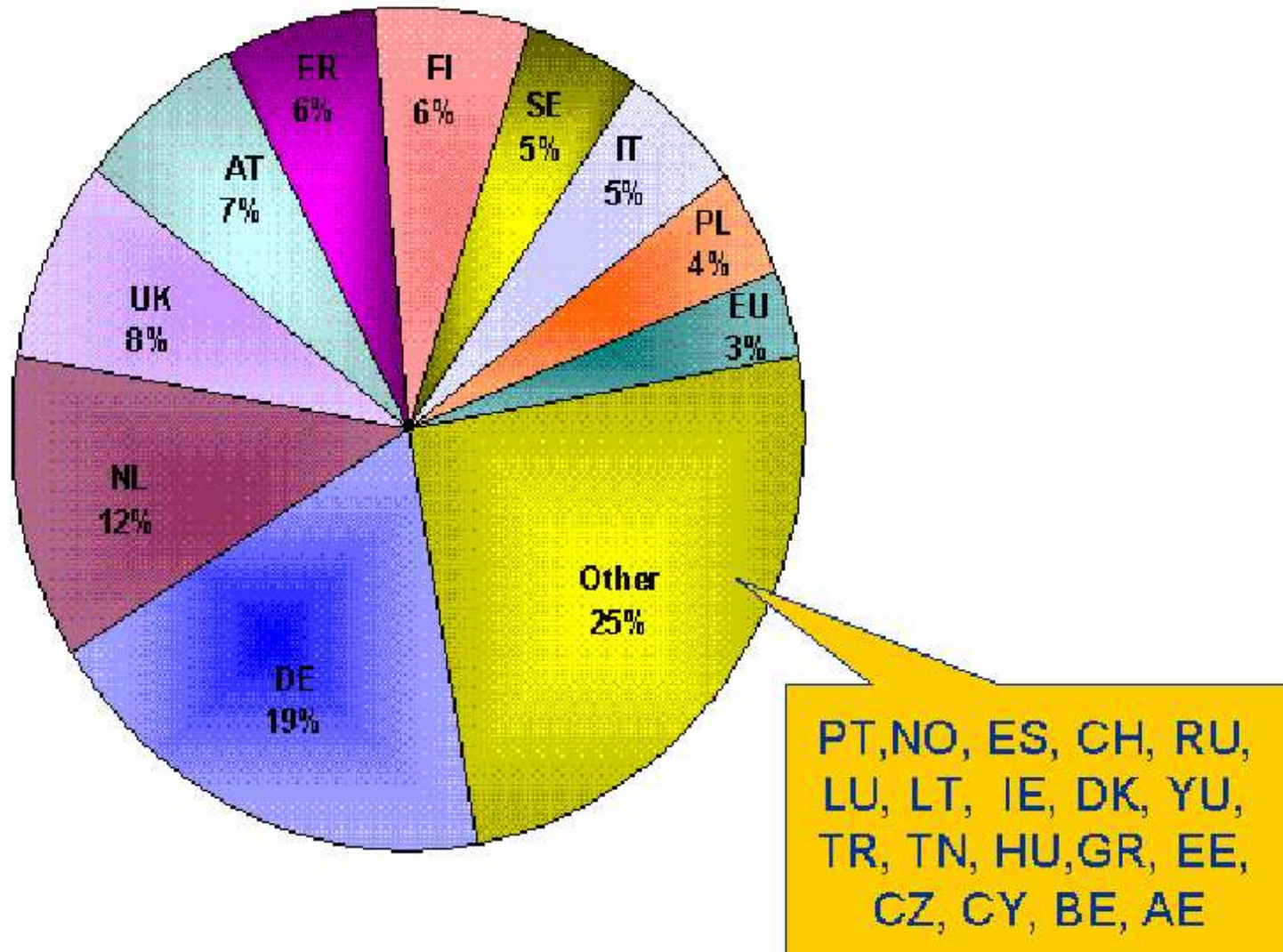


Einführung in IPv6

{chrome|mm|pablo|ruediger|tho}@koeln.ccc.de

- Überblick über Neuerungen
 - Ziele des neuen Protokolls
 - IPv6 Protokoll-Suite
 - IPv6 – Adressen
- Migration von IPv4 zu IPv6
 - 6Bone
 - 6to4
 - Tunnelbroker
- Praktischer Einsatz für IPv6 unter Linux/Windows
- RTFM

IPv6 Allocation by country



Quelle: <http://www.ripe.net/rs/statistics/>

RFCs zu IPv6

- S. Deering, R. Hinden:
Internet Protocol, Version 6 (IPv6) Specification
RFC 2460, December 1998.
- R. Hinden, S. Deering:
IP Version 6 Addressing Architecture
RFC 2373, July 1998.
- ICMPv6 (*RFC 2463*), OSPF for IPv6 (*RFC 2740*), IP Encapsulating Payload (ESP) (*RFC 2406*), Mobility Support in IPv6 (*draft-ietf-mobileip-ipv6-13.txt*) . . .

Probleme bei IPv4

- Verfügbare Adressen gehen zu Neige (~ 2006–2011)
- Routingtabellen der core–Router werden zu groß /
Performanceprobleme bei den Routern
- (Fast) fehlende Unterstützung für mobile Knoten
- Keine besonderen Sicherheitsfeatures
- Schlechte Ausnutzung des Adressraums (\Rightarrow NAT)

Ziele von IPv6 (I)

- Expanded Addressing Capabilities

128 Bit–Adressen (IPv4 hat 32 Bit–Adressen),
Autokonfiguration, Anycast–Adressen

- Header Format Simplification

Einige IPv4–Header wurden entfernt, schnellere Bearbeitung
in den Routern möglich

- Improved Support for Extensions and Options

Umstrukturierung des IP–Headers, größere Flexibilität

Ziele von IPv6 (II)

- Flow Labeling Capability

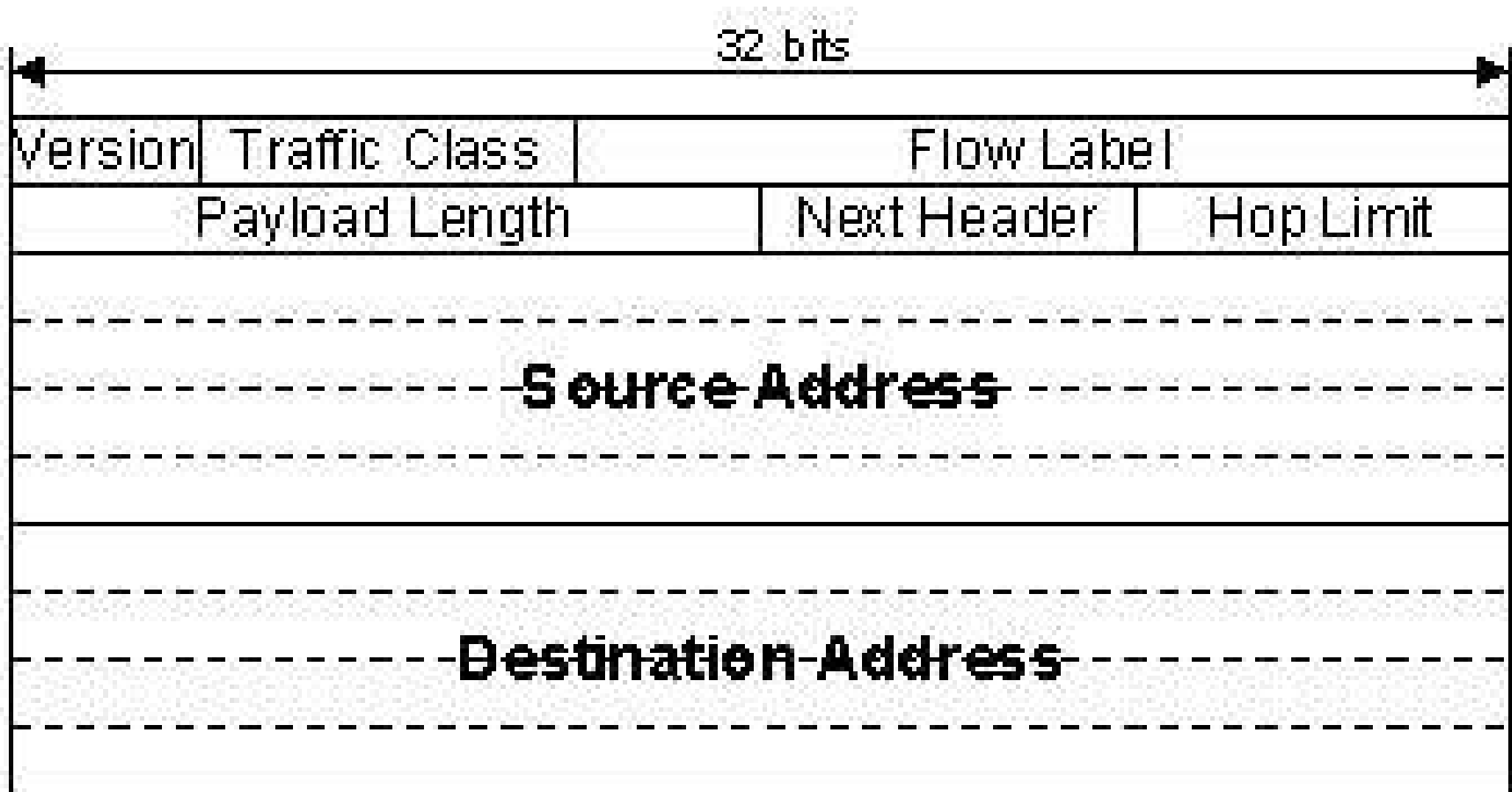
So genannte *flows* erlauben verbesserten QoS und real-time–Anwendungen

