HSLU Hochschule Luzern

Einführendes Labor zu Informations- und Cyber-Sicherheit

Eindenken in die Welt der Cyber Security mit praktischen Beispielen

Informatik
27. Oktober 2025

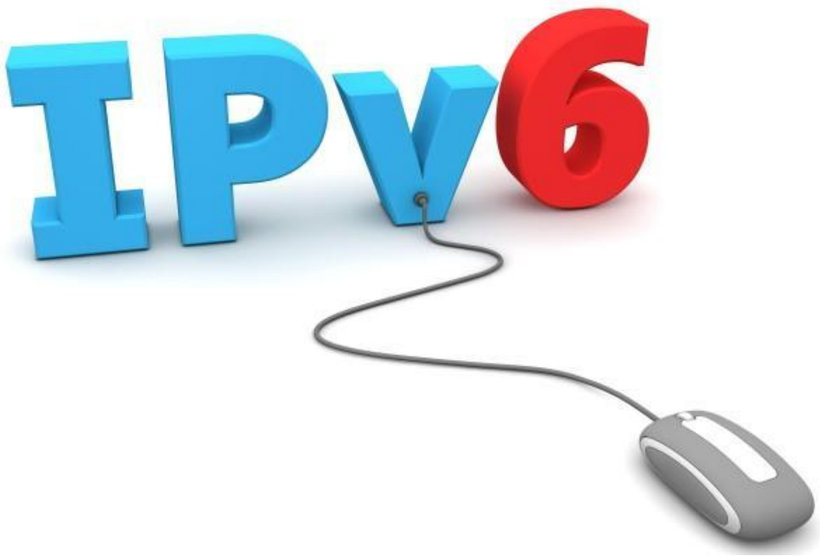
FH Zentralschweiz

Mikrofon- und Verständnis-Test

HSLU Hochschule Luzern

Einführendes Labor Sicherheit

IPv6 - das Netz der Zukunft



Informatik
27. Oktober 2025

FH Zentralschweiz

HSLU Hochschule Luzern

Organisatorisches

Organisatorisches

Informatik
27. Oktober 2025

FH Zentralschweiz

AACSB ACCREDITED

Laborabgaben

- **Labor „Protokollanalyse“** und **„Man-in-the-Middle“**
  - Probleme?
  - Wer es nicht bis zum nächsten Labor schafft, → Mail an mich schreiben zur Verlängerung
- **Musterlösung**
  - Nicht vergessen: Musterlösung anschauen
  - Musterlösung nach Abgabe im ILIAS verfügbar.

3 ARP Request Poisoning: Senden von gefälschten ARP-Paketen

Im Folgenden werden Sie ein `ARP Request Poisoning` vornehmen, um als MITM zwischen dem `Ubuntu Client` und dem `lokalen Gateway` zu interagieren. Als Angreifer benutzen Sie hierfür den Client `Kali Linux`. Um als MITM agieren zu können, müssen Sie sich zuerst eine Übersicht verschaffen, welche Hosts im Netzwerk aktiv sind. Hierzu verwenden wir an dieser Stelle das Tool `Nmap`. «Nmap ist ein Werkzeug zum Scannen und Auswerten von Hosts in einem Computernetzwerk und fällt somit in die Kategorie der Portscanner»[1].

[1] Wikipedia: https://de.wikipedia.org/wiki/Nmap

1. Wechseln Sie zum Kali Linux-Client und öffnen Sie ein neues Terminal.
2. Machen Sie sich kurz über das Tool `Nmap` schlau:

```
> nmap -h
```

Investieren Sie hier nicht zu viel Zeit. Sie werden sich mit `Nmap` in einem anderen Labormodul genauer auseinandersetzen.

3. Sie führen an dieser Stelle einen SYN Stealth-Scan[1] im angegebenen Netzwerk aus:

```
> nmap -sS 192.168.1.0/24
```

Zum Ausführen von `nmap` brauchen Sie `root`-Rechte: `sudo -i`.

[1] SYN Stealth-Scan, https://nmap.org/book/synscan.html

4. Wie viele Hosts wurden identifiziert?

4 Hosts (Firewall, Windows und 2 VMs)

SW1 – 17.02.	**SW2** – 24.02.	**SW3** – 03.03.	**SW4** – 10.03.	**SW5** – 17.03.	**SW6** – 24.03.	**SW7** – 31.03.
Einführung und Organisatorisches • ILIAS und Wochenplan • Aufbau der Vorlesung • Vorstellung der Modulinhalte • Besuch des Labors	Open Your Mind • SolarWinds-Hack • Def. Lateral Thinking • Zusammenhang zwischen Lateral Thinking ↔ Cyber Security • Kreativitätstechniken und Fallbeispiele Theorie	Protokollanalyse • Protokolle & Schichten im Netzwerk • Ausgewählte Protokolle und Funktionen • Standards • Analyse mit Wireshark Theorie	Protokollanalyse-Lab • ARP, IP, ICMP, TCP & UDP, DNS & DHCP, HTTP & HTTPS und FTP-Protokoll • Analyse-Werkzeuge • Praktischer Umgang und Analyse Labor	Man in the Middle (MitM) Lab • Unterschiedliche Arten von Man in the Middle (MitM)-Attacken • Methode: ARP Request Poisoning • Methode: DNS Spoofing Labor	Strategie und Taktik • Theorie zu Strategie & Taktik in der Cyber Security • Strategie und Taktik übertragen in die IT Security Theorie	Internet Protocol Version 6 (IPv6) • Motivation: Warum? • IPv6-Adressierung • IPv6-Paket-Header • Spezial-IPv6-Adressen • Stateless Address Autoconfiguration • Andere Verfahren zur Adressvergabe Theorie
Zimmer: I.S1A_230/1 Montag, 15:30 – 17:50 Uhr	Zimmer: I.S1A_230/1 Montag, 15:30 – 17:50 Uhr	Zimmer: I.S1A_230/1 Montag, 15:30 – 17:50 Uhr	Labor: I.S1A_403/404 Montag, 15:30 – 17:50 Uhr *(siehe Einteilung)*	Labor: I.S1A_403/404 Montag, 15:30 – 17:50 Uhr *(siehe Einteilung)*	Zimmer: I.S1A_230/1 Montag, 15:30 – 17:50 Uhr	Zimmer: I.S1A_230/1 Montag, 15:30 – 17:50 Uhr
			Labor: I.S1A_404 18:30 – 20:50 Uhr *(siehe Einteilung)*	Labor: I.S1A_404 18:30 – 20:50 Uhr *(siehe Einteilung)*		

Heute

SW8 – 07.04.	**SW9** – 14.04.	**SW10** – 28.04.	**SW11** – 05.05.	**SW12** – 12.05.	**SW13** – 19.05.	**SW14** – 26.05.
IPv6 Lab • Analyse IPv6 • Reihenfolge der Nachrichten / Ablauf • Praktische Konfiguration • Tunnel- und IPv4/IPv6-Deployment-Szenarien Labor	Password Cracking • Theorie der sicheren Passwörter • Salting • Windows VM-Passwort hacken Labor	Firewall-Technologien • Definition Firewall • Vom Packetfilter zu Application Layer Gateways (ALGs) • Web Application Firewall (WAF) • Firewall-Architekturen Theorie	Firewall Basics Lab • Konfiguration und Funktionen von Firewalls • Schrittweises Kennenlernen der Funktionsweise & des technischen Ablaufs Labor	Firewall Advanced Lab • Application Layer Gateways (ALGs) • Next Generation-Firwall (NextGen) • Einsatzgebiet und Funktionen Labor	Web Authentication Lab • Grundlagen und Techniken zur Authentifizierung • Cookie-basierte Authentifizierung • OAuth 2.0 • WebAuthn Labor	Physische Sicherheit • Von Schloss und Schlüsseln • Pick a lock: Lernen von Lockpicking-Techniken Theorie (Gastreferent)
Labor: I.S1A_403/404 Montag, 15:30 – 17:50 Uhr *(siehe Einteilung)*	Labor: I.S1A_403/404 Montag, 15:30 – 17:50 Uhr *(siehe Einteilung)*	Zimmer: I.S1A_230/1 Montag, 15:30 – 17:50 Uhr	Labor: I.S1A_403/404 Montag, 15:30 – 17:50 Uhr *(siehe Einteilung)*	Labor: I.S1A_403/404 Montag, 15:30 – 17:50 Uhr *(siehe Einteilung)*	Labor: I.S1A_403/404 Montag, 15:30 – 17:50 Uhr *(siehe Einteilung)*	Zimmer: I.S1A_230/1 Montag, 15:30 – 17:50 Uhr
Labor: I.S1A_404 18:30 – 20:50 Uhr *(siehe Einteilung)*	Labor: I.S1A_404 18:30 – 20:50 Uhr *(siehe Einteilung)*		Labor: I.S1A_404 18:30 – 20:50 Uhr *(siehe Einteilung)*	Labor: I.S1A_404 18:30 – 20:50 Uhr *(siehe Einteilung)*	Labor: I.S1A_404 18:30 – 20:50 Uhr *(siehe Einteilung)*	

Labor IPv6 vor Ort

Zoom-Link für Theorie-Sessions (Achtung: nur Streaming, kein Hybrid-Unterricht!):

HSLU Hochschule Luzern ILIAS HSLU

Dashboard

Meine HSLU

Magazin > Informatik > Ausbildung > Unterlagen zum Unterricht > 04. IPv6

04. IPv6

Aktionen

Ebene hoch | Inhalt | Info | Einstellungen | Lernfortschritt | Export | Rechte

Labor nächsten Montag

Vorlesung

1. Vorlesungsunterlagen

Tests

2. Vorbereitungsfragen zu IPv6

Bitte beantworten Sie diese Fragen zu Hause als Vorbereitung auf die Laborübung. Sie müssen 90% der Antworten richtig beantworten, um die Prüfung zu bestehen. Bitte recherchieren und informieren Sie sich, um die Fragen zu beantworten und sich gut auf die Übung vorzubereiten. Geschätzter Zeitaufwand: ca. 30 - 60 min.

Lernfortschritt:

Übungen

3. Laborübung - Protokollanalyse IPv6

Lernfortschritt:

Vorbedingungen, die für einen Zugriff erfüllt werden müssen: » Anzeigen

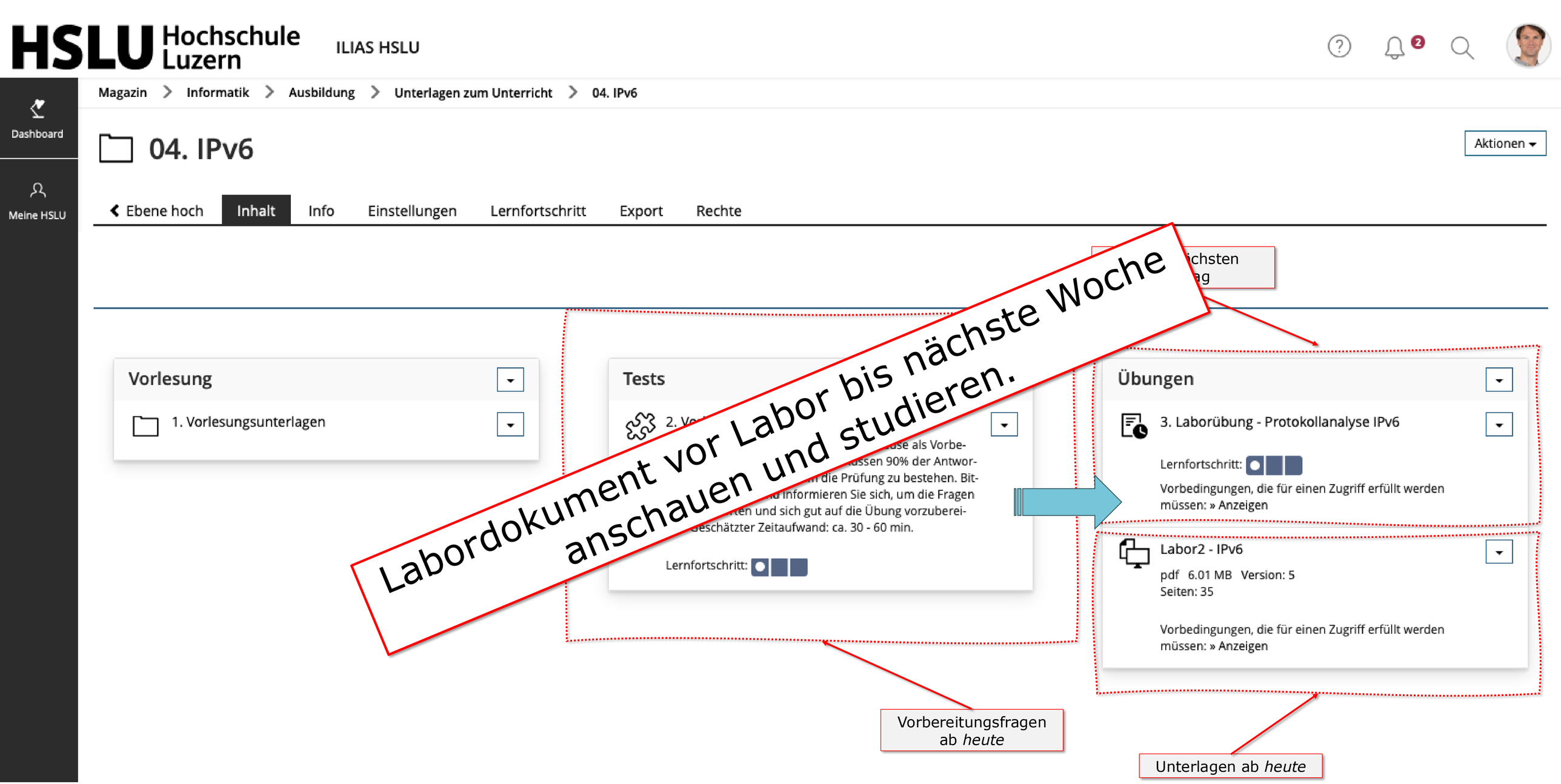
Labor2 - IPv6

pdf 6.01 MB Version: 5
Seiten: 35

Vorbedingungen, die für einen Zugriff erfüllt werden müssen: » Anzeigen

Vorbereitungsfragen ab *heute*

Unterlagen ab *heute*


HSLU Hochschule Luzern
ILIAS HSLU
Dashboard
Meine HSLU
Magazin > Informatik > Ausbildung > Unterlagen zum Unterricht > 04. IPv6
04. IPv6
Aktionen
Ebene hoch
Inhalt
Info
Einstellungen
Lernfortschritt
Export
Rechte
Vorlesung
1. Vorlesungsunterlagen
Tests
Lernfortschritt:
Übungen
3. Laborübung - Protokollanalyse IPv6
Lernfortschritt:
Vorbedingungen, die für einen Zugriff erfüllt werden müssen: » Anzeigen
Labor2 - IPv6
pdf 6.01 MB Version: 5
Seiten: 35
Vorbedingungen, die für einen Zugriff erfüllt werden müssen: » Anzeigen
Labordokument vor Labor bis nächste Woche anschauen und studieren.
Vorbereitungsfragen ab heute
Unterlagen ab heute

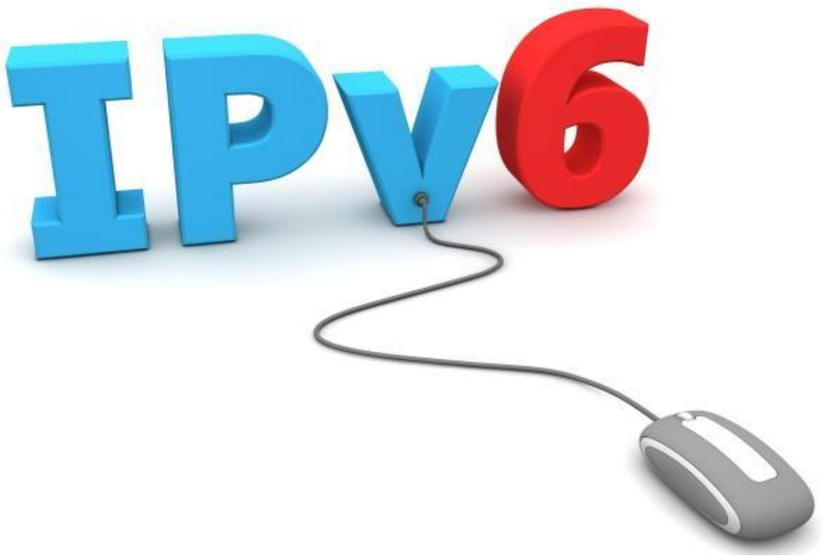
HSLU Hochschule Luzern

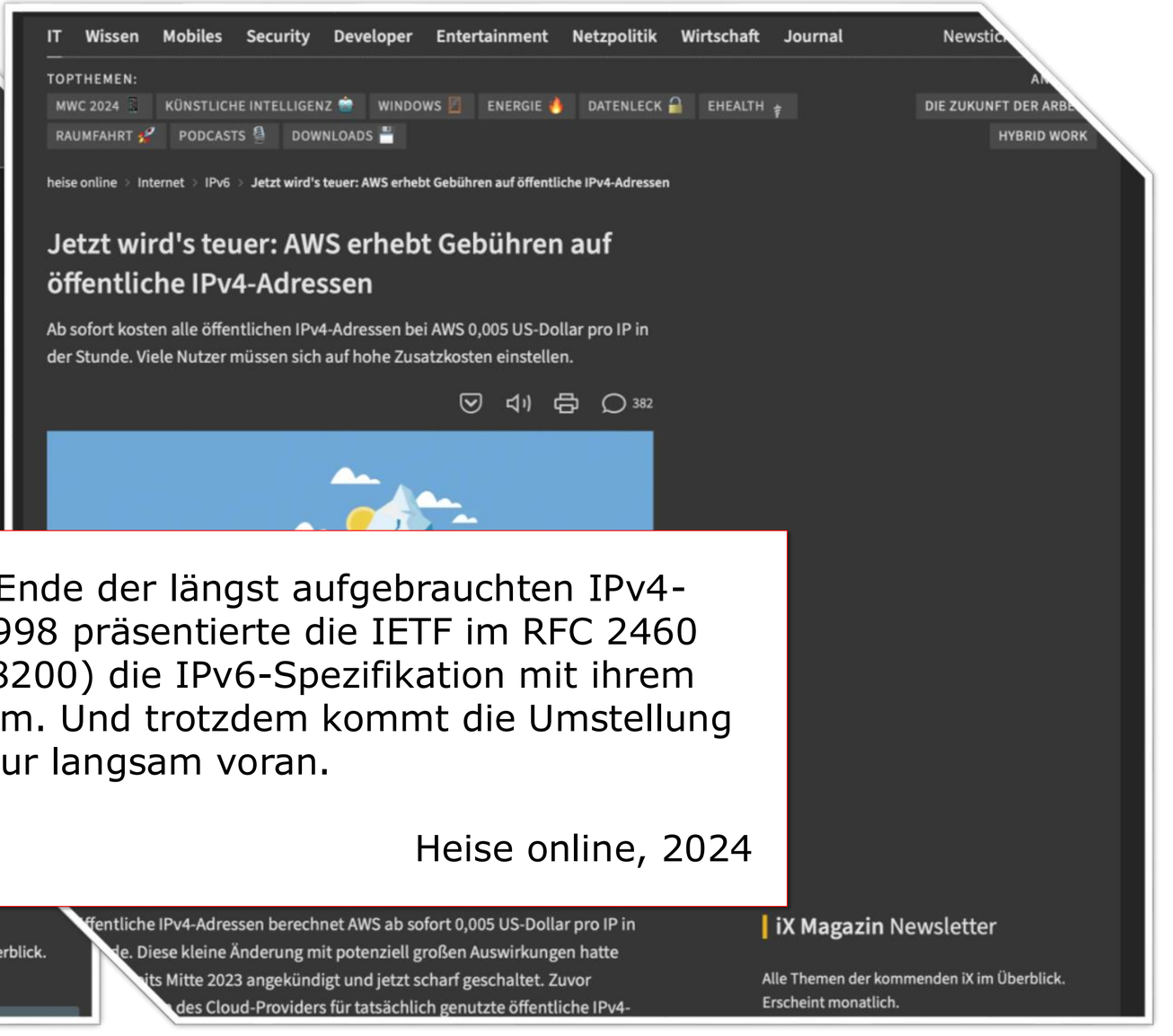
Generelle Fragen & Feedback?

Informatik
27. Oktober 2025

Intro Lab

IPv6 - das Netz der Zukunft





Seit 15 Jahren wird über das Ende der längst aufgebrauchten IPv4-Adressen debattiert. Bereits 1998 präsentierte die IETF im RFC 2460 (2017 ersetzt durch den RFC 8200) die IPv6-Spezifikation mit ihrem nahezu unbegrenzten Adressraum. Und trotzdem kommt die Umstellung auf IPv6 nur langsam voran.

Heise online, 2024

Motivation

IPv6 bietet folgende Verbesserungen gegenüber IPv4

1. Zustandslose Adressen-Autokonfiguration für einfachere Netzwerkverwaltung
2. Erweiterung des Adressraums von 32 auf 128 Bit und Eliminierung von NAT
3. Verbesserte Header-Struktur mit weniger Verarbeitungsaufwand
4. Effizienteres Routing ohne Fragmentierung von Paketen
5. Eingebaute Sicherheit auf Netzwerkschicht (IPsec)
6. Eingebaute Quality of Service (QoS), die verzögerungsempfindliche Pakete unterscheidet

IPv6 das zukünftige Internet-Protokoll

Einführung

- Gegenwärtige Version von IP (Version 4) ist 42 Jahre alt
- Es wurden neue Bedürfnisse geweckt, welche durch neue Technologien gedeckt werden müssen
- IETF hat neue IP-Version vorgeschlagen, um gegenwärtige Probleme zu lösen

Erfolg von IPv4

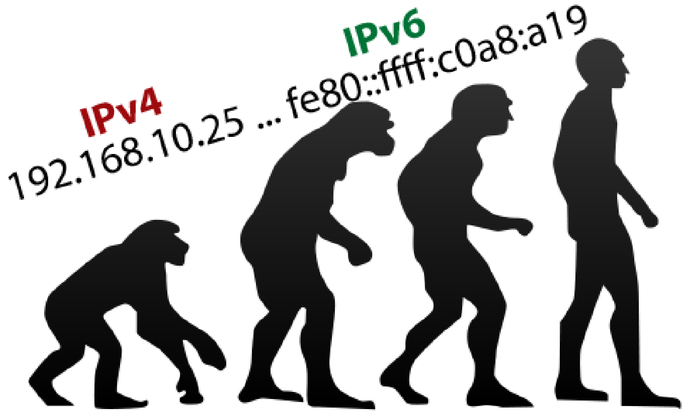
IP hat "viel mitgemacht" in seiner bisherigen Lebensgeschichte, aber:

- grundsätzliches Prinzip funktioniert immer noch
- viele neue Hardwaretypen (Smartphones, IoT, …)

Skalierung

- von einigen wenigen zu mehreren hundert Millionen Computer
- Geschwindigkeit von wenigen KBit/s bis Gbits/s
- drastische Änderungen in der Grössenordnung der Hardware

Motivation für Wechsel



Adressbereich

- 32 Bit-Adressraum ermöglicht Millionen von Netzwerken .... Aber viele sind Klasse C-Netzwerke mit zuwenig Adressen für viele Unternehmungen
- 2^{14} Klasse-B-Netzwerk-Adressen sind aufgebraucht

Neue Servicetypen

- verschiedene Applikationen benötigen unterschiedliche Ansprüche an Zuverlässigkeit und Geschwindigkeit
- gegenwärtiges IP verwendet oft nicht implementierte Servicetypen

Multicasting und Anycast

- Bieten neue Möglichkeiten zur Verbreitung von multimedialen Inhalten oder Anpassung an besondere Verhältnisse

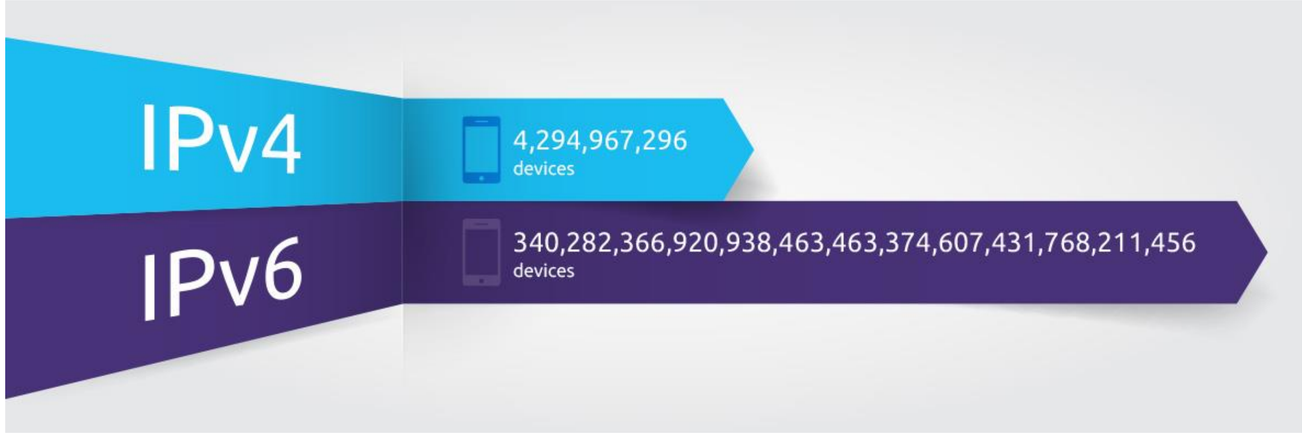
UND verschiedene Forderungen nach Verbesserungen, darunter:

- Optimiertes Routing und Parsing von IP-Paketen für eine einfachere und effizientere Datenübertragung
- Verbesserte Konfigurierbarkeit durch Autokonfiguration, um die Einrichtung zu vereinfachen.

Historie: Name und Versionsnummer

- Vorgängige Versionen als *Internet Protocol next Generation* (IPng) bezeichnet
- Verschiedene Vorschläge als IPng bezeichnet
- Eines wurde ausgewählt und mit der nächsten Nummer bezeichnet

→ Resultat: IPv6 (1998: RFC2460)

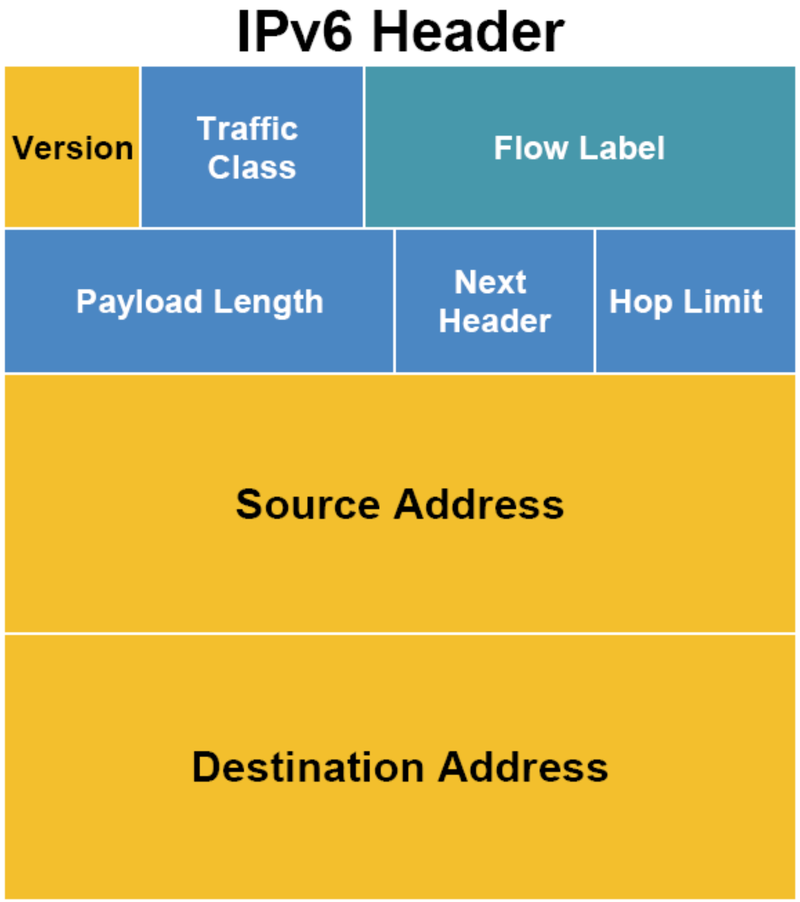


Neue Features

- Adressraum: IPv6 Adressen sind **128 Bit lang**
- Headerformat: total unterschiedlich gegenüber IPv4
- Headererweiterungen: zusätzliche Informationen werden in optionalen Headererweiterungen, gefolgt von den Daten, gespeichert
- Unterstützung von Audio und Video: Flow Labels und Quality of Service erlauben Aufbau geeigneter Audio- und Video-Verbindungen
- Erweiterungen: neue Features können einfach zugefügt werden

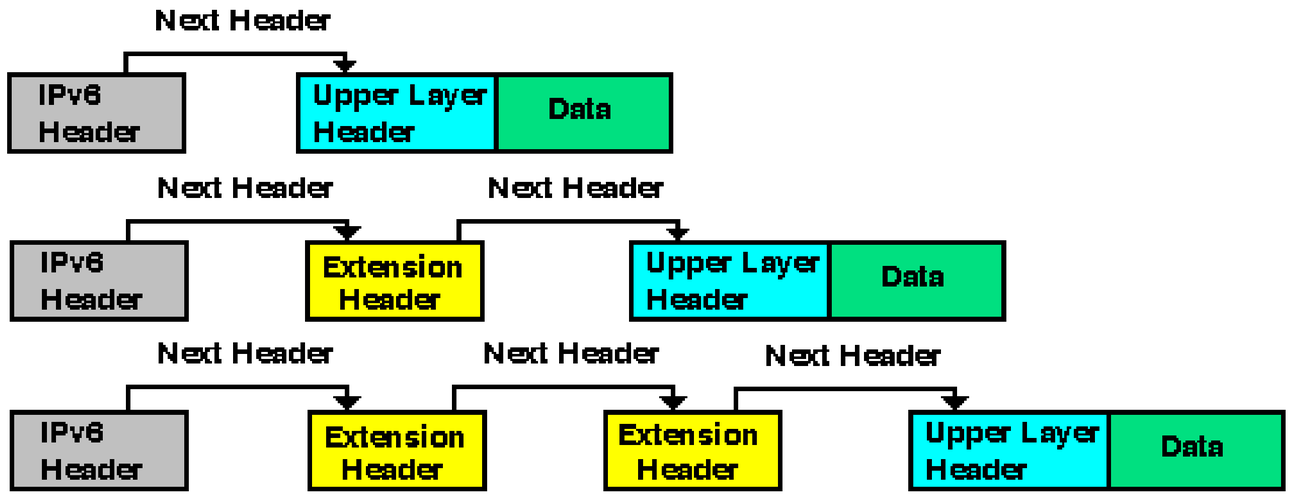
IPv6-Paket: Aufbau und Struktur

- Basis-Header hat feste Länge von 40 Bytes
- "unterwegs" keine Fragmentierung erlaubt
- Weniger Felder als bei IPv4
- Funktion einiger Felder gleich geblieben aber anders benannt



IPv6: Optionale Erweiterungs-Header

- IPv6 besitzt primär nur einen Basisheader.
- Werden weitere Funktionalitäten benötigt können zusätzliche Header hinzugefügt werden.
- Es gibt Empfehlungen, in welcher Reihenfolge diese Header aneinander gefügt werden sollten.



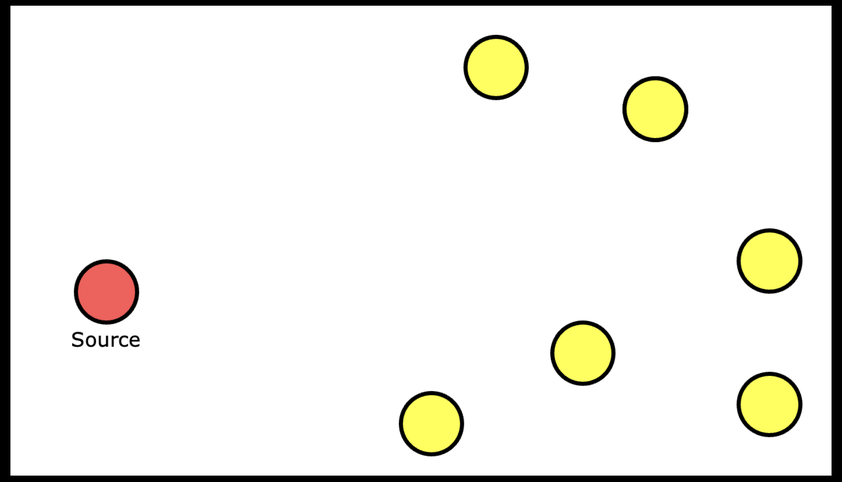
IPv6-Adressierung

- 128-Bit Adressen
- Beinhalten Network Prefix und Host Suffix
- Keine Adressklassen - Prefix/Suffix-Grenzen können überall sein
- Spezielle Typen von Adressen
  - *unicast* - single destination computer
  - *multicast* - multiple destinations; eventuell nicht innerhalb der gleichen Site
  - *anycast/cluster* - Sammlung von Computern mit demselben Prefix; Paket wird an ein Mitglied des Clusters ausgeliefert
- IPv4-Broadcast sind Subsets von Multicast.
- Cluster-Adressierung erlaubt Verdopplung der Dienste
  - wird auch als Anycast bezeichnet!

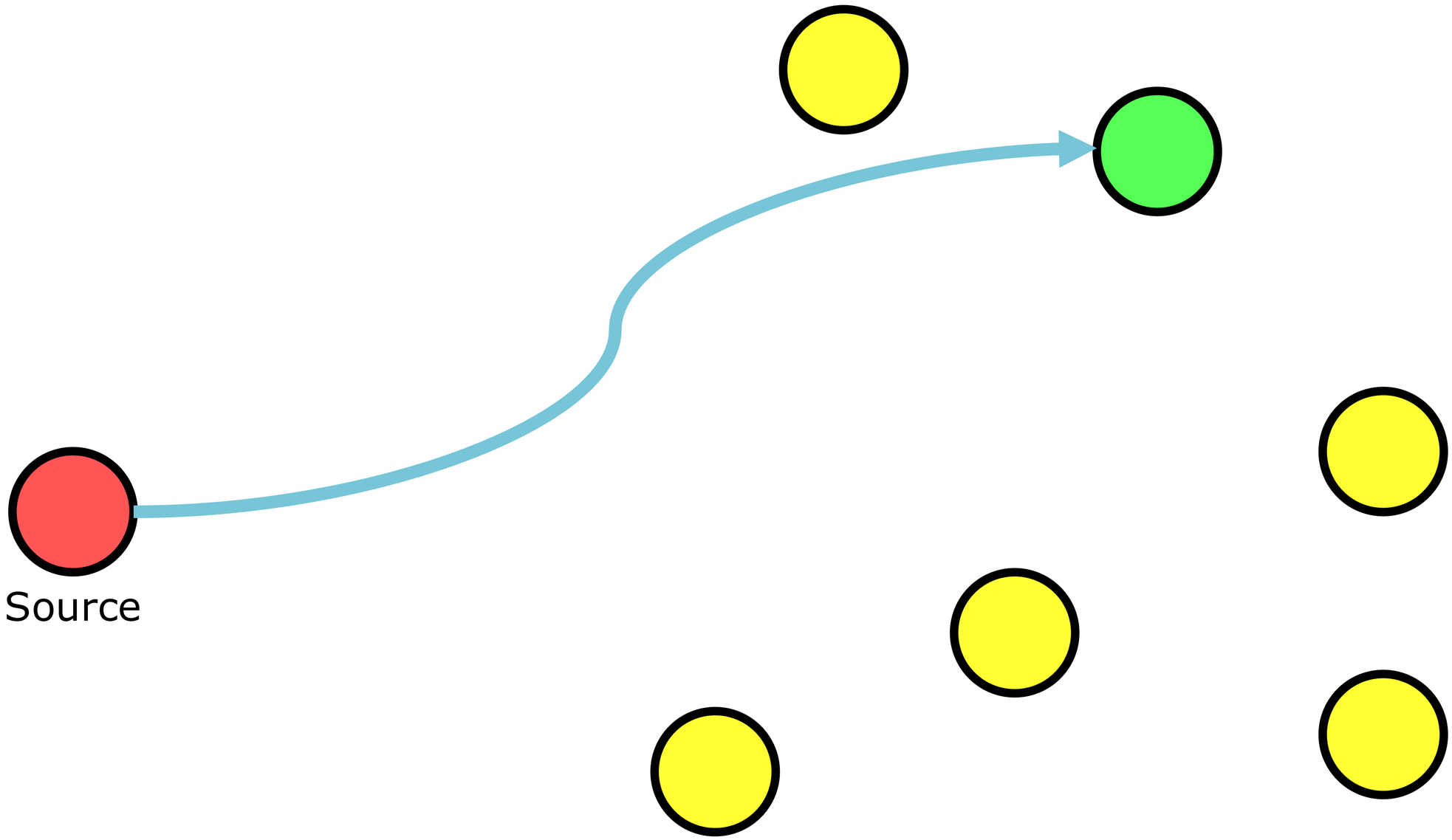
HSLU Hochschule Luzern

Beispiel / Visualisierung am Whiteboard/iPad

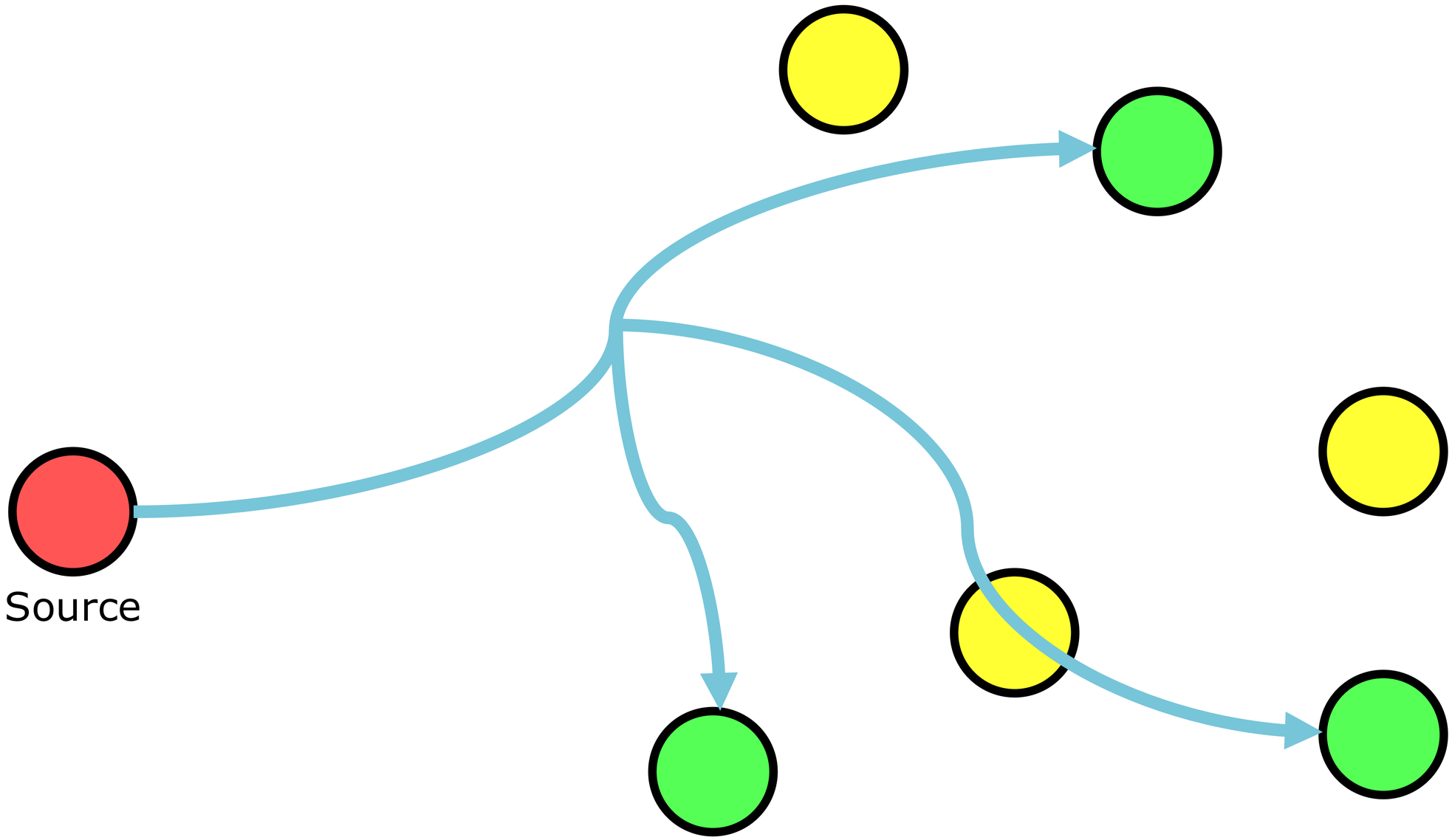
→ Unicast vs. Multicast vs. Anycast



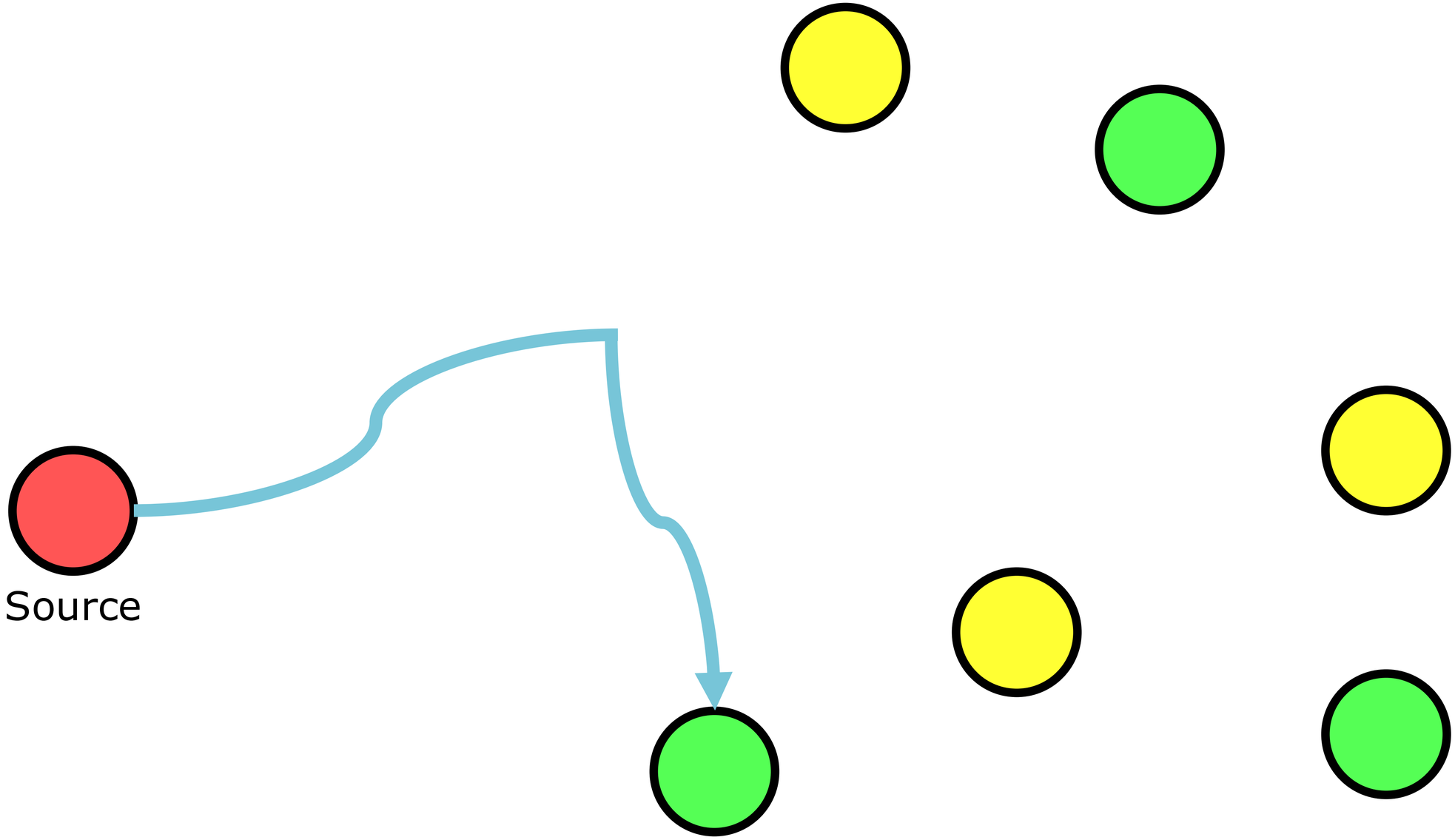
unicast



multicast



anycast



IPv6-Adressnotation

- 128-Bit Adressen in *dotted decimal*-Format; benötigt 16 Zahlen:

  105.220.136.100.255.255.255.255.0.0.18.128.140.10.255.255

- Gruppen von 16 Bit Zahlen in *hexadezimal* Format, unterteilt mit Doppelpunkten:

  69DC:8864:FFFF:FFFF:0:1280:8C0A:FFFF

- IPv6-Adressen mit 96 führenden Nullen wird als IPv4 Adresse interpretiert

- *Zero-compression* - eine Reihe von Nullen kann unterdrückt werden

  FF0C:0:0:0:0:0:0:B1

  wird zu

  FF0C::B1

Kurzschreibweisen – Beispiele und RFC 5952

Die folgenden Schreibweisen sind Repräsentationen der gleichen IPv6-Adresse.

1. 2001 : 0db8 : 0000 : 0000 : 0001 : 0000 : 0000 : 0001
2. 2001 : db8 : 0 : 0 : 1 : 0 : 0 : 1
3. 2001 : 0db8 : 0000 : 000 : 1 : 00 : 0 : 1
4. 2001 : db8 : : 0 : 1 : 0 : 0 : 1
5. 2001 : db8 : 0 : 0 : 1 : : 1
6. 2001 : DB8 : 0 : 0 : 1 : : 1
7. 2001 : db8 : : 1 : 0 : 0 : 1

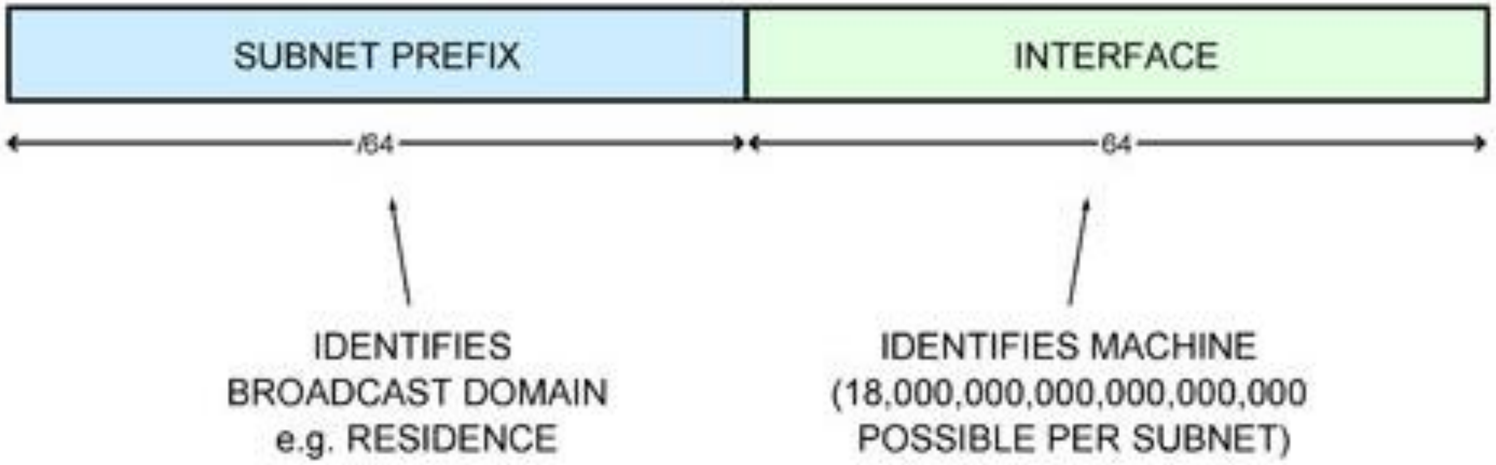
Verbindliche Notationsregeln nach RFC 5952

- Alle alphabetischen Zeichen (Buchstaben a, b, c, d, e, f) werden grundsätzlich klein geschrieben.
- Alle führenden Nullen eines Blocks werden grundsätzlich weggelassen.
- Einer oder mehrere aufeinanderfolgende 4er Nullerblöcke werden durch zwei Doppelpunkte ("::") gekürzt.
- Die Kürzung zu zwei Doppelpunkte ("::") darf nur einmal bei der längsten Folge von Nullerblöcken durchgeführt werden. Oder bei gleicher Länge, die erste von links.

Adress-Aufteilung

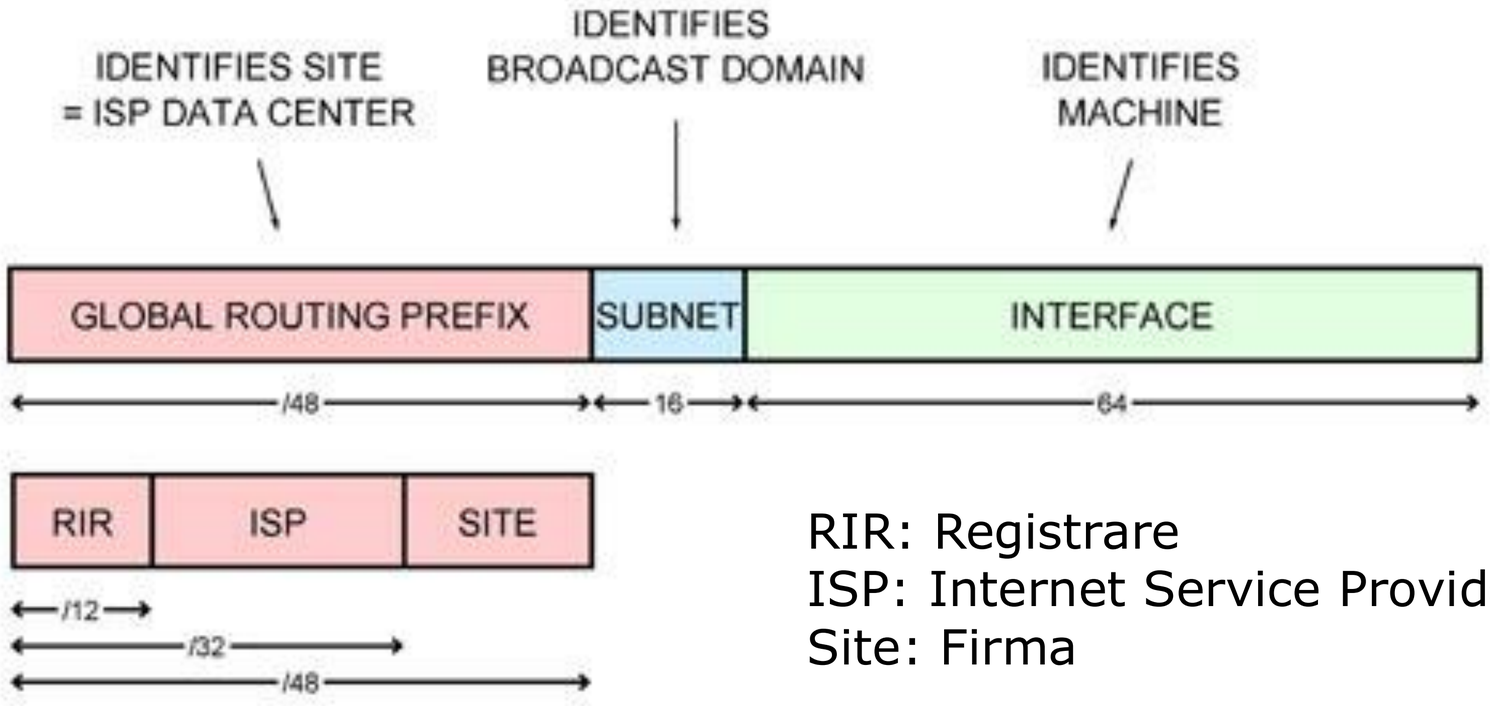
Endsysteme haben immer

- 64 Bit Net/Subnet Prefix
- 64 Bit Interface ID



Adress-Aufteilung

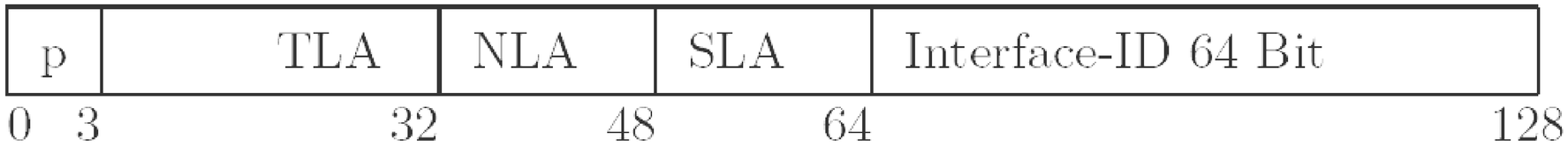
Der Prefix wird folgendermassen aufgebaut:



RIR: Registrare
ISP: Internet Service Provider
Site: Firma

Adress-Aufteilung/Struktur detailliert

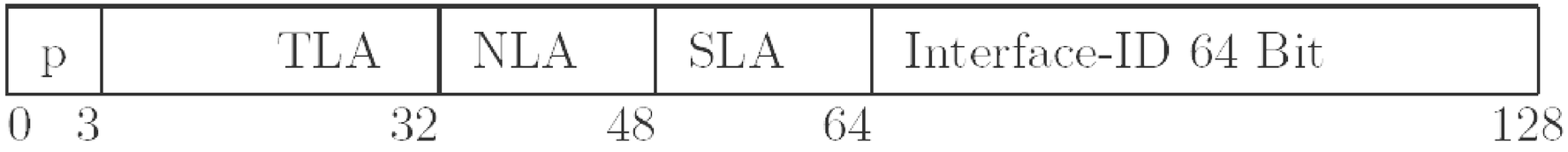
Aufteilung in zwei 64 Bit Teile



- p: Prefix (=001 bei Unicast-Adressen!)
- TLA: Top Level Aggregator (Superprovider) (inkl. einen Reserve-Bereich RES)
- NLA: Next Level Aggregator (Provider/grosse Unternehmen/Reseller)
- SLA: Site Level Aggregator (Subnetzwerke)

Aufteilung hat mehrmals geändert!!

IPv6: Adress-Hierarchie in der Praxis



- Endkunden (zB. Firmen) mit Subnetzen bekommen immer ein /48
- LAN-Segmente sind immer ein /64
- 16 Bit SLA-Space ! 65536 Subnetze, oder eine interne Hierarchie mit (z.B.) 256 Standorten (8 Bit) und dort jeweils 256 Netzsegmenten
- Bei Umzügen zwischen Providern kann die Adressstruktur beibehalten werden, nur das /48-Prefix muss angepasst werden

Hat z. B. ein Netzwerkgerät die IPv6-Adresse

`2001:0db8:85a3:08d3:1319:8a2e:0370:7347/64`

so lautet das Präfix

`2001:0db8:85a3:08d3::/64`

und der Interface-Identifier:

`1319:8a2e:0370:7347`

Der Provider bekam von der RIR wahrscheinlich das Netz

`2001:0db8::/32`

zugewiesen und der Endkunde vom Provider möglicherweise das Netz

`2001:0db8:85a3::/48`

oder aber nur:

`2001:0db8:85a3:0800::/56`

Wichtige IPv6-Adressen

Adresse	Typ	Beschreibung
::/128 (128 0-Bits)	nicht spezifiziert	zeigt das Fehlen einer Adresse an.
::1/128 (127 0-Bits und ein 1-Bit)	Loopback (localhost)	die Adresse auf sich selbst

Wichtige IPv6-Adressen

Adresse	Typ	Beschreibung
::/128 (128 0-Bits)	nicht spezifiziert	zeigt das Fehlen einer Adresse an.
::1/128 (127 0-Bits und ein 1-Bit)	Loopback (localhost)	die Adresse auf sich selbst
2000::/3 (...)	Global Unicast	*normale IPv6-Adressen*
2001:0600:... 2001:09ff:... ...		IPv6-Adresse aus Europa (RIPE)
2001:04xx:... 2001:0018:... ...		IPv6-Adresse aus US (ARIN)
2001:02xx:... 2001:0axx:... ...		IPv6-Adresse aus Asien (APNIC)

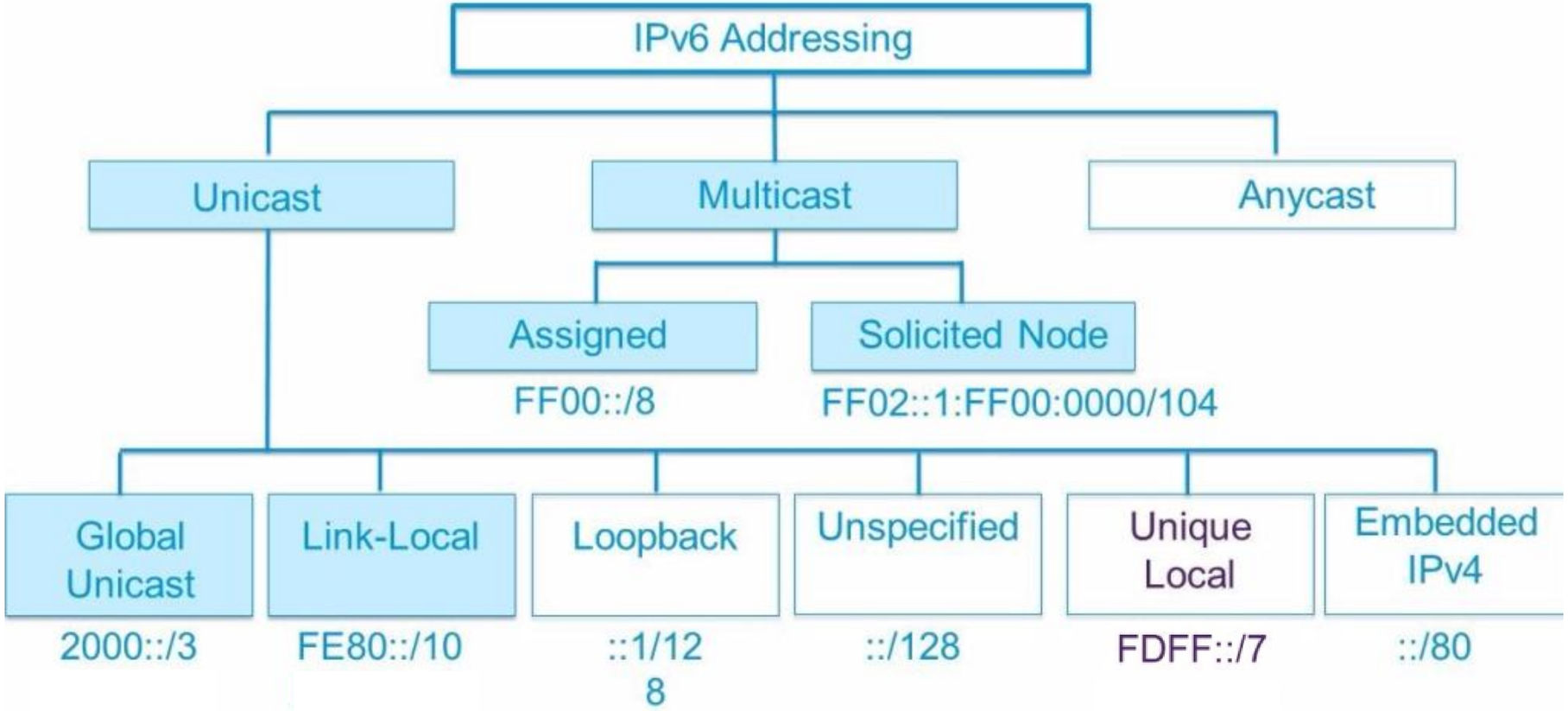
Wichtige IPv6-Adressen

Adresse	Typ	Beschreibung
::/128 (128 0-Bits)	nicht spezifiziert	zeigt das Fehlen einer Adresse an.
::1/128 (127 0-Bits und ein 1-Bit)	Loopback (localhost)	die Adresse auf sich selbst
2000::/3 (...)	Global Unicast	*normale IPv6-Adressen*
2001:0600:... 2001:09ff:... ...		IPv6-Adresse aus Europa (RIPE)
2001:04xx:... 2001:0018:... ...		IPv6-Adresse aus US (ARIN)
2001:02xx:... 2001:0axx:... ...		IPv6-Adresse aus Asien (APNIC)
fe80::/64	Link-Local Unicast	Lokales Netzwerk, von Routern nicht weitergeleitet
fd00::/64	Side-Local Unicast	Private IPv6-Adressen. Weitergeleitet von internen Routern im Unternehmen, aber nicht von Border-Routern ins Internet.

