausfall des Nachbarn als Routing-Informationen in das Netzwerk gesendet.

### Überblick Eigenschaften

- ✓ Alle Link-State-Router kennen die gesamte Topologie des Netzwerks (Netzplan).
- ✓ Der verwendete mathematische Algorithmus berechnet aus dem Netzplan schleifenfreie Pfade, wobei i. d. R. der Weg mit der höchsten Bandbreite als Route in die Routing-Tabelle übernommen wird.
- ✓ Schnelle Konvergenz: Routing-Informationen (LSAs – Link State Advertisements) werden **sofort** (und nur dann) gesendet, wenn sich Änderungen ergeben haben.
- ✓ Da die Updates nur die Informationen zu den Änderungen enthalten, belasten sie die Bandbreite nur gering. Aufgrund der verwendeten Metrik (Costs, steht i. d. R. für die Bandbreite) ist ein Link State Protocol für große Netzwerke geeignet. Zudem können große Netzwerke in unterschiedliche Routing-Bereiche (Areas bei OSPF) eingeteilt werden.
- ✓ Link State Protocols bieten viele zusätzliche Features, die dem Administrator eine Reihe von Möglichkeiten bieten, das Routing zu optimieren – dies setzt jedoch ein detailliertes Wissen um die komplexe Funktionsweise voraus. Daher gelten Link State Protocols als schwierig zu konfigurieren und schwierig zu überwachen.

### Distance Vector Routing

Bei einem Distance-Vector-Routing-Protokoll werden Wege in Zielnetzwerke, die Routen, als Routing-Informationen ausgetauscht.

Diese Routing-Informationen, die Routing-Updates, werden

- ✓ zu periodischen Zeiten (Zeitintervallen)
- ✓ **nur** zwischen direkt benachbarten Routern

versendet und empfangen.

Routing-Updates enthalten zudem weitere Informationen zu den übermittelten Routen:

- ✓ **die Distanz** (Entfernung bis zum Zielnetzwerk): hier der Hop Count – hierbei wird die Anzahl der Router, die zwischen dem weiterleitenden Router und dem Zielnetzwerk liegen, gezählt; verwendet wird der Weg, der die wenigsten Router passiert (Router = Hop; Anzahl der Router = Hop Count)
- ✓ **den Vektor**, der die Zielrichtung beschreibt („Wohin sollen Daten gesendet werden, wenn das Zielnetzwerk erreicht werden soll?“). Als Vektor wird die IP-Adresse des direkten Nachbar-Routers (Next Hop) verwendet, von dem die Informationen erhalten wurden.

## Arbeitsweise

- ✓ Zu Beginn kennt jeder Router nur seine direkt verbundenen Netzwerke, die er in Routing-Updates an direkte Nachbarn sendet.
- ✓ Gleichzeitig empfängt er Routing-Updates seiner Nachbarn, die Routen zu Zielnetzwerken enthalten, die seinen Nachbarn bekannt sind.
- ✓ Neu erlernte Routen werden in die nächsten Routing-Updates eingefügt, und diese wieder an andere direkte Nachbarn gesendet. Der Distance-Vector-Router sendet alle ihm bekannten Routen (auch die von anderen Nachbarn erlernten) in regelmäßigen Abständen wiederum an seine anderen Nachbarn weiter.
- ✓ Nach dieser Logik, dem Bellman-Ford-Algorithmus, lernen alle Router im Netzwerk nach und nach alle Routen zu entfernten Netzwerken („Schneeballsystem“).

Grundsätzlich werden empfangene Routing-Updates überprüft, und es werden folgende Einträge in die Routing-Tabelle übernommen:

- ✓ unbekannte Routen
- ✓ bessere Routen: kleinere Metrik – geringerer Hop Count – für bekannte Routen, unabhängig von welchem Nachbarn die Updates erhalten wurden
- ✓ neue Informationen zu bekannten Routen: neue Metrik – höherer oder geringerer Hop Count – für bekannte Routen, wenn die Updates vom gleichen Nachbarn erhalten wurden

## Problematik und Lösungen

Die Verwendung von periodischen Updates und die Eigenschaft, dass auch Routen mit höherer Metrik akzeptiert werden, falls die Information vom gleichen Nachbarn empfangen wurde, ist problematisch.

Durch diese Arbeitsweise ist die „rechtzeitige“ Verteilung neuer Informationen und somit die Integrität der Informationen nicht immer gewährleistet.

Die Verteilung von fehlerhaften (nicht mehr aktuellen) Informationen kann Routing Loops verursachen, indem Router sich die Pakete gegenseitig zusenden, ohne dass das Paket das Zielnetzwerk erreicht:

- ✓ Fehlerhafte Informationen führen zu fehlerhaften Einträgen in der Routing-Tabelle.
- ✓ Fehlerhafte Einträge in der Routing-Tabelle bewirken eine falsche Weiterleitung der Pakete.

Um Routing Loops zu vermeiden und Abgleichungen im Netzwerk (Konvergenzen) zu beschleunigen, sind Distance-Vector-Protokolle um einige zusätzliche Funktionen (RIPv2, Cisco-Protokolle) erweitert worden, dazu zählen:

### ✓ Maximum Hop Count

Eine maximale Metrik von 15 (Metrik 16 = Zielnetzwerk nicht erreichbar). Verhindert den Count-to-Infinity-Effekt. Dabei senden sich Router gegenseitig Routing-Informationen über ein bereits ausgestorbenes Netzwerk zu, wobei die Metrik kontinuierlich erhöht wird.

### ✓ Split Horizon

Routing-Updates, die über eine bestimmte Schnittstelle empfangen wurden, dürfen über die gleiche Schnittstelle nicht wieder propagiert werden.

### ✓ Route Poisoning

Routen in ausgestorbenen Netzwerken werden für eine gewisse Zeitspanne mit einer Metrik von 16 propagiert („vergiftet“), anstatt die Route in zukünftigen Updates einfach nicht mehr zu propagieren.

### ✓ Poison Reverse

Empfängt ein Router eine „vergiftete“ Route, bestätigt er den Erhalt, indem er die Route an den gleichen Nachbarn zurücksendet, dabei wird temporär die Split-Horizon-Regel außer Kraft gesetzt.

### ✓ Triggered Updates

Wenn sich Änderungen im Netzwerk ergeben, dürfen diese sofort propagiert werden, d. h., Router müssen nicht mehr auf den Ablauf des entsprechenden Zeitintervalls (Update Timer) warten, sondern senden die Informationen sofort.

## Überblick Eigenschaften

- ✓ Router besitzen nur aus Sicht der Nachbar-Router Informationen über das Netzwerk.
- ✓ Langsame Konvergenz (Abgleichung der Routing-Tabelle) bei Änderungen in der Netztopologie durch periodische Updates
- ✓ Nicht schleifenfrei trotz zusätzlicher Funktionen
- ✓ In größeren Netzwerken relativ hohe Belastung der Bandbreite, aufgrund umfangreicher Updates (viele Routen in Updates)
- ✓ In Netzwerken mit unterschiedlichen Bandbreiten auf Verbindungen ist die Metrik (Hop Count) unzureichend, da nur die Anzahl der Router bis zum Zielnetzwerk als Entscheidungskriterium für die Auswahl der „besten“ Route verwendet wird.
- ✓ Es gibt eine maximale Metrik: Hop Count = 15, d. h., ein Zielnetzwerk darf nicht weiter als 15 Hops entfernt sein. Deshalb ist Distance Vector Routing nicht für große Netzwerke geeignet.
- ✓ Einfache Arbeitsweise, einfache Konfiguration und einfaches Troubleshooting

## Path Vector Routing

Path-Vector-Protokolle gehören zu den Distance-Vector-Protokollen. Statt einen einzelnen Kostenwert (wie z. B. Hop Count) weiterzugeben, wird der gesamte Pfad (nähere Informationen zu den betroffenen Routern) weitergegeben. Dadurch kann die Count-to-Infinity-Problematik verhindert werden.

## Arbeitsweise

Jeder Router sendet periodisch alle Pfade, die er erreichen kann und über welche er erreichbar ist, an alle Nachbarn. Die Router können dann mit diesen Informationen ihre Routing-Tabelle aufbauen. Die gesendeten Pfade werden vom Router darauf überprüft, ob er selbst, also seine Routerkennung, in dem Pfad enthalten ist. Ist er selbst nicht enthalten, speichert er den Pfad in seiner Routing-Tabelle. Dadurch erhält jeder Router Kenntnis über das gesamte Netz und kennt alle Wege, die er benötigt. Da die Tabelle so schnell sehr groß wird, wird bei vielen Protokollen, z. B. dem BGP (Border Gateway Protocol), nur der kürzeste Pfad zu einem Zielnetzwerk weitergegeben. Die entstehende Routing-Tabelle wird dadurch auf die bestmöglichen Pfade reduziert.

Zusätzlich dazu, dass Änderungen unmittelbar weitergegeben werden, werden alle Routing-Einträge in bestimmten Zyklen an alle relevanten benachbarten Router gesendet, um sicherzustellen, dass im Netz jederzeit größtmögliche Konsistenz herrscht.

## 7.6 Statisches und dynamisches Routing

### Statisches und dynamisches Routing im Vergleich

Statische Routen sind vom Administrator fest eingestellte Pfade zu Zielnetzwerken. Die administrative Konfiguration von statischen Routen (**Weiterleitungsregeln**) ist – neben der Verwendung von dynamischen Routing-Protokollen – eine etablierte Methode, die gewährleistet, dass Router Pakete in entfernte Netzwerke senden können.

Wenn sich allerdings Änderungen in der Topologie des Netzwerks ergeben, muss die Routing-Tabelle hinsichtlich der statisch erzeugten Einträge auf allen Routern im gesamten Netzwerk administrativ (manuell) angepasst werden. Daher hat die Verwendung dynamischer Routing-Protokolle gegenüber dem statischen Routing immer einen entscheidenden Vorteil: Die Routing-Tabelle auf allen Routern im Netzwerk ist immer, auch bei Änderungen der Topologie des Netzwerks, auf dem aktuellsten Stand.

### Gründe für statisches Routing anstelle dynamischen Routings

- ✓ **Administrative Kontrolle über die Verbindungen**

Durch die statische Konfiguration der Weiterleitung sind die Wege der Pakete immer vorhersehbar. Da sich bei statischem Routing der Inhalt der Routing-Tabellen nicht verändert (außer bei Ausfall einer Verbindung), ist auch eine Fehleranalyse (Troubleshooting) einfacher.

- ✓ **Bestehende Sicherheitsrichtlinien, die dynamische Protokolle verbieten**

Wenn Protokolle Daten austauschen, bietet der Austausch immer eine Möglichkeit für Angriffe. Das gilt auch für Routing-Protokolle, selbst wenn die Kommunikation der Protokolle i. d. R. durch kryptografische Methoden geschützt wird. Bei Verwendung von statischen Routen werden keine Routing-Informationen über das Netzwerk übertragen, die ausgespäht werden könnten.

- ✓ **Kleine Netzwerke**

Routing-Protokolle senden und empfangen Informationen über das Netzwerk. Sie benötigen abhängig vom verwendeten Protokoll eine unterschiedliche Bandbreite. Deshalb ist der Einsatz von Routing-Protokollen in kleineren Netzwerken, mit meist nur sporadischen, geringfügigen Änderungen innerhalb der Topologie, nicht empfehlenswert.

- ✓ **Ersatzrouten**

Statische Routen können auch als Ersatzrouten (backup route) verwendet werden. Falls die eigentliche Verbindung, die i. d. R. über ein Routing-Protokoll gelernt wurde, ausfällt, wird die statische Route verwendet.

- ✓ **Default Route**

Eine Default Route ist eine Route in das Netzwerk ::/0 für IPv6 bzw. 0.0.0.0/0 für IPv4, die meist für Verbindungen in das Internet benötigt werden. Sie werden auf Edge-Routern (Router, die an der Grenze des eigenen Netzwerks stehen) meist statisch konfiguriert und zeigen auf den **ISP (Internet Service Provider)**.

Eine Default Route kann über dynamische Routing-Protokolle im Netzwerk weiterverteilt werden (**Redistribution**). 

### Unterscheidung statischer Routen

Statische Routen unter IPv6 werden nach Art ihrer Konfiguration bzw. nach ihrem Informationsgehalt unterschieden. Eine statische Route muss immer das Präfix und die Präfix-Länge enthalten, kann aber (je nach Router) unterschiedlich angelegt werden:

<b>Directly attached (direkt verbunden)</b>	Angabe der ausgehenden Schnittstelle
<b>Recursive (rekursiv)</b>	Angabe der Next-Hop-Adresse
<b>Fully specified (voll spezifiziert) – empfohlen</b>	Angabe der ausgehenden Schnittstelle und der Next-Hop-Adresse

Die Angabe einer Fully-specified-IPv6-Route ermöglicht das effizienteste Routing und ist für bestimmte Verbindungsarten, wie BMAs (Broadcast Multiaccess Networks, z. B. Ethernet-Netzwerke), notwendig.

Bei Verwendung der link-lokalen IPv6-Adresse des Nachbarn als Next Hop (empfohlen) muss die ausgehende Schnittstelle zwingend mit angegeben werden, da der Router diese nicht selbst ermitteln kann.



**RFC 2461** empfiehlt, falls möglich, die link-lokale Adresse des Next-Hop-Routers als Next-Hop-Adresse innerhalb einer voll spezifizierten statischen Route zu verwenden (RFC 2461).

Bei dieser Vorgehensweise ist, eine aktive Verbindung vorausgesetzt, die korrekte Arbeitsweise der Redirect-Funktion und die Erreichbarkeit des Next-Hop-Router, z. B. auch im Falle eines Renumberings (Umadressierung von Netzwerken), sichergestellt, da sich die link-lokale IPv6-Adresse nicht ändert.



Dynamische Routing-Protokolle sind nach ihren Standards ebenfalls verpflichtet, ausschließlich die link-lokale IPv6-Adresse als Next-Hop-IPv6-Adresse innerhalb der Routing-Tabelle zu verwenden.

Ausnahmen dieser Empfehlung sind z. B.

- ✓ Routen über 6to4- oder Teredo-Tunnel, die voll spezifizierte Routen über eine **Aggregatable-Global-IPv6-Adresse** erfordern, oder
- ✓ Routen über Punkt-zu-Punkt-WAN-Verbindungen, die als direkt verbunden konfiguriert werden können, falls nicht anders möglich.

## 7.7 Konfiguration statischer Routen – MS Server Manager

### Beispiel

Am Beispiel eines MS Windows Server 2016, der als Router eingesetzt werden soll, wird gezeigt, wie eine statische Route eingerichtet wird.

#### Routing- und RAS-Dienst installieren und einrichten

Zunächst muss auf dem Server der Routing- und RAS-Dienst installiert und eingerichtet werden. Diese Dienste sind für die Routing-Funktion zuständig und standardmäßig nicht aktiviert.

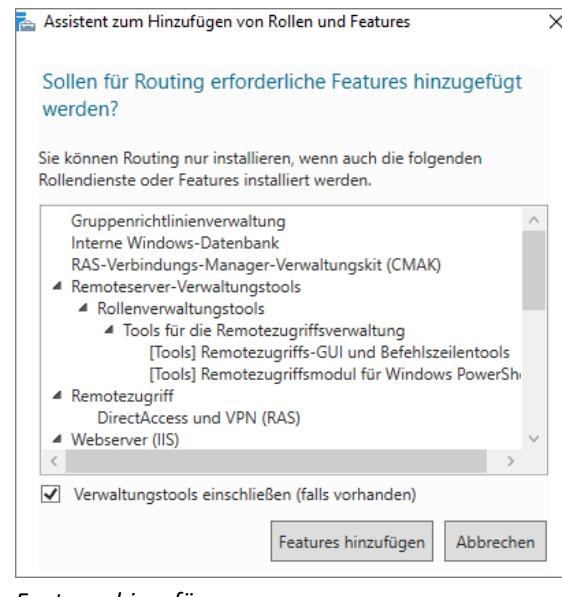
- Starten Sie den Server-Manager.
- Klicken Sie in der Startseite des Server-Managers auf *Rollen und Features hinzufügen*. Klicken Sie jeweils auf *Weiter* bis zum Fenster *Serverrollen auswählen*.
- Wählen Sie aus der Liste *Remotezugriff* aus.

Nach dem Auswählen erscheinen im linken Bereich kontextbezogene Menüpunkte. Klicken Sie in den folgenden Fenstern auf *Weiter* bis Sie zum Fenster *Rollendienste auswählen* kommen.

Hier wählen Sie *Routing* aus. Es werden zusätzliche Features angezeigt, deren Vorauswahl Sie mit *Features hinzufügen* bestätigen.

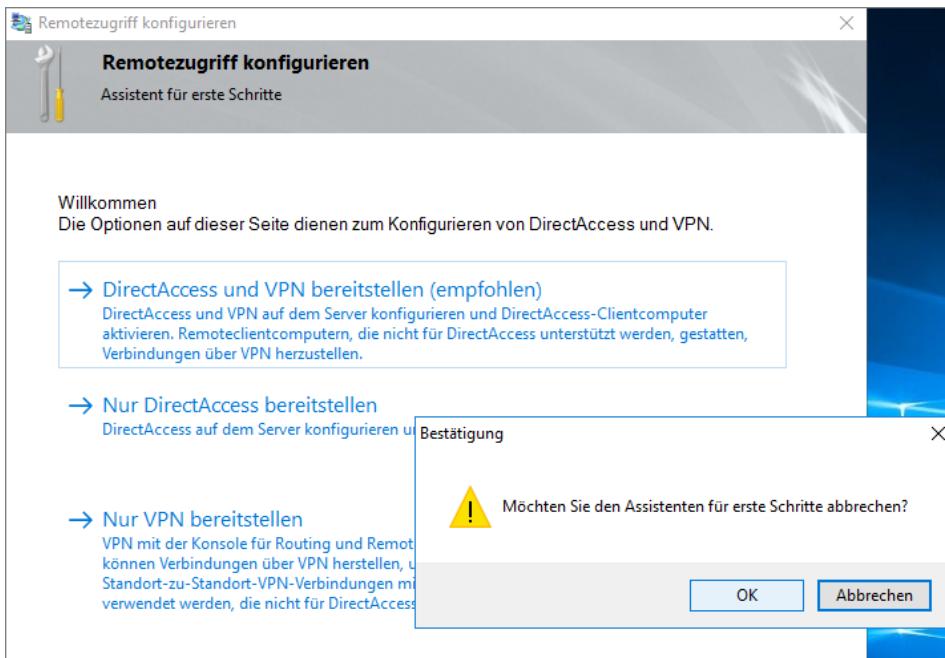


Anschließend ist zusätzlich *Direct Access* ausgewählt. Das kann nicht abgewählt werden und muss mit installiert werden, auch wenn es hier nicht gebraucht wird. (Direct Access ist eine VPN-Technologie von Microsoft, die verschiedene Methoden und Protokolle kombiniert, u. a. auch IPv6 Übergangstechniken.)



Bestätigen Sie die weiteren Fenster mit *Weiter* bzw. *Installieren*.

Nach der Installation wird *Konfiguration nach Bereitstellung* angefordert. Das bezieht sich auf das hier nicht verwendete Direct Access. Damit die Meldung verschwindet, muss der *Assistent für erste Schritte* einmal gestartet und über das Schließen-Symbol in der rechten oberen Ecke beendet werden.



*Assistent für erste Schritte einmal starten und wieder schließen*

Anschließend kann über den Servermanager im Tools-Menue die Konsole *Routing und RAS* gestartet werden, alternativ über die Verwaltung oder aus dem Startbildschirm durch Eingabe von *Routing und RAS* und Auswahl des entsprechenden Suchergebnisses.

Zunächst muss *Routing und RAS* konfiguriert werden. Dazu in der Managementkonsole über das Aktionsmenue *Routing und RAS konfigurieren und aktivieren* auswählen. In den folgenden Fenstern wird *Benutzerdefinierte Konfiguration* und *LAN-Routing* gewählt.

<p>Setup-Assistent für den Routing- und RAS-Server</p> <p><b>Konfiguration</b> Sie können eine beliebige Kombination an Diensten wählen, oder Sie können diesen Server benutzerdefiniert anpassen.</p> <ul style="list-style-type: none"> <li><input type="radio"/> RAS (DFÜ oder VPN) Ermöglicht Remoteclients, eine Verbindung mit diesem Server über eine Erreichbarkeit verbinden oder eine sichere VPN-Internetverbindung herzustellen.</li> <li><input type="radio"/> Netzwerkklassierung (NAT) Ermöglicht internen Clients, eine Internetverbindung mit einer einzelnen öffentlichen IP-Adresse herzustellen.</li> <li><input type="radio"/> VPN-Zugriff und NAT Ermöglicht Remoteclients, eine Verbindung mit diesem Server über das Internet, und lokalen Clients eine Internetverbindung über eine einzige öffentliche IP-Adresse herzustellen.</li> <li><input type="radio"/> Sichere Verbindung zwischen zwei privaten Netzwerken Verbindet dieses Netzwerk mit einem Remotelnetzwerk, wie z.B. einer Zweigstelle.</li> <li><input checked="" type="radio"/> Benutzerdefinierte Konfiguration Wählen Sie eine beliebige Routing- und RAS-Featurekombination aus.</li> </ul> <p style="text-align: center;"><a href="#">&lt; Zurück</a> <a href="#">Weiter &gt;</a> <a href="#">Abbrechen</a></p>	<p>Setup-Assistent für den Routing- und RAS-Server</p> <p><b>Benutzerdefinierte Konfiguration</b> Sie können die ausgewählten Dienste in der Routing- und RAS-Konsole konfigurieren, nachdem dieser Assistent fertig gestellt wurde.</p> <p>Wählen Sie die Dienste aus, die auf diesem Server aktiviert werden sollen.</p> <ul style="list-style-type: none"> <li><input type="checkbox"/> VPN-Zugriff</li> <li><input type="checkbox"/> DFÜ-Zugriff</li> <li><input type="checkbox"/> Bei Bedarf herstellende Verbindungen (für Zweigstellenrouting)</li> <li><input type="checkbox"/> NAT</li> <li><input checked="" type="checkbox"/> LAN-Routing</li> </ul> <p style="text-align: center;"><a href="#">&lt; Zurück</a> <a href="#">Weiter &gt;</a> <a href="#">Abbrechen</a></p>
--	--

Zum Abschluss der Konfiguration wird noch der Dienst gestartet und die Konsole ist zur Konfiguration bereit.



## Mit Routing- und RAS-Diensten arbeiten

Nachdem der Dienst eingerichtet ist, stehen in der Baumstruktur der Navigationsleiste die entsprechenden Menüpunkte zur Verfügung.

- ▶ Wählen Sie im Kontextmenü von *Statische Routen* den Eintrag *Neue statische Route*.  
Das Dialogfenster *Statische IPv6-Route* zur Einrichtung einer statischen Route wird geöffnet.

Eine neue statische Route erfordert folgende Parameter:

<b>Schnittstelle</b>	Die ausgehende Schnittstelle für Datenverkehr in das IPv6-Zielnetzwerk
<b>Ziel</b>	Das IPv6-Zielnetzwerk
<b>Präfixlänge</b>	Die Länge des Präfix in Bit
<b>Gateway</b>	Die link-lokale IPv6-Adresse (fe80::) des direkten Nachbar-Routers auf dem Weg ins IPv6-Zielnetzwerk, falls möglich
<b>Metrik</b>	<p>Ein Wert zur Gewichtung der Route, falls multiple Routen in das IPv6-Zielnetzwerk zur Verfügung stehen</p> <p>Ein kleinerer Wert entspricht einer höheren Gewichtung (default 256).</p>

- ▶ Überprüfen Sie die korrekte Arbeitsweise der statischen Routen bzw. die korrekte Zustellung von Paketen in entfernte Netzwerke über die festgelegten Pfade mit dem folgenden Konsolenbefehl:  
`tracert -6 IPv6-Ziel-Adresse`

```
C:\>tracert -6 google.de

Routenverfolgung zu google.de [2a00:1450:4001:810::2003]
über maximal 30 Hops:

 1  3 ms   2 ms   1 ms  fritz.box [2003:5e:c5e:f00:5e49:79ff:fe7a:a1fb]
 2  21 ms  20 ms  21 ms  2003:0:3f02:203::1
 3  22 ms  22 ms  21 ms  2003:0:3f02:228::2
 4  20 ms  19 ms  19 ms  2003:0:3f02:c1::1
 5  *       *       *   Zeitüberschreitung der Anforderung.
 6  22 ms  21 ms  21 ms  2003:0:1304:8010::2
 7  24 ms  22 ms  22 ms  2a00:1450:8000:48::1:1
```

```

8 23 ms 22 ms 22 ms 2001:4860:0:1::216e
9 23 ms 22 ms 23 ms 2001:4860:0:1::75d
10 22 ms 21 ms 22 ms fra15s09-in-x03.1e100.net
    2a00:1450:4001:810::2003]

Ablaufverfolgung beendet.

```

*Traceroute über ipv6*

## 7.8 Konfiguration statischer Routen – **netsh**

Eine alternative Möglichkeit, statische Routen zu konfigurieren, bietet das Kommandozeilen-Tool **netsh**.

- ▶ Starten Sie **netsh** und wechseln Sie in den IPv6-Kontext.

```
C:\>netsh
netsh>int ipv6
netsh interface ipv6>
```

*Aufruf von Netshell und Wechsel in den IPv6 Kontext*

### Route anlegen

Eine neue Route legen Sie mit dem Konsolenbefehl **add route** an. Er bietet folgende Syntax und Parameter:

**add route** <prefix/prefix-length> <interface> <next-hop> <metric=zahl>

<b>add route</b>	Befehl zum Anlegen einer statischen Route
<b>Parameter:</b>	
<b>prefix/prefix-length</b>	IPv6-Präfix/Länge der Präfix in Bit
<b>interface</b>	Schnittstellenbezeichnung oder Index-Nummer der ausgehenden Schnittstelle
<b>next-hop</b>	die link-lokale IPv6-Adresse (fe80::) des direkten Nachbar-Routers
<b>metric</b>	Angabe der Gewichtung (default 256)

Wenn die Route erfolgreich angelegt wurde, bestätigt das System mit OK.

```
netsh interface ipv6>add route fd00::/8 LAN fe80::214:cdff:fe01:67 metric=8
OK.

netsh interface ipv6>
```

*Hinzufügen einer Route für die Schnittstelle LAN*

### Anzeigen der Routing-Tabelle

Die Routing-Tabelle können Sie sich mit dem Konsolenbefehl **show route** im netsh interface ipv6 Kontext anzeigen lassen.

**show route**

Anhand des letzten Eintrages wird die eben angelegte Route erläutert:

<b>Veröff.</b>	Gibt an, ob die Route für die automatische IPv6-Konfiguration veröffentlicht/ angekündigt wird (vgl. Kap. 5)
<b>Typ</b>	Herkunft der Route (Manuell= statisch angelegt, ansonsten Protokollname)
<b>Met</b>	Metrik; interne Gewichtung zur bevorzugten Benutzung einer Route; je kleiner der Wert, desto bevorzugter die Route
<b>Präfix</b>	Netzadresse des Zielnetzwerks inklusive Netzmasken-Länge
<b>Idx</b>	Interne Index-Nummer der Schnittstelle  Im oberen Konfigurationsbeispiel wurde die Schnittstelle unter ihrer Bezeichnung LAN referenziert; diese Schnittstelle hat hier die interne Index-Nummer 10.
<b>Gateway/ Schnittstelle</b>	Nachbarrouter, an den Datenverkehr ins angegebene Zielnetzwerk gesendet werden muss  <i>oder</i> Eigene Schnittstelle, über die der Datenverkehr ins angegebene Zielnetzwerk gesendet werden muss

netsh interface ipv6>show route						
Veröff.	Typ	Met	Präfix	Idx	Gateway/Schnittstelle	
Nein	Manuell	16	::/0	12	fe80::5e49:79ff:fe7a:a1fb	
Nein	System	256	::1/128	1	Loopback Pseudo-Interface	
Nein	Manuell	8	fd00::/8	12	fe80::214:cdf:fe01:67	
Nein	Manuell	16	fd00:a:b:c::/64	12	LAN	
Nein	System	256	fd00:a:b:c:21d:9ff:fede:b8ec/128	12	LAN	

Auflisten der IPv6 Routen, in der Mitte der manuell eingerichtete Eintrag

**Löschen einer Route**

Sie löschen eine Route mit dem Konsolenbefehl **delete route**. Er bietet folgende Syntax und Parameter:

► **delete route <prefix/prefix-length> <interface> <next-hop>**

Wenn die Route erfolgreich gelöscht wurde, bestätigt das System mit OK.

netsh interface ipv6>delete route fd00::/8 LAN fe80::214:cdf:fe01:67
OK.
netsh interface ipv6>

Löschen einer Route

## 7.9 Dynamisches Routing

### Neuerungen bei IPv6-fähigen Routing-Protokollen

Routing-Protokolle verwalten die Routing-Tabelle eines Routers dynamisch – durch Austausch von Routing-Informationen.

Generell neue Eigenschaften IPv6-fähiger Routing-Protokolle:

- ✓ Senden von 128-Bit-Präfix-Informationen (anstelle 32 Bit)
- ✓ Verwenden als Next-Hop-Adresse ausschließlich die link-lokale IPv6-Adresse der Nachbarrouter (Aggregatable-Global- oder Unique-Local-IPv6-Adressen werden ausdrücklich nicht verwendet)
- ✓ Arbeiten link-based (verbindungsorientiert), da unter IPv6 mehrere Präfixe auf einer Verbindung möglich sind. IPv4-Routing-Protokolle arbeiten i. d. R. subnetwork-based (angeschlossenes logisches Netzwerk).

### Übersicht dynamischer Routing-Protokolle mit IPv6-Unterstützung

IPv6-fähige Versionen der populärsten IPv4-Routing-Protokolle sind verfügbar und teilweise bereits in Verwendung, wie z. B. BGP-4+ im IPv6-Internet.

**Interne Routing-Protokolle:** routen innerhalb eines „autonomous systems“ (AS)

- ✓ **RIPng for IPv6**  
RFC 2080 - RIPng for Ipv6  
Distance Vector Routing Protocol für Ipv6 (basiert auf RIPv2)
- ✓ **OSPFv3**  
RFC 2740 - OSPF for IPv6  
Link State Routing Protocol für IPv6 (basiert auf OSPFv2)
- ✓ **EIGRPv6**  
Cisco  
Ein proprietäres Routing Protocol der Firma Cisco Systems.  
Advanced Distance Vector Protocol
- ✓ **i/IS-IS for IPv6**  
ISO/EIC 10589 - Intermediate System to Intermediate System  
Protokoll aus der OSI Protokollfamilie.  
Link State Routing Protocol

**Externe Protokolle:** routen zwischen „autonomous systems“ (AS) – Internet Routing

- ✓ **BGP-4+** (multiprotocol BGP)  
RFC 2858 – Multiprotocol Extensions for BGP-4  
RFC 2545 – Use of BGP-4 Multiprotocol Extensions for IPv6 Interdomain Routing  
Das de facto von Providern und Organisationen im IPv6-Internet verwendete externe Routing-Protokoll.  
Path Vector Routing Protocol
- ✓ **OLSR**, Optimized Link State Routing nach RFC 3626 ist ein Routing Protokoll für mobile ad hoc Netze. Status ist experimentell. OLSR wird u.a. bei Freifunk genutzt.