- Authentication and Privacy Capabilities

Authentifizierung, Integrität, (optional) Vertraulichkeit [confidentiality]

- Leichte Migration von IPv4 zu IPv6 / Koexistenz beider Systeme

IPv6 Protokoll-Suite



IPv6–Adressen

- 128 Bit Adressen $\approx 3,4 * 10^{38}$ Adressen

Beispielsweise: apple.kame.net —

2001:200:0:8002:210:f3ff:fe03:4d0

- Anzahl Adressen mehr als ausreichend
- Führende Nullen können weggelassen werden
- Einmalig kann Serie von Nullen durch :: abgekürzt werden

FF01:0:0:0:0:0:0:43 → FF01::43

IPv6-Adresstypen

- **Unicast:** In der Regel einem NIC zugeordnet
Spezielle Unicast-Adressen:
loopback (::1), unspecified (::)
- **Multicast:** Adressiert Menge von NICs, ähnlich wie IPv4
Einsatz beispielsweise für Multimedia-Anwendungen
- **Anycast:** Adressiert auch Menge von NICs, Nachricht wird jedoch an einen beliebigen NIC verschickt

Link-Local/Site-Local Adressen

- link-local gedacht für den Einsatz im lokalen Netz
- Kein Routing von link-local Adressen
- link-local Adressen haben Präfix aus dem Bereich fe80 – febf
Aktuell wird nur fe80 genutzt
- site-local ähnlich wie private Adressräume in IPv4
- Präfix fec0 – feff, aktuell wir nur fec0 benutzt

Globale/Multicast Adressen

- Globale Adressen beginnen mit $2xxx$ bzw. $3xxx$
- Momentan gebräuchliche Präfixe:
 - $3ffe$ → 6Bone
 - 2002 → 6to4
 - 2001 → Produktionseinsatz
- Multicast Adressen beginnen mit $ffxy$
 x bzw y geben Multicast–Scope bzw –Typ an

Broadcast

- IPv4-Broadcast ist in IPv6 ein Multicast
- Ping an alle Systeme link-local:

```
$ping6 -I eth0 ff02::1
```