Wichtige IPv6-Adressen

Adresse	Typ	Beschreibung
::/128 (128 0-Bits)	nicht spezifiziert	zeigt das Fehlen einer Adresse an.
::1/128 (127 0-Bits und ein 1-Bit)	Loopback (localhost)	die Adresse auf sich selbst
2000::/3 (...)	Global Unicast	*normale IPv6-Adressen*
2001:0600:... 2001:09ff:... ...		IPv6-Adresse aus Europa (RIPE)
2001:04xx:... 2001:0018:... ...		IPv6-Adresse aus US (ARIN)
2001:02xx:... 2001:0axx:... ...		IPv6-Adresse aus Asien (APNIC)
fe80::/64	Link-Local Unicast	Lokales Netzwerk, von Routern nicht weitergeleitet
fd00::/64	Side-Local Unicast	Private IPv6-Adressen. Weitergeleitet von internen Routern im Unternehmen, aber nicht von Border-Routern ins Internet.
ff00::/8	Multicast	Für multicast-Kommunikation und mehr!
ff01::1		Interface (ff01:...), alle Knoten (alle IPs der Netzwerkkarte/NIC)
ff01::2		Interface (ff01:...), alle Router (sende an alle Router des NICs)
ff02::1	Insbesondere hier: *Link-Local* Multicast (lokales Netzwerk)	**Lokales Netzwerk** (ff02:...)**, alle Knoten (früher Broadcast!)** WICHTIG
ff02::2		**Lokales Netzwerk** (ff02:...)**, alle Router** WICHTIG
ff02::1:2		**Alle DHCPv6-Server im aktuellen lokalen Netzwerk** (ff02:...)
ff05::1:3 → *Side-local* multicast		Alle firmeninternen (ff05:... = side-local multicast) DHCPv6-Server

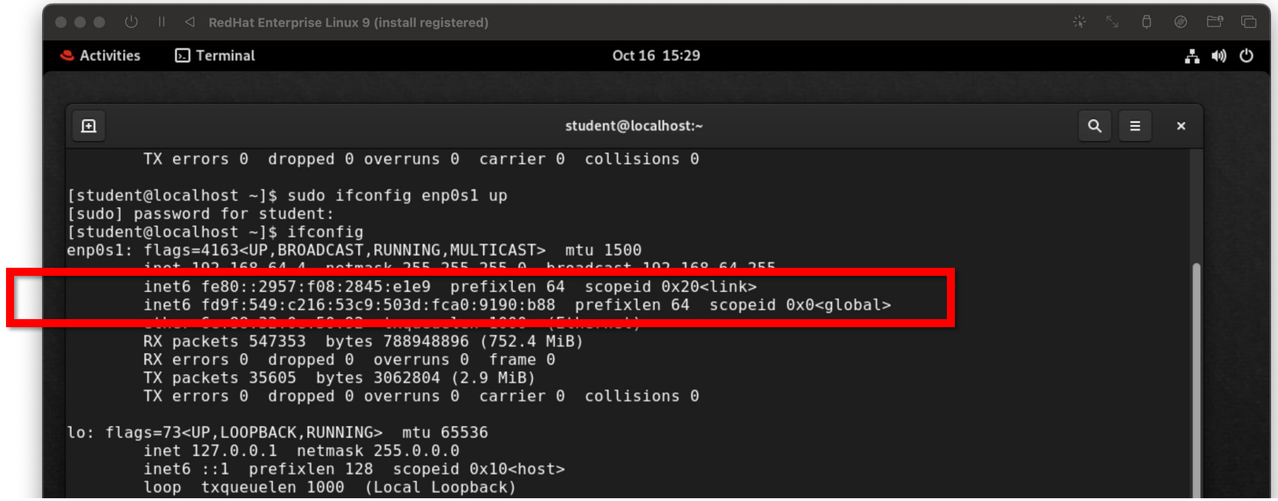
IPv6-Adressarchitektur

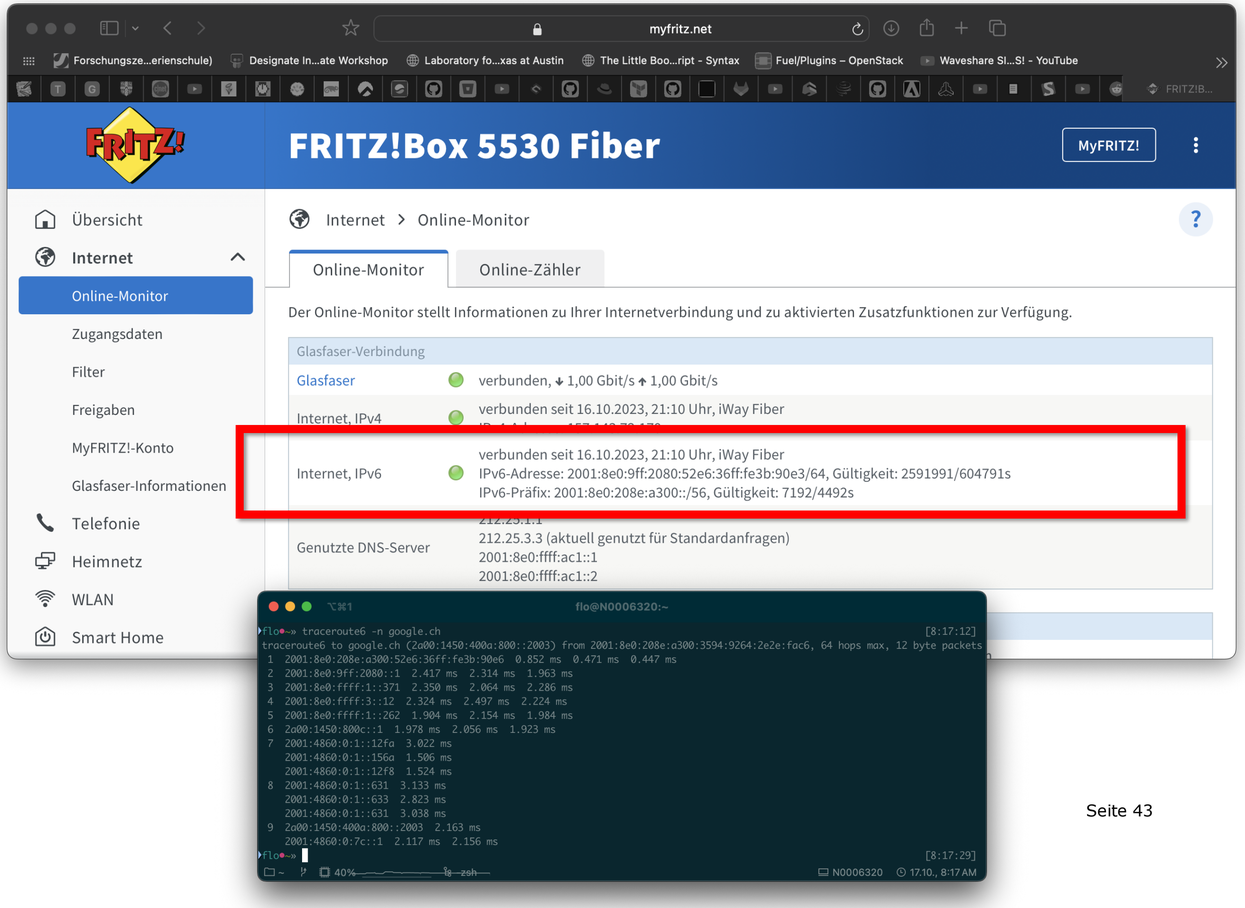


IPv6-Adresse in der Praxis

- Linux Terminal mit IPv6 (*Bild oben)*
  1. Link-local fe80::-Adresse (nur im LAN)
  2. Side-local fdxx::-Adresse (gilt nur in internen Netzen / wird nicht ins Internet geroutet)
  3. *keine öffentliche (globale) IPv6-Adresse*

- FRITZ!Box Heimanschluss mit IPv6 (*Bild unten*)
  1. Öffentliche IPv6-Adresse (WAN):
     2001:8e0:9ff:2080:52e6:36ff:fe3b:90e3
     WAN-Präfix: 2001:8e0:9ff:2080::/64
     (Provider: iWay Fiber)
  2. Zugewiesener Netzwerk-Präfix für Heim-LAN:
     2001:8e0:208e:a300::/56
     tatsächlich genutztes Präfix für das Heimnetz:
     2001:8e0:208e:a300::/64
     Beispiel-IP im internen LAN:
     2001:8e0:208e:a300:4cc:dbcb:9674:1b0c





Verfahren zur IPv6-Adresskonfiguration

Autokonfiguration mit SLAAC, Stateless DHCPv6 und Stateful DHCPv6.

Informatik
27. Oktober 2025

Adresszuteilung #1 - SLAAC

SLAAC: stateless address auto configuration

- Kein DHCP notwendig
- Automatische Adresszuweisung direkt mit dem IPv6-Router
- Automatische Router-Erkennung und Check, ob Adresse schon vorhanden (Nachbar-Erkennung)
- Kein ARP notwendig (da MAC-Teil der IPv6 Adresse)
- Addon für Zusatz-Informationen: DHCPv6 (NTP-Server, ...)

SLAAC Step 1: generating link-local addr.

Generate a link local address (FE80), from MAC address
state: tentative

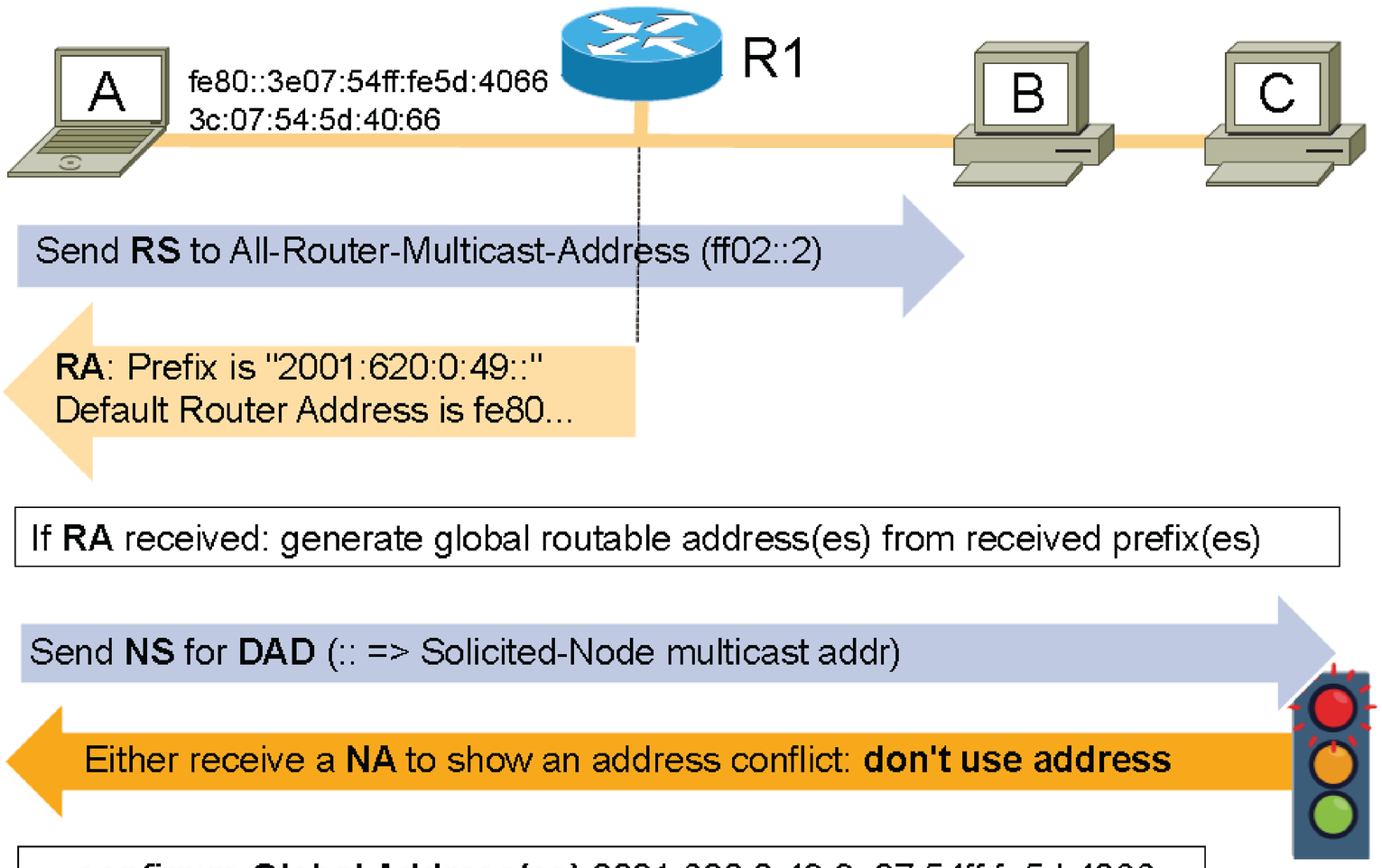
Send **NeighSolicit** for **DupAddrDetect** (:: => Solicited-Node multicast addr)

Either receive a **NeighAdvert** to show an address conflict: **stop autoconfig**

or change state of link local address to: *preferred*
fe80::3e07:54ff:fe5d:4066

Quelle: switch.ch

SLAAC Step 2: generating global addr.



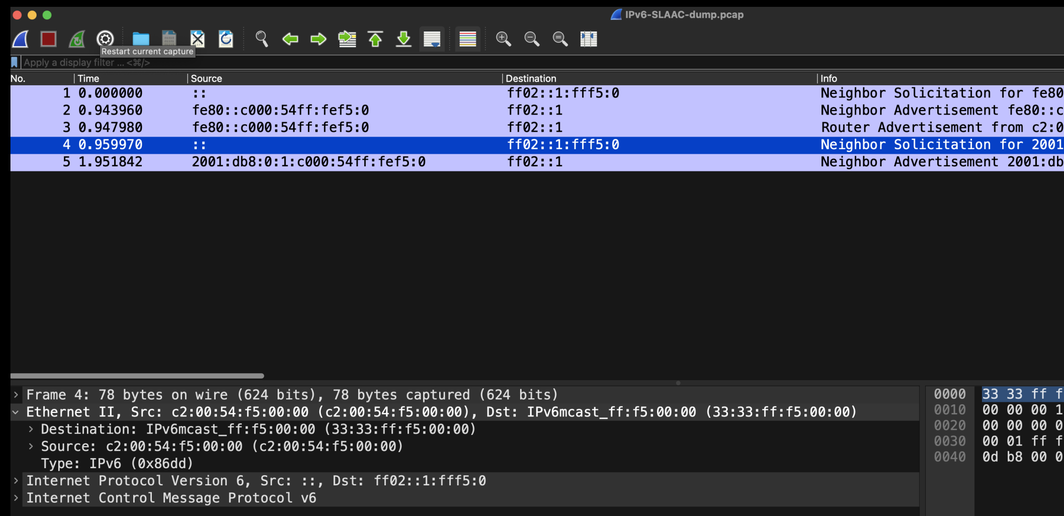
Quelle: switch.ch

Aufgabe

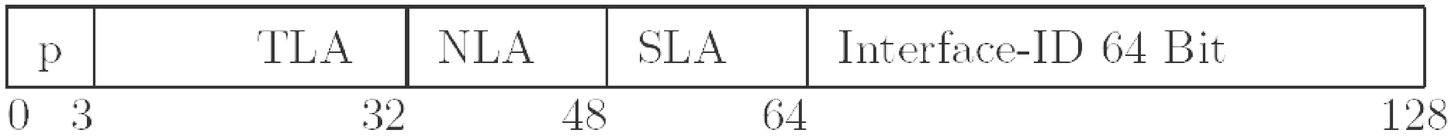
Öffnen Sie den Wireshark-Dump „IPv6-SLAAC-dump.pcapng“ (im ILIAS) und ordnen Sie die Pakete den Schritten zu.

1. Neighbor Solicitation for DAD
2. Router Solicitation
3. Router Advertisment → Network Prefix
5. Neighbor Soliciation #2 for DAD

Wichtig: Verstehen der vielen Multicasts!



IPv6-Multicast



- Format der IPv6-Multicast-Adressen immer ff... (→ p = 111)
- Es gibt keine Broadcasts mehr. Neu: alles Multicasts!
  - Link-Local-Multicast ff02:... *Multicast-Adressen werden als «spezielle» Ziel-Adressen genutzt. Adressierung.*
    - ff02::01 → Ziel für Neighbor-Solicitation (= «Hat schon jemand diese IP-Adresse?») (*alle Teilnehmer/Knoten*)
    - ff02::02 → Ziel für Router-Solicitation-Anfragen (= «Wer ist Router im aktuellen Netz?») (*alle Router*)
  - Spezielle Multicast-Adressen, zB. bei Neighbor-Discovery (Duplicate Adress Detection (DAD))
    - ff02::1:ff<low 24bit> → siehe Wireshark-Dump-Beispiel! (*alle Teilnehmer mit best. IP-Adresse/Adressbereich)*
- Multicasts für Anwendungen: Global Scope ff0e:... (wie Multicast 224.* bei IPv4), kaum Applikationen
- Referenz: RFC2375, RFC3306, RFC3307, RFC3513

ff00::/8		**Für multicast-Kommunikation und mehr!**
ff01::01 ff01::02 ff02::01 ff02::02	**Multicast**	Interface, alle Knoten (alle IPs der Netzwerkkarte/NIC) Interface, alle Router (sende an alle Router des NICs) **Lokales Netzwerk, alle Knoten (früher Broadcast!!)** WICHTIG **Lokales Netzwerk, alle Router** WICHTIG

Mängel der Autokonfiguration (1)

- Neighbor Discovery fehlt einen Mechanismus, um autorisierte Nachbarn zu bestimmen.
- Bedrohungen
  - Umleitungen,
  - Adressen stehlen,
  - Denial of Service,
  - Advertisement und Parameter spoofing
- Es wurden **keine Sicherheitsmassnahmen**, für die zu den DHCP und ARP äquivalenten Protokollen entwickelt
  - Neighbor Discovery Protocol ist unsicher
  - Spoofing von "stateless autoconfiguration messages" möglich

Mängel der Autokonfiguration (2)

- Jeder kann sich als Router ausgeben → Traffic wird über ihn umgeleitet
- Es existieren Mechanismen wie ***Stateless DHCPv6*** oder ***dynamisches DNS***, bei der sich Hosts selbst im DNS eintragen (RFC3736) können

Adresszuteilung #2 - Stateless DHCPv6

- Mit SLAAC (via Router Advertisments) können nur IP-Adressen und DNS-Server zugewiesen werden.
- Weitere Informationen der Netzwerk-Konfiguration kann ein IPv6-Client via SLAAC *nicht* bekommen. Beispielsweise werden NTP-Zeit-Server *nicht* per SLAAC zugewiesen.
- DHCPv6 ist das einzige Verfahren, welches diese und weitere Angaben innerhalb der Autokonfiguration ergänzen kann. Um wie bei IPv4 mit DHCPv4 die gleichen Funktionalitäten für IPv6 zu ermöglichen, wurde *Stateless DHCPv6* definiert.
- Stateless DHCPv6 ist eine Ergänzung zu SLAAC. Es werden nach der SLAAC-Autokonfiguration weitere Informationen von einem DHCPv6-Server abgefragt. Der Vorgang sonst entspricht dem vom IPv4.
- **Praxis**: Per Router-Advertisement werden die IP-Grundparameter verteilt und mit DHCPv6 alles weitere. Die Autokonfiguration bleibt dabei "stateless".

Adresszuteilung #3 - Stateful DHCPv6

- Eine weitere Möglichkeit ist die Adressvergabe *nur* mit DHCP – wie man es von IPv4 her kennt. Hierbei wird *kein* SLAAC gemacht.
- Um wie bei IPv4 mit DHCPv4 die gleichen Funktionalitäten für IPv6 zu ermöglichen, wurde *Stateful DHCPv6* definiert.
- Es wird genauso wie bei IPv4 die IP-Adresse beim DHCP-Server angefragt und diese dann vom DHCPv6-Server zugewiesen.
- Abweichend von DHCPv4 läuft bei DHCPv6 die Kommunikation über die UDP-Ports 546 (Client) und 547 (Server)
- Weitere Informationen wie NTP- und/oder DNS-Server werden in diesem Zuge zusätzlich mitgeteilt und können vom IPv6-Client übernommen werden.

Adresszuteilung #4 - Übersicht

Weitere Konfigurationen möglich

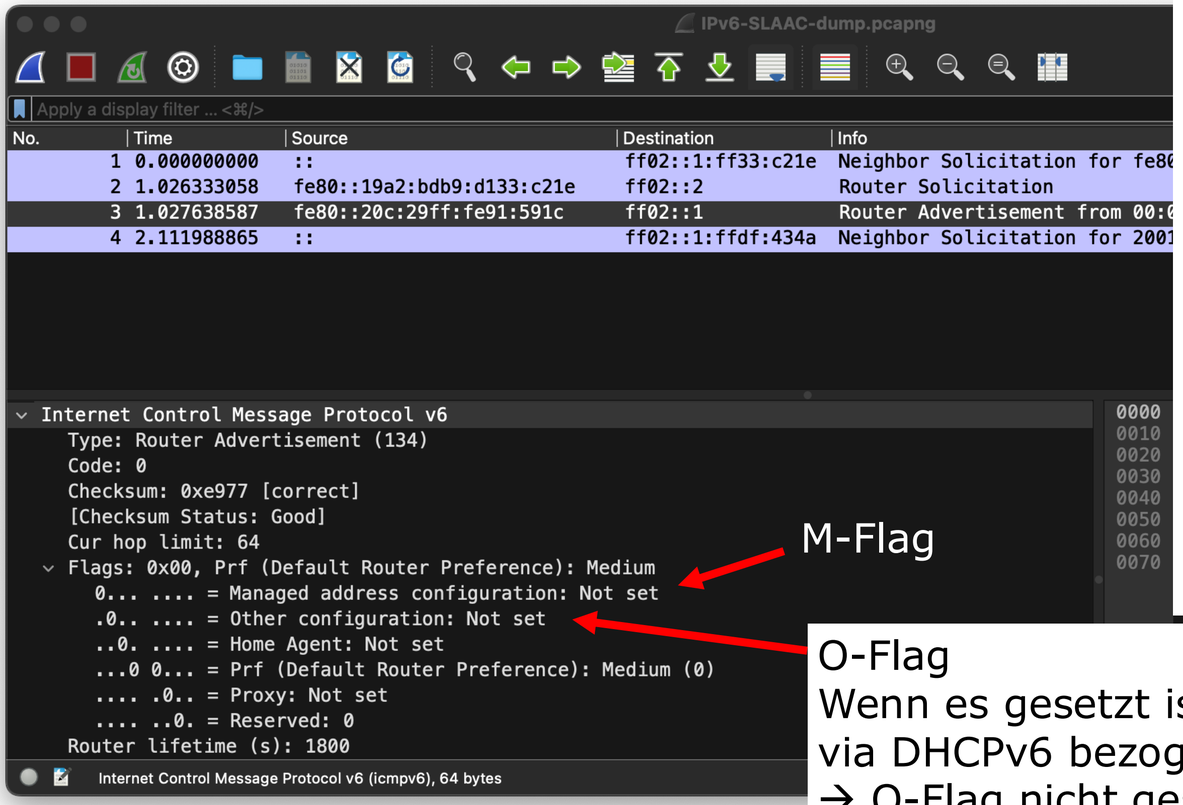
- **Manuell**
  1. Manuelle Adress-Konfiguration vom Netzwerk-Administrator auf dem Client: nur lokal
  2. Manuelle Adress-Konfiguration vom Administrator: global (Internet) (benötigt Router)
- **Automatisch**
  3. SLACC (ohne DHCP) *(präferiert!)*
  4. Stateless DHCPv6: SLACC + Infos via DHCP
  5. Stateful DHCPv6
     (Vergabe von Adressen wie bisher via DHCPv6)
  6. Hybride Verfahren ...

IPv6 autoconfiguration options

Address Autoconfiguration Method	ICMPv6 RA (Type 134) Flags M Flag	O Flag	ICMPv6 RA (Type 134) ICMPv6 Option Prefix Info A Flag	L Flag	Prefix Derived from	Interface ID Derived from	Other Configuration Options
Link-Local (always configured)	N/A	N/A	N/A	N/A	Internal (fe80::)	M-EUI-64 or Privacy	Manual
SLAAC	Off	Off	On	On	RA	M-EUI-64 or Privacy	Manual
Stateful (DHCPv6)	On	On	Off	On	DHCPv6	DHCPv6	DHCPv6
Stateless DHCPv6	Off	On	On	On	RA	M-EUI-64 or Privacy	DHCPv6
Combination Stateless & DHCPv6 (results in up to 3 IPv6 addresses per network prefix)	On	On	On	On	RA and DHCPv6	M-EUI-64 or Privacy and DHCPv6	DHCPv6

Abbildung: Im Router Advertisment-Packet kündigt der IPv6-Router die Adresskonfigurationsmöglichkeiten via *Flags* an.

Beispiel – Wireshark mit Router Advertisment für SLAAC



M-Flag

IPv6 autoconfiguration options

Address Autoconfiguration Method	ICMPv6 RA (Type 134) Flags M Flag	O Flag	ICMPv6 RA (Type 134) ICMPv6 Option Prefix Info A Flag	L Flag	Prefix Derived from	Interface ID Derived from	Other Configuration Options
Link-Local (always configured)	N/A	N/A	N/A	N/A	Internal (fe80::)	M-EUI-64 or Privacy	Manual
SLAAC	Off	Off	On	On	RA	M-EUI-64 or Privacy	Manual
Stateful (DHCPv6)	On	On	Off	On	DHCPv6	DHCPv6	DHCPv6
Stateless DHCPv6	Off	On	On	On	RA	M-EUI-64 or Privacy	DHCPv6
Combination Stateless & DHCPv6 (results in up to 3 IPv6 addresses per network prefix)	On	On	On	On	RA and DHCPv6	M-EUI-64 or Privacy and DHCPv6	DHCPv6

O-Flag
Wenn es gesetzt ist, sollen weitere Netzwerk-Konfigurationsinformationen via DHCPv6 bezogen werden.
→ O-Flag nicht gesetzt: keine Infos abrufbar, SLAAC (siehe auch M-Flag)
→ O-Flag gesetzt: Stateless DHCPv6 mit SLAAC *oder* Stateful DHCP

IPv6-Headerstruktur

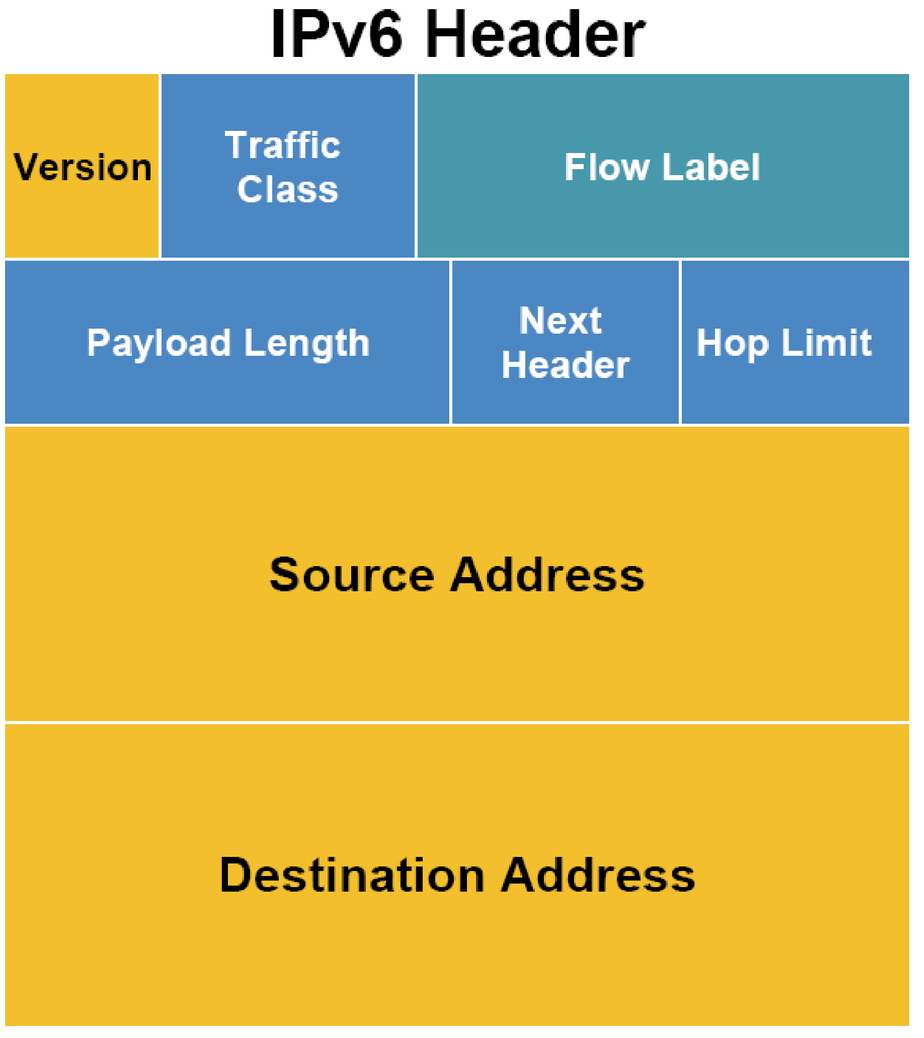
Basisheader vs. Header Extensions

Informatik
27. Oktober 2025

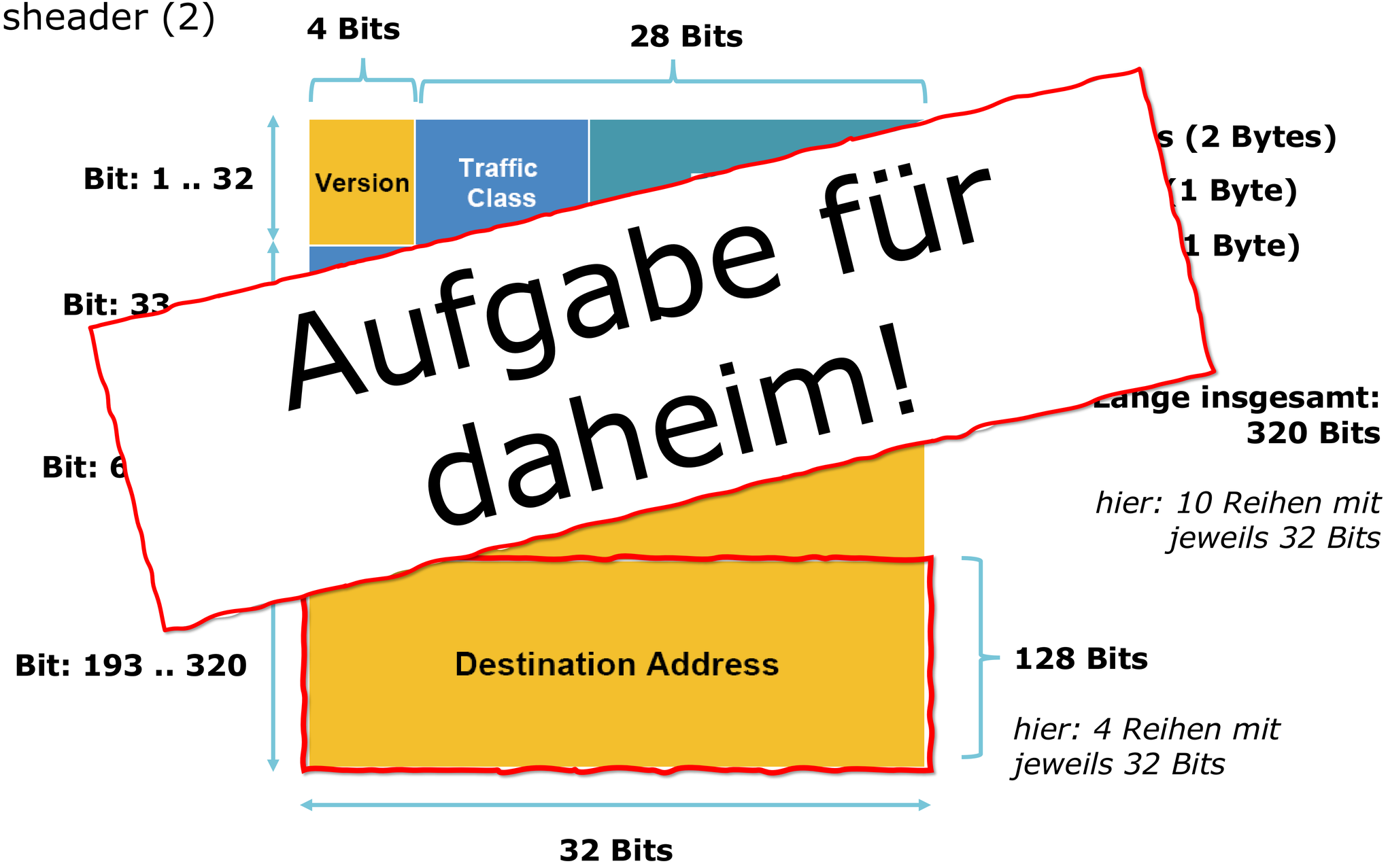
IPv6-Basisheader (1)

- Enthält weniger Informationen als IPv4
- Basisheader vs. Header Extensions
  - Fixe Länge: Vorteil ist, dass auch einfache Geräte den Basis-Header parsen können
  - Erweiterung dennoch möglich: NEXT HEADER-Feld zeigt auf ersten Erweiterungsheader
- FLOW LABEL: Wird verwendet, um Paket einem Datenfluss zwischen 2 Applikationen zuzuordnen
  - Traffic Klasse
  - Specific Path
  - Router verwenden FLOW LABEL, um Pakete entlang eines vorbereiteten Pfads zu übermitteln
- NEXT HEADER-Feld: Identifiziert den Typ des nächsten Kopfdatenbereiches.
  - Dieser kann entweder wie gewöhnlich auf ein Protokoll höherer Schicht zeigen, wie bspw. TCP (Typ 6) oder UDP (Typ 17), oder auf den ersten Erweiterungsheader

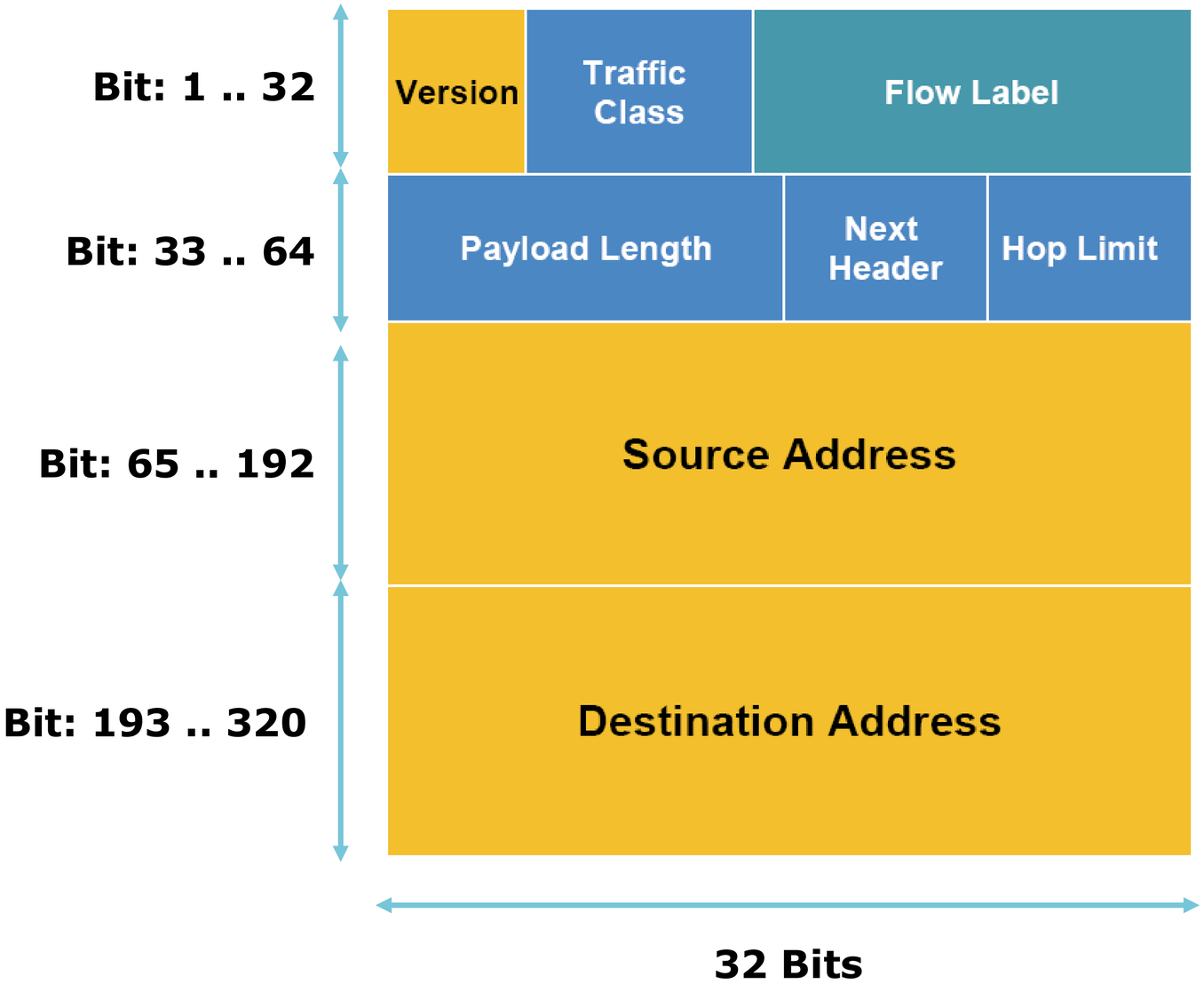
IPv6-Basisheader (2)



IPv6-Basisheader (2)



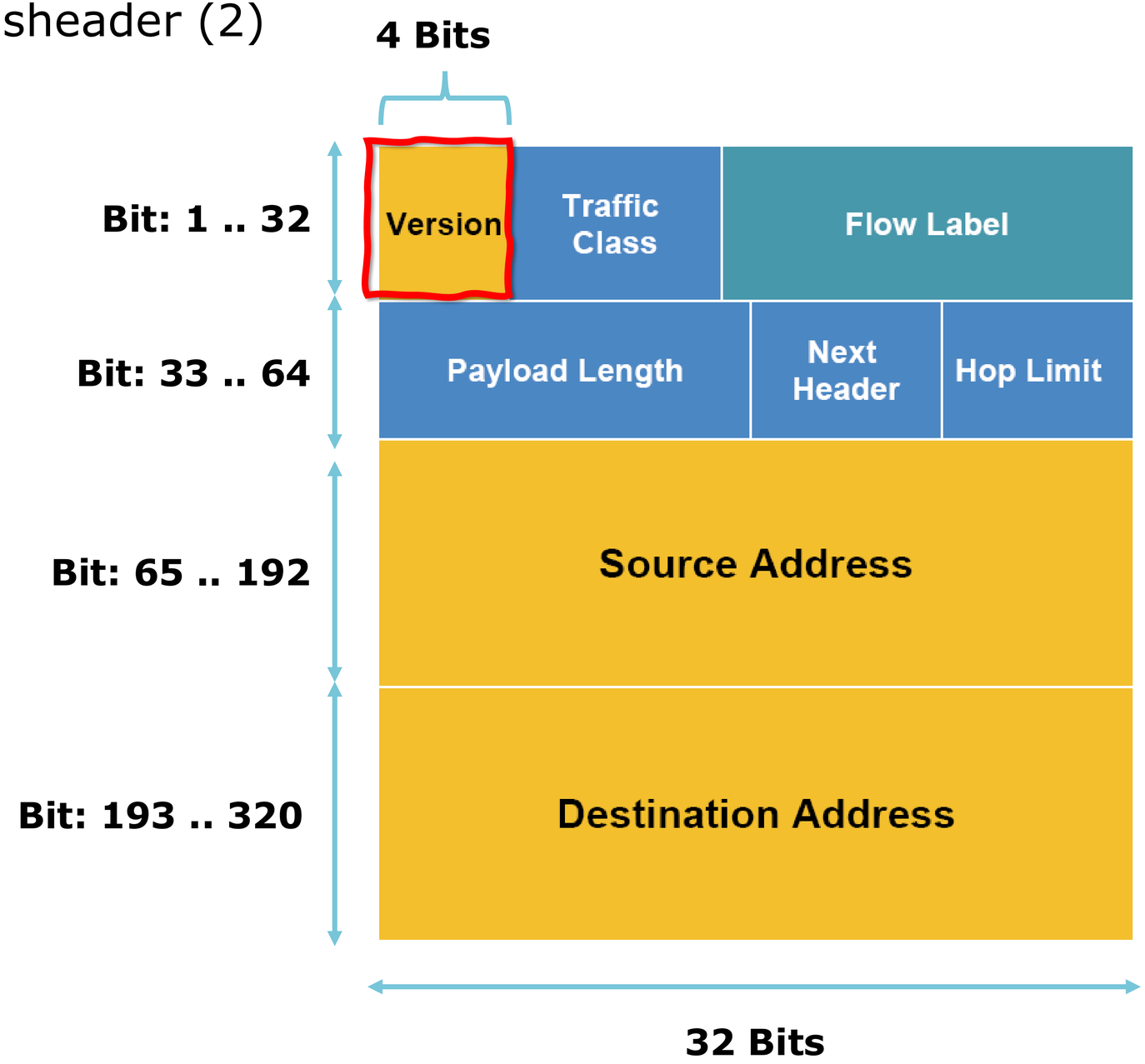
IPv6-Basisheader (2)



Länge insgesamt: 320 Bits

hier: 10 Reihen mit jeweils 32 Bits

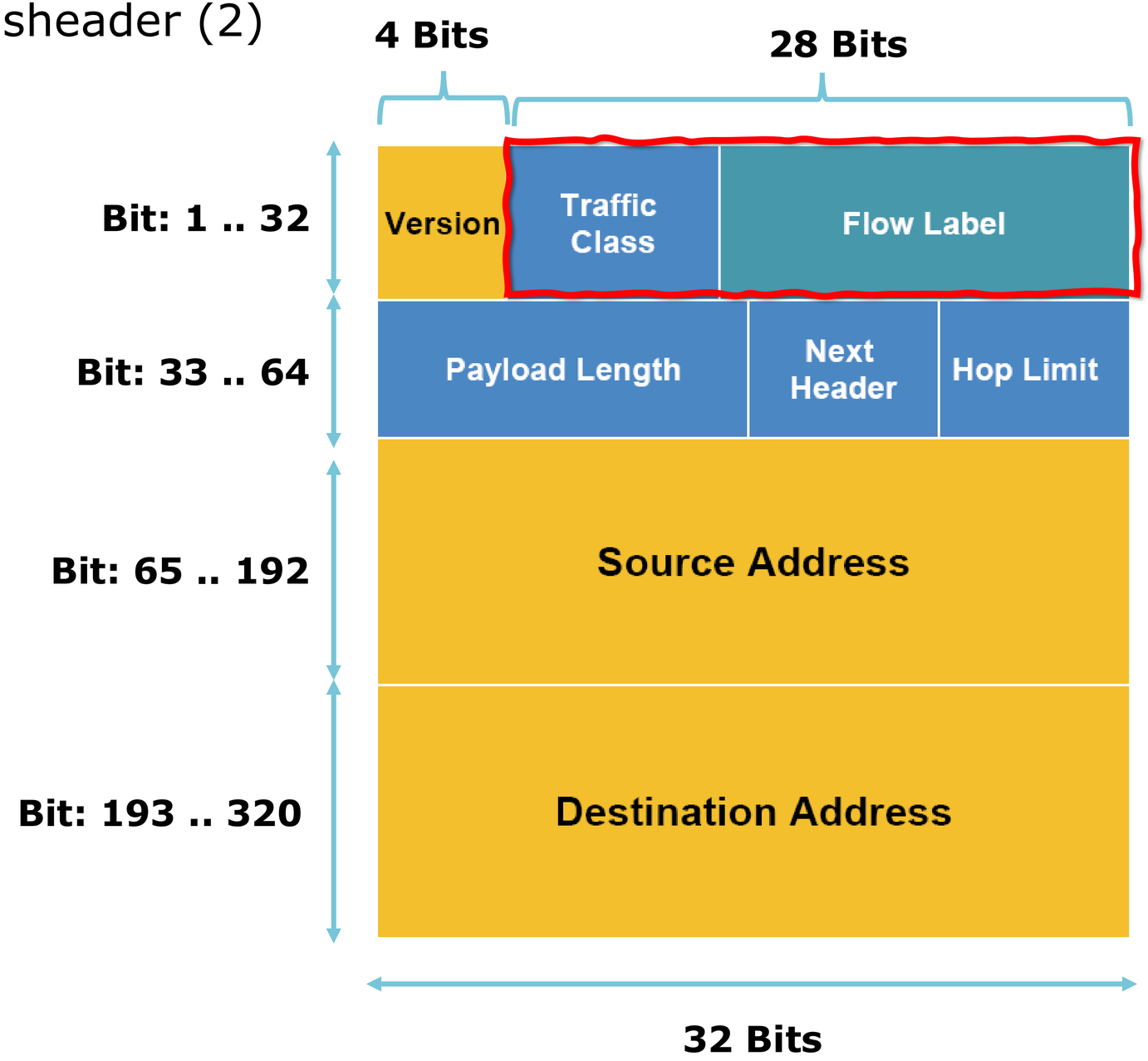
IPv6-Basisheader (2)



Länge insgesamt: 320 Bits

hier: 10 Reihen mit jeweils 32 Bits

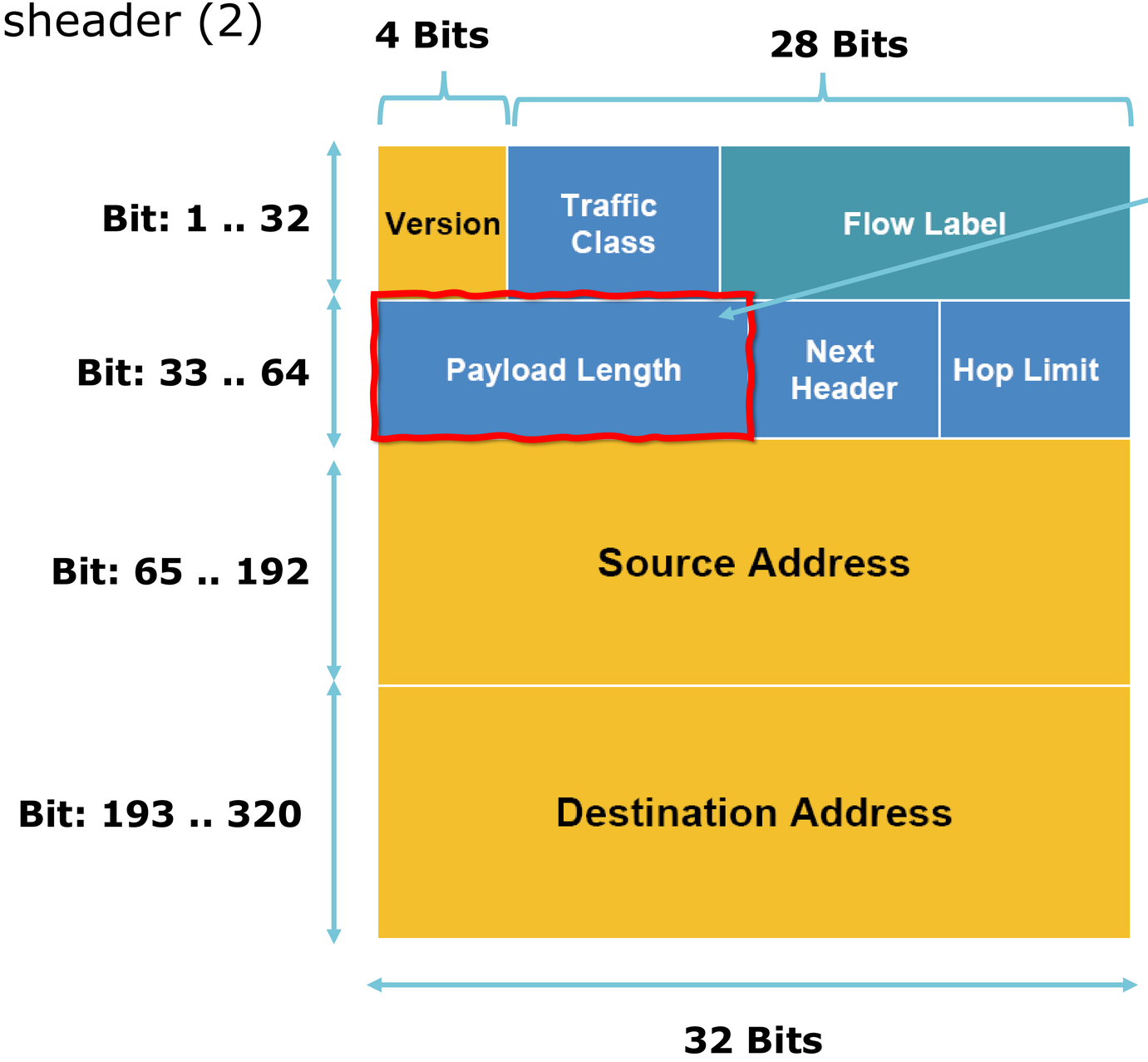
IPv6-Basisheader (2)



Länge insgesamt: 320 Bits

hier: 10 Reihen mit jeweils 32 Bits

IPv6-Basisheader (2)

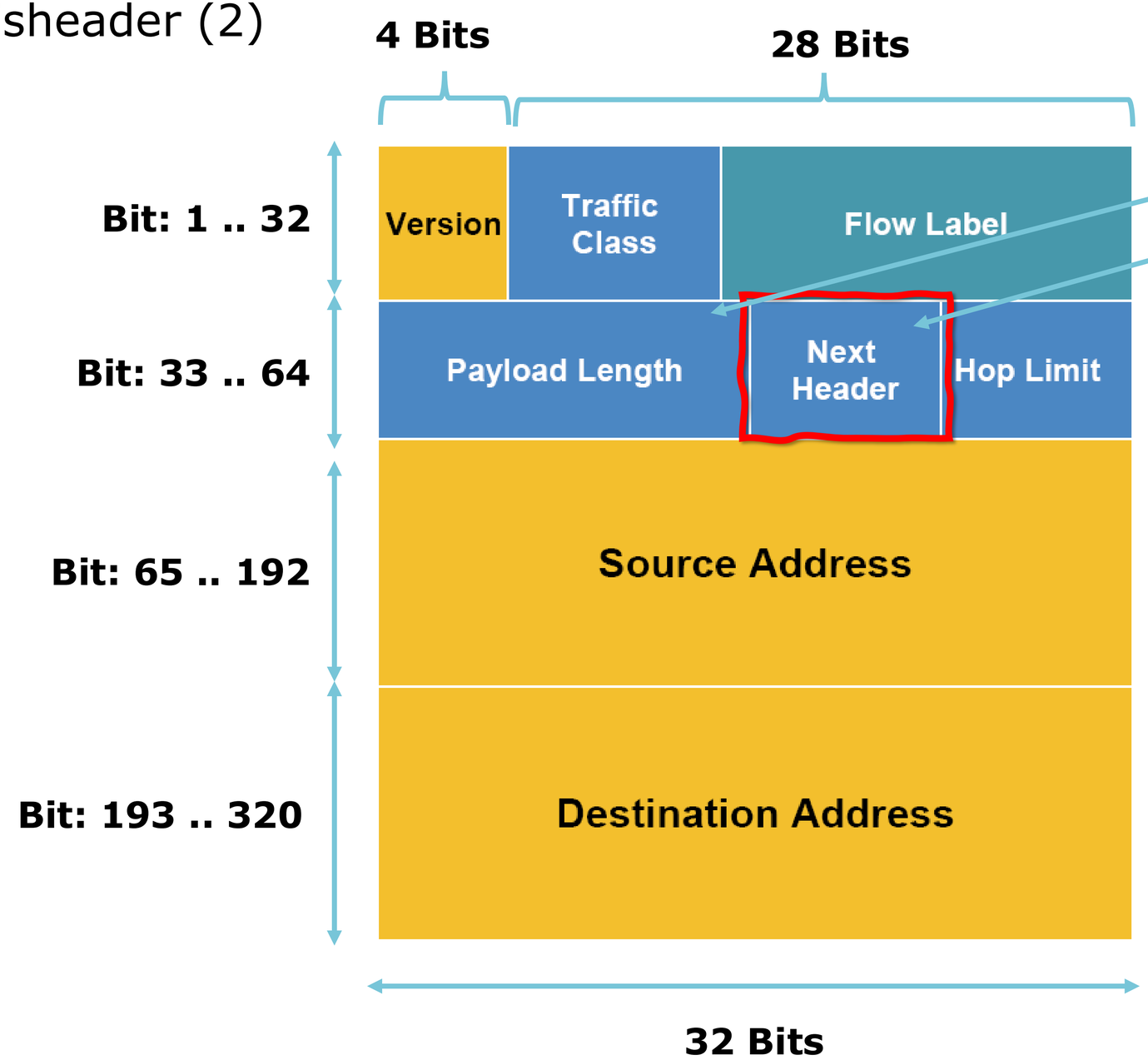


16 Bits (2 Bytes)

Länge insgesamt: 320 Bits

hier: 10 Reihen mit jeweils 32 Bits

IPv6-Basisheader (2)



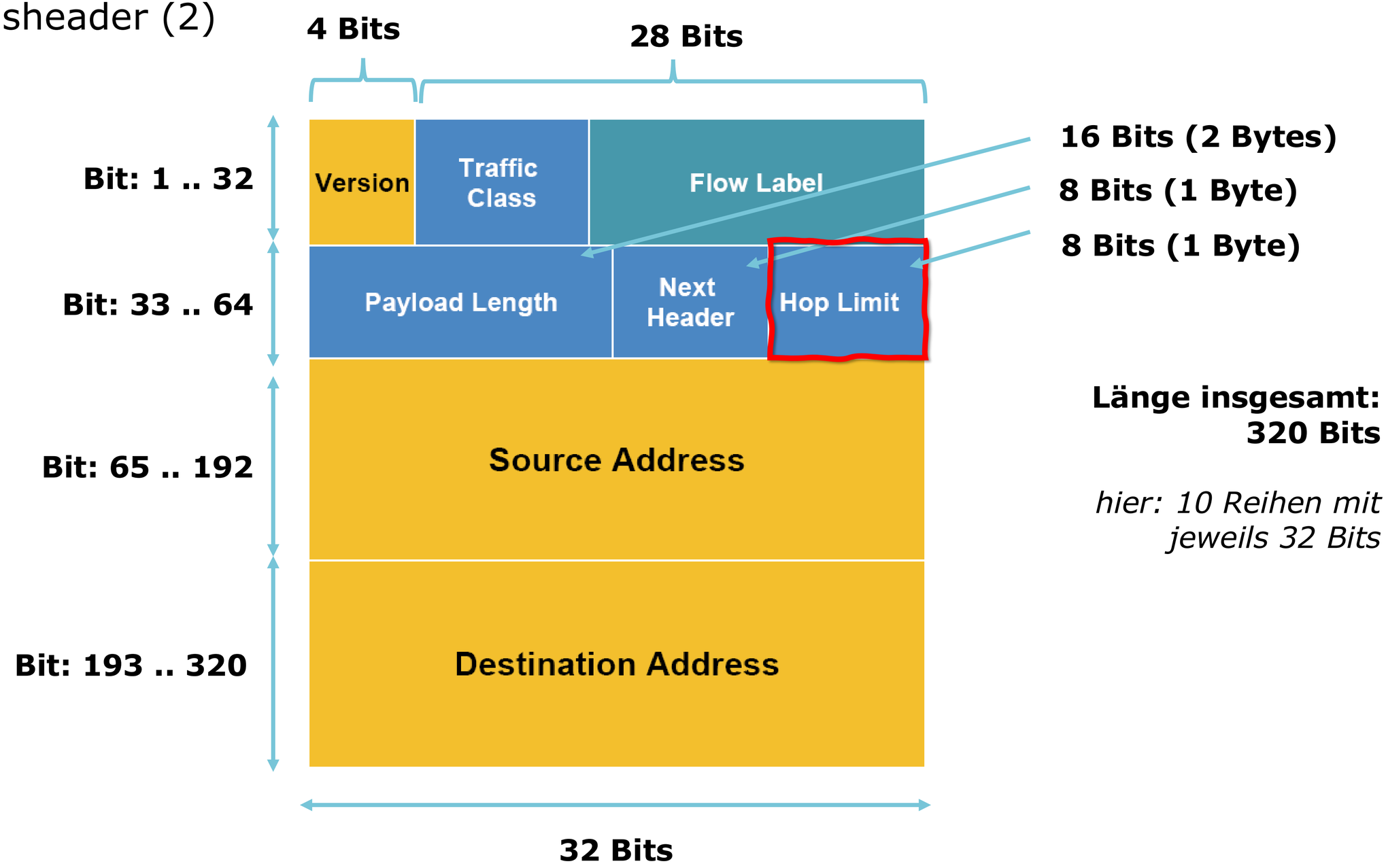
16 Bits (2 Bytes)

8 Bits (1 Byte)

Länge insgesamt: 320 Bits

hier: 10 Reihen mit jeweils 32 Bits

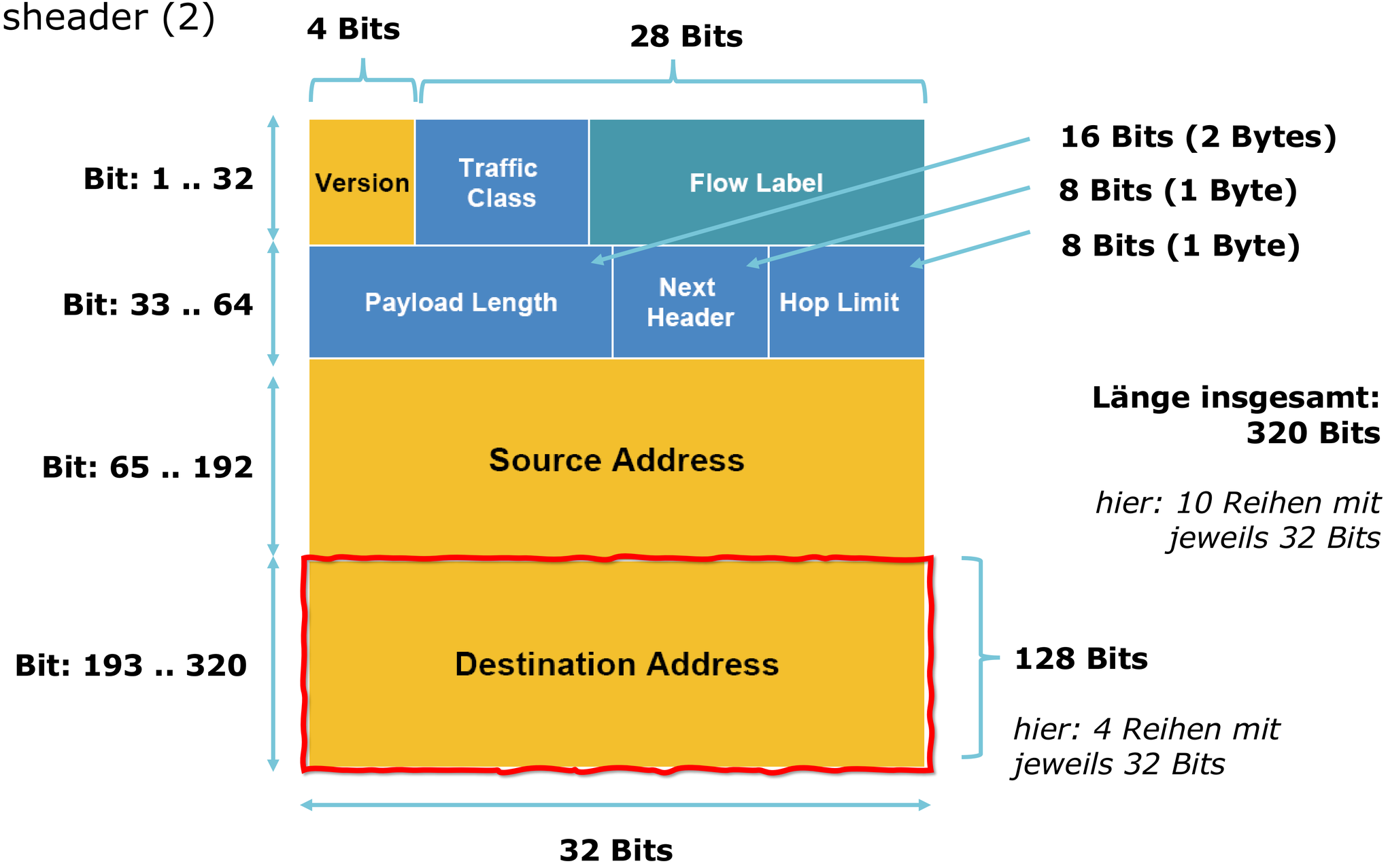
IPv6-Basisheader (2)



Länge insgesamt: 320 Bits

hier: 10 Reihen mit jeweils 32 Bits

IPv6-Basisheader (2)



Wireshark-Dump

Payload-Länge ist in diesem Paket 60 Bits: 00 3c (hex) = 60 dezimal

IP-Version ist 6: 6 (hex) = 0110 (binary, 4 B

kein Traffic c flow label Bits)

```
0000                                                                  `.
0010  6                                                               ..........
0020  2                                                               ..............
0030  00                                                   01 10      .......".#.<....
0040  08                                                   39 cf 88 08 00   .t........9....
0050  27                                  04 00 17 00 18 00 08 00 02   '...............
0060  00 0                  00 0c 27 fe 8f 95 00 00 0e 10 00 00   ......'.........
0070  15 18                                                           ..
```

Aufgabe für daheim!

Dest. Address: ff02::1:2 (Multicast, alle DHCPv6-Server im lokalen Netzwerk)

Next header ist UDP! 11 hex = 17 dezimal → UDP

Hop limit: 1 → Nur link local, im internen Netz

Source Address: (link local !) fe80::a00:27ff:fefe:8f95

Wireshark-Dump

Vorsicht: hexadezimale Darstellung!
1 Ziffer entspricht 4 Bits.

6 (hex) =
0110 (binary, 4 Bits)

```
0000                                                  60 00                  `.
0010   00 00 00 3c 11 01 fe 80 00 00 00 00 00 00 0a 00   ...<............
0020   27 ff fe fe 8f 95 ff 02 00 00 00 00 00 00 00 00   '...............
0030   00 00 00 01 00 02 02 22 02 23 00 3c ad 08 01 10   .......".#.<....
0040   08 74 00 01 00 0e 00 01 00 01 1c 39 cf 88 08 00   .t.........9....
0050   27 fe 8f 95 00 06 00 04 00 17 00 18 00 08 00 02   '...............
0060   00 00 00 19 00 0c 27 fe 8f 95 00 00 0e 10 00 00   ......'.........
0070   15 18                                             ..
```

Wireshark-Dump

Vorsicht: hexadezimale Darstellung!
1 Ziffer entspricht 4 Bits.

6 (hex) =
0110 (binary, 4 Bits)

f (hex) =
1111 (binary, 4 Bits)

```
0000                                                  60 00                  `.
0010   00 00 00 3c 11 01 fe 80 00 00 00 00 00 00 0a 00   ...<............
0020   27 ff fe fe 8f 95 ff 02 00 00 00 00 00 00 00 00   '...............
0030   00 00 00 01 00 02 02 22 02 23 00 3c ad 08 01 10   .......".#.<....
0040   08 74 00 01 00 0e 00 01 00 01 1c 39 cf 88 08 00   .t.........9....
0050   27 fe 8f 95 00 06 00 04 00 17 00 18 00 08 00 02   '...............
0060   00 00 00 19 00 0c 27 fe 8f 95 00 00 0e 10 00 00   ......'.........
0070   15 18                                             ..
```

Wireshark-Dump

Vorsicht: hexadezimale Darstellung!
1 Ziffer entspricht 4 Bits.

Zwei Ziffern → 2 x 4 Bits = 1 Byte

6	**0**	**=**	**(hex)**
0110	**0000**		**(binary)**
4 Bits	**+ 4 Bits**		

```
0000                                                  60 00                  `.
0010   00 00 00 3c 11 01 fe 80 00 00 00 00 00 00 0a 00   ...<............
0020   27 ff fe fe 8f 95 ff 02 00 00 00 00 00 00 00 00   '...............
0030   00 00 00 01 00 02 02 22 02 23 00 3c ad 08 01 10   .......".#.<....
0040   08 74 00 01 00 0e 00 01 00 01 1c 39 cf 88 08 00   .t.........9....
0050   27 fe 8f 95 00 06 00 04 00 17 00 18 00 08 00 02   '...............
0060   00 00 00 19 00 0c 27 fe 8f 95 00 00 0e 10 00 00   ......'.........
0070   15 18                                             ..
```

Wireshark-Dump

28 bits = 7 x 4 Bits

Vorsicht: hexadezimale Darstellung!
1 Ziffer entspricht 4 Bits.

0	**0**	**0**	**0**	**0**	**0**	**0**	**(hex)**
0000	**0000**	**0000**	**0000**	**0000**	**0000**	**0000**	**(binary)**
4 Bits	**+ 4+4 Bits**		**+ 4+4 Bits**		**+ 4+4 Bits**		

```
0000                                                    60 00                   `.
0010   00 00 00 3c 11 01 fe 80 00 00 00 00 00 00 0a 00   ...<............
0020   27 ff fe fe 8f 95 ff 02 00 00 00 00 00 00 00 00   '...............
0030   00 00 00 01 00 02 02 22 02 23 00 3c ad 08 01 10   .......".#.<....
0040   08 74 00 01 00 0e 00 01 00 01 1c 39 cf 88 08 00   .t.........9....
0050   27 fe 8f 95 00 06 00 04 00 17 00 18 00 08 00 02   '...............
0060   00 00 00 19 00 0c 27 fe 8f 95 00 00 0e 10 00 00   ......'.........
0070   15 18                                             ..
```

Wireshark-Dump - Dekodierung

IP-Version ist 6:
6 (hex) =
0110 (binary, 4 Bits)

```
0000                                                  60 00                  `.
0010   00 00 00 3c 11 01 fe 80 00 00 00 00 00 00 0a 00   ...<............
0020   27 ff fe fe 8f 95 ff 02 00 00 00 00 00 00 00 00   '...............
0030   00 00 00 01 00 02 02 22 02 23 00 3c ad 08 01 10   .......".#.<....
0040   08 74 00 01 00 0e 00 01 00 01 1c 39 cf 88 08 00   .t.........9....
0050   27 fe 8f 95 00 06 00 04 00 17 00 18 00 08 00 02   '...............
0060   00 00 00 19 00 0c 27 fe 8f 95 00 00 0e 10 00 00   ......'.........
0070   15 18                                             ..
```

Wireshark-Dump - Dekodierung

IP-Version ist 6:
6 (hex) =
0110 (binary, 4 Bits)

kein Traffic oder flow label
(insgesamt 28 Bits)

```
0000                                                  60 00                  `.
0010   00 00 00 3c 11 01 fe 80 00 00 00 00 00 00 0a 00   ...<............
0020   27 ff fe fe 8f 95 ff 02 00 00 00 00 00 00 00 00   '...............
0030   00 00 00 01 00 02 02 22 02 23 00 3c ad 08 01 10   .......".#.<....
0040   08 74 00 01 00 0e 00 01 00 01 1c 39 cf 88 08 00   .t.........9....
0050   27 fe 8f 95 00 06 00 04 00 17 00 18 00 08 00 02   '...............
0060   00 00 00 19 00 0c 27 fe 8f 95 00 00 0e 10 00 00   ......'.........
0070   15 18                                             ..
```

Wireshark-Dump - Dekodierung

Payload-Länge ist in diesem Paket 60 Bytes: 00 3c (hex) = 60 dezimal

IP-Version ist 6: 6 (hex) = 0110 (binary, 4 Bits)

kein Traffic oder flow label (insgesamt 28 Bits)

```
0000                                                  60 00                  `.
0010   00 00 00 3c 11 01 fe 80 00 00 00 00 00 00 0a 00   ...<............
0020   27 ff fe fe 8f 95 ff 02 00 00 00 00 00 00 00 00   '...............
0030   00 00 00 01 00 02 02 22 02 23 00 3c ad 08 01 10   .......".#.<....
0040   08 74 00 01 00 0e 00 01 00 01 1c 39 cf 88 08 00   .t.........9....
0050   27 fe 8f 95 00 06 00 04 00 17 00 18 00 08 00 02   '...............
0060   00 00 00 19 00 0c 27 fe 8f 95 00 00 0e 10 00 00   ......'.........
0070   15 18                                             ..
```

Wireshark-Dump - Dekodierung

Payload-Länge ist in diesem Paket 60 Bytes: 00 3c (hex) = 60 dezimal

IP-Version ist 6: 6 (hex) = 0110 (binary, 4 Bits)

kein Traffic oder flow label (insgesamt 28 Bits)

```
0000                                                  60 00                  `.
0010   00 00 00 3c 11 01 fe 80 00 00 00 00 00 00 0a 00   ...<............
0020   27 ff fe fe 8f 95 ff 02 00 00 00 00 00 00 00 00   '...............
0030   00 00 00 01 00 02 02 22 02 23 00 3c ad 08 01 10   .......".#.<....
0040   08 74 00 01 00 0e 00 01 00 01 1c 39 cf 88 08 00   .t.........9....
0050   27 fe 8f 95 00 06 00 04 00 17 00 18 00 08 00 02   '...............
0060   00 00 00 19 00 0c 27 fe 8f 95 00 00 0e 10 00 00   ......'.........
0070   15 18                                             ..
```

Next header ist UDP!
11 hex = 17 dezimal → UDP

Wireshark-Dump - Dekodierung

Payload-Länge ist in diesem Paket 60 Bytes: 00 3c (hex) = 60 dezimal

IP-Version ist 6: 6 (hex) = 0110 (binary, 4 Bits)

kein Traffic oder flow label (insgesamt 28 Bits)

```
0000                                                  60 00                  `.
0010   00 00 00 3c 11 01 fe 80 00 00 00 00 00 00 0a 00   ...<............
0020   27 ff fe fe 8f 95 ff 02 00 00 00 00 00 00 00 00   '...............
0030   00 00 00 01 00 02 02 22 02 23 00 3c ad 08 01 10   .......".#.<....
0040   08 74 00 01 00 0e 00 01 00 01 1c 39 cf 88 08 00   .t.........9....
0050   27 fe 8f 95 00 06 00 04 00 17 00 18 00 08 00 02   '...............
0060   00 00 00 19 00 0c 27 fe 8f 95 00 00 0e 10 00 00   ......'.........
0070   15 18                                             ..
```

Next header ist UDP!
11 hex = 17 dezimal → UDP

Hop limit: 1
→ Nur link local, im internen Netz

Wireshark-Dump - Dekodierung

Payload-Länge ist in diesem Paket 60 Bytes: 00 3c (hex) = 60 dezimal

IP-Version ist 6: 6 (hex) = 0110 (binary, 4 Bits)

kein Traffic oder flow label (insgesamt 28 Bits)

```
0000                                                  60 00                  `.
0010   00 00 00 3c 11 01 fe 80 00 00 00 00 00 00 0a 00   ...<............
0020   27 ff fe fe 8f 95 ff 02 00 00 00 00 00 00 00 00   '...............
0030   00 00 00 01 00 02 02 22 02 23 00 3c ad 08 01 10   .......".#.<....
0040   08 74 00 01 00 0e 00 01 00 01 1c 39 cf 88 08 00   .t.........9....
0050   27 fe 8f 95 00 06 00 04 00 17 00 18 00 08 00 02   '...............
0060   00 00 00 19 00 0c 27 fe 8f 95 00 00 0e 10 00 00   ......'.........
0070   15 18                                             ..
```

Next header ist UDP! 11 hex = 17 dezimal → UDP

Hop limit: 1 → Nur link local, im internen Netz

Source Address: (link local !) fe80::a00:27ff:fefe:8f95

Wireshark-Dump - Dekodierung

Payload-Länge ist in diesem Paket 60 Bytes: 00 3c (hex) = 60 dezimal

IP-Version ist 6: 6 (hex) = 0110 (binary, 4 Bits)

kein Traffic oder flow label (insgesamt 28 Bits)

```
0000                                                  60 00                  `.
0010   00 00 00 3c 11 01 fe 80 00 00 00 00 00 00 0a 00   ...<............
0020   27 ff fe fe 8f 95 ff 02 00 00 00 00 00 00 00 00   '...............
0030   00 00 00 01 00 02 02 22 02 23 00 3c ad 08 01 10   .......".#.<....
0040   08 74 00 01 00 0e 00 01 00 01 1c 39 cf 88 08 00   .t.........9....
0050   27 fe 8f 95 00 06 00 04 00 17 00 18 00 08 00 02   '...............
0060   00 00 00 19 00 0c 27 fe 8f 95 00 00 0e 10 00 00   ......'.........
0070   15 18                                             ..
```

Dest. Address: ff02::1:2 (Multicast, alle DHCPv6-Server im lokalen Netzwerk)

Next header ist UDP! 11 hex = 17 dezimal → UDP

Hop limit: 1 → Nur link local, im internen Netz

Source Address: (link local !) fe80::a00:27ff:fefe:8f95

IPv6-Basisheader (3)

Weitere Änderungen im IPv6-Header

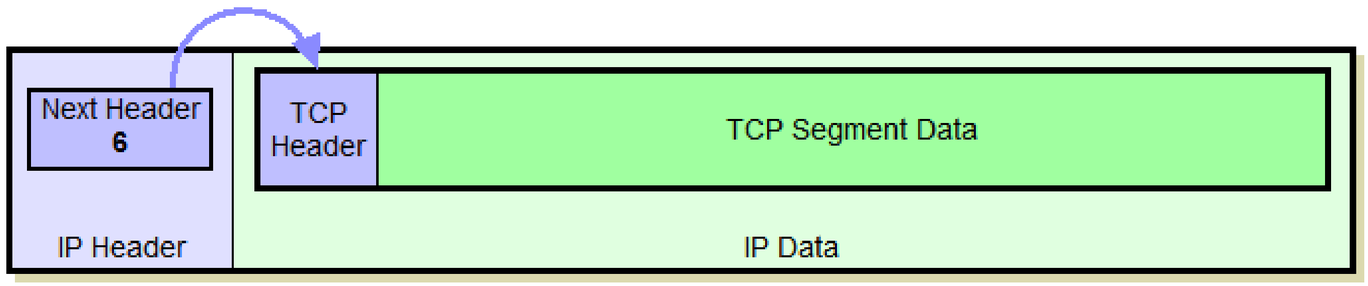
- Checksum: entfernt um die Verarbeitungszeit auf den Hops (Routern) zu minimieren
- Options: erlaubt, aber nicht im, sondern nach dem IPv6-Header, angezeigt durch das «Next Header»-Feld

ICMPv6: neue Version von ICMP

- zusätzliche Messagetypen: "Packet to Big»
- Multicastgruppen Funktionen

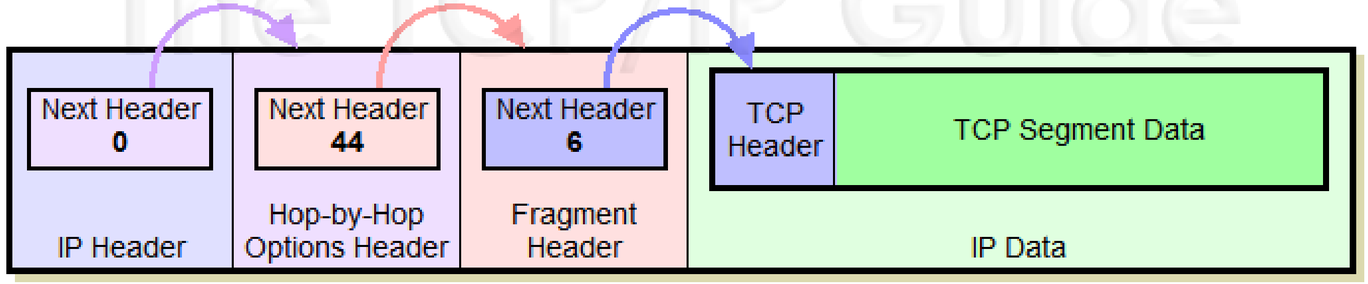
IPv6: NEXT HEADER-Feld

- NEXT HEADER-Feld: Identifiziert den Typ des nächsten Kopfdatenbereiches.
- Dieser kann entweder wie gewöhnlich auf ein Protokoll höherer Schicht zeigen, wie bspw. TCP (Typ 6) oder UDP (Typ 17), oder auf den ersten Erweiterungsheader



IPv6 Datagram With No Extension Headers Carrying TCP Segment

The TCP/IP Guide



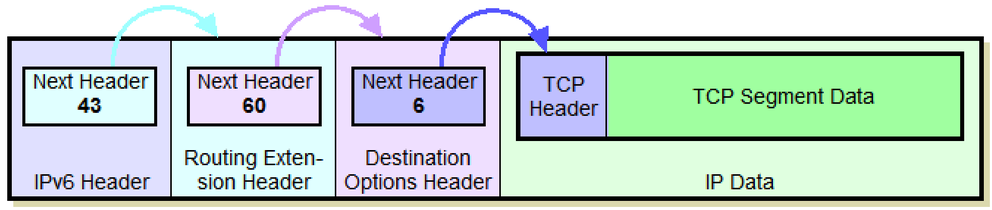
IPv6 Datagram With Two Extension Headers Carrying TCP Segment

Quelle: http://www.tcpipguide.com/free/t_IPv6DatagramExtensionHeaders-2.htm

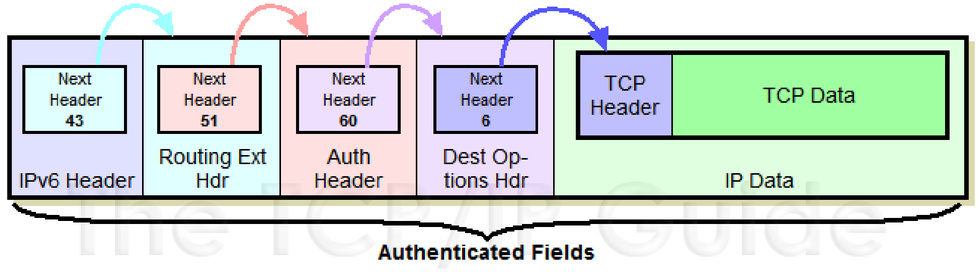
"Parsen" des IPv6 Headers

Basisheader hat feste Grösse von 40 Bytes
- NEXT HEADER-Feld im Basisheader definiert den Typ des Headers
- befindet ich am Ende des Basisheaders

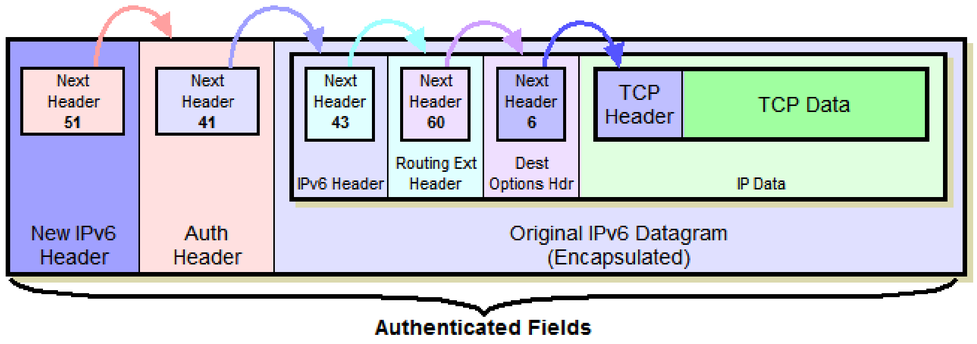
Einige Erweiterungsheader besitzen variable Grösse
- NEXT HEADER Feld in der Erweiterung definiert Typ
- HEADER LEN gibt die Grösse der Erweiterungsheader



Original IPv6 Datagram Format (Including Routing Extension Header and Destination-Specific Destination Options Extension Header)



IPv6 AH Datagram Format - IPSec Transport Mode



IPv6 AH Datagram Format - IPSec Tunnel Mode

Quelle: http://www.tcpipguide.com/free/t_IPSecAuthenticationHeaderAH-2.htm

Next Header

Zurzeit gibt es nur eine begrenzte Anzahl Header-Typen:

1. Hop-by-Hop Options-Header
2. Destination Options-Header
3. Routing-Header
4. Fragmentation-Header
5. Authentication-Header
6. Encapsulating Security Payload-Header
7. Destination Options-Header 2

Nummer	Extension Header
0	Hop-by-Hop Options
43	Routing
44	Fragment
51	Authentication Header (AH)
50	Encapsulation Security Payload (ESP)
60	Destination Options
135	Mobility
139	Host Identity Protocol
140	Shim6 Protocol
59	No Next Header (Ende eines Header-Stapels)

Next Header

Zurzeit gibt es nur eine begrenzte Anzahl Header-Typen:

1. Hop-by-Hop Options-Header
2. Destination Options-Header
3. Routing-Header
4. Fragmentation-Header
5. Authentication-Header
6. Encapsulating Security Payload-Header
7. Destination Options-Header 2

Next-Header-Feld soll aber auch – wie gehabt (wie bei IPv4) – Protokolle höherer Schichten anzeigen:

- ICMP (Typ 1)
- TCP (Typ 6)
- UDP (Typ 17)
- RDP (Typ 27)
- ...

Nummer	Extension Header
0	Hop-by-Hop Options
43	Routing
44	Fragment
51	Authentication Header (AH)
50	Encapsulation Security Payload (ESP)
60	Destination Options
135	Mobility
139	Host Identity Protocol
140	Shim6 Protocol
59	No Next Header (Ende eines Header-Stapels)
1	ICMP
6	TCP
17	UDP
...	...

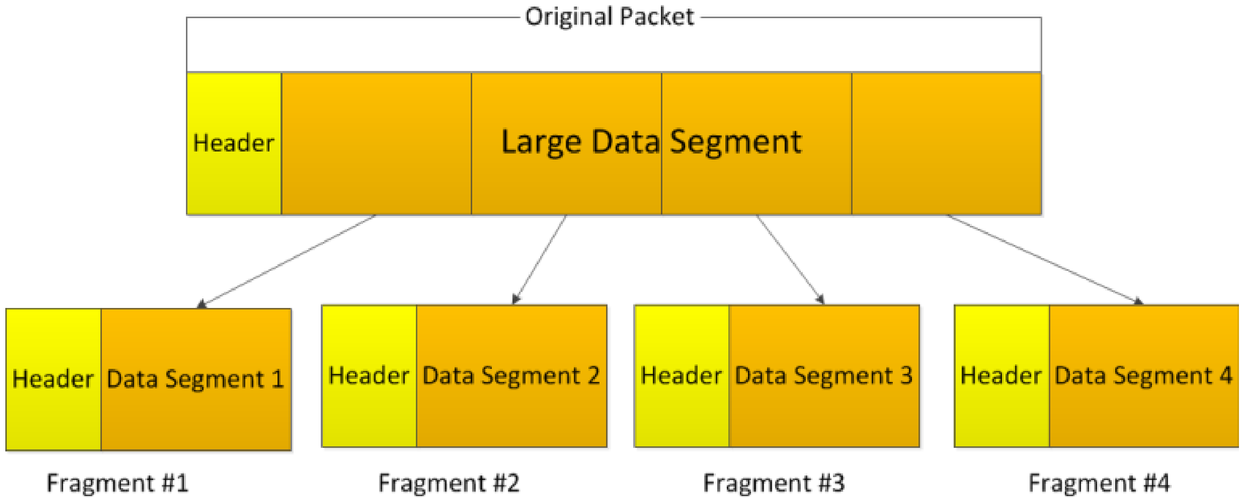
Verwendung mehrerer Header

- Effizient: Header nur so gross wie benötigt
- Flexibel: neue Header können für neue Features hinzugefügt werden
- Schrittweise Entwicklung

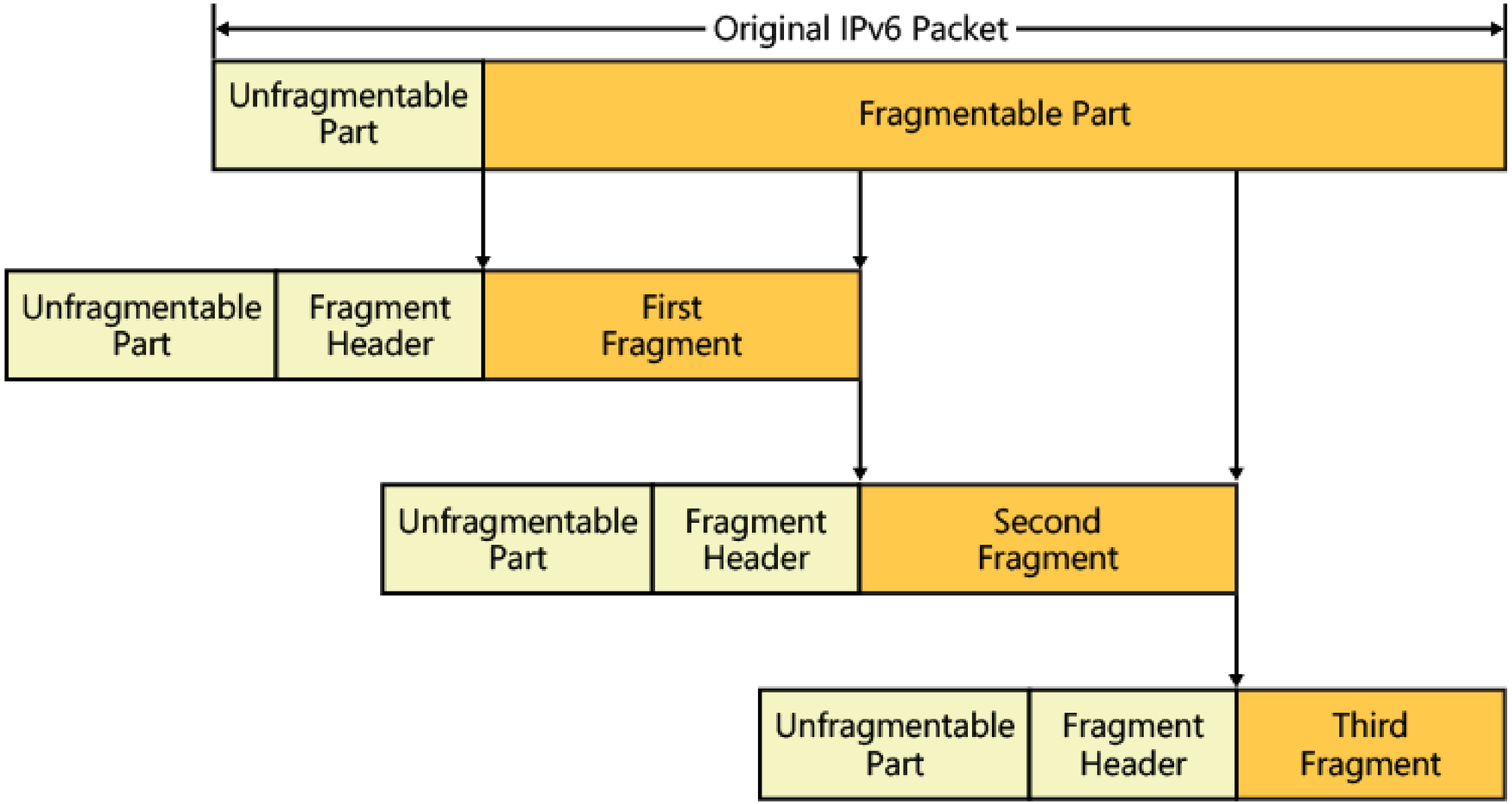
  neue Verarbeitungen für neue Features können in Testumgebung hinzugefügt werden; die anderen Router ignorieren diese Header

Fragmentierung (1)

- Fragmentierungsinformation wird in separatem Erweiterungsheader festgehalten
- Jedes Fragment besitzt Basisheader und eingefügten Fragementierungsheader
- Ganzes Datagramm, einschliesslich des Originalheaders kann fragmentiert sein



Fragmentierung IPv6 (2)



Quelle: https://www.microsoftpressstore.com/articles/article.aspx?p=2225063&seqNum=4

Fragmentierung und MTU

- IPv6-Quelle/Client (und nicht dazwischen liegende Router) sind für die Fragmentierung verantwortlich
  - Router droppen Pakete welche grösser als MTU sind
  - Quelle muss also Pakete fragmentieren, um Ziel zu erreichen
- Source bestimmt MTU des Pfades
  - ausschlaggebend ist kleinstes MTU zwischen Quelle und Ziel
  - fragmentiert Pakete, um in dieses MTU reinzupassen
- Einsatz von: Path MTU Discovery
  - Quelle sendet Testmitteilungen von verschiedener Grösse bis Ziel erreicht wird
  - Muss dynamisch gehandhabt werden: Pfad kann sich während Übermittlung ändern!

Übergangsmechanismen und Koexistenz mit IPv4

Informatik
27. Oktober 2025

Transition von IPv4 zu IPv6

- Nicht alle Clients und Router können gleichzeitig umgestellt und IPv6-fähig gemacht werden
  - keinen "flag day” (= Tag, an dem die gesamte Welt von IPv4 auf Version 6 (IPv6) umsteigt)
  - aber wie geht ein gemischter Betrieb von IPv4 und IPv6?
- Zwei Möglichkeiten
  - *Dual Stack*: Parallelbetrieb von IPv4 und IPv6
  - *Tunneling*: IPv6-Pakete werden über IPv4 zu einem IPv6-fähigen Router im Internet gesendet und dort weitergeleitet.

DO WE NEED AN IPV6 FLAG DAY?

JUNE 6, 2012 • BY BURT KALISKI • DOMAIN NAMES

In recent interviews about World IPv6 Launch I've been asked by several different people whether or not I think there needs to be some kind of a "Flag Day" on which the world all together switches from Internet Protocol version 4 (IPv4) to the version 6 (IPv6).

I don't think a flag day is needed. World IPv6 Launch is just the right thing.

It's worth looking at some previous flag-type days to get a better sense of why.

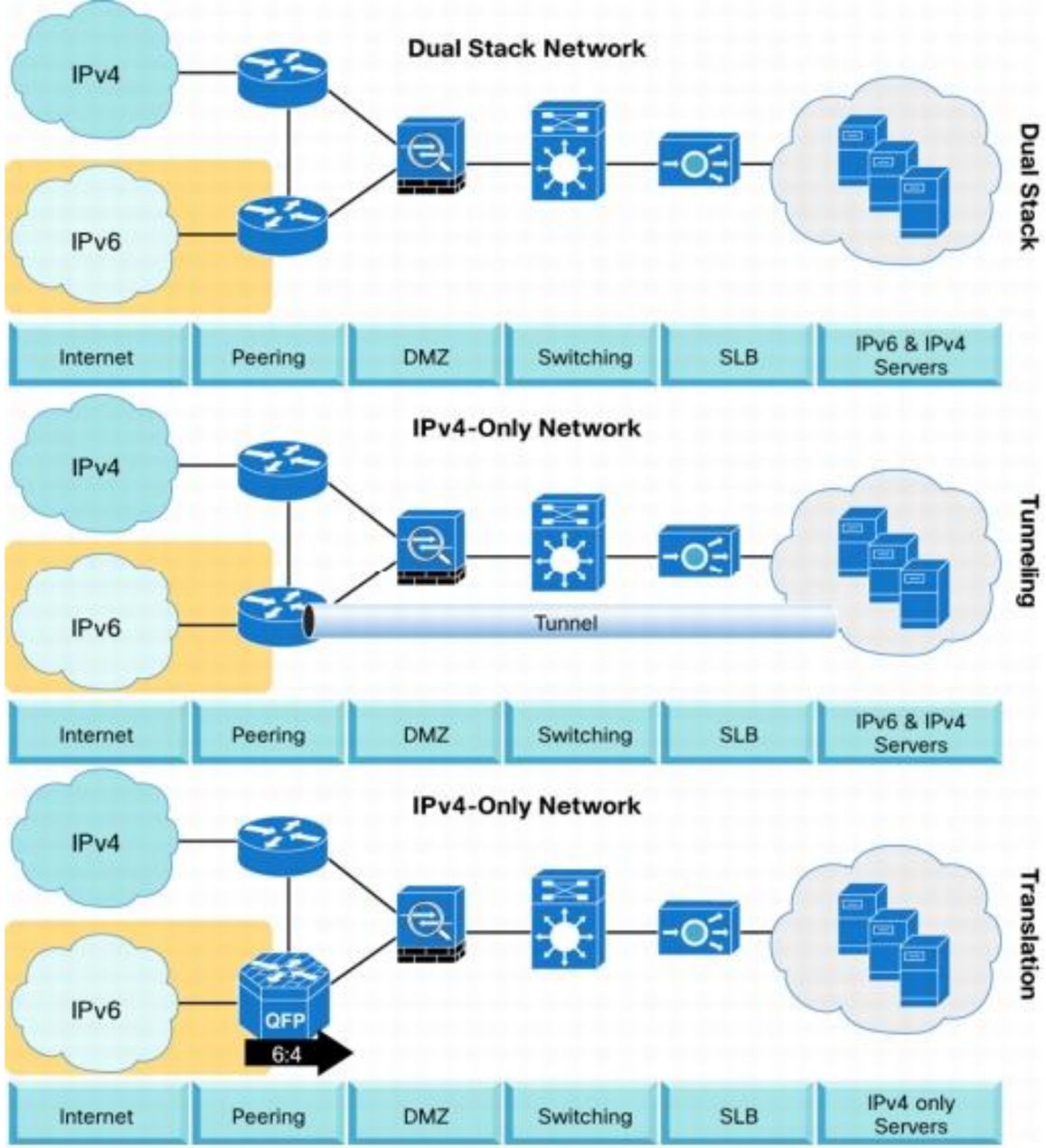
A classic example of a flag day converting entirely from one system to another is Högertrafikomläggningen or "H Day," September 3, 1967, when Sweden switched all road traffic from the left side to the right. After decades of debate and four years of planning, nation realigned both the roadways and the direction of vehicle headlights over a d to match its immediate neighbors.

Transition von IPv4 zu IPv6

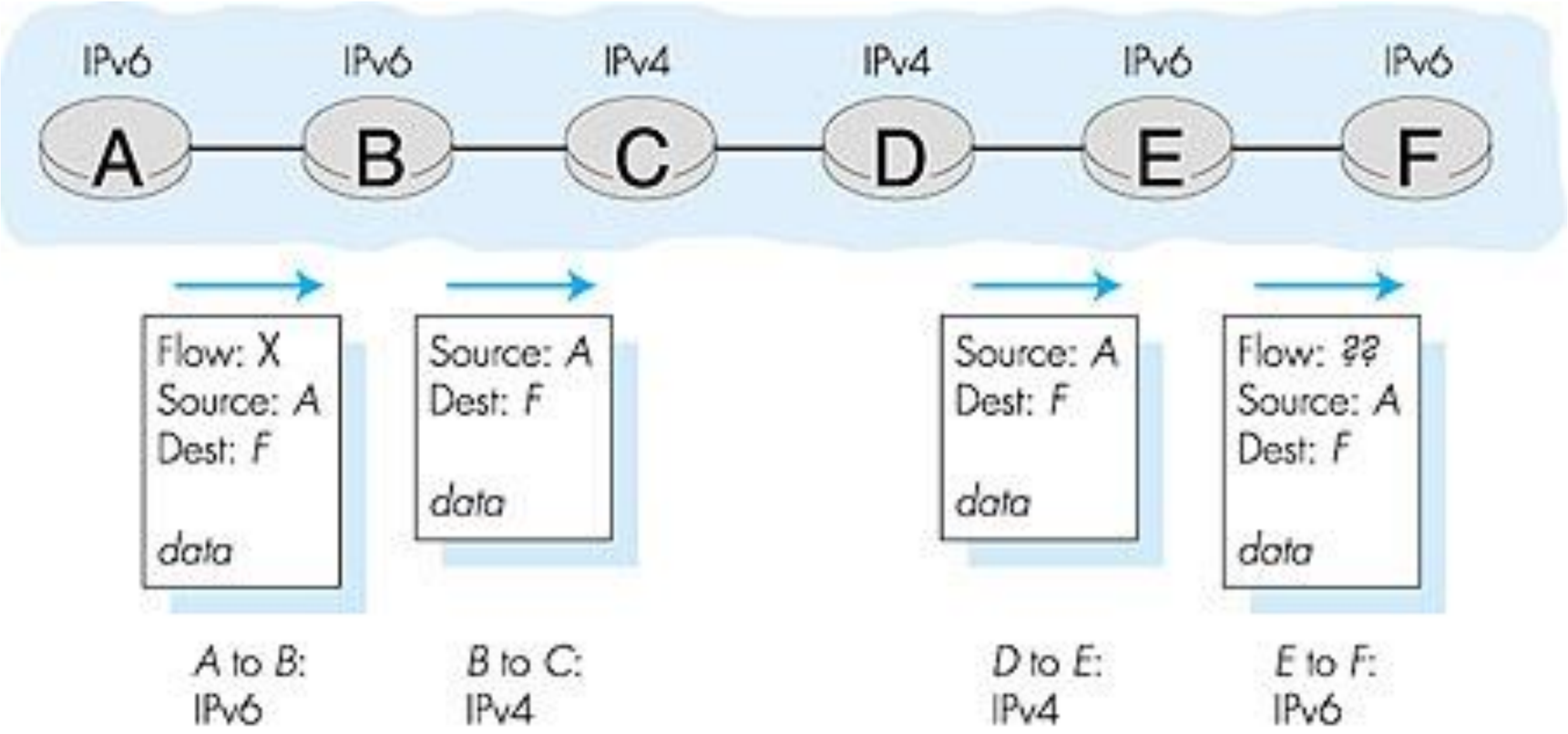
Verschiedene Szenarien denkbar

- *Dual-Stack:* Parallelbetrieb von IPv4 und IPv6. Längerfristig würde man dann auf IPv4 verzichten.
  Alle bestehenden Dienste sind weiterhin unter ihrer gewohnten IPv4-Adresse erreichbar. Nach und nach kann man bestehende Dienste per IPv6 erreichbar machen.
- *Tunneling*: IPv6-Packete werden beim Client oder Router in IPv4-Pakete verpackt, zu einem IPv6-fähigen Knoten im Internet getunnelt, dort entpackt und weitergeschickt.

Idee bei beiden Lösungen: Falls ein DNS-Server eine IPv6-Adresse bei der Namensauflösung zurückliefert (AAAA-Record) → IPv6; Falls *keine* IPv6-Adresse verfügbar → IPv4.

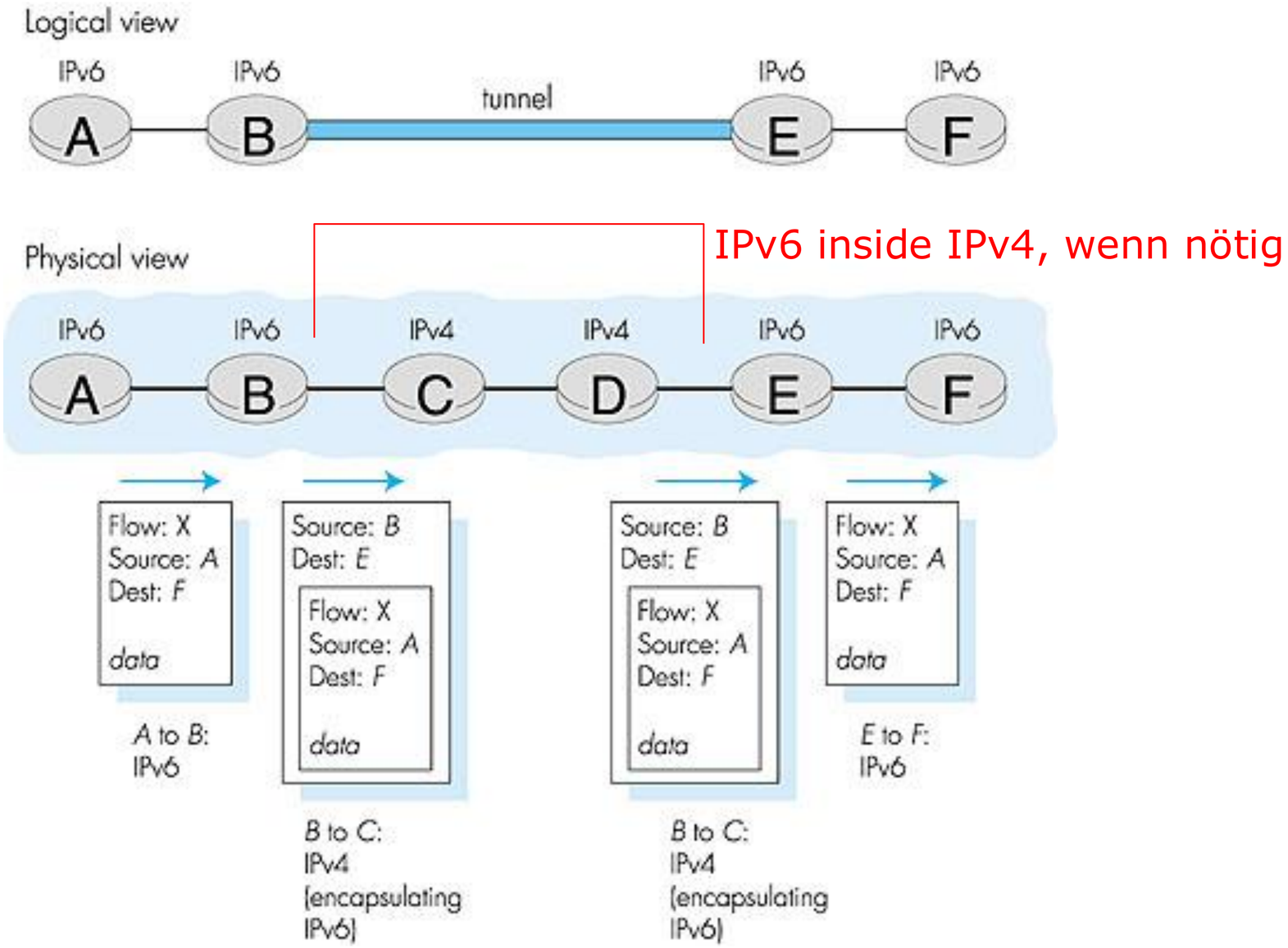


Tunneling



Quelle: www.heise.de

Tunneling



Quelle: www.heise.de

Links zu IPv6

- https://www.computerweekly.com/de/definition/IPv6-Internet-Protocol-Version-6 - kleine IPv6 Einführung
- https://test-ipv6.com/ - zum Testen der IPv6 Connectivity
- https://www.ultratools.com/tools/ipv6Info - IPv6 lookup
- https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Internetsicherheit/isi_lana_leitfaden_IPv6_pdf.pdf?__blob=publicationFile&v=1 – Leitfaden zur sicheren IPv6 Konfiguration
- https://www.space.net/~gert/RIPE/ - Umfangreiche Dokumentensammlung zu IPv6