Netze

Modul 5: IPv6 und Autokonfiguration

Prof. Dr. Hannes Tschofenig

30. Oktober 2025

Semesterplanung — Vorlesungen

Modul	Dozent	Datum	Thema
1	Rademacher	2. Oktober 2025	Einführung, OSI-Referenzmodell und Topologien
2	Rademacher	9. Oktober 2025	Übertragungsmedien und Verkabelung
3	Rademacher	16. Oktober 2025	Ethernet und WLAN
4	Tschofenig	23. Oktober 2025	IPv4, Subnetze, ARP, ICMP
5	Tschofenig	30. Oktober 2025	IPv6 und Autokonfiguration
6	Tschofenig	6. November 2025	Netzwerksegmentierung
7	Tschofenig	13. November 2025	Routing
8	Rademacher	20. November 2025	Transportschicht und UDP
9	Rademacher	27. November 2025	TCP
10	Rademacher	4. Dezember 2025	DNS und HTTP 1
11	Tschofenig	11. Dezember 2025	HTTP 2 und QUIC
12	Tschofenig	18. Dezember 2025	TLS und VPN
/	/	8. Januar 2026	`Bei Bedarf / TBA`
13	Tschofenig	15. Januar 2026	Messaging
14	Rademacher	22. Januar 2026	Moderne Netzstrukturen

Semesterplanung — Übungen und Praktika

ID	KW	Art	Thema
	40	/	/
UE-1	41	Übung	Topologien und OSI
UE-2	42	Übung	Übertragungen bspw. Kabel
P-1	43	Praktikum	Laboreinführung und Netzwerktools
S-1	44	Video	IPv4
P-2	45	Praktikum	Adressierung
P-3	46	Praktikum	IPv4 und Autokonfiguration
P-4	47	Praktikum	IPv6 und Autokonfiguration
P-5	48	Praktikum	Routing
P-6	49	Praktikum	Switching
P-7	50	Praktikum	Transportprotokolle
S-2	51	Experiment	VPN
S-2	52	Experiment	VPN
	2	/	/
P-8	3	Praktikum	DNS
P-9	4	Praktikum	Webkommunikation

UE - Übung laut Stundenplan in den Seminarräumen

P - Praktikum in C055

S - Selbststudium KEINE Präsenz

Praktikum: Erinnerung

Die Auslastung in den Gruppen ist noch immer sehr unterschiedlich.

Tag	Zeit
Montag	10:45 – 12:15
Donnerstag	15:15 – 16:45

Hinweis

- Bitte über **LEA** in eine der beiden Gruppen ummelden, falls noch nicht geschehen.
- Die **Donnerstagsgruppe** ist speziell für Studierende des **Bachelorstudiengangs Informatik** vorgesehen.
- **Nächste Woche** werden die Arbeiten **im Labor weitergeführt**.
  Dort gibt es **noch freie Kapazitäten** – nutzen Sie diese Gelegenheit!

Motivation

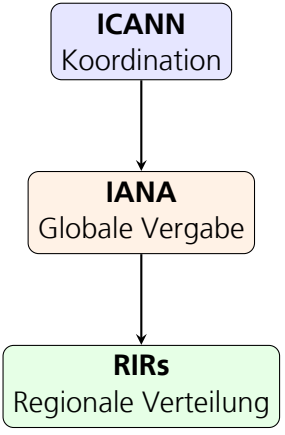
Adressknappheit im IPv4-Netz

- Der IPv4-Adressraum umfasst nur **32 Bit** – rund 4,3 Milliarden Adressen.
- Mit dem rasanten Wachstum des Internets wurde dieser Raum zunehmend knapp.
- Die Einführung von **CIDR (RFC 1519)** verlangsamte die Erschöpfung, konnte sie jedoch nicht dauerhaft verhindern.
- Als Übergangslösung entstand der gemeinsame Adressgebrauch mittels:
  - **Network Address Translation (NAT)** [11]
  - **Network Address Port Translation (NAPT)** [12]

Organisation der IP-Adressverwaltung

Globale Hierarchie der IP-Adressvergabe:

- **ICANN**: Koordiniert die globale Verwaltung von IP-Adressen, DNS und Protokollnummern.
- **IANA**: Operativer Teil von ICANN; verwaltet den globalen IP-Adresspool und weist große Blöcke an die RIRs zu.
- **RIRs**: Vergeben IP-Adressen regional an Provider und Organisationen.
  Beispiele: **RIPE NCC (Europa)**, **ARIN (Nordamerika)**, **APNIC**, **LACNIC**, **AFRINIC**.

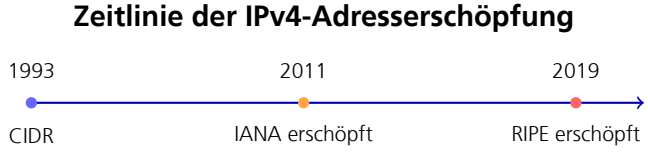


ICANN beaufsichtigt IANA → RIRs verteilen Adressen regional.