- Ping an alle Router link-local:

```
$ping6 -I eth0 ff02::2
```

Address Autoconfiguration

- Rolle von DHCP übernimmt *stateful address configuration*
Mittels *router advertisements* wird Konfiguration verbreitet
- Neuheit ist *stateless address configuration* mit ICMP6
Gebildet aus MAC-Adresse, Kontrolle auf Eindeutigkeit
mittels *neighbour solicitation* mit Absender-Adresse ::
Falls Adresse bereits vergeben ist, muss NIC per Hand
konfiguriert werden

Neighbour Discovery/Address Resolution

- Durch Neighbour Discovery finden hosts bspw. Router
- Rolle von ARP übernimmt ICMP6 (*router solicitation*)
- Router Discovery:

Router schicken ICMP6–Nachricht (*router advertisement*) an FF02::1

Clients können mit *router solicitation* an FF02::2 ein *router advertisement* anfordern

IPv6 Extension Headers

- IPv4 hat *options field*, wird aber nicht oft benutzt
- IPv6 definiert *extension header*:
 - Folgen nach IPv6 header und vor transport header/payload
 - Optional, damit flexibel
 - Unterstützung für security, source routing, fragmentation, network management...
 - Extension header werden meistens auf dem Weg zum Ziel nicht beachtet \Rightarrow schnellere Verarbeitung an den Routern

6Bone

- Geschaffen um Migration von IPv4 nach IPv6 zu vereinfachen
- Testnetz für IPv6
- Alle IPv6-Adressen mit Präfix 3ffe:: (bspw. freenet6)
- 6Bone läuft (vermutlich) noch bis 1. Juli 2006
Danach Abschaltung und alle Systeme müssen andere Adresse benutzen
- `apt-get install freenet6 ⇒ 6Bone`

- RFC 3056
- Adressen für IPv6, basierend auf IPv4-Adressen
- Kein registrierter Tunnel
- IPv6-Pakete werden über IPv4-Pakete transportiert (Kapselung)
- Aus einer IPv4-Adresse wird ein /48 IPv6-Netz 2^{16} subnets mit jeweils 2^{64} hosts

IPv4-Adresse \Rightarrow 6to4-Adresse:

```
$printf "2002:%02x%02x:%02x%02x::/48\n" 134 130 118 178  
2002:8682:76b2::/48
```

- Komplettes 2002:8682:76b2::/48 steht nun als Netz zur Verfügung
- Benötigt wird noch IPv6-Adresse eines 6to4 Relay Router
- Anycast-Adresse des nächsten Routers: 2002:c058:6301:::
(RFC 3068)

Konfiguration unter RedHat

```
/etc/sysconfig/network
```

```
NETWORKING_IPV6=yes
```

```
IPV6_DEFAULTDEV=tun6to4
```

```
/etc/sysconfig/network-scripts/ifcfg-eth0
```

```
IPV6INIT=" `yes" `
```

```
IPV6TO4INIT=" `yes" `
```

```
IPV6TO4_RELAY=192.88.99.1
```

Tunnelbroker

- So genannte Tunnelbroker vermitteln IPv6 Adressen/Subnetz
- Teilweise werden 2001-Adressen vergeben, teilweise nur 3ffe
- Beispiele für Tunnelbroker
 - <http://www.sixxs.net>
 - <http://www.freenet6.net>
`#apt-get install freenet6`
Registrierung für Subnetz

IPv6 mit Linux

```
$ip -f inet6 addr show
```

```
...
```

```
2: eth0: <BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast \
                                   qlen 100
```

```
    inet6 fe80::200:e8ff:feed:4024/10 scope link
```

```
4: sit1@NONE: <POINTOPOINT,NOARP,UP> mtu 1480 qdisc noqueue
```

```
    inet6 3ffe:b80:3:3301::2/128 scope global
```

```
    inet6 fe80::8682:76b2/10 scope link
```

```
...
```

IPv6 mit Linux

Kernel muss IPv6 unterstützen (/proc/net/if_inet6 existiert)

```
$grep -i ipv6 /usr/src/linux/.config
```

```
CONFIG_IPV6=m
```

...

