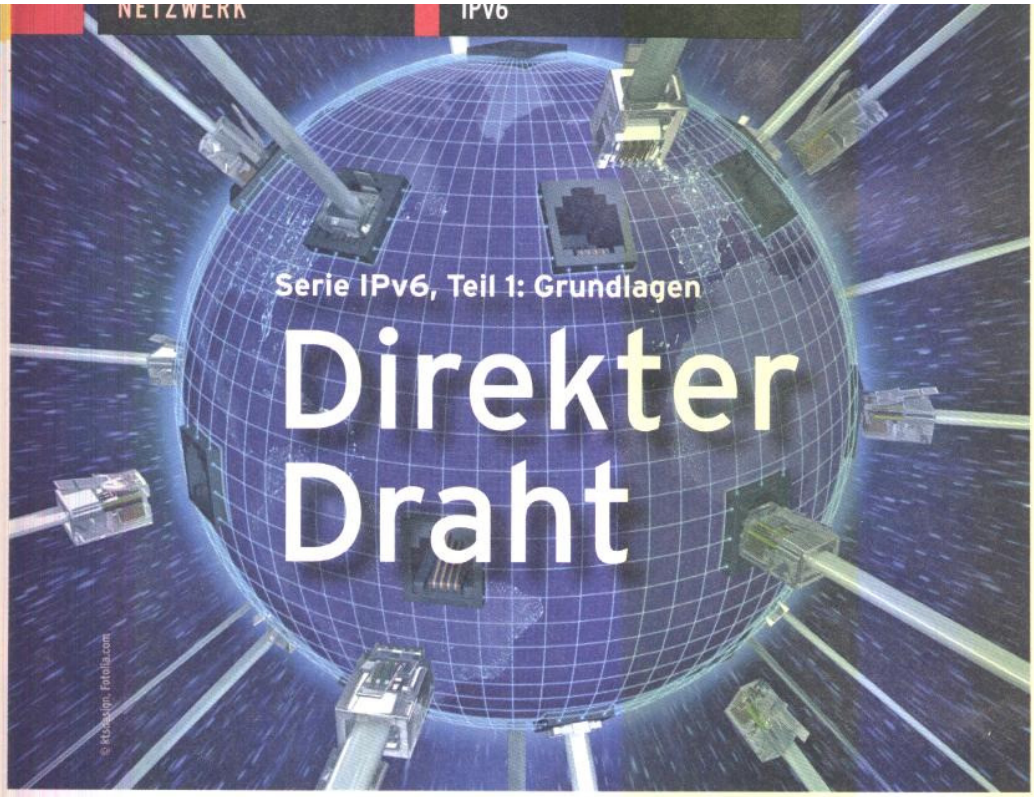

Serie IPv6, Teil 1: Grundlagen

Direkter Draht

Kaffeemaschine spricht mit Mobiltelefon – das geheimnisvolle IPv6 macht's möglich, denn es gibt eine unvorstellbar große Anzahl verfügbarer Adressen. Was steckt hinter der Technik der Zukunft? Thorsten Scherf

Jeder kennt es, jeder benutzt es – aber nur wenige kennen die genaue Geschichte und Funktionsweise: Die Rede ist vom Internetprotokoll [1]. Anfangs nur zum Verbinden einiger weniger Rechner gedacht, baut heute auf dieser Technik das gesamte Internet auf. Die aktuelle Version 4 des Protokolls hat jedoch ein Problem – es gehen so langsam die Adressen aus. Zeit also, einen Blick auf den Nachfolger IPv6 zu werfen.

Bei den heute als Internet bekannten Strukturen handelt es sich um eine Ansammlung verschiedener Protokolle. Um deren Aufgaben schematisch zu beschreiben, existiert das so genannte OSI-Schichtenmodell (Abbildung 1). Es ordnet den einzelnen Schichten die Protokolle des TCP/IP-Stack zu.

Auf den Layern 1 (Übertragen der Bits) und 2 (Sichern) stehen die Protokolle, mit denen Sie physikalische Verbindungen aufbauen – also beispielsweise Ethernet, aber auch Token Ring oder FDDI. Auf dem Layer 3 (Vermittlung) steht das Protokoll, das sich um das Verteilen der einzelnen Netzwerkpakete kümmert, das Internetprotokoll. Layer 4 (Transport) stellt den jeweiligen Anwendungen die Pakete bereit. Typischerweise kommen hier UDP und TCP zum Einsatz.

Auf den Layern 5 (Kommunikation), 6 (Darstellung) und 7 (Anwendung) finden sich die Anwendungsprotokolle, darunter beispielsweise HTTP, FTP und SMTP. Damit die einzelnen Daten, die ja in vielen einzelnen Paketen verpackt vorliegen, überhaupt den Weg zum Ziel finden, sind eine richtige Konfiguration und ein grundlegendes Verständnis des Internetprotokolls auf der Schicht 3 unerlässlich.

Die einzelnen, als IP-Pakete bezeichneten Datenpäckchen zeigen immer den gleichen Aufbau: Auf den so genannten Header mit den Informationen zum Zustellen des Pakets folgt ein Body mit den zu übertragenden Daten. Der Header umfasst in der Regel 20 Bytes, optionale Informationen erweitern ihn bis auf 60 Bytes. Der eigentliche Nutzdatenteil enthält je nach Netzwerk bis zu 65 535 Bytes. Im Ethernet ist eine Größe von 1500 Bytes üblich.

Zwei wichtige Felder im IP-Header enthalten die Empfänger- und die Absenderadresse. Diese beiden Felder sind in der aktuellen Version 4 des Internetprotokolls je genau 32 Bit groß, also 4 Bytes. Das ermöglicht es, zurzeit genau 4 294 967 296 (also 2^{32}) unterschiedliche Adressen anzusteuern.

In den Anfangstagen des Internets glaubten die Verantwortlichen, dass diese gut vier Milliarden Adressen für alle Zeiten reichen würden. Niemand kam auf den Gedanken, dass irgendwann nahezu jedes Telefon, jeder Fernseher und viele andere Geräte über eine eigene IP-Adresse verfügen würden, um ständig neue Daten aus dem Internet zu erhalten. Doch genau dieser Fall tritt ein – es gehen langsam die Adressen aus. Nach aktuellen Hochrechnungen der Internet Assigned Numbers Authority (IANA, [2]) sind spätestens Anfang 2013 die letzten freien IPv4-basierten Adressen vergeben.

Mit Umstrukturierungen des bestehenden Adressenraums in A-, B- und C-Netze mit fest definierten Subnetzmasken und später mit frei wählbaren Netzmasken und Adressenumsetzungen mit Hilfe von Network Address Translation (NAT) versuchten Techniker bereits früh, dieses Problem in den Griff zu bekommen. In mancherlei Hinsicht machte dies das Leben in der IP-Welt aber immer schwieriger und der Markt an IP-fähigen Endgeräten wächst nach wie vor.

Ausweg: IPv6

Aus diesem Grunde arbeiten bereits seit vielen Jahren Experten auf der ganzen Welt an einer neuen Version des Internetprotokolls. Sie trägt heute den Namen IPv6, früher hieß sie einmal IPng (IP next generation). Anders als bei der aktuellen Version 4 kommt hier ein jeweils 128 Bit – also 16 Bytes – großes Feld beim Adressieren zum Einsatz. Das erlaubt es, 2128 unterschiedliche Adressen anzusteuern, also ungefähr 340 Sextillionen.

Möchten Sie sich diese Zahl veranschaulichen, so stellen Sie sich vor, dass Sie bei diesem Adressraum jeden Quadratmillimeter der Erdoberfläche mit mehreren Hundert Milliarden IP-Adressen belegen könnten. Dabei ist die Anzahl der möglichen IP-Adressen nur ein Vorteil von vielen.

Die Verantwortlichen wollten die Fehler der ersten Protokoll-Implementierung nicht wiederholen, so floss eine Menge neuer Features und Fehlerkorrekturen mit in die neue Protokollversion ein. Die ersten Patches für den Linux-Kernel 2.1.8 stellte Pedro Roque bereit. Seitdem hat sich einiges getan.

Das Header-Format hat sich im Vergleich zur alten Version stark vereinfacht (Abbildung 2). Statt wie bislang zwischen 20 und 60 Bytes hat der Paketkopf nun eine feste Größe von 40 Bytes. Die verteilen sich auf einen Basis- und mehrere

OSI-Schichtenmodell	
Anwendung	HTTP, DNS, FTP, SMTP, ...
Darstellung	
Kommunikation	
Transport	TCP und UDP
Internet	IPv4, IPv6, ICMP, IGMP
Sicherung	Ethernet, FDDI, TokenRing
Bitübertragung	

Abbildung 1: Das OSI-Schichtenmodell bestimmt die Zusammenarbeit und die Aufgaben der einzelnen Protokolle.

Extension-Header, wobei letztere eher selten zum Einsatz kommen. Der kleinere IP-Header bietet den Vorteil, dass Router und andere Geräte zum Weiterleiten von IP-Paketen diese nun schneller und effizienter verarbeiten können.

Ein weiterer Vorteil von IPv6 liegt ganz klar im Bereich mobiler Endgeräte, also beispielsweise IP-fähigen Handys, PDAs oder auch Notebooks. Sie sind dann immer unter der gleichen Adresse zu erreichen – egal in welchem Netzwerk auf der Welt sie sich gerade befinden.

Dies ermöglicht ein so genannter Home Agent (HA). Er befindet sich im Heimatnetzwerk des eigentlichen Clients und empfängt die gerade aktuelle Adresse (Care of Address, CoA) des mobilen Endgerätes über so genannte Binding Updates des Geräts, sobald dieses im gerade aktuellen Netzwerk eine IP-Adresse bezogen hat. Spricht nun ein Gerät das Endgerät über die reguläre IP-Adresse des Heimatnetzwerks an, leitet der Home Agent die Anfrage einfach an die aktuelle IP des Geräts weiter. Dieses Feature heißt in der IPv6-Welt „Mobile-IP".

Auch im Bereich der Dienstequalität (Quality of Service, QoS) bietet das neue Protokoll einige Vorteile. So existieren im IP-Header zwei spezielle Felder, die die Dringlichkeit der Pakete definieren. Auf diese Weise legen Sie beispielsweise schon beim Aufbau der Verbindung bestimmte Anforderungen an diese fest. Dieser Funktion kommt gerade in solchen Umgebungen eine große Bedeutung zu, in denen die Zustelldauer der Pakete und die Bandbreite eine große Rolle spielen – also etwa bei Video- und Audio-Telefonie oder bei Onlinespielen.

Außerdem bringt beispielsweise die vorgeschriebene Integration von IPsec (IP-Security) auch zusätzliche kryptographische Mechanismen mit in das Protokoll. Dies soll das sichere Authentifizieren und Verschlüsseln der Datenpakete gewährleisten. Einige dieser Neuerungen existieren bereits als Aufsatz für die aktuelle IP-Version, so auch IPsec. Weitere Änderungen und Neuheiten, die IPv6 mit sich bringt, finden Sie auf der im Netz aufgeführten Liste [3].

Adressen-Schema

Angesichts der Größe des Adressraums eignet sich bei IPv6 die bei IPv4 verwendete dezimale Darstellung der Adressen nicht mehr. Schließlich müssten Sie hier nicht nur vier Oktetts mit jeweils 1 Byte darstellen, sondern acht Oktetts mit jeweils 2 Byte. Aus diesem Grund kommt bei IPv6 eine hexadezimale Schreibweise zum Einsatz. Ein Beispiel für eine solche Adresse wäre:

```
2001:0dc4:55b1:08d3:1216:8b2e:0170:3344
```

Die Adresse setzt sich dabei aus drei unterschiedlichen Segmenten zusammen. Das hintere (rechte) Segment nimmt die 64 Bit lange Interface-ID ein – im Beispiel »1216:8b2e:0170:3344« . Sie bestimmt, für welchen Rechner innerhalb eines Subnetzes ein IP-Paket bestimmt ist. Die ersten 64 Bit der Adresse (»2001:0dc4:55b1:08d3«) setzen sich aus Global Routing Prefix und Subnetz-ID zusammen.