Abkürzungsverzeichnis:

ICANN	Internet Corporation for Assigned Names and Numbers
IANA	Internet Assigned Numbers Authority
RIR	Regional Internet Registry
RIPE NCC	Réseaux IP Européens Network Coordination Centre
ARIN	American Registry for Internet Numbers
APNIC	Asia-Pacific Network Information Centre
LACNIC	Latin American and Caribbean Internet Addresses Registry
AFRINIC	African Network Information Centre

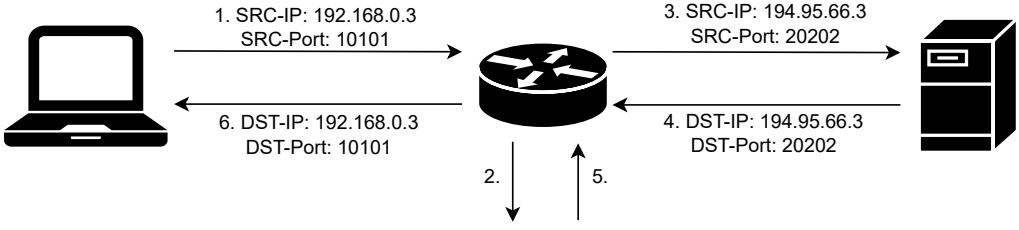
IPv4-Adresserschöpfung



Historische Entwicklung:

- **1993:** Einführung von CIDR zur effizienteren Nutzung des Adressraums.
- **3. Februar 2011: IANA** vergibt ihre letzten fünf /8-Blöcke an die RIRs → globaler Pool leer.
- **2011–2019:** Regionale Internet-Registries (RIRs) geben ihre letzten Bestände nach strengen Vergaberegeln aus.
- **25. November 2019: RIPE NCC (Europa)** meldet vollständige Erschöpfung.

Network Address Port Translation



LAN-Port	LAN-IP	WAN-Port
10101	192.168.0.3	20202
...	...	...

NAT-Tabelle

Auswirkungen von NAT/NAPT

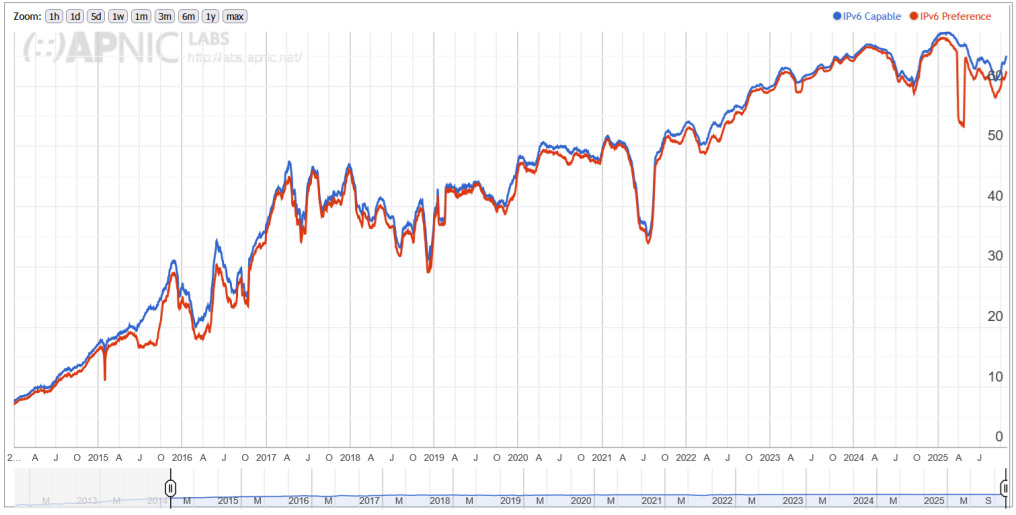
- Obwohl NAT und NAPT technisch einfache Konzepte sind, hatten sie weitreichende Folgen für die Internetarchitektur: Die **Ende-zu-Ende-Erreichbarkeit** ging verloren.
- **NAT-Bindings** sind temporär und müssen regelmäßig erneuert werden, um bestehende Verbindungen aufrechtzuerhalten.
- Eine Studie von 2010 mit 34 Home-Gateway-Modellen [15] zeigte deutliche Unterschiede zwischen UDP- und TCP-Verbindungen: UDP-Bindings liefen frühestens nach 30 s ab, TCP-Bindings nach durchschnittlich 239 s.
- Ursprünglich als **Übergangslösung** entwickelt, sind NATs bis heute allgegenwärtig.
- Eine kurze historische Einordnung bietet Lixia Zhang [30].

Hinweis: NATs erfordern für viele moderne Anwendungen (z. B. VoIP, P2P, WebRTC, Instant Messaging, Spiele) zusätzliche **NAT-Traversal-Mechanismen** wie STUN [25], TURN [26]oder ICE [20].

IPv6 – Überblick

- Ursprünglich standardisiert durch RFC 2460 (1998) [16], inzwischen durch RFC 8200 (2017) [9] ersetzt – IPv6 ist heute als **Internet Standard (STD 86)** festgelegt.
- Zentrale Unterschiede gegenüber IPv4:
  - **Vergrößertes Adressfeld:** 128 Bit statt 32 Bit
  - **Vereinfachtes Header-Format** (keine Prüfsumme, keine Fragmentierung durch Router)
  - **Bessere Unterstützung** von Erweiterungs-Headern und Optionen
  - Einführung eines **Flow Labels** zur Kennzeichnung von Datenströmen
- Die **Sicherheitsarchitektur IPsec** wurde ursprünglich in RFC 2401 [3] beschrieben und durch RFC 4301 [27] aktualisiert; sie ist auch auf IPv4 anwendbar.

Verwendung von IPv6 in Deutschland



Datenquelle: APNIC Labs – IPv6 Measurements (Deutschland),

Zugriff: Oktober 2025

IPv6 Adressen

IPv6 – Adressen (1/2)

- Länge einer IPv6-Adresse: **128 Bit**
- Darstellung in **acht Gruppen** zu jeweils **16 Bit** (vier Hexadezimalziffern), getrennt durch Doppelpunkte (`:`)

Beispiel

```
2001:0db8:0000:0000:0000:0005:eec1:0008
```

- Jede Gruppe entspricht **16 Bit** bzw. vier Hexadezimalziffern.
- IPv6-Adressen identifizieren (wie IPv4-Adressen) **einzelne Netzwerk-Interfaces**.

Beispielpräfix `2001:db8::/32` stammt aus RFC 3849 [17] und ist für Dokumentationszwecke reserviert.

IPv6 – Adressen (2/2): Komprimierung

- Eine zusammenhängende Folge von **Nullgruppen** kann ausgelassen und durch `::` ersetzt werden.
- Führende Nullen innerhalb einer Gruppe dürfen ebenfalls weggelassen werden.
- Hinweis: Das Kürzel `::` darf pro Adresse nur **einmal** verwendet werden.

Beispiel

```
2001:0db8:0000:0000:0000:0005:eec1:0008
2001:0db8::0005:eec1:0008
2001:db8::5:eec1:8
```

IPv6 – Präfixnotation (1/2)

- IPv6 verwendet **Adresspräfixe**, die mit einem Schrägstrich (/) angegeben werden.
- Beispiel: `2001:db8::/32` bezeichnet den Adressbereich eines Netzes.
- **n** gibt die Länge des Netzpräfixes an – also die Anzahl der führenden Bits, die das Netz definieren.
- Der Präfixanteil wird in der Regel vom Provider oder einer übergeordneten Instanz festgelegt.

IPv6 – Präfixe und Adressbereiche (2/2)

- Provider erhalten üblicherweise größere Präfixe und teilen diese in kleinere, dem Bedarf angepasste Subnetze für ihre Kund*innen auf.

Beispiel für Präfixhierarchie	
`2001:db8::/32`	Provider-Netz
`2001:db8:1::/48`	Kundennetz
`2001:db8:1:1::/64`	Subnetz im Kundennetz

Hinweis: /64 ist die Standardgröße für ein einzelnes IPv6-Subnetz.

IPv6 – Adresstypen

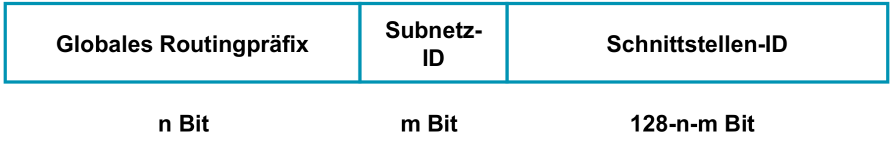
IPv6 unterscheidet drei grundlegende Adresstypen:

- **Unicast:** 1:1-Kommunikation – eindeutig einer einzelnen Schnittstelle zugeordnet.
- **Multicast:** 1:n-Kommunikation – eine Nachricht wird an mehrere Empfänger gesendet.
- **Anycast:** 1:1 von n – Zustellung an die **nächstgelegene** (topologisch nächste) Schnittstelle.

Typ	Kommunikation	Beispiel / Hinweis
Unicast	$1 \rightarrow 1$	`2001:db8::1` (ein Host)
Multicast	$1 \rightarrow n$	`ff02::1` (alle Knoten), `ff02::2` (alle Router)
Anycast	$1 \rightarrow 1$ (nächstes)	`2001:db8::1:1` (z. B. DNS-Server-Cluster)
Broadcast	entfällt	durch Multicast ersetzt

Hinweis: **Broadcast** wurde in IPv6 abgeschafft, um Netzlast zu reduzieren; **Multicast** übernimmt diese Funktion gezielter.

Globale Unicast-Adressen (1/2)



- Allgemeine Struktur einer globalen Unicast-Adresse nach RFC 4291 [8] (aktualisiert durch RFC 8200 [9]).
  - Das **globale Routingpräfix** kennzeichnet den Adressbereich einer Organisation oder eines Providers.
  - Die **Subnetz-ID** ermöglicht die Bildung mehrerer hierarchischer Subnetze innerhalb der Organisation.
  - Die **Schnittstellen-ID** identifiziert eine einzelne Netzwerkschnittstelle innerhalb eines Subnetzes.
- IPv6 verwendet in der Regel **64 Bit lange Schnittstellen-IDs** (vgl. RFC 7421).
- Global gültige IPv6-Adressen beginnen mit dem Präfix `2000::/3` – d. h. sie starten mit den Bits `001`. Ein großer Teil der heute vergebenen Adressräume liegt im Bereich `2001::/16`.

IPv6 ohne 64-Bit Interface Identifier

- In IPv6-Netzen werden **Interface-Identifier** standardmäßig mit einer Länge von **64 Bit** verwendet (RFC 4291, RFC 7421 [5]).
- Es gibt jedoch einige **Ausnahmen**, in denen kürzere Präfixe genutzt werden:
  - **Punkt-zu-Punkt-Verbindungen** (z. B. Routerlinks): Verwendung von `/127` laut RFC 6164 [21].
  - **Loopback-Adresse:** `::1/128`
  - **Spezielle Infrastruktur- oder Management-Netze:** teilweise `/126` oder `/120` (nicht RFC-standardisiert, aber in der Praxis anzutreffen)
  - **Multicast- und Anycast-Adressen:** besitzen keinen klassischen Interface-Identifier.
- Der 64-Bit-Identifier ist erforderlich, wenn **SLAAC (Stateless Address Autoconfiguration)** eingesetzt wird.

Hinweis: RFC 7421 bestätigt die 64-Bit-Grenze für Unicast-Adressen, erlaubt aber abweichende Präfixlängen für spezielle Einsatzzwecke.

Generierung der Schnittstellen-ID

- Wie kann eine eindeutige Schnittstellen-ID automatisch erzeugt werden?
  - Abgeleitet aus der MAC-Adresse (EUI-64, Appendix A in [8]) → nächste Folie
  - Basierend auf einem Zufallsverfahren (Privacy Extensions [23])
  - Durch ein kryptografisches Verfahren (Cryptographically Generated Addresses - CGA) [2] für die Verwendung in SEcure Neighbor Discovery [19])

Historische Entwicklung

- Früher: Bildung der Schnittstellen-ID nach dem **EUI-64-Verfahren** aus der MAC-Adresse
- Heute: Das EUI-64-Verfahren erzeugt in der Regel eindeutige Interface-IDs, ist aber aus **Datenschutzgründen** in Geräten von Endverbrauchern (Laptops, Smartphones) kaum noch gebräuchlich (vgl. RFC 4941 [23] und RFC 7721 [7]). Stattdessen werden die Privacy Extensions benutzt.
- EUI-64-Verfahren wird aber noch in **IoT-Geräte** benutzt, z.B. in IPv6 over 6LoWPAN [29].

Ableitung der Schnittstellen-ID aus der MAC-Adresse (EUI-64)

- Bei der Bildung einer IPv6-Adresse kann eine 48-Bit-MAC-Adresse (EUI-48) in eine 64-Bit-Adresse (EUI-64) umgewandelt werden.
- EUI steht für **Extended Unique Identifier**.

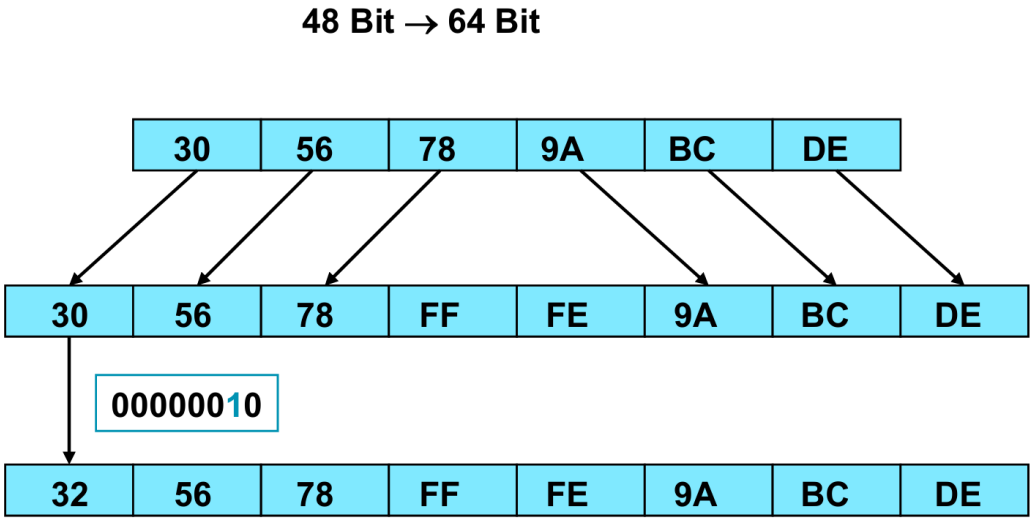
Verfahren

1. Einfügen der Bytes `FF` und `FE` in der Mitte der MAC-Adresse (nach dem 24. Bit).
2. Invertieren des **U/L-Bits** (2. Bit im ersten Oktett - Least Significant Order):
   - $0 \rightarrow 1$: Adresse wurde lokal generiert
   - $1 \rightarrow 0$: Adresse ist global eindeutig

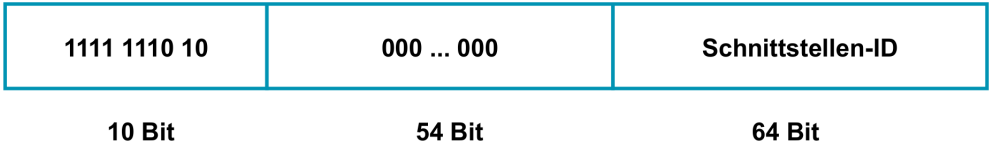
Beispiel

MAC: `00-1A-2B-3C-4D-5E` $\rightarrow$ EUI-64: `021A:2BFF:FE3C:4D5E`

Beispiel


48 Bit → 64 Bit
30
56
78
9A
BC
DE
30
56
78
FF
FE
9A
BC
DE
00000010
32
56
78
FF
FE
9A
BC
DE

Link-Lokale Unicast-Adressen (Link-Local Addresses)



- Beginnen mit dem Präfix `FE80::/10` und werden automatisch auf jedem IPv6-fähigen Interface konfiguriert.
- Werden z. B. für **Neighbor Discovery** (RFC 4861 [28]) und Routing-Protokolle (z. B. OSPFv3, RIPng) verwendet.
- Gültig nur innerhalb eines **lokalen Links**; sie werden **nicht durch Router weitergeleitet**.

Vergleich

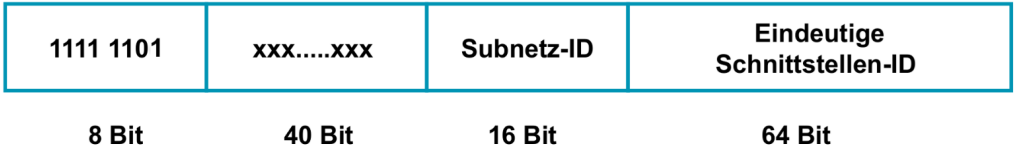
Entspricht den automatisch vergebenen IPv4-Adressen im Bereich `169.254.0.0/16` [6].

Bildung einer Link-Lokal gültigen Unicast-Adresse

- Präfix für ‘Link Local’ Adresse: FE80::/10, und
- 64 Bit eindeutige Schnittstellen ID aus MAC-Adresse nach EUI-64

1111 1110 10	000 ... 000	Eindeutige Schnittstellen-ID
10 Bit	54 Bit	64 Bit

Unique Local Addresses (ULAs)



- Funktional vergleichbar mit privaten IPv4-Adressen; werden in **privaten Netzwerken (Intranets)** eingesetzt.
- ULAs sind in RFC 4193 [13] definiert.
- Beginnen mit dem Präfix `FD00::/8` (die oberen 8 Bit sind `1111 1101`).
- Die **Global ID** (40 Bit) wird zufällig generiert, um eine hohe Wahrscheinlichkeit der Eindeutigkeit zu gewährleisten.
- Die **Subnetz-ID** (16 Bit) identifiziert ein Subnetz innerhalb der Organisation („Site“).

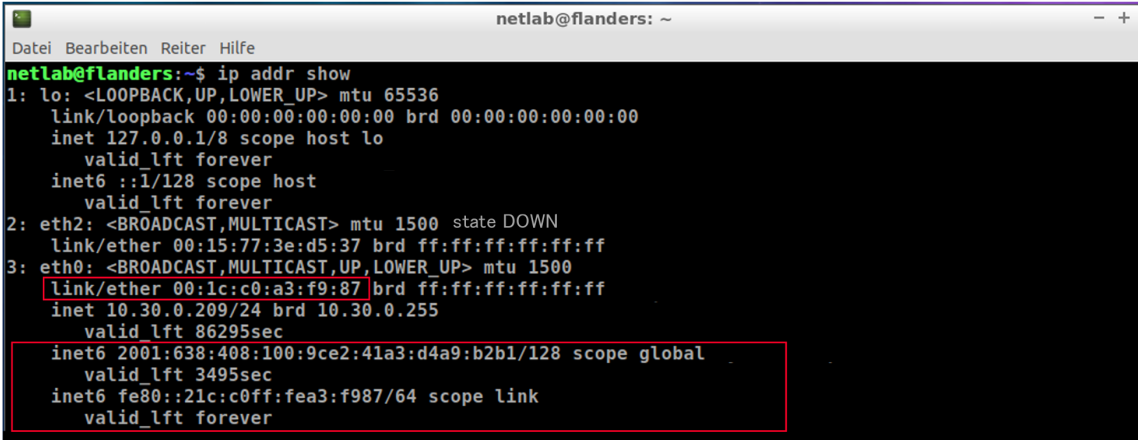
Beispiel

```
fd12:3456:789a:1::1/64
```

Einsatz von Unique Local Addresses (ULAs) heute

- **ULAs** (RFC 4193 [13]) werden für interne IPv6-Kommunikation verwendet – sie sind nicht global routbar.
- Hauptanwendungsbereiche:
  - Private oder isolierte Netzwerke (Labore, Testumgebungen, Campusnetze)
  - Kombination mit globalen Adressen (Dual-Adressierung für interne + externe Kommunikation)
  - Heimnetze mit stabilen internen Adressen, unabhängig vom ISP-Präfix
  - IoT- oder industrielle Steuerungsnetze ohne Internetzugang
- Vorteile:
  - Stabiler interner Adressraum, keine Abhängigkeit vom Provider
  - Keine Weiterleitung im Internet

Beispiel aus dem Labor



Spezielle IPv6-Adressen

Nicht spezifizierte Adresse (`::`)

- Zeigt an, dass einem Interface noch keine gültige Adresse zugewiesen ist.
- Wird nur als **Quelladresse** verwendet, z. B. bei der automatischen Adresskonfiguration.
- Wird niemals dauerhaft zugewiesen oder als Zieladresse verwendet.
- Entspricht in IPv4 der Adresse `0.0.0.0`.

Loopback-Adresse (`::1`)

- Kennzeichnet die **Loopback-Schnittstelle** – Pakete werden an das eigene Gerät gesendet.
- Wird nicht über Router weitergeleitet.
- Entspricht der IPv4-Adresse `127.0.0.1`.

Multicast-Adressen

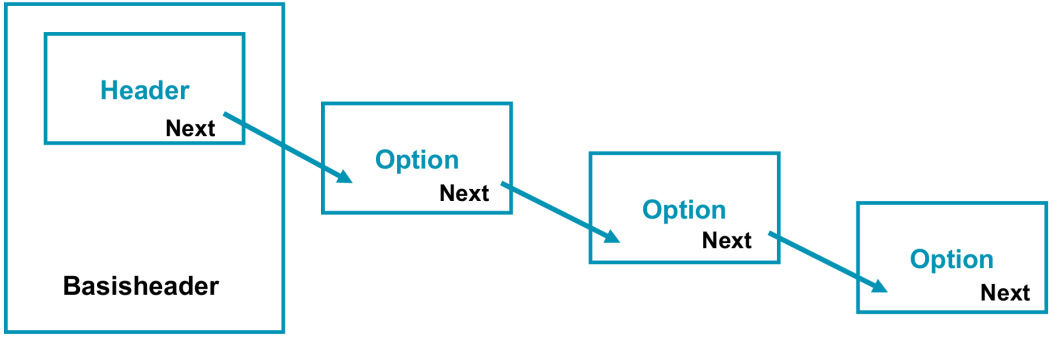
- Beginnen mit dem Präfix `FF00::/8`.
- Die Felder **Flags** und **Scope** enthalten Informationen über Aufbau und Geltungsbereich der Adresse.
- Die **Group ID** identifiziert die jeweilige Multicast-Gruppe.
- Weitere Details in RFC 4291 [8] und RFC 3306 [14].
- Beispiele:
  - `FF02::1` – *All Nodes* (vergleichbar mit IPv4-Broadcast `255.255.255.255`)
  - `FF02::2` – *All Routers*

Hinweis: IPv6 ersetzt Broadcasts vollständig durch Multicast (z. B. für Neighbor Discovery und Routing-Protokolle).

