

IPv6- und Ipv4-Adressierung im LAN und Internet



1. Was ist ein "Dual-Stack für TCP/IP" ?

Ein Protokoll-Treibermodul oder Protokoll-Objekt, das sowohl die Verarbeitung und Adressierung von Internet- und LAN-Adressen nach dem Standard von IPv4 und zugleich von IPv6 erlaubt!

2. Was sind die Vorteile von IPv6 gegenüber von IPv4?

- ✓ *unvorstellbare Größe des Adressraumes*
- ✓ *selbstständige Adressvergabe der Netzteilnehmer bei Eintritt ins Netzwerk - ein DHCP-Server wird überflüssig!*
- ✓ *selbstständige Adressierung des eigenen Interfaces inklusive Überprüfung ob ein anderer IPv6-Teilnehmer die gleiche Adresse besitzt.*
- ✓ *automatische Unterscheidung zwischen Router-und Host-Adressesierung (Knoten).*
- ✓ *ipsec ist automatisch enthalten.*

3. Was zeigen die Linux-Befehle: /sbin/ifconfig und alternativ /sbin/ip -6 addr show?

Es werden acht Viererblöcke an Hexziffern, die durch Doppelpunkt von einander getrennt sind, dargestellt. Damit wird auch gleichzeitig rückgemeldet, dass die Funktionalität von IPv6 zur Verfügung steht.

4. Was ist eine "verbindungslokale IPv6-Adresse"?

Hierbei handelt es sich um eine Adresse, die ähnliche Funktionen hat wie eine lokale loopback-Adresse in IPv4! Mit dieser Adresse kann nur der lokale PC-Service kommunizieren. (Nicht routbar! Nur im Netzsegment nutzbar!)

5. Was haben die beiden nachfolgenden IPv6-Adressen gemeinsam?

2001:0db8:f1f1:0100:0000:0000:feed:0001 und 2001:db8:f1f1:100::feed:1

Die beiden Adressen sind gleich, da die "stellenauffüllenden" Nullen nicht geschrieben werden. ":0000:" oder ":0000:0000:" hat die gleiche Bedeutung wie beispielsweise "::" werden. Die vier bzw. 8 Nullen sind zwar da, werden aber üblicherweise nicht geschrieben.

6. Was bedeutet die Anzeige der Adresse bei Windows: "2001:db8:feed:f101::feed:1/64%1"

Diese bezeichnet das Netzwerkinterface mit der Interface-ID: "%1" mit der Zugehörigkeit zum Netzwerk der Nummer: 2001:db8:feed:f101/64 und der Host-Nummer: :0000:0000:feed:0001.

7. Was hat die verbindungslokale Adresse: `fe80::213:e8ff:fe73:bccb/64`, die es bei jedem Netzwerkinterface immer gibt, mit der Mac-Hardware-Adresse: `00:13:e8:73:bc:cb` zu tun?

Die verbindungslokale Adresse beginnt immer mit "fe80". Dieses Präfix bezeichnet man als "link-local" und außerdem enthält sie die Mac-Adresse der Netzwerkkarte.

8. Was liefert der Linux-Befehl: `"ping6 -I eth4 fe80::213:e8ff:fe73:bccb"` und was der Windows-Befehl: `"ping -6 fe80::213:e8ff:fe73:bccb%4"`?

Diese beiden Befehle senden icmp-Testpakete der Version IPv6 an die jeweiligen Adressen, wobei sie die "link-local" Adressierung mit der integrierten MAC-Adresse und Interface-ID verwenden! Hier werden also nur die lokalen Interfaces angetestet. Will man eine LAN-Verbindung testen benötigt man nicht den locale-scope (=fe80), sondern den "global scope" mit der Netzadressierung.

9. Kann man bereits mit IPv6 ins Internet?

Man benötigt ein fest zugeordnetes Adress-Präfix, welches einem als Teilnehmer fest zugeordnet wird z.B. sixxs und Hurricane-Electric.

10. Gibt es NATTING und Private-IPv6-Adressbereiche und wie werden diese genutzt?

Es gibt kein Natting wie bei IPv4. Es gibt lediglich sogenannte Unique Local Addresses (ULA). Diese Adressbereiche nach RFC 4193 bieten ähnliche Funktionen wie die privaten IPv4-Adressbereiche. Die privaten Adressen (ULAs) sollen aber im IPv6 weltweit eindeutig zuordbar sein.

Es gibt selbsterzeugte (Adressen beginnen mit "fd") und global durch das RIPE zugewiesene Adressbereiche (diese beginnen mit den Ziffern "fc")

ULA-Adressen sind öffentlich zu registrieren. Alternativ kann man Side-Locale-Adressbereiche verwenden.

11. Wie werden ULAs für den privaten Gebrauch erzeugt?

Zunächst markiert das Präfix "fd" einen selbsterzeugten Adressbereich. Anschließend folgt eine zufällige 40 Bit-lange Site-Kennung inklusive einer willkürlich gewählte Subnetz-ID. Die durch die RIPE vergebenen allgemein öffentlich geltenden Adressen enthalten im Präfix ein „fc“. (Sehen Sie auch die Antworten zur Frage 13.)

12. Wozu dient das auf der nächsten Seite befindliche Shell-Skript?

Das Skript hilft bei der Berechnung geeigneter IPv6-Adressen für den eigenen nicht registrierten Gebrauch von IPv6-Adressen.

```

#!/bin/sh
#  @(#) generate-rfc4193-addr.sh (ULA) (c) Sep 2004 - Dec 2006  Holger Zuleger
#  do what the name suggest
#  firstpart = 64-Bit NTP time
#  secondpart = EUI-64 Identifier or 48 Bit MAC-Adress
#  sha1sum ($firstpart | $secondpart )
#  use least significant 40 Bits of sha1sum
#  build global prefix (locally assigned == FD00::/8)
#  (M1) 11. May 2006
#  - a check added to complain if firstpart or secondpart is empty
#  - firstpart calculation changed in such a way, that only one transmit
#    time is stored (ntpd since version 4.2.0 use a list of ntp servers)
#  (M2) 27. Aug 2006
#  Fixed bug in using reference time instead of transmit timestamp.
#  Thanks to Marc A. Donges for finding this out
#  (M3) 4. Sep 2006
#  Use ntpq instead of ntpdate because the latter is deprecated.
#  This requires a local running and synchronized ntpd, but
#  speeds up the execution time
#  (M4) 29. Dec 2006
#  set LC_ALL=C at the beginning of the script, to be sure the grep command
#  used to scan the output of the ifconfig command finds the expected string
#  Thanks to Ted Percival for finding this out
PATH=/usr/local/bin:/bin:/usr/bin:/usr/sbin:/sbin
debug=0
USE_NTPQ=1
NTPSERVER=pool.ntp.org
#(M4)
LC_ALL=C
export LC_ALL
#(M3)
if test $USE_NTPQ -eq 1
then
    if time=`ntpq -c rv | grep clock=`
    then
        test $debug -eq 1 && echo "$time"
        firstpart=`echo $time | sed -e "s/clock=//" -e "s/ .*//" -e "s/\./\\\\"`
    else
        echo "no local ntpd running" 1>&t2
        exit 1
    fi
else
    #(M1)
    #(M2)
    firstpart=`ntpddate -d -q $NTPSERVER 2>/dev/null | sed "/transmit timestamp/q" |
        sed -n "/transmit time/s/^transmit timestamp: *([^\ ]*) .*/\1/p" |
        tr -d "."`
    fi
    secondpart=`ifconfig eth0 |
        grep "inet6 addr: fe80" |
        sed -n "s|^:::([^\ ]*)/.*\|1|p" |
        tr -d ":"`
    #(M1)
    if test -z "$firstpart" -o -z "$secondpart"
    then
        echo "$0: installation error: check if ntpdate and ifconfig is in search path"
        exit 1
    fi
    test $debug -eq 1 && echo "Firstpart: $firstpart"
    test $debug -eq 1 && echo "Secondpart: $secondpart"
    test $debug -eq 1 && echo "123456789o123456789o123456789o123456789o123456789o123456789o"
    test $debug -eq 1 && echo "${firstpart}${secondpart} | sha1sum
    globalid=`echo ${firstpart}${secondpart} | sha1sum | cut -c31-40`
    test $debug -eq 1 && echo $globalid
    echo fd${globalid} | sed "s|\\(....)\\(....)\\(....)\\|1:2:3::/48|"

```

13. Welche IPv6-ULA-Präfixe eignen sich für eigene Tests im LAN und werden auf keinem Fall im Internet durch Router weitergeleitet?

Das IPv6-Präfix 2001:db8::/32 ist eigentlich für Dokumentationszwecke gedacht (RFC 3849). Da es laut dem "APNIC" niemals im Internet geroutet werden soll, eignet es sich für erste Gehversuche im eigenen LAN.

14. Welches Gerät garantiert die korrekte Funktion von "Stateless Auto-Configuration" ?

In einem Netzwerk ohne Router besteht das Problem, das der Host nur dann eine gültige IPv6-Adresse erzeugen kann, wenn dem Host ein Adress-Präfix-Vergabe mitgeteilt wird. Dies muss ein Router-Ersatzgerät liefern. Diese Art der Adressverteilung aufgrund einer Clientanfrage nennt sich: "Stateless Auto-Configuration". Hierbei wird mittels der verbindungslokalen Adressanfrage, indem der Host an die Multicast-Adresse ff02::2 eine "Solicitation Message" sendet, der Router aufgefordert dem Host ein gültiges IPv6-Präfix zuzuteilen.

15. Gibt es eine Möglichkeit IPv6 anonym zu surfen?

***Nein!** Gegenwärtig werden geeignete Proxies entwickelt. Man kann wohl den Hostanteil der IPv6-Adresse ändern, das Präfix bleibt jedoch erhalten. Abhilfe schafft hier nur ein TOR-Server. Linux kann seit 2013 Natting im Kernel aktivieren.*

<i>Einige wichtige IPv6-Befehle und der dynamische IPv6-Adresswechsel:</i>	
<u>Windows:</u>	<code>netsh interface ipv6 show privacy</code> <code>netsh interface ipv6 show address</code> <code>netsh interface ipv6 setglobal randomizedidentifier=enable</code>
<u>Linux:</u>	beachte die Datei im etc-Verzeichnis: <code>sysctl.conf</code> <code>net.ipv6.conf.<Interface>.use_tmpaddr=2</code> <code>net.ipv6.conf.<Interface>.tmp_preferred_lft=</code> <div style="text-align: right;"><code><Anzahl Sekunden></code></div> <code>net.ipv6.conf.<Interface>.tmp_valid_lft=<Anzahl Sekunden></code> <div style="text-align: right;">Beispiel: 3600 = 1 Stunde</div> <div style="text-align: right;">86400 = 24 Stunden</div>
Anzeige der IPv6-Adresse: <code>ip -6 addr show</code>	

Protokolle

Quelle: <http://www.heise.de/netze/artikel/IPv6-fuer-kleine-Netze-221783.html> von Reiko Kaps

- siehe: [Ellenlange Zahlen](#) [Kabel rein und los?](#) [Netzwerk-Diplomatie](#)
[Windows als Routenplaner](#) [LAN 6.0](#) [Fazit](#)

IPv6 für kleine Netze

Erfahrungen mit dem neuen Internet-Protokoll unter Linux, Mac OS X, Windows XP und Vista



Der Nachfolger des aktuellen Internetprotokolls sollte die globalen Netze vor der Verknappung nutzbarer Adressen bewahren. Die Entwickler haben allerdings noch weit mehr eingebaut, sodass IPv6-Netze besser und unkomplizierter zu verwalten sind. Nachdem mittlerweile alle nennenswerten Betriebssysteme IPv6 sprechen, wagen wir einen Versuch, das neue Protokoll in einem lokalen Netzwerk einzusetzen.

Das Internetprotokoll der nächsten Generation ([IPv6](#)) adressiert unvorstellbare Mengen an Computern und Netzwerkgeräten – eine ausführliche Beschreibung liefert der Beitrag [Das Mega-Netz](#) auf heise Netze. Außerdem verspricht ein erster Blick auf das Protokoll eine einfache und quasi automatische Einrichtung von Netzwerken – anders als bei [IPv4](#) kommt es ohne eine zentrale Vergabestelle für gültige Adressen in einem [LAN](#) aus.

Um Kollisionen von Adressen zu beheben, überprüfen IPv6-Rechner selbstständig, ob ihre Adresse in Netz bereits benutzt wird. Zudem entsorgt IPv6 die aus IPv4 bekannte Netzwerkmaske und Broadcast-Adressen, was die Zahl der Fehlerquellen bei der Netzwerk-Einrichtung nochmals senkt. Diese Gründe und die Tatsache, dass das Protokoll mit den aktuellen Betriebssystemen frei Haus geliefert wird, machen Lust auf erste Versuche mit IPv6 in einem kleinen Netz. Ein Test-LAN mit [Linux](#), Mac OS X, [Windows XP](#) und Vista soll zeigen, wie die Betriebssysteme mit dem Protokoll umgehen und was für Dienste sie darin anbieten und nutzen können.

Alle aktuellen Betriebssysteme bringen die Unterstützung für IPv6 mit. Da die Vorgängerversion IPv4 alles andere als überflüssig ist, besitzen Windows, Mac OS X und Linux einen sogenannten Dual-Stack für [TCP/IP](#), der beide Protokoll-Versionen parallel anbietet.

IPv6 eingebaut

```
rek@dhcp-157: ~ - Befehlsfenster - Konsole
Sitzung Bearbeiten Ansicht Lesezeichen Einstellungen Hilfe

eth0      Protokoll:Ethernet Hardware Adresse 00:80:AD:85:FD:CB
inet Adresse:192.168.1.1 Bcast:192.168.1.255 Maske:255.255.255.0
inet6 Adresse: fe80::280:adff:fe85:fdcb/64 Gültigkeitsbereich:Verbindung
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:0 errors:3845 dropped:0 overruns:0 frame:0
TX packets:0 errors:110 dropped:0 overruns:0 carrier:330
collisions:0 Sendewarteschlangenlänge:1000
RX bytes:0 (0.0 b) TX bytes:0 (0.0 b)
Interrupt:169 Basisadresse:0xa000

rek@dhcp-157:~$
```

Ein aktuelles Windows XP mit Service Pack 2 enthält das IPv6-Protokoll, man muss es aber von Hand aktivieren. In [Windows Vista](#) ist es ab Werk eingeschaltet. Gleiches gilt für Ubuntu, OpenSuse und Mac OS X, das sich ohne Zusatz-Software in einem IPv6-Netz zurechtfindet.

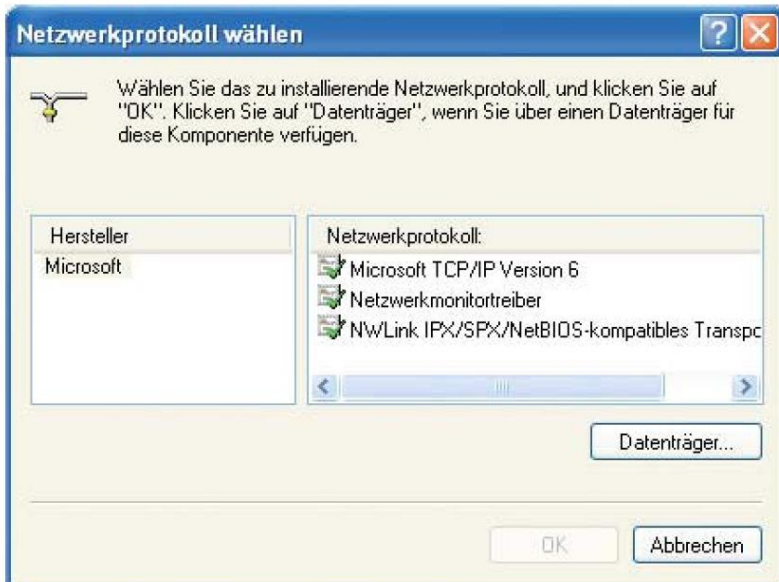
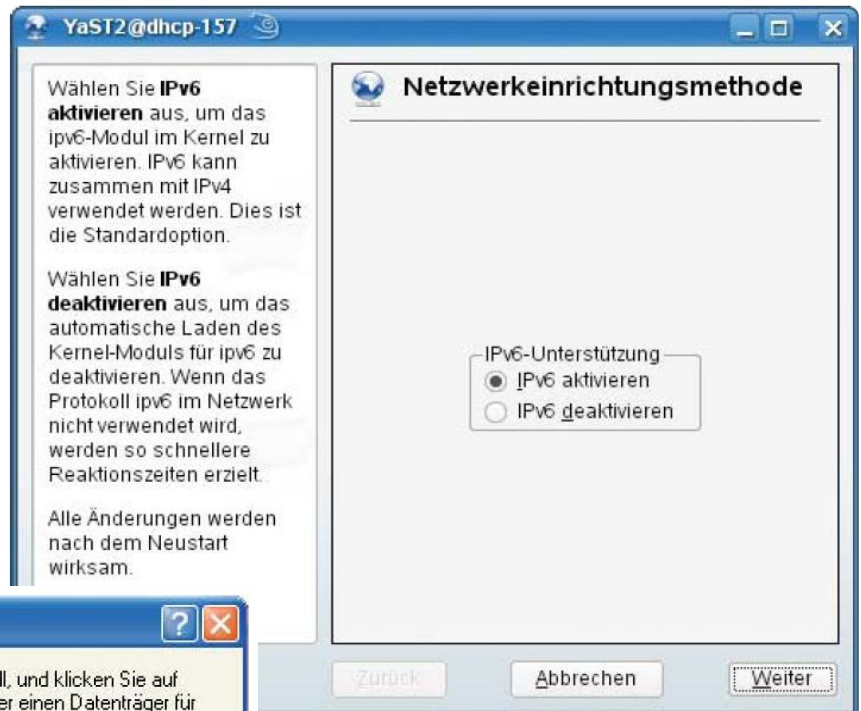
Das Linux-Kommando ifconfig zeigt, dass der Rechner das IPv6-Protokoll unterstützt. Die Netzwerkkarte (eth0) besitzt eine verbindungslokale IPv6-Adresse. Ob das Betriebssystem tatsächlich mit IPv6 umgehen kann,

lässt sich per Einrichtungsdialog oder Kommandozeile auf allen genannten Systemen schnell ermitteln: Unter Windows XP genügt ein Blick in die Eigenschaften der Netzwerkkarte. Taucht dort das Protokoll auf, ist es aktiv. Der Installieren-Knopf in diesem Dialog fügt es bei Bedarf hinzu. Wenn ein Aufruf von `ipconfig` auf der Windows-Eingabeaufforderung eine Zeile wie beispielsweise "Verbindungslokale IPv6-Adresse: fe80::9dfc:7fe:e8ef:4b3f%8" anzeigt, bringt Windows XP alle Voraussetzungen für IPv6 mit???

Das Einrichtungswerkzeug Yast unter OpenSuse kann auf den ersten Blick IPv6 nur ein- oder ausschalten. Für fest vergebene Adressen muss man das Feld für IPv4-Adressen nutzen.

Die Unterstützung für IPv6 lässt sich auf allen Linux-Fassungen schnell überprüfen, wenn man das Programm `/sbin/ifconfig` auf einer Unix-Shell aufruft. Alternativ zeigt das Kommando `/sbin/ip -6 addr show` ebenfalls die IPv6-Einstellungen an.

Unter einem aktuellen Windows XP muss IPv6 von Hand aktiviert und konfiguriert werden. Im Netzwerk- und Einrichtungs-



center von Windows Vista und im Bereich Netzwerke der Systemeinstellungen unter Mac OS X finden sich Einrichtungsdialoge für das Protokoll, die auf Windows XP und Ubuntu vollständig fehlen. Die Verwaltungszentrale Yast unter OpenSuse 10.2 kann das Protokoll unter dem Punkt Netzwerkgeräte pro Schnittstelle lediglich an- oder abschalten.

Ellenlange Zahlen

IPv4- und IPv6-Adressen unterscheiden sich in ihrer Länge und ihrer Notation. IPv4-Adressen sind 32 Bit lang, aufgeschrieben werden sie üblicherweise als Dezimalzahlen von 0 bis 255 (acht Bit) zwischen denen je ein Punkt steht. Beim Nachfolge-Protokoll vervierfacht sich die Adresslänge auf 128 Bit, was eine Dezimalschreibweise sehr unübersichtlich macht. Daher nutzt man Hexadezimalzahlen und unterteilt die Folge in Gruppen mit je 16 Bit, die durch Doppelpunkte getrennt werden.

Selbst diese Schreibweise produziert noch sehr lange Adressen, doch vereinfachen einige Regeln den Umgang: Führende Nullen in den Zahlenblöcken können wegfallen. Pro Adresse kann man eine zusammenhängende Folge aus Nullen ebenfalls streichen und durch zwei aufeinanderfolgende Doppelpunkte ersetzen. Durch die Kürzung wird die vollständige Adresse 2001:0db8:f1f1:0100:0000:0000:feed:0001 zu 2001:db8:f1f1:100::feed:1, die – nach etwas Gewöhnung – besser im Kopf bleibt.

Die variierenden Netzwerkmasken aus IPv4 gibt es im neuen Protokoll nicht mehr. Adressbereiche und Subnetze gibt man mit einem Anhängsel an, der durch einen Schrägstrich (/) vom Rest der Adresse getrennt wird. Üblicherweise adressieren in IPv6 die ersten 64 Bit das Netz, der Rest den [Host](#). Die IPv6-Adresse 2001:db8:feed:f101::feed:1 bezieht sich daher auf das Netz 2001:db8:feed:f101/64. In der Ausgabe des

Windows-Hilfsmittels ipconfig steht am Ende dieser Adressen außerdem ein Prozentzeichen und eine weitere Nummer, die die Schnittstelle kennzeichnet (Interface-ID).

Kabel rein und los?

Das kommende Internetprotokoll adressiert nicht mehr nur Geräte, sondern verspricht auch eine vereinfachte Einrichtung des Netzwerks. Es unterteilt die Knoten eines Netzes in zwei Gruppen: [Router](#), die fremde Pakete annehmen und weiterleiten, und Hosts, zu denen alle anderen Knoten gehören. Die Unterscheidung ist wichtig, denn während Router die Netzwerk-Präfixe verwalten und eine feste Adresse besitzen, wählen die Hosts die Kennung für ihre Netzwerkkarten selbst aus. IPv6-Adressen beziehen sich immer explizit auf einen Gültigkeitsbereich (Scope), wobei jede Karte mehrere Adressen besitzen kann. Auf Rechnern mit IPv6-Netzwerkunterstützung findet man nach dem Start mindestens eine Adresse, die sich auf den Bereich der Verbindung (link-local oder verbindungslokal) bezieht und die mit dem Präfix fe80::/64 beginnt. Diesem Adressvorspann folgen 64 Bit, die der Host aus der Hardware-Adresse der Netzwerkkarte ableitet.

```
C:\Dokumente und Einstellungen\cttest>ipconfig

Windows-IP-Konfiguration

Ethernetadapter LAN:

    Verbindungsspezifisches DNS-Suffix:
    IP-Adresse (Autokonfig.) . . . . . : 192.168.111.10
    Subnetzmaske . . . . . : 255.255.255.0
    IP-Adresse . . . . . : 2001:db8:100:f101:7d48:34c8:1c1f:450d
    IP-Adresse . . . . . : 2001:db8:100:f101:2c0:9fff:fe78:6172
    IP-Adresse . . . . . : fe80::2c0:9fff:fe78:6172%4
    Standardgateway . . . . . : 192.168.111.11
                                fe80::20f:eaff:fe28:d293%4

Ethernetadapter ...

C:\Users\ct>ipconfig

Windows-IP-Konfiguration

Drahtlos-LAN-Adapter Drahtlosnetzwerkverbindung:

    Medienstatus. . . . . : Medium getrennt
    Verbindungsspezifisches DNS-Suffix:

Ethernet-Adapter LAN-Verbindung:

    Verbindungsspezifisches DNS-Suffix:
    IPv6-Adresse . . . . . : 2001:db8::4532:9bc8:a886:33b9
    Temporäre IPv6-Adresse . . . . . : 2001:db8::841c:9b4a:f99a:60d2
    Verbindungslokale IPv6-Adresse . . : fe80::4532:9bc8:a886:33b9%8
    IPv4-Adresse (Auto. Konfiguration): 169.254.51.185
    Subnetzmaske . . . . . : 255.255.0.0
    Standardgateway . . . . . :

Tunneladapter LAN-Verbindung*:

    Verbindungsspezifisches DNS-Suffix:
    Verbindungslokale IPv6-Adresse . . : fe80::5efe:169.254.51.185%10
    Standardgateway . . . . . :

Tunneladapter LAN-Verbindung* 2:

    Verbindungsspezifisches DNS-Suffix:
    Standardgateway . . . . . :

Tunneladapter LAN-Verbindung* 6:

    Medienstatus. . . . . : Medium getrennt
    Verbindungsspezifisches DNS-Suffix:

C:\Users\ct>
```

Die Ausgabe von ipconfig unter Windows XP verschweigt den Gültigkeitsbereich der IPv6-Adressen. Mit dieser verbindungslokalen Adresse können IPv6-Rechner innerhalb eines LAN kommunizieren. So kann man

beispielsweise [ICMP](#)-Nachrichten mit dem Linux-Befehl `ping6 -I eth0 LINKLOCAL-ADRESSE` oder unter Windows mit `ping -6 LINKLOCAL-ADRESSE%INTERFACE-ID` senden und empfangen. Die Angabe der Netzwerkkarte per Parameter oder der Interface-ID ist notwendig, denn verbindungslokale Adressen besitzen auf allen Netzwerkkarten des Rechners das gleiche Präfix `fe80::/64`. Der Rechner weiß daher nicht, über welche Schnittstelle er die Pakete senden soll, denn das Präfix bezeichnet wie unter IPv4 ein Netzwerk. Dienste im Netzwerk funktionieren damit allein noch nicht. Über die verbindungslokale Adresse erhält der Host Informationen über die Anwesenheit von anderen IPv6-Hosts und -Routern. Der Rechner benötigt deshalb eine Adresse, die im LAN erreichbar ist und in der IPv6-Notation dem globalen Gültigkeitsbereich zugerechnet wird (Global Scope).

Nischen fürs LAN der nächsten Generation

Um dauerhaft nutzbare Adressen zu erhalten, gibt es drei Methoden: Man kann sich – zum Beispiel auf dieser [Liste echter IPv6-Anbieter](#) – einen [DSL](#)-Provider suchen, der Netzwerk-Präfixe für IPv6 bereitstellt. Gültige Netzwerk-Präfixe verteilen auch die Tunnel-Provider [sixxs](#) oder [go6](#), die IPv6-Daten durch IPv4-Verbindungen leiten. Will man nicht ins Internet, kann man fürs Erste einen reservierten Adress-Präfix auswählen. Bekommt man später ein im Internet gültiges Präfix zugeteilt, ist die Umstellung sehr einfach.

Die für abgegrenzte Bereiche reservierten sogenannten Unique Local Addresses (ULA), wie sie das [RFC 4193](#) beschreibt, bieten ähnliche Funktionen wie private IPv4-Adressen. Im Unterschied zu ihren IPv4-Verwandten sollen sie jedoch weltweit eindeutig sein – [Network Address Translation](#) entfällt bei IPv6 vollständig. Die Autoren des Standards unterschieden dabei selbsterzeugte und global durch das [RIPE](#) zugewiesene Adressbereiche – letztere beginnen mit den Ziffern `fc`. Das Präfix `fd` markiert selbsterzeugte Adressbereiche, die um eine zufällig erzeugte 40 Bit lange Site-Kennung und eine willkürlich gewählte Subnetz-ID erweitert wird. Bei der Berechnung dieser Zahlen hilft ein kleines [Shell-Skript](#), das beispielsweise das Präfix `"fdcd:1e7f:30ce::/48"` ausgibt. Aller Wahrscheinlichkeit nach sind diese Adressen weltweit eindeutig. Router sollen diese Adressen trotzdem nur innerhalb von abgegrenzten Standorten (Sites) wie beispielsweise Firmennetzen und zwischen ihnen weiterleiten, jedoch nicht im globalen Internet. Der Begriff Site ist allerdings nicht eindeutig definiert. *Nur wenn der lokale DNS-Server (AAAA-Rec.) und der Router IPv6 kann!!!*

Das IPv6-Präfix `2001:db8::/32` ist eigentlich für Dokumentationszwecke gedacht ([RFC 3849](#)). Da es laut dem Asia Pacific Network Information Center (APNIC) niemals im Internet geroutet werden soll, eignet es sich für erste Gehversuche in einem LAN, das nicht per IPv6 ans Internet angeschlossen ist. Wir nutzen deshalb im Artikel dieses Präfix, zumal es vergleichsweise kurz ist. Bei manueller Vergabe von IPv6-Adressen kann das ein Vorteil sein. Will man später auf ein anderes Präfix wechseln, sollte man allerdings die Möglichkeiten zur automatischen Konfiguration nutzen. Eine einzige Änderung auf dem Router schaltet dann das Netzwerk auf ein neues Netzwerk-Präfix um:

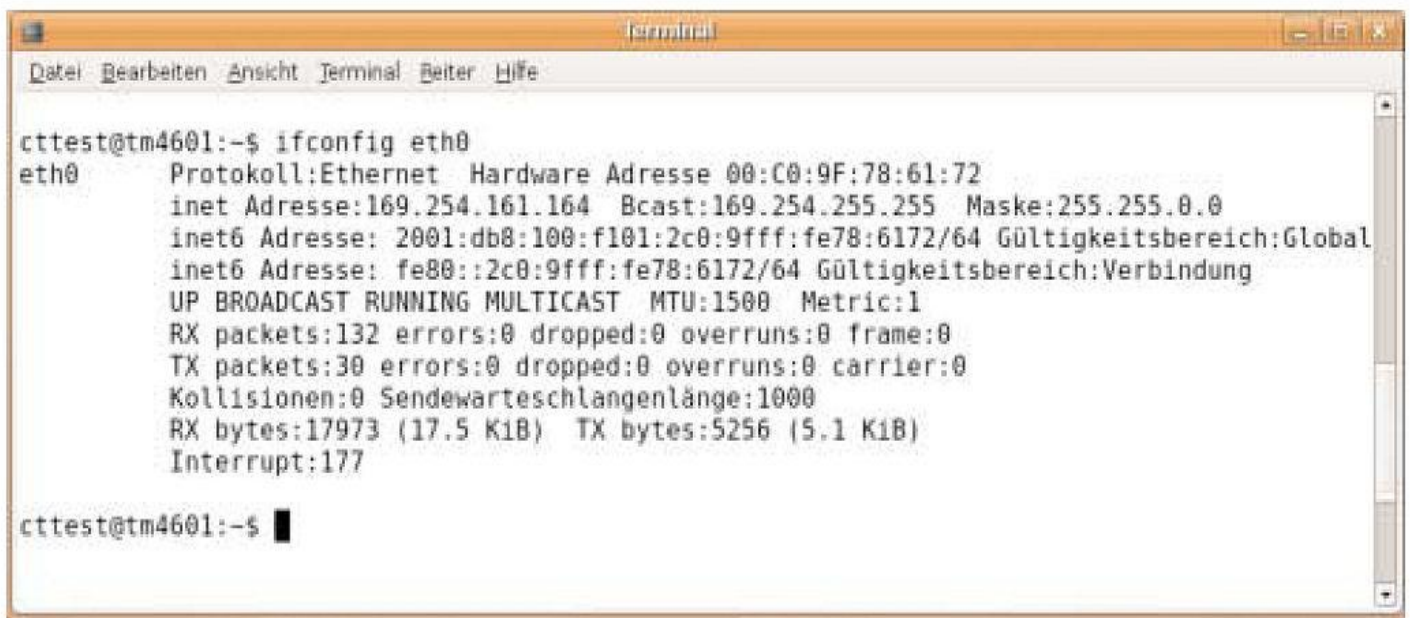
Damit ein Host eine gültige Adresse erzeugt, muss ein Router ihm mitteilen, welches Adress-Präfix er innerhalb des LAN nutzen soll. Diese Art der Adressverteilung nennt sich Stateless Auto-Configuration und unterscheidet sich von dem unter IPv4 bekannten [DHCP](#)-Protokoll: Mit Hilfe der verbindungslokalen Adresse sendet der Host an die [Multicast](#)-Adresse `ff02::2` eine Bitte (Solicitation Message), ihm einen IPv6-Präfix mitzuteilen.

Netzwerk-Diplomatie

Der LAN-Router antwortet auf die Solicitation Message mit einer Ankündigung (Advertisement-Message), die ein Adress-Präfix für dieses Netzwerk enthält. Daraus und aus der Hardware-Adresse seiner Schnittstelle erzeugt der Host seine IPv6-Adresse. Er prüft nun durch eine Anfrage im Netz, ob die Adresse bereits in LAN belegt ist (Duplicate Address Detection, [RFC 4862](#)). Ist die Adresse frei, weist der Host die Adresse der Schnittstelle zu und aktiviert sie.

Problem: Bei Hardwarewechsel (Netzwerkkarte) ändert sich die MAC- und damit die IPv6-Adresse!!!

Abhilfe bietet hier eine dynamische Vergabe der Mac-Adressen.



```
cttest@tm4601:~$ ifconfig eth0
eth0      Protokoll:Ethernet  Hardware Adresse 00:C0:9F:78:61:72
          inet Adresse:169.254.161.164  Bcast:169.254.255.255  Maske:255.255.0.0
          inet6 Adresse: 2001:db8:100:f101:2c0:9fff:fe78:6172/64 Gültigkeitsbereich:Global
          inet6 Adresse: fe80::2c0:9fff:fe78:6172/64 Gültigkeitsbereich:Verbindung
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:132 errors:0 dropped:0 overruns:0 frame:0
          TX packets:30 errors:0 dropped:0 overruns:0 carrier:0
          Kollisionen:0 Sendewarteschlangenlänge:1000
          RX bytes:17973 (17.5 KiB)  TX bytes:5256 (5.1 KiB)
          Interrupt:177

cttest@tm4601:~$
```

Ubuntu-Linux hat seine IPv6-Adresse automatisch von einem Router erhalten. Der globale Gültigkeitsbereich in der Ausgabe des Kommandos `ifconfig` zeigt das an. Ein Rechner, der als Router arbeiten soll, benötigt eine statische IPv6-Adresse. Auf einem Debian- oder Ubuntu-System mit einer Netzwerkkarte erweitert man dazu die Datei `/etc/networks/interfaces` um die Zeilen

```
iface eth0 inet6 static
    address 2001:0db8::1
    netmask 64
```

und startet das Netzwerk mit dem Befehl `/etc/init.d/networking restart` neu. Für die Verteilung des Präfix (Advertisement) ist unter Linux das Programm `radvd` zuständig, das mit `apt-get` installiert wird. Die Einrichtungsdatei `/etc/radvd.conf` muss vor dem ersten Start wenigstens folgende Einträge enthalten:

```
interface eth0 {
    AdvSendAdvert on;
    prefix 2001:db8::0/64
    {
    };
};
```

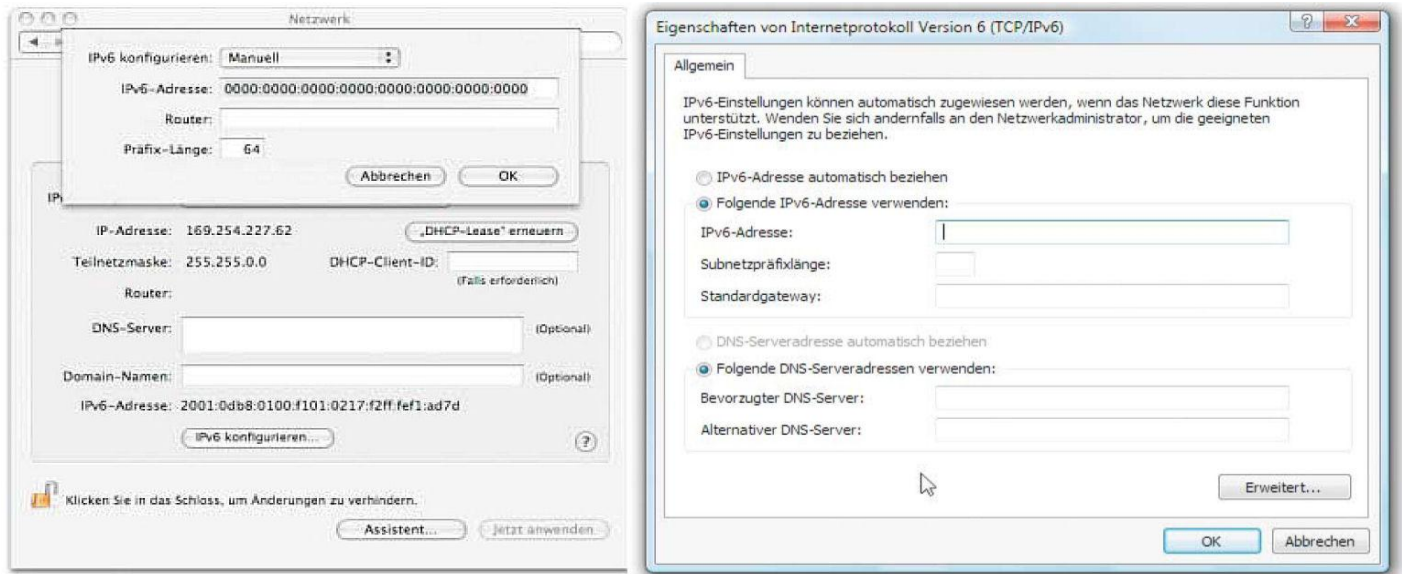
Der Eintrag `prefix` in der dritten Zeile enthält den verwendeten Netzwerkbereich, der bei Bedarf jederzeit änderbar ist. Auf dem Rechner muss außerdem das Forwarding für IPv6 aktiviert sein, denn sonst scheitert der Start von `radvd`. Der Wert `1` in der System-Variable `/proc/sys/net/ipv6/conf/all/forwarding` schaltet dieses Verhalten für alle Netzwerkkarten im System ein. Das Ubuntu-Startskript für `radvd` setzt die Einstellung selbstständig. Andere Linux-Ausgaben wie OpenSuse 10.2 steuern das Forwarding beispielsweise über Einträge in `/etc/sysconfig/`. Zur Not schaltet in einer Root-Shell das Kommando `echo "1" > /proc/sys/net/ipv6/conf/all/forwarding` die Weiterleitung bis zum nächsten Neustart ein.

