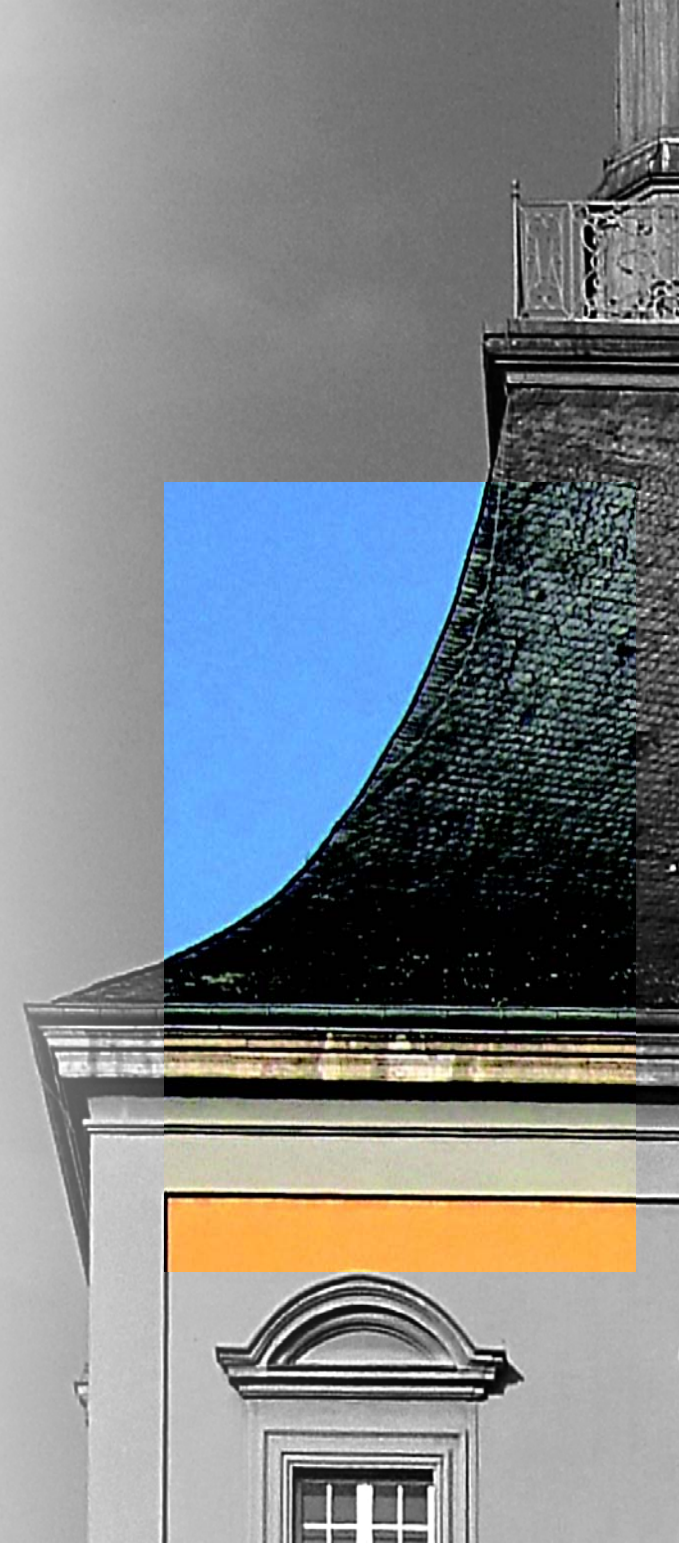


VORLESUNG
NETZWERKSICHERHEIT

SOMMERSEMESTER 2022
MO. 14-16 UHR



KAPITEL 5

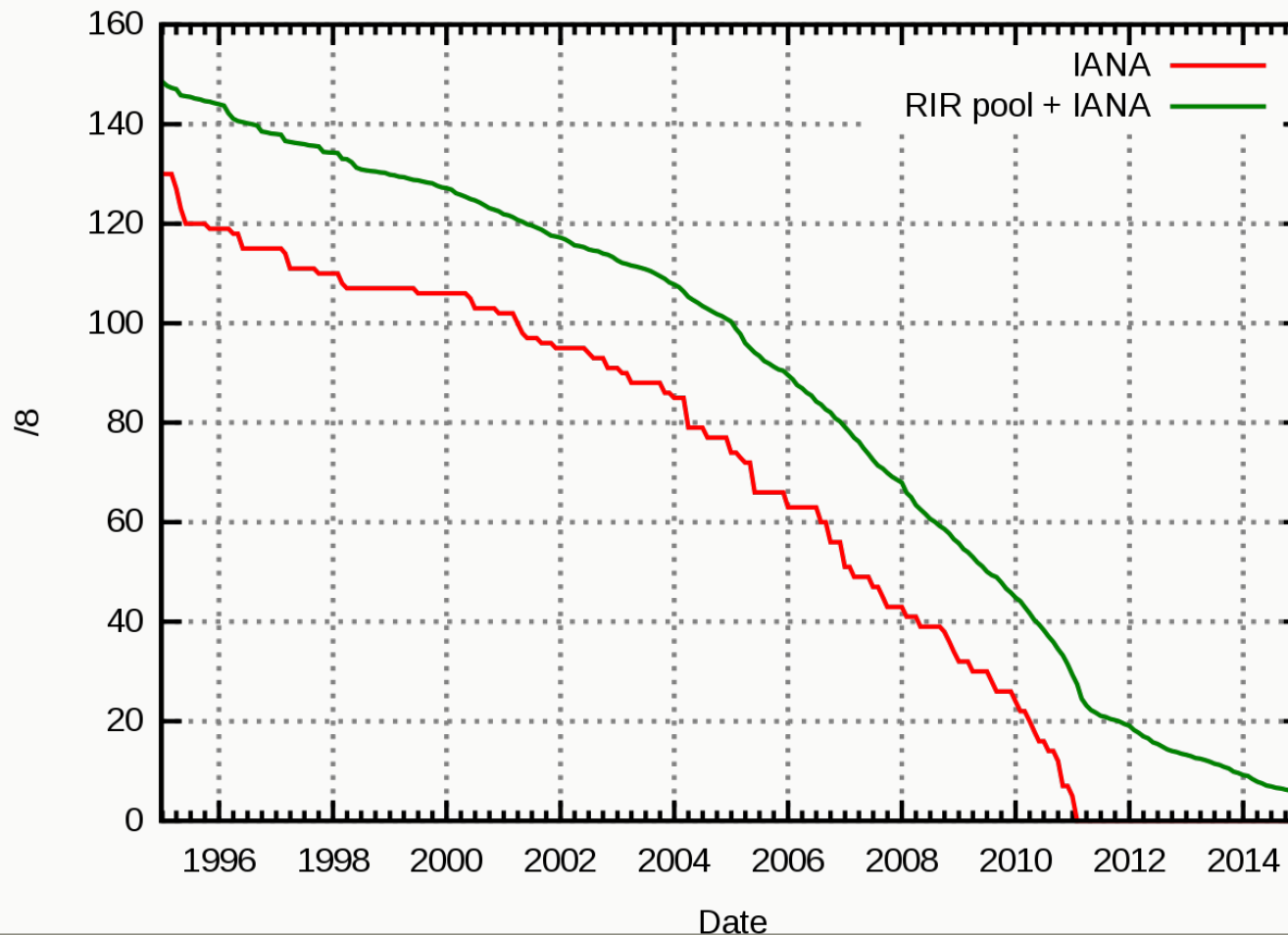
INTERNET PROTOCOL V6

- Motivation
- IPv6 in Action / Migration IPv4 => IPv6
- IPv6-Prokoll
- IPv6-Adressen & -Addressauflösung
- Autokonfiguration / DHCPv6
- ICMPv6
- IPSec
- Fragmentierung
- Multicast
- Netzwerkarchitektur