IDRPv6 ist ebenfalls ein externes Routing-Protokoll (aus der OSI-Protokollfamilie) für IPv6, das jedoch nicht genutzt wird.



## 7.10 RIPng

### Grundlagen zu RIPng

RIPng – RIP next generation basiert auf RIPv2 und ist definiert in RFC 2080.

RIPng ist wie RIPv2 ein Distance Vector Routing Protocol und

- ✓ sendet Routen in Updates,
- ✓ sendet Updates in regelmäßigen Abständen (30 Sekunden Update Time) nur an direkte Nachbarn,
- ✓ verwaltet die Routing-Tabelle durch Empfang regelmäßiger Routing-Updates.

RIPng ist, wie RIPv2, aufgrund der Metrik (Hop Count, maximaler Hop Count 15) nur geeignet für kleinere Topologien mit nahezu gleichen Bandbreiten auf den Verbindungen.

Im Vergleich mit einem Link State Routing Protocol (wie OSPF) benötigt es, wegen der periodischen Updates, mehr Bandbreite, stellt aber geringere Anforderungen an CPU und RAM, da lediglich die Routing-Tabelle verwaltet wird.

RIPng zeichnet sich hauptsächlich durch seine Einfachheit bei der Konfiguration und beim Troubleshooting aus.

### Neuerungen von RIPng

- ✓ Verwaltung der 128-Bit-Prefix, Verwendung der Link-local-IPv6-Adresse als Next Hop, arbeitet verbindungsorientiert
- ✓ Neuer UDP-Port: 521
- ✓ Kommunikation über Multicast-IPv6-Adresse: ff02::9
- ✓ Verwaltet neues „Route Change Flag“, das bei Änderungen von Routen gesetzt wird
- ✓ Neuer Timer: Garbage-collect 120 Sekunden (ersetzt RIPv2 Flush Timer)
- ✓ Verwendet keine Autosummarization (automatische Verteilung von zusammengefassten Routen an Netzwerkgrenzen) mehr
- ✓ Verwendet den AH/ESP IPv6 Extension Header zur Authentifizierung
- ✓ Verwendet einen Namen – dadurch ist es jetzt möglich, mehrere Rip Prozesse zu konfigurieren

### Verfügbarkeit

- ✓ RIPng ist auf Hardware-Routern unterschiedlicher Hersteller (z. B. Cisco Systems) und neuerer Generation i. d. R. verfügbar.
- ✓ Auf Linux-Systemen muss zusätzliche Software installiert und konfiguriert werden (z. B. quagga).
- ✓ Auf Windows Server steht RIPng nicht zur Verfügung.

### RIPv2 vs. RIPng

	RIPv2	RIPng
Distance Vector Protocol	Ja	
Metrik	Hop Count (maximum Hop Count 15, Metrik 16 = unreachable)	
Administrative Distanz	120	
Vermeidung von Schleifen	Split Horizon, Route Poisoning, Poison Reverse, Hold Down	

	RIPv2	RIPng
Triggered Updates	Ja (inkrementell)	
Timer (in Sekunden)	Update: 30 Invalid: 180 Flush: 240	Update: 30 Timeout: 180 Garbage-collect: 120
UDP Port	520	521
Multicast IP	224.0.0.9	ff02::9 (link local scope)
Next-Hop	IPv4-Adresse (der ausgehenden Schnittstelle des sendenden Routers)	Link-Local-IPv6-Adresse (der ausgehenden Schnittstelle des sendenden Routers)
Funktionalität/Aktivierung	Netzwerk-basierend (wird durch Angabe von Net-IDs auf Schnittstellen aktiviert)	Link-basierend (wird direkt auf einer Schnittstelle aktiviert)
Information in Updates	32 Bit Net-ID, Länge der Netzmaske, Metrik	128-Bit-Prefix, Länge der Prefix, Metrik, Route Change Flag
Automatische Routenzusammenfassung	Ja, default (abschaltbar)	Nein
Manuelle Routenzusammenfassung	Administrativ einstellbar	
Authentifizierung	Ja – Plain Text/MD5 Hash	Ja – AH/ESP Extension Header
Redistribution	Eine Weiterverteilung von Routing-Informationen aus anderen Informationsquellen (direkt verbunden, statisch angelegt, aus andern Routing-Protokollen) muss auf RIPng-fähigen Geräten – vom Hersteller – bereitgestellt werden.	
Filterung	Eine Funktion, um Routing-Updates reglementieren zu können, muss auf RIPng-fähigen Geräten – vom Hersteller – bereitgestellt werden.	

## 7.11 OSPFv3

### Grundlagen zu OSPFv3

OSPFv3 (Open Shortest Path First Version 3) basiert auf OSPFv2 und ist in RFC 5340 definiert. Das Protokoll unterstützt IPv6 und IPv4 und ist wie OSPFv2 ein Link State Routing Protocol. Es bietet folgende grundlegenden Eigenschaften:

- ✓ sendet lokale Link-Informationen (Verbindungsinformationen) in Updates
- ✓ sendet Updates (LSAs – Link State Advertisements) nur bei Änderungen an alle OSPF Router im Netzwerk
- ✓ verwaltet die Informationen aller anderen Router in der LSDB – Link State Database
- ✓ berechnet mithilfe des Dijkstra-Algorithmus aus den Topologie-Informationen innerhalb der LSDB die besten Wege in Zielnetzwerke und schreibt sie in die Routing-Tabelle
- ✓ tauscht mit direkten Nachbarn Hello-Pakete aus
- ✓ verwaltet Nachbarschafts-Tabelle, LSDB und Routing-Tabelle

OSPFv3 ist, wie OSPFv2, nicht allein aufgrund der Metrik (auf Bandbreite basierend) für große Topologien mit unterschiedlichen Bandbreiten auf den Verbindungen geeignet, sondern kann große Netzwerke in unterschiedliche Areas (Routing-Bereiche) unterteilen, was enorme Vorteile für ein effizienteres Routing bedeutet.

Im Vergleich mit einem Distance Vector Routing Protocol wie RIP benötigt es weniger Bandbreite (sendet nur bei einer Änderung Informationen über die Änderung), aber mehr RAM, da mehr Informationen verwaltet werden müssen, und mehr CPU, da die Routing-Tabelle bei jeder Änderung innerhalb der LSDB neu berechnet werden muss.

OSPFv3 stellt eine Reihe nützlicher Funktionen zur Verfügung und ist sehr schnell, was es zum meistverwendeten internen Routing-Protokoll in großen Netzwerken werden ließ, ist jedoch komplex in seiner Arbeitsweise – was sich bei Konfiguration und Troubleshooting bemerkbar macht.

### Neuerungen von OSPFv3

- ✓ Verwaltung der 128-Bit-Präfix, Verwendung Link-Local-IPv6-Adresse als Next Hop, arbeitet verbindungsorientiert
- ✓ Kann für IPv6 UND IPv4 verwendet werden
- ✓ Kann multiple Prozesse auf einer Verbindung verwalten (z. B. eine pro Stack)
- ✓ Neue Multicast-IPv6-Adressen: ff02::5 und ff02::6
- ✓ Modifizierter Paket-Header
- ✓ Neue LSA-Typen: LSA Type 8 „Link“, LSA Type 9 intra-area-prefix“
- ✓ Umbenennung von LSA-Typen: LSA Type 3 „interarea-prefix“, LSA Type 4 „interarea-router“
- ✓ Einführung von Gültigkeitsbereichen (scopes) für LSAs: link-lokal (Type 9), AS (Type 5), area (alle anderen LSAs)
- ✓ Neue Netzwerkarten (network types) für NBMA-Verbindungen
- ✓ Verwendet den AH/ESP IPv6 Extension Header zur Authentifizierung

### Verfügbarkeit von OSPFv3

- ✓ OSPFv3 ist auf Hardware-Routern unterschiedlicher Hersteller (z. B. Cisco System) und neuerer Generation i. d. R. verfügbar.
- ✓ Auf Linux-Systemen muss zusätzliche Software installiert und konfiguriert werden (z. B. quagga).
- ✓ Auf Windows Server 2003 und Server 2008 steht OSPFv3 nicht zur Verfügung.

### OSPFv3 vs. OSPFv2

Übersicht der Unterschiede/Gemeinsamkeiten zwischen OSPFv2 und OSPFv3:

	OSPFv2	OSPFv3
Link State Routing Protocol	Ja	
Metrik	Kosten (i. d. R. basierend auf Bandbreite)	
Administrative Distanz	110	
Pakettypen	Identisch	
Nachbarschaften	Identisch	
LSA-Kommunikation	Identisch	
Router-ID	Identisch (32 Bit)	
Link-ID	Identisch (32 Bit)	
Area-ID	Identisch (32 Bit)	
Area-Typen	Identisch	
Multicast	224.0.0.5 224.0.0.6	ff02::5 ff02::6

	OSPFv2	OSPFv3
Next Hop	IPv4 Adresse (der ausgehenden Schnittstelle des sendenden Routers)	link local IPv6 Adresse (der ausgehenden Schnittstelle des sendenden Routers)
Funktionalität/Aktivierung	netzwerk-basierend (wird durch Angabe von NetIDs auf Schnittstellen aktiviert)	link-basierend (wird direkt auf einer Schnittstelle aktiviert)
Information in Updates	32 Bit Net-ID	128-Bit-Präfix
Authentifizierung	MD5	AH/ESP Extension Header
LSA-Typen	Type 1: Router Type 2: Network Type 3: Network Summary Type 4: ASBR Summary Type 5: AS External Type 7: NSSA External	Type 1: Router Type 2: Network Type 3: Inter-area-prefix Type 4: Inter-area-router Type 5: AS External Type 7: NSSA External Type 8: Link Type 9: Intra-area-prefix
NBMA Network-Typen	Non-broadcast Point-to-multipoint	Non-broadcast Point-to-multipoint Broadcast
OSPF-Prozesse	Ein Prozess pro Schnittstelle	Multiple Prozesse pro Schnittstelle

## 7.12 IDRIPv2

IDRIPv2 (Inter-Domain Routing Protocol) basiert auf IDRIP (ISO 10747) und ist der IPv6-fähige Nachfolger von IDRIP.

IDRIP ist ein externes Routing-Protokoll (Internet-Routing) aus der OSI-Protokollfamilie, das auf BGP-4 (RFC 2711) basiert und Routing Domains (autonome Systeme) miteinander verbindet, d. h. für das IPv6-Internet-Routing verwendet werden kann.

BGP-4 ist jedoch das im globalen Netz (Internet) von Providern und Organisationen de facto verwendete externe Routing-Protokoll, das als BGP-4+ – Multiprotocol BGP für IPv6 erweitert wurde (RFC 2858 – Multiprotocol Extensions for BGP-4 und RFC 2545 – Use of BGP-4 Multiprotocol Extensions for IPv6 Interdomain Routing) und momentan für das IPv6-Internet-Routing verwendet wird.

IDRIP kann mit BGP-4 zusammenarbeiten. Dazu existieren einige ältere Draft-Dokumente bei der IETF, die aber nicht weiterentwickelt wurden.

Mittlerweile wurde IDRIP von der IETF als *historical* eingeordnet – da es **nicht** verwendet wurde/wird. IDRIPv2, dem IPv6-fähigen Nachfolger von IDRIP, kann der gleiche Stellenwert zugeordnet werden.

Unter <http://www.bgp4.net/rs6> ist eine Liste mit aktuellen IPv6-Root-Servern im Internet zu finden – einige sind via telnet oder ssh erreichbar, und Sie können sich z. B. die IPv6-Routen anschauen (Login-Informationen werden im Banner angezeigt).



Nachfolgend ein Beispiel für eine entsprechende Telnet-Session.

```
► c:\> telnet route-server.ip.att.net  
----- route-server.ip.att.net -----  
----- AT&T IP Services Route Monitor -----
```

The information available through route-server.ip.att.net is offered by AT&T's Internet engineering organization to the Internet community.

This router maintains eBGP peerings with customer-facing routers throughout the AT&T IP Services Backbone:

IPv4:	IPv6:	City:
12.122.124.12	2001:1890:ff:ffff:12:122:124:12	Atlanta, GA
12.122.124.67	2001:1890:ff:ffff:12:122:124:67	Cambridge, MA
12.122.127.66	2001:1890:ff:ffff:12:122:127:66	Chicago, IL
12.122.124.138	2001:1890:ff:ffff:12:122:124:138	Dallas, TX
12.122.83.238	2001:1890:ff:ffff:12:122:83:238	Denver, CO
12.122.120.7	2001:1890:ff:ffff:12:122:120:7	Fort Lauderdale, FL
12.122.125.6	2001:1890:ff:ffff:12:122:125:6	Los Angeles, CA
12.122.125.44	2001:1890:ff:ffff:12:122:125:44	New York, NY
12.122.125.106	2001:1890:ff:ffff:12:122:125:106	Philadelphia, PA
12.122.125.132	2001:1890:ff:ffff:12:122:125:132	Phoenix, AZ
12.122.125.165	2001:1890:ff:ffff:12:122:125:165	San Diego, CA
12.122.126.232	2001:1890:ff:ffff:12:122:126:232	San Francisco, CA
12.122.125.224	2001:1890:ff:ffff:12:122:125:224	Seattle, WA
12.122.126.9	2001:1890:ff:ffff:12:122:126:9	St. Louis, MO
12.122.126.64	2001:1890:ff:ffff:12:122:126:64	Washington, DC

\*\*\* Please Note:

Ping and traceroute delay figures measured here are unreliable, due to the high CPU load experienced when complicated show commands are running.

For questions about this route-server, send email to: jayb@att.com

\*\*\* Log in with username 'rvviews', password 'rvviews' \*\*\*

login:

Nach dem Login kann man sich mit ? eine Befehlsübersicht ausgeben lassen. Mit Befehl ? erhält man weitere Informationen:

```
--- JUNOS 17.1R1-S1 built 2017-04-07 08:21:13 UTC
rvviews@route-server.ip.att.net> ?
Possible completions:
  ping                  Ping remote target
  quit                 Exit the management session
  set                  Set CLI properties, date/time, craft interface
message
  show                 Show system information
  traceroute           Trace route to remote host
rvviews@route-server.ip.att.net>
rvviews@route-server.ip.att.net> show bgp summary
```

zeigt eine Übersicht der verwalteten BGP-Routen

Sie können die Ausgabe mit der Leertaste fortsetzen oder mit der Taste **Q** abbrechen.

Ausloggen mit quit oder exit:

► rvviews@route-server.ip.att.net> quit

# 8 Übergangsmechanismen

## In diesem Kapitel erfahren Sie

- ✓ was Dual-Stack bedeutet
- ✓ was Dual-Stack Lite bedeutet
- ✓ welche Tunnelmechanismen es gibt
- ✓ was ein Übersetzungsverfahren ist

## Voraussetzungen

- ✓ IPv6-Adressaufbau
- ✓ Routing und DNS

## 8.1 Voraussetzungen für Migration auf IPv6

Um in der Praxis eine erfolgreiche Migration auf IPv6 durchführen zu können, müssen für einen bestimmten Zeitraum des Netzwerkbetriebes Übergangsmechanismen angeboten werden.

- ✓ Im sog. **Dual-Stack**-Betrieb sind die im Netz vorhandenen Geräte sowohl über IPv4 als auch über IPv6 erreichbar, wobei es keine Rolle spielt, welcher der beiden IP-Stacks auf den einzelnen Geräten aktiviert ist; dies wird vom „Gesamtsystem“ erkannt und geregelt.
- ✓ Das **Tunneling** sorgt dafür, dass der Dual-Stack-Betrieb auch über Netzbereiche hinweg (z. B. Internet), deren IP-Stack-Konfiguration unbekannt ist, funktioniert.

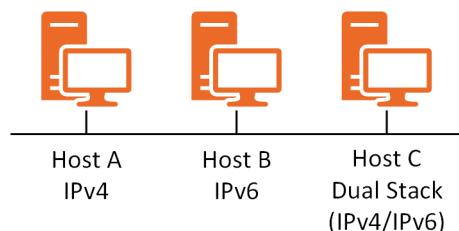
Übersetzungsverfahren werden angewendet, wenn Geräte mit verschiedenen Protokollen in der Vermittlungsschicht miteinander kommunizieren müssen

## 8.2 Dual-Stack

Als **Dual-Stack** wird der gleichzeitige Betrieb von IPv4 und IPv6 auf einem Netzwerk-Gerät bezeichnet. Dies ist eine Methode zur Integration und Migration von IPv6 in einem IPv4-Netzwerk.

Dual-Stack kann sowohl auf einen Host (DSH; **Dual-Stack-Host**) als auch auf einem Router (DSR; **Dual-Stack-Router**) implementiert sein. Voraussetzung hierfür ist die IPv6-Unterstützung durch das jeweilige Betriebssystem. Die vorhandenen Betriebssysteme müssen gegebenenfalls durch Software-Updates Dual-Stack-fähig gemacht werden (z. B. durch Einspielen von Service-Packs).

Im Dual-Stack-Betrieb können IPv4- und IPv6-Geräte miteinander kommunizieren, obwohl die Protokolle in ihrem „originären“ Zustand nicht miteinander kompatibel sind. So kann im nebenstehenden Schaubild Host C (Dual-Stack) sowohl mit Host A als auch mit Host B kommunizieren, nicht aber Host A (nur IPv4) mit Host B (nur IPv6).



Die vorhandenen Dienste im IPv4-Netzwerk können also weiter wie gewohnt betrieben werden. Neu hinzugefügte IPv6-Dienste können nach und nach und ohne störende Konsequenzen für den Betrieb des Gesamtsystems schrittweise implementiert werden.

So hängt z. B. die Wahl des jeweiligen Protokolls für einen Verbindungsaufbau u. a. vom Nameserver ab:

- ✓ Der Nameserver selbst kann, muss aber nicht zwangsläufig für Dual-Stack-Betrieb ausgerüstet sein. Er muss nur über eines der Protokolle erreichbar sein.
- ✓ Der Nameserver-Dienst muss neben den bekannten IPv4-A-Records auch IPv6-AAAA-Records auflösen können. Der Nameserver kann auf Anforderung beide Adressen zurückliefern.

Dual-Stack-Hosts, denen die DNS-Namensauflösung eine IPv4-Adresse liefert, kommunizieren über IPv4, wird eine IPv6-Adresse geliefert, kommuniziert der Host über IPv6. Aus diesem Grund können auf IPv4 basierende Anwendungen und Dienste in einem IPv6-Netzwerk weiterhin ohne Einschränkung ausgeführt werden.

Neben den offensichtlichen Vorteilen des Dual-Stack-Verfahrens, das technisch den problemlosen Parallelbetrieb beider IP-Versionen erlaubt, gibt es allerdings auch Aspekte, die insbesondere in der System-Administration als unangemessen aufwendig und deshalb eher nachteilig betrachtet werden.

- ✓ Administrative Doppelbelastung durch die Protokoll-Migration: Unterstützung von zwei separaten IP-Stacks in Einrichtung, Konfiguration, Fehlerbehebung, ...
- ✓ Unterschiedliche Kommandos für IPv4- und IPv6-Protokolle
- ✓ Der Nameserver muss in der Lage sein, beide Eintragungstypen (A-Host, AAAA-Host) auflösen zu können.
- ✓ Durch eine zusätzlich erforderliche Reverse-Lookupzone für IPv6 wird ein erhöhter Konfigurationsaufwand des Nameservers erforderlich.
- ✓ Für Router entstehen ein höherer Speicherbedarf sowie eine höhere CPU-Auslastung für die zusätzlichen Routing-Tabellen sowie ein erhöhter Konfigurationsaufwand für die IPv6-Routing-Protokolle.

## 8.3 Dual-Stack Lite (DS-Lite)

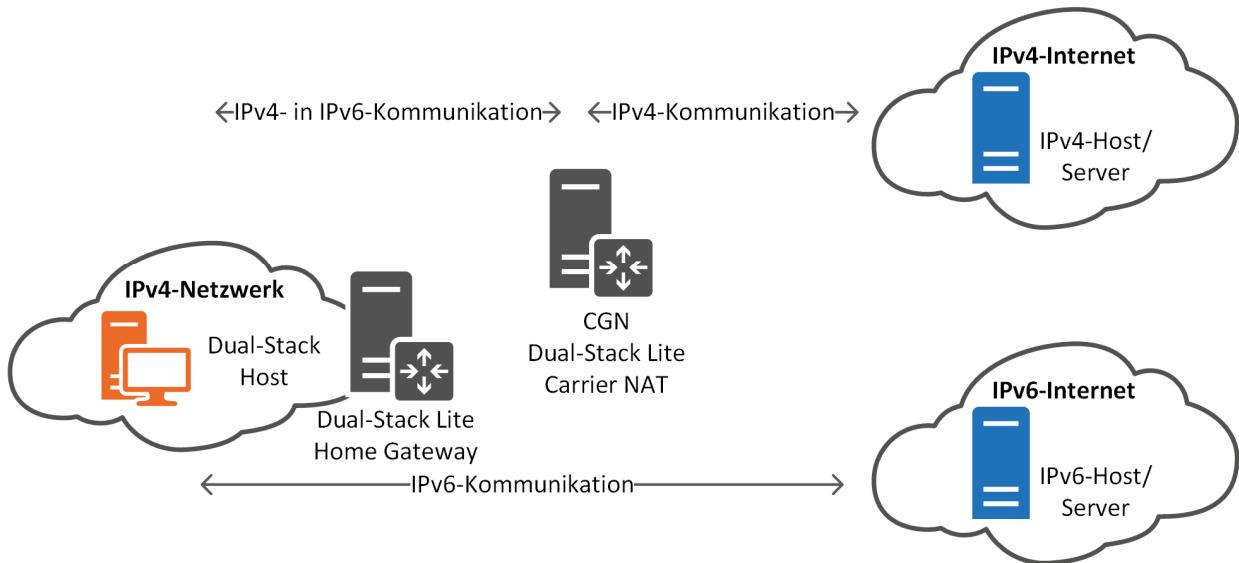
Durch die Adressknappheit bei IPv4 können viele Provider ihren Kunden keine öffentlichen IPv4-Adressen zur Verfügung stellen. Dem Kunden wird stattdessen eine Private IP zugewiesen, die der Provider an den Netzgrenzen auf eine Public IP umsetzt. IPv6 ist davon nicht betroffen. Dieses „Mischverfahren“ aus Private IPv4 und Globalem IPv6 Präfix wird als **Dual-Stack Lite (DS-Lite)** bezeichnet.

Im Dual-Stack-Lite-Verfahren weist der Provider dem Kunden ein globales IPv6-Netz für den Router mit passendem Präfix zu. Somit ist eine native IPv6-Kommunikation mit IPv6-Zielnetzwerken gewährleistet.

Weiterhin weist der Provider dem Kunden zusätzlich eine private (nicht öffentliche) IPv4-Adresse zu.

Somit besitzt der Kunde nun auf seinem Router ein globales IPv6-Präfix sowie eine private IPv4-Adresse mit den dazugehörigen Default Gateways des Provider-Routers (**CGN** – Carrier-grade NAT; beim Provider durchgeführtes NAT).

## ISP – Internet Service Provider



Mit dem Dual-Stack-Lite-Verfahren spart der Provider die Vergabe der knapp gewordenen globalen IPv4-Adressen an den Kunden. Der Provider übernimmt hier selbst das NAT von privaten IPv4-Adressen auf globale IPv4-Adressen durch den Einsatz von Carrier-Grade-NAT-Routern.

### Vorgangsbeschreibung

Wenn ein Anwender eine Webseite, z. B. [www.herd़t.com](http://www.herd़t.com) aufruft, wird als Erstes eine DNS-Abfrage nach der IP-Adresse von [www.herd़t.com](http://www.herd़t.com) ausgelöst. Die Anfrage könnte wie folgt aussehen:

```
> nslookup www.herd़t.com
Server: fritz.box
Address: 192.168.0.1
```

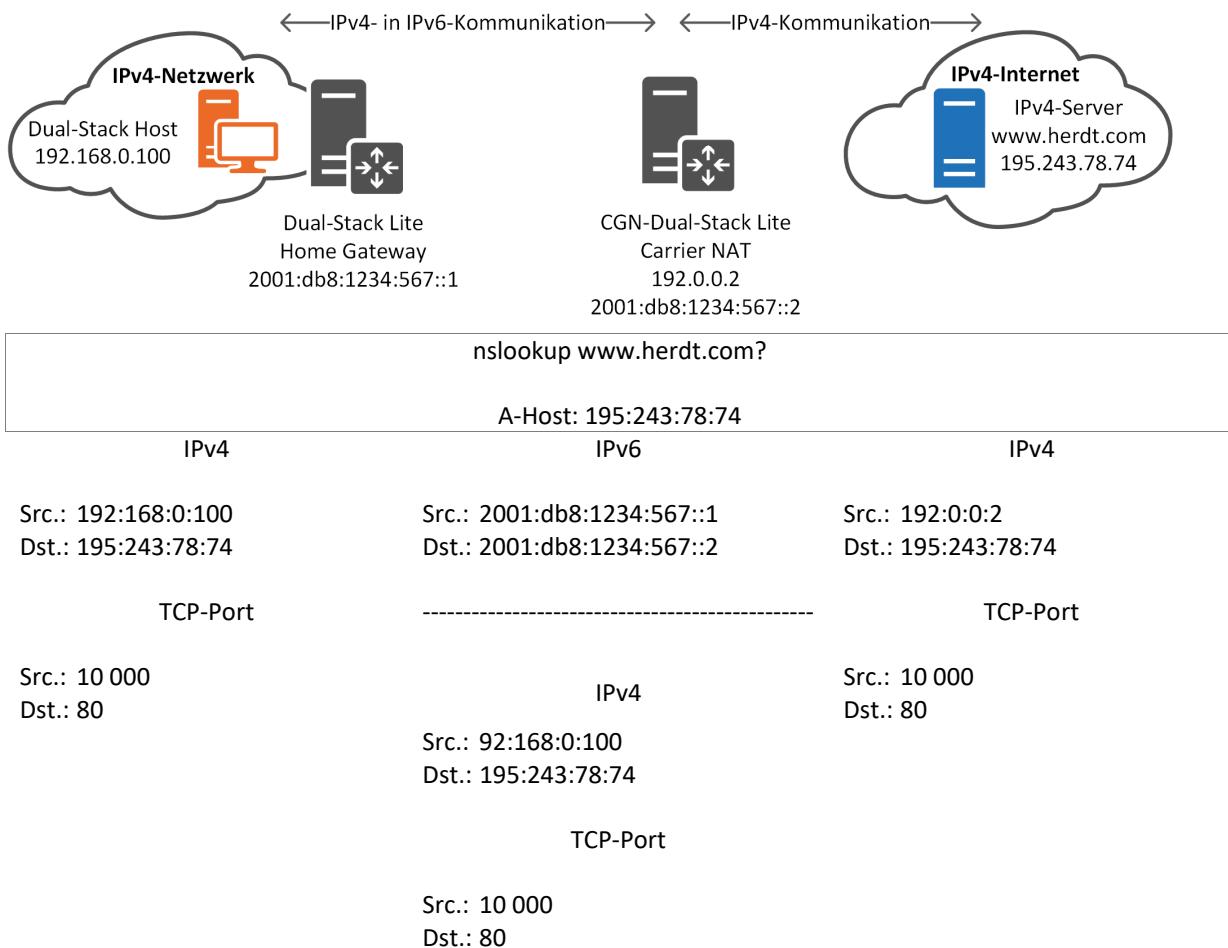
Nicht autorisierende Antwort:  
Name: www.herd़t.com  
Address: 195.243.78.74

Die Antwort des DNS-Servers bietet im Dual-Stack-Lite-Verfahren drei Optionen:

- ✓ Enthält die Antwort **nur eine IPv6-Adresse**, dann baut der Host eine Verbindung direkt über die globale IPv6-Adresse zum Zielhost auf.
- ✓ Enthält die Antwort **sowohl eine IPv4- als auch eine IPv6-Adresse**, dann wird bevorzugt die IPv6-Adresse für die Verbindungsaufnahme benutzt.
- ✓ Enthält die Antwort – wie im Beispiel – **nur eine IPv4-Adresse**, dann kommt das Dual-Stack-Lite-Verfahren zum Einsatz.

In unserem Beispiel sendet der Client ein IPv4-Datagramm mit seiner privaten IPv4-Adresse an seinen Dual-Stack-Lite Home Gateway-Router. Dieser Dual-Stack-Lite Home Gateway-Router kapselt das IPv4-Datagramm in ein IPv6-Datagramm und sendet es an den CGN-Dual-Stack-Router des Providers.

Der CGN-Dual-Stack-Router des Providers extrahiert das IPv4-Datagramm und leitet es auf die globale IPv4-Adresse des Ziels (195.243.78.74) weiter.

**ISP – Internet Service Provider**

Für die IPv6-Verbreitung ist DS-Lite positiv zu betrachten. In der Praxis zeigen sich allerdings gravierende Nachteile für Kunden, die auf eine öffentliche IPv4-Adresse am Router angewiesen sind. Der Heimrouter ist anders als bei nativem IPv4 durch das zusätzliche Carrier Grade NAT nicht mehr aus dem IPv4 Internet erreichbar. Dienste wie der Zugriff auf Heimautomation, NAS, Webcam oder IP-Telefonie (VoIP) sind gar nicht oder nur eingeschränkt möglich.

## 8.4 Tunnelmechanismen

### Grundlagen

Ein Tunnel überträgt ein eingebettetes Netzwerkprotokoll in einem anderen Netzwerkprotokoll. In der Folge werden vier Tunnelmechanismen beschrieben, die in der Praxis am häufigsten Verwendung finden.

<b>6in4-Tunnel</b>	Stellt über ein IPv4-Netzwerk eine Verbindung ins IPv6-Internet her
<b>6to4-Tunnel</b>	Verbindet zwei IPv6-Netzwerke über eine öffentliche IPv4-Infrastruktur
<b>4in6</b>	Transportiert IPv4 Pakete über eine IPv6 Verbindung
<b>6rd</b>	IPv6 rapid deployment, ähnlich wie 6to4 aber mit Adressbereich des Providers
<b>ISATAP-Tunnel</b>	Verbindet Dual-Stack Nodes über IPv4 Netzwerke
<b>Teredo-Tunnel</b>	Verbindet ein IPv4-Netzwerk mit einem IPv6-Netzwerk über unbekannte öffentliche Netze, wobei an der Schnittstelle zwischen dem öffentlichen Netz und dem IPv6-Netz ein sog. Teredo-Server zum Einsatz kommen muss

Das Tunneling mit ISATAP und Teredo wird erst mit der Dual-Stack-Implementierung auf einem Netzwerkhost möglich, denn mit der Installation und Aktivierung des IPv6-Protokolls werden die beiden Tunneladapter für den „ISATAP“- bzw. den „Teredo“-Tunnel installiert, die in der Folge für die Kommunikation mit etwaigen Tunnelendpunkten zur Verfügung stehen.