In der Regel nimmt das Global Routing Prefix dabei die ersten 48 Bits (links) ein. Dies gibt der Internet Service Provider (ISP) an die eigenen Kunden weiter (»2001:0dc4:55b1«). Die sind dann in der Lage, mit den verbleibenden 16 Bits der Subnetz-ID (»08d3«) bis zu 65 535 eigene Teilnetze aufzubauen. Global Routing Prefix und Subnetz-ID zusammen bestimmen das so genannte Subnetz-Präfix. Benötigt ein Kunde mehr Teilnetze, so besteht die Möglichkeit, die Subnetz-ID entsprechend zu vergrößern.

Ein großer Vorteil von IPv6 liegt beispielsweise in der Möglichkeit, Netzwerkkarten automatisch zu konfigurieren. Diese Funktion heißt auch „Stateless Autoconfiguration". Möchte ein IPv6-Host seine Netzwerkkarte aktivieren, so generiert er zunächst die Interface-ID. Dies geschieht nach dem Zufallsprinzip oder auf Basis der MAC-Adresse. Natürlich besteht außerdem die Möglichkeit, die ID fix zu definieren.

Von allen verfügbaren Routern erhält der Client das passenden Subnetz-Präfix. Interface-ID und Subnetz-Präfix bestimmen dann die IPv6-Adresse des Clients. Von diesen darf er durchaus mehrere besitzen. Unter IPv6 ist es nicht üblich, eine einmal erhaltene IP-Adresse später zu löschen.

Die eigentliche Konfiguration findet komplett auf den Routern statt. Auf den Clients selbst brauchen Sie keine Angaben mehr zu machen. Benötigt ein Client neben der IP-Adresse selbst noch einen DNS- und Gateway-Server, lässt sich hierfür auf IPv6-fähige DHCP-Server zurückgreifen. Um die IP-Adresse kümmern die sich aber nicht mehr, das ist bei IPv6

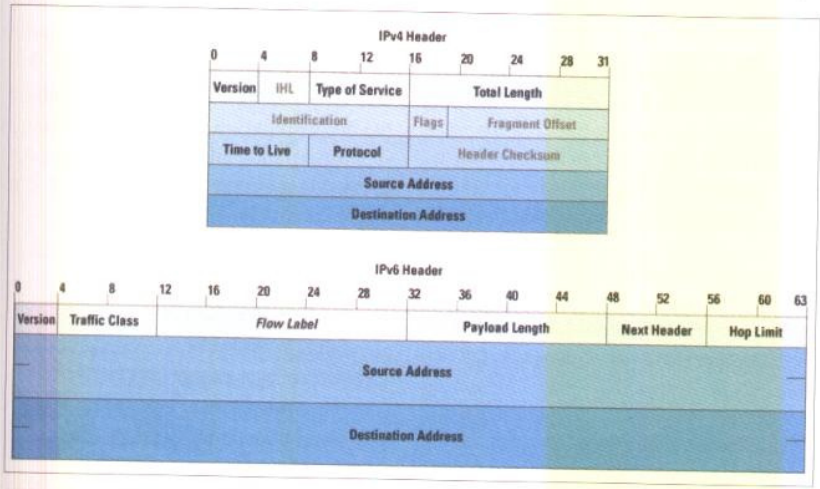

Abbildung 2: Der Header eines IPv6-Pakets enthält weniger Daten als der unter Version 4, ist also leichter von Programmen zu verarbeiten.

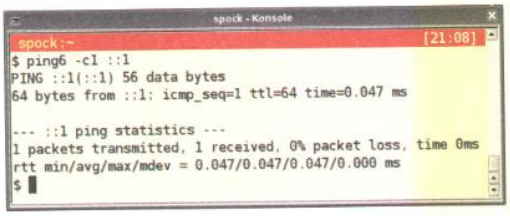

Abbildung 3: Der Befehl »ping6« setzt IPv6-Testpakete ab. Hier ist ein IPv6-Ping auf die Loopback-Adresse zu sehen.

Aufgabe der Router – die lästige Pflege von Lease-Datenbanken entfällt.

Wie bereits weiter oben angesprochen, bestehen IPv6-Adressen aus jeweils acht hexadezimalen 2-Byte-Blöcken. Die einzelnen Blöcke trennen Sie mittels Doppelpunkt voneinander. Beginnt ein Block mit einer oder mehreren Nullen, ist es erlaubt, diese einfach wegzulassen. Bestehen mehrere zusammenhängende Blöcke ausschließlich aus Nullen, dürfen Sie diese einmalig durch zwei aufeinander folgende Doppelpunkte ersetzen.

Vereinfacht

Die IPv6-Adresse »2001:0d81:0000:06e3:0000:5b1a:1010:1234« ließe sich nach der ersten Regel auch als »2001:d81:0:6e3:0:5b1a:1010:1234« schreiben. Die Adresse »2001:ab6:0:0:0:0:1234:35ce« wäre nach der zweiten Regel mit »2001:ab6::1234:35ce« gleichbedeutend. Aufeinanderfolgende Null-Blöcke dürfen Sie lediglich einmal ersetzen, da sonst keine Eindeutigkeit mehr gewährleistet wäre.

Um nun zwischen Interface-ID und Subnetz-Präfix zu unterscheiden, kommt die so genannte CIDR-Notation (Classless Inter Domain Routing) zu Einsatz. Die Anzahl der Bits, die das Subnetz-Präfix einnimmt, folgt als dezimale Zahl und mit einem Schrägstrich getrennt hinter der eigentlichen Adresse. Ein Beispiel: Die Adresse »2001:0ab6: 65b1:03a1: 2116:7a1e:1a50:4357/64« mit einem Präfix zu 64 Bit, bestimmt das Subnetz-Präfix mit »2001:0ab6: 65b1:03a1« und mit »2116:7a1e:1a50:4357« die Interface-ID.

Link-lokale Adressen

Unter IPv6 gehören bestimmte Adressen vordefinierten Bereichen an: Link-lokale Adressen (»fe80::/10«) kommen bei der initialen Kommunikation mit Routern und üblicherweise nur im lokalen Netzwerksegment zum Einsatz. Router leiten diese Adressen nicht weiter.

Besitzt ein Client im lokalen Netzwerk noch keine Adresse, konfiguriert er sich selbstständig mit Hilfe einer Link-lokalen Adresse. Mit Hilfe des Neighbor Discovery Protocol (NDP) sucht der Rechner dafür innerhalb seiner Nachbarschaft nach verfügbaren Routern, um von diesen gültige Präfixe für jene Netzwerksegmente zu erfragen, an denen er angeschlossen ist.

Befindet der Rechner sich in mehreren Netzwerksegmenten, erhält er mehrere Präfixe von unterschiedlichen Routern. Zusammen mit dem Interface-Identifier ergeben sich hieraus dann eine oder mehrere IP-Adressen für diesen Rechner.

Meist handelt es sich bei diesen um so genannte Unicast-Adressen die auch außerhalb des lokalen Netzwerksegments Gültigkeit besitzen.

Unicast-Adressen (»2000::/3«) dienen als globale, weltweit eindeutig IPs. Router leiten diese Adressen entsprechend ihren Tabellen an den jeweils nächsten Hop weiter. Diese Art der Adressen bezieht ein Client entweder von einem Router (Stateless Address Configuration) oder einem DHCP-Server (Stateful Address Configuration). In der IPv6-Welt spielt DHCP jedoch eine untergeordnete Rolle, sodass die Vergabe in den meisten Fällen durch einen Router stattfindet. In der Linux-Welt existiert für diese Aufgabe ein Tool mit dem Namen Radvd.

Bei Multicast-Adressen (»ff00::/8«) verbergen sich anders als bei Unicast mehrere Empfänger hinter einer Adresse. Somit lässt sich mit einem einzelnen Paket eine Vielzahl von Empfängern ansprechen. Beispielsweise sendet ein Client bei seiner initialen Autoconfiguration mittels NDP eine Anfrage an die Multicast-Adresse »ff02::2«. Alle Router des lokalen Netzwerksegments hören auf eine solche Adresse und antworten auf entsprechende Anfragen.

Ähnlich wie bei IPv4 existieren auch bei IPv6 einige spezielle Adressen wie beispielsweise »localhost« (»::1/128«, entspricht »127.0.0.1«) oder eine nicht definierte Adresse (»::/128«, entspricht »0.0.0.0«). Wollen Sie mit einem Ping das Loopback-Gerät testen, geben Sie hierfür das Kommando »ping6 -c1 ::1« ein. Die Ausgabe sollte ähnlich aussehen wie in Abbildung 3. Dies setzt eine funktionierende IPv6-Netzwerkkonfiguration voraus. Die meisten Distributionen aktivieren das Protokoll jedoch automatisch bei der Installation. Es ist also sehr wahrscheinlich, dass dieser kleine Test auch bei Ihrem Rechner klappt. *(agr/ofr)* ■

Kurze Geschichte des Internets

Ende der 60er Jahre hatte das US-Verteidigungsministerium (Department of Defense, DoD) die Idee, ein Computernetzwerk zu entwickeln, das weltumspannend arbeiten und maximale Ausfallsicherheit bieten sollte. Bei den Design-Ideen spielte der damalige Konflikt zwischen den Machtblöcken eine große Rolle. Der erste funktionstüchtige Entwurf des neuen Netzwerks trug den Namen Arpanet ([4], Advanced Research Projects Agency Network). Zum Vermitteln der Daten teilten die Forscher diese in kleine Happen auf und verpackten sie in mehrere Netzwerkpakete, um sie zwischen zwei Rechnern auszutauschen.

Das Netzwerk erfreute sich immer größerer Beliebtheit, immer mehr Universitäten und Forscher wollten Zugang. Die Folge: Innerhalb des Arpanet entstanden neue kleine Netz-Inseln, die das bestehende Netz immer weiter vergrößerten und unüberschaubarer machten. Aus Sicherheitsgründen trennte das Militär seinen Teil des Netzes Anfang der 80er Jahre ab, es war ab diesem Zeitpunkt nur noch über streng gesicherte Gateways zu erreichen.

Auf diese Weise entstand auf der einen Seite ein ausschließlich militärisch genutzter Teil, das so genannte Milnet, und auf der anderen Seite der Forschungsteil des alten Arpanet. Durch den Split des Netzwerks, die wachsende Beliebtheit der Kommunikation über vernetzte Rechner und mit dem Start der TCP/IP-Protokollfamilie [1] entstand aus dem Arpanet im Laufe der folgenden Jahre das Netzwerk, das heute den Namen Internet trägt.

Infos

[1] TCP/IP-Protokollfamilie: [http://de.wikipedia.org/wiki/TCPIP]
[2] Hochrechnungen zum Ende von IPv4: [http://www.potaroo.net/tools/ipv4/]
[3] Status IPv6: [http://ipv6.com/articles/general/timeline-of-ipv6.htm]
[4] Arpanet: [http://de.wikipedia.org/wiki/Arpanet]

IPv6 in der Praxis

Schleichweg

Mit einer Handvoll Änderungen in wenigen Konfigurationsdateien machen Sie Ihr Netz samt Router, Webserver und Clients fit für die schöne neue IPv6-Welt. Thorsten Scherf

Nachdem der erste Teil der Serie die Grundlagen zu IPv6 vorgestellt hat, geht es nun darum, Ressourcen über das neue Protokoll anzusprechen. Aktuelle Linux-Distributionen bringen IPv6-Support bereits von Haus aus mit. Das gilt sowohl für die Kernel- als auch für die Applikations-Seite. Benötigen Sie einen angepassten Kernel, müssen Sie bei der IPv6-Konfiguration darauf achten, alle notwendigen Module im Abschnitt »The IPv6 Protocol« auszuwählen.