IPv6 Protokoll

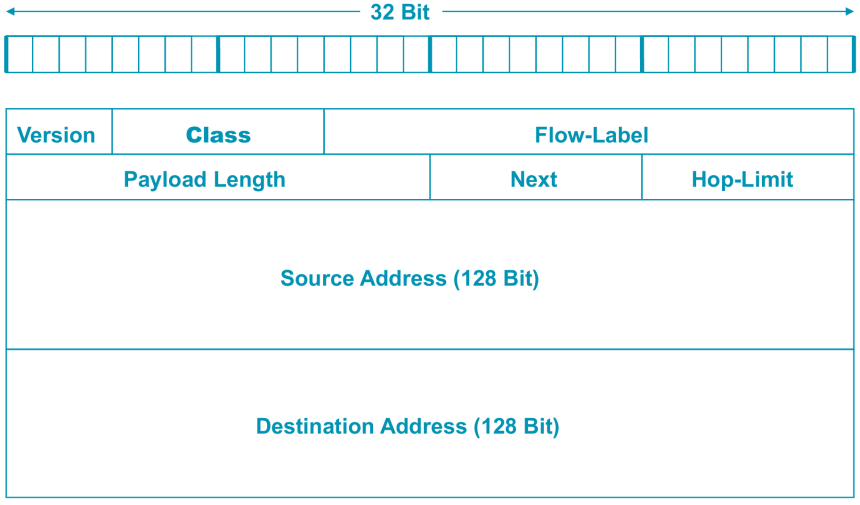
Prinzipieller Aufbau des IPv6-Headers

- Der **IPv6-Basis-Header** besitzt eine feste Länge von **40 Byte** und enthält nur die wichtigsten Paketinformationen.
- Zusätzliche Optionen werden in **verketteten Erweiterungs-Headern** abgelegt, die nur von den Komponenten ausgewertet werden, für die sie relevant sind.
- Die Reihenfolge der Erweiterungs-Header ist **standardisiert** (vgl. RFC 8200).



Beispielhafte Reihenfolge: IPv6 Header → Hop-by-Hop → Routing → Fragment → Destination → TCP/UDP.

Aufbau des Basisheaders (1)



Aufbau des IPv6-Basis-Headers (2)

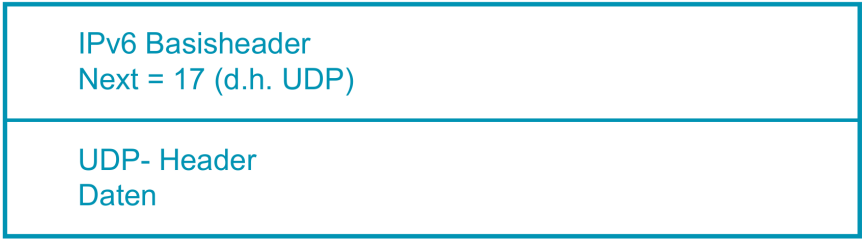
- Der IPv6-Basis-Header besitzt eine **feste Länge von 40 Byte**.
- **Version:** Immer der Wert 6.
- **Traffic Class:** Kennzeichnet Priorität bzw. Dienstgüte (Quality of Service).
- **Flow Label:** Eindeutige Kennung für Datenströme (z. B. Echtzeitverkehr).
- **Payload Length:** Länge der Nutzdaten in Byte (max. 65 535 Byte, größere Werte über „Jumbo Payload“ gemäß RFC 2675 [4]).
- **Next Header:** Typ des nächsten Headers (z. B. Erweiterungs-Header oder Transportprotokoll wie TCP).
- **Hop Limit:** Maximale Anzahl an Routern, die das Paket durchlaufen darf (entspricht **TTL** in IPv4).
- **Quell- und Zieladresse:** Jeweils 128 Bit lang.

Hinweis: Der IPv6-Basis-Header ersetzt mehrere Felder des IPv4-Headers und ist dadurch deutlich einfacher aufgebaut.

Kennzeichnung verschiedener Headertypen im NEXT Feld

Nummer	Typ des Headers
0	Hop-by-Hop-Optionen
60	Ziel-Optionen
43	Header für Routing
44	Header für Fragmentierung
51	Header für Authentisierung
50	Header für Verschlüsselung
59	Kein weiterer Header
6	TCP
17	UDP

Beispiel für ein minimales Paket bei IPv6

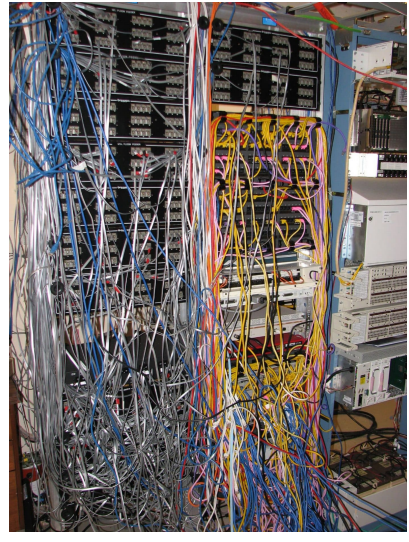


Beispiel für ein komplexes Paket bei IPv6

Autokonfiguration

Motivation für Autokonfiguration

- Netzwerkgeräte benötigen eine **minimale Grundkonfiguration**:
  - IP-Adresse
  - Standard-Router (Default Gateway)
  - DNS-Server-Adresse
- Eine manuelle Konfiguration ist insbesondere in **mobilen Szenarien** unpraktikabel.
- Geräte ohne Benutzeroberfläche (z. B. im **Internet of Things**) können häufig nicht manuell konfiguriert werden.
- **Ziel der Autokonfiguration:** Automatische Vergabe aller notwendigen Netzparameter für eine reibungslose Integration in das aktuelle Netz.



Verfahren zur Autokonfiguration

IPv4

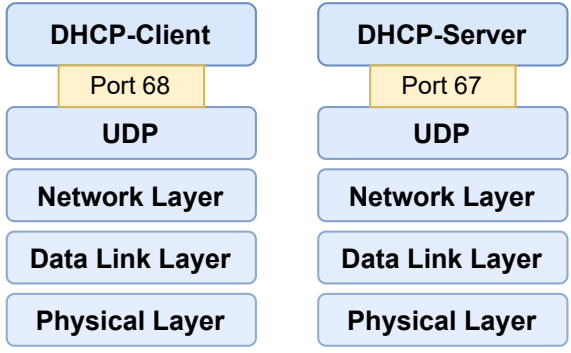
- **Bootstrap Protocol (BOOTP)** [1]
  - Veraltet, Vorgänger von DHCP.
- **Dynamic Host Configuration Protocol (DHCPv4)** [10]
  - Standard in nahezu allen IPv4-Netzen.
  - DHCPv4-Server weist IP-Adressen und weitere Netzparameter zu.
  - **Stateful:** Zentrale Adressverwaltung durch Server.

IPv6

- **Stateless Address Autoconfiguration (SLAAC)** [24]
  - Host generiert Adresse selbst (Präfix + Interface-ID).
  - Kein zentraler Server erforderlich.
- **DHCPv6** [22]
  - Stateful Pendant zu DHCPv4.
  - Kombination häufig: **SLAAC** für Adressen, **DHCPv6** für DNS (vgl. RFC 8106 [18]).

DHCPv4 – Übersicht

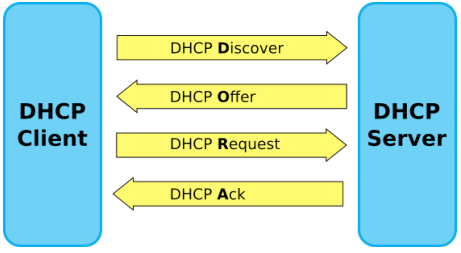
- **DHCP (Dynamic Host Configuration Protocol)** ist ein **Client-Server Protokoll** zur automatischen Netzwerkkonfiguration [10].
- Ein DHCP-Server liefert auf Anfrage mindestens folgende Konfigurationsparameter:
  - IP-Adresse,
  - Subnetzmaske (**Option 1**),
  - Lease Time (**Option 51**),
  - DHCP-Server-Identifier (**Option 54**),
  - Standard-Gateway-Adresse (**Option 3**)[a]
- Adressen werden aus einem vordefinierten **Adresspool** dynamisch vergeben.
- Weitere Optionen siehe IANA.



DHCP nutzt **UDP-Port 67/68** und arbeitet auf der Anwendungsschicht.

[a] Option 3 ist nicht verpflichtend, wird aber häufig mitgeschickt.

Ablauf von DHCPv4 (DORA)



- **DHCPDISCOVER:** Wird als Broadcast (`255.255.255.255`) mit der Absenderadresse `0.0.0.0` gesendet, um verfügbare DHCP-Server zu finden.
- **DHCPOFFER:** Enthält die vorgeschlagene IP-Adresse, Lease Time, Subnetzmaske und die Adresse des anbietenden DHCP-Servers.
- **DHCPREQUEST:** Der Client wählt eines der Angebote aus und fordert diese Adresse (und ggf. weitere Parameter) vom DHCP-Server an.
- **DHCPACK:** Der Server bestätigt die Zuweisung – der Client kann die Adresse nun verwenden.

Ablauf der Stateless Address Autoconfiguration (SLAAC)

1. **Lokale Adressbildung**
   Der Host erzeugt eine Link-Local-Adresse der Form `FE80::<Interface-ID>` und überprüft sie mit der **Duplicate Address Detection (DAD)** (Abschnitt 5.4 in [24]).
2. **Erkennung erreichbarer Router**
   Router senden regelmäßig **Router Advertisements (RA)** an `FF02::1` („All Nodes“). Der Host kann zusätzlich eine **Router Solicitation (RS)** an `FF02::2` („All Routers“) senden, um die Antwort zu beschleunigen. Die Quelladresse des empfangenen RAs wird als Standardrouter gespeichert.
3. **Empfang von Präfixinformationen**
   Über Optionen im RA, insbesondere die **Prefix Information Option (PIO)**, erhält der Host Netzpräfixe und Flags.

Ergebnis

Der Host kombiniert ein empfangenes Präfix mit seiner Interface-ID und bildet daraus eine globale IPv6-Adresse.

SLAAC – Beispiel und Einschränkungen

Beispielhafte Adressbildung

- **Interface-Identifier:** `a667:06ff:fe7d:6082`
- **Empfangener Präfix:** `2001:0dc8:0100:f101::/64`
- **Gebildete globale Adresse:**
  `2001:0dc8:0100:f101:a667:06ff:fe7d:6082`

(Präfix aus Router Advertisement, Interface-ID lokal generiert.)

Einfach, aber mit Einschränkungen

- Keine Bereitstellung weiterer Konfigurationsparameter (z. B. DNS-, NTP- oder Suchdomäneninformationen).
- Erweiterung durch RFC 8106 [18] („Router Advertisement Options for DNS Configuration“) oder ergänzend durch **DHCPv6**.

Hinweis

In modernen Netzwerken werden **SLAAC** und **DHCPv6** häufig kombiniert, um vollständige Konfigurationsinformationen bereitzustellen.

DHCPv6 – Überblick

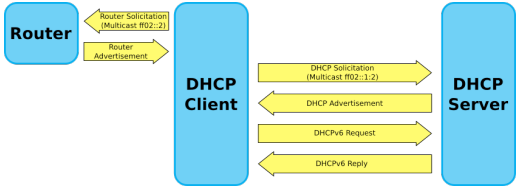
- **DHCPv6** (RFC 8415 [22]) ist das IPv6-Pendant zu DHCPv4, arbeitet aber **ohne Broadcasts**.
- Kommunikation erfolgt über die **Multicast-Adresse** `FF02::1:2`.
- Verwendet die UDP-Ports **546 (Client)** und **547 (Server)**.
- Kann Adressen (**stateful**) oder nur Zusatzinformationen (**stateless**) bereitstellen.

Wichtiger Unterschied zu DHCPv4

Der Standard-Router wird nicht über DHCPv6, sondern über **Router Advertisements** (Neighbor Discovery, RFC 4861 [28]) bekanntgegeben.

Weitere Details zum Ablauf folgen auf der nächsten Folie.

DHCPv6 – Ablauf der Kommunikation



- Der **DHCPv6-Client** erfährt über ein **Router Advertisement (RA)**, ob DHCPv6 Adressen (M-Bit) oder Zusatzinfos (O-Bit) liefern soll.
- Anschließend sendet er eine **Solicit**-Nachricht an `ff02::1:2`, um verfügbare Server zu finden.
- Server antwortet mit **Advertise** und bietet Konfigurationen an.
- Der Client sendet **Request**, um ein Angebot auszuwählen.
- Der Server bestätigt mit **Reply** und übermittelt die finalen Parameter (z. B. IPv6-Adresse, DNS, NTP).
- Spätere Nachrichten wie **Renew** oder **Rebind** dienen der Verlängerung der Zuweisung.

Quellen I

[1] Bootstrap Protocol.
RFC 951, September 1985.

[2] Arkko, J., and Bagnulo, M.
Cryptographically Generated Addresses (CGA) Extension Field Format.
RFC 4581, October 2006.

[3] Atkinson, R., and Kent, S.
Security Architecture for the Internet Protocol.
RFC 2401, November 1998.

[4] Borman, D. A., Deering, D. S. E., and Hinden, B.
IPv6 Jumbograms.
RFC 2675, August 1999.

[5] Carpenter, B. E., Chown, T., Gont, F., Jiang, S., Petrescu, A., and Yourtchenko, A.
Analysis of the 64-bit Boundary in IPv6 Addressing.
RFC 7421, January 2015.

[6] Cheshire, S., Aboba, D. B. D., and Guttman, E.
Dynamic Configuration of IPv4 Link-Local Addresses.
RFC 3927, May 2005.

[7] Cooper, A., Gont, F., and Thaler, D.
Security and Privacy Considerations for IPv6 Address Generation Mechanisms.
RFC 7721, March 2016.

[8] Deering, D. S. E., and Hinden, B.
IP Version 6 Addressing Architecture.
RFC 4291, February 2006.

[9] Deering, D. S. E., and Hinden, B.
Internet Protocol, Version 6 (IPv6) Specification.
RFC 8200, July 2017.

Quellen II

[10] Droms, R.
Dynamic Host Configuration Protocol.
RFC 2131, March 1997.

[11] Egevang, K. B., and Francis, P.
The IP Network Address Translator (NAT).
RFC 1631, May 1994.

[12] Egevang, K. B., and Srisuresh, P.
Traditional IP Network Address Translator (Traditional NAT).
RFC 3022, January 2001.

[13] Haberman, B., and Hinden, B.
Unique Local IPv6 Unicast Addresses.
RFC 4193, October 2005.

[14] Haberman, B., and Thaler, D.
Unicast-Prefix-based IPv6 Multicast Addresses.
RFC 3306, September 2002.

[15] Hätönen, S., Nyrhinen, A., Eggert, L., Strowes, S., Sarolahti, P., and Kojo, M.
An experimental study of home gateway characteristics.
In *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement* (New York, NY, USA, 2010), IMC '10, Association for Computing Machinery, p. 260–266.

[16] Hinden, B., and Deering, D. S. E.
Internet Protocol, Version 6 (IPv6) Specification.
RFC 2460, December 1998.

[17] Huston, G., Lord, A., and Smith, P.
IPv6 Address Prefix Reserved for Documentation.
RFC 3849, July 2004.

Quellen III

[18] Jeong, J. P., Park, S. D., Beloeil, L., and Madanapalli, S.
IPv6 Router Advertisement Options for DNS Configuration.
RFC 8106, March 2017.

[19] Kempf, J., Arkko, J., Zill, B., and Nikander, P.
SEcure Neighbor Discovery (SEND).
RFC 3971, March 2005.

[20] Keränen, A., Holmberg, C., and Rosenberg, J.
Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal.
RFC 8445, July 2018.

[21] Matsuzaki, Y., Kohno, M., Narten, D. T., Bush, R., Nitzan, B., and Colitti, L.
Using 127-Bit IPv6 Prefixes on Inter-Router Links.
RFC 6164, April 2011.

[22] Mrugalski, T., Siodelski, M., Volz, B., Yourtchenko, A., Richardson, M., Jiang, S., Lemon, T., and Winters, T.
Dynamic Host Configuration Protocol for IPv6 (DHCPv6).
RFC 8415, November 2018.

[23] Narten, D. T., Draves, R. P., and Krishnan, S.
Privacy Extensions for Stateless Address Autoconfiguration in IPv6.
RFC 4941, September 2007.

[24] Narten, D. T., Jinmei, T., and Thomson, D. S.
IPv6 Stateless Address Autoconfiguration.
RFC 4862, September 2007.

[25] Petit-Huguenin, M., Salgueiro, G., Rosenberg, J., Wing, D., Mahy, R., and Matthews, P.
Session Traversal Utilities for NAT (STUN).
RFC 8489, February 2020.

[26] Reddy.K, T., Johnston, A., Matthews, P., and Rosenberg, J.
Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN).
RFC 8656, February 2020.

Quellen IV

[27] Seo, K., and Kent, S.
Security Architecture for the Internet Protocol.
RFC 4301, December 2005.

[28] Simpson, W. A., Narten, D. T., Nordmark, E., and Soliman, H.
Neighbor Discovery for IP version 6 (IPv6).
RFC 4861, September 2007.

[29] Thubert, P., Bormann, C., Toutain, L., and Cragie, R.
IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Routing Header.
RFC 8138, April 2017.

[30] Zhang, L.
A retrospective view of network address translation.
Network, IEEE 22 (10 2008), 8 – 12.