Ist das Router-Programm per `/etc/init.d/radvd` start erfolgreich gestartet, verteilt `radvd` sofort das Adress-Präfix an angeschlossene IPv6-Rechner. Steckt man einen neuen Rechner ans LAN, kann es einige Sekunden bis Minuten dauern, bis er eine globale Adresse erzeugt hat.

Ist man im Besitz eines Heim-Routers, der mit Router-Linux-Distributionen wie [OpenWRT](#) kompatibel ist, kann man die schlanke Software dort arbeiten lassen. Das Programm `radvd` steht als Zusatz-Software zum [Download](#) bereit. Die Variante [DD-WRT](#) bringt es bereits mit – ein Beitrag im Wiki des Projekts [erklärt](#) die Einrichtung.

Windows als Routenplaner

Auch Windows XP ab Service-Pack 2 und Windows Vista können IPv6-Präfixe verteilen. Die Einrichtung erfolgt mit dem Programm netsh, dessen Kommandos standardmäßig persistent sind, also über einen Neustart des Rechners hinaus erhalten bleiben.



Die Dialoge von Mac OS X und Windows Vista erlauben die komfortable Eingabe von statischen Adressen. Bei anderen Betriebssystemen muss dafür meist die Kommandozeile herhalten. Eine statische IPv6-Adresse vergibt man unter Vista mit dem Einrichtung dialog, der sich hinter dem Punkt "Status anzeigen" der Netzwerkkarte im Netzwerk- und Freigabecenter versteckt. Alle weiteren Kommandos muss man auf einer Eingabeaufforderung eingeben, die mit Administratorrechten läuft. Unter XP setzt folgende Eingabe in die Kommandozeile eine IPv6-Adresse:

```
netsh interface ipv6 set address "LAN-Verbindung" 2001:db8::1
```

Der Wert "LAN-Verbindung" bezieht sich dabei auf die verwendete Netzwerkverbindung, deren Name von System zu System wechseln kann. Ein Aufruf des Kommandos ipconfig zeigt die Bezeichnungen der Schnittstellen an. Der nächste Befehl richtet eine Route für das Präfix auf das Interface ein:

```
netsh interface ipv6 add route 2001:db8::/64 "LAN-Verbindung" publish=yes
```

Ein weiterer Befehl aktiviert auf der Netzwerkschnittstelle das Advertisement, sodass der Windows-Rechner ab sofort den LAN-Präfix im Netzwerk verteilt:

```
netsh interface ipv6 set interface "LAN-Verbindung" advertise=enabled
```

Soll der Rechner tatsächlich IPv6-Pakete weiterleiten, muss dieser Befehl den zusätzlichen Parameter forwarding=enabled enthalten. Auf den angeschlossenen Rechnern sollte nun in der Ausgabe von ipconfig unter Windows respektive von ifconfig unter Linux eine Zeile erscheinen, die eine IPv6-Adresse mit dem Präfix 2001:db8 enthält. Mac OS X zeigt die Adresse auch in den Dialogen der Netzwerkeinstellungen an.

IPv6-Dienste

Arbeitet im Netz beispielsweise ein IPv6-tauglicher [SSH](#)-Server, kann man nun per PuTTY oder mit dem Kommando ssh auf ihn zugreifen. Ob der SSH-Server IPv6 spricht, verrät der lokal einzugebende Befehl netstat -A inet6 -a. Der Eintrag AddressFamily=inet6 in der Datei /etc/ssh/sshd_config zwingt SSH, ausschließlich über IPv6 Daten zu transportieren. Per Vorgabe nutzt der Server beide Protokollversionen.

Besteht das Netz aus mehreren Vista-Rechnern, sollte man im Idealfall in der Netzwerkumgebung die Freigaben der anderen Computer sehen. Der Dateiaustausch über das IPv6-Netzwerk unter Vista-Rechnern funktionierte in unserem Test reibungslos. Das in Debian und Ubuntu mitgelieferte Samba-Paket war per IPv6 nicht zu einer Zusammenarbeit mit Windows Vista zu überreden.

LAN 6.0

Will man zwischen Windows und Mac OS X Dateien tauschen, muss IPv4 im Netz aktiv bleiben. Ein Ausweg sind die verschiedenen FTP-Server. Mac OS X bietet Dateien per Apple Filing Protocol (AFP) auch über IPv6 an, was allerdings nur Mac-Clients verstehen. Einen Versuch sind möglicherweise Anleitungen wie die von David Holder – [Samba und Vista mit IPv6](#) – wert, die Samba nachträglich mittels xinetd IPv6-tauglich machen.

Beim Drucken über das Netzwerk sieht es dank Internet-Printing-Protokoll etwas besser aus: Der Druckserver CUPS steht für Unix und Mac OS X bereit, sodass IPv6-Clients Druckaufträge sicher abliefern können.

Sehr viele Linux- und Unix-Server, die in den aktuellen Distributionen enthalten sind, sprechen bereits das IPv6-Protokoll. Dazu zählen Webserver wie Apache in Version 2, der [DNS](#)-Server BIND, die Terminal- und Dateitransfer-Software SSH und zahlreiche andere Programme. Eine [Liste](#) der unterstützten Programme findet sich bei **deepspace6**. Allerdings entwickeln sich die dort erwähnten Programme ständig weiter, sodass die Seite nicht immer auf dem allerneuesten Stand ist.

Zentrale Adressvergabe

Der Router verteilt bei der zustandslosen Konfiguration (**Stateless Autoconfiguration**) nur das Netzwerk-Präfix und die Größe der Pakete ([MTU](#)) – Einstellungen wie die Netzwerk-Gateway und DNS müssen entweder per Hand nachgetragen oder über einen DHCPv6-Dienst erfragt werden.

Für ein LAN, das nicht per IPv6 ans Internet angeschlossen ist, reicht die zustandslose Konfiguration völlig aus. Auf DHCPv6 gehen wir daher hier nicht weiter ein – siehe dazu [RFC 4704](#).

Decknamen

Für ein Versuchsnetz lohnt sich die Einrichtung eines DNS (Domain Name System) eigentlich nicht. Wer es trotzdem versuchen will, muss gegenüber dem von IPv4 Gewohnten [wenig Neues lernen](#): Ein zusätzlicher Eintrag (**AAAA-Record**) nimmt die neuen Informationen auf – das wars auch schon. Daneben hat Microsoft für die Namensauflösung **PNRP (Peer Name Resolution Protocol)** vorgeschlagen, das das Problem ähnlich wie das [Bonjour-Protokoll](#) (siehe auch [für Entwickler](#)) dezentral lösen will.

Einfacher und schneller ist die Textdatei hosts auf den angeschlossenen Rechnern angepasst: Sie liegt unter Unix-ähnlichen Betriebssystemen im Verzeichnis **/etc**. Auf Windows XP und Vista findet man sie im Ordner **%WINDIR%\system32\drivers\etc**.

Pro Zeile nimmt die Datei jeweils eine IP-Adresse und einen symbolischen Namen für diese Adresse auf, etwa in der Form 2001:db8::1 ip6_router. Ruft man das Linux-Kommando **ping6** (unter Windows **ping -6**) nun mit dem gewählten Alias-Namen statt der IPv6-Adresse auf, antwortet der Rechner mit ICMP-Nachrichten, und auch andere Dienste wie beispielsweise Webserver sind mit diesem Namen ansprechbar.

Fazit *(am 12.11.2007 heute überholt!)*

Das kommende Internetprotokoll bringt bei der Einrichtung von Netzen interessante Funktionen mit. Ein tatsächliches Plug & Play auf dem Ethernet-Kabel bietet es allerdings nicht. Die zustandslose Konfiguration und die bereits nach dem Rechnerstart vorhandenen **verbindungslokalen Adressen** vereinfachen jedoch die Einrichtung eines lokalen Netzwerks deutlich. Nutzt man ausschließlich das aktuelle Windows-Betriebssystem oder Mac OS X in einem Netz, kann die Umstellung auf IPv6 sinnvoll sein. Der Dateiaustausch funktioniert dort reibungslos. Werkelt jedoch ein Zoo von Linux-Boxen, Mac-Rechnern und Windows-PCs im LAN, reicht das neue Protokoll für die Aufgaben eines lokalen Netzes noch nicht vollständig aus. Dazu fehlt einigen wichtigen Anwendungen wie beispielsweise dem Dateiserver Samba noch die vollständige Unterstützung für IPv6. Beim Internet-Zugang sieht es allerdings anders aus. Einige Anbieter stellen kostenlose **IPv6-Tunnel** bereit und vereinzelt bekommt man sogar echte DSL-Internet-Zugänge samt Adressbereich für das zukünftige Internetprotokoll.

Weitere Infos und Hinweise:

<http://www.sixxs.net/fag/connectivity/?faq=native>

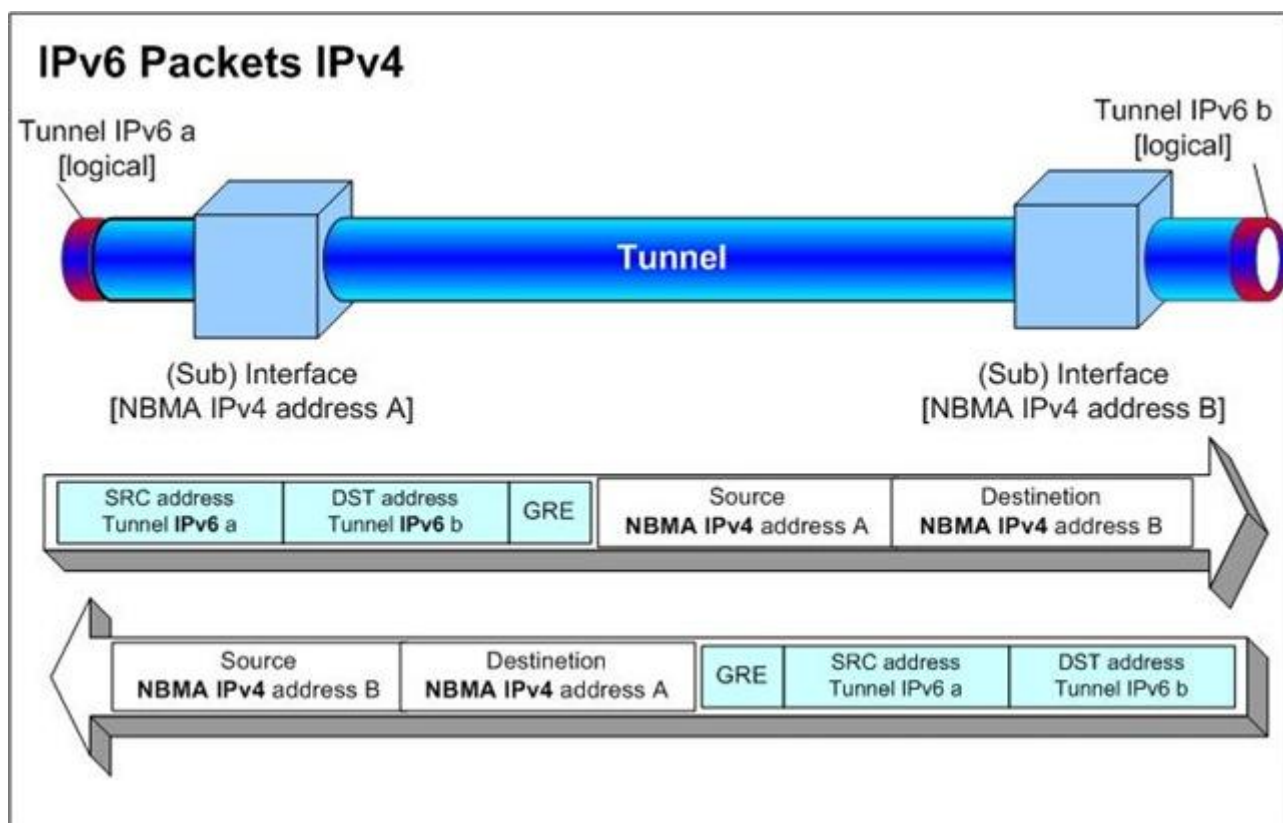
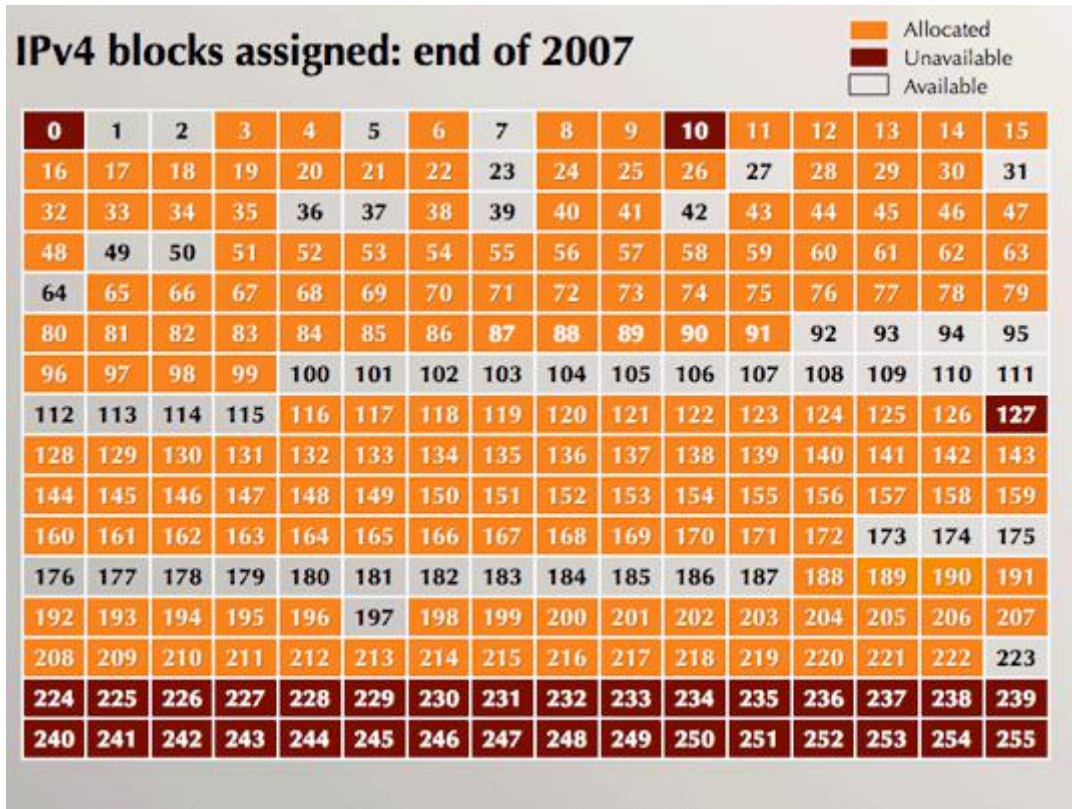
<http://www.heise.de/netze/artikel/IPv6-Das-Mega-Netz-221708.html>

<http://ipv6.net/>

<http://mirrors.bieringer.de/Linux+IPv6-HOWTO-de/>

<http://www.teltarif.de/internet/ipv6/>

IP-Adressvergabe zur Jahreswende 2007/2008:



The Cable Guy DNS-Verbesserungen in Windows Server 2008

Quelle: <http://technet.microsoft.com/de-de/magazine/2008.01.cableguy.aspx> Joseph Davies

Dieser Artikel basiert auf einer Vorabversion von Windows Server 2008. Die in diesem Artikel enthaltenen Informationen können jederzeit geändert werden.

Microsoft hat einen DNS-Serverdienst (Domain Name System) in die Versionen von Windows Server seit Windows NT 4.0 aufgenommen. DNS ist eine hierarchische, verteilte Datenbank, die Zuordnungen von DNS-Domännennamen zu verschiedenen Datentypen wie IP-Adressen enthält. In Windows Server 2008 umfasst der DNS-Serverdienst den neuen Zonenladevorgang im Hintergrund, Verbesserungen zur Unterstützung von IPv6, Unterstützung für schreibgeschützte Domänencontroller (Read-only Domain Controllers, RODCs) und die Möglichkeit, globale Namen mit einer einfachen Bezeichnung zu hosten.

Zonenladevorgang im Hintergrund

Der DNS-Serverdienst in Windows Server® 2008 beschleunigt den Datenabruf durch Implementieren des Zonenladevorgangs im Hintergrund. In der Vergangenheit kam es in Unternehmen mit Zonen, die eine große Anzahl von Datensätzen in Active Directory® enthielten, zu Verzögerungen von bis zu einer Stunde oder mehr, wenn der DNS-Serverdienst in Windows Server 2003 versuchte, beim Neustart die DNS-Daten aus Active Directory abzurufen. Während dieser Verzögerungen war der DNS-Server für DNS-Clientanforderungen für seine gehosteten Zonen nicht verfügbar.

Um dieses Problem zu lösen, ruft der DNS-Serverdienst in Windows Server 2008 nach dem Start Zonendaten aus Active Directory im Hintergrund ab, sodass er auf Anforderungen von Daten aus anderen Zonen antworten kann. Wenn der Dienst startet, erstellt er einen oder mehrere Ausführungsthreads, um die in Active Directory gespeicherten Zonen zu laden. Da separate Threads zum Laden der Active Directory-basierten Zonen vorhanden sind, kann der DNS-Serverdienst auf Abfragen antworten, während der Zonenladevorgang gerade ausgeführt wird. Wenn ein DNS-Client Daten in einer Zone anfordert, die bereits geladen wurde, antwortet der DNS-Server entsprechend. Wenn Daten in einer Zone angefordert werden, die noch nicht vollständig abgerufen wurde, ruft der DNS-Server stattdessen die spezifischen Daten aus Active Directory ab.

Diese Möglichkeit zum Abrufen bestimmter Daten aus Active Directory während des Zonenladevorgangs stellt einen zusätzlichen Vorteil gegenüber der Speicherung von Zoneninformationen in Dateien bereit, da der DNS-Serverdienst nun sofort auf Anforderungen antworten kann. Wenn die Zone in Dateien gespeichert ist, muss der Dienst die Datei sequenziell lesen, bis die Daten gefunden werden.

Verbesserte Unterstützung für IPv6

IPv6, das in früheren Ausgaben dieser Rubrik behandelt wurde, ist eine neue Sammlung von Standardinternetprotokollen. IPv6 wurde entwickelt, um viele Probleme der aktuellen Version (IPv4) wie Adressenknappheit, Sicherheit, automatische Konfiguration und den Bedarf an Erweiterbarkeit zu lösen.

Ein Unterschied in IPv6 besteht darin, dass die Adressen 128 Bit lang sind, während IPv4-Adressen nur 32 Bit lang sind. IPv6-Adressen werden in Hexadezimalnotation mit Doppelpunkt ausgedrückt. Jede Hexadezimalziffer entspricht 4 Bit der IPv6-Adresse. Eine vollständig dargestellte IPv6-Adresse besteht aus 32 Hexadezimalziffern in 8 Blöcken, die durch Doppelpunkte getrennt sind. Ein Beispiel für eine vollständig dargestellte IPv6-Adresse ist FD91:2ADD:715A:2111:DD48:AB34:D07C:3914.

Die Forwardnamensauflösung für IPv6-Adressen verwendet den IPv6-Host-DNS-Datensatz, der als AAAA-Datensatz (ausgesprochen „Quad-A“) bezeichnet wird. Für die Reversenamensauflösung verwendet IPv6 die IP6.ARPA-Domäne, und jede Hexadezimalziffer in der 32-stelligen IPv6-Adresse wird in umgekehrter Reihenfolge zu einer eigenständigen Ebene in der umgekehrten Domänenhierarchie. Der Reverse-Lookup-Domänenname für die Adresse FD91:2ADD:715A:2111:DD48:AB34:D07C:3914 ist beispielsweise 4.1.9.3.C.7.0.D.4.3.B.A.8.4.D.D.1.1.1.2.A.5.1.7.D.D.A.2.1.9.D.F.IP6.ARPA.

Der DNS-Serverdienst in Windows Server 2003 unterstützt die Forward- und Reversenamensauflösung für IPv6. Die Unterstützung ist jedoch nicht vollständig integriert. Um beispielsweise einen IPv6-Adressdatensatz (den gerade erörterten AAAA-Datensatz) im Windows Server 2003-DNS-Manager-Snap-In zu erstellen, müssen Sie mit der rechten Maustaste auf die Zone klicken, auf „Other New Records“ (Weitere neue Einträge) klicken und dann auf IPv6-Host (AAAA) als Ressourceneintragstyp doppelklicken. Um einen AAAA-Datensatz im DNS-Manager-Snap-In für Windows Server 2008 hinzuzufügen, klicken Sie mit der rechten Maustaste auf

den Zonennamen und klicken dann auf „Neuer Host“ (A oder AAAA).Im Dialogfeld „Neuer Host“ können Sie eine IPv4- oder eine IPv6-Adresse eingeben.**Abbildung 1** zeigt ein Beispiel.

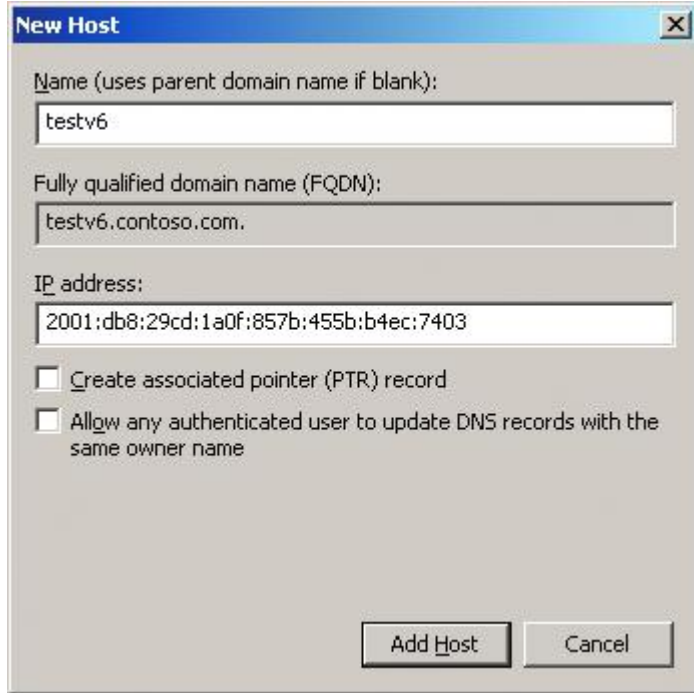


Abbildung 1 Dialogfeld „Neuer Host“

Ein weiteres Beispiel für bessere Unterstützung für IPv6 ist die Unterstützung von Reverse-IPv6-Zonen.Um eine Reverse-Lookupzone im DNS-Manager-Snap-In für Windows Server 2003 zu erstellen, müssen Sie den Reverse-Zonennamen manuell auf der Seite „Reverse Zone Lookup Name“ (Reverse-Lookupzonennamen) des Assistenten zum Erstellen neuer Zonen eingeben.Ein Beispiel für einen Reverse-Zonennamen in DNS ist 1.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa (für das IPv6-Subnetzpräfix 2001:db8:0:1::/64, vollständig dargestellt als 2001:0db8:0000:0001::/64).

IPv6-Reverse-Zonen im DNS-Manager-Snap-In für Windows Server 2008 sind jetzt in den Assistenten zum Erstellen neuer Zonen integriert.Der Assistent verfügt über eine neue Seite, auf der Sie zur Auswahl einer IPv4- oder IPv6-Reverse-Lookupzone aufgefordert werden.Für eine IPv6-Reverse-Lookupzone müssen Sie nur das IPv6-Subnetzpräfix eingeben, und der Assistent erstellt die Zone automatisch.**Abbildung 2** zeigt ein Beispiel.

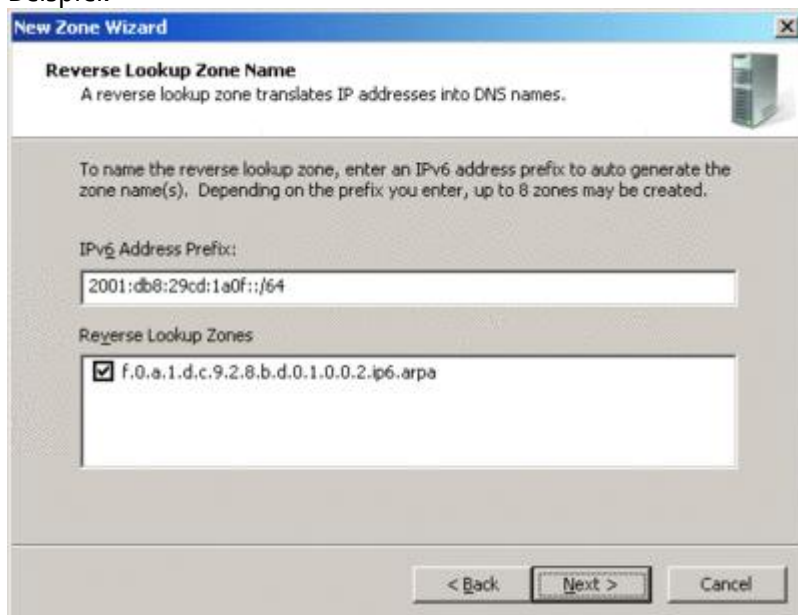


Abbildung 2 Benennen einer IPv6-Reverse-Lookupzone (Klicken Sie zum Vergrößern auf das Bild)

Eine weitere Verbesserung bei Reverse-Zonen ist die Art und Weise, wie das DNS-Manager-Snap-In IPv6-Zeigerdatensätze (Pointer, PTR) anzeigt.**Abbildung 3** veranschaulicht, wie das DNS-Manager-Snap-In für Windows Server 2003 einen PTR-Datensatz anzeigt.

Wenn der Name nicht gefunden wird, sendet der DNS-Client zusätzliche Namensabfragen für die Kombination aus der einfachen Bezeichnung und den Suffixen in seiner DNS-Suffixsuchliste (bei entsprechender Konfiguration). Wenn keiner dieser Namen aufgelöst wird, fordert der Client die Auflösung mithilfe der einfachen Bezeichnung an.

Der DNS-Server sucht die einfache Bezeichnung in der GlobalNames-Zone. Wenn sie dort angezeigt wird, sendet der DNS-Server die aufgelöste IPv4-Adresse oder den FQDN zurück an den DNS-Client. Andernfalls konvertiert der DNS-Clientcomputer den Namen in einen NetBIOS-Namen und verwendet NetBIOS-Verfahren einschließlich WINS zur Namensauflösung. Am DNS-Clientdienst müssen keine Änderungen vorgenommen werden, um die Auflösung der einfachen Bezeichnung in der GlobalNames-Zone zu aktivieren.

Windows Server 2008 und Windows Vista® unterstützen das NetBIOS über TCP/IP-Protokoll (NetBT). NetBT verwendet NetBIOS-Namen zum Identifizieren von Sitzungsschicht-NetBIOS-Anwendungen. Obwohl die NetBIOS-Namensauflösung mit WINS für aktuelle Versionen von Windows, die Windows Sockets-basierte Netzwerkanwendungen und DNS zur Namensauflösung benötigen, nicht erforderlich ist, stellen viele Microsoft-Kunden WINS in ihren Netzwerken bereit, um ältere NetBT-Anwendungen zu unterstützen und die Namensauflösung für einfache Bezeichnungen in ihren Organisationen bereitzustellen. Einfache Bezeichnungen beziehen sich in der Regel auf wichtige, bekannte und häufig verwendete Server in einer Organisation wie E-Mail Server, zentrale Webserver oder die Server für Branchenanwendungen.

Damit diese einfachen Bezeichnungen organisationsweit nur mithilfe von DNS aufgelöst werden können, könnte es erforderlich sein, den DNS-Domänen Ihrer Organisation A-Datensätze hinzuzufügen, sodass ein Windows-basierter DNS-Client den Namen unabhängig von dessen zugewiesenem DNS-Domänensuffix oder der Suffixsuchliste auflösen kann.

Angenommen die contoso.com-Organisation hat einen zentralen Webserver namens CWEB, der Mitglied der central.contoso.com-Domäne ist. Um eine einfache Bezeichnung für den Server CWEB zu implementieren, wenn DNS-Clients das DNS-Domänensuffix wcoast.contoso.com, central.contoso.com oder ecoast.contoso.com zugewiesen werden kann, muss der Netzwerkadministrator zwei zusätzliche A-Datensätze für cweb.wcoast.contoso.com und cweb.ecoast.contoso.com erstellen. Sie sollten jedoch beachten, dass manuell erstellte A-Datensätze für einfache Bezeichnungen bei Änderungen in der IPv4-Adresszuweisung oder bei neuen Namen verwaltet werden müssen.

Wenn contoso.com für ältere NetBT-Anwendungen WINS bereits verwendet, kann ein Netzwerkadministrator die Namensauflösung für die einfache Bezeichnung CWEB durch Hinzufügen eines einzelnen statischen WINS-Datensatzes zur WINS-Infrastruktur implementieren. Wenn sich die IPv4 Adresse ändert, muss nur der einzelne statische WINS-Datensatz geändert werden. Da einfache Bezeichnungen auf WINS leichter verwaltet werden können, verwenden viele Windows-basierte Netzwerke statische WINS-Datensätze für einfache Bezeichnungen.

Um eine Lösung für die Auflösung einfacher Bezeichnungen im DNS bereitzustellen, die sich ebenso leicht wie statische WINS-Datensätze verwalten lässt, unterstützt der DNS-Serverdienst in Windows Server 2008 eine neue Zone namens „GlobalNames“ zum Speichern von einfachen Bezeichnungen. Der Replikationsbereich dieser Zone ist in der Regel eine Gesamtstruktur, die die Auflösung für einfache Bezeichnungen über eine vollständige Active Directory-Gesamtstruktur bereitstellt. Zusätzlich kann die GlobalNames-Zone die Auflösung einfacher Bezeichnungen in einer Organisation unterstützen, die mehrere Gesamtstrukturen enthält, wenn Sie Ressourceneinträge zur Dienstidentifizierung (Service Location, SRV) zum Veröffentlichen des Speicherorts der GlobalNames-Zone verwenden.

Im Unterschied zu WINS soll die GlobalNames-Zone die Auflösung von einfachen Bezeichnungen für einen begrenzten Satz von Hostnamen bereitstellen, bei denen es sich in der Regel um die zentralen und wichtigen Server einer Organisation handelt, die von der IT-Abteilung verwaltet werden. Die GlobalNames-Zone sollte nicht zum Speichern der Namen von Desktopcomputern oder anderen Servern verwendet werden, deren IPv4-Adressen sich ändern können, und unter keinen Umständen werden von ihr dynamische DNS-Updates unterstützt. Sie wird am häufigsten für Aliasressourceneinträge (CNAME) zum Zuordnen einer einfachen Bezeichnung zu einem vollqualifizierten Domännennamen (Fully Qualified Domain Name, FQDN) verwendet. Für Netzwerke, die derzeit WINS verwenden, enthält die GlobalNames-Zone normalerweise Ressourceneinträge für von der IT-Abteilung verwaltete Namen, die bereits statisch in WINS konfiguriert sind.

Die GlobalNames-Zone stellt die Auflösung von einfachen Bezeichnungen nur dann bereit, wenn auf allen maßgeblichen DNS-Servern Windows Server 2008 ausgeführt wird. Doch auf anderen DNS-Servern, die für keine Zone maßgeblich sind, können ältere Versionen von Windows oder andere Betriebssysteme ausgeführt werden. Die GlobalNames-Zone muss in der Gesamtstruktur eindeutig sein.

Für maximale Leistung und Skalierbarkeit sollte die GlobalNames-Zone in Active Directory integriert werden, und Sie sollten jeden maßgeblichen DNS-Server mit einer lokalen Kopie dieser Zone konfigurieren. Dies ist erforderlich, damit die Bereitstellung der GlobalNames-Zone über mehrere Gesamtstrukturen hinweg unterstützt wird.

Weitere Informationen zur DNS-Unterstützung in Windows und zum Bereitstellen der GlobalNames-Zone finden Sie auf der Microsoft DNS-Webseite unter microsoft.com/dns.

Joseph Davies ist technischer Redakteur bei Microsoft und lehrt und schreibt seit 1992 über Themen im Bereich der Windows-Netzwerke. Er hat fünf Bücher für Microsoft Press verfasst und ist Autor des monatlich erscheinenden TechNet-Artikels „The Cable Guy“.

© 2008 Microsoft Corporation und CMP Media, LLC. Alle Rechte vorbehalten. Die nicht genehmigte teilweise oder vollständige Vervielfältigung ist nicht zulässig.

Protokolle Kapitel: [Artikelanfang](#) [Adressnotation](#) [Einfacher Header](#) [Adressumstellung](#)

von Dusan Zivadinovic, 05.04.07

IPv6: Das Mega-Netz - IPv6 wird Wirklichkeit

IPv6, die Neufassung des Internet-Protokolls, ist seit dem Erscheinen von Windows Vista auf den meisten populären PC-Betriebssystemen zusätzlich zu IPv4 aktiv. Damit dürfte es das langsame Ende der IPv4-Ära gerade noch rechtzeitig einläuten, dem es eine Reihe von Vorteilen voraus hat.

Die Entwicklung des Internet-Protokolls Version 6, [IPv6](#), begann bereits 1995, weil damals schon absehbar war, dass der durch den aktuellen Standard [IPv4](#) gebotene Adressraum knapp wird. Erste IPv6-Implementationen für PC-Betriebssysteme erschienen zur Jahrtausendwende (BSD), und in der Folge hielt IPv6 auch auf Mac OS X, [Linux](#) und schließlich Windows Einzug. Firmen wie Cisco, die Router für Internet-Backbones herstellen, haben es ebenfalls schon längst im [Programm](#). Es gibt auch den einen oder anderen Provider, der IPv6 anbietet; [Titan Networks](#), [rh-tec](#) oder auch [Space.Net](#) etwa. Auch gibt es frei erhältliche [PPP](#)-Treiber für die Provider-Anwahl.


- [IDC-Studie: Software-Lizenzen und Software-Product-Lifecycle-Management gemeinsam betrachten](#)
Dieses Whitepaper befasst sich mit Lizenz-Management anhand einer Studie von IDC und zeigt wie Hersteller und Kunden von SPLM-Technologien (Software Product Lifecycle Management) profitieren.
- [Information und Sicherheit: Wichtige Aspekte für Unternehmen in der heutigen Zeit](#)
Diese Marktstudie präsentiert interessante Zahlen und hält eine Reihe von Überraschungen parat.
- [Leitfaden zur Integration eines Softwarekopierschutzes in moderne Anwendungen](#)
In diesem Dokument gibt SafeNet dem Architekten einen Leitfaden an die Hand, der die wichtigsten Aspekte beinhaltet.

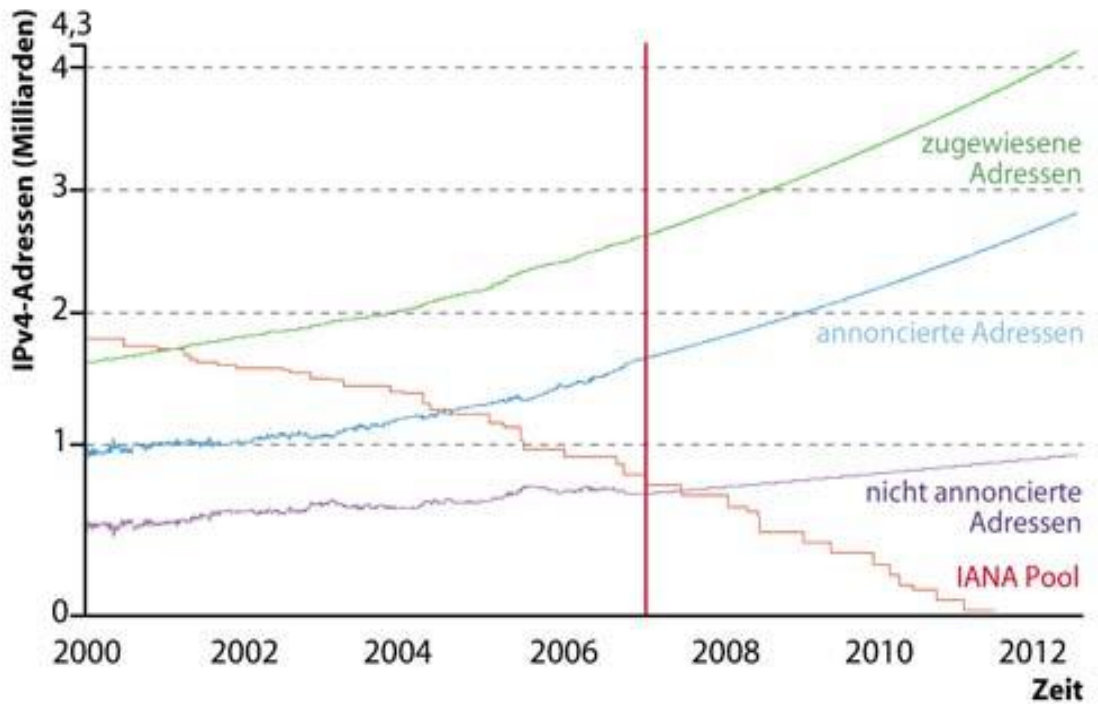
Aber auf breiter Front hat sich das Protokoll zur Enttäuschung seiner Verfechter bisher nicht durchgesetzt. Den Backbone-Betreibern und Providern in den USA oder auch Europa pressierte es bisher kaum – in diesen Ländern haben die meisten Provider deutlich mehr [IP](#)-Adressen als Kunden, sodass sich daraus teilweise das Phlegma erklärt, das bisher die Einführung verhinderte.

Nicht kleckern ...