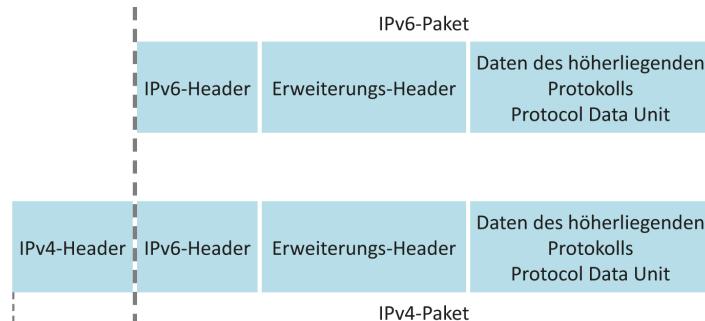
Tunnelverfahren werden eingesetzt, wenn bei einer Verbindung zwischen zwei IPv6-Netzen ein oder mehrere IPv4-Netze überbrückt werden müssen oder IPv4-Netze mit IPv6-Netzen kommunizieren wollen. In diesen Fällen wird das IPv6-Paket auf der Strecke durch die IPv4-Netze mit dem dazugehörigen IPv4-Header eingekapselt und am Tunnelendpunkt (Übergang IPv4-Netz zu IPv6-Netz) entsprechend wieder entpackt. Dieses Verfahren bezeichnet man als 6in4-Tunneling.

### 6in4-Tunnel

Der 6in4-Tunnel stellt ein Verfahren dar, in dem ein IPv6-Datenpaket in ein IPv4-Paket eingekapselt wird.

Ein so gewonnenes IPv4-Datagramm besteht aus

- ✓ einem IPv4-Header
- ✓ einem IPv6-Header
- ✓ einem IPv6 Extension Header (nur wenn erforderlich)
- ✓ dem IPv6-Payload



Im Protokoll-Feld des IPv4-Header wird über den Eintrag „41“ angegeben, dass es sich um ein eingekapseltes IPv6-Paket handelt. Der Tunnelendpunkt wird mit dem Protokoll-Feld 41 darauf hingewiesen, dass das Paket eingekapselt ist und für die Weiterleitung in ein IPv6-Netz „ausgepackt“ werden muss.

Durch den hinzugefügten IPv4-Header verringert sich die MTU des Datenpaketes um 20 Bytes auf 1480 Bytes. Endpunkte von 6in4-Tunneln können wechselnde IPv4-Adressen haben und sich auch hinter einem NAT-Router befinden.

Der bekannte Anbieter SixXS hat leider zum 6.6.2017 seinen Betrieb eingestellt, nachdem schon zuvor keine Neuanmeldungen und Konfigurationsänderungen möglich waren. Dort gehostete Tunnel/Netze ließen sich problemlos z. B. in der Fritz!Box direkt oder über die Tools AICCU (Automatic IPv6 Connectivity Client Utility) oder bei der Verwendung hinter NAT mit ANYIYA (Anything In Anything) auf Clients einrichten.

<b>IPv6-Unterstützung</b>	
<input checked="" type="checkbox"/> Unterstützung für IPv6 aktiv	
<b>IPv6-Anbindung</b>	
<input type="radio"/> Immer eine native IPv4-Anbindung nutzen (empfohlen) Zunächst wird eine native IPv4-Verbindung aufgebaut. Falls per DHCP ein 6RD-Server gelernt wurde, wird ein 6RD-Tunnel aufgebaut. Ansonsten wird versucht, eine native IPv6-Verbindung aufzubauen (Dual Stack).	
<input checked="" type="radio"/> Immer eine native IPv6-Anbindung nutzen Ihr Internetanbieter muss für diese Betriebsart natives IPv6 an Ihrem Anschluss unterstützen.	
<input type="radio"/> Immer ein Tunnelprotokoll für die IPv6-Anbindung nutzen IPv6 mit einem Tunnelprotokoll über eine herkömmliche IPv4-Anbindung verwenden. Für diese Betriebsart ist keine IPv6-Unterstützung durch Ihren Internetanbieter notwendig.	
<b>Verbindungseinstellungen</b>	
Wählen Sie ein Tunnelprotokoll	
<input type="radio"/> 6to4 Die Nutzung von 6to4 ist über nahezu jede IPv4-Adresse möglich und erfordert keine Anmeldung. Der nächstgelegene Tunnelendpunkt wird automatisch ermittelt.	
<input checked="" type="radio"/> SixXS Vor der Nutzung ist es notwendig, einen Tunnel mit Heartbeat-Unterstützung unter <a href="http://www.sixxs.net">http://www.sixxs.net</a> zu beantragen.	
Benutzername:	<input type="text" value="xxxxxx"/>
Kennwort:	<input type="text" value="****"/>
Tunnel-ID:	<input type="text" value="T38xxx"/>
<input type="radio"/> 6RD	
Die Nutzung von 6RD erfordert die Angabe des Tunnelendpunktes und der im Tunnel genutzten IPv6-Adresse.	
IPv4-Adresse des Tunnelendpunktes:	<input type="text" value="192"/> : <input type="text" value="88"/> : <input type="text" value="99"/> : <input type="text" value="1"/>
IPv6-Präfix	<input type="text" value="2002"/> / <input type="text" value="16"/>
IPv4-Maskenlänge	<input type="text" value="0"/>
<input type="radio"/> 6in4	
Geben Sie den Tunnelendpunkt an.	
IPv4-Adresse des Tunnelendpunktes:	<input type="text" value="0"/> : <input type="text" value="0"/> : <input type="text" value="0"/> : <input type="text" value="0"/>
IPv6-Adresse des Tunnelendpunktes	<input type="text" value="::"/>
Lokale IPv6-Adresse	<input type="text" value="::"/>
IPv6-Präfix	<input type="text" value="::"/> / <input type="text" value="0"/>

**Verschiedene Möglichkeiten der IPv6-Nutzung bei einer Fritz!Box**

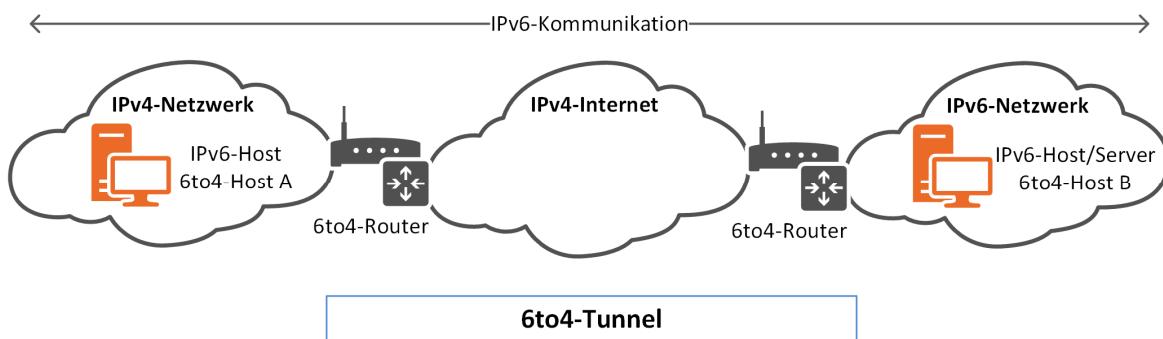
Die Beweggründe für die Einstellung des Dienstes sind auf der Webseite [www.sixxs-net](http://www.sixxs-net) erläutert, die als Archiv weiter online bleibt. Eine Übersicht über Tunnelanbieter findet sich in Wikipedia unter [https://de.wikipedia.org/wiki/Liste\\_von\\_IPv6-Tunnelbrokern](https://de.wikipedia.org/wiki/Liste_von_IPv6-Tunnelbrokern) oder etwas ausführlicher in der englischen Version [https://en.wikipedia.org/wiki/List\\_of\\_IPv6\\_tunnel\\_brokers](https://en.wikipedia.org/wiki/List_of_IPv6_tunnel_brokers).

### 6to4-Tunnel

6to4-Tunnel erlauben die Kommunikation zwischen zwei IPv6-Netzwerken über eine IPv4-Infrastruktur, ohne dass hierfür ein dedizierter Tunnel konfiguriert werden muss.

An den Schnittstellen vom IPv6- zum IPv4-Netzwerk wird ein 6to4-Router installiert und konfiguriert. An der Schnittstelle zum öffentlichen IPv4-Netzwerk (Internet) ist die IPv4-Adresse eine öffentliche IPv4-Unicast-Adresse.

Der untere Balken umfasst in seiner Breite den Tunnelbereich, hier von Router zu Router.



Der Präfix für diesen Tunnelmechanismus lautet (2002::/16) und setzt sich wie folgt zusammen:

- ✓ 6to4-Präfix
- ✓ IPv4-Adresse des 6to4-Routers in hexadezimaler Schreibweise
- ✓ Subnet-Identifier
- ✓ Interface-Identifier des Zielhostes

2002	IPv4-Adresse	Subnet-ID	Interface-ID
16 Bit	32 Bit	16 Bit	64 Bit

Der Kommunikationsablauf stellt sich dann wie folgt dar:

- ✓ Der 6to4-Router generiert seine 6to4-IPv6-Adresse aus seiner öffentlichen IPv4-Adresse und kündigt den generierten Präfix im Subnet über die Autokonfiguration der 6to4-Hosts an.
- ✓ Möchte nun ein 6to4-Host mit einem 6to4-Host außerhalb seines Subnetzes kommunizieren, so wird das IPv6-Paket an den 6to4-Router geschickt.
- ✓ Der Router verpackt die IPv6-Pakete in IPv4-Pakete und adressiert diese an den Empfänger-6to4-Router.
- ✓ Der Empfänger-Router entpackt die IPv4-Pakete und leitet dann das ursprüngliche IPv6-Paket weiter.

Die Kommunikation zwischen 6to4-Hosts im selben Netz verläuft direkt über IPv6 (kein Tunnel). Für diese Kommunikation wird kein Router benötigt.

Die Verwendung von NAT ist bei Einsatz eines 6to4-Tunnels nicht möglich, denn der Router besitzt auf der internen Schnittstelle eine private IPv4-Adresse. Diese Adresse würde dann im 6to4-Präfix enthalten sein. Der Empfänger-Host würde somit eine private IPv4-Adresse als Absenderadresse erhalten und kann dem Absender-Host keine Antwort senden.

## 6rd (IPv6 Rapid Deployment)

6rd ist seit 2010 in RFC 5969 standardisiert und ähnelt 6to4 mit der Abweichung, dass statt eines fest definierten Präfixes (2002::/16) ein Präfix aus dem Pool des Providers verwendet wird. Es ermöglicht Providern somit eine einfache Zuteilung von IPv6-Präfixen an Endkunden mit wenigen Änderungen an der Infrastruktur. Dabei wird die IPv4-Adresse des Kunden im IPv6-Präfix des Kunden eingebunden. Die 6rd-Adresse setzt sich zusammen aus dem 6rd-Präfix (n), der IPv4-Adresse (oder Teilen davon)(o) und den Subnetzbits(m) sowie der 64 Bit langen Interface ID. Der IPv6-Traffic wird vom Router des Kunden über das Providernetzwerk zu dessen 6rd-Gateway getunnelt. Dort findet dann der Übergang ins öffentliche IPv6-Netz statt.

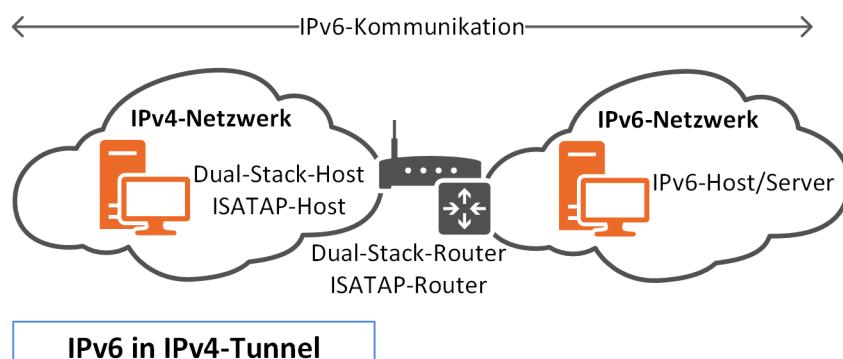
n bits	o bits	m bits	128-n-o-m bits
6rd Präfix	IPv4 Adresse	Subnet ID	Interface ID
6rd delegated prefix			

### Zusammensetzung der 6rd Adresse nach RFC 5969

Die Länge der verschiedenen Teile des 6rd Präfixes sind nicht festgelegt. Verwendet der Provider einen 32 Bit Präfix blieben z.B. noch 24 Bits der IPv4 Adresse und 8 Bit für Subnetze.

## ISATAP-Tunnel

Wenn sich ein Dual-Stack-Host in einem IPv4-Netzwerk mit einem IPv6-fähigen Router befindet und mit einem IPv6-Host kommunizieren will, kann das Intra-Site Automatic Tunnel Addressing Protocol (**ISATAP**) eingesetzt werden. Das Protokoll wurde von Cisco und Microsoft entwickelt und ist in RFC 5214 beschrieben. Der Unterschied zum weiter unten beschriebenen Teredo-Tunneling besteht darin, dass ISATAP vollständig manuell konfiguriert werden kann.



Im Gegensatz zum 6to4-Tunneling kann mit ISATAP nicht ein gesamtes Netzwerk über IPv4 hinweg verbunden werden. ISATAP setzt einen Dual-Stack auf dem Host voraus und wird als virtuelles Interface mit spezieller IPv6-Adresse eingebunden.

Das Unicast-Präfix einer ISATAP-Adresse kann ein globales, Link-Local- oder 6to4-Präfix sein.

Beispiel:

ISATAP-Präfix	Subnet-Präfix	IPv4-Adresse
64 Bit	32 Bit	32 Bit

Dabei wird im Subnet-Präfix angezeigt, ob es sich bei der nachfolgenden IPv4-Adresse um eine private oder eine öffentliche IPv4-Adresse handelt.

- ✓ 00 00 5e fe für private IPv4-Adressen
- ✓ 00 02 5e fe für öffentliche IPv4-Adressen

In den letzten 32 Bit ist dann die IPv4-Adresse (in IPv4-Notation) eingebettet.

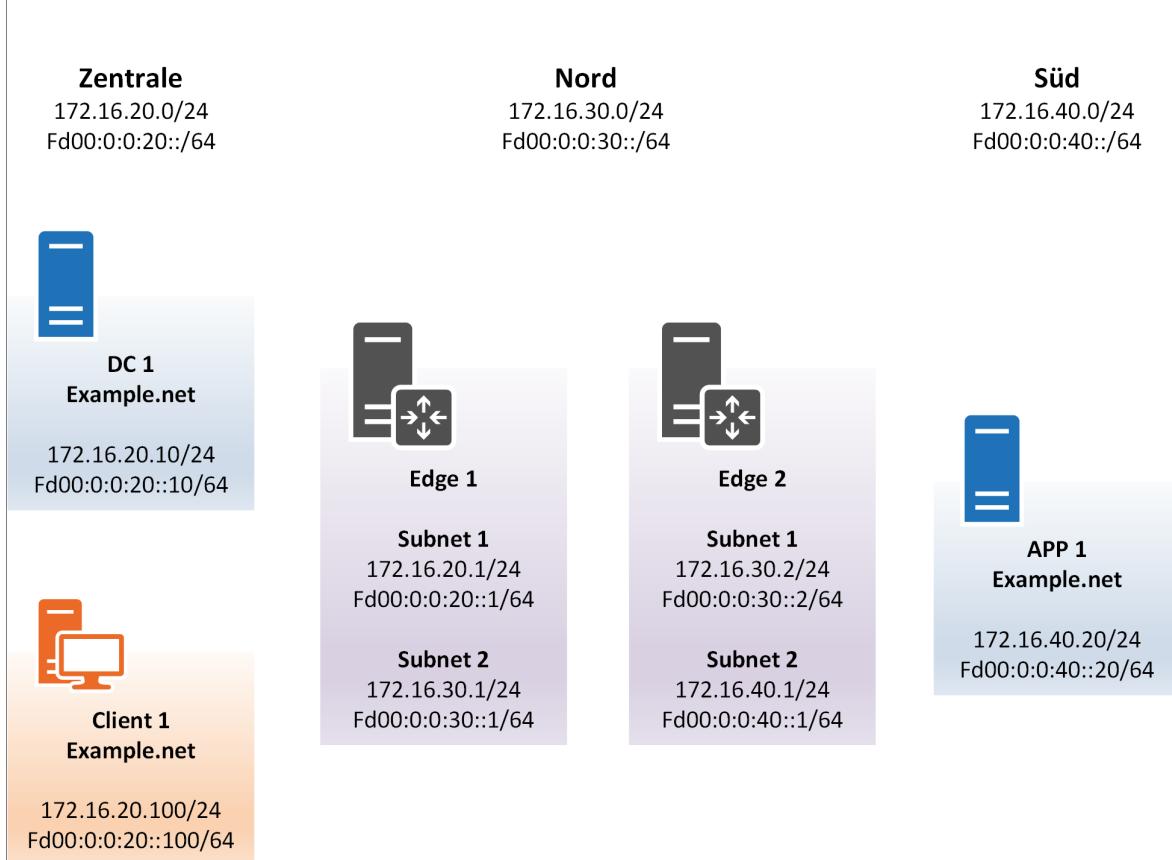
Der Kommunikationsablauf stellt sich wie folgt dar:

- ✓ Beim Start des ISATAP-Interfaces wird über eine DNS-Abfrage versucht, einen ISATAP-Router zu erreichen. Hierzu ist im Vorfeld ein A-Host-Eintrag auf die interne IPv4-Schnittstelle des ISATAP-Routers im DNS zu konfigurieren.
- ✓ Ist die DNS-Anfrage erfolgreich, wird ein IPv6-in-IPv4-Tunnel zu diesem ISATAP-Router aufgebaut und eingerichtet. Aus dem Subnet-Präfix und der Gateway-Adresse wird dann eine IPv6-ISATAP-Adresse generiert.

### Praxisbeispiel ISATAP-Tunnel

- Richten Sie Ihre Konfiguration nach dem Netzwerkplan ein, um das Praxisbeispiel nachzuvollziehen.
- ✓ Der DC1 ist als Domänencontroller sowie als DNS-Server für die Domäne **example.net** konfiguriert.
- ✓ Edge1 und Edge2 sind als Router mit dem Routing- und RAS-Dienst konfiguriert.
- ✓ Auf App1 ist der Webserver IIS installiert und konfiguriert.

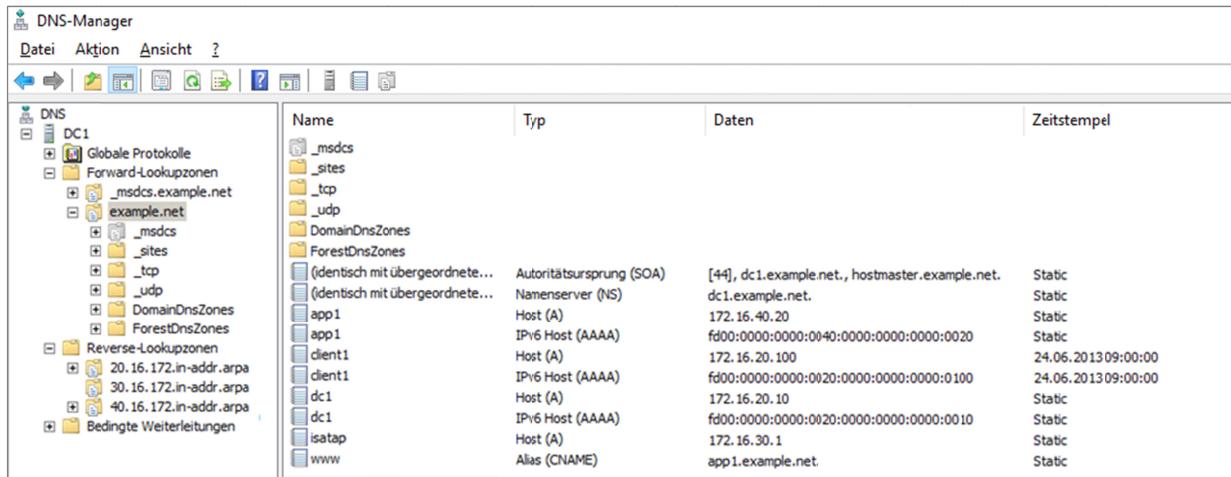
### Netzwerkplan ISATAP



### Konfiguration des DNS-Servers DC1

Auf DC1 wird der DNS-Dienst für die Unterstützung von ISATAP konfiguriert.

- Erstellen Sie einen A-Host-Eintrag `isatap` auf das ISATAP-Routerinterface Subnet2 172.16.30.1 in der Forward-Lookupzone `example.net`.



The screenshot shows the Windows DNS Manager interface. On the left, the tree view shows a domain structure under 'DNS'. A selected node is 'example.net' under 'Forward-Lookupzonen'. On the right, a table lists various DNS entries:

Name	Typ	Daten	Zeitstempel
_msdcsv	(identisch mit übergeordnete...)	Autoritätsursprung (SOA) [44], dc1.example.net, hostmaster.example.net.	Static
_sites	(identisch mit übergeordnete...)	Namenserver (NS) dc1.example.net.	Static
_tcp		Host (A) 172.16.40.20	Static
_udp		IPv6 Host (AAAA) fd00:0000:0000:0400:0000:0000:0000:0020	Static
DomainDnsZones			
ForestDnsZones			
app1	Host (A)	172.16.20.10	24.06.2013 09:00:00
client1	IPv6 Host (AAAA)	fd00:0000:0000:0020:0000:0000:0000:0100	24.06.2013 09:00:00
dc1	Host (A)	172.16.20.10	Static
isatap	IPv6 Host (AAAA)	fd00:0000:0000:0020:0000:0000:0000:0010	Static
www	Host (A)	172.16.30.1	Static
	Alias (CNAME)	app1.example.net.	Static

#### ISATAP-Eintrag im DNS-Server

- ▶ Aktivieren Sie die ISATAP-Unterstützung im DNS-Serverdienst durch den Konsolenbefehl `dnscmd /config /globalqueryblocklist wpad`.
- ▶ Starten Sie den DNS-Serverdienst neu, damit die geänderte Konfiguration übernommen wird.

#### Konfiguration des ISATAP-Routers Edge1

Konfiguration des Dual-Stack-Routers als ISATAP-Router:

- ▶ Konfigurieren Sie mit `netsh` den Router für die Routerankündigungen sowie das Erstellen und Veröffentlichen einer Default-Route nach Subnet2:
 

```
netsh interface ipv6 set interface "subnet1" advertise=enabled
forwarding=enabled
netsh interface ipv6 add route fd00:0:0:20::/64 "subnet1" publish=yes
netsh interface ipv6 add route ::/0 "subnet2" fe80::1 publish=yes
```
- ▶ Aktivieren Sie ISATAP auf dem Router-Interface "subnet2":
 

```
netsh interface isatap set router 172.16.30.1
```
- ▶ Ermitteln Sie den Schnittstellenindex für das ISATAP-Interface:
 

```
netsh interface ipv6 show address
```

Die ISATAP-Schnittstelle sollte jetzt folgende Adresse aufweisen:

`fe80::5efe:172.16.30.1` → Index 16

- ▶ Aktivieren Sie die ISATAP-Schnittstelle für die Routerankündigungen und -weiterleitungen (hier wird an Stelle der Schnittstellenbezeichnung der eben ermittelte Index eingesetzt):
 

```
netsh interface ipv6 set interface 16 forwarding=enabled advertise=enabled
```
- ▶ Erstellen Sie ein logisches ISATAP-Subnet mit der IPv6-Adresse `fd00:0:0:100::/64`:
 

```
netsh interface ipv6 add route fd00:0:0:100::/64 16 publish=yes
```

Test der Konfiguration des App1- und Client1-Computers

Im nächsten Schritt müssen Sie sicherstellen, dass beide Computer die Routerankündigungen von Edge1 erhalten und mit dem gelieferten ISATAP-Subnet und der eigenen privaten IPv4-Adresse eine ISATAP-Adresse generieren.

- Deaktivieren Sie auf dem App1- und dem Client1-Computer die Netzwerk-Interfaces und aktivieren Sie sie anschließend wieder.
  - Führen Sie auf beiden Computern den folgenden Kommandozeilen-Befehl aus und suchen Sie im Ergebnis nach dem Eintrag des ISATAP-Adapters:

```
Ipconfig -all
```

```
Tunneladapter isatap.example.net:

Verbindungsspezifisches DNS-Suffix: example.net
Beschreibung. . . . . : Microsoft-ISATAP-Adapter
Physische Adresse . . . . . : 00-00-00-00-00-00
DHCP aktiviert. . . . . : Nein
Autokonfiguration aktiviert . . . : Ja
IPv6-Adresse. . . . . : fd00::100:0:5efe:172.16.20.100(Bevorzugt)
Verbindungslokale IPv6-Adresse . . . : fe80::100:0:5efe:172.16.20.100%12(Bevorzugt)
Standardgateway . . . . . : fe80::5efe:172.16.30.1%12

DNS-Server . . . . . . . : fd00:0:0:20::10
                           172.16.20.100
NetBIOS über TCP/IP . . . . . : Deaktiviert
```

*ipconfig auf Client1*

- ▶ Testen Sie auf Client1, können Sie über einen Browser den Verbindungsauflauf zu App1 testen, indem Sie folgende URL eingeben:

Tunneladapter isatap.example.net:

*ipconfig auf App1*

<http://app1.example.net>

<http://www.example.net>

**oder** mit dem  
Kommandozeilen-  
Befehl auf die ISATAP-  
Adresse von App1:

```
c:\>ping  
fd00::100:0:5efe  
:172.16.40.20
```

```
Ping wird ausgeführt für
fd00::100::5efe:172.16.40.20 mit 32 Bytes Daten:

Antwort von fd00::100::5efe:172.16.40.20 Zeit=2ms
Antwort von fd00::100::5efe:172.16.40.20 Zeit=1ms
Antwort von fd00::100::5efe:172.16.40.20 Zeit=1ms
Antwort von fd00::100::5efe:172.16.40.20 Zeit=1ms

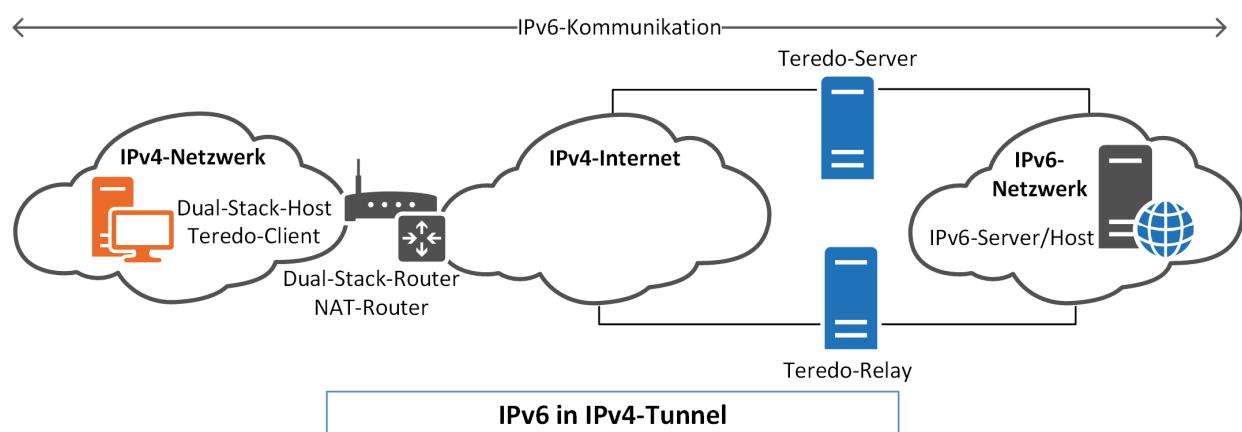
Ping-Statistik für fd00::100::5efe:172.16.40.20
Pakete: Gesendet = 4, Empfangen = 4, Verloren = 0
(0% Verlust),
Ca. Zeitangaben in Millisek.:
Minimum = 1ms, Maximum = 2ms, Mittelwert = 1ms
```

### *Traceroute zu App1*

## Teredo-Tunnel

Auch Teredo-Tunnel leiten in IPv6 gekapselte IPv4-Inhalte in ein IPv6-Netzwerk weiter. Dabei muss allerdings ein sog. Teredo-Server als Tunnelendpunkt angegeben werden. Dieser packt die vom Teredo-Client eingepackten IPv4-Pakete wieder aus und stellt sie im IPv6-Netzwerk zu. In Microsoft-Netzwerken ist hierzu ein MS-Teredo-Server vorkonfiguriert; diese Konfiguration erlaubt es, ohne eigenen Konfigurationsaufwand beispielsweise über einen SoHo-Router mit einem IPv6-Netzwerk zu kommunizieren. Es besteht jedoch grundsätzlich die Möglichkeit, einen beliebigen von verschiedenen Institutionen oder Firmen angebotenen Teredo-Server für diese Funktion in Anspruch zu nehmen. Listen mit Adressen dieser Server sind im Internet verfügbar; die Serveradresse muss dann im Teredo-Client eingetragen werden.

Ein Teredo-Relay-Server sorgt für die Aufrechterhaltung der Kommunikation (Rückverbindung).



Ein Teredo-Server muss eine öffentliche IPv4-Adresse besitzen und bietet dem Client eine Adresszuweisung sowie das automatische Host-zu-Host-Tunneling an. Teredo unterstützt NAT und bietet Clients, die sich hinter einem NAT-Router befinden, einen stabilen Tunnel in ein IPv6-Netzwerk.

Die Teredo-Clients müssen mit der Adresse des Teredo-Servers vorkonfiguriert werden. Wegen der hohen Latenzzeit bei der Einrichtung des Teredo-Tunnels sollte dieses Verfahren nur Anwendung finden, wenn andere Tunnel-Verfahren nicht realisierbar sind.

Teredo-IPv6-Adressen haben folgenden Aufbau:

Teredo-Präfix	Teredo-Server IPv4-Adresse	Flags NAT-Typ	Teredo-Client Port	Teredo-Client IPv4-Adresse
32 Bit	32 Bit	16 Bit	16 bit	32 Bit

Der Kommunikationsablauf stellt sich wie folgt dar:

- ✓ Ein Teredo-Client (Dual-Stack) verbindet sich mit dem vorkonfigurierten Teredo-Server, ermittelt den Adress-Präfix und richtet einen IPv6-in-IPv4-Tunnel ein.
- ✓ Über einen regelmäßigen Paketaustausch zwischen dem Teredo-Client und dem Teredo-Server wird die Tunnelverbindung gehalten.
- ✓ Die IPv6-Kommunikation erfolgt dann über den Teredo-Relay.
- ✓ Beide, Teredo-Server und Teredo-Relay, müssen zwischen dem IPv4-Internet und dem IPv6-Netzwerk eingerichtet werden.

Unter Linux existiert das entsprechende Tool unter dem Namen **Miredo**.

### Praxisbeispiel Teredo-Tunnel

Voraussetzung:

- ✓ Windows 7 Client
- ✓ Internet-Gateway
- ✓ Firewall-Port: UDP 3544 offen

Überprüfung der Konfiguration des Teredo-Clients:

- ▶ Öffnen Sie in Windows 7 die Konsole mit dem cmd-Befehl.
- ▶ Lassen Sie sich zunächst alle verfügbaren IPv6-Schnittstellen anzeigen. Geben Sie dazu folgenden Befehl an der Konsole ein:  
`netsh interface ipv6 show interfaces`

```
PS C:\> netsh interface ipv6 show interfaces

Idx      Met          MTU        State           Name
---  -----
 1       50  4294967295  connected    Loopback Pseudo-Interface 1
12       50         1800  disconnected  isatap.{16FA2F47-3CB7-4C1A-822F-38EAC88C30}
 11      10         1500  connected    LAN-Verbindung
```

*Verfügbare interfaces*

Hier wird ersichtlich, dass der Teredo-Adapter in der Ausgangskonfiguration nicht für den Tunnelbetrieb eingerichtet ist.

- ▶ Lassen Sie sich nun die momentane Konfiguration des Teredo-Adapters mit folgendem Konsolenbefehl anzeigen:  
`netsh interface ipv6 show teredo`

```
C:\> netsh interface ipv6 show teredo
Teredo-Parameter
-----
Typ          : Client
Servername   : teredo.ipv6.microsoft.com.
Clientaktual.-intervall : 30 Sekunden
Clientport    : unspecified
Status        : dormant
```

*Teredo Status vor Verbindungsauftbau*

<b>Typ</b>	Der Teredo-Gerätetyp wird hier als Client ausgewiesen.
<b>Servername</b>	Als Teredo-Server ist standardmäßig der Microsoft-Teredo-Server eingetragen. (Dies kann jederzeit mit dem cmd-Befehl <code>netsh interface ipv6 set teredo client &lt;teredo.example.com&gt;</code> auf einen anderen Teredo-Server angepasst werden.)
<b>Clientaktual.-intervall</b>	Der Zeitraum, nach dem ein initiiertter Tunnel bei Nichtaktivität (kein Datenfluss) beendet wird
<b>Clientport</b>	Nicht manuell zugewiesen (Standardmäßig ist hier der Port UDP 3455 aktiv.)
<b>Status</b>	Dormant (= ruhend, inaktiv)

Die Konfiguration des Teredo-Tunnels erfolgt beim Verbindungsauftbau. Durch eine Verbindungsanfrage zu einem IPv6-Host wird der Teredo-Tunnel initialisiert und automatisch eingerichtet.

Die Adressen von heise online sind über IPv4 und IPv6 erreichbar und bieten sich zum Testen an.

- Geben Sie folgenden Kommandozeilen-Befehl ein (wichtig: -6 zum Erzwingen der IPv6-Kommunikation):  
 ping -6 www.six.heise.de

```
PS C:\> ping -6 www.six.heise.de

Ping wird ausgeführt für www.six.heise.de
[2a02:2e0:3fe:1001:7777:772e:2:85] mit 32 Bytes Daten:
Antwort von 2a02:2e0:3fe:1001:7777:772e:2:85: Zeit=24ms
Antwort von 2a02:2e0:3fe:1001:7777:772e:2:85: Zeit=25ms
Antwort von 2a02:2e0:3fe:1001:7777:772e:2:85: Zeit=21ms
Antwort von 2a02:2e0:3fe:1001:7777:772e:2:85: Zeit=22ms

Ping-Statistik für 2a02:2e0:3fe:1001:7777:772e:2:85:
Pakete: Gesendet = 4, Empfangen = 4, Verloren = 0
(0% Verlust),
Ca. Zeitangaben in Millisek.:
Minimum = 21ms, Maximum = 25ms, Mittelwert = 23ms
```

#### *Test der IPv6 Konnektivität*

Beim ersten ping erhalten Sie eine Zeitüberschreitung. Dies weist auf den Vorgang der Initialisierung des Tunnels hin (und kann auch mehrere Ping-Intervalle umfassen). Die hohen Latenzen (hier: 90 ms) sind spezifisch für Teredo-Tunnel, machen sich aber in Anwendungen (z. B. Browser) für den Benutzer durch systeminterne Puffermechanismen nicht bemerkbar.

Endet das ping-Kommando mit Fehlermeldungen oder grundsätzlichen Zeitüberschreitungen, kann es sein, dass der Client den Tunnel noch nicht initialisiert und aufgebaut hat. Rufen Sie deshalb das Kommando ein zweites Mal auf.

- Sehen Sie sich die nun initiierte Teredo-Client-Konfiguration erneut auf der Konsole an:  
 netsh interface ipv6 show teredo

```
C:\> netsh interface ipv6 show teredo
Teredo-Parameter
-----
Typ : Client
Servername : teredo.ipv6.microsoft.com.
Clientaktual.-intervall : 30 Sekunden
Clientport : unspecified
Status : qualified
Clienttyp : Teredo client
Netzwerk : unmanaged
NAT : restricted
NAT-spezifisches Verhalten : UPNP: nein, Portbeibehaltung; ja
Lokale Zuordnung : 172.16.0.200:58265
Externe NAT-Zuordnung : 91.32.88.248:58265
```

#### *Teredo Status nach Verbindungsaufbau*

Der Status ist nun qualified, also „eingerichtet“. Als zusätzliche Konfigurationsparameter sehen wir die automatisch eingerichtete NAT-Konfiguration mit der Zuordnung der temporär zugewiesenen öffentlichen zur privaten IPv4-Adresse.

- Um die IPv6-Global Unique Adress des Teredo-Clients anzuzeigen, verwenden Sie den Konsolen-Befehl ipconfig -all

Tunneladapter LAN-Verbindung 2:

```
Verbindungsspezifisches DNS-Suffix: 
Beschreibung . . . . . : Teredo Tunneling Pseudo Interface
Physische Adresse . . . . . : 00-00-00-00-00-00
DHCP aktiviert . . . . . : Nein
Autokonfiguration aktiviert . . . . : Ja
IPv6-Adresse . . . . . : 2001:0:5ef5:79fd:1805:1c66:a4df:a707(Bevorzugt)
Verbindungslokale IPv6-Adresse . . : fe80:1805:1c66:a4df:a707%13(Bevorzugt)
Standardgateway . . . . . : :: 
NetBIOS über TCP/IP . . . . . : Deaktiviert
```

#### *IPv6-Adresse des Teredo Clients*

Wenn sich der Teredo-Client in einem verwalteten Netzwerk (Active Directory Domain) befindet, kann es sein, dass der Teredo-Adapter deaktiviert ist.



- Um den Adapter zu aktivieren, geben Sie an der Konsole folgenden Befehl ein:  
netsh interface ipv6 set teredo enterpriseclient

Der Teredo-Typ wird auf enterpriseclient gesetzt, der Tunnel kann wie beschrieben genutzt werden.

- Um den Pfad zum Zielhost anzuzeigen, benutzen Sie den tracert-Befehl mit dem Parameter -6  
tracert -6 www.six.heise.de  
*oder* tracert -d -6 www.six.heise.de  
(Anzeige ohne DNS-Namen)

```
C:\> tracert -6 www.six.heise.de

Routenverfolgung zu www.six.heise.de [2a02:2e0:3fe:1001:7777:772e:2:85]
über maximal 30 Hops:

 1  329 ms    91 ms    89 ms  6to4.fra1.he.net [2001:470:0:150::2]
 2  98 ms     99 ms    99 ms  gigabitethernet2-6.core1.fra.he.net [2001:470:0]
 3  89 ms     87 ms    97 ms  te3-1.c101.f.de.plusline.net [2001:7f8::3012:0:1]
 4  90 ms     93 ms   109 ms  te6-1.c13.f.de.plusline.net [2a02:2e0:1::le]
 5  88 ms     93 ms    97 ms  www.six.heise.de [2a02:2e0:3fe:1001:7777:772e:2:85]
```

#### *Traceroute über Teredo*

- Abschließend deaktivieren Sie den Teredo-Tunnel manuell in der Konsole:  
netsh interface ipv6 set teredo disable

```
C:\> netsh interface ipv6 show teredo
Teredo-Parameter
-----
Typ          : disabled
Servername   : teredo.ipv6.microsoft.com.
Clientaktual.-intervall : 30 Sekunden
Clientport    : unspecified
Status        : offline
Fehler       : keiner
```

#### *Deaktivierter Teredo Adapter*

Der Status des Tunnels ist nun von dormant zu offline gewechselt, der Tunnel ist nicht mehr aktivierbar.

- Um ihn wieder zu aktivieren, geben Sie den folgenden Konsolen-Befehl ein:  
netsh interface ipv6 set teredo enable

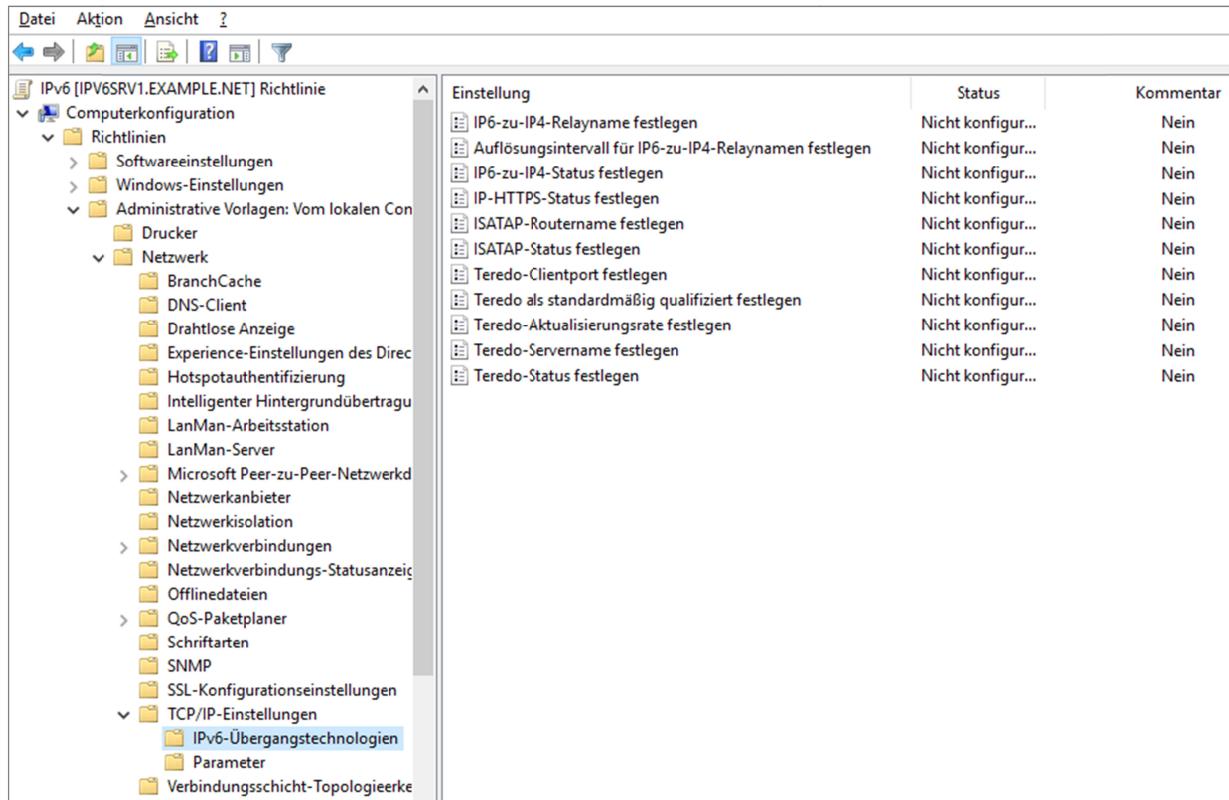
## Konfiguration über Gruppenrichtlinien

Über die lokalen Gruppenrichtlinien haben Sie die Möglichkeit, die Standardeinstellungen der Teredo-Clients anzupassen.

- Geben Sie folgenden Kommandozeilen-Befehl ein:  
gpedit

*oder* Starten Sie den lokalen Gruppenrichtlinien-Editor über den Eintrag von gpedit im Menüpunkt *Ausführen* des Windows Start-Menüs.

Im Menüpunkt *Computerkonfiguration* können Sie unter dem Knoten *Administrative Vorlagen, Netzwerk, TCP/IP Einstellungen, IPv6-Übergangstechnologien* unter anderem die Einstellungen für den Teredo-Client vornehmen.



## Gruppenrichtlinien IPv6-Übergangstechnologien

### Tunnel im Vergleich

Mechanismus	Vorteile	Nachteile
6in4	Einfache Konfiguration durch Router oder Tools. Funktioniert auch hinter NAT	Anmeldung erforderlich
6to4	Einfache Konfiguration, manuell konfigurierter Tunnel	Öffentliche IPv4-Adresse notwendig, keine NAT-Unterstützung
ISATAP	Öffentliche oder private IPv4-Adresse nutzbar, manuell konfigurierter Tunnel	
Teredo	NAT-Unterstützung automatischer Tunnel	Hohe Latenzzeit, Rechner ungeschützt im Internet, keine feste IP

## 8.5 Übersetzungsverfahren

Als **Übersetzungsverfahren** bezeichnet man Verfahren für den Übergang der Protokolle IPv4 auf IPv6, basierend auf der Übersetzung von IP-Adressen. Die IPv4-Adressen werden hierbei so umgeschrieben, dass die IPv4-Datenpakete über IPv6-Netzwerke übertragen werden können.

Bei dieser Protokollübersetzung muss darauf geachtet werden, dass hierbei die Anwendungsschicht nicht beeinflusst wird, da hier sonst die Anwendungen angepasst werden müssen.

Übersetzungsverfahren sind immer dann notwendig, wenn Geräte nicht Dual-Stack-fähig sind, also IPv4-Geräte mit IPv6-Geräten oder umgekehrt kommunizieren müssen. Dabei wurden im Laufe der Entwicklung von IPv6 verschiedene Standards entwickelt, teilweise erweitert oder wieder zurückgezogen.

### Stateless IP/ICMP Translator (SIIT)

Die zustandslose (=stateless) Protokollübersetzung von IP- und ICMP-Protokollen (**SIIT; Stateless IP/ICMP Translation**) ist in der RFC 6145 definiert und mit RFC 6791 aktualisiert. Zustandslos bedeutet, dass jedes Paket für sich übersetzbare ist.

Das Verfahren eignet sich nur für die Unicast-Kommunikation von nativen IPv4- und nativen IPv6-Hosts. IPv4-Multicast-Adressen sind mit SIIT nicht abbildbar.

### Network Address Translation – Protocol Translation (NAT – PT)

Network Address Translation – Protocol Translation (NAT-PT) ist eine Adress- und Protokollumsetzung auf Layer 3 des OSI-Modells. Am Übergang zwischen dem IPv4- und dem IPv6-Netzwerk wird ein NAT-PT-Router (Network Address Translation – Protokoll Translation) implementiert.

Das in RFC 2766 beschriebene Protokoll NAT-PT wurde mit RFC 4966 bereits 2007 zurückgezogen und gilt seitdem als veraltet.

### NAT64

Im Gegensatz zur Stateless Translation bei SIIT arbeitet NAT64 stateful. Hierbei wird eine IPv4-Adresse mit dem dafür reservierten Präfix 64:ff9b::/96 versehen. Die IPv4-Adresse wird dabei ebenfalls hexadezimal notiert. Aus **192.168.16.16** wird dann **64:ff9b::c0:a8:10:10**. Alternativ kann die IPv4-Adresse dezimal mit Punkten notiert werden. Die Übersetzung funktioniert ähnlich wie beim herkömmlichen NAT, indem der NAT64-Router die Verbindungen in einer NAT-Tabelle speichert und so die ein- und ausgehenden Verbindungen einander zuweisen kann (stateful). Ein NAT64-Router kann also IPv6-Verbindungen aus lokalen Netzen in öffentliche IPv4-Netze umsetzen. NAT64 ist in RFC 6146, „Stateful NAT64“, definiert.

### NAT66

Eine Sonderform der Übersetzung stellt NAT66 dar. Es ist in RFC 6296 (Status experimentell), „IPv6-to-IPv6 Network Prefix Translation“, definiert.

Während NAT64 eine Adress- und Protokoll-Übersetzung beschreibt, geht es bei NAT66 um den Austausch von IPv6-Präfixen. Das in RFC6296 beschriebene Verfahren spricht deshalb auch von Network Prefix Translation - NPTv6. Damit soll das Ende-zu-Ende-Prinzip trotz verschiedener Adressbereiche erhalten bleiben. Anwendungsfall wäre z.B. die 1:1-Umsetzung von Unique Local Unicast Adressen in Global Unicast Adressen. Die dabei auftretenden Probleme ähneln denen bei IPv4 NAT.

Bei IPv4 ist NAT eine, wenn auch etablierte, Notlösung. Die Notwendigkeit von NAT bei IPv6 ist daher generell umstritten. Bei der Entwicklung von IPv6 war NAT zunächst nicht vorgesehen. Die zugehörige RFC6269 datiert von 2011 und ist als experimentell eingestuft.

# 9 Sicherheit und IPv6

## In diesem Kapitel erfahren Sie

- ✓ welche Bedrohungen es gibt
- ✓ welche neuen Probleme bei IPv6 auftreten

## Voraussetzungen

- ✓ Überblick über das IPv4-Protokoll
- ✓ Grundlagen des IPv6-Protokolls
- ✓ Adressaufbau von IPv6
- ✓ Grundkenntnisse Verschlüsselung

## 9.1 Sicherheitsprobleme und Probleme bei IPv6

### Allgemeines über Sicherheitsprobleme

Auch die Einführung von IPv6 und die langsame Etablierung des Protokolls ändern nichts an der Tatsache, dass das Internet sicherheitstechnische Schwachstellen bietet. Die Absicht, die Erneuerung des Protokolls auch dazu zu nutzen, die Datenübertragung mittels in IPv6 integrierter Sicherheitsmechanismen weniger angreifbar zu machen, wurde nicht erfüllt. Vielmehr bieten einige Protokolländerungen Raum für verstärkte oder neue Angriffsszenarien.

Weiterhin gilt, dass sich im Bereich der Bedrohungen und Gefahren typische Szenarien und Kategorien etabliert haben, die die heutige Bedrohungslage in Unternehmen und für Privatanwender darstellen. IPv6 schützt weder vor Fehlkonfigurationen von Diensten und Anwendungen (vielmehr erhöht sich die Chance solcher Fehler aufgrund der immensen Komplexität einiger Bestandteile von IPv6), noch vor dem Versagen von Sicherheitsrichtlinien (policies) oder vor menschlichem Versagen und persönlichen Schwächen (social engineering).

Auch heute ist die Liste der Angriffsszenarien lang. In der Folge finden Sie eine Aufstellung der aktuellen Bedrohungen durch Gefahren aus dem Internet. Diese sind unabhängig vom verwendeten Protokollstapel (IPv4/IPv6) vorhanden und werden aktiv genutzt.

- ✓ Drive-by-Downloads von Schadsoftware
- ✓ Trojaner und Würmer
- ✓ Angriffe auf Datenbanken per SQL-Injection
- ✓ Angriffe auf Webseiten mittels Cross Site Scripting (XSS)
- ✓ Viren-Baukästen/Exploits Kits
- ✓ Botnetze
- ✓ Denial-of-Service-Attacken
- ✓ Phishing
- ✓ Datendiebstahl und Datenverluste
- ✓ Rogueware/Scareware
- ✓ SPAM

## Neue Gefährdungen bei IPv6

Die hauptsächliche Gefährdung beim Einsatz von IPv6 ergibt sich daraus, dass jedes Gerät nun mit einer eindeutigen Adresse ausgestattet werden kann und damit ein vollwertiger, jederzeit identifizierbarer Teil des Internet wird. Sofern keine Maßnahmen wie z. B. Firewalls eingesetzt werden, ist das Gerät damit aus dem Internet direkt erreichbar. Dies ist bisher in der Regel Servern und exponierten Rechnern größerer Behörden- oder Firmenstrukturen vorbehalten, die eine feste IPv4-Adresse zugewiesen bekommen und auch heute schon besonders geschützt werden. Mit dem Einsatz von IPv6 wird jedes einzelne Gerät direkt adressierbar und ist nicht mehr, wie heute noch vielfach üblich, vom Internet abgetrennt, indem es in einem privaten Netzwerk hinter einem NAT-Router verborgen und damit grundlegend geschützt ist. Dadurch ergeben sich viele neue Möglichkeiten, aber auch neue Gefahren, da potenziell jedes einzelne Gerät nun individuell in die Lage versetzt werden muss, sich gegen Angriffe von außen selbst zu schützen.

Eine Firewall ist hierbei ein probates Mittel. Inwieweit jedoch Paketfiltersysteme, die mit IPv6 umgehen können, auf einem Gerät vorhanden und aktiviert sind, liegt oft nicht in der Entscheidung des Administrators/ Benutzers. Gerade mobile Geräte (Smartphones/Touchpads) besitzen oft keine Paketfilterstrukturen, oder, wenn diese vorhanden sind, sind sie standardmäßig nicht aktiv und durch den „normalen“ Nutzer auch nur schwer und manchmal nur unter Anwendung nicht immer einwandfrei als „legal“ zu bezeichnende Maßnahmen einzurichten.

**Die Aktivierung der Privacy Extensions oder die Installation einer Firewall ist erst nach einem vermeintlich illegalen Jailbreak bzw. Rooten möglich!**



Zudem sind insbesondere Smartphones unbemerkt in fremden, oft ungeschützten WLAN- oder UMTS-Netzen eingebucht, wodurch eine zusätzliche Gefahrenlage für die Unternehmen besteht. Sie sind dort möglicherweise weltweit über IPv6 erreichbar und angreifbar.

## 9.2 Die Privatsphäre

Anonymität, also der Schutz der Privatsphäre, ist eines der wichtigsten Güter nicht nur des Internets. Leider trägt der etablierte Identifikationsmechanismus von IPv6 dazu bei, das allgemeine Gefährdungspotenzial hier deutlich zu erhöhen.

### Wiedererkennbare Host-ID

Mithilfe des modifizierten **EUI-64** (64-Bit Extended Unique Identifier)-Formats wird aus der Layer 2-Adresse (MAC) der Hostanteil der IPv6-Adresse mittels „Stretching“ gewonnen. Wie bereits beschrieben, führt dies zu der Tatsache, dass sich damit die Hosts – insbesondere mobile Hosts – weltweit und unabhängig vom Netzwerkanteil (Netz-ID) anhand dieses EUI-64-Adresssteils (Host-ID) identifizieren lassen. Anders als unter IPv4, wo ein Standortwechsel – grob gesprochen – die Vergabe einer neuen IP-Adresse und damit ggf. eine erneute Anonymität im Netz nach sich zieht, gilt unter IPv6, dass, wenn ein Gerät einmal penetriert ist, auch der Wechsel in ein anderes Netzsegment auf dieser Ebene nicht mehr vor ungewollter Identifikation und deren potenziellen Folgen schützt.

Dieses Problem kann dadurch umgangen werden, dass statt des EUI-64-Formats eine in regelmäßigen Abständen vom Gerät zufällig generierte Host-ID zum Einsatz kommt, die nur für einen bestimmten Zeitraum gültig ist und immer wieder automatisch geändert wird. Dieser Mechanismus wird als **Privacy Extension** bezeichnet, das dabei vorgesehene Vorgehen ist in RFC 4941 beschrieben.

## Privacy Extensions

Um die Verfolgung der Benutzer zu erschweren, führte man mit der RFC 4941 die IPv6 Privacy Extensions ein. Privacy Extensions erzeugen einerseits einen zufällig generierten statischen, andererseits einen zusätzlichen regelmäßig wechselnden Interface Identifier. Somit wird die Identifikation über die IPv6-Adresse deutlich erschwert. Die wechselnden Adressen haben nur eine begrenzte Zeit Gültigkeit. Moderne Betriebssysteme unterstützen diese Funktion, bei Microsoft sind Privacy Extensions ab Vista, Server 2008 standardmäßig aktiv. In den Servervarianten wird nur der statische, in den Clientbetriebssystemen zusätzlich der wechselnde Host-Identifier verwendet.

```
Beschreibung. . . . . : Broadcom NetXtreme 57xx-Gigabit-Controller
Physikalische Adresse . . . . . : 00-1D-09-DF-7A-8F
DHCP aktiviert. . . . . : Ja
Autokonfiguration aktiviert . . . . . : Ja
IPv6-Adresse. . . . . : 2001:db3:300:8d45:75d1:f3d9:9175:bea7
(Bevorzugt)
Temporäre IPv6-Adresse. . . . . : 2001:db3:300:8d45:b848:8b31:53fc:f05c
(Bevorzugt)
Verbindungslokale IPv6-Adresse . . . : fe80::75d1:f3d9:9175:bea7%11(Bevorzugt)
```

*Auszug aus ipconfig: Die (dauerhafte) verbindungslokale IPv6-Adresse ist nicht aus der MAC-Adresse abgeleitet, die temporäre IPv6-Adresse hat einen anderen Host-Identifier, der regelmäßig wechselt.*

In Linux-Systemen müssen Privacy Extensions vom Systemverwalter meist über eine *root shell* mit dem folgenden Kommando aktiviert werden:

- ▶ `sysctl -w net.ipv6.conf.all.use_tempaddr=2`

Dauerhaft kann dies in der Datei »/etc/sysctl.conf« durchgeführt werden.

Googles Smartphone-Betriebssystem **Android** setzt auf Linux, das zufällige und wechselnde IPv6-Adressen erzeugen kann. Allerdings hat Google die dafür nötigen Einstellungen nicht aktiviert, sodass bislang jede Android-Version ohne die Privatsphäre-schützenden IPv6-Adressen auskommen muss. Diese lassen sich nicht einfach einschalten, denn die Mobilfunk-Provider und Handy-Hersteller verhindern den dafür nötigen Root-Zugang. Zwei Befehle würden genügen, und ein gerootetes Android surft über die wechselnden und nicht aus der Hardware abgeleiteten IPv6-Adressen. Wie auch auf iPhones bleibt nur der Weg über das nachträgliche Freischalten des Root-Zugangs oder über die Installation von **Custom-ROMs**: Mit dem für solche Verwaltungsaufgaben nötigen Root-Benutzer lassen sich dann wieder die **Sysctl**-Werte setzen, die die Privacy Extensions für IPv6 aktivieren. Steht auf dem Telefon das Kommando `sysctl` bereit, könnte dies durch folgende Befehle erreicht werden:

- ▶ `su`
- ▶ `sysctl -w net.ipv6.conf.default.use_tempaddr=2`
- ▶ `sysctl -w net.ipv6.conf.all.use_tempaddr=2`

Nach einem Neustart vergisst Android diese Einstellungen jedoch wieder. Sie können die beiden Befehle allerdings in eine Datei namens **/data/local/userinit.sh** schreiben. Existiert diese Datei, führt z. B. das beliebte Custom-ROM **Cyanogenmod** die darin aufgelisteten Befehle beim Systemstart aus.

Apples Betriebssystem Mac OS X kann auch über die Privacy Extensions ermittelte IPv6-Adressen erzeugen und einsetzen. Sie sind allerdings versionsabhängig nicht von Haus aus aktiviert. Die Privacy Extensions lassen sich im Terminal mit dem Befehl

- ▶ `sudo sysctl -w net.inet6.ip6.use_tempaddr=1`

aktivieren. Das vorangestellte `sudo` fragt nach dem Passwort und führt dann den Befehl `sysctl` mit Administrator-Rechten aus. Dauerhaft wird dies durch folgenden Befehl erreicht:

- ▶ `sudo sh -c 'echo net.inet6.ip6.use_tempaddr=1 >> /etc/sysctl.conf'`

Bei Apples iOS für iPhones/iPads gibt es verschiedene Möglichkeiten. Bis zur Version 4.3 waren auch dort die Privacy Extensions abgeschaltet. Erst ein Update aktiviert die Erweiterung. Im Unterschied zu Mac OS X steht aber auf den Mobilbetriebssystemen für iPhone und iPad kein vom Hersteller vorgesehener Weg offen, die Privacy Extensions zu aktivieren oder abzuschalten. Wollen Sie auf Geräten mit der IOS-Version kleiner als 4.3 die Privacy Extensions einschalten, können Sie das nur, wenn Sie einen Administrator-Zugang zum Betriebssystem haben (Jailbreak): In diesem Fall reicht der Aufruf von

► `sudo sysctl -w net.inet6.ip6.use_tempaddr=1`

oder der Eintrag

► `net.inet6.ip6.use_tempaddr=1`

in die Datei `/etc/sysctl.conf`. Dazu starten Sie im Terminal einen Editor mit root-Rechten, beispielsweise mit

► `sudo pico /etc/sysctl.conf`

und fügen die Zeile am Ende der Datei an. Nach einem Neustart der WLAN-Schnittstelle respektive einem Neustart des Geräts sollte die IPv6-Adresse nun mit den Privacy Extensions zu sehen sein.

Der durch die Privacy Extensions erzwungene Wechsel des Host-Identifiers bringt andererseits neue Probleme mit sich. Bei jedem Systemstart, bei jedem Neuaktivieren der Netzwerkkarte oder des Protokollstapels (oder aber mindestens einmal pro Tag), wird ein neuer zufälliger Host-Identifier erzeugt. Bei Viren- oder Wurmbefall der Systeme, den der Administrator erst am nächsten Tag entdeckt, kann er meist die protokollierten IP-Adressen in seinen Logprotokollen (Firewalls) den Systemen nicht mehr konkret zuordnen, weil der Adresswechsel dort nicht verzeichnet wird.

Alle aktuellen Smartphone-Betriebssysteme haben inzwischen Privacy-Extensions standardmäßig aktiviert.



## Zufällige Präfixe

Die beschriebenen Privacy Extensions sorgen dafür, dass das einzelne Gerät nicht über den Host-Identifier identifiziert werden kann. Bleibt die Netzkennung die gleiche, weiß man aber, dass das Gerät die Verbindung aus dem immer gleichen Netz aufbaut. Damit lässt sich der Anschluss auf einen Zugangsprovider oder eine Firma zurückverfolgen. Die Daten dazu sind öffentlich mit dem Tool **Whois** oder entsprechenden Webformularen auch im Browser abfragbar. Das geht unter anderem direkt bei der für Europa zuständigen Vergabestelle RIPE ([www.ripe.net](http://www.ripe.net)).