Für die Applikationen ist es entscheidend, dass die Skripte zur Initialisierung des Netzwerks IPv6 unterstützen, was jedoch bei allen aktuellen Linux-Distributionen ohnehin funktioniert. Zusätzlich gibt es einige Werkzeuge, die bei der Administration eines IPv6-Systems helfen. Dazu zählt etwa »iptables-ipv6«, mit dessen Hilfe die IPv6-Erweiterung des Paketfilters gelingt. Ein Blick in die Paketverwaltung der verwendeten Distribution zeigt alle verfügbaren Tools.

Vorüberlegungen

Bevor Sie mit der Konfiguration der Netzwerkumgebung beginnen, gilt es aber, erst einmal einige Überlegungen anzustellen: Möchten Sie lediglich auf Ressourcen im eigenen LAN zugreifen oder wollen Sie die IPv6-Pakete auch in das globale Netz routen? Es gibt zwei gewichtige Gründe, diese Entscheidung bereits im Vorfeld zu treffen. Um die Pakete nur im eigenen Netzwerk zu transportieren, genügt eine eindeutige lokale Unicast-Adresse, eine so genannte Unique Local Address (ULA).

Solche Adressen entsprechen den Kreisen für private IPv4-Adressen, zum Beispiel »192.168.0.0/24«. Möchten Sie diese ULA selbstständig bestimmen, steht hierfür der IPv6-Adressraum »fd00::/7« zur Verfügung. Wollen Sie jedoch die IPv6-Pakete auch außerhalb Ihres eigenen Netzwerks transportieren, benötigen Sie zwingend eine eindeutige, globale Unicast-Adresse. Solche Adressen beginnen üblicherweise mit dem Präfix »2000::/3«.

Wo bekommen Sie eine solche globale IPv6-Adresse? Dafür gibt es mehrere mögliche Antworten: Zum einen bieten viele Internet Service Provider (ISP) bereits Support für IPv6 an, auf der anderen Seite existieren so genannte Tunnel-Broker. Diese tunneln die IPv6-Pakete über IPv4-Pakete und befördern sie auf diese Weise in die IPv6-Welt. Einer der bekannteren Anbieter dieser Art heißt Sixxs. Er bietet neben dem Bezug von einzelnen IPv6-Adressen auch die Weiterleitung ganzer IPv6-Subnetze an.

Somit lässt sich jeder Rechner im heimischen LAN mit einer öffentlich erreichbaren IPv6-Adresse versorgen. Das Besondere daran: Es funktioniert selbst dann, wenn die IPv4-Adresse des eigenen Tunnel-Endpunktes dynamisch zugewiesen ist und als NAT-Adresse für das lokale Netz dient.

IPv6-Adressen beziehen

Um an die begehrte IPv6-Adresse zu kommen, erstellen Sie als Erstes einen Account auf der Sixxs-Webseite [1]. Im Anschluss können Sie dann direkt einen Tunnel mit eigenem Subnetz beantragen. Nach erfolgreicher Anmeldung und der Annahme des Antrags erscheinen die relevanten Daten auf der Sixxs-Webseite (Abbildung 1).

Für den Aufbau des Tunnels benötigen Sie eine entsprechende Clientsoftware. Sixxs bietet dazu das Tool Aiccu (Automatic IPv6 Connectivity Client Utility) an. Sie laden es direkt von der Sixxs-Webseite herunter [2]. Fedora-Benutzer installieren die Anwendung einfach mit Hilfe des Befehls »yum install aiccu« aus dem Standard-Software-Repository heraus.

In der Konfigurationsdatei »/etc/aiccu.conf« tragen Sie den von Sixxs erhaltenen Benutzernamen nebst Passwort und die entsprechende Tunnel-ID ein (Listing 1). Durch Aufruf von »/etc/init.d/aiccu« starten Sie die Anwendung und bauen den Tunnel auf. Ein abschließendes Ping auf die IPv6-Webseite von Google bestätigt die korrekte Funktionsweise des Tunnels (Abbildung 2).

Rufen Sie im Browser nun auch einmal die URL [http://ipv6.whatismyv6.com] auf. Als Ergebnis bekommen Sie Ihre von Sixxs zugewiesene IPv6-Adresse angezeigt (Abbildung 3). Solange der Tunnel aktiv bleibt, sind Sie nun unter dieser Adresse aus dem IPv6-Netz zu erreichen.

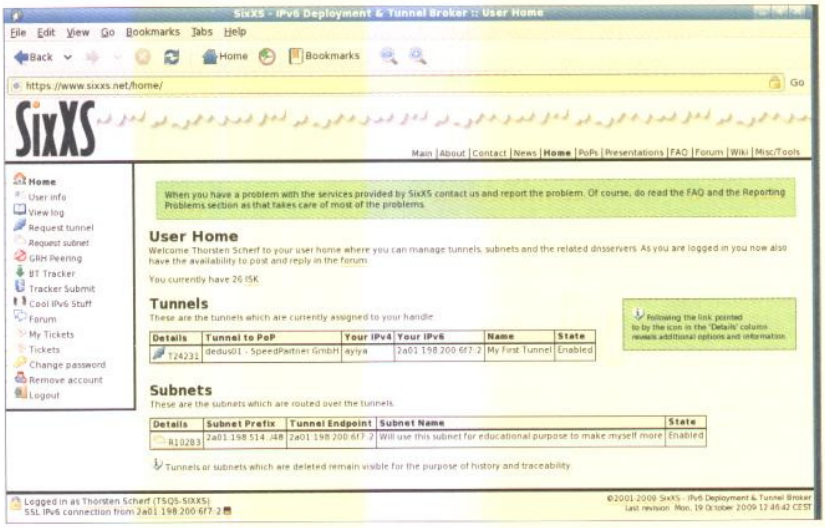

Abbildung 1: Über die Sixxs-Webseite können Sie ein ganzes IPv6-Subnetz beantragen.