Zu Beginn der Internet-Ära ging man recht sorglos mit dem Adressraum um; die IPv4-Spezifikation sieht 2^{32} Adressen vor, also rund 4,3 Milliarden und das schien damals mehr als genug. Fast 640 Millionen Adressen wurden für spezielle Zwecke [reserviert](#) und mit dem Rest ging man sehr großzügig um. So bekam etwa die University of California in Berkeley (UCB) rund 16,8 Millionen IP-Adressen, die sie kaum je ausschöpfen wird.

Große Teile dieser Adressräume liegen daher brach, aber eine Neuordnung wäre zu aufwendig. Zudem würde man so die Fahnenstange nur unwesentlich verlängern – der IPv4-Adressraum reicht ja nicht einmal aus, um jedem Menschen wenigstens eine Adresse zuzuteilen. Spürbar ist die Knappheit bereits in Südamerika oder Asien, wo die Nachfrage derzeit stark zunimmt, aber nurmehr deutlich kleinere Adressräume erhältlich sind. Man setzt daher diverse Techniken ein, um das Problem zu lindern, beispielsweise Classless Inter-Domain

Einer viel beachteten Prognose zufolge gehen der IANA, oberste Adressenvergabestelle des Internet, die letzten freien IP-Adressen unter Umständen schon im Jahr 2011 aus.  Aber das sind nur Tropfen auf den heißen Stein. Glaubt man den verschiedenen Prognosen, dann hat die [IANA](#), die oberste Adressenvergabestelle, die letzten freien IP-Adressen in wenigen

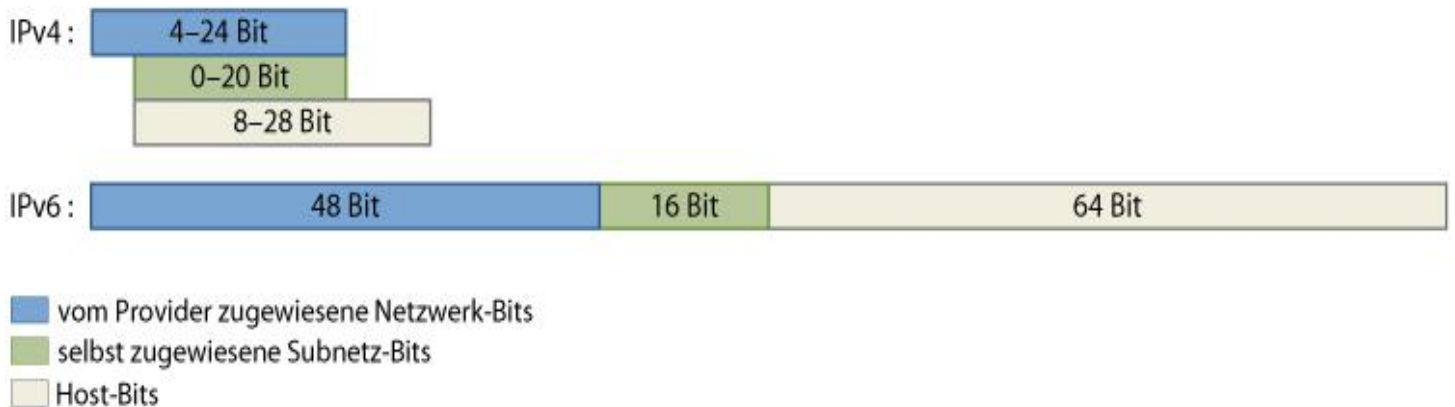



Jahren an ihre Unterorganisationen auf den verschiedenen Kontinenten zugeteilt; eine der [meistbeachteten Studien](#) geht von 2011 aus. Im Jahr 2012 haben demnach wiederum die IANA-Vertretungen ihre Adressräume komplett den Providern zugewiesen, und spätestens dann dürften die Folgen auch auf Surfer durchschlagen:

Wenn alle IPv4-Adressen gleichzeitig im Verkehr sind, müssen Nutzer, die einen Anschluss ohne feste IP-Adresse haben, darauf warten, dass eine IP-Adresse aus dem dynamisch zugewiesenen Bereich frei wird, bevor sie ins Internet können. Das dürfte die weitaus meisten DSL-Surfer betreffen, aber auch Nutzer mit [Modem](#)-, ISDN- oder Handy-Zugang.

... sondern klotzen

IPv6 definiert einen weit größeren Adressraum, nämlich 2^{128} oder 340 282 366 920 938 463 374 607 431 768 211 456 Adressen, also rund 340,28 Sextillionen. Das genügt, um jeden Quadratmillimeter der Erdoberfläche inklusive Ozeanen mit rund 600 Billionen Adressen zu bepflanzen. Weil man nicht knausern muss, ist es nicht nur möglich, Mobiltelefonen, Fahrzeugen oder beliebigen Haushaltsgeräten eigene IP-Adressen zuzuweisen, sondern man kann an der Verwaltung sparen und die Adressen ad hoc zuteilen.



Platz für jeden MP3-Player und jede Waschmaschine: Das IPv6-Netz schraubt den Adressraum auf über 340 Sextillionen IP-Adressen hoch.  Viele Provider haben sich im Stillen schon auf IPv6 vorbereitet – über hundert Anbieter in Deutschland haben bereits IPv6-Adressräume, nur sind sie noch nicht bei jedem im praktischen Einsatz oder nicht für Privatkunden erhältlich.

Zu den großen Vorteilen der IPv6-Spezifikation gehören auch die IP-Autokonfiguration anhand der [MAC-Adresse](#), Renumbering für den leichteren Wechsel ganzer Firmennetze zwischen Providern, Jumbogramme für Pakete bis zu 4 GByte Größe, schnelleres Routing, Punkt-zu-Punkt-Verschlüsselung gemäß [IPSec](#) sowie die Erreichbarkeit unter derselben Adresse in wechselnden Netzwerken (Mobile IPv6). Diese Techniken erläutern wir im Weiteren ausführlich.

Die IPv6-originiären Verfahren [Multicast](#) und [Quality of Service](#), die nachträglich auch bei IPv4 Einzug gehalten haben, behandeln wir in separaten Beiträgen. Multicast ersetzt [Broadcast](#) und verhilft zu einer effizienteren Bandbreitennutzung beim Video- und Audio-Streaming an mehr als einen Empfänger. Mit Quality of Service lassen sich Datenströme priorisieren, um zeitkritische Anwendungen vor Paketverlust zu bewahren. Das soll zum Beispiel bei der IP-Telefonie Verzögerungen oder Aussetzer verhindern.

Adressnotation

Um IPv6-Adressen kompakt darstellen zu können, greift man zur hexadezimalen Notation. Dabei sind die 128 Bit in acht Blöcke von je 16 Bit unterteilt; als Trennzeichen dienen jeweils Doppelpunkte. Die ersten vier Blöcke, also 64 Bit, werden für das Routing genutzt und bezeichnen das Netz-Präfix. Die darauf folgenden 64 Bit führen zum [Host](#).

Folgen von Nullen lassen sich abgekürzt darstellen, sodass manche Adressen noch etwas kompakter dargestellt werden können. Mit "::1" kann man die Host-Local-Adresse angeben, welche aus 15 Nullen und einer 1 besteht und dem Local Host 127.0.0.1 bei IPv4 entspricht. Diese und ähnliche Details sind bereits ausführlich erläutert, unter anderem in einem [c't-Beitrag](#).

In URLs kollidiert der Doppelpunkt mit der Portangabe, daher werden IPv6-Nummern in URLs wie in diesem Beispiel in eckige Klammern gesetzt:

```
http://[2001:0db8:85a3:08d3:1319:8a2e:0370:7344]:80/
```

Wie schon bei IPv4 kann jedes Netzwerk-Interface mehr als eine IP-Adresse haben – sie werden jedoch automatisch generiert. Startet man einen mit IPv6 bestückten PC, weist er sich selbst zunächst die Link-lokale Adresse für die Kommunikation im [LAN](#) zu. Die ersten 64 Bit haben immer den Präfix fe80 und die restlichen 48 sind Nullen:

```
fe80:0000:0000:0000
```

Für die zweiten 64 Bit wird die MAC-Adresse des Netzwerk-Interfaces in das Nummerierungssystem EUI-64 des [IEEE](#) umgewandelt (Extended Unique Identifier). Zusammen mit der ersten Hälfte sind das 128 Bit. Ein Beispiel sieht so aus:

```
fe80:0000:0000:0000:4231:65ff:fedc:1faa
```

oder

```
fe80::4231:65ff:fedc:1faa
```

Bevor ein Host eine solche Adresse nutzen kann, muss er per Neighbor Solicitation im LAN fragen, ob sie bereits belegt ist. Falls die fragliche Adresse ein anderer Host für sich annonciert (Neighbor Advertisement), kann der anfragende Host erst nach manueller IP-Adresseinstellung im LAN kommunizieren. Normalerweise sollte eine Kollision aber nicht vorkommen, denn schon MAC-Adressen sind für jedes Netzwerk-Interface weltweit individuell – aber sie lassen sich per Hand manipulieren. Wenn ein solcher Fall auftritt, ist es also ratsam, das LAN einer Prüfung zu unterziehen, weil möglicherweise Unbefugte eine gültige MAC-Adresse gekapert und per MAC-Spoofing ins LAN eingedrungen sind.

Mit der Link-lokalen Adresse kann ein Host nur im LAN kommunizieren; für öffentliche Verbindungen braucht er eine zusätzliche, die er anhand von Router-Antworten selbst generiert. Ein IPv6-Router sendet dafür auf Host-Anfragen das Präfix des öffentlichen Adressblocks, Lease Timeout, [MTU](#) und Hop Count (bei IPv4 [TTL](#) genannt). Ein Host kann nun prinzipiell eine öffentliche IP-Adresse aus Präfix und Suffix bilden; das Suffix ist seine EUI-64-Adresse. Praktisch an dem Verfahren ist, dass sich der Router nicht merken muss, welche IPs er bereits vergeben hat. Das Präfix kann kleiner als 64 Bit sein und wird dann mit Nullen auf 64 Bit aufgefüllt. Wechselt man den Provider, genügt es, dem Router das neue Präfix einzuimpfen, und die Hosts hüpfen

automatisch mit (Router Renumbering, [RFC 2894](#)), manuelle Eingriffe bleiben dem Administrator erspart.

Dieses Verfahren erleichtert es jedoch Dritten, Nutzerprofile aufzuzeichnen, denn die EUI-64 ist statisch. Deshalb hat man nachträglich ein Verfahren spezifiziert, das für die öffentliche IP-Adresse das Suffix ausgehend von der MAC-Adresse und einer pseudozufälligen Zahl erzeugt ([RFC 3041](#)). Dieser "zufällige" 64-Bit-Block ist nur temporär und kann zum Beispiel täglich gewechselt werden.


Freilich hat das Verfahren auch Schwächen und Nachteile. Sobald der Host einen [DNS](#)-Namen hat, den man per reverse DNS lookup auffinden kann, ist es untergraben, weil Lauscher den DNS-Namen wechselnden IP-Adressen zuordnen können. Netzwerkadministratoren dürfte es Analysen und Problemlösungen erschweren, weil es verschleiert, wie viele Hosts ein Netzwerkproblem verursachen.

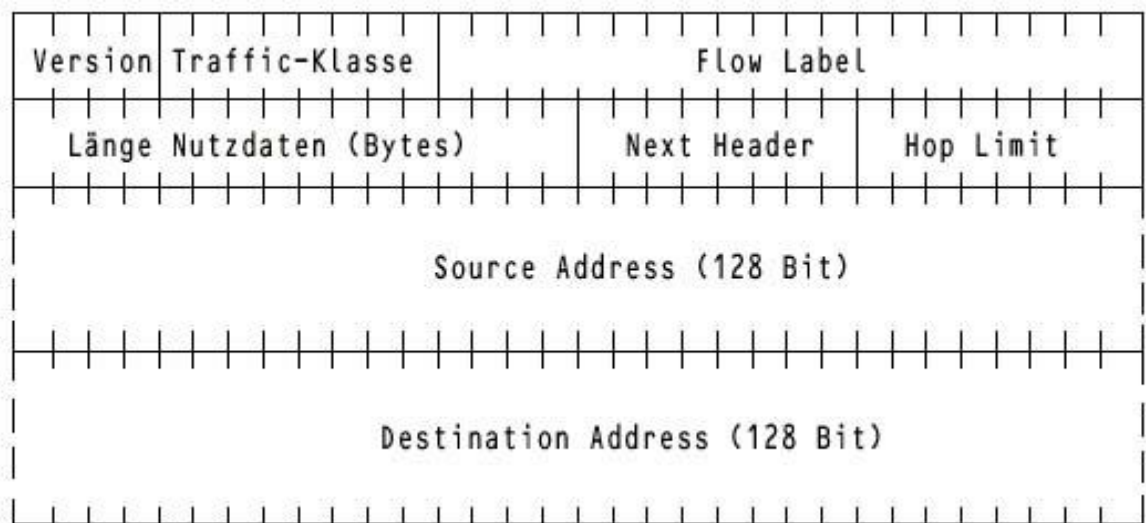
Über die automatische Adresszuteilung hinaus braucht es noch das automatische Finden des Nameservers. Da die Autokonfiguration keine Informationen über Host-, Domainnamen, DNS- oder auch [NTP](#)-Server erzeugt, muss man diese Informationen auf anderen Wegen beziehen. Für den öffentlichen DNS könnte man einen [DHCPv6](#)-Server verwenden. Dieser ist dadurch entlastet, dass er über die Adressvergabe im LAN nicht Buch führen muss (stateless DHCPv6, [RFC 3736](#)).