Dieses Verhalten würde bei IPv4 einer festen IP-Adresse bei DSL bzw. eines einzelnen IPv4-(Sub-)Netzes bei Firmen entsprechen. In vielen Fällen ist das so gewollt, da man mit einer festen Adresse Dienste nach extern besser zur Verfügung stellen kann. Legt man dennoch Wert auf einen wechselnden IPv6-Netzpräfix, kann der ambitionierte Heimanwender oder der Administrator der Firma dafür Sorge tragen, dass der Präfix sich ebenfalls regelmäßig ändert. Im Kapitel zum Adressaufbau wurde beschrieben, dass in der Regel vom Provider ein Netz mit der Länge /48 oder /56 zugewiesen wird. Da davon meist nur wenige /64-Netze benötigt werden, kann mit geeigneter Router-Software für wechselnde Präfixe gesorgt werden. Bei einer /56-Zuteilung wären das immer noch  $2^8 = 256$  Möglichkeiten, bei /48 schon  $2^{16} = 65536$ . Ob das dem Schutz der Privatsphäre wirklich dient, ist jedoch umstritten. Bei der Registry ist das zugewiesene Präfix registriert, und daran ändert sich durch das Verfahren nichts (siehe auch <http://heise.de/-1445607>).

## Handhabung bei Zugangs providern

Die großen Zugangsprovider, sofern sie IPv6 unterstützen, teilen ihren Kunden wechselnde Präfixe zu, allerdings nur wenn der Zugangsrouter neu gestartet wurde. Eine tägliche Zwangstrennung gibt es bei den aktuellen Tarifen nicht mehr. Der Kunde bekommt meist ein /56 Präfix zugeteilt, das mit geeigneten Endgeräten auf mehrere /64 Netze verteilt werden kann. Die meisten Heimrouten werden freilich nur ein /64-Netz von den 256 möglichen nutzen. Von den zunächst vorgesehenen festen Präfixen wurde angesichts der Datenschutzzdiskussion Abstand genommen. Wer einen festen Präfix benötigt, sollte sich eher an kleinere Anbieter wenden, die die IPv6 in dieser Form anbieten. Dort wird in der Regel ein Eintrag auf den Kunden in der RIPE-Datenbank erstellt.

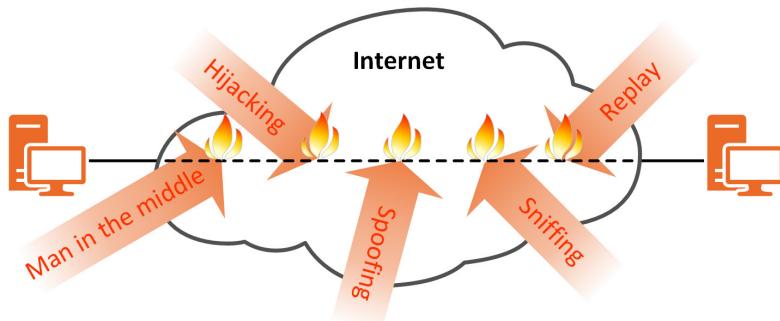
## Webhosting

Im Webhosting-Bereich machen wechselnde Präfixe und wechselnde Host-IDs wenig Sinn, da auf den Servern Dienste laufen, die jederzeit erreichbar sein müssen. Hier erhält der Kunde meist eine /64-Zuteilung für jeden dedizierten Server.

## 9.3 Eingebaute Sicherheit

### IPsec bei IPv4

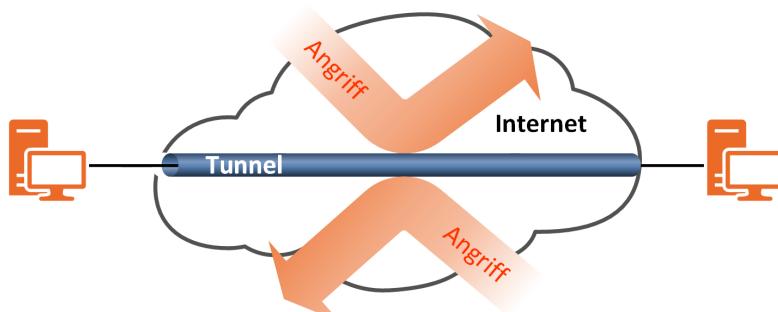
IPSec (IP Security) wird als Framework in den IPv4-Stack implementiert. Damit können die üblichen Sicherheitsansprüche an die Kommunikation (Authentizität, Vertraulichkeit, Integrität) erfüllt werden. IPSec wird im Gegensatz zu SSL in der Internetschicht des DoD-Modells (Vermittlungsschicht des OSI-Modells) realisiert. Als wesentliche Protokolle werden **Authentication Header (AH)**, **Encapsulated Security Payload (ESP)** sowie **Internet Key Exchange (IKE)** zum Austausch der Schlüssel verwendet. Mithilfe von zwei lokalen Datenbanken, der SPD (Security Policy Database) und der SAD (Security Association Database), werden die Ansprüche und Methoden definiert.



Ungesicherte Verbindung zweier Netzwerke über das Internet

Das Internet-Key-Exchange-Protokoll IKEv1 dient der automatischen Schlüsselverwaltung für IPsec. Es verwendet den Diffie-Hellman-Schlüsselaustausch (DH) für einen sicheren Austausch von Schlüsseln über ein unsicheres Rechnernetz. IKE wird über UDP transportiert und nutzt standardmäßig den Port 500 als Quell- und Ziel-Port. Wird IKE und IPsec jedoch hinter einer NAT-Firewall betrieben, wird von den meisten IPsec-Implementierungen in diesem Fall UDP-Port 4500 verwendet.

IPSec kann die zu übertragenden IP-Datagramme mittels AH (Authentication Header) und/oder ESP (Encapsulating Security Payload) schützen. AH dient der Authentisierung, des Schutzes der Adressinformation und sichert die IP-Kopfinformationen mittels Hash-Verfahren. Dadurch kann die Integrität der Protokoll-Header gegen Manipulationen gesichert werden. Das führt zu dem Problem, dass AH nicht über NAT-Router geführt werden können. AH schützt weiterhin auch nicht die Nutzinformationen des IP-Datagramms. Für diesen Zweck kann ESP eingesetzt werden. IPsec kann im Tunnelmodus auf Routern den Verkehr durch ein Transitnetz (Tunnelmodus) oder im Ende-zu-Ende-Modus den Verkehr zwischen zwei Hosts (Transportmodus) schützen.



Gesicherte (Tunnel-)Verbindung über das Internet

## IPsec bei IPv6

IPSec wurde bei IPv6 bereits im Protokolldesign berücksichtigt. Dazu werden die Erweiterungs-Kopfdaten (Extension Headers) verwendet. Im Kopf des IP-Datagramms sind zu diesem Zweck zwei Erweiterungsheader vorgesehen:

Authentication Header (AH)	Typ 51	RFC 4302
Encapsulating Security Payload (ESP)	Typ 50	RFC 4303

Auch hier kann AH und ESP gleichzeitig eingesetzt werden und der Verkehr durch Tunnel oder Ende-zu-Ende-Modus geschützt werden. Durch die Einführung von IKEv2 hat man insbesondere die Erkennung und Behandlung von Timeouts/Tunnelabbrüchen, sowie die NAT-T (NAT Traversal Transport) verbessert.

Bevor AH oder ESP angewendet werden kann, müssen die beteiligten Partner ihre Schlüssel austauschen, die für die Absicherung verwendet werden. Hier können statische Schlüssel (manuell) auf beiden Kommunikationspartnern vereinbart werden. In den meisten Kommunikationssituationen ist dies aber nicht praktikabel. Deswegen wird der Schlüsselaustausch meist mithilfe des Internet Key Exchange (IKE)-Protokolls realisiert. IKE wird in der RFC 2049 definiert, das Nachfolgeprotokoll IKEv2 in der RFC 4306 beschrieben. Es ist die technische Umsetzung des **Internet Security Association and Key Management Protocol (ISAKMP)**. ISAKMP definiert die Regeln und Wege für die Authentifizierung von Kommunikationspartnern, Erstellung und Verwaltung von Sicherheitszuordnungen (Security Association; SA), Schlüsselerzeugung und -verwaltung sowie Maßnahmen gegen Replay-Attacken.

IKE dient der automatischen Schlüsselerzeugung, der sicheren Schlüsselübertragung und der Schlüsselerneuerung. IKE bedient sich dabei der beiden Methoden OAKLEY und SKEME und nutzt dazu den UDP-Port 500. Die Schlüsselübertragung wird mithilfe der Diffie-Hellman (DH)-Algorithmen durchgeführt. Bei der Authentisierung werden Schlüsselpaare (Preshared Keys, PSK) oder X.509-Zertifikate eingesetzt.

Bei der Verwendung von Microsoft-Betriebssystemen kann hier auch das Kerberos-Protokoll eingesetzt werden, wenn die Partner einer AD-Domäne angehören.

IKE arbeitet in zwei Phasen:

- ✓ Aushandlung einer Security Association (SA, Vertrauensstellung) über den Hauptmodus (Main Mode) oder Aggressiv-Modus (Aggressive Mode), wobei der Main Mode Vorrang hat.
- ✓ Erzeugung einer SA im Schnellmodus (Quick Mode)

Eine Security Association (SA) ist eine Vereinbarung zwischen den beiden kommunizierenden Seiten und besteht aus den Punkten:

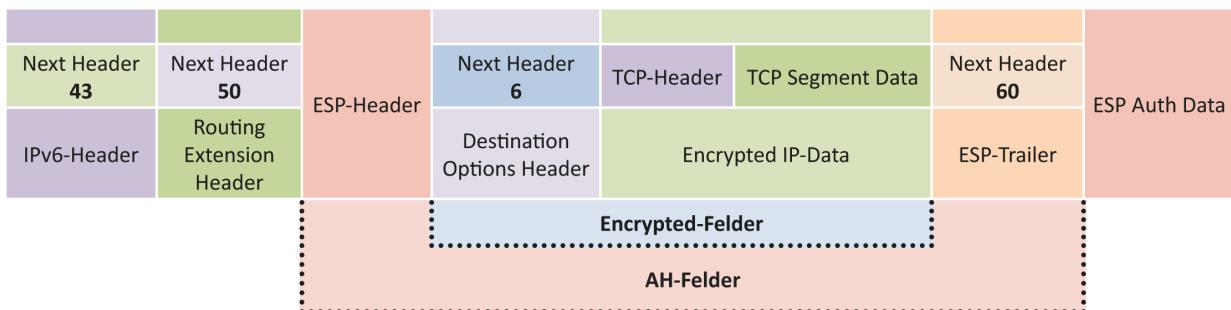
- ✓ Identifikation (entweder per PSK oder Zertifikat)
- ✓ Festlegung des zu verwendenden Schlüsselalgorithmus für die IPsec-Verbindung
- ✓ von welchem (IP-)Netz die IPsec-Verbindung erfolgt
- ✓ zu welchem (IP-)Netz die Verbindung bestehen soll
- ✓ Zeiträume, in denen eine erneute Authentisierung erforderlich ist
- ✓ Zeitraum, nach dem der IPsec-Schlüssel erneuert werden muss

IKE ist für Partner mit wechselnden IP-Adressen (Dynamische Zugänge) weniger geeignet, da die Verbindungsabbrüche durch IP-Adresswechsel schlecht erkannt werden (Dead Peer Detection).

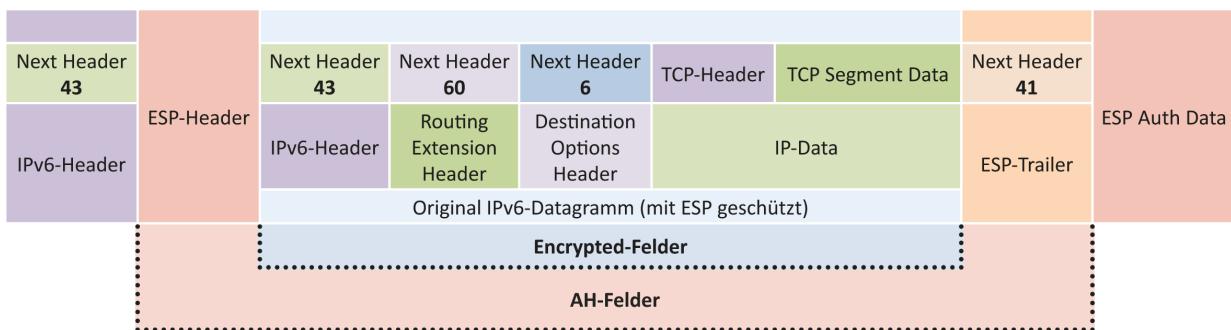
Mit IKEv2 wurde diesem Rechnung getragen. IKEv2 benötigt für die Einrichtung einer Sicherheitszuordnung (SA) nur noch 4 UDP-Nachrichten.



Original IPv6-Datagramm-Format (mit Routing Extension Header und Destination Options Header)



IPv6-Datagramm-Format - IPSec im Transport Modus



IPv6-ESP-Datagramm-Format - IPSec im Tunnelmodus

## 9.4 Wegfall von NAT

### Warum NAT bei IPv4

Die Adressknappheit öffentlicher IPv4-Adressen zwingt die Internet-Nutzer, ihre internen (privaten) IP-Adressen mittels NAT-Routern auf öffentliche IP-Adressen umzuschreiben (vgl. Kap 2). Als Nebeneffekt wird für den im Internet befindlichen Host die interne Struktur verdeckt, weil er nur die öffentliche Adresse des Grenzrouters sieht. Das schafft zusätzliche Sicherheit, bringt aber auch Probleme mit sich. Da die Ports und IPs durch den NAT-Router umgeschrieben werden, lassen sich somit keine Verbindungen schaffen, die diese Originaladressen/Ports benötigen. Insbesondere bei der Verwendung von L2TP/IPSec gibt es damit Probleme.

Viele Firmen kombinieren die Strategien NAT/Firewall, um mehr Sicherheit durch Verdeckung der internen Infrastruktur zu schaffen. In Situationen, in denen Ende-zu-Ende-Beziehungen aufgebaut werden müssen, kann dies über zwei NAT-Strukturen nicht funktionieren, sodass die NATs teilweise geöffnet werden müssen (Portforwarding). Andere Lösungen bestehen in der Anwendung von VPN-Netzen zwischen diesen Kommunikationspartnern.

## Warum kein NAT bei IPv6

IPv6 basiert auf dem Ansatz, dass grundsätzlich eine durchgehende Ende-zu-Ende-Beziehung möglich ist. Damit ist NAT zunächst überflüssig. Auch die Adressknappheit entfällt als Argument für eine NAT-Lösung. Somit stellt auch IPSec kein Problem mehr dar. Das Problem des Sicherheitsansatzes ist nun, dass der Grenzrouter eine ungehinderte Ende-zu-Ende-Kommunikation zulässt. Beim Einsatz von globalen Adressen ist somit jeder Knoten im internen Netzwerk direkt erreichbar. In Zukunft spielen deshalb Paketfilter-Firewalls eine wichtige Rolle. Hier sind die Hersteller der Router/Firewall-Strukturen gefordert. Bei vielen Systemen zeigt sich, dass zum Teil bei vorhandenem IPv6 Stack keine Paketfilter für dieses Protokoll existieren, diese vom Hersteller aus deaktiviert oder unzureichend implementiert sind.

## Firewall

Die Aufgabe einer Firewall ist zunächst, den durchquerenden Datenverkehr anhand von Filterregeln (Access-Lists) zu gestatten oder zu verbieten. Neben den Paketfiltern bieten moderne Firewalls die Möglichkeit, anhand von Statusflags den ein- und ausgehenden Verkehr zu filtern. Die Stateful Inspection Firewalls (SPI – Dynamische Paketfilterung) können auch Dienste filtern, welche mit einfachen Paketfiltern (Access-Lists) nicht behandelt werden können.

## Paketfilter statt NAT

Den Regelwerken der Paketfilter auf den Grenzroutern wie auch auf den einzelnen Hosts kommt damit eine wesentliche Bedeutung zu. Bei modernen Paketfiltern kann mit zustandsorientierten Regeln die gleiche einfache Sicherheit erreicht werden, die auch mit NAT-Filtern gewährleistet wurde. Der Initialverkehr darf dabei nur vom Client in Richtung Server gehen.

NAT selbst bietet ja nicht wirklich einen Schutz, da es lediglich die IP-Adressen versteckt und Zugriffe nicht unterbindet. Mit Source-Routing und Kenntnis/Erraten der internen IP-Adressen baut ein Angreifer sogar Verbindungen zu Systemen hinter einem NAT-Router auf, sofern ein vorhandener Paketfilter dies nicht unterbindet. Nicht alle Paketfilter unterstützen solche „stateful-inspection“-Methoden. Linux-Kernel können erst seit Version 2.6.20 (aus dem Jahr 2007) IPv6 zustandsorientiert filtern. Ältere Distributionen lassen sich deshalb nicht als IPv6-Firewall einsetzen.

Das nachfolgende Beispiel einer Linux ip6tables stellt sicher, dass nur interne Hosts mit der Kommunikation zu Hosts im Internet aufbauen dürfen, Hosts aus dem Internet dagegen können keine Verbindungen zu den internen Hosts aufbauen.

Durch vier Regeln wird sichergestellt, dass alle Pakete,

- ✓ die zu **bereits aufgebauten Verbindungen** gehören ("ESTABLISHED") oder
- ✓ **Fehlermeldungen** für diese Verbindungen enthalten ("RELATED"),
- ✓ sowie **neue Verbindungen** ("NEW")

zugelassen werden, wenn sie aus dem internen LAN stammen. Bei einigen Diensten, wie z. B. FTP-Servern, die dynamische Ports zum internen Host benötigen, können mit Modulen, wie z. B. `nf_conntrack_ftp`, die Fähigkeiten des Paketfilters erweitert werden. Dazu muss der Systemverwalter die entsprechenden Module laden, z. B. mit "`modprobe nf_conntrack_ftp`".

```

01 LAN=eth0
02 INTERNET=ppp0
03 IPT=/sbin/ip6tables
04
05 $IPT -P FORWARD DROP
06 $IPT -F FORWARD
07 $IPT -A FORWARD -m state -state ESTABLISHED,RELATED -j ACCEPT
08 $IPT -A FORWARD -i $LAN -o $INTERNET -m state -state NEW -j ACCEPT

```

*Listing: iptables*

Das Listing zeigt eine einfache Firewallkonfiguration mit ip6tables:

- ✓ In der Zeile 05 wird in der Table FORWARD das Verwerfen von Paketen als Standard eingestellt.
- ✓ Zeile 06 löscht eventuell vorhandene Regeln.
- ✓ Zeile 07 lässt Pakete, welche (statusbezogen) für eingerichtete Verbindungen gelten, oder Pakete, welche in Beziehung zu einer bestehenden Verbindung (related) sind, passieren.
- ✓ Zeile 08 akzeptiert Pakete, die vom internen LAN kommen und in Richtung Internet-Adapter transportiert werden.

Mit diesen Regeln lässt die Firewall ausgehenden Internet-Verkehr zu, eingehenden nur dann, wenn dazu schon eine Verbindung existiert (established), z. B. die Antwort auf eine Anfrage an einen Webserver. Eingehende Verbindungen ohne Entsprechung werden blockiert.

Bei IPv6 wird es entsprechend schwieriger, die Firewall selbst zu schützen. Die IPv4-Erreichbarkeit kann mittels der Anweisungen in der INPUT- und OUTPUT-Kette alle Pakete an die Firewall verwerfen. Dies funktioniert bei IPv6 nicht. Grund dafür ist das nicht vorhandene ARP. Dieses wurde durch NDP ersetzt, das auf Schicht 3 arbeitet. Dagegen arbeitet ARP auf Schicht 2. Die Filter in IPv6 ignorieren die Schicht 2-Anfragen und lassen diese ungehindert passieren. Die MAC-Adressenauflösung funktioniert hier unabhängig von den iptables-Regeln. Somit kann man in der IPv6-Regel nicht alle Pakete von und zu dem System verwerfen, weil dies auch die ICMPv6-Nachrichten einschließen würde.

Bei IPv6 erfolgt die MAC-Adressenauflösung mit ICMPv6 auf OSI-Layer 3 (NDP). Verwerfen die ip6tables-Regeln sämtliche Pakete, dann schließt das auch die ICMPv6-Nachrichten ein, und die MAC-Adressenauflösung der Firewall funktioniert nicht mehr. Somit müssen nun in den IPv6-Regeln mindestens die ICMPv6-Nachrichten zugelassen werden, aber aus Sicherheitsgründen auch wieder nicht alle Typen. Mit der RFC 4890 hat man eine detaillierte Anleitung, die beim Bilden der IPv6-Regelketten herangezogen werden kann.

## 9.5 Sicheres DHCPv6

### Warum DHCPv6?

Die Stateless Address Autoconfiguration konfiguriert verbindungslokale und weitere nicht verbindungslokale Adressen durch das Austauschen von Router Solicitation- und Router Advertisement-Nachrichten mit den benachbarten Routern.

Bei der Stateful Address Autoconfiguration wird für die Zuweisung von nicht verbindungslokalen Adressen ein DHCPv6-Server benötigt. Auch bei statusfreier Adresszuweisung kann ein Host den DHCPv6-Dienst für Optionen in Anspruch nehmen. Dafür müssen im Router die Flags **M** (Managed Address Configuration) und **O** (Other Configuration) gesetzt sein (vgl. Stateless/Stateful DHCP in Kapitel 5). In Active Directory-Umgebungen gilt auch für IPv6, dass DHCP-Dienste im AD zuvor autorisiert werden müssen, bevor sie ihre Arbeit verrichten können.

## Änderungen im Protokoll gegenüber IPv4

Mithilfe von DHCP Unique Identifier (DUID) und Identity Association Identifier (IAID) können DHCP-Clients mit einer Reservierung im Bereich versehen werden, die einen zusätzlichen Schutz gegen DHCP-Angriffe bietet.

Ethernet-Adapter LAN-Verbindung:

```
Verbindungsspezifisches DNS-Suffix:
Beschreibung. . . . . : Broadcom NetXtreme 57xx-Gigabit
Physische Adresse . . . . . : 00-1D-09-DE-B8-EC
DHCP aktiviert. . . . . : Nein
Autokonfiguration aktiviert . . . . . : Ja
Verbindungslokale IPv6-Adresse . . . : fe80::21d:9ff:fedec%12 (Bevorzugt)
IPv4-Adresse . . . . . : 172.16.20.100 (Bevorzugt)
Subnetzmaske . . . . . : 255.255.255.0
Standardgateway . . . . . : fe80::5e49:79ff:fe7a:a1fb%12
DHCP-Server . . . . . : 172.16.20.1
DHCPv6-IAID . . . . . : 268442889
DHCPv6-Client-DUID. . . . . : 00-01-00-01-1D-3B-C1-2A-00-1D-09-DE-B8-EC
DNS-Server . . . . . : fe80::1d1f:c5db:b9c4:fd81%12
192.168.55.1
NetBIOS über TCP/IP . . . . . : Aktiviert
```

DUID und IAIID über ipconfig ausgelesen

Bei IPv4 war die Reservierung mit der MAC-Adresse verbunden. Die DUID und IAIID stehen auch nach einem Tausch der Netzwerkkarten oder einem Neustart von Maschinen zur Verfügung.

Die bei IPv4 noch möglichen Attacken auf DHCP-Server durch Aufbrauchen aller verfügbaren DHCP-Adressen (DHCP-Starvation) oder Vortäuschen eines DHCP-Servers (DHCP-Spoofing) werden durch die zusätzlichen Möglichkeiten bei IPv6 deutlich erschwert.

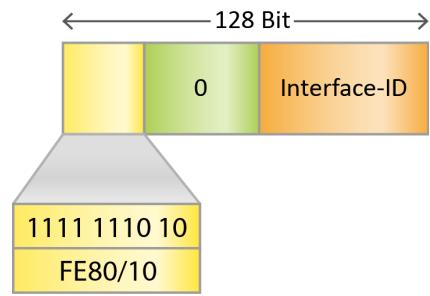
DHCPv6-Nachrichten lassen sich durch Authentisierung schützen (OPTION\_AUTH). Dabei können zwei Verfahren eingesetzt werden (RFC 3118):

- ✓ Delayed Authentication Protocol
- ✓ Reconfigure Key Authentication Protocol

Dies beinhaltet neben dem Replay-Schutz die Identifikation des DHCP-Servers durch den DHCP-Client mittels Token (HMAC-MD5). Zusätzlich wird für Delayed Authentication jeweils ein Schlüssel (Key) auf jedem DHCP-Client benötigt, welcher initial erzeugt wird und dem DHCP-Server bekannt sein muss.

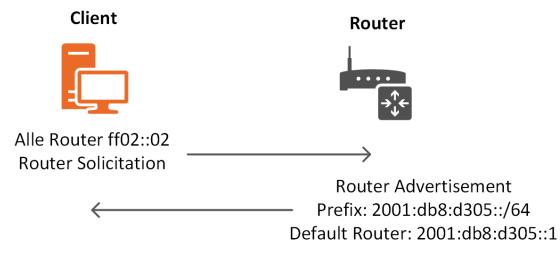
## Autokonfiguration

Eine Eigenschaft des IPv6-Protokolls, die mögliche zustandslose, automatische Konfiguration der IPv6-Adresse durch die einzelnen Systeme (Stateless Automatic Autoconfiguration, SLAAC) wurde bereits beschrieben. Damit muss kein DHCP-Server mehr eingesetzt werden, ein weiterer „Single Point of Failure“ ist eliminiert. Wie schon dargestellt, erzeugen die Systeme aus ihrer MAC-Adresse einen Identifier. Mit diesem wird die Link-Local-Adresse erzeugt, indem sie den Identifier an das Netz „fe80/64“ anhängen. Schon mit der Adresse können Hosts im lokalen Netz kommunizieren. Dieser Automatismus stellt sicher, dass wie unter IPv4 mit APIPA (Windows) oder Avahi (Linux), unter dem Namen „Verbindungslokale IP-Adresse“ via Zeroconf (welches adäquat zu IPv4 APIPA funktioniert) eine Kommunikation im lokalen Segment möglich ist.



Die Link-Local-Adresse kann der Rechner selbst und ohne zusätzliche Hilfe erzeugen.

Per **Router Solicitation** an die Router-Multicast-Adresse werden nun alle Router im selben Netz aufgefordert, sich zu melden. Diese übermitteln dabei das globale Präfix. Der anfragende Host errechnet sich daraus eine globale IP-Adresse, indem es den Identifier an das globale Präfix anhängt. Antworten mehrere Router oder enthält das Advertisement mehrere Präfixe, erzeugt sich das System aber auch mehrere IPv6-Adressen.



*Mit der Link-Local-Adresse kommuniziert der Client mit allen Routern im Netz über die Multicast-Adresse „ff02::2“.*

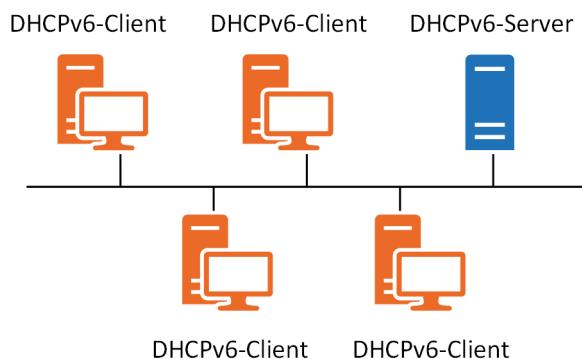
### Fälschungssicherheit

Die IPv6-Firewall-Regeln müssen, wie weiter oben beschrieben, diese ICMPv6-Nachrichten zulassen, damit die Autokonfiguration funktioniert. Diese Meldungen lassen sich aber nicht zustandsorientiert filtern, weil der Router sie per Multicast verschickt hat. Router-Solicitation- und -Advertisement-Nachrichten sind aber auch nicht authentifizierbar. Somit kann ein Angreifer ebenfalls solche Router-Advertisement-Nachrichten verschicken und damit Router erfolgreich fälschen. Das kann für MiM (Man in the Middle)-Angriffe missbraucht werden. Im Internet stehen dazu bereits entsprechende Werkzeuge zur Verfügung.

In der Praxis ist die Stateless Autoconfiguration für moderne Netze ohnehin nicht wirklich geeignet. Mit der Autokonfiguration können der Netzpräfix und die Routen zugewiesen werden, aber weder die DNS-Server noch weitere Optionen werden dabei berücksichtigt. Mit RFC 5006 hat man dies für DNS-Server nachgeholt, dies wird aber leider noch nicht von allen Betriebssystemen unterstützt und besitzt den Status „experimental“. Da in den Netzen ohnehin per IPv4 DHCP-Dienste Informationen wie die DNS-Domäne, NTP-Server oder PXE-Bootserver verteilt werden, kann man auch die Optionen für IPv6 mit verteilen, vorausgesetzt der DHCP-Server unterstützt dies. Dabei wird durch das Betriebssystem per Autokonfiguration die IPv6-Adresse gebildet und zusätzlich über DHCP die Optionen erfragt, z. B. DNS-Server und DNS-Domain. Windows 7 und Vista nutzen dies bereits. Dazu muss im Router-Advertisement (RA) das „OtherConfig“-Flag gesetzt sein.

### Stateful DHCP und DNS

Schließlich existieren auch Umgebungen, in denen der DNS-Server die Hostnamen der Systeme nach deren Boot dynamisch registrieren sollen, wie das für Active Directory-Strukturen typisch ist. Linux-Systeme mit BIND9 tun sich damit traditionell schwerer, weshalb viele Administratoren bei IPv4 einen Umweg nehmen und den DHCP-Server berechtigen, die Informationen in der DNS-Zone dynamisch zu ändern.



*An der betagten IP-Ausgabe per DHCP führt trotz IPv6-Autokonfiguration selten ein Weg vorbei.*

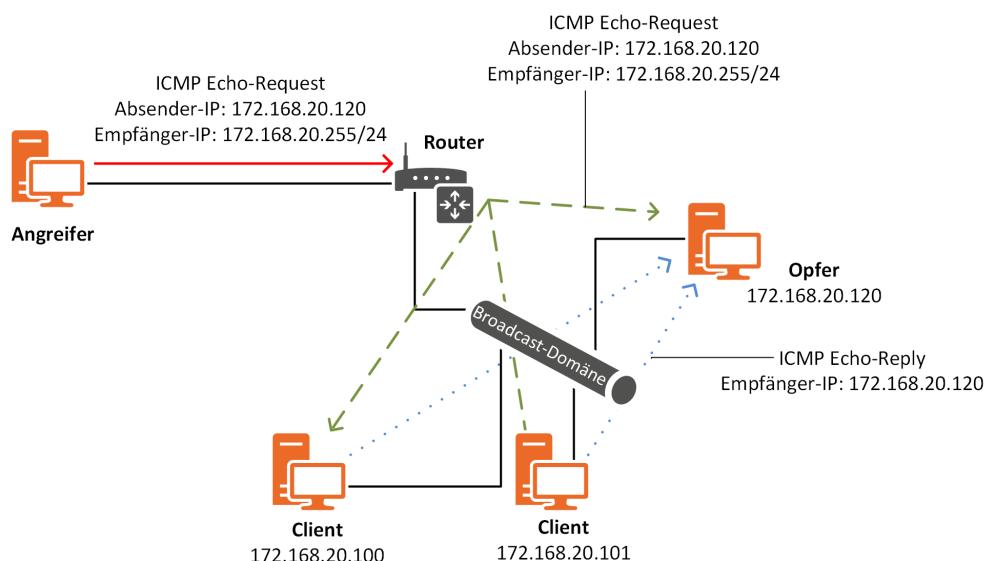
Mit IPv6 kann aber weder die Autokonfiguration noch das zustandslose DHCP genutzt werden, da der DHCP-Server die IPv6-Adresse des Clients kennen muss, um sie einzutragen. Dazu muss ein Stateful-DHCP-Server vorhanden sein, der IPv6-Adressen aus einem Pool vergibt. Windows-Clients unterstützen das automatisch, wenn das Router-Advertisement das „ManagedFlag“ enthält (vgl. Kap. 5). Für Linux-Systeme muss der entsprechende DHCP-Client nachgerüstet und konfiguriert werden. Auch hier gilt, dass man sich durch redundante Strukturen gegen die Folgen des Ausfalls eines DHCP-Servers wappnen sollte.