IPv6-Adressen fürs LAN

Damit auch andere Rechner aus dem lokalen Netz via IPv6 auf das Internet zugreifen können, benötigen diese ebenfalls eine entsprechende Adresse. Für deren Zuweisung gibt es mehrere Lösungsansätze. Zum einen können Sie eine IPv6-Adresse aus dem zugewiesenen Subnetz manuell in die Netzwerk-Konfigurationsdatei eintragen. Im gezeigten Beispiel lautet das von Sixxs zugewiesene Subnetz-Präfix »2a01:198:514::/48«. Da es nur 48 Bits in Anspruch nimmt, verbleiben 16 Bits für die Definition eigener Netze – damit lassen sich bis zu 65 535 Subnetze aufbauen.

Hier genügt zunächst ein einzelnes Subnetz, sodass Sie den Rest der Adresse lediglich mit Nullen ausfüllen. Die Interface-ID besteht nur aus einer einzelnen Ziffer: 1. Die so erzeugte IP-Adresse tragen Sie auf einem Fedora-System dann zusätzlich zur bereits bestehenden IPv4-Konfiguration in die Datei »ifcfg-eth0« unter »/etc/sysconfig/network-scripts/« ein. Verwenden Sie ein anderes Interface, müssen Sie den Dateinamen entsprechend anpassen.

Die komplette Konfiguration könnte so aussehen, wie das Listing 2 zeigt. Nach einem Neustart des Netzwerks mittels »/etc/init.d/network restart« zeigt der Aufruf von »ip -6 a s eth0« und »ip -6 r« die konfigurierte IPv6-Adresse mit dem Default-Gateway an – jedenfalls dann, wenn alles geklappt hat.

Weitere Rechner lassen sich auf dieselbe Weise konfigurieren, einfacher geht es jedoch mit Hilfe der Stateless-Autokonfiguration. Hierzu setzen Sie die Anweisung »IPV6_AUTOCONF« aus der Konfigurationsdatei der Netzwerkkarte einfach auf den Wert »NO«. Daraufhin sendet der Client selbstständig so genannte Solicitation Messages an die Multicast-Adresse »ff02::2«. Als Absenderadresse kommt die verbindungslokale Adresse (»fe80::«) zum Einsatz, die das System bei aktivem IPv6 automatisch erzeugt.

Auf einem Fedora-System aktivieren Sie den IPv6-Support gegebenenfalls über die Anweisung »NETWORKING_IPV6 = yes« in der Datei »/etc/sysconfig/network«. Allerdings verwenden alle gängigen Linux-Distributionen diesen Wert in den aktuellen Releases als Vorgabe, sodass Sie hier meist nichts ändern müssen. Überprüfen Sie im Zweifelsfall einfach mit Hilfe von »ip«, ob der Rechner eine solche verbindungslokale Adresse besitzt (Listing 3, oben)

Existiert im LAN ein IPv6-fähiger Router, dann antwortet er beim Empfang der Solicitation Messages eines Clients mit einer so genannten Advertisement Message. Dieses Antwort-Paket enthält unter anderem das Subnetz-Präfix für Ihr Netzwerk. Aus der Hardware-Adresse der Netzwerkkarte generiert der Client dann die Interface-ID, die zusammen mit dem Subnetz-Präfix die IPv6-Adresse ergibt.

Klappt alles, dann besitzt der Rechner nun neben der verbindungslokalen Adresse auch eine globale Adresse aus dem oben genannten Subnetz (Listing 3, unten). Über diese Adresse kann der Client durch den IPv6-Tunnel mit anderen Rechnern im Internet kommunizieren.

Aber zuerst muss er überhaupt wissen, dass er die Pakete in den Tunnel senden soll. Diese Information bildet in Gestalt des angegebenen Default-Gateway jedoch bereits einen Teil der Advertisement Message des Routers.

Routing von IPv6

Bleibt die spannende Frage: Wie lässt sich solch ein IPv6-fähiger Router konfigurieren? Zum einen gibt es Hardware wie beispielsweise die Fritzbox, die den notwendigen IPv6-Support bereits mitbringt. Solche Geräte können selbstständig einen IPv6-Tunnel zu einem Tunnel-Broker aufzubauen. Zum anderen gibt es mit »radvd« auch eine passende Software

Listing 1: »/etc/aiccu.conf«

```
username TSQ5-SIXXS
password password
tunnel_id T24231
server tic.sixxs.net
protocol tic
ipv6_interface sixxs
verbose true
daemonize true
automatic true
requiretls false
```

Listing 2: »ifconfig-eth0«

```
IPV6INIT=YES
IPV6_AUTOCONF=NO
IPV6ADDR=2a01:198:514::1/64
IPV6_DEFAULTGW=2a01:198:200:6f7::2
```

```
[root@tiffy ~]$ /etc/init.d/aiccu start
Starting AICCU (Automatic IPv6 Connectivity Configuration Utility) services: Tunnel Information for T24231:
POP Id       : dedus01
IPv6 Local   : 2a01:198:200:6f7::2/64
IPv6 Remote  : 2a01:198:200:6f7::1/64
Tunnel Type  : ayiya
Adminstate   : enabled
Userstate    : enabled
                                                           [  OK  ]
[root@tiffy ~]$ ifconfig sixxs
sixxs     Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet6 addr: 2a01:198:200:6f7::2/64 Scope:Global
          inet6 addr: fe80::198:200:6f7:2/64 Scope:Link
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1280  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:500
          RX bytes:0 (0.0 b)  TX bytes:48 (48.0 b)

[root@tiffy ~]$ ping6 -c3 ipv6.google.com
PING ipv6.google.com(ipv6.google.com) 56 data bytes
64 bytes from ipv6.google.com: icmp_seq=1 ttl=56 time=71.6 ms
64 bytes from ipv6.google.com: icmp_seq=2 ttl=56 time=73.7 ms
64 bytes from ipv6.google.com: icmp_seq=3 ttl=56 time=72.2 ms

--- ipv6.google.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2075ms
rtt min/avg/max/mdev = 71.606/72.564/73.796/0.940 ms
[root@tiffy ~]$
```