Entweder fest einkompilieren oder besser als Modul

```
#modprobe ipv6
```

```
$ip addr show
```

zeigt device sit0 sowie inet6-Adresse bei ethx

Fertig :-)

IPv6 mit Linux

In /etc/network/interface:

```
auto sixxs
```

```
iface sixxs inet6 v4tunnel
```

```
    address 2001:768:1900:43::2
```

```
    netmask 64
```

```
    endpoint 195.143.155.2
```

```
    ttl 64
```

```
up ip link set mtu 1280 dev sixxs
```

```
up ip route add 2000::/3 via 2001:768:1900:43::1 dev sixxs
```


IPv6 mit Linux

In /etc/network/interface:

```
iface tunnel4 inet6 v4tunnel
```

```
    address 2001:768:191f:ffff::1
```

```
    netmask 64
```

```
    endpoint 213.168.105.158
```

```
    ttl 64
```

```
up ip route add 2001:768:191f:ffff::2/64 dev tunnel4
```

```
up ip route add 2001:768:191f:1000::/64 dev tunnel4
```

```
up ip tunnel change tunnel1 ttl 64
```

Performance per freenet6

```
$ping6 -c 3 apple.kame.net
```

```
PING apple.kame.net (2001:200:0:8002:210:f3ff:fe03:4d0) 56 data  
64 bytes from apple.kame.net: icmp_seq=1 ttl=54 time=425 ms  
64 bytes from apple.kame.net: icmp_seq=2 ttl=54 time=407 ms  
64 bytes from apple.kame.net: icmp_seq=3 ttl=54 time=425 ms
```

```
--- apple.kame.net ping statistics ---
```

```
3 packets transmitted, 3 received, 0% packet loss, time 2020ms  
rtt min/avg/max/mdev = 407.788/419.672/425.627/8.436 ms
```

Zum Vergleich per IPv4:

```
rtt min/avg/max/mdev = 279.491/281.699/286.071/3.151 ms
```

Performance per SixXS

```
$ping6 -c 3 apple.kame.net
```

```
PING apple.kame.net (2001:200:0:8002:210:f3ff:fe03:4d0) 56 data
```

```
64 bytes from apple.kame.net: icmp_seq=1 ttl=49 time=289 ms
```

```
64 bytes from apple.kame.net: icmp_seq=2 ttl=49 time=285 ms
```

```
64 bytes from apple.kame.net: icmp_seq=3 ttl=49 time=285 ms
```

```
--- apple.kame.net ping statistics ---
```

```
3 packets transmitted, 3 received, 0% loss, time 10590ms
```

```
rtt min/avg/max/mdev = 285.322/286.976/289.887/2.064 ms
```

Zum Vergleich per IPv4:

```
rtt min/avg/max/mdev = 284.580/285.263/286.531/0.897 ms
```

IPv6 mit Window\$

Pablo erzählt ein bisschen zu IPv6 unter Window\$

Siehe: <http://research.microsoft.com/msripv6/>

Programme mit IPv6-Support

- zebra, tinydns, bind...
- konqueror, galeon, lynx, w3m...
- apache, sendmail, ssh...
- lftp, bsd-ftpd, lukemftp...
- libc6
- iproute, tcpdump, ping6...

- [/usr/share/doc/HOWTO/en-txt/Linux+IPv6-HOWTO.gz](#)
- <http://people.debian.org/~csmall/ipv6/>
- <http://www.sixxs.net/>
- <http://debian.fabbione.net/>
- Comer, Douglas: Internetworking with TCP/IP (1–3),
- RFCs 2460, 2373, 2463, 2406...

So long and thanks for all the fi sh

Fragen? Anmerkungen? Kritik?

⇒ Jetzt oder später an

{chrome|mm|pablo|ruediger|tho}@koeln.ccc.de