## 9.6 Bedrohungsszenarien

### Broadcast-Attacken

IPv6 hat den Broadcast abgeschafft.

Bei IPv4 war es möglich, mit gefälschten Broadcast-Paketen einzelne Hosts oder bei entsprechender Last auch komplette Netzwerke lahmzulegen. Bei der sogenannten Smurf-Attacke wird ein ICMP Echo Request an die Broadcast-Adresse mit der Absenderadresse des anzugreifenden Hosts geschickt. Die Antwortpakete kommen dann auf dem „Opfer“-Rechner an, je nach Größe des Netzes eine erhebliche Belastung. Durch falsch konfigurierte Router kann der Angriff noch verstärkt werden. Dem wurde aber auch bei IPv4 schon dadurch ein Riegel vorgeschoben, dass Hosts auf Echo Requests an Broadcasts nicht mehr antworten.



Das **Neighbor-Discovery-Protokoll** (NDP) nutzt Multicast-Adressen, die dem Broadcast ähneln, um die MAC-Adresse des Kommunikationspartners zu ermitteln. Die Multicast-Gruppen sind jedoch so gewählt, dass die Kommunikation in vielen Fällen nur mit dem einen Rechner stattfindet, dessen MAC-Adresse ermittelt werden soll.

**DHCPv6** verwendet die beiden Multicast-Adressen `ff02::1:2` und `ff05::1:3`. Die erste ist die Link-Local-Multicast-Adresse (`ff02`), die für alle DHCP-Agenten (Server und Relays) verwendet wird. Die zweite Adresse ist als Site-Local-Multicast-Adresse (`ff05`) DHCP-Servern vorbehalten.

Durch diese Änderungen wird erreicht, dass in den meisten Fällen nur noch die beteiligten Systeme die Pakete im IP-Stapel bearbeiten. Für die IP-Adressen gelten dann auch entsprechende Multicast-MAC-Adressen. IPv6-Multicast-Adressen werden auf MAC-Adressen abgebildet, indem die letzten vier Byte der Adresse in die MAC 33-33-00-00-00-00 eingesetzt werden. Im Vergleich zur Broadcast-Kommunikation unter IPv4 wird damit eine deutlich geringere Netzwerklast erreicht.

### ARP Spoofing

Bei ARP Spoofing wird die Absender-MAC-Adresse gefälscht. Das klappt bei IPv6 nicht mehr, da ARP durch NDP ersetzt wurde.

### NDP

Ein Angreifer erzielt den gleichen Effekt mit NDP-Spoofing. Die IPv6-Entwickler haben aber diesen Angriffsvektor erkannt und Modifikationen des Protokolls in RFC 39791 (Secure Neighbor Discovery Protocol, SEND) spezifiziert. SEND verwendet kryptographisch erzeugte Adressen mit öffentlichen Schlüsseln für die Authentifizierung der NDP-Nachrichten. IPv6-Router müssen sich über Zertifikate ausweisen, die jedes IPv6-Gerät über eine Certificate Authority (CA, Zertifizierungsstelle) abgleicht. In Desktopsystemen ist SEND bisher nicht verbreitet.

## Netscan

Bei Netscan wird ein Netz nach IP-Adressen und Diensten durchsucht. Beim großen Adressraum von IPv6 ist diese Methode nicht effektiv zu bewältigen.

## Portscan

Beim Portscan wird ein Host, Router oder Server nach verwundbaren Stellen abgesucht. Mit den verfügbaren Tools (IPv6) sind genauso Portscans für TCP/UDP möglich. Portscans können von Router/Firewall erkannt und geblockt werden.

## Denial of Service

Beim Denial of Service werden Dienste mit einer Flut unnützer Anfragen beschäftigt und sind dann nicht mehr erreichbar. Da das auf Anwendungsebene passiert, sind IPv4 und IPv6 gleichermaßen betroffen. Da die Angriffe meist bestimmte Muster haben, können sie von Routern/Firewall erkannt und ausgefiltert werden.

## Man in the Middle

Der Klassiker, bei dem beiden Kommunikationspartnern die erwartete Gegenstelle vorgegaukelt wird. Im Gegensatz zu Denial-of-Service-Angriffen ist es mit MITM-Angriffen möglich, die kompletten IPv6-Verbindungen mitzuhören und zu verändern. Zum Beispiel stellt das `thc-toolkit` die notwendigen Tools bereit. Damit lassen sich unter anderem folgende Angriffe ausführen:

- ✓ Router Advertisement Spoofing: einen zusätzlichen Default-Router im Netzwerk etablieren
- ✓ Neighbor Advertisement Spoofing: Link-Layer-Adresse des Opfers fälschen und den Verkehr über den Angreifer leiten (entspricht einem ARP-Spoof in IPv4)
- ✓ Per zusätzlichem DHCPv6-Server falsche DNS-Einträge verteilen: Der Netzwerkverkehr kann dann über einen vom Angreifer kontrollierter Server umgeleitet werden.

Ein großer Teil dieser Angriffsmöglichkeiten kann durch die Verwendung von SEND (siehe NDP) abgewehrt werden.

## SPAM

Bei SPAM-Bekämpfung auf Anwendungsebene gibt es keinen Unterschied zwischen IPv4 und IPv6. Der Rat, SPAM-verdächtige E-Mails gar nicht erst anzunehmen, bleibt. Das geschieht heute über Blacklisting. Dabei werden bekannte Mailserver an zentrale Stellen gemeldet, die dann ihre Informationen mit anderen teilen. So entsteht eine „schwarze Liste“, mit deren Hilfe ein Großteil des SPAM-Aufkommens vorab gefiltert werden kann.

Bei IPv6 ist diese Methode wegen des riesigen Adressraumes nicht mehr durchführbar. Sobald die Spammer IPv6 für sich entdeckt haben, könnte eine große SPAM-Welle auf uns zurollen. Abhilfe schafft das auch heute schon angewandte Grey-Listing. Dabei wird jede E-Mail grundsätzlich erst einmal abgelehnt. Ein „seriöser“ Mailserver wird nun einen zweiten Versuch starten, der dann die Zustellung vornimmt. Ein Spammer hat für so etwas keine Zeit und springt sofort zu seinem nächsten Ziel. Diese Methode klappt mit IPv6 und IPv4.

## Sicherheitsprobleme bei Übergangstechnologien

### Dual-Stack

Die gleichzeitige Verwendung von IPv4 und IPv6 erhöht die Sicherheitsanforderungen, da nun zwei Protokolle zu schützen sind. Zusätzliche Maßnahmen im Bereich Firewall sind damit verpflichtend.

### Extension Header-Attacken

Ein Angreifer sendet

- ✓ Pakete mit vielen Extension Header
- ✓ Extension Header mit vielen Optionen
- ✓ ungültige Extension Header oder Optionen (Fuzzing)
- ✓ Extension Header, in denen Informationen versteckt sind (Covert Channel)
- ✓ sehr viele Pakete mit Router-Alert-Option

Einsatzziele von Extension Header-Attacken sind das Umgehen von Sicherheitseinrichtungen wie z. B. RA-Guard, Angriffe auf das Endsystem, um dessen Robustheit zu testen, oder Angriffe auf Router, um diese zu beschäftigen.

### Tunnel

Ungewollte bzw. unbeabsichtigte Tunnel über Teredo/ISATAP können Verbindung zu externen Netzen herstellen, auch wenn dies nicht gewollt ist. Hier sollte der Administrator durch Blocken der entsprechenden Ports bzw. Protokolle vorbeugend tätig werden.

### Netzwerk Address- und Protokoll-Translation

Adress- und Protokoll-Translation ermöglicht die Kommunikation zwischen IPv6-Knoten in IPv4-Netzwerken und IPv4-Knoten in IPv6-Netzwerken. Die RFC 2766 [Network Address Translation – Protokoll Translation (NAT-PT)] wurde durch RFC 4966 abgelöst, wegen einer Reihe von Sicherheitsproblemen, welche teilweise nicht lösbar waren oder deren Lösung sich so komplex darstellte, dass sie nicht mehr praktikabel war. Diese (veraltete) Spezifikation war zudem auch unverträglich mit DNSSEC und hätte deren Deployment beeinträchtigt.

Derzeit laufen Untersuchungen zur Lösung des NAT, ohne die Nachteile der ursprünglichen Spezifikation zu erhalten. Ein aktuelles Draft ist *draft-ietf-behave-v6v4-xlate-stateful* mit dem Titel „Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers“. Hierbei werden Diskussionen um NAT64, NAT46, NAT66, NAT44, Carrier Grade NAT und Large Scale NAT geführt.

Die Translation als Mechanismus ist mit einigen grundsätzlichen Nachteilen verbunden und sollte darum nur dann eingesetzt werden, wenn kein anderer Übergangsmechanismus möglich ist. Das Gateway stellt dabei einen Flaschenhals dar, und die Übersetzungsarbeit beeinträchtigt die Performance. Es gelten die gleichen Einschränkungen wie beim Einsatz von IPv4-NAT (IPSec und DNSSEC) (vgl. Kapitel 8).

## 9.7 Zusammenfassung

Zurückblickend kann man sagen, dass beim Design des inzwischen mehr als 20 Jahre alten IPv6-Protokolls Beachtliches geleistet wurde. Die Selbstorganisationsmöglichkeiten am IPv6-Stack und der konsequente Verzicht auf Broadcast-Verfahren sind kluge Entscheidungen gewesen. Allerdings haben sich seit Erscheinen der RFC 1897 im Januar 1996 die Betriebssysteme und Anforderungen weiterentwickelt, IPv6 dagegen kaum.

Die Autokonfiguration ermöglicht problemlose Kommunikation im Netzabschnitt, benötigt aber für übergreifende Kommunikation Router Advertisements für das Gateway und, sofern vom Client unterstützt, DNS-Informationen. In größeren Umgebungen ist damit ein DHCPv6-Server unverzichtbar.

DNS und insbesondere Firewalls (SPI) gewinnen an Bedeutung. Ohne IPSec erhöht IPv6 auch die Sicherheit in einem Unternehmensnetz nicht. Hier stellt sich die Frage, inwieweit der Administrator sicherstellen kann, ob die Maschinen eines Unternehmens nicht weltweit (unbeabsichtigt) erreichbar sind.

# 10 Mobile IPv6 und Migration

## In diesem Kapitel erfahren Sie

- ✓ was Mobile IPv6 bedeutet
- ✓ welche Perspektiven der IPv6-Einsatz mit sich bringt
- ✓ was bei der Migration von IPv4 auf IPv6 grundsätzlich zu berücksichtigen ist
- ✓ welche grundlegenden Überlegungen bei einer Migration zu berücksichtigen sind

## Voraussetzungen

- ✓ IPv6-Adressaufbau
- ✓ Routing

## 10.1 Mobile IPv6

Mobile IPv6 definiert einen Standard, der technisch eine ständige und unterbrechungsfreie Verbindung der mobilen Clients wie Smartphone, Tablet PC, Laptop oder anderer mobiler Endgeräte gewährleistet, während sie sich zwischen unterschiedlichen Netzwerken bewegen. Die potentielle Neueinwahl in jeweils andere Netze, die der mobile Nutzer unter IPv4 mit z. T. aufwändigen manuellen Konfigurationsaufgaben vornehmen muss, entfällt damit.

Der Standard ist derzeit noch experimentell und muss von Clients und – bei Verwendung von **route optimization** (siehe Abschnitt „Funktionsweise von Mobile IPv6“) – auch von Routern unterstützt werden. In der Erweiterung **Proxy Mobile IPv6** (RFC 5213) wird eine netzseitige Implementierung beschrieben, bei der keine clientseitige Änderungen notwendig ist. Eine zukünftige Verbreitung von Mobile IPv6 hängt von der Unterstützung durch die Netzwerkinfrastrukturen und die Betriebssysteme ab. Einen Schub könnte die Vergabe von nativen IPv6-Adressen durch die Mobilfunkbetreiber bringen.

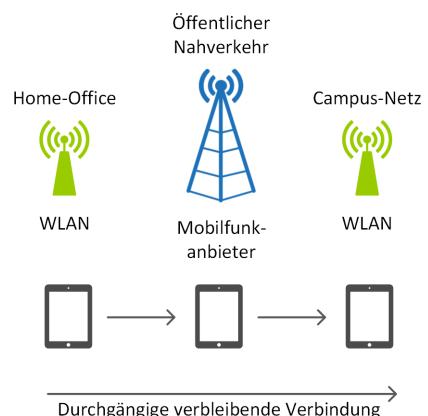
### Übersicht zu unterbrechungsfreien Verbindungen

Die Idee der unterbrechungsfreien Verbindung geht schon auf IPv4 zurück und wird in der RFC 2002 beschrieben. Unter IPv6 soll jetzt die Möglichkeit der unterbrechungsfreien Verbindung trotz Netzwechsels mit Mobile IPv6 Einzug halten.

Die Beschreibung zu Mobile IPv6 ist in der RFC 6275 von Juli 2011 zu finden und löst damit die Beschreibung in der RFC 3775 aus dem Jahr 2004 ab.

Wird eine Verbindung zwischen zwei Kommunikationspartnern aufgebaut, basiert diese auf den IP-Adressen dieser Kommunikationspartner.

Beim Wechsel eines der mobilen Clients zwischen zum Beispiel einem WLAN- und einem UMTS-Netz ändert sich die IP-Adresse des Clients; die Verbindung wird unterbrochen und muss mit einer neuen IP-Adresse neu aufgebaut werden. Datenübertragungen können verloren gehen, müssen wiederholt gesendet werden oder führen zu insgesamt fehlerhaften Übertragungen.



Zwar ist die Verbindung innerhalb eines WLANs zwischen verschiedenen Accesspoints ohne Weiteres möglich, ebenso die Aufrechterhaltung einer Verbindung innerhalb eines Mobilfunknetzes oder die netzübergreifende Verbindung, jedoch erfolgt die Weiterleitung über die Sicherungsschicht und basiert auf den Fähigkeiten der einzelnen Übertragungstechnologien.

Hier liegt jedoch der Knackpunkt: Beim Wechsel der jeweiligen Übertragungstechnologie kommt es zwangsläufig auch zu einem Wechsel der IP-Adresse, die Verbindung muss neu hergestellt werden. Nicht abgeschlossene Transaktionen gehen verloren, es sei denn, in den Anwendungen sind Möglichkeiten implementiert, die den Abbruch erkennen, um die Transaktion bei einem IP-Wechsel wieder aufzunehmen.

IPv6 bietet demgegenüber eine komplette Lösung auf der Netzwerkschicht, nämlich IP-basierend, an und ermöglicht somit eine Lösung, die von unterschiedlichen Anbindungstechnologien und Softwareimplementierungen völlig unabhängig ist.

## Funktionsweise von Mobile IPv6

Befindet sich das mobile Gerät in seinem Heimatnetz, enthält die Adresse das Netzwerkpräfix des lokalen Netzes und die IP-Pakete werden normal zum Ziel geroutet. Sobald das mobile Gerät in ein fremdes Netz wechselt, erhält das mobile Gerät eine Adresse über z. B. die statusorientierte oder statuslose Autokonfiguration.

<b>Mobile Node (MN)</b>	Mobiles Gerät, Mobiler Knoten
<b>Correspondent Node (CN)</b>	Kommunikationspartner des mobilen Gerätes
<b>Home Link</b>	Heimatnetzwerk des mobilen Gerätes
<b>Home Agent (HA)</b>	Router, der an das Heimatnetzwerk angeschlossen ist und für die Bindung der festen Adresse des MN verantwortlich ist
<b>Home Address</b>	Globale-Unicast-Adresse des MN
<b>Foreign Link</b>	Bezeichnung der Fremdnetzwerke, von denen der MN gerade seine Adresse empfangen hat
<b>Care-of-Address (CoA)</b>	Aktuelle Adresse des MN, die er in einem Fremdnetzwerk erhalten hat
<b>Binding</b>	Hergestellter Zusammenhang zwischen der Home Address und der Care-of-Address
<b>Binding-Update</b>	Aktuelle Care-of-Adresse, die vom MN an den CN gesendet wird

### Begriffe zu Mobile IPv6

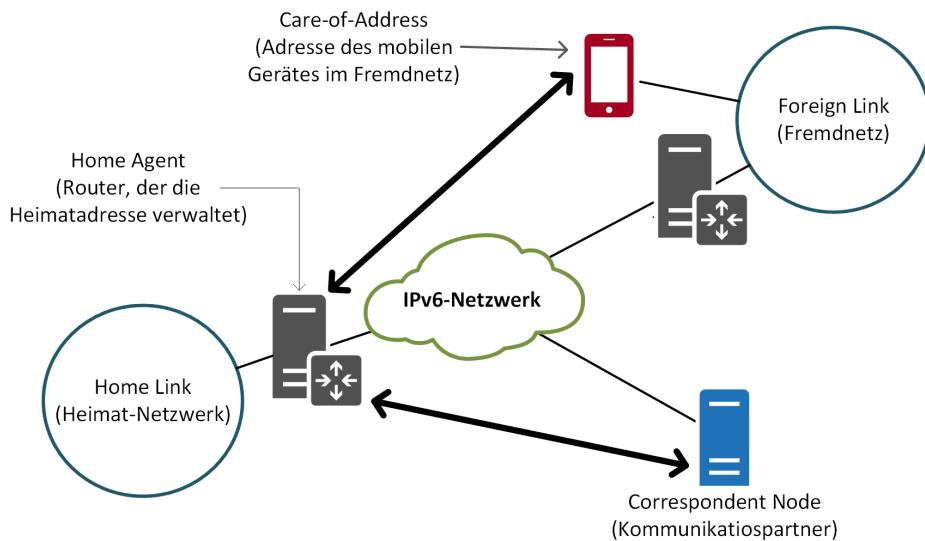
Die Adresse, die dem mobilen Gerät über z. B. die statusorientierte oder statuslose Autokonfiguration zugewiesen wird, entspricht der **Care-of-Address** und wird dem **Home Agent** bekanntgegeben. Dazu wird ein **Binding Update** an den Home Agent gesendet, und dieser bestätigt mit einem Binding Acknowledgement. Daher kennt der Home Agent immer die aktuelle CoA des mobilen Gerätes, denn sobald das Netz wieder gewechselt wird, erfolgt ein erneutes Binding Update. In RFC 6275 werden zwei Modi beschrieben:

- ✓ Bidirectional Tunneling
- ✓ Route Optimization

### Bidirectional Tunneling

Im ersten Fall benutzt der Kommunikationspartner bei einem Verbindungsaufbau zum mobilen Gerät die **Home Address**, die dann auch zum Home Agent führt.

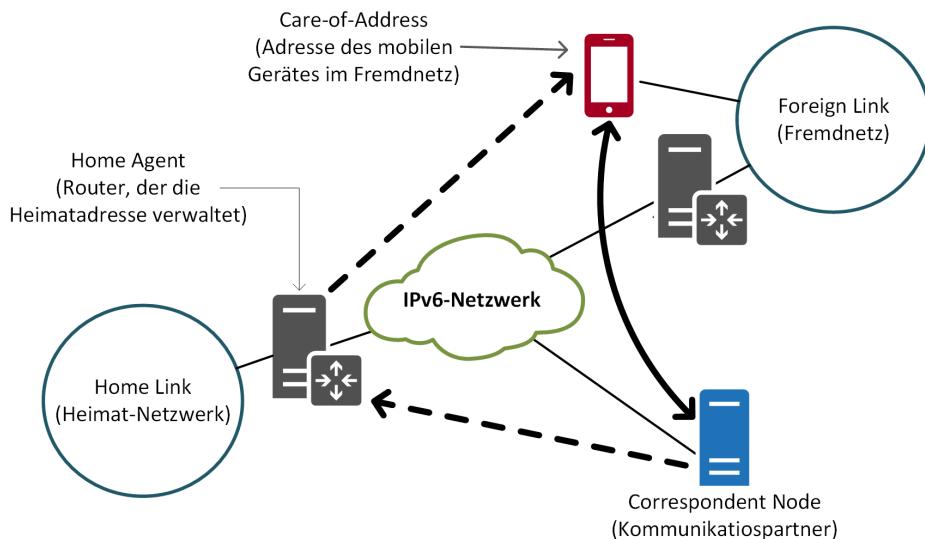
Wenn der Kommunikationspartner keine Unterstützung für Mobile IPv6 hat, erfolgt ein **Bidirectional Tunneling**. In diesem Fall werden Pakete vom Kommunikationspartner an den Home Agent gesendet, der diese wiederum durch einen Tunnel zum mobilen Gerät weiterleitet. Das mobile Gerät sendet die Antwort durch einen Tunnel zum Mobile Agent zurück, der diese dann an den Kommunikationspartner weiterleiten kann.



*Bidirectional Tunneling*

### Route Optimization

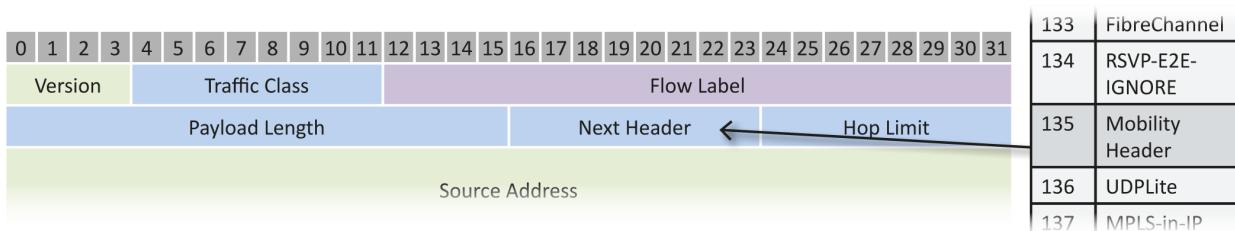
Im zweiten Fall findet eine **Route Optimization** statt. Bedingung ist, dass der Kommunikationspartner Mobile IPv6 unterstützt. Die Kontaktaufnahme vom Kommunikationspartner zum mobilen Gerät erfolgt wieder über den Home Agent. Danach verschickt das mobile Gerät ein Binding Update direkt an den Kommunikationspartner und übernimmt dessen Adresse in seine Binding Update List. Der Kommunikationspartner speichert das Binding in seinem Binding Cache und kann jetzt seine Pakete direkt an das mobile Gerät senden. Dabei kommt ein Typ-2-Routing-Header zum Einsatz, der speziell für Mobile IPv6 vorgesehen ist und die Care-of-Address und die Zieladresse miteinander verbinden kann.



*Route Optimization*

## Mobility Header (MH) als Erweiterung des IPv6-Headers

In Kapitel 3 wurde die flexible Gestaltung des IPv6-Headers beschrieben. Optionale Header werden im Next Header-Feld angekündigt. Ein **Mobile IPv6-Erweiterungsheader** wird durch den Wert 135 im Next Header beschrieben (siehe [www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml](http://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml)).



Das Feld „Next Header“ im IPv6-Header

## Aufbau des Mobility Headers

Der Mobility Header (MH) enthält Informationen über den Status der mobilen IPv6-Verbindung. Hierbei ist insbesondere das Feld **MH Type** interessant. Es sind verschiedene MH-Typen definiert, die Auskunft über die Art der Nachricht geben; siehe Tabelle unter

<https://www.iana.org/assignments/mobility-parameters/mobility-parameters.xhtml#mobility-parameters-1>.

Der Mobility Header hat folgenden Aufbau:



Aufbau des Mobility Headers

Die einzelnen Felder haben folgende Bedeutung:

<b>Payload Protocol</b>	Entspricht dem Next-Header-Feld des IPv6-Headers und verweist auf den nächsten Erweiterungsheader
<b>Header Length</b>	Länge des Mobility Headers ohne die ersten 8 Byte
<b>MH Type</b>	Typ der Mobility Nachricht, z. B. <ul style="list-style-type: none"> <li>✓ 0 <b>BRR</b> (Binding Refresh Request) RFC 3775</li> <li>✓ 5 <b>BU</b> (Bindung Update) RFC 3775, RFC 4140</li> <li>✓ 6 <b>BA</b> (Binding Acknowledgement) RFC 3775</li> </ul>
<b>Reserved</b>	Vom Sender auf 0 gesetzt und vom Empfänger ignoriert
<b>Checksum</b>	Prüfsumme
<b>Data</b>	Je nach verwendetem MH Type

Für das folgende Beispiel ist der MH Type auf 5 (Binding Update) gesetzt. Ein Binding Update wird versendet, wenn sich die Care-of-Address (CoA) des Clients ändert. Die Adresse wird dabei dem IPv6-Header entnommen. Alternativ kann in den Mobility Options auch eine andere CoA mitgeteilt werden.



Die Nutzdaten bestehen aus:

<b>Sequence Number</b>	16 Bit Integer, um Zuordnung und Reihenfolge der Updates sicherzustellen
<b>Flags AHLK</b>	A → Bestätigung (Acknowledgement) erforderlich H → Home Flag (Paket an Home Agent gerichtet, A erforderlich) L → Link Local Address Compatibility (Host-ID nicht verändert) K → Key Management Mobility Capability (wenn nicht gesetzt, muss die Security Association nach jedem Netzwechsel neu ausgehandelt werden)
<b>Reserviert</b>	12 Bit reserviert
<b>Lifetime</b>	Verbleibende Gültigkeit der Care-of-Address in 4s Einheiten
<b>Mobility Options</b>	Weitere Optionen sind möglich und werden durch Header Length angekündigt, z. B. abweichende CoA.

Nach erfolgtem Update wird die Zuordnung in den Binding Cache des Home Agents geschrieben. Ist die Bindung bereits vorhanden, wird sie überschrieben.

Home address	Care-of address	Sequence no.	Lifetime	Flags
20011:db3:8:1:a:b:c:d	20011:db3:10:1:a:b:c:d	11	250	A/H/K/L
20011:db3:8:1:a:b:c:9	20011:db3:12:1:a:b:c:9	2000	400	A/H/L

*Bindungs Cache-Einträge im Home Agent*

## Sicherheit in Mobile IPv6

Da die Kommunikation zwischen dem mobilen Gerät und dem Partner anfällig für verschiedene Attacken wie Man in the Middle, Session Hijacking oder DoS ist, gibt es verschiedene Sicherheitsmaßnahmen.

Die Verwendung wird in RFC 6275 vorgegeben:

Binding Updates und Binding-Bestätigungen müssen wie folgt über IPSec (vgl. Kapitel 10) gesichert werden:

- ✓ ESP (Encapsulation Security Payload) der Binding Updates und der Bestätigung zwischen dem mobilen Gerät und dem Home Agent muss unterstützt und eingesetzt werden. Dadurch wird eine Übernahme der CoA durch Angreifer verhindert.
- ✓ ESP für die Home Test Init und die Home Test-Mitteilung, die zwischen dem mobilen Gerät und dem Home Agent über einen Tunnel laufen, muss unterstützt und sollte benutzt werden.
- ✓ ESP für ICMPv6 zur Präfix-Anfrage muss unterstützt und sollte genutzt werden.
- ✓ ESP für die Daten zwischen mobilem Gerät und dem Home Agent kann unterstützt und eingesetzt werden.

Die Verbindung zwischen mobilem Gerät und Kommunikationspartner über Route Optimization muss nicht, kann aber über IPSec geschützt werden.

Wie im ersten Abschnitt des Kapitels erwähnt, ist der Status experimentell und als Zukunftsszenario zu verstehen. Ähnliche Ansätze verfolgen z.B. LISP (Locator/Identifier Separation Protocol), RFCs 6830-6836 und MPTC (Multipath TCP), RFCs 6181, 6182, 6356, 6824, 6897. LISP und MPTC sind aber beide nicht ausdrücklich auf IPv6 beschränkt.

## 10.2 Anmerkungen zu Migrationsszenarien

In den vorangegangenen Kapiteln wurden verschiedenen Möglichkeiten zur Nutzung von IPv6 vorgestellt. Nun ist es sicherlich weder angeraten noch praktikabel, sämtliche Möglichkeiten in einer Netzwerkumgebung gleichzeitig zu nutzen.

Im Folgenden werden die in der Unterlage beschriebenen Verfahren überblicksartig zu grundsätzlichen Aussagen hinsichtlich einzelner Migrationsaspekte in unterschiedlichen Netzwerkszenarien genutzt.

### IPv6 im LAN

Dank Autokonfiguration ist heute jedes aktuelle Betriebssystem in der Lage, IPv6 ohne weitere soft- oder hardwaretechnische Zusatzelemente zu nutzen. Sofern es nicht administrativ abgeschaltet wurde, ist IPv6 auch aktiviert. So ist zumindest im eigenen Netzabschnitt eine IPv6-Kommunikation möglich, theoretisch könnte sogar auf IPv4 verzichtet werden. In der Praxis sieht es allerdings in den meisten Umgebungen so aus, dass sich noch eine große Anzahl von Geräten im Einsatz befindet, die nicht IPv6-fähig ist. Die am häufigsten genutzte Möglichkeit wird deshalb sein, diese Geräte mittels der Dual-Stack-Technik anzusprechen.

Für eine reine IPv6-Umgebung müssten solche nicht IPv6-fähigen Geräte über eine Protokollumsetzung eingebunden werden. Beispiele für nicht IPv6-fähige Geräte sind u. a. Netzwerkdrucker, Webcams oder IP-Telefone.

Auch WLAN-Access Points sind oft nicht IPv6-fähig. Da ein WLAN-Access Point in der Regel als Bridge, also auf OSI Schicht 2 arbeitet, ist der Einsatz von IPv6 über solche Geräte für die Benutzer dennoch kein Problem. Allerdings ist die Administration nur mit IPv4 möglich. Gleicher gilt auch für gemanagte Switches.

### IPv6 im Firmennetz

Sobald Router im Netzwerk eingesetzt sind, ist es erforderlich, den Endgeräten die Router-Adresse als Default Gateway bekannt zu geben. Das Router Advertisement geschieht dabei durch den Router selbst. Neben seiner IPv6-Adresse wird zudem die Netzschicht als Unique Local oder Unique Global Präfix bekannt gegeben.

Um eine übergreifende Kommunikation zu ermöglichen, wird darüber hinaus ein DNS-Server benötigt. Hier kann weiterhin der statisch oder per DHCP bekannte IPv4-Nameserver verwendet werden. Ob eine Migration des DNS-Servers auf IPv6 an dieser Stelle sinnvoll ist, bleibt der strategischen Planung im eigenen Netz überlassen. Eine IPv6-Nameserver-Adresse kann wie in Kapitel 5 beschrieben über DHCPv6 oder nach RFC 6106 über den Router bekannt gegeben werden. Selbstverständlich ist weiterhin auch ein statischer Eintrag auf den Clients möglich.

Die diesbezüglichen Konfigurationsmöglichkeiten eines IPv6-fähigen Routers wurden im Kapitel 5 beschrieben. Dabei ist insbesondere zu beachten, dass je nach vorhandener Infrastruktur die entsprechenden Flags **O** (other configuration) oder **M** (managed configuration) gesetzt werden.

### Kleines Unternehmen, SoHo, Homeuser

Solange im Netz nur ein Router eingesetzt ist, über den die Verbindung ins Internet oder zu einem Firmennetz erfolgt, ist es ratsam, alle Konfigurationsaufgaben über diesen Router vorzunehmen. Clients generieren demzufolge selbst einen Host Identifier und erhalten vom Router den Global Routing Präfix für die Internetkommunikation sowie die Nameserver-Adresse. Damit ist eine problemlose IPv6-Kommunikation über die Netzwerkgrenzen hinweg gewährleistet.

### Woher bekommt der Router IPv6?

Im Kapitel 8 wurden dazu verschiedene Möglichkeiten aufgezeigt. Für die Anbindung von Firmennetzen ist es möglich, ein VPN auf Schicht 2 aufzubauen, sodass beliebige Schicht 3-Protokolle und somit auch IPv6 übertragen werden können. Eine andere Möglichkeit stellt die temporäre Verwendung von ISATAP dar, was aber feste IP-Adressen auf beiden Seiten voraussetzt.

Wenn es um eine direkte Internetanbindung mittels fest zugewiesenenem IPv6-Präfix geht, ist die erste Wahl natürlich ein Provider, der einen IPv6-Adressbereich nativ zur Verfügung stellen kann. Dieser Bereich muss dann noch entsprechend aufgeteilt werden (vgl. Kap 4). Im einfachsten Fall wird aus dem zugewiesenen Adressbereich nur ein einziges /64-Netz genutzt. Eine native Anbindung bietet zudem die höchste Verfügbarkeit, da nur ein ISP Ansprechpartner ist, mit dem die Verfügbarkeit per Vertrag vereinbart wird.

Ist eine native Anbindung nicht möglich und ein fester IPv6-Präfix gewünscht, kann ein Tunnelbroker verwendet werden. Eine Auflistung findet sich u. a. in Wikipedia unter [https://de.wikipedia.org/wiki/Liste\\_von\\_IPv6-Tunnelbrokern](https://de.wikipedia.org/wiki/Liste_von_IPv6-Tunnelbrokern). Die Nutzung erfordert im Allgemeinen eine Public IPv4-Adresse am Internetanschluss. Die Verfügbarkeit ist somit von der eigenen Internetanbindung und der Zuverlässigkeit des ISP abhängig, ist aber im Allgemeinen ausreichend.

Ist kein fester IPv6-Präfix notwendig, können andere Übergangslösungen genutzt werden, die größtenteils ohne Anmeldung funktionieren (vgl. Kapitel 8).

### Was ist mit Teredo?

Das beschriebene Teredo-Verfahren ist gut geeignet, um von einzelnen Clients (und nicht von ganzen Netzwerken) zu Test-/Übungszwecken eine IPv6-Konnektivität ins Internet herzustellen. Dabei sollten Sie es allerdings belassen, für eine dauerhafte Anbindung ist Teredo nicht geeignet. Insbesondere ist zu beachten, dass der PC dabei ungeschützt (allenfalls gesichert durch die Betriebssystem-eigene Firewall) im Internet steht und dabei zahlreichen Angriffen ausgesetzt sein kann. In Firmennetzen kommt so eine Anbindung aus Sicherheitsgründen nicht infrage und sollte in der Firmenfirewall auf IPv4-Ebene geblockt sein. Eine ständige Verfügbarkeit von Teredo ist nicht gewährleistet. So gab es beispielsweise im Juli 2013 einen längeren Ausfall der Teredo-Server von Microsoft. Alternative Server stehen zwar zur Verfügung, müssen aber erst konfiguriert werden (vgl. Kapitel 8).

## Abschließende Überlegungen

Die Umstellung eines kleinen Netzes ist – wie gezeigt – ohne großen Aufwand möglich. Bei Enterprise-Umgebungen ist dagegen eine umfangreiche Vorplanung unumgänglich. Allerdings ist nicht zu erwarten, dass trotz der Rückbesinnung auf die Möglichkeiten direkter Ende-zu-Ende-Verbindungen die Verantwortlichen dieser Netze eine vollständige Umsetzung dieses Konzeptes in Betracht ziehen werden.

Seit den 1990er-Jahren werden Netze aus gutem Grund durch Firewalls geschützt und hinter NAT versteckt. Die Gründe sind durch das IPv6-Protokoll nicht weggefallen, sodass man weiterhin auf bewährte Techniken setzen muss. Firmennetze werden weiterhin durch ausgefeilte Firewall-Konzepte nach außen geschützt werden müssen. Von innen nach außen sind unautorisierte Zugriffe die Ausnahme. Dort wird der Datenverkehr über interne Server (Webproxy, Mail, DNS usw.) abgewickelt, und nur diese Server haben Zugang ins Internet. Das kann paradoxerweise dazu führen, dass sich der Bedarf an öffentlichen IP-Adressen im internen Netz verringert. Der Bedarf an öffentlichen Adressen entsteht also nicht primär in Firmen- oder privaten Netzwerken, sondern im Bereich Webhosting und im „Internet der Dinge“ (z. B. Car2Car-Kommunikation, Hausautomation – Smart Home, Smart Metering, Ladeinfrastruktur für E-Mobilität).

Der große Vorteil von IPv6 besteht für weltweit tätige Unternehmen in der Möglichkeit, einen einheitlichen Adressraum zu nutzen. National tätige Unternehmen können hier auch intern mit Global Unique Präfixen arbeiten. Der beauftragte ISP sollte in der Lage sein, die verschiedenen Subnetze innerhalb seines Netzes an verschiedene Standorte zu routen. International wird das wegen der regionalen Zuweisung der Adressblöcke nicht funktionieren, sodass weltweit tätige Unternehmen intern auf Local Unique Präfix zurückgreifen werden. Die Standortvernetzung kann dann über VPN erfolgen.

An einzelnen Standorten wird ein Übergang ins Internet ermöglicht, die internen Local-Unique-Adressen werden dabei in öffentliche Global-Unique-Adressen umgesetzt. Dadurch wird eine hohe Redundanz erzielt, und Providerwechsel sind ohne interne Umnummerierung möglich. Ein providerunabhängiges Netz (PI) wie bei IPv4 und der damit verbundene Administrationsaufwand ist nicht mehr erforderlich.

# Anhang: Testumgebung

## In diesem Kapitel erfahren Sie

- ✓ wie die Testumgebung konfiguriert wird
- ✓ welche grundlegenden Tools verwendet werden

## Voraussetzungen

- ✓ Windows-Eingabeaufforderung

## A.1 Die Testumgebung

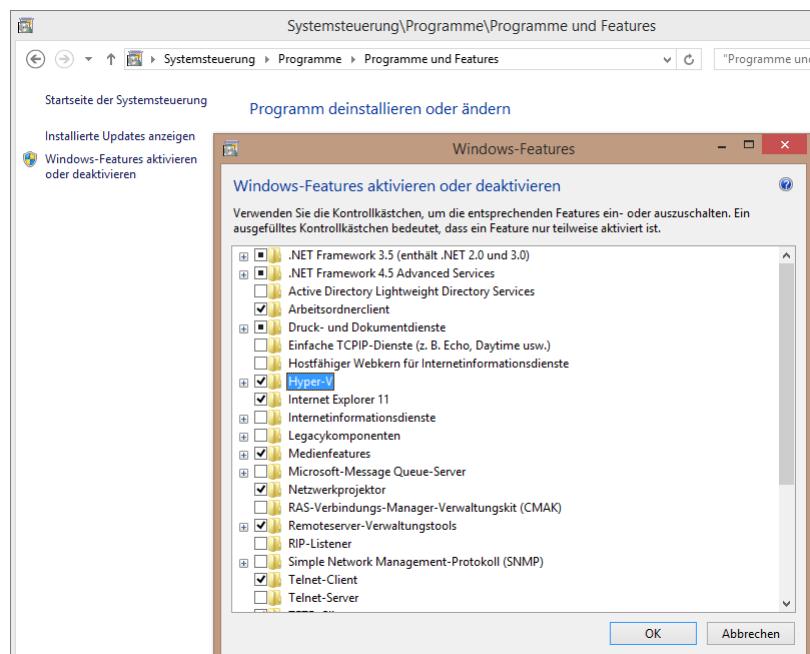
Die praktischen Übungen gehen von einer einheitlichen Testumgebung aus. Damit können die im Buch beschriebenen Grundlagen und Migrationsszenarien überprüft und nachgebildet werden. Die Umgebung wird zunächst als reine IPv4-Umgebung aufgebaut. Sie besteht aus

- ✓ zwei Windows 2012R2/2016 Servern
- ✓ zwei Windows 7 Clients oder höher

Alternativ können auch andere Betriebssysteme zum Einsatz kommen, die Übungen beschränken sich hier auf eine Windows-Umgebung. Die Übungen können mit dedizierten Rechnern aufgebaut werden. In der Praxis wird aber eher eine Virtualisierung zum Einsatz kommen. Im Folgenden wird die Konfiguration mit Microsofts Hyper-V beschrieben, das in den aktuellen Clientbetriebssystemversionen ab der Professional Variante aktiviert werden kann. Alternativ können auch VMware, Virtualbox oder andere Virtualisierungslösungen verwendet werden.

## A.2 Hyper-V aktivieren

Auf Windowsservern wird Hyper-V als Rolle hinzugefügt und steht nach einem Neustart zur Verfügung. Auf den Clientbetriebssystemen wird Hyper-V über die „Systemsteuerung/Programme/Programme und Features/Windows-Features aktivieren oder deaktivieren“ aktiviert. Bei Windows 10 führt eine Suche nach Hyper-V zum entsprechenden Menüpunkt.



Aktivieren von Hyper-V unter Windows 8

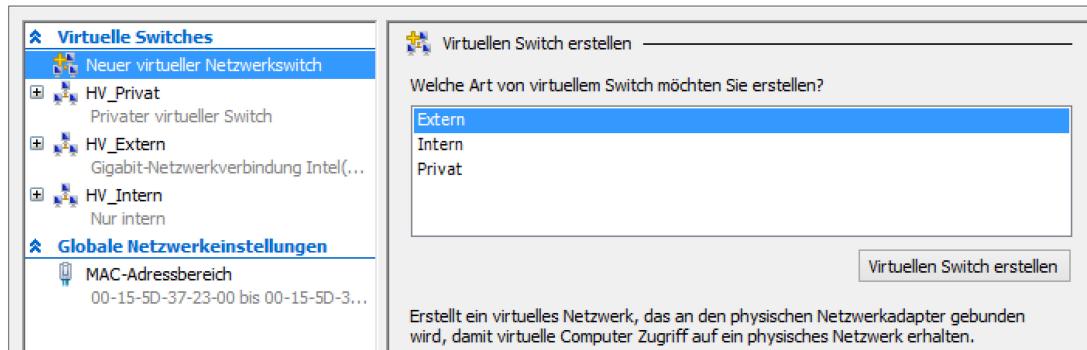
## A.3 Netzwerke einrichten mit Hyper-V-Netzwerkmanager

Die verschiedenen Netzwerktypen:

Netzwerktyp	Eigenschaften
Extern	Hyper-V und Wirtsrechner nutzen eine direkte Anbindung ans LAN
Intern	Es wird auf dem Wirtsrechner eine zusätzliche virtuelle Netzwerkkarte bereitgestellt über die ein Zugriff auf das virtuelle Netzwerk vom Typ „Intern“ möglich ist. Es besteht keine Verbindung ins LAN.
Privat	Vom virtuellen Netzwerk ist keine direkte Verbindung nach aussen oder zum Wirtsrechner möglich. Dazu wäre im virtuellen Netzwerk ein Router mit einer zusätzlichen Schnittstelle vom Typ „Intern“ oder „Extern“ notwendig.

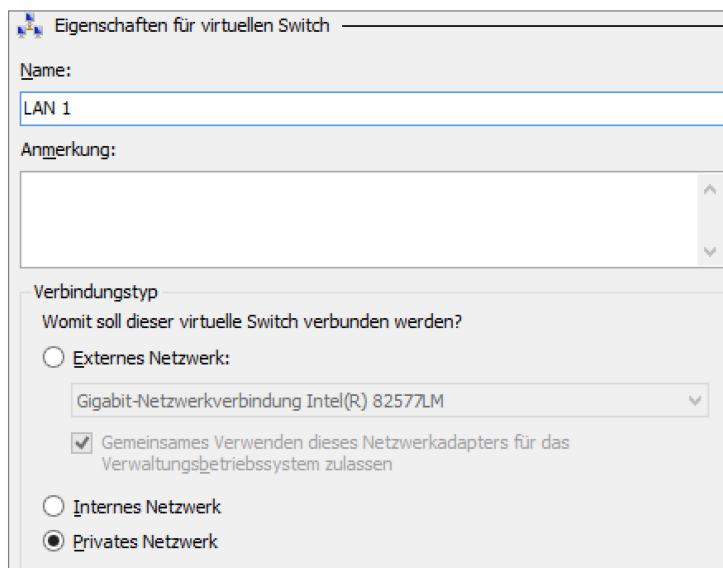
Für unsere Zwecke benötigen wir 2(3) Netzwerke vom Typ Privat.

Zunächst wird der Hyper-V-Manager aufgerufen (Suche nach Hyper-V). Im Aktionsfeld des Hyper-V-Managers wird dann der Punkt „Manager für virtuelle Switches“ aufgerufen. Es öffnet sich das entsprechende Fenster.



### Manager für virtuelle Switches

Wir richten zunächst einen virtuellen Switch vom Typ Privat ein. Dazu „Neuer virtueller Netzwerkswitch“ auswählen und im rechten Bereich „Privat“ auswählen und „Virtuellen Switch erstellen“ anklicken.



### Neuer virtueller Switch vom Typ Privat

Im folgenden Fenster wird ein Name vergeben, hier „LAN 1“ und unten rechts auf „Anwenden“ geklickt. Der neue virtuelle Switch wird erstellt. Den Vorgang für LAN 2 (und LAN 3) wiederholen und abschließend auf OK klicken.

## A.4 Virtuelle Maschinen konfigurieren

Nachdem die Netzwerke eingerichtet wurden können nun die virtuellen Maschinen erstellt werden. Dazu im Aktionsbereich des Hyper-V-Managers auf „Neu/Virtueller Computer“ klicken. Im den darauf folgenden Dialogen werden der Name der virtuellen Maschine, die Generation der VM, der Arbeitsspeicher, die Verbindung, virtuelle Festplatte und Installationsmedium, i.d.R. ein Installationsimage als ISO-Datei, nacheinander abgefragt. Weitere „Hardware“ wie z.B. Netzwerkarten können nach der Erstellung über die Einstellungen der jeweiligen VM hinzugefügt werden.



Der Name ist nur die Bezeichnung des Fensters in dem die VM läuft. Der Hostname wird im Rahmen der Installation bzw. Konfiguration vergeben. Generation 2 kann nur bei Betriebssystemen ab Windows 8/Server 2012 verwendet werden.

Folgende Rechner/VMs werden benötigt:

Alle Server 2048 GB RAM (min 1024), dynamisch, die Clients 1024 GB RAM dynamisch. Virtuelle Festplatten 128 GB dynamisch. Netzwerke laut Tabelle

VM-Name	Netzwerk
Server 1 (DNS, Active Directory)	LAN 1
Server 2 (Router)	LAN 1, LAN 2, ggf. LAN 3
Client 1	LAN 1
Client 2	LAN 1

Die Testumgebung ist beliebig erweiterbar.

Nach dem Einrichten kann die erste VM gestartet werden. Dazu wird im Hyper-V-Manager die betreffende VM markiert und über das Kontextmenü oder den Aktionsbereich verbunden und anschließend gestartet. Die Installation verläuft wie bei einem physischen Rechner.

Für IP-Adressen in den einzelnen Netzen gilt folgendes Schema

Name	Start IP	Netzmaske
LAN 1	172.16.20.0	255.255.255.0
LAN 2	172.16.30.0	255.255.255.0
LAN 3	172.16.40.0	255.255.255.0

Für die Konfiguration gilt folgende Aufteilung:

Funktion	Hostbyte	Dienste	
Server1	.10	AD, DNS, DHCP	Der DHCP Server wird im Verlauf der Übungen eingerichtet! Am Anfang sind alle IP-Adressen fest vergeben!
Server2 Standardgateway	.254	Router	im jeweiligen Netz

Als Second Level Domain wird **example.net** nach RFC 2606 verwendet.

Die IPv6-Adressen werden im Verlauf der Übungen konfiguriert.

Verwendet werden folgende IPv6-Präfixe:

Name	Präfix	
LAN 1	FD00:20:0:1::	/64
LAN 2	FD00:30:0:1::	/64
LAN 3	FD00:40:0:1::	/64

## A.5 Standard-Tools

Bei den Übungen werden grundlegende Netzwerktools verwendet. Damit kann die Netzwerkkonfiguration geprüft oder eingestellt werden.

In den Kapiteln des Buchs werden die hier beschriebenen Tools und die damit verbundenen Vorgehensweisen als bekannt vorausgesetzt. Die Tools werden direkt in der Eingabeaufforderung (Shell) ausgeführt.

- ✓ Ping, IPconfig
- ✓ Nslookup
- ✓ Netshell
- ✓ Wireshark

### Ping, IPconfig

Ping nutzt das ICMP-Protokoll und ermöglicht eine einfache Konnektivitätsüberprüfung. Diese sollte immer zu Beginn der Übungen durchgeführt werden, um sicherzustellen, dass die beteiligten Rechner im Netz erreichbar sind.

- Geben Sie in der Eingabeaufforderung folgenden Befehl ein:  
ping <IP-Adresse>.

Ohne weitere Optionen werden vier Datenpakete an den Zielrechner geschickt, die im Idealfall beantwortet werden. Dies funktioniert sowohl mit IPv4- als auch mit IPv6-Adressen, insofern die Betriebssystem-eigene Firewall dies nicht verhindert. Bei Verwendung von Hostnamen kann mit dem Schalter -4 bzw. -6 direkt nach dem ping Befehl IP4 bzw. IPv6 erzwungen werden.

```
C:\> ping fd00:a:b:c:5e49:79ff:fe7a:a1fb

Ping wird ausgeführt für fd00:a:b:c:5e49:79ff:fe7a:a1fb mit 32 Bytes Daten:
Antwort von fd00:a:b:c:5e49:79ff:fe7a:a1fb: Zeit=1ms
Antwort von fd00:a:b:c:5e49:79ff:fe7a:a1fb: Zeit=1ms
Antwort von fd00:a:b:c:5e49:79ff:fe7a:a1fb: Zeit=2ms
Antwort von fd00:a:b:c:5e49:79ff:fe7a:a1fb: Zeit=4ms

Ping-Statistik für fd00:a:b:c:5e49:79ff:fe7a:a1fb:
Pakete: Gesendet = 4, Empfangen = 4, Verloren = 0
(0% Verlust),
Ca. Zeitangaben in Millisek.:
Minimum = 1ms, Maximum = 4ms, Mittelwert = 2ms
```

*Ping an den Server über die IPv6-Adresse*

```
C:\>ping -6 ix.de
Ping wird ausgeführt für ix.de [2a02:2e0:3fe:1001:302::] mit 32 Bytes Daten:
Antwort von 2a02:2e0:3fe:1001:302::: Zeit=23ms
.....
.....
C:\>ping -4 ix.de
Ping wird ausgeführt für ix.de [193.99.144.80] mit 32 Bytes Daten:
Antwort von 193.99.144.80: Bytes=32 Zeit=22ms TTL=249
```

#### *Ping mit IPv6 / IPv4 erzwungen*

Sollte der Test scheitern, ist zunächst die Ursache zu suchen, bevor weitergehende Netzwerkkonfigurationen getestet werden. Findet sich die „angepingte“ IP-Adresse im ARP bzw Neighborcache des Hosts ist grundsätzliche Konnektivität (Layer 2) vorhanden, der Ping (Layer 3) wird aber durch eine Firewall verhindert.

- ▶ Zeigen Sie mit dem Befehl ipconfig /all detailliertere Informationen über die Konfiguration der Netzwerkschnittstellen an.

Ethernet-Adapter Ethernet:

```
Verbindungsspezifisches DNS-Suffix:
Beschreibung. . . . . : Microsoft Hyper-V Network Adapter
Physische Adresse . . . . . : 00-15-5D-37-23-19
DHCP aktiviert. . . . . : Nein
Autokonfiguration aktiviert . . . : Ja
IPv6-Adresse. . . . . : fd00:20:0:1::1(Bevorzugt)
Verbindungslokale IPv6-Adresse . . : fe80::b09c:b9fe:835b:65de%6
                                         (Bevorzugt)
IPv4-Adresse . . . . . : 172.16.20.10(Bevorzugt)
Subnetzmaske . . . . . : 255.255.255.0
Standardgateway . . . . . : fd00:20:0:1::254
                           172.16.20.1
DHCPv6-IAID . . . . . : 33559901
DHCPv6-Client-DUID. . . . . : 00-01-00-01-21-4F-29-A1-00-15-5D-37-23-19
DNS-Server . . . . . : ::1
                           127.0.0.1
NetBIOS über TCP/IP . . . . . : Aktiviertipconfig/all
```

#### *Ausgabe für die LAN-Verbindung*

### nslookup

**nslookup** ist ein Kommandozeilentool, das auf zwei Arten bedient werden kann.

- ✓ Im **Befehlszeilenmodus** werden Befehle direkt über die Eingabeaufforderung eingegeben.
- ✓ Im **interaktiven Modus** wird mit dem Befehl nslookup in den nslookup-Kontext gewechselt. Danach meldet sich das Programm mit IP-Adresse und Namen des Default-Nameservers.

Standardeinstellung ist die Abfrage von IP-Adressen (A, AAAA) oder Hostnamen (PTR). Für alle anderen Abfragen muss der Querytyp mit set q=<querytype> geändert werden. Gültige Werte für querytype sind z. B. A, AAAA, PTR, NS, SOA, MX, SRV.

Bei komplexeren Abfragen wie verschiedene Abfragen zu einer Zone empfiehlt es sich, im **interaktiven Modus** zu arbeiten.

- ▶ Wollen Sie z. B. den SOA Record einer DNS-Zone auflösen, geben Sie in der Eingabeaufforderung (interaktiver Modus) folgenden Befehl ein:  
`set q=SOA <Zonenname>`.

Wenn die Zone existiert, erhalten Sie eine bestätigende Antwort.

- ▶ Um anschließend die autoritativen Nameserver abzufragen, müssen Sie den Querytyp ändern:  
`set q=ns`

Eine erneute Eingabe der Zone liefert nun mindestens zwei Nameserver, die für die Zone zuständig sind.

- ▶ Wechseln Sie den Nameserver, der die Abfrage beantworten soll:  
`server <ip-ipadresse> oder <server hostname> (bzw. FQDN)`

Für einzelne Abfragen empfiehlt sich hier die erste Methode, der Aufruf von `nslookup` mit Parametern und Optionen. Dem Befehl `set q` im interaktiven Modus entspricht `-type` im Befehlszeilenmodus. Der folgende Screenshot zeigt die Vorgehensweise am beschriebenen Beispiel.

```
C:\>nslookup ix.de
Server:      UnKnown
Address:     fe80::5e49:79ff:fe7a:a1fb

Nicht autorisierende Antwort:
Name:        ix.de
Addresses:   2a02:2e0:3fe:1001:302::
              193.99.144.80

C:\>nslookup -type=A ix.de
Server:      UnKnown
Address:     fe80::5e49:79ff:fe7a:a1fb

Nicht autorisierende Antwort:
Name:        ix.de
Address:     193.99.144.80

C:\>nslookup -type=AAAA ix.de
Server:      UnKnown
Address:     fe80::5e49:79ff:fe7a:a1fb

Nicht autorisierende Antwort:
Name:        ix.de
Address:     2a02:2e0:3fe:1001:302::
```

*nslookup verschiedener Arten auf „www.heise.de“*

Je nach Programmversion werden neben den abgefragten Informationen noch zusätzliche Infos mit ausgegeben. `nslookup` ist ebenfalls geeignet, die Konnektivität innerhalb einer Domäne zu testen.



## netshell

Mit dem Tool **netshell** können Sie wie bei **nslookup** ebenfalls im Befehlszeilen- oder im interaktiven Modus arbeiten.

Im **Befehlszeilen-Modus** werden dem Befehl `netsh` alle benötigten Parameter und Optionen mitgegeben. Das empfiehlt sich, wenn ein einzelner Befehl ausgeführt werden soll.

- ▶ Geben Sie in der Konsole folgenden Befehl ein:  
`netsh interface ipv6 show addresses`

```
C:\>netsh interface ipv6 show addresses

Schnittstelle 1: Loopback Pseudo-Interface 1

AdressTyp DAD-Status Gültigkeit Bevorzugt Adresse
-----
Andere Bevorzugt infinite infinite ::1

Schnittstelle 6: Ethernet

AdressTyp DAD-Status Gültigkeit Bevorzugt Adresse
-----
Manuell Bevorzugt infinite infinite fd00:20:0:1::1
Andere Bevorzugt infinite infinite fe80::b09c:b9fe:835b:65de%6
```

*Einzelner netsh-Befehl, ohne in den Kontext zu wechseln*

In den **interaktiven Modus** wechseln Sie, indem Sie in der Eingabeaufforderung netsh eingeben. Die Eingabeaufforderung wechselt in den netshell-Kontext, erkennbar am Prompt netsh>. Der interaktive Modus ist in unterschiedlichen Hierarchieebenen organisiert, in die Sie durch Eingabe definierter Schlagworte wechseln können.

- ▶ Eine Liste der für die jeweilige Ebene gültigen Schlagworte erhalten Sie, indem Sie folgenden Befehl eingeben:  
help oder ?

Durch Eingabe eines einzelnen Punktes (.) gelangen Sie eine Ebene zurück. Die Bearbeitung der Befehle innerhalb eines Kontextes ist wesentlich handlicher, da nicht jedes Mal der gesamte netshell-Pfad mit eingegeben werden muss. Bei fehlerhafter Eingabe von Befehlen zeigt netshell die korrekte Syntax und Beispiele.

- ▶ Wechseln Sie in den netsh-Kontext:  
netsh
- ▶ Wechseln Sie in den gewünschten (Sub-)Kontext:  
interface
- ▶ Wechseln Sie in den gewünschten (Sub-)Kontext:  
ipv6
- ▶ Lassen Sie sich die Befehle des (Sub-)Kontext ipv6 anzeigen:  
?



Ab Windows 8 erscheint ein Hinweis „In zukünftigen Versionen von Windows wird die Netsh-Funktionalität möglicherweise nicht mehr für "TCP/IP" verfügbar sein.“ mit einem Verweis auf die aktuellen PowerShellbefehle.

```
C:\>netsh
netsh>interface
netsh interface>ipv6
netsh interface ipv6>

Befehle in diesem Kontext:
6to4           - Wechselt zum "netsh interface ipv6 6to4"-Kontext.
?              - Zeigt eine Liste der Befehle an.
add            - Fügt einen Konfigurationseintrag zu einer Tabelle hinzu.
delete         - Löscht einen Konfigurationseintrag aus einer Tabelle.
dump           - Zeigt ein Konfigurationsskript an.
help           - Zeigt eine Liste der Befehle an.
isatap          - Wechselt zum "netsh interface ipv6 isatap"-Kontext.
reset          - Setzt die IP-Konfigurationen zurück.
set            - Legt Konfigurationsinformationen fest.
show           - Zeigt Informationen an.
```

*Wechsel durch den netsh-Kontext*

In diesem Kontext können Sie sich nun z. B. die Interface-Konfigurationen anzeigen lassen.

**netsh interface ipv6> show addresses**

```
Schnittstelle 1: Loopback Pseudo-Interface 1

Adresstyp DAD-Status Gültigkeit Bevorzugt Adresse
-----
Andere Bevorzugt infinite infinite ::1

Schnittstelle 6: Ethernet

Adresstyp DAD-Status Gültigkeit Bevorzugt Adresse
-----
Manuell Bevorzugt infinite infinite fd00:20:0:1::1
Andere Bevorzugt infinite infinite fe80::b09c:b9fe:835b:65de%6
```

*Der Befehl show addresses aus dem netsh-Kontext*

**netsh interface ipv6>show addresses Ethernet**

```
Adresse fd00:20:0:1::1 Parameter
-----
Schnittstellen-LUID : Ethernet
Bereichskennung : 0.0
Gültigkeitsdauer : infinite
Bevorzugte Gültigkeitsdauer : infinite
DAD-Status : Bevorzugt
Adresstyp : Manuell
Als Quelle überspringen : false

Adresse fe80::b09c:b9fe:835b:65de%6 Parameter
-----
Schnittstellen-LUID : Ethernet
Bereichskennung : 0.6
Gültigkeitsdauer : infinite
Bevorzugte Gültigkeitsdauer : infinite
DAD-Status : Bevorzugt
Adresstyp : Andere
Als Quelle überspringen : false
```

*Detaillierte Anzeige der Adressen der Schnittstelle Ethernet*

Netshell ergänzt abgekürzte Befehle bei Eindeutigkeit intern automatisch. So wird z. B. der gekürzt eingegebene Befehl sh add ausgeführt als show addresses. Aus Gründen der Übersichtlichkeit wird im Buch auf die verkürzte Schreibweise verzichtet.



Netshell ermöglicht auch eine einfache Sicherung und das Zurückspielen der Netzwerkkonfiguration, was insbesondere in Testsituationen wiederholte Konfigurationsarbeit ersparen kann.

- ▶ Sichern der aktuellen Interface-Einstellung für IPv6 in die Datei myconfig6.netsh:  
netsh interface ipv6 dump >myconfig6.netsh
- ▶ Zurückschreiben der Konfiguration:  
netsh -f myconfig6.netsh

Die Sicherungsdatei wird in dem Ordner gespeichert, aus dem heraus der Befehl abgesetzt wurde, standardmäßig im Benutzerprofil des angemeldeten Users. Der Befehl sollte aus einer Shell mit Administratorrechten ausgeführt werden.

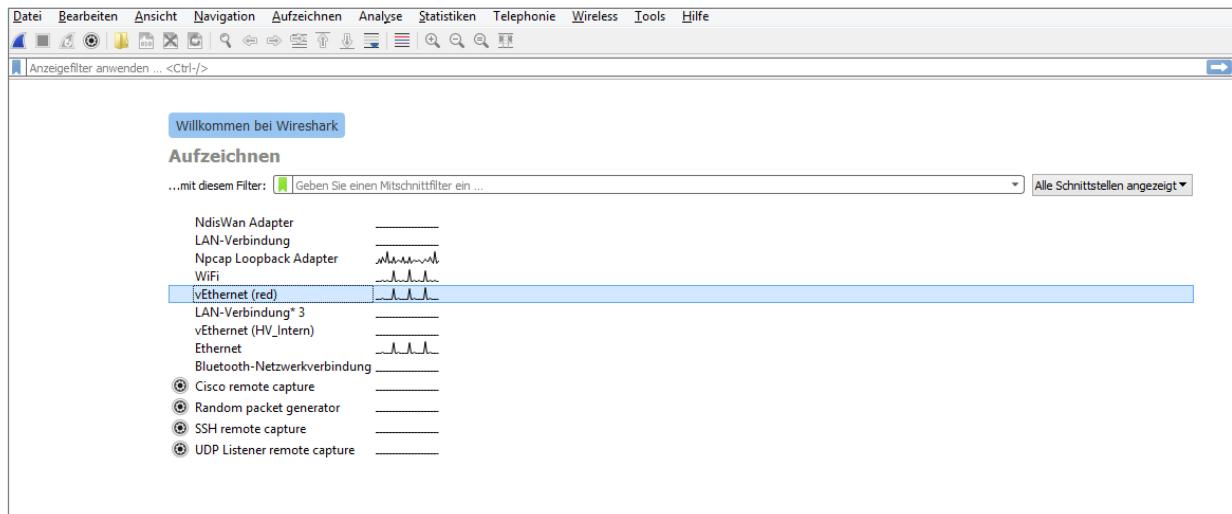
Eine Auflistung der entsprechenden Linuxbefehle finden Sie im Linux IPv6 HowTo von Peter Bieringer unter <http://www.tldp.org/HOWTO/Linux+IPv6-HOWTO/index.html>, hier insbesondere in den Kapiteln 6 und 7.



## Wireshark

Mit dem Werkzeug Wireshark (früher Ethereal) können Datenpakete an den Netzwerkschnittstellen eines Rechners mitgeschnitten umfangreich gefiltert und ausgewertet werden.

Aktuelle Versionen und Onlinedokumentation unter <https://www.wireshark.org/>



Screenshot Wireshark Startseite

## Netshell versus PowerShell

In den aktuellen Versionen der Microsoft PowerShell haben neue Befehle Einzug gehalten, die den im Buch verwendeten Netshell-Befehlen entsprechen und diese zukünftig ablösen sollen. Da der Umgang mit der PowerShell komplexer ist als die hier verwendeten cmd oder Netshellbefehle, wurde im Buch auf die Verwendung der PowerShell verzichtet. Eine Einführung in die PowerShell bietet z. B. die HERDTunterlage *PowerShell 5 – Grundlagen und Verwaltung des Active Directory*.

In Kurzform sollen einige der verwendeten cmd und netshell Kommandos in der PowerShell-Variante gezeigt werden.

Ping	Test-Connection, Test-NetConnection																														
<pre>PS C:\&gt; Test-Connection ix.de</pre> <table border="1"> <thead> <tr> <th>Source</th><th>Destination</th><th>IPV4Address</th><th>IPV6Address</th><th>Bytes</th><th>Time (ms)</th></tr> </thead> <tbody> <tr> <td>DENKSTATION</td><td>ix.de</td><td>193.99.144.80</td><td>2a02:2e0:3fe:1001:302::</td><td>32</td><td>23</td></tr> <tr> <td>DENKSTATION</td><td>ix.de</td><td>193.99.144.80</td><td>2a02:2e0:3fe:1001:302::</td><td>32</td><td>22</td></tr> <tr> <td>DENKSTATION</td><td>ix.de</td><td>193.99.144.80</td><td>2a02:2e0:3fe:1001:302::</td><td>32</td><td>21</td></tr> <tr> <td>DENKSTATION</td><td>ix.de</td><td>193.99.144.80</td><td>2a02:2e0:3fe:1001:302::</td><td>32</td><td>22</td></tr> </tbody> </table>	Source	Destination	IPV4Address	IPV6Address	Bytes	Time (ms)	DENKSTATION	ix.de	193.99.144.80	2a02:2e0:3fe:1001:302::	32	23	DENKSTATION	ix.de	193.99.144.80	2a02:2e0:3fe:1001:302::	32	22	DENKSTATION	ix.de	193.99.144.80	2a02:2e0:3fe:1001:302::	32	21	DENKSTATION	ix.de	193.99.144.80	2a02:2e0:3fe:1001:302::	32	22	
Source	Destination	IPV4Address	IPV6Address	Bytes	Time (ms)																										
DENKSTATION	ix.de	193.99.144.80	2a02:2e0:3fe:1001:302::	32	23																										
DENKSTATION	ix.de	193.99.144.80	2a02:2e0:3fe:1001:302::	32	22																										
DENKSTATION	ix.de	193.99.144.80	2a02:2e0:3fe:1001:302::	32	21																										
DENKSTATION	ix.de	193.99.144.80	2a02:2e0:3fe:1001:302::	32	22																										

Tracert	Test-NetConnection mit dem Schalter -TraceRoute
---------	---

```
PS C:\> Test-NetConnection -ComputerName ix.de -TraceRoute

ComputerName      : ix.de
RemoteAddress    : 2a02:2e0:3fe:1001:302::
InterfaceAlias   : vEthernet (red)
SourceAddress    : 2003:5e:c5e:f00:aca4:8a7:f88:f2d9
PingSucceeded    : True
PingReplyDetails (RTT) : 22 ms
TraceRoute       : 2003:0:3f02:203::1
                  2003:0:3f02:203::1
                  2003:0:3f02:22d::2
                  2003:0:3f02:c3::1
TimedOut         : 2003:0:1306:400c::2
TimedOut         : 2a02:2e0:12:31::2
TimedOut         : 2a02:2e0:3fe:0:c::1
TimedOut         : 2a02:2e0:3fe:1001:302::
```

Ipconfig	Get-NetIPConfiguration
----------	------------------------

```
PS C:\> Get-NetIPConfiguration

InterfaceAlias      : vEthernet (red)
InterfaceIndex      : 21
InterfaceDescription: Hyper-V-Adapter - virtuelles Ethernet #2
NetProfile.Name     : FRITZ!Box 7490
IPv6Address         : fd00:a:b:c:aca4:8a7:f88:f2d9
                      2003:5e:c5e:f00:aca4:8a7:f88:f2d9
IPv4Address         : 192.168.55.133
IPv6DefaultGateway : fe80::5e49:79ff:fe7a:a1fb
IPv4DefaultGateway : 192.168.55.1
DNSServer          : fe80::5e49:79ff:fe7a:a1fb
                      192.168.55.1

InterfaceAlias      : vEthernet (HV_Intern)
InterfaceIndex      : 27
InterfaceDescription: Hyper-V-Adapter - virtuelles Ethernet #3
.....
....
```

Mit dem Schalter –InterfaceIndex kann die Ausgabe auf eine Netzwerkschnittstelle begrenzt werden

```
PS C:\> Get-NetIPConfiguration -InterfaceIndex 3

InterfaceAlias      : WiFi
InterfaceIndex      : 3
InterfaceDescription: Intel(R) Centrino(R) Advanced-N 6200 AGN
NetAdapter.Status   : Disconnected
```

Netsh interface [ipv4 ipv6] show interface	Get-NetAdapter / Get-NetIPInterface																									
PS C:\> Get-NetAdapter	<table> <thead> <tr> <th>Name</th><th>InterfaceDescription</th><th>ifIndex</th><th>Status</th><th>MacAddress</th></tr> </thead> <tbody> <tr> <td>vEthernet (HV_Intern)</td><td>Hyper-V-Adapter ....#3</td><td>27</td><td>Up</td><td>00-15-..</td></tr> <tr> <td>vEthernet (red)</td><td>Hyper-V-Adapter ....#2</td><td>21</td><td>Up</td><td>F0-DE-..</td></tr> <tr> <td>Bluetooth-Netzwerkverb.</td><td>Bluetooth-Gerät (PAN)</td><td>5</td><td>Disconnected</td><td>C4-17-..</td></tr> <tr> <td>Ethernet</td><td>Gigabit-Netzwerkver</td><td>4</td><td>Up</td><td>F0-DE-..</td></tr> </tbody> </table>	Name	InterfaceDescription	ifIndex	Status	MacAddress	vEthernet (HV_Intern)	Hyper-V-Adapter ....#3	27	Up	00-15-..	vEthernet (red)	Hyper-V-Adapter ....#2	21	Up	F0-DE-..	Bluetooth-Netzwerkverb.	Bluetooth-Gerät (PAN)	5	Disconnected	C4-17-..	Ethernet	Gigabit-Netzwerkver	4	Up	F0-DE-..
Name	InterfaceDescription	ifIndex	Status	MacAddress																						
vEthernet (HV_Intern)	Hyper-V-Adapter ....#3	27	Up	00-15-..																						
vEthernet (red)	Hyper-V-Adapter ....#2	21	Up	F0-DE-..																						
Bluetooth-Netzwerkverb.	Bluetooth-Gerät (PAN)	5	Disconnected	C4-17-..																						
Ethernet	Gigabit-Netzwerkver	4	Up	F0-DE-..																						

Eingeschränkt auf Schnittstelle 4, Ausgabe als Liste

PS C:\> Get-NetAdapter -ifindex 4   fl																																				
<table> <tbody> <tr> <td>Name</td><td>:</td><td>Ethernet</td></tr> <tr> <td>InterfaceDescription</td><td>:</td><td>Gigabit-Netzwerkverbindung Intel(R) 82577LM</td></tr> <tr> <td>InterfaceIndex</td><td>:</td><td>4</td></tr> <tr> <td>MacAddress</td><td>:</td><td>F0-DE-F1-17-DF-60</td></tr> <tr> <td>MediaType</td><td>:</td><td>802.3</td></tr> <tr> <td>PhysicalMediaType</td><td>:</td><td>802.3</td></tr> <tr> <td>InterfaceOperationalStatus</td><td>:</td><td>Up</td></tr> <tr> <td>AdminStatus</td><td>:</td><td>Up</td></tr> <tr> <td>LinkSpeed(Gbps)</td><td>:</td><td>1</td></tr> <tr> <td>MediaConnectionState</td><td>:</td><td>Connected</td></tr> <tr> <td>ConnectorPresent</td><td>:</td><td>True</td></tr> <tr> <td>DriverInformation</td><td>:</td><td>Driver Date 2013-03-28 Version 12.6.47.1 NDIS 6.30</td></tr> </tbody> </table>	Name	:	Ethernet	InterfaceDescription	:	Gigabit-Netzwerkverbindung Intel(R) 82577LM	InterfaceIndex	:	4	MacAddress	:	F0-DE-F1-17-DF-60	MediaType	:	802.3	PhysicalMediaType	:	802.3	InterfaceOperationalStatus	:	Up	AdminStatus	:	Up	LinkSpeed(Gbps)	:	1	MediaConnectionState	:	Connected	ConnectorPresent	:	True	DriverInformation	:	Driver Date 2013-03-28 Version 12.6.47.1 NDIS 6.30
Name	:	Ethernet																																		
InterfaceDescription	:	Gigabit-Netzwerkverbindung Intel(R) 82577LM																																		
InterfaceIndex	:	4																																		
MacAddress	:	F0-DE-F1-17-DF-60																																		
MediaType	:	802.3																																		
PhysicalMediaType	:	802.3																																		
InterfaceOperationalStatus	:	Up																																		
AdminStatus	:	Up																																		
LinkSpeed(Gbps)	:	1																																		
MediaConnectionState	:	Connected																																		
ConnectorPresent	:	True																																		
DriverInformation	:	Driver Date 2013-03-28 Version 12.6.47.1 NDIS 6.30																																		

Erweiterte Ausgabe und weitere Filter sind möglich

Netstat	Get-NetTCPConnection																																													
PS C:\> Get-NetTCPConnection -state established	<table> <thead> <tr> <th>LocalAddress</th><th>LocalPort</th><th>RemoteAddress</th><th>RemotePort</th><th>State</th></tr> </thead> <tbody> <tr> <td>2003:5e:c5e:f00:aca4:8a7:f88:f2d9</td><td>18660</td><td>2a00:1450:400c:c06::10</td><td>993</td><td>Established</td></tr> <tr> <td>2003:5e:c5e:f00:d4dd:899e:9364:6dd6</td><td>7246</td><td>2a01:4f8:b16:1000::28</td><td>445</td><td>Established</td></tr> <tr> <td>192.168.55.133</td><td>59387</td><td>192.168.55.114</td><td></td><td>3389</td></tr> <tr> <td>192.168.55.133</td><td>59053</td><td>144.76.175.90</td><td></td><td>143</td></tr> <tr> <td>192.168.55.133</td><td>58064</td><td>23.97.215.12</td><td></td><td>443</td></tr> <tr> <td>192.168.55.133</td><td>57943</td><td>144.76.175.90</td><td></td><td>143</td></tr> <tr> <td>192.168.55.133</td><td>57942</td><td>144.76.175.89</td><td></td><td>993</td></tr> <tr> <td>Established</td><td></td><td></td><td></td><td></td></tr> </tbody> </table>	LocalAddress	LocalPort	RemoteAddress	RemotePort	State	2003:5e:c5e:f00:aca4:8a7:f88:f2d9	18660	2a00:1450:400c:c06::10	993	Established	2003:5e:c5e:f00:d4dd:899e:9364:6dd6	7246	2a01:4f8:b16:1000::28	445	Established	192.168.55.133	59387	192.168.55.114		3389	192.168.55.133	59053	144.76.175.90		143	192.168.55.133	58064	23.97.215.12		443	192.168.55.133	57943	144.76.175.90		143	192.168.55.133	57942	144.76.175.89		993	Established				
LocalAddress	LocalPort	RemoteAddress	RemotePort	State																																										
2003:5e:c5e:f00:aca4:8a7:f88:f2d9	18660	2a00:1450:400c:c06::10	993	Established																																										
2003:5e:c5e:f00:d4dd:899e:9364:6dd6	7246	2a01:4f8:b16:1000::28	445	Established																																										
192.168.55.133	59387	192.168.55.114		3389																																										
192.168.55.133	59053	144.76.175.90		143																																										
192.168.55.133	58064	23.97.215.12		443																																										
192.168.55.133	57943	144.76.175.90		143																																										
192.168.55.133	57942	144.76.175.89		993																																										
Established																																														

Netsh interface ipv6 add address	New-NetIPAddress
<pre>PS C:\&gt; New-NetIPAddress -InterfaceIndex 21 -IPAddress fd00:a:b:c::1 -PrefixLength 64   IPAddress      : fd00:a:b:c::1  InterfaceIndex : 21  InterfaceAlias : vEthernet (red)  AddressFamily   : IPv6  .....</pre>	

New-NetIPAdress erstellt eine neue IP-Adresse, Set-NetIPAddress ändert und Remove-NetIPAddress löscht eine Adresse

Netsh interface IPv6 show route	Get-NetRoute
	<pre>PS C:\&gt; Get-NetRoute -AddressFamily ipv6 -InterfaceIndex 21  ifIndex DestinationPrefix          NextHop RouteMetric PolicyStore ----- ----- ----- ----- 21      ff00::/8                  ::: 256 ActiveStore 21      fe80::aca4:8a7:f88:f2d9/128    ::: 256 ActiveStore 21      fe80::/64                  ::: 256 ActiveStore 21      fd00:a:b:c:d4dd:899e:9364:6dd6/128    ::: 256 ActiveStore 21      fd00:a:b:c:b53f:8510:2bf1:9f14/128    ::: 256 ActiveStore 21      fd00:a:b:c:aca4:8a7:f88:f2d9/128    ::: 256 ActiveStore 21      fd00:a:b:c::/64                fe80::5e49:79ff:fe7a:a1fb 16 ActiveStore 21      fd00:a:b:c::/64                ::: 16 ActiveStore 21      2003:5e:c5e:f00:d4dd:899e:9364:6dd6/128  ::: 256 ActiveStore 21      2003:5e:c5e:f00:b53f:8510:2bf1:9f14/128  ::: 256 ActiveStore 21      2003:5e:c5e:f00:aca4:8a7:f88:f2d9/128    ::: 256 ActiveStore 21      2003:5e:c5e:f00::/64                ::: 16 ActiveStore 21      2003:5e:c5e:f00::/56                fe80::5e49:79ff:fe7a:a1fb 16 ActiveStore 21      ::/0                                fe80::5e49:79ff:fe7a:a1fb 16 ActiveStore</pre>

Im Beispiel wurde die Ausgabe auf IPv6 und Interface 21 eingeschränkt.  
Hinzufügen, Ändern und Löschen einer Route mit New-, Set-, Remove-NetRoute

Nslookup	Resolve-DnsName
----------	-----------------

```
PS C:\> Resolve-DnsName ix.de

Name                           Type      TTL     Section      IPAddress
----                           ----      --      -----      -----
ix.de                         AAAA    66844  Answer
2a02:2e0:3fe:1001:302::       A        66844  Answer      193.99.144.80
ix.de
```

arp	netsh interface ipv6show neighbors	Get-NetNeighbor
-----	------------------------------------	-----------------

```
PS C:\> Get-NetNeighbor -AddressFamily ipv6 -InterfaceIndex 21

ifIndex  IPAddress           LinkLayerAddress      State      PolicyStore
-----  -----           -----           -----
21      ff0e::c            33330000000c      Permanent  ActiveStore
21      ff02::1:ffff1:9f14  3333ffff19f14    Permanent  ActiveStore
.....
```

*Neighborcache nach IPv6 und Schnittstelle 21 gefiltert*



**6**

6in4-Tunnel	109, 110
6rd	109
6to4-Router	111
6to4-Tunnel	109, 111

**A**

Access-Lists	129
Active Directory-Struktur	132
AD-Domäne	127
admin-lokal	45
Adressbereiche	40
Addressierung	34
Addressierung, klassenlose	23
Adressknappheit	30, 128
Adresslänge	28, 31
AfriNIC	39
AICCU (Automatic IPv6 Connectivity Client Utility)	110
Android	124
Angriffsszenarien	122
Anonymität	123
Anwendungsschicht	8
Anycast	29, 46
Anycast-Adresse	46
APIPA	41, 131
APNIC	39
Application Layer	8
ARIN	39
ARP	9, 46, 130
ARP Cache Poisoning	15
ARP Spoofing	133
ARP, Gratuitous	14
ARPA	7
Arpanet	7
AS-Nummern	87
Asynchronous Transfer Mode (ATM)	79
Ausfallsicherheit	29, 46
Authentication Header	126
Authentizität	126
Autokonfiguration	29, 41, 132
Automatische Konfiguration	52
Autonome Systeme (AS)	86
Autonomous System Number (ASN)	86
Avahi	131
AYIYA (Anything In Anything)	110

**B**

Betriebssysteme	124
BGP	86, 88

BGP (Border Gateway Protocol)	86	DNS-Grundlagen IPv6	58
BGP-4+	99	DNSSec	135
BGP-Router	29	DNS-Zone	132
Bidirectional Tunneling	137	DoD	7
BIND9	132	DoD-Modell	126
Binding Update	137	Dual Stack	6
Bitübertragungsschicht	8	Dual Stack Node	47
Blacklisting	134	Duales Zahlensystem	49
Bootstrap Protocol (BOOTP).	56	Dual-Stack	106
Border Gateway Protocol	29	Dual-Stack Lite	107
Broadcast	29, 31, 44, 46	Dual-Stack-Host	106
Broadcast-Adresse	17	Dual-Stack-Router	106
Broadcast-Attacken	133	DUID	131
Broadcast-Domain	82		

**C**

Care-of-Address	137	EGP (Exterior Gateway-Protokolle)	86, 88
Carrier Sense (CS)	78	EIGRPv6	99
CGN - Carrier-grade NAT	107	Einmalige Adresse	34
Checksumme	31	Encapsulated Security Payload	126
CIDR	23	Ende-zu-Ende-Beziehung	129
CIDR-Notation	36	Ende-zu-Ende-Konzept	30
Client-Server	84	Ende-zu-Ende-Modus	126
Collision Detection (CD)	78	Ethernet Frame	79
Count-to-Infinity-Problematik	92	EUI-64	37, 123
Custom-ROM	124	Everything over IP	78
Cyanogenmod	124	Extension Header-Attacken	135
		Extension Headers	127
		Exterior Gateway Protocol	88

**D**

Darstellungsschicht	8	Fault-Tolerance	46
Data Link Layer	8	Fehlernachrichten ICMPv6	72
Datagramm	76	Fibre Distributed Data Interface (FDDI)	80
DCHP-Solicit-Message	57	Filterregeln	129
Dead Peer Detection	127	Firewall	123
Default Gateway	83	Flag	32, 44
Denial of Service	134	Flow Label	32
Destination Address	12	Forward Lookup	59
Destination IP	88	Fragment Offset	11, 12, 32
DHCP	57	Fragmentierung	31, 77
DHCP-Advertise-Message	57	Frame Relay	80
DHCP-Reconfigure	57	Fritz!Box	110
DHCP-Release	57	Fully specified	93
DHCP-Reply	57	Funktionsweise des Routing	88
DHCP-Request	57		
DHCP-Server	41		
DHCP-Spoofing	131		
DHCP-Starvation	131		
DHCPv6	55, 130, 133		
Direct Access	94		
Directly attached	93		
Distance Vector Routing	89, 90	Gefährdung durch IPv6	123
Distanz (Routing)	90	Gerätenummer	37
DNS	31	Global Routing Präfix	38
		Globaler Multicast	45

**G**

Global-Unicast-Adressen	43	IP	6	Maschen-Topologie	85
Grey-Listing	134	IP over Everything	78	Maximum Hop Count	91
		ip6tables	129, 130	Maximum Segment Size (MSS)	76
		IP-Adressheader	29	Maximum Transmission	
		IPconfig	147	Unit (MTU)	76
		IPsec	30, 31, 126, 128	Migrationsszenarien	141
		IPv4	6, 28	MLD	31
		IPv4-kompatible Adresse	47	Mobile Geräte	123
		IPv4-zugeordnete Adresse	48	Mobile IP	29
		IPv5	30	Mobile IPv6	136
		IPv6	6, 28	Mobile IPv6, Funktionsweise	137
		IPv6 im Firmennetz	141	Mobile IPv6, Sicherheit	140
		IPv6 im LAN	141	Mobility Header	139
		IPv6-Präfix	34	MTU	80
		ISAKMP	127	mtupath.exe	77
		ISATAP	110	Multi Protocol Label	
		ISATAP-Tunnel	109	Switching (MPLS)	80
		IS-IS	88	Multicast	29, 31, 44, 46, 132
		ISO/EIC 10589	99	Multicast-Gruppen	44
		ISP	38	Multihomed AS	87
				Multihomed Stub-AS	87
				Multihoming	29
<b>K</b>					
		Kerberos	127	<b>N</b>	
		Kompatibilitätsadressen	47	NAT	123, 128, 129, 135
		Konfiguration statischer Routen	97	NAT64	121
		Konfiguration statischer Routen, MS Server	94	NAT66	121
		Konvergenz (Netzwerk)	90	NDP	46, 130, 133
		Kosten (Routing)	89	NDP-Spoofing	133
		Kunde	87	Neighbor Advertisement	73
				Neighbor Advertisement Spoofing	134
<b>L</b>					
		L2TP	128	Neighbor Discovery	41
		LACNIC	39	Neighbor Solicitation	73
		Informationsmeldungen ICMPv6	73	Neighbor-Discovery-Protokoll	133
		Integrität	126	Neighborhood Discovery Protocol	9
		Inter-AS-Routing	87	Netscan	134
		Interface-ID	37	netshell	149
		Interface-Identifier	34	Network Access Layer	8
		interface-lokal	45	Network Address Translation	135
		Interior Gateway Protocols	88	Network Address Translation – Protocol Translation (NAT-PT)	121
		Internet Header Length (IHL)	11	Network Layer	8
		Internet Key Exchange	126	Netzbereich	36
		Internet Layer	8	Netzkennung	125
		Internet Security Association and Key Management Protocol	127	Netz-Notation	36
		Internet-Key-Exchange	126	Netzwerkkennung	17
		Internet-Protokoll	6	Netzwerkklassen	21
		Internetschicht	8	Netzwerkmodelle	83
		InterNIC	10	Netzzugangsschicht	8
		Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)	112	Neuerungen	28
iOS	125	MAC-Adresse	37, 41	Next Header	32
		Man in the Middle	132, 134	Next Hop	88
		Managed Address Configuration	130	NRO	40
				nslookup	148

**O**

OAKLEY	127
OLSR	99
OPTION_AUTH	131
Options	31
organisations-lokal	45
OS X	124
OSI-Referenzmodell	7
OSPF	88
OSPF (Open Shortest Path First Protocol)	86
OSPFv3	99, 101
Other Configuration	130
OtherConfig	132

**R**

Recursive	93	RIPng for IPv6	99
Reginal Internet Registries (RIR)	86	RIPv2	100
Replay-Schutz	131	RIR	38, 39
Ressource Records (RR)	58	Route Optimization	137
Reverse Lookup	60	Route Poisoning	91
RFC 114	9	Router	
RFC 2002	136	Advertisement	54, 73, 130, 132
RFC 2049	127	Router Advertisement Spoofing	134
RFC 2080	99, 100	Router Solicitation	73, 130, 132
RFC 2373	44	Routing	82, 86
RFC 2461	94	Routing- und RAS-Dienst	
RFC 2464	78	einrichten	96
RFC 2474	11	Routing- und RAS-Dienst	
RFC 2526	46	installieren	94
RFC 2545	99	Routing-Tabellen	83
RFC 2740	99		

**P**

Paketfilter	129
Paketfilter-Firewall	129
Paketfiltersysteme	123
Path MTU Discovery (PMTUD)	77
Path Vector Routing	89, 92
Payload	11
Peers	87
Peer-to-Peer	84
Physical Layer	8
Ping	147
Point to Point (PPP)	80
Point to Point Protocol over Ethernet (PPPoE)	80
Poison Reverse	91
Portforwarding	128
Portscan	134
Präfix	41, 48
Praxisbeispiel Konfiguration des DHCPv6-Servers unter MS Windows Server 2016	69
Praxisbeispiel Stateless Autoconfiguration	61
Praxisbeispiel Stateless Auto-configuration mit DHCPv6	67
Presentation Layer	8
Preshared Keys, PSK	127
Privacy Extensions	38, 123, 125
Private IP	29
Privatsphäre	123
Privatsphäre, Schutz	125
Protocol	12
Protokoll Translation	135
Provider	39, 87
Providerunabhängige Adressen	29
Punkt-zu-Punkt-Topologie	84

**S**

RFC 2766	121, 135	SAD	126
RFC 2858	99	Schlüssel	127
RFC 3626	99	Schutz, Privatsphäre	125
RFC 3879	42	Schwachstellen, Internet	122
RFC 3971	133	Scope	45
RFC 4193	42	Scope-ID	41
RFC 4291	36, 41, 53	Security Association	127
RFC 4306	127	Security Association Database	126
RFC 4861	53, 72	Security Policy Database	126
RFC 4862	53	SEND	133
RFC 4884	72	Server schützen	123
RFC 4890	130	Session Layer	8
RFC 4941	38, 123	Sicherheitsansprüche	126
RFC 4966	121, 135	Sicherheitsmechanismen, IPv6	122
RFC 5006	55, 132	Sicherheitsrichtlinien	122
RFC 5969	112	Sicherungsschicht	8
RFC 6145	121	SIT, Stateless IP/ICMP	121
RFC 6275	136	Site-Local- Adressen	42
RFC 6296	121	site-lokal	45
RFC 6791	121	Sitzungsschicht	8
RFC 6890	48	SKEME	127
RFC 6918	23	SLAAC	53, 131
RFC 7136	36	Smartphones	123
RFC 791	12, 16, 21	Smurf-Attacke	133
RFC 792	15	Source Address	12
RFC 8106	55	Source-Routing	129
RFC 8200	33	SPAM	134
RFC 826	13	SPD	126
RFC 950	24	Split Horizon	91
RFC 959	9	SSL	126
RIP (Routing Information Protocol)	86, 88	Standard	28
RIPE	125	Stateful Address	
RIPE NCC	39	Autoconfiguration	130
RIPng	100	Stateful Autoconfiguration	52, 55
		Stateful Inspection	129

Stateless Address Autoconfiguration	53	Time to Live	12, 32	<b>V</b>
Stateless Autoconfiguration	52, 53, 132	Topologien	82, 84	Vektor (Routing) 90
Stateless Automatic Autoconfiguration	131	Token Ring/Token Pass	79	Verbindung, unterbrechungsfreie 136
Stateless-DHCP	53	Total Length	11, 32	Vereinfachtes Routing 30
Statische Konfiguration	52	Transit-AS	87	Vermittlungsschicht 8, 28
Statische und dynamische Routen	83	Transport Layer	8	Vertraulichkeit 126
Statisches und dynamisches Routing, Vergleich	92	Transportschicht	8	VPN 128
Stern-Topologie	85	Triggered Updates	92	
Stub-AS	87	Tunnel	135	
Subnetting	23, 48	Tunneling	106	
Subnetz	34, 48	Tunnelmodus	126	<b>W</b>
Subnetzmaske	17	Type of Service (ToS)	11, 32	Webcode 5
Subnetz-Präfix	46			Webhosting 126
		Übersetzungsverfahren	121	Whois 39, 125
		Übungsdatei	5	Wichtige Ressource Records 60
		UDP	9, 126	Wireshark 74, 152
		ULA	42	WLAN 123
		UMTS	123	
		Unicast	46	<b>X</b>
		Unicast-Adressen	41	X.509-Zertifikate 127
		Unique Local	42	
		Unique-Lokale-Adressen	29	
		Unspecified Address	46	<b>Z</b>
		Unterscheidung statischer Routen	93	Zone-ID 41
		URL-Notation	36	Zufällige Präfixe 125
				Zurückverfolgen eines Anschlusses 125

**T**

TCP/IP	7
Teredo	110
Teredo-Client	116
Teredo-Server	116
Teredo-Tunnel	94, 109, 116
Testumgebung	144
Tier-1 Provider	87

**U**

Übersetzungsverfahren	121
Übungsdatei	5
UDP	9, 126
ULA	42
UMTS	123
Unicast	46
Unicast-Adressen	41
Unique Local	42
Unique-Lokale-Adressen	29
Unspecified Address	46
Unterscheidung statischer Routen	93
URL-Notation	36

**W**

Webcode	5
Webhosting	126
Whois	39, 125
Wichtige Ressource Records	60
Wireshark	74, 152
WLAN	123

**X**

X.509-Zertifikate	127
Zone-ID	41
Zufällige Präfixe	125
Zurückverfolgen eines Anschlusses	125