Abbildung 2: Nach dem Start der Clientanwendung baut diese den IPv6-Tunnel auf und Sie erhalten eine globale IPv6-Adresse. Damit greifen Sie nun auf externe IPv6-Ressourcen zu.

für Linux. Die Anwendung findet sich in den Software-Repositories der meisten Linux-Distributionen, sodass die Installation mit dem jeweiligen Paketmanager leicht gelingt. Unter Fedora rufen Sie dazu »yum install radvd« auf.

In die Konfigurationsdatei »/etc/radvdv.conf« tragen Sie anschließend das von Sixxs zugewiesen Präfix ein (Listing 4). Nach einem Start des Dienstes mittels »/etc/init.d/radvd start« lauscht Radvd nun auf entsprechende Clientanfragen und sendet sogar selbstständig in regelmäßigen Zeitintervallen Informationen über das konfigurierte IPv6-Präfix in das Netzwerk.

Damit der Router nun auch alle Anfragen der Clients über den IPv6-Tunnel weiterleitet, müssen Sie noch das IP-Forwarding aktivieren. Das erledigen Sie im einfachsten Fall mit dem Befehl:

```
sysctl -w net.ipv6.conf.all.forwarding=1
```

Für eine dauerhafte Aktivierung setzen Sie diese Variable einfach statisch in die Datei »/etc/sysctl.conf«.

Listing 3: Adressen-Check

```
# ip -6 a s eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qlen
100
    inet6 fe80::208:74ff:fea8:dd17/64 scope link
       valid_lft forever preferred_lft forever

# ip -6 a s eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qlen
100
    inet6 2a01:198:514:0:21b:77ff:fe40:9e8e/64 scope
global dynamic
       valid_lft 2587384sec preferred_lft 600184sec
    inet6 fe80::208:74ff:fea8:dd17/64 scope link
       valid_lft forever preferred_lft forever
```

Listing 4: »/etc/radvdv.conf«

```
interface eth0
{
  AdvSendAdvert on;
  prefix 2a01:198:514::/64
  {
    AdvOnLink on;
    AdvAutonomous on;
    AdvRouterAddr on;
  };
};
```

Paketfilter

Danach sind Ihre Rechner ab sofort unter einer IPv6-Adresse direkt aus dem Internet zu erreichen, also nicht mehr hinter einer NAT-Adresse versteckt. Das bringt neben vielen Vorteilen freilich auch etliche Gefahren mit sich. Es führt kein Weg daran vorbei, auf dem zentralen IPv6-Router entsprechende Paketfilter-Regeln zu definieren, um den Zugriff von außen auf das Netzwerk einzuschränken.

Eine Beispiel-Konfiguration für einen solchen Paketfilter zeigt Listing 5. Speichern Sie das Skript als »/root/fw.sh« und aktivieren Sie dann die Regeln über den Befehl »sh /root/fw.sh«. Der Aufruf »/etc/init.d/iptables-save« sorgt dafür, dass die Regeln auch nach einem Neustart des Rechners aktiv bleiben.

Der Paketfilter erlaubt jeden Traffic aus dem eigenen Subnetz über den IPv6-Tunnel nach außen. Er lässt eingehende Pakete jedoch nur dann zu, wenn als Protokoll entweder SSH oder aber Bittorrent zum Einsatz kommen. Bei Bedarf erweitern Sie das Skript nach Belieben um zusätzliche Protokolle. Eine sehr gute Einführung in das Thema Paketfilter auf Basis von Netfilter finden Sie unter [3].

Eigene IPv6-Dienste

Falls Sie jetzt auf den Geschmack gekommen sind und eventuell eigene IPv6-Dienste unter einer statischen Adresse anbieten möchten, dann helfen eventuell die folgenden Abschnitte weiter.

Einer der am meisten verwendeten Internetdienste ist sicherlich das WWW. In der Unix-Welt kommt hierfür besonders oft Apache als Webserver zum Einsatz. Das Folgende geht davon aus, dass Sie in der Datei »/etc/httpd/conf/httpd.conf« nach dem Muster von Listing 6 bereits mehrere virtuelle Hosts für IPv4 konfiguriert haben.

Soll dieser virtuelle Host nun über eine IPv6-Adresse zu erreichen sein, müssen Sie die erste Zeile des im Listing gezeigten Abschnitts entsprechend abändern, sodass sie die IPv6-Adresse enthält:

```
<VirtualHost [2a01:198:514:0:21b:77ff:fe40:?
9e8e]:80>
```

Auch eine Kombination mit der alten IPv4-Adresse funktioniert. Wichtig: Die IPv6-Adresse muss in eckigen Klammern stehen, da sonst der Port-Trenner nicht eindeutig ist. Schließlich trennt man unter IPv6 die einzelnen Blöcke ebenfalls mit einem Doppelpunkt. Nach dem Neustart des Dienstes mit »/etc/init.d/httpd restart« lauscht dieser nun auch auf der angegebenen IPv6-Adresse auf eingehende Anfragen. Die Ausgabe von Netstat bestätigt das (Listing 7).

Generell lässt sich sagen, dass auf aktuellen Linux-Distributionen die meisten Serverdienste bereits IPv6-fähig sind. Einige, etwa der Apache-Webserver, benötigen zum korrekten Funktionieren noch einige Anpassungen, andere laufen bereits ohne manuelle Änderungen problemlos auch unter IPv6. Stellen Sie beispielsweise eine SSH-Verbindung zu einem IPv6-Host her, so sollte der Connect ohne Probleme funktionieren.

Die hierfür verantwortliche Einstellung in der Konfigurationsdatei »/etc/ssh/sshd_config« lautet »AddressFamily any« und ist in der Standardeinstellung bereits ak-

Abbildung 3: Über die Webseite [http://ipv6.whatismyv6.com] sehen Sie die IPv6-Adresse, mit der Sie im Internet unterwegs sind.

tiv. Möchten Sie den Connect ausschließlich für IPv6 erlauben, lautet die Anweisung »AddressFamily IPv6«.

Zum Schluss noch einige Hinweise zum Thema DNS-Namensauflösung: In der IPv4-Welt funktioniert die Auflösung eines Rechnernamens in die dazu passende IP-Adresse über einen so genannten A-Eintrag. Der findet sich in den Zonendateien eines DNS-Servers. Unter IPv6 erfolgt dieses Mapping jedoch über den AAAA-Eintrag. Listing 8 zeigt ein Beispiel für die Auflösung eines IPv4- und eines IPv6-Namens.

Möchten Sie Ihre eigenen Dienste im DNS nun auch unter Ihrer IPv6-Adresse registrieren, müssen Sie dazu entsprechende Änderungen an der Zonendatei des DNS-Servers vornehmen. Nähere Informationen zu diesem Thema bietet beispielsweise die ausführliche Sixxs-FAQ [4].

Listing 5: IPtables für IPv6

```
# Alle bestehenden Regeln löschen
ip6tables -F
ip6tables -X

# Zugriff über das Loopback-Device erlauben
ip6tables -A INPUT -i lo -j ACCEPT
ip6tables -A OUTPUT -o lo -j ACCEPT

# Uneingeschraenkter Zugriff auf den
IPv6-Tunnel vom Router aus
ip6tables -A OUTPUT -o sixxs -j ACCEPT

# Zugriff aus dem lokalen Subnetz ebenfalls
erlauben
ip6tables -A INPUT -i eth0   -j ACCEPT
ip6tables -A OUTPUT -o eth0   -j ACCEPT

# Source-Routing-Pakete gefaehrlich, nicht
zugelassen
ip6tables -A INPUT -m rt --rt-type 0 -j
DROP
ip6tables -A FORWARD -m rt --rt-type 0 -j
DROP
ip6tables -A OUTPUT -m rt --rt-type 0 -j
DROP

# Verbindungslokale Adressen erlaubt
ip6tables -A INPUT -s fe80::/10 -j ACCEPT
ip6tables -A OUTPUT -s fe80::/10 -j ACCEPT

# Multicast-Pakete zulassen
ip6tables -A INPUT -s ff00::/8 -j ACCEPT
ip6tables -A OUTPUT -s ff00::/8 -j ACCEPT

# ICMP-Protokoll zur Fehlersuche zulassen
ip6tables -I INPUT  -p icmpv6 -j ACCEPT
ip6tables -I OUTPUT -p icmpv6 -j ACCEPT
ip6tables -I FORWARD -p icmpv6 -j ACCEPT

# Uneingeschraenkter Zugriff auf den
IPv6-Tunnel aus dem Subnetz
ip6tables -A FORWARD -m state --state NEW
-i eth0 -o sixxs -s 2a01:198:514::/48 -j
ACCEPT
ip6tables -A FORWARD -m state --state
ESTABLISHED,RELATED -j ACCEPT

# Ueber den Tunnel eingehende
SSH-Verbindungen zulassen
#ip6tables -A FORWARD -i sixxs -p tcp -d
2a01:198:514::1  --dport 22 -j ACCEPT

# Ueber den Tunnel eingehenden
Bittorrent-Traffic erlauben
ip6tables -A FORWARD -i sixxs -p tcp -d
2a01:198:514::1 --dport 33600:33604 -j
ACCEPT

# Alles andere ist verboten
ip6tables -P INPUT   DROP
ip6tables -P FORWARD DROP
ip6tables -P OUTPUT  DROP
```

IPv6 kommt

Ist der Einstieg in die IPv6-Welt erst einmal geschafft, fällt die Konfiguration der passenden Umgebung kaum schwerer als mit IPv4. Langfristig werden Sie um den Einsatz des neuen Protokolls sowieso nicht herumkommen, sodass sich die Beschäftigung mit der neuen Technik bereits jetzt lohnt. *(jlu/ofr)* ■

Infos

[1] Sixxs: [https://www.sixxs.net]

[2] Aiccu: [https://www.sixxs.net/tools/aiccu/]

[3] Netfilter-Tutorial: [http://www.frozentux.net/documents/iptables-tutorial/]

[4] DNS-Server-Konfiguration für IPv6: [http://www.sixxs.net/faq/dns/?faq=ipv6glue]

Listing 6: Apache-Host

```
<VirtualHost 192.168.0.1:80>
ServerAdmin webmaster@worlddomination.org
DocumentRoot /var/www/html/worlddomination/
ServerName www.worlddomination.org
ErrorLog logs/worlddomination.org-error_log
TransferLog logs/worlddomination-access_log
</VirtualHost>
```

Listing 7: Netstat

```
# netstat -tln |grep '80\>'
tcp  0  0  192.168.0.99:80              0.0.0.0:*
LISTEN
tcp  0  0  2a01:198:514:0:21b:77ff::80  :::*
LISTEN
```

Listing 8: IPv6-DNS

```
# dig +short -t A www.google.com
www.l.google.com.
74.125.43.103
74.125.43.99
74.125.43.105
74.125.43.106
74.125.43.104
74.125.43.147

# dig +short -t AAAA ipv6.google.com
ipv6.l.google.com.
2a00:1450:8006::63
2a00:1450:8006::67
2a00:1450:8006::69
2a00:1450:8006::68
2a00:1450:8006::93
2a00:1450:8006::6a
```