Microsoft geht einen radikaleren Weg und schlägt ein serverloses Protokoll vor, das [Peer Name Resolution Protocol](#), das Windows XP und [Vista](#) schon mitbringen.

Einfacher Header

Router müssen beim IPv4-Protokoll Checksummen prüfen und Pakete fragmentieren. Prinzipiell ist das nicht aufwendig, aber beim enormen Durchsatz aktueller Leitungen erfordert auch das reichlich Rechenleistung. Beides, die Fragmentierung und die Checksummen der IP-[Header](#), sind in IPv6 ersatzlos gestrichen. Eine Prüfsumme führt nun nur der TCP-Header. Fehlerhafte Pakete erkennt somit ausschließlich der Empfänger, und er fordert auch den Sender auf, betreffende Pakete neu zu schicken.

Bei IPv6-Headern sind die Adressfelder auf 64-Bit-Grenzen ausgerichtet. Das spart Rechenleistung in Routern und verspricht höheren Durchsatz.  Mangels Fragmentierung müssen Router



nun zu große Pakete grundsätzlich verwerfen und den Sender per [ICMP](#)-Nachricht über den Fehler informieren. Der Sender setzt dann die maximale Paketgröße für diese Route herab (MTU, Maximum Transmission Unit). Dieses Verfahren namens Path MTU Discovery ist bei IPv4 nur optional, und um es nutzen zu können, muss ein IPv4-Sender dafür das Don't-Fragment-Bit setzen.

Wenn die per ICMP-Block geschickte Fehlermeldung des Routers unterwegs verloren geht, beispielsweise wegen einer falsch konfigurierten [Firewall](#), schlägt die Path MTU Discovery fehl. In diesen seltenen Fällen muss der Sender die kleinstmögliche MTU verwenden. Diese beträgt bei IPv4 nur 68 Byte. Bei Routen, die prinzipiell höhere MTUs erlauben, wird so der Anteil der Verwaltungsinformationen unnütz erhöht und die Kapazität der Leitung nicht ausgeschöpft. Dieser Effekt wird bei IPv6 drastisch gemildert, denn alle IPv6-Geräte müssen mindestens 1280-Byte-Pakete befördern können. Natürlich dürfen sie wie IPv4 aber auch kleinere Pakete befördern.

Bei IPv6-Headern ist die Länge nicht mehr variabel und die Adressfelder sind auf 64-Bit-Grenzen ausgerichtet (64 Bit aligned). Das spart Rechenleistung in Routern und verspricht höheren Durchsatz. Flags wie das Don't-Fragment-Bit werden nicht mehr im Header übertragen, sondern als Teil von Optionen zwischen dem IP-Layer und [UDP/TCP](#).

Bewegliche Adressen

Mobile IPv6 erlaubt es, etwa mit einem Laptop an beliebigen Orten mit der heimischen IP-Adresse zu arbeiten, beispielsweise auf Konferenzen oder irgendwo in einem WLAN-[Hotspot](#). Bei IPv6 hat man für diesen Zweck eine ICMP-Umleitungsnachricht eingeführt, mit der der Laptop auf der Konferenz einem Agenten im heimischen Netz mitteilt, unter welchen IPs er gerade erreichbar ist. Der Agent stellt dann einkommende Verbindungen dorthin durch.

Sicherheitsexperten dürfte dieses Szenario alarmieren, denn man muss befürchten, dass [Cracker](#) die [banking.weltbank.de](#) für ihre Zwecke umleiten wollen. Daher darf der Agent nicht einfach auf Zuruf Verkehr umleiten, sondern der Administrator muss mit kryptographischen Methoden eine Authentifizierung sicherstellen. Eine adäquate Verschlüsselung bringt IPv6 in Gestalt von IPsec mit ([RFC 2411](#)). Das gibt es zwar auch schon bei IPv4, aber dort hauptsächlich im Tunnel-Modus für VPNs und nicht für Punkt-zu-Punkt-Verbindungen wie bei IPv6.

In der freien Software-Szene hat sich IPv6 inzwischen etabliert. Es gibt eine ganze Reihe an IPv6 unterstützenden Applikationen. Anfangs gab es nur Nameserver und diverse Tools wie [Ping](#) oder Traceroute, inzwischen aber auch Webserver, Browser oder auch Mailer, sodass man zumindest im LAN schon IPv6 üben konnte.

Im LAN zahlt sich IPv6 aus, weil es keinen Broadcast mehr gibt, der hohe Netzlast provozieren kann. Für alle unter IPv4 über Broadcast abgewickelten Übertragungen gibt es Multicast. Dadurch laufen in großen geschwittenen [Ethernets](#) die Neighbor-Discovery-Pakete praktisch nur auf den Strängen, auf denen der Host sich befindet.

Was ist eigentlich mit IPv5?

Es gab nie ein IPv5. Im IP-Header gibt es ein Versionsfeld, das bei IPv4 4 und bei IPv6 6 enthält. Es gab ein experimentelles Protokoll für Echtzeit-Ströme, für das an dieser Stelle eine 5 reserviert wurde. Dieses Protokoll hieß ST-2 und ist von [RSVP](#) ersetzt worden. ST-2 sollte Audio- und Videosignale per Multicast übertragen können und die Bandbreiten-Reservierungsvorteile von ATM in IP-Netze bringen. (*Felix von Leitner*)

Adressumstellung

Das in der Praxis größte Problem für den kompletten Umstieg, Adressumstellungen im laufenden Betrieb, schafft IPv6 dank Renumbering aus der Welt. Die Routing-Tabellen dürften dadurch deutlich verkleinert werden, denn bei den IPv4-Routing-Tabellen gibt es fragmentierte Adressbereiche: So kann der Gesamtbereich XY einem Provider gehören, aber der darin liegende Unterbereich Z einem ehemaligen Kunden, der zu einem anderen ISP umgezogen ist.

- [Disaster Recovery im Unternehmen – Bericht 2009](#)
Welche Rolle spielt eine effiziente und schnelle Disaster Recovery Konzeption für Unternehmen? Dieser Frage widmet sich diese Marktstudie von Symantec, in der Mitarbeiter aus 800 EMEA-Unternehmen befragt wurden.
- [Sicherheitsverstöße und deren Ursachen](#)
Datensicherheitsverstöße gefährden die Grundsatzsubstanz eines Unternehmens und es gilt diese mit allen Mitteln zu vermeiden. In diesem Dokument lesen Sie alles über die Thematik.
- [Aktueller Stand der Bedrohungen im Internet](#)
Wie ist es um die Sicherheit im Internet bestellt? Und vor allem die Sicherheit der Unternehmen, die den Zugriff auf das weltweite Datennetz gestatten? Dieser Bericht von Symantec zeigt ernüchternde Ergebnisse.

Teilte man solche Bereiche auf, wären sie nicht mehr als Aggregat durch eine Route zusammenfassbar und die weltweite Routing-Tabelle würde explodieren. Das vermeidet das Renumbering von IPv6, und Kunden, die mit ganzen Firmennetzen den Provider wechseln wollen, können den Schritt wagen, weil der Aufwand deutlich kleiner ist.

Wechseln die Kunden zugleich von IPv4 auf IPv6, gewährleisten verschiedene LAN- und Internet-Verfahren den schmerzfreien Übergang. Der Wechsel zu IPv6 beginnt im LAN. Dort gibt es grundsätzlich Geräte, die auf Layer 1 (etwa Hubs), auf Layer 2 ([Switches](#)) und auf Layer 3 (Router) aufsetzen. Layer-1-Geräte sind für IPv6 transparent, weshalb für diese keine Anpassung erforderlich ist. Layer-2-Devices wie Switches müssen Multicast beherrschen. Das ist bei allen modernen Switches der Fall.

Wenn man sich auf LANs beschränkt, dann markieren die Layer-3-Geräte (Router) im Heimbereich meist die Grenze des eigenen Netzes und sind daher für das Weiterleiten von IPv6-Verkehr zunächst nicht wichtig. Man kann also heute im LAN prima mit IPv6 arbeiten. Will man das LAN per IPv6 ans Internet anbinden, sieht es im Heimbereich routerseitig recht düster aus. Doch mit zwei LAN-Karten kann man PCs zu Routern aufrüsten: eine für die [PPPoE](#)-Verbindung zum IPv6-Provider und eine für die Anbindung des LAN. Linux-basierte Router-Distributionen können dann die IPv6-Fähigkeiten des Kernels zur Verfügung stellen.

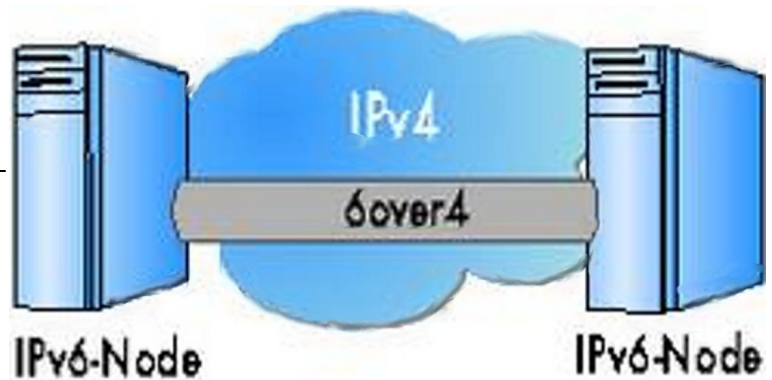
Zweisprachig

Die verbreiteten PC-Betriebssysteme sind für einen nahtlosen Umstieg auf IPv6 gerüstet. Dafür haben Linux, Mac OS X oder auch Windows Dual Stacks, mit denen sie IPv6 und IPv4 gleichzeitig beherrschen ([RFC 4213](#)). Bei Windows Vista ist IPv6 von Haus aus [aktiviert](#) und nicht wie beim Vorgänger XP lediglich mitgeliefert. Mit Vista ausgerüstete PCs greifen auf Netzwerkdienste wo immer möglich, also automatisch via IPv6, zu.

Um die zurzeit nur inselartige Ausbreitung von IPv6 zu stützen, also IPv6-Verkehr über die IPv4-Infrastruktur hinweg zu ermöglichen, gibt es mehrere Techniken. Man kann relativ leicht mit ein paar statischen Tunneln IPv6-Inseln über das Internet verbinden. Das hat zum Beispiel das 6BONE-Projekt demonstriert, das nach erfolgreicher Testphase Mitte 2006 abgeschaltet worden ist.

Inseln zu Kontinenten: Erste IPv6-Netze nutzen zum Beispiel Multicast-Tunnel, um IPv4-Infrastrukturen zu überbrücken, bis sie diese schließlich ablösen. Bei 6over4 tauschen zwei IPv6-Hosts Daten über einen per Multicast aufgebauten Tunnel durch das IPv4-Netz aus. Damit das auch mit dynamischen IPv4-Adressen funktioniert, hat das [Centro Studi e Laboratori Telecomunicazioni](#) (CSELT) Tunnel-Broker erfunden, bei denen man nach dem Einwählen automatisch seinen Tunnel aktivieren kann. Via 6to4 können zwei IPv6-Hosts

Daten in IPv4-Pakete enkapsulieren und über das IPv4-Netz austauschen. Teredo, eine Technik, die in Vista eingebaut ist, tunnelt IPv6-Verkehr via UDP durch NAT-Router. Weiterführende Informationen zu diesen und anderen Transitionsverfahren gibt es zum Beispiel beim IPv6-Dienst [Six Access](#).



Ausblick

Für die meisten Anwender wird IPv6 trotz der Implementation auf ihren PCs zunächst wohl nur im Verborgenen wirken. Mac OS X oder auch Vista bringen dafür gar keine zusätzlichen grafischen User-Interfaces mit, IPv6 werkelt dort von Haus aus im Autopilotmodus und ermöglicht zumindest ansatzweise Surfen ohne Benutzereingriffe.

Im Bunde mit den verschiedenen Transitionsmechanismen dürfte vor allem Vista erheblichen Auftrieb verschaffen. Einen ersten Schub gab es bereits, als IPv6 in Japan und Süd-Korea eingeführt wurde. Ab 2008 wird es auch für die Backbones in den USA Pflicht, was wohl den entscheidenden Dominostein zugunsten von IPv6 umstoßen dürfte. Erste IPv6-Router als fertige Boxen für den Privatanwender dürften dann nicht lange auf sich warten lassen. ([je](#))