

# IPv6 - Merkbätter

Michael Dienert

14. Dezember 2021

## Inhaltsverzeichnis

<b>1 IPv6-Adressen</b>	<b>1</b>
1.1 Aufbau, Schreibweise . . . . .	1
1.2 Typen von IPv6-Adressen . . . . .	3
1.3 Unspecified Address . . . . .	3
1.4 Loopback Address . . . . .	3
1.5 Link Local Unicast Addresses . . . . .	3
1.6 Global Scope Unicast Addresses . . . . .	3
1.7 Multicastadressen . . . . .	4
1.7.1 Solicited Node Address . . . . .	4
<b>2 SLAAC - Stateless Address AutoConfiguration</b>	<b>4</b>
2.1 Ablauf von SLAAC und Duplicate Address Detection . . . . .	4
<b>3 Adressauflösung</b>	<b>5</b>
<b>4 Interface Identifier</b>	<b>5</b>
4.1 EUI64 . . . . .	6
4.2 Temporäre IID . . . . .	6
4.3 Stable Privacy IID . . . . .	6
<b>5 IPv6-Header</b>	<b>7</b>
<b>6 ICMPv6-Neighbor-Solicitation-Message</b>	<b>7</b>

## 1 IPv6-Adressen

### 1.1 Aufbau, Schreibweise

- IPv6 Adressen sind 128 bit lang
- die bevorzugte Darstellung einer IPv6 Adresse ist:

```
x : x : x : x : x : x : x : x
```

wobei jedes x eine 16-bit Dualzahl ist, die mit max. 4 Hexadezimalziffern dargestellt wird. Dabei **sollen** führende Nullen in jedem der 8 Blöcke weggelassen werden. Beispiele:

```
ABCD:EF01:2345:6789:ABCD:EF01:2345:6789
2001:DB8:0:0:8:800:200C:417A
```

- in der Notation einer IPv6-Adresse darf *höchstens einmal* "::" auftauchen. Die beiden Doppelpunkte stehen dabei für eine oder mehrere aufeinanderfolgende Gruppen von 16 - Null-Bits. Die Adressen:

```
2001:DB8:0:0:8:800:200C:417A    a unicast address
FF01:0:0:0:0:0:0:101           a multicast address
0:0:0:0:0:0:0:1                the loopback address
0:0:0:0:0:0:0:0                the unspecified address
1:0:0:0:2:0:0:3
```

Dürfen so gekürzt werden:

```
2001:DB8::8:800:200C:417A a unicast address
FF01::101 a multicast address
::1 the loopback address
:: the unspecified address
1::2:0:0:3 oder 1:0:0:0:2::3
```

- Netze im IPv6-Umfeld werden so dargestellt:

```
Netzadresse/Präfixlänge
```

- Bei einer beliebigen IPv6-Adresse sollte immer die Präfixlänge mit angegeben werden:

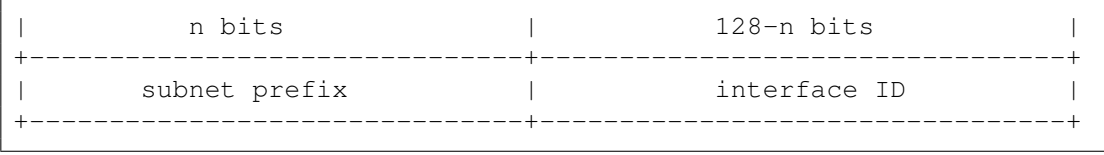
```
Netzadresse:    2001:7C0:caff:e00f::/64
Hostadresse:    2001:7C0:caff:e00f:1234:5678:9abc:def0/64
```

- Die Zahl *n* sei die Präfixlänge. Dann nennt man die vorderen *n* bits den **Präfix**
- Die hinteren 128-n bits nennt man Interface Identifier (IID)

Beispiel: IPv6-Host-Adresse (node=Host):



Aufgeteilt in Präfix und IID:



## 1.2 Typen von IPv6-Adressen

Es gibt 5 verschiedene Arten von IPv6-Adressen, wobei die ersten Bits entscheiden, zu welcher Art eine Adresse gehört:

Address type	Binary prefix	IPv6 notation
Unspecified	00...0 (128 bits)	::/128
Loopback	00...1 (128 bits)	::1/128
Multicast	11111111	FF00::/8
Link-Local unicast	1111111010	FE80::/10
Global Unicast	(everything else)	

Achtung! Die früher geltenden Einschränkungen für Global-Unicast-Adressen wurden gestrichen!

## 1.3 Unspecified Address

Die Adresse 0:0:0:0:0:0:0:0 = :: nennt man *Unspecified Address*. Sie wird als Quelladresse verwendet, wenn ein Host noch keine IPv6 Adresse hat (s.u.). Ausserdem dient das Netz ::/0 als Ziel der Default-Route.

## 1.4 Loopback Address

Die Adresse 0:0:0:0:0:0:0:1 = ::1 nennt man *Loopback Address*. Sie wird als Zieladresse verwendet, wenn ein Host sich selbst ein Paket senden möchte.

## 1.5 Link Local Unicast Addresses

Link-Local-Adressen haben folgende Form:

10									
bits			54 bits			64 bits			
+-----+	+-----+			+-----+			+-----+		
1111111010	0000000	0				interface ID			
+-----+	+-----+			+-----+			+-----+		

Sie beginnen also immer mit der binären Folge 1111 1110 1000 0000, Hexadezimal: **FE80**

Sie gelten nur innerhalb eines Layer2-Subnetzes (Broadcastdomäne, Switch, VLAN) und dürfen von einem Router nicht weitergeleitet werden.

## 1.6 Global Scope Unicast Addresses

Alle globalen Unicast-Adressen die **nicht** mit binär 000 beginnen, müssen eine Präfixlänge von 64 haben! Sie haben also folgenden Aufbau:

	n bits		m bits		64 bits	
+-----+		+-----+		+-----+		+-----+
	global routing prefix		subnet ID		interface ID	
+-----+		+-----+		+-----+		+-----+

## 1.7 Multicastadressen

Multicastadressen beginnen immer mit 11111111=FF. Die allerwichtigsten sind folgende:

```
FF02:0:0:0:0:0:0:1 = FF02::1  All Nodes innerhalb des Layer2-Subnetzes (Link)
FF02:0:0:0:0:0:0:2 = FF02::2  All Routers innerhalb des Layer2-Subnetzes
FF05:0:0:0:0:0:0:2 = FF05::2  All Routers innerhalb eines Clusters von Subnetzen
```

### 1.7.1 Solicited Node Address

Eine spezielle, reservierte Multicastadresse ist die *Solicited Node Address*:

```
FF02:0:0:0:0:0:1:FFXX:XXXX
```

Um eine Solicited Node Address zu bilden, ersetzt ein Hostrechner die XX:XXXX durch die letzten 24bit einer IPv6-Adresse.

Beispiel:

```
IPv6-Adresse:                4037::01:800:200E:8C6C
korrespondierende Solicited Node Address:  FF02::1:FF0E:8C6C
```

Sie wird beim *Neighbor Discovery Protocol* (NDP) verwendet. Die Solicited Node Address ist dort Zieladresse einer *Neighbor Solicitation Nachricht* (s.u.).

## 2 SLAAC - Stateless Address AutoConