

Internet Protocol v6

Ingo Blechschmidt

Linux User Group Augsburg e. V.

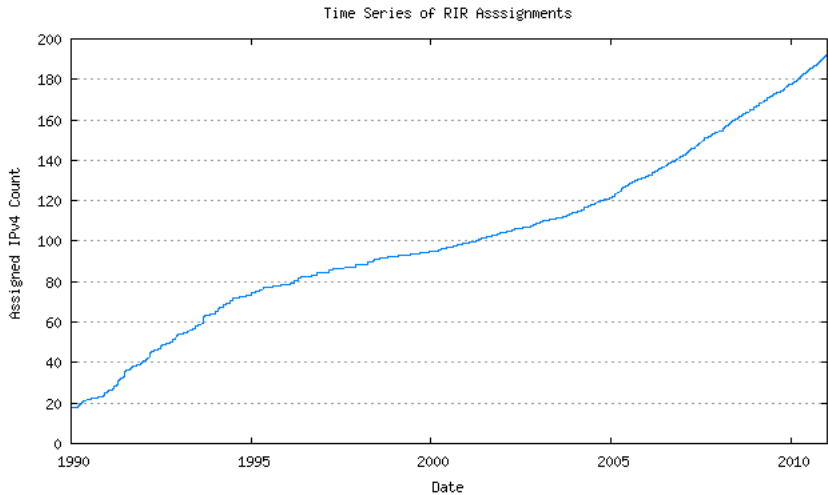
5. Januar 2011

Inhalt

- 1 Probleme von IPv4
 - Adressknappheit
 - Weitere Probleme
- 2 Überblick über IPv6
 - Plan zum Übergang zu IPv6
 - Adressen
 - Autokonfiguration
 - Mobilitätserweiterungen
- 3 IPv6 unter Linux
 - Clients
 - Router
 - IPv6-Zugang über Tunnel

Problem von IPv4: Adressknappheit

- Grundlage der IPv4-Internetkommunikation:
Eindeutige Adressierung eines jeden Computers
durch IP-Adressen
z. B. 213.179.141.18, 80.81.7.3, ...
- Problem:
Es gibt nur $256^4 = 2^{32}$ (≈ 4 Mrd.) Adressen,
wobei viele auch für spezielle Zwecke reserviert
sind.
- Prognose:
Erschöpfung des IP-Adressraums im November
2011.





IANA END Date RIR END Date

Weitere Probleme von IPv4

- Oftmals wegen Network Address Translation (NAT) keine Ende-zu-Ende-Kommunikation
- Keine automatische Adresskonfiguration
- Verschlüsselung und Signierung nicht standardmäßig
- Kein „Roaming“ bei Netzwerkwechseln (mobile Geräte)

Lösung: IPv6

- Lösung all dieser Probleme: IPv6
- Erste Veröffentlichung: RFC 2460 (1998)
- Hauptmerkmal:
Erweiterung der Adresslänge auf 128 Bit
↪ theoret. 2^{128} (\approx 340 Sextillionen) Adressen
- Viele weitere Verbesserungen!
- Aber: Nicht abwärtskompatibel mit IPv4 (gleich)
- (v5 vergeben für Internet Stream Protocol v2)

Migrationsplan

- Offensichtlich: Vollständige Umstellung auf IPv6 zu einem festen Zeitpunkt unmöglich
- Stattdessen: Lange Übergangsphase mit Dual-Stack-Betrieb
- Erhalt von IPv4- und IPv6-Adressen durch Internetanbieter (Parallelnutzung problemlos möglich)
- Bevorzugung von IPv6-Verbindungen durch Clients
- Irgendwann: Abschaltung von IPv4

Adressen bei IPv6

- 128 Bit, geschrieben als acht Blöcke von je vier Hexziffern
- Beispiel:
2001:08e0:abcd:014d:0000:0000:0000:0001
2001:8e0:abcd:14d:0:0:0:1
2001:8e0:abcd:14d::1

- Konvention:

Vordere 64 Bit: „Präfix“, identifiziert das Subnetz
Hintere 64 Bit: Interface ID, identifiziert eine Netzwerkschnittstelle innerhalb eines Netzes

- Üblich: Mehrere Adressen pro Interface

Adressen bei IPv6 (Forts.)