MOTIVATION FÜR IPV6

Geringe Anzahl IPv4 Adressen

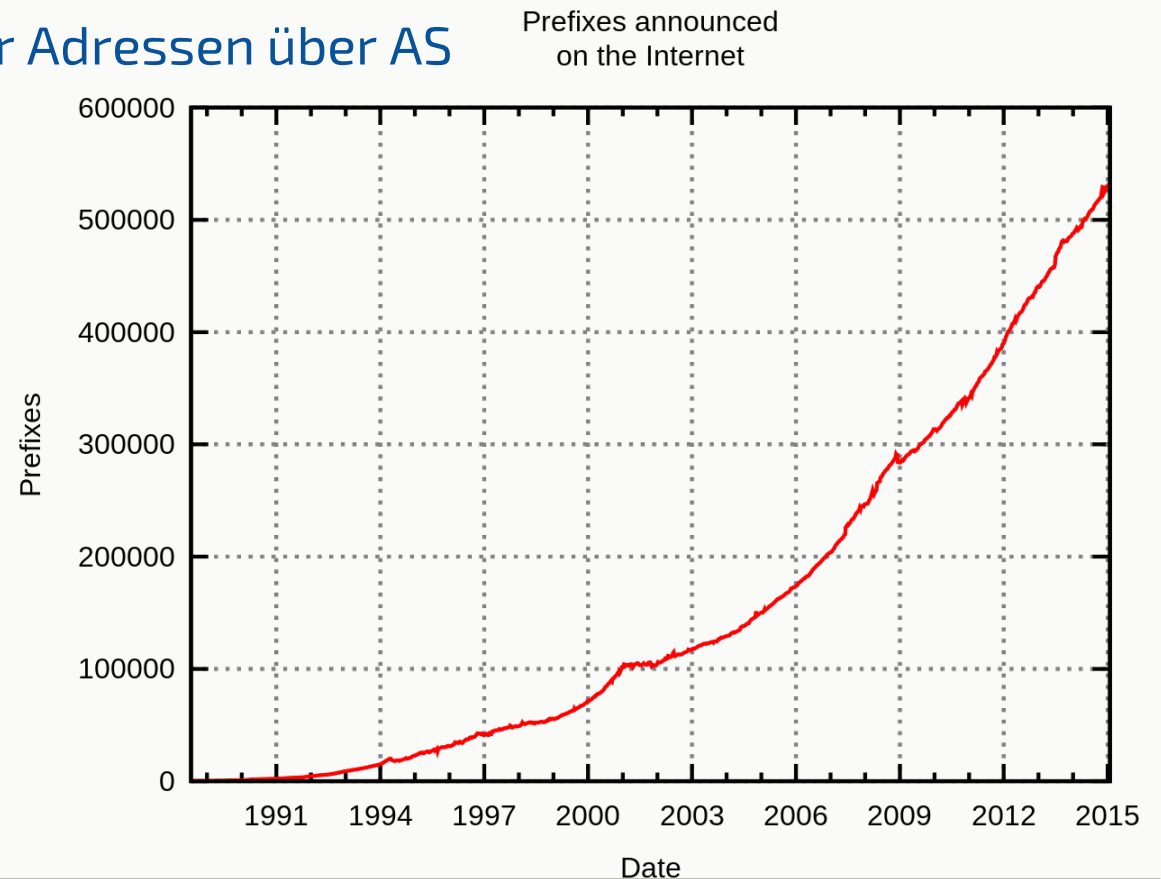
Free /8



MOTIVATION FÜR IPV6

Geringe Anzahl IPv4 Adressen

- Anzahl der Prefixe in Routingtabellen ist sehr groß
 - Grund: Große Fragmentierung der Adressen über AS
 - Probleme: 512k-Day



Geringe Anzahl IPv4 Adressen

- Anzahl der Prefixe in Routingtabellen ist sehr groß
 - Grund: Große Fragmentierung der Adressen über AS
 - Probleme: 512k-Day

Automatische Netzwerkkonfiguration gewünscht

- IPv4LL (169.254.0.0/16) hat sich nicht bewährt

Weniger Komplexität / Mehr Flexibilität

- "Kleinerer Standardheader"
- Einfache Erweiterbarkeit durch optionale Header (Daisy-Chained)

Sicherheit Out-of-the-Box

- IPSec-Header als optionale Header direkt verfügbar

Größerer Adressraum

- Größe der IP-Adresse von 32 Bits auf 128 Bits
 - von 4.294.967.295 IP-Adressen
auf 340.282.366.920.938.463.463.374.607.431.768.211.455

Optimierter Protokollheader

- Verbesserung der Paketweiterleitung

Stateless Autoconfiguration

- Knoten bestimmen / berechnen ihre eigene, weltweit eindeutige Adresse selbst

Multicast

- Verbesserung der 1-zu-n-Kommunikation (Kein Broadcast mehr)

IPV6 VS. IPV4 (FORTS.)

Jumbogramme

- Große Pakete zur Effizienzsteigerung von Nutzdaten

Implizite Sicherheit

- Optionale Header für Authentifikation und Verschlüsselung der Nutzdaten

Quality-of-Service

- QoS-Markierung von Paketen / Priorisierung von Netzverkehr

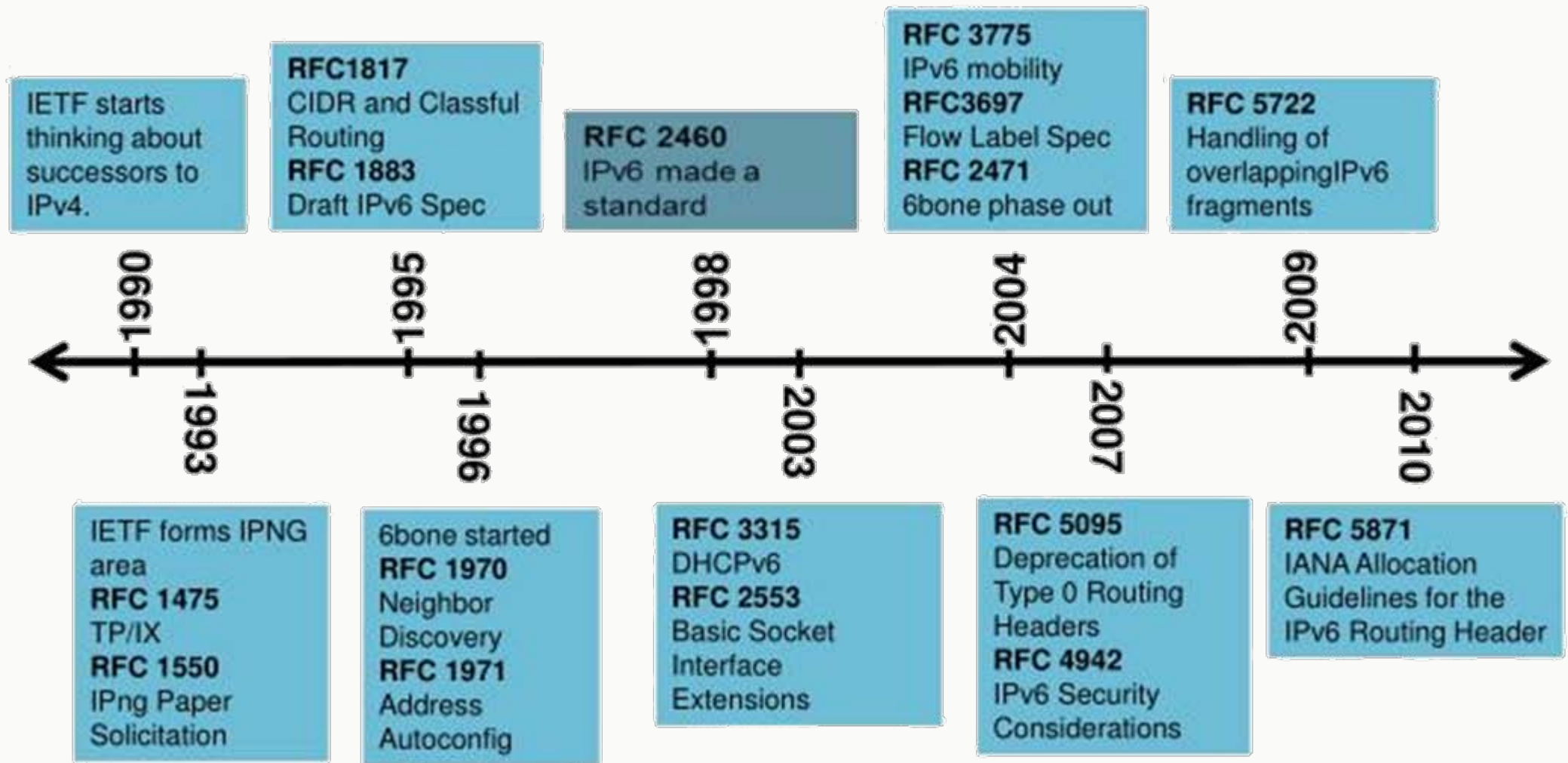
Anycast

- Redundante Dienste ohne eindeutige Adressen

Mobiles Internet

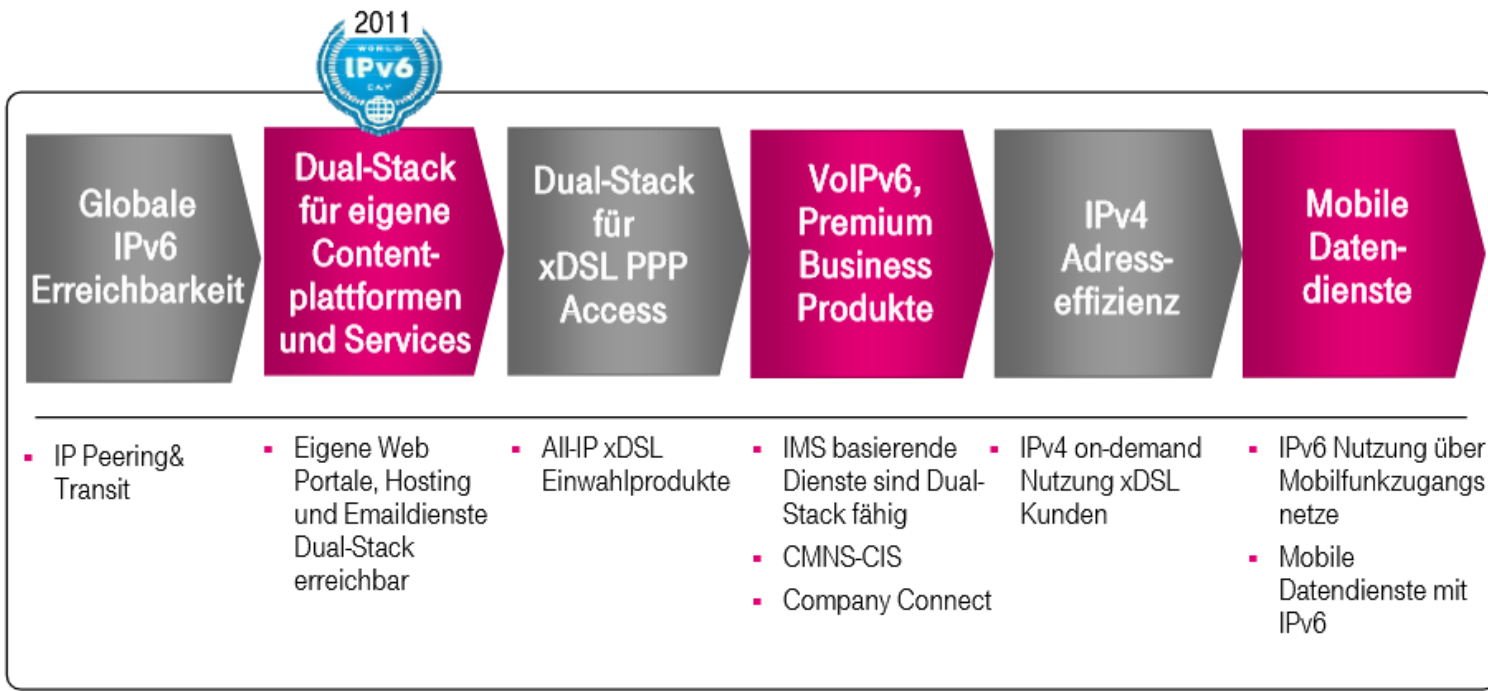
- Bessere Handhabung von mobilen Knoten und Roaming

IPV6-HISTORIE



IPV6 IN ACTION?

Telekom Deutschland – IPv6 für IP Dienste.



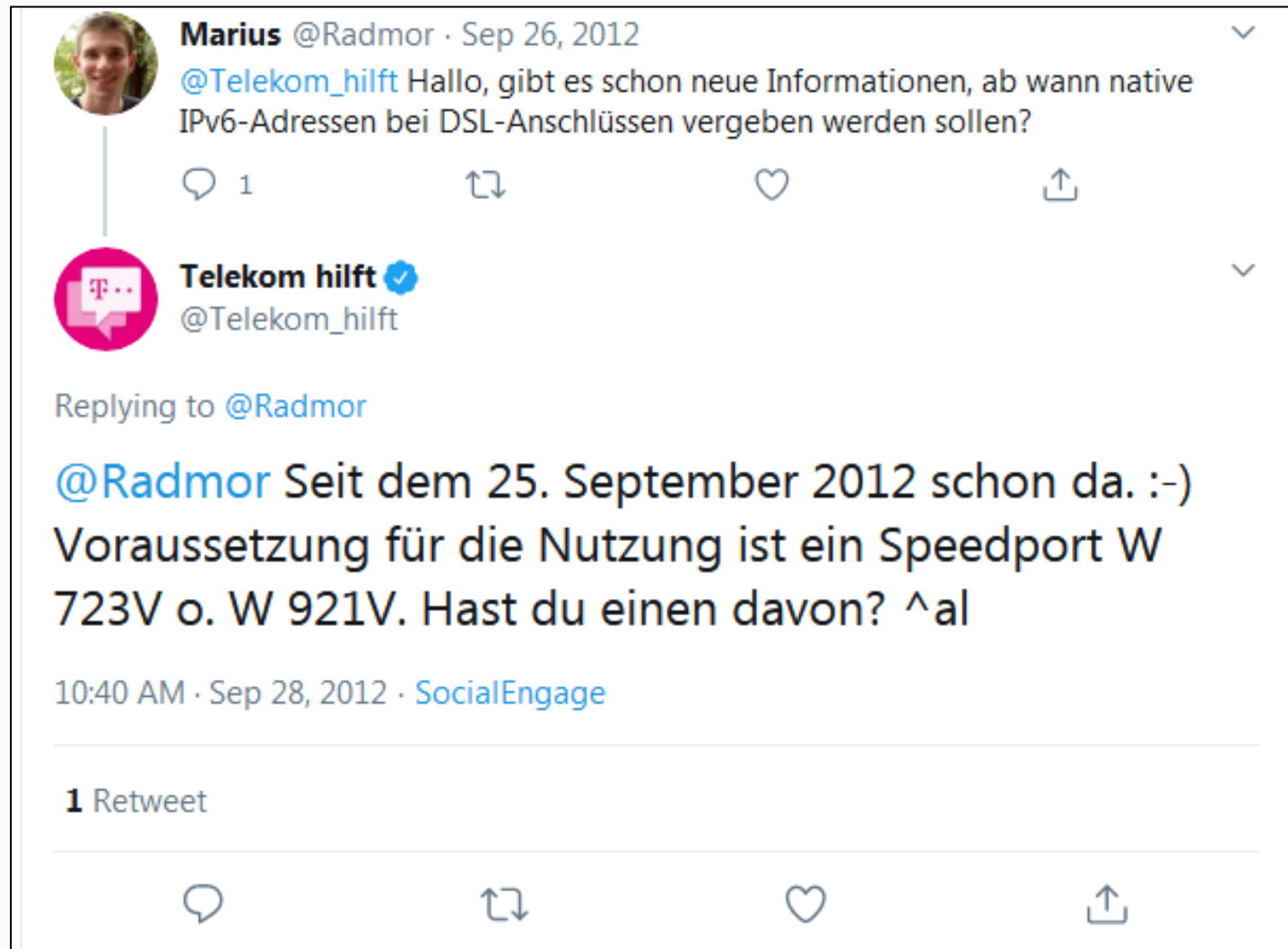
Lösung für begrenzte IPv4 Adressressourcen ist die konsequente Einführung



Deutsche Telekom

26.03.2012

IPV6 IN ACTION



heise online › News › 07/2015 › Telekom startet IPv6-Einführung im Mobilfunknetz

2015

Telekom startet IPv6-Einführung im Mobilfunknetz

Den Angaben zufolge soll die Technik einführen. Um die IPv6-Adresse oder Dateifreigaben h


Lesezeit: 1 Min.  In Pocket

heise online › News › 08/2018 › IPv4-Dämmerung: Telekom testet IPv6-only-Kommunikation im Mobilfunk

2018

IPv4-Dämmerung: Telekom testet IPv6-only-Kommunikation im Mobilfunk

Die technischen Voraussetzungen für den alleinigen Betrieb mit IPv6 gibt es zwar, nun wird aber deren Einsatz in einem öffentlichen Mobilfunknetz absehbar

Lesezeit: 1 Min.  In Pocket



Telekom Mobilfunk: Neuer IPv6-Zugang für alle Kunden

[mobiFlip.de](#) - 29 Jan 2020

Bei Problemen sollte man auf den Standard APN internet.telekom (Dual Stack) zurückwechseln. Telekom schließt zahlreiche Shops. Deutsche ...

Deutsche Telekom: Neuer IPv6-only-Zugang zum mobilen ...