- Nebenbei: Können so kurz wie v4-Adressen sein:
2001:388:f000::285 (5+13 Zeichen) vs.
60.234.76.50 (12 Zeichen)
- Spezielle Adressen:
 $::0 \hat{=} :: \hat{=} 0.0.0.0$
 $::1 \hat{=} 127.0.0.1$
- Notation für Netze, Beispiel:
Zum Netz *2000::/3* gehören genau die Adressen, deren vorderste drei Bits mit denen von *2000::* übereinstimmen, also *2000::* bis *3fff:....*

Vergebene Adressbereiche (Auszug)

Netz	Bereich	Zweck
::/4	0000:: bis 0fff:...	Verschiedenes
2000::/3	2000:: bis 3fff:...	Global Unicast
fc00::/7	fc00:: bis fdff:...	Unique Local Unicast
fe80::/10	fe80:: bis febf:...	Link-Local Unicast
ff00::/8	ff00:: bis ffff:...	Multicast

- Netze von Internetanbietern für Endnutzer:
mindestens /64, besser /48
- Bei /64: Nur Interface ID (hintere 64 Bits) frei wählbar, also nur ein Subnetz möglich
- Bei /48: $2^{64-48} = 2^{16} = 65.536$ Subnetze möglich

Private Adressen

- Private Adressen bei IPv4:
192.168.0.0/16, 172.16.0.0/12, 10.0.0.0/8
- Probleme:
private Adressen können ins Internet gelangen,
nicht weltweit eindeutig
- Bei IPv6: Unique Local Unicast

<i>fc</i> oder <i>fd</i>	eindeutige Site-ID	Subnetz-ID
8 Bits	40 Bits	16 Bits

/64
- Beispiel: *fc9e:21a7:a92c:2323::1*

Sicherheitsbedenken

- Philosophie von IPv6:
Öffentliche IP-Adressen für alle Geräte;
keine Network Address Translation
- Somit Wegfall des gewissen Schutzes durch NAT
- Einfache Abhilfe: Filterung aller eingehenden,
nicht angeforderten Verbindungen durch Router

- Nebenbei: Absuchen von Netzwerken nach
erreichbaren Computern bei IPv6
unpraktikabel, 2^{64} (≈ 18 Trillionen)
durchzuprobierende Adressen

Autokonfiguration

- Adresskonfiguration bei IPv4:
manuell oder DHCP
- Probleme:
DHCP ist nicht zustandslos,
keine Unterstützung für Adressmigration

- Bei IPv6: Zustandslose Autokonfiguration durch
link-lokale Adressen und Router Discovery
- Clients konfigurieren sich selbst,
keine Verwaltung durch einen Server nötig

Link-lokale Adressen

- Netz *fe80::/10* für nicht-geroutete link-lokale Adressen (ähnlich wie *169.254.0.0/16* bei IPv4)
- Automatische Einrichtung einer link-lokalen Adresse für jede Netzwerkschnittstelle, über ihre EUI-64 (grob: = MAC)
- Duplicate Address Detection (DAD) zur Vermeidung von Adresskollisionen

- Beispiel:

MAC-Adresse *00:e0:7d:e5:0b:ff*

↪ IPv6-Adresse *fe80::2e0:7dff:fee5:bff*

Router Discovery

- Nach Einrichtung einer link-lokalen Adresse:
Senden einer Router Solicitation (ICMPv6) an die Multicast-Adresse *ff02::2* (alle Router des Netzsegments)
- Antwort: Router Advertisement, enthält zulässiges Netzpräfix (64 Bits)
- Wahl von *Netzpräfix : Interface ID* als IP-Adresse
- Wieder Duplicate Address Detection

Optionale Ergänzung: DHCPv6

- Zustandsloses DHCPv6 als optionale Ergänzung, beispielsweise zur Bekanntgabe von DNS-, Zeit- oder Druckerservern
- Oftmals wegen Multicast überflüssig, beispielsweise bezeichnet *ff05::101* alle NTP-Server derselben Site
- DNS-Angabe in Router Advertisements vorgeschlagen (experimenteller RFC 5006)

Datenschutzerweiterungen

- Problem: Feste Interface ID in IP-Adresse, eindeutige Identifizierbarkeit möglich
- Abhilfe: Privacy Extensions, Clients geben sich zusätzlich temporäre Adressen mit zufälligem Interface-Teil
- Bei Kombination mit zeitlich veränderlichem Präfix gleiche Anonymität wie bei IPv4

Mobilitätserweiterungen

- Problem bei IPv4:
Bei Netzwechsel Änderung der IP-Adresse,
somit Abbruch aller bestehenden Verbindungen
- Lösung bei IPv6: stets gültige Heimatadresse
- Weiterleitung von an die Heimatadresse
gesendeten Paketen an jeweils aktuelle Adresse
durch einen Heimatagenten (bspw. den
Heimrouter)
- Nach erster Kontaktaufnahme direkte
Kommunikation ohne Heimatagent

IPv6 auf Clients

- Falls ipv6-Kernelmodul geladen und IPv6-Router vorhanden:
Keine weitere Einrichtung erforderlich,
Autokonfiguration erledigt alles!
- Warnung: Öffentliche Erreichbarkeit des Clients
bei funktionierendem IPv6-Router,
Firewall-Einrichtung nötig!

IPv6 auf Routern

- Zum Reagieren auf Router Solicitations: `radvd`
- `/etc/radvd.conf`:

```
interface eth0 {  
    AdvSendAdvert on;  
    prefix 2001:db8:dead:c0de::/64 {  
        AdvRouterAddr on;  
    };  
};
```

- `# echo 1 > /proc/sys/net/ipv6/conf/all/forwarding`

IPv6-Zugang über 6to4

- 6to4-Gateway unter Anycast-IPv4-Adresse
192.88.99.1
- Damit erhält man ein ganzes /48er-Netzwerk unterhalb von 2002!
- Direkte IPv4-Verbindung (ohne NAT) benötigt

- Beispiel:

```
# ip tunnel add tun6to4 mode sit ttl 64  
    remote any local 84.157.46.16
```

```
# ip link set dev tun6to4 up
```

```
# ip -6 addr add 2002:549d:2e10::/16 dev tun6to4
```

```
# ip -6 route add 2002::/3 via ::192.88.99.1  
    dev tun6to4 metric 1
```

(549d:2e10 ist einfach 84.157.46.16 im Hexformat,
84.157.46.16 die öffentliche IPv4-Adresse.)

IPv6-Zugang über Teredo

- Falls IPv4-Verbindung nur hinter NAT vorhanden: Teredo
- Tunnelung des IPv6-Verkehrs über einen öffentlichen Teredo-Server, bspw. *teredo.ipv6.microsoft.com* oder *teredo-debian.remlab.net*
- Aber: Anders als bei 6to4 keine Möglichkeit, ein gesamtes Subnetz ins IPv6-Internet zu bringen
- *# apt-get install miredo*
- Mehr ist nicht zu tun, schon die Standardkonfiguration funktioniert!

Nicht erwähnt

- Protokollvereinfachungen, u. a. für höhere Routingeffizienz
- Besserer Umgang mit Paketfragmentierung
- IPv6 und DNS
- Neighbor Discovery als Teil von ICMPv6 und besseren ARP-Ersatz
- Secure Neighbor Discovery (SEND, RFC 3971)
- Funktionsweise von Duplicate Address Detection
- Multicast
- Jumbogramme
- Sicherheitsbedenken

Siehe auch

- **Statistik über die IPv4-Adressknappheit:**
<http://www.potaroo.net/tools/ipv4/>
- **Wikipedia zu IPv6:**
<http://en.wikipedia.org/wiki/IPv6>
- **Vortrag mit mehr Details:**
<http://www.openwall.com/presentations/IPv6/>
- **6to4 unter Linux:**
<http://www.atoomnet.net/ipv62002.php>
<http://www.wuglug.org.uk/IPv6>
- **IPv6-only-Webseiten zum Testen:**
<http://ipv6.google.com/>,
<http://six.heise.de/>