[Caschys Blog \(Blog\)](#) - 29 Jan 2020

[View all](#)

2020

Entwicklung des Standards berücksichtigt Übergangsstrategien

- Langsames und methodisches Vorgehen beim Übergang erwartet

Einführungsphase “6bone”

- 1995 – 2006 (Ende am 6.6.2006)
- Verbindung lokaler IPv6-Netzwerke über IPv4-VPN
- Erlaubte erste Tests, ohne IPv6-fähige Internet-Core-Router

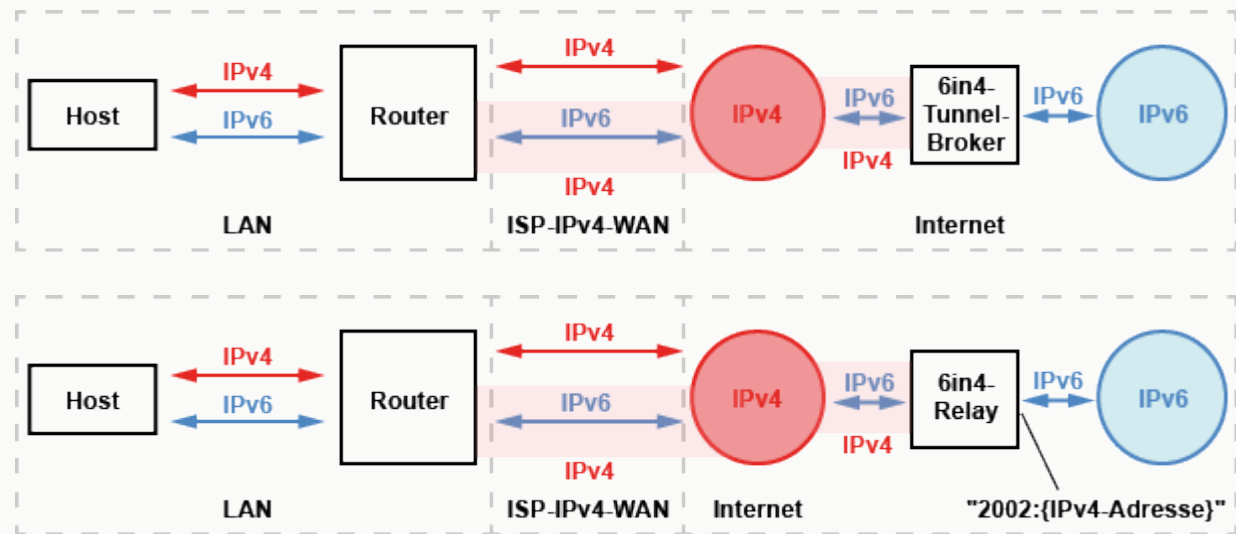
Entwicklung des Standards berücksichtigt Übergangsstrategien

- Langsames und methodisches Vorgehen beim Übergang erwartet

- IPv4-IPv6-Tunnel

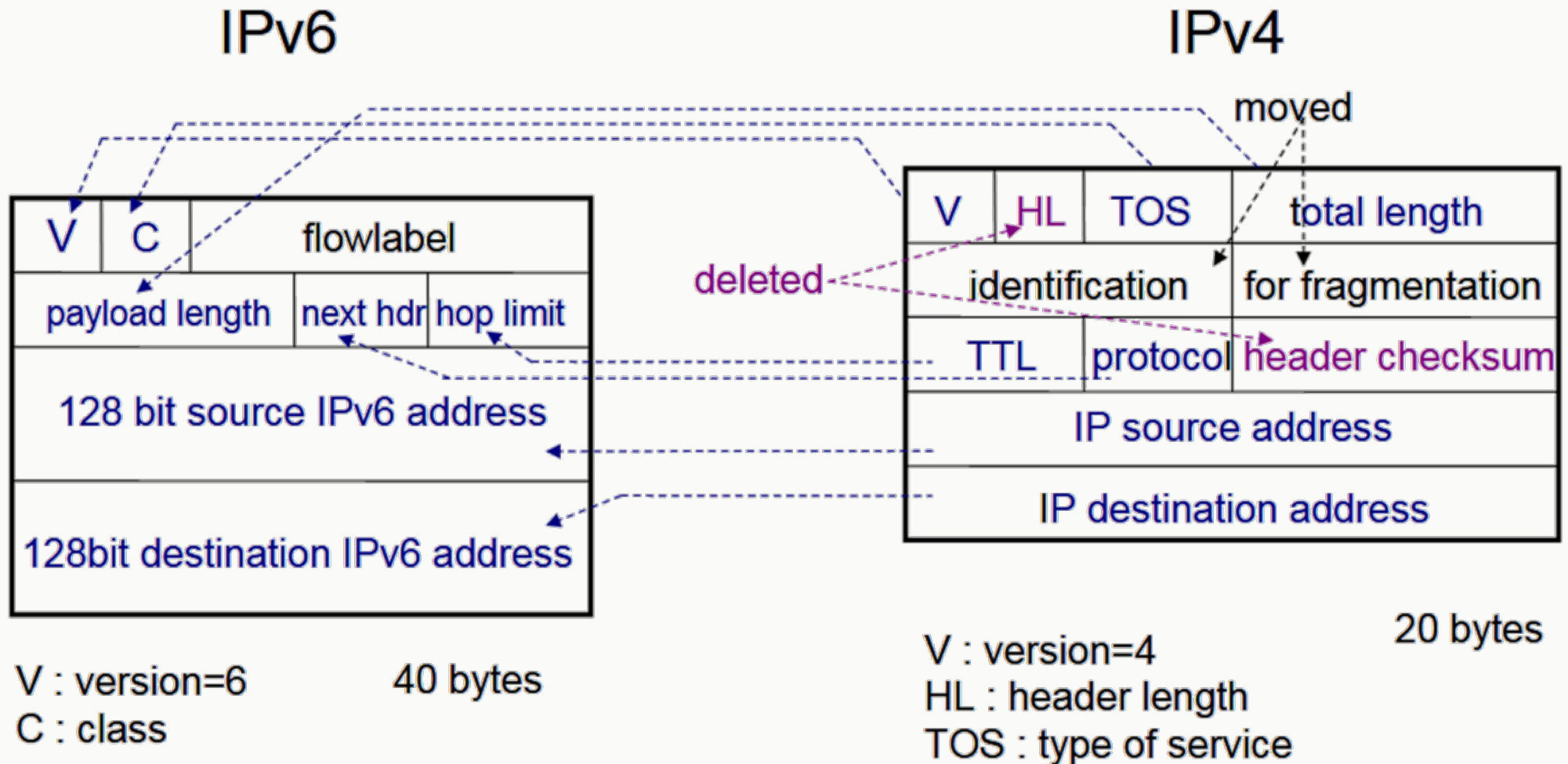
- 4in6-Tunnel
(Tunnel-Broker)

- 4to6-Tunnel
(2002::/16 Bereich)



- Dual Stack (Notwendige Koexistenz beider Protokoll für den Übergangszeitraum)

HEADER: IPV6 VS. IPV4



HEADER: IPV6 VS. IPV4 (FORTS.)

Keine Checksum?

- Nicht benötigt, da höhere Layer Prüfmechanismen haben
- Router müssen nicht bei jedem Pakete die Checksum (nochmal) berechnen
 - Geschwindigkeitsvorteile!
- IP-Adressen sind 64 Bit aligned (ausgerichtet an 64-Bit-Grenzen)
 - Mehr Geschwindigkeit beim Speicherzugriff auf 64-Bit-Rechnern

Payload length

- Die Länge der Daten nach dem 40 Byte Header in Byte
- Bei Jumbogrammen ($\geq 2^{16}$ Bytes) steht hier 0
 - Tatsächliche Länge im Erweiterungs-Header

HEADER: IPV6 VS. IPV4 (FORTS.)

Hop Limit

- In IPv4 “Time-To-Live”
- Wird bei jedem Hop (durch den Router) um Eins verringert
- Wenn der Wert anschließend 0 ist, soll der Router das Paket verwerfen und eine ICMP-Nachricht an den Absender senden

Next Header

- Entspricht dem “Protocol”-Feld in IPv4
- Zum Multiplexing übergeordneter Transport-Layer-Protokolle
- Manche Nummern entsprechen der IPv4-Nummer (6 = TCP, 17 = UDP)
- Neue Nummern, etwa 58 = ICMPv6

HEADER: IPV6 VS. IPV4 (FORTS.)

Flow Label

- 24-Bit-Markierung für Pakete (default 0) – gesetzt nur vom Absender
- Identifiziert einen Netzwerkstrom (etwa für QoS, Bandbreiten oder Latenzanforderungen)
- Bisher im Grunde nicht verwendet

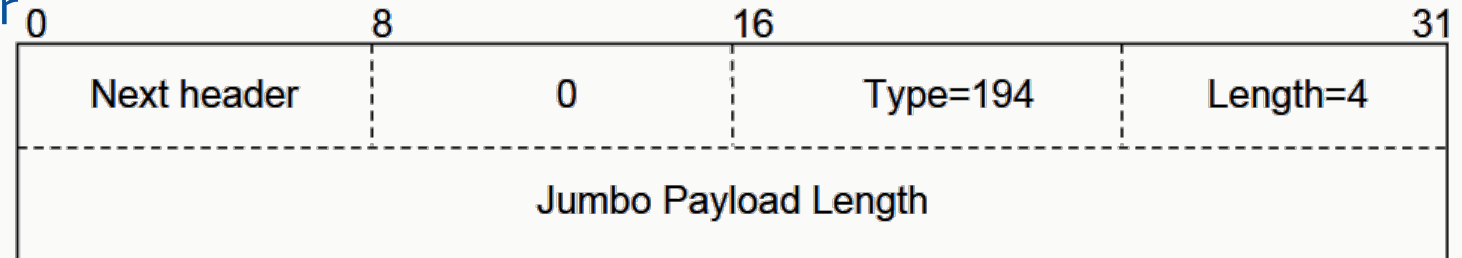
Entspricht "ToS" in IPv4 – gesetzt vom Absender

- Wert entspricht einer Art "Priorität" der Nutzdaten
- Bisher im Grunde nicht verwendet

HEADER: IPV6 VS. IPV4 (FORTS.)

Erweiterungs-Header (Hop-by-Hop ausgewertet)

- Folgen direkt auf den Standard-Header
 - Source-Routing (Routing Header 0)
 - Erlaubt die Angabe von Hops, die ein Paket auf dem Pfad zum Ziel verwenden soll
 - Nicht verwendet (zu hohes Missbrauchspotential)
 - Jumbogram-Header



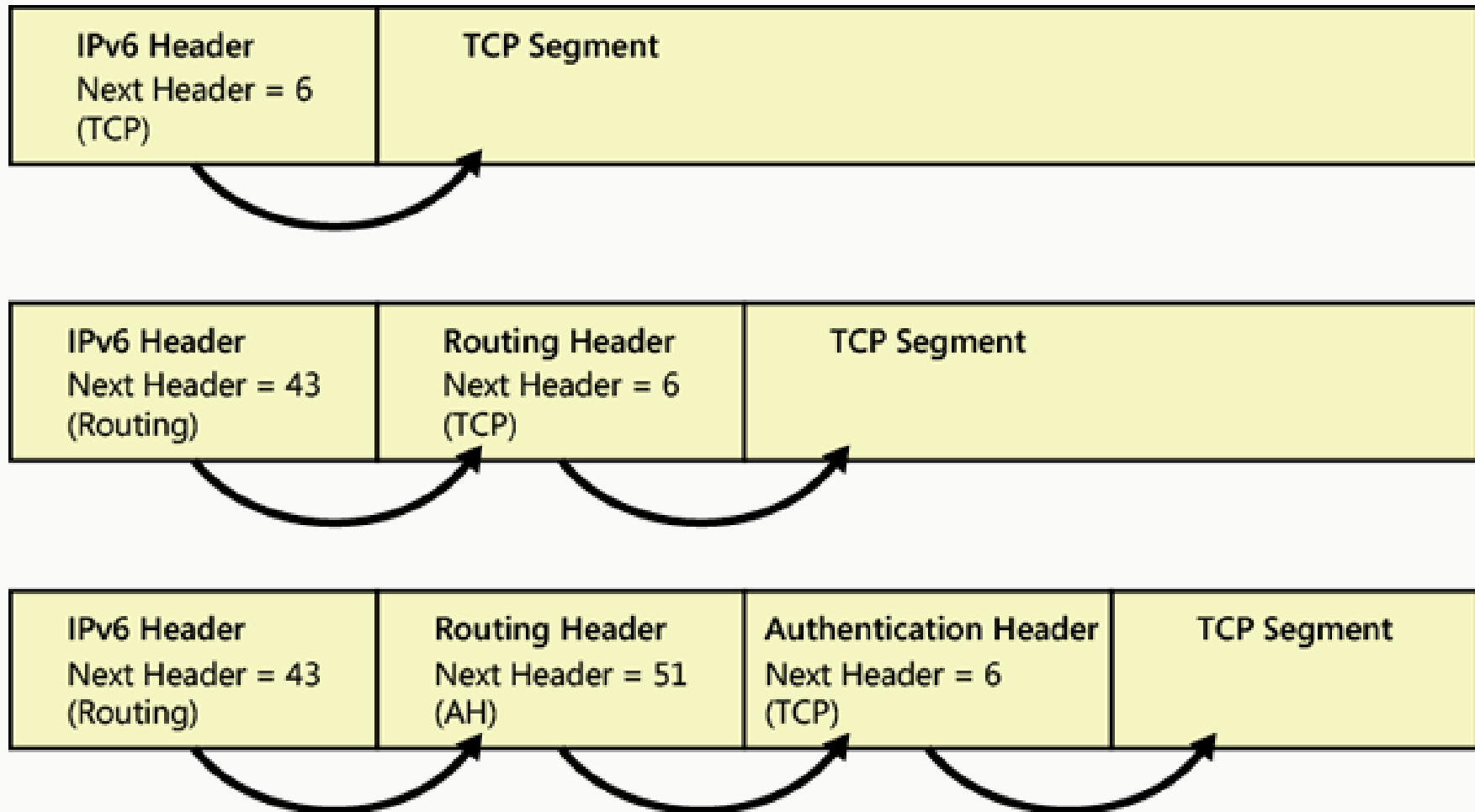
- MTU für Jumbogramme muss größer 2^{16} sein
- Keine Fragmentierung möglich
- Type (2 signifikanten Bit bestimmen, was passiert, wenn der Header nicht verarbeitet werden kann)

HEADER: IPV6 VS. IPV4 (FORTS.)

Erweiterungs-Header (Nur am Ziel ausgewertet)

- Fragmentation Header
- Authentication Header (IPSec AH)
- Encapsulation Security Payload Header (IPSec ESP)
- Destination Options
- Header übergeordneter Layer (TCP, UDP, ICMPv6, etec.)

HEADER: IPV6 VS. IPV4 (FORTS.)



128 Bit Adressen

IPv4: Dotted Dezimal Notation (127.0.0.1)

IPv6: Hexadezimale Notation

```
2001 : 0db8 : 0000 : 0000 : 0000 : 0000 : 0000 : 0001
```

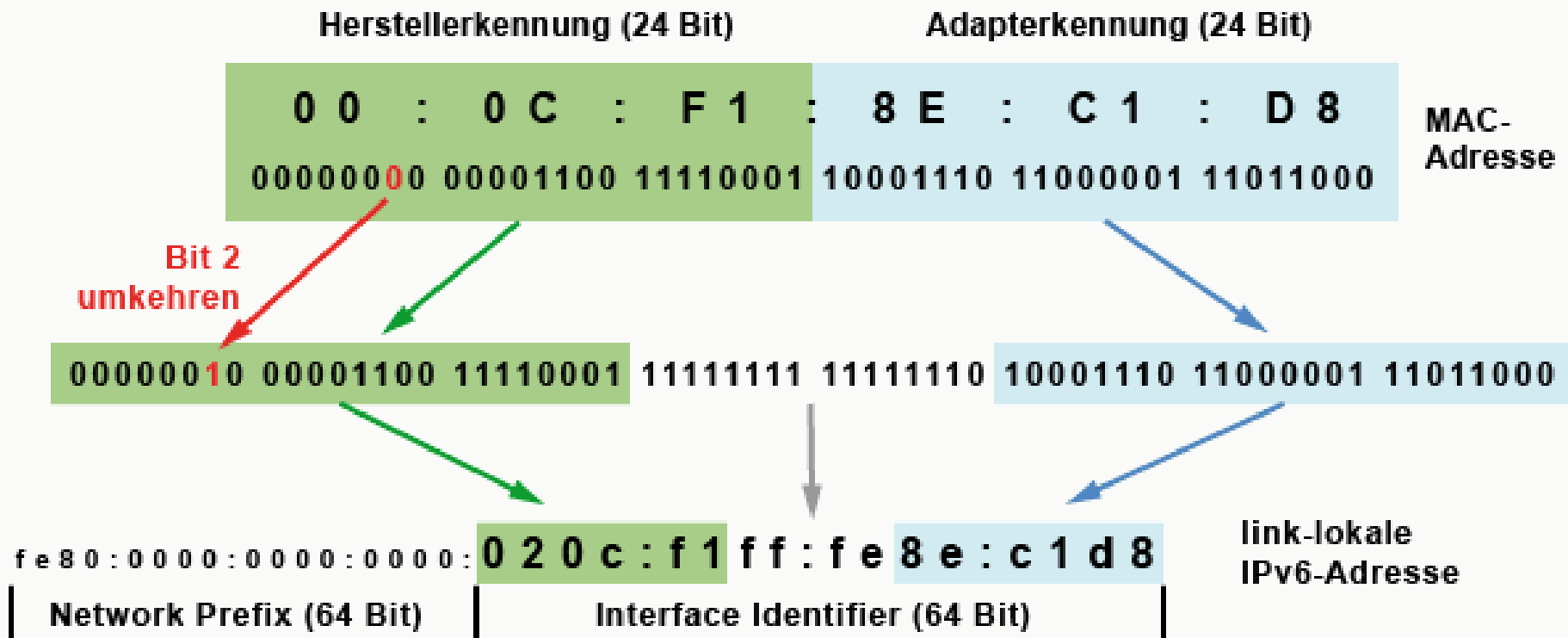
Network Prefix (64 Bit)

Interface Identifier (64 Bit)

- Notationsregeln (z.B. Abkürzungen / Zusammenfassungen) in RFC 5952
 - Einzig gültige Schreibweise der o.g. Adresse: 2001:db8::1
- Verwendung in URLs mit []
 - `http://[2001:db8::1%25eth0]:80/` (Interface-Wahl mit % - urlencoded %25)
- Verwendung in UNC's:
 - `\\2001:db8::1\share` (normale Schreibweise)
 - `\\2001-db8--1.ipv6-literal.net\share` (angepasste Schreibweise)

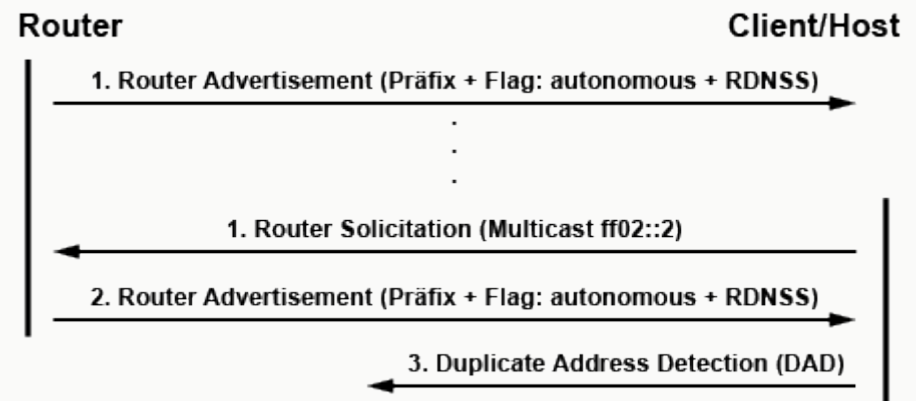
ICMPv6 / Neighbor Discovery Protocol (NDP)

- 1) Wahl der eigenen link-local IP-Adresse
 - 1) Stateless Address Autoconfiguration (SLAAC)



ICMPv6 / Neighbor Discovery Protocol (NDP)

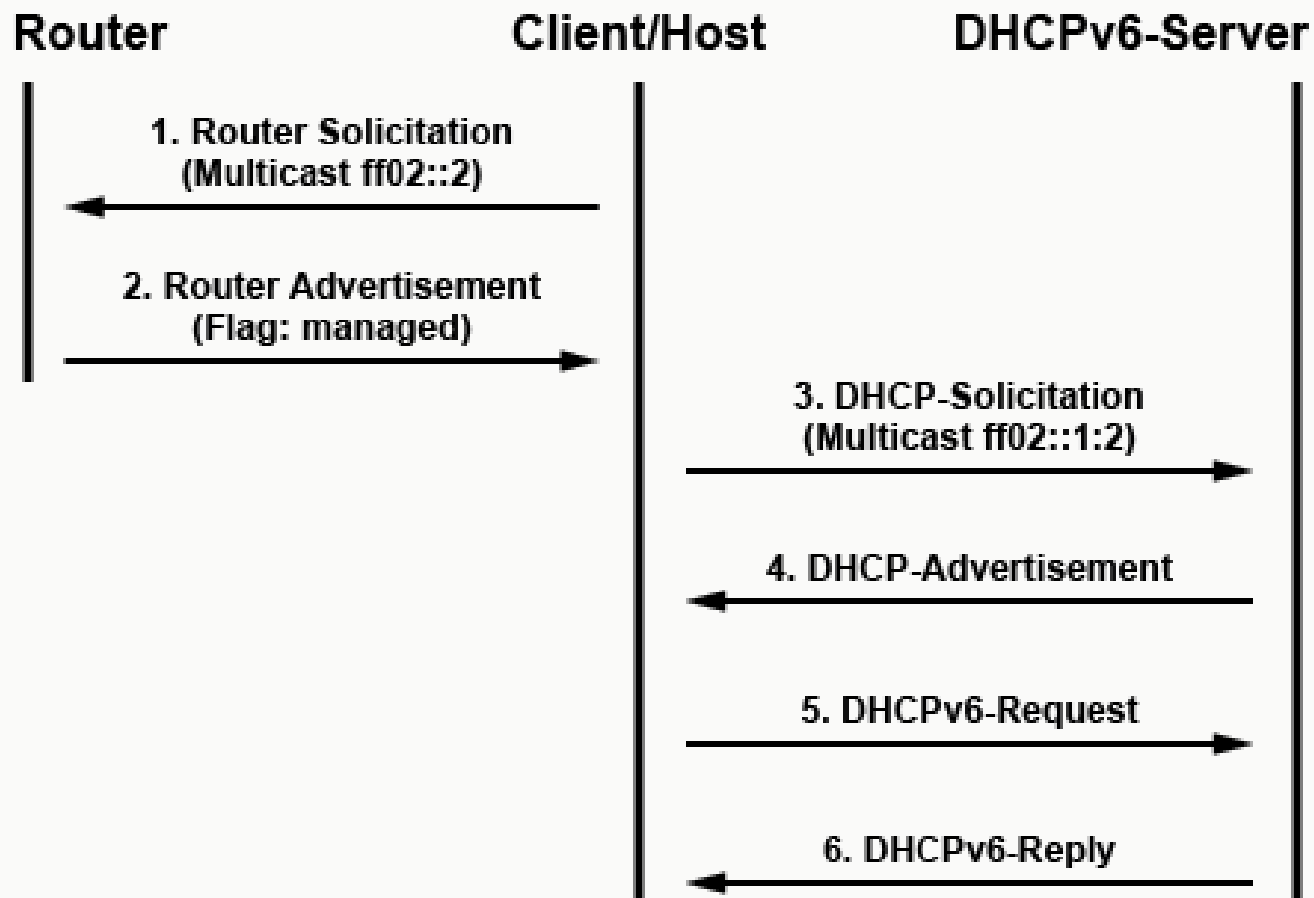
- 1) Wahl der eigenen link-local IP-Adresse
 - 1) Stateless Address Autoconfiguration (SLAAC)
 - 2) Prüfen der gewählten IP-Adresse "Duplicate Address Detection (DAD)"
 - 1) Neighbor Solicitation (NS)
 - 2) Neighbor Advertisement (NA)
- 2) Wahl der eigenen globalen IP-Adressen (Prefix des globalen Adressblocks)
 - 1) Router Solicitation (RS)
 - 2) Router Advertisement (RA)
 - 3) Prüfen der IP-Adresse (DAD)



Ein Rechner ist durch den aus der MAC erzeugten Interface Identifier in unterschiedlichen Prefixen eindeutig erkennbar

- Privacy Extensions (RFC 4941) erzeugen weitere Adresse mit zufälligem Interface Identifier
 - Nehme aktuellen NTP-Zeitstempel (64 Bit) und die MAC-Adresse
 - Erstelle SHA1-Hashs mit einer Länge von 64 Bit
 - Ergebnis ist der neue zufällige Interface Identifier
- Regelmäßiger Wechsel des Interface Identifiers (stündlich / täglich / manuell)
 - Temporär Weiterbetrieb des vorigen Identifiers
 - Neue Verbindungen mit neuem Identifier
- Nur für globale IP-Adressen, nicht für link-local IP-Adressen

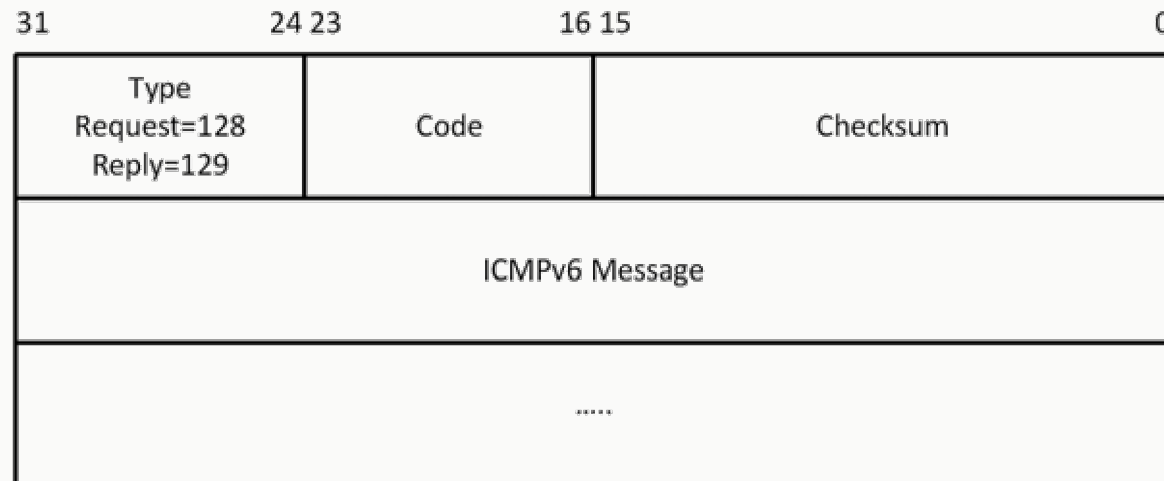
DHCPv6 (Stateful Address Configuration)



DHCPv6 (Stateful Address Configuration)

- Eigentlich nicht benötigt, aber nicht alle Clients / Optionen funktionieren mit SLAAC
 - Beispiel: RDNSS (Rekursiver DNS-Server zur Namensauflösung)
 - Erst 2010 als Standard in SLAAC (nicht von allen implementiert)
 - Feste IP-Adresse eines Servers unabhängig von der Netzwerkkarte
- Streit um SLAAC / DHCPv6
 - Google Android => SLAAC / kein DHCPv6
 - Microsoft => DHCPv6 / kein SLAAC
 - Aufgrund unterschiedlicher Implementierungen in den meisten Netzwerken sowohl SLAAC als auch DHCPv6 mit allen möglichen Optionen konfiguriert

Übertragung von Statusinformationen und Fehlermeldungen (analog zu ICMP)



(Note: IPv6 header is prepended)

Zusätzlich: Versand von NDP-Nachrichten (Solicitation / Advertisement)

- Neighbor Discovery (IPv4: ARP); Types: 135, 136
- Inverse Neighbor Discovery (IPv4: RARP); Types: 141, 142
- Multicast Listener Discovery (IPv4: IGMP); Type: 130

Übertragung von Statusinformationen und Fehlermeldungen

NOTWENDIG FÜR DEN BETRIEB VON IPV6

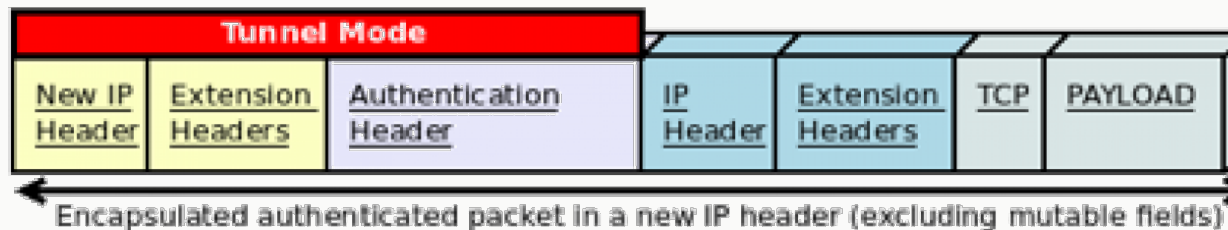
Daher: Blockieren im Paketfilter (wie bei IPv4) nicht empfohlen!

Idee: Jede Verbindung mit IPv6 wird mittels IPSec abgesichert

- Realität: Nicht!
- Verwendung von IPSec “nativ” als IPv6-Extension-Header
 - Next Header: 50 (Encapsulating Security Header)
 - Verschlüsselung des Datenpakets
 - Authentifikation des Kommunikationspartners (auch ohne Verschlüsselung möglich)
 - Next Header: 51 (Authentication Header)
 - Authentifikation des Kommunikationsinhalts
 - Integritätscheck des Datenpakets / Verhindert Replay-Angriffe
- Unterschiedliche Modi:
 - Transport (Host to Host IPv6+IPSec => Originales IP-Paket mit AH/ESP-Header)
 - Tunnel (Gateway to Gateway/Host IPv6+IPSec => Umverpacktes IPv6-Paket)

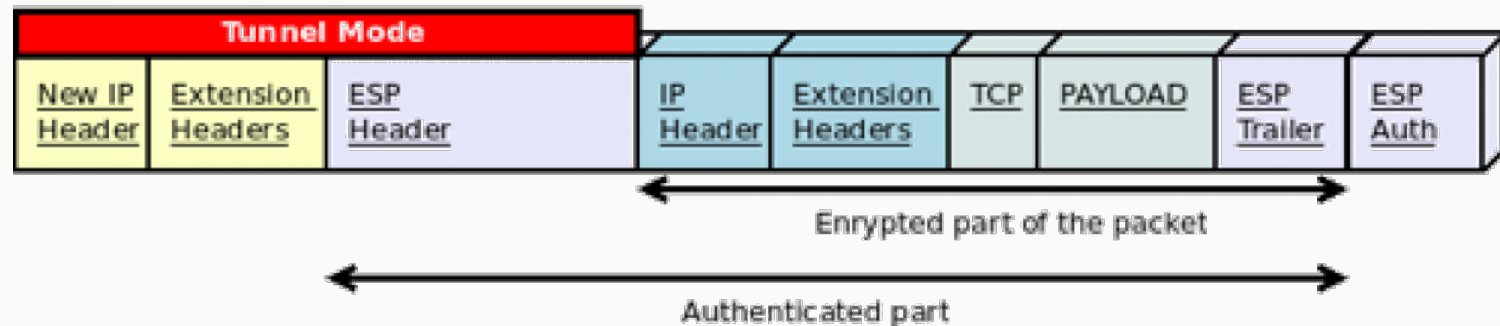
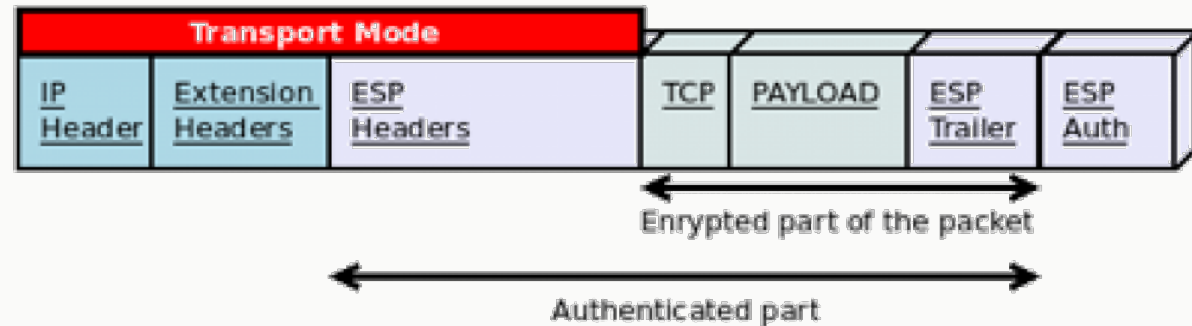
IPSEC TRANSPORT VS. TUNNEL

Authentication Header



IPSEC TRANSPORT VS. TUNNEL

Encapsulating Security Payload



Fragmentierung bei IPv6 nur noch durch den Absender möglich

- IPv4: Fragmentierung auch durch Router auf dem Weg
 - Erzeugt eine hohe Last in den Routern
- IPv6: ICMP-Nachricht (Typ 2: Packet Too Big) an den Absender
 - Reaktion:
 - Anwendung erzeugt kleinere Pakete, die keine Fragmentierung benötigen
 - IPv6 fragmentiert mit Fragment Extension Header
 - Next Header: 44
 - Ursprünglicher Next Header wird in Fragmentation-Header verschoben
- Fragmentierung wird im Internet wenig verwendet (ca. 8% des Internettraffics)

Multicast erlaubt den Versand von Paketen an unbekannte Empfänger

- Multicast-Adressen bilden eigene Scopes (zugewiesen durch IANA)
- Zuordnung von Hosts und Diensten mittels Multicast-Gruppen
 - Jede Gruppe hat eine eigene Adresse (aus dem Prefix ff::/8)
 - 4 Bits für Flags und 4 Bits für den Gültigkeitsbereich (Scope)
 - Scope ff01 gilt nur am lokalen Interface (Host-intern)
 - Scope ff02 gilt nur link-local (LAN)
- Häufig genutzte Gruppen:
 - ff0X::1 : alle IPv6-Stationen (es gibt kein Broadcast mehr in IPv6)
 - ff0X::2 : alle Router
 - ff0X::f : UPnP
 - ff0X::101 : alle Zeitserver (NTP)
 - ff0X::1:2 : DHCPv6-Server

Beispiel:

- Ping an link-local Hosts

\$ > ping6 ff02::1

```
[matze@tschita] ~ $ ping6 -c 5 ff02::1
PING ff02::1(ff02::1) 56 data bytes
64 bytes from fe80::dafc:93ff:fe55:1ae0%wlan0: icmp_seq=1 ttl=64 time=0.062 ms
64 bytes from fe80::3a10:d5ff:fe3b:ee3c%wlan0: icmp_seq=1 ttl=64 time=2.86 ms (DUP!)
64 bytes from fe80::ba27:ebff:fe2c:d8a3%wlan0: icmp_seq=1 ttl=255 time=7.45 ms (DUP!)
64 bytes from fe80::32d3:2dff:fe74:ad66%wlan0: icmp_seq=1 ttl=255 time=133 ms (DUP!)
64 bytes from fe80::32d3:2dff:feb5:11a2%wlan0: icmp_seq=1 ttl=255 time=133 ms (DUP!)
64 bytes from fe80::6ead:f8ff:fe72:f14%wlan0: icmp_seq=1 ttl=255 time=133 ms (DUP!)
64 bytes from fe80::1002:e75f:a210:8eb4%wlan0: icmp_seq=1 ttl=255 time=134 ms (DUP!)
64 bytes from fe80::faad:cbff:fe19:1a22%wlan0: icmp_seq=1 ttl=255 time=135 ms (DUP!)
64 bytes from fe80::facf:c5ff:fe2e:2e8%wlan0: icmp_seq=1 ttl=255 time=228 ms (DUP!)
64 bytes from fe80::dafc:93ff:fe55:1ae0%wlan0: icmp_seq=2 ttl=64 time=0.049 ms
64 bytes from fe80::3a10:d5ff:fe3b:ee3c%wlan0: icmp_seq=2 ttl=64 time=2.77 ms (DUP!)
64 bytes from fe80::32d3:2dff:feb5:11a2%wlan0: icmp_seq=2 ttl=255 time=6.26 ms (DUP!)
64 bytes from fe80::32d3:2dff:fe74:ad66%wlan0: icmp_seq=2 ttl=255 time=8.20 ms (DUP!)
64 bytes from fe80::ba27:ebff:fe2c:d8a3%wlan0: icmp_seq=2 ttl=255 time=8.20 ms (DUP!)
64 bytes from fe80::6ead:f8ff:fe72:f14%wlan0: icmp_seq=2 ttl=255 time=47.5 ms (DUP!)
64 bytes from fe80::feal:83ff:fe63:7619%wlan0: icmp_seq=1 ttl=255 time=1049 ms (DUP!)
64 bytes from fe80::feal:83ff:fe63:7619%wlan0: icmp_seq=2 ttl=255 time=49.1 ms (DUP!)
64 bytes from fe80::faad:cbff:fe19:1a22%wlan0: icmp_seq=2 ttl=255 time=90.6 ms (DUP!)
64 bytes from fe80::1002:e75f:a210:8eb4%wlan0: icmp_seq=2 ttl=255 time=107 ms (DUP!)
64 bytes from fe80::facf:c5ff:fe2e:2e8%wlan0: icmp_seq=2 ttl=255 time=158 ms (DUP!)
64 bytes from fe80::dafc:93ff:fe55:1ae0%wlan0: icmp_seq=3 ttl=64 time=0.030 ms
64 bytes from fe80::3a10:d5ff:fe3b:ee3c%wlan0: icmp_seq=3 ttl=64 time=3.17 ms (DUP!)
64 bytes from fe80::32d3:2dff:feb5:11a2%wlan0: icmp_seq=3 ttl=255 time=7.15 ms (DUP!)
64 bytes from fe80::32d3:2dff:fe74:ad66%wlan0: icmp_seq=3 ttl=255 time=8.82 ms (DUP!)
64 bytes from fe80::ba27:ebff:fe2c:d8a3%wlan0: icmp_seq=3 ttl=255 time=8.82 ms (DUP!)
64 bytes from fe80::1002:e75f:a210:8eb4%wlan0: icmp_seq=3 ttl=255 time=28.4 ms (DUP!)
64 bytes from fe80::6ead:f8ff:fe72:f14%wlan0: icmp_seq=3 ttl=255 time=68.2 ms (DUP!)
64 bytes from fe80::feal:83ff:fe63:7619%wlan0: icmp_seq=3 ttl=255 time=68.8 ms (DUP!)
64 bytes from fe80::faad:cbff:fe19:1a22%wlan0: icmp_seq=3 ttl=255 time=110 ms (DUP!)
64 bytes from fe80::dafc:93ff:fe55:1ae0%wlan0: icmp_seq=4 ttl=64 time=0.049 ms
64 bytes from fe80::3a10:d5ff:fe3b:ee3c%wlan0: icmp_seq=4 ttl=64 time=3.11 ms (DUP!)
64 bytes from fe80::32d3:2dff:feb5:11a2%wlan0: icmp_seq=4 ttl=255 time=28.5 ms (DUP!)
64 bytes from fe80::32d3:2dff:fe74:ad66%wlan0: icmp_seq=4 ttl=255 time=28.5 ms (DUP!)
64 bytes from fe80::ba27:ebff:fe2c:d8a3%wlan0: icmp_seq=4 ttl=255 time=28.5 ms (DUP!)
64 bytes from fe80::1002:e75f:a210:8eb4%wlan0: icmp_seq=4 ttl=255 time=50.6 ms (DUP!)
64 bytes from fe80::feal:83ff:fe63:7619%wlan0: icmp_seq=4 ttl=255 time=90.7 ms (DUP!)
64 bytes from fe80::6ead:f8ff:fe72:f14%wlan0: icmp_seq=4 ttl=255 time=90.7 ms (DUP!)
64 bytes from fe80::faad:cbff:fe19:1a22%wlan0: icmp_seq=4 ttl=255 time=128 ms (DUP!)
^C
--- ff02::1 ping statistics ---
4 packets transmitted, 4 received, +34 duplicates, 0% packet loss, time 3002ms
rtt min/avg/max/mdev = 0.030/83.791/1048.974/168.931 ms, pipe 2
```

Gateway-Router erhalten nicht mehr nur eine IP(v4), sondern einen IP(v6)-Prefix vom Einwahlserver

- Es gibt endlich genug Adressen
 - /32 Netze an ASe / ISPs
 - /48- oder /56-Subnetze an Endkunden
 - Für SLAAC ist mind. ein /64-Subnetz notwendig
- Daher: kein NAT mehr!
 - Direkte Erreichbarkeit aller Endpunkte aus dem Internet (theoretisch)
 - Abhängig vom eingesetzten Router (Firewall ersetzt NAT)
 - Telekom-Router (Speedport) blockt eingehenden IPv6-Verkehr
 - Fritzbox erlaubt "Exposed Hosts"
 - Wie ist es bei anderen Geräten?

OFFENSICHTLICHE UNTERSCHIEDE IPV4 – IPV6

Feature	IPv4	IPv6
Adressierung	32 Bits	128 Bits
Paketstruktur	Header mit variabler Größe	Feste Größe des Headers + Erweiterungsheader
Adressauflösung	ARP	ICMPv6 NS/NA (+MLD)
Auto-Konfiguration	DHCP	ICMPv6 RS/RA & DHCPv6 (+MLD)
Fehlerisolation	ICMPv4	ICMPv6
IPSec-Unterstützung	Optional	Optional
Fragmentierung	In Hosts und Routern	Nur in Hosts
Multicast	Nur Multicast-Anwendungen	Nötig für Neighbor-Discovery
Netzwerkarchitektur	Private Adressen + NAT	Globale Adressen + Firewall

Vielen Dank für die Aufmerksamkeit!

Fragen?

Nächste Vorlesung:

- Montag, 30. Mai 2022

Nächste Übung:

- Dienstag, 24. Mai 2021 – 16 Uhr
- Abgabe des Übungszettels 6 bis morgen – 16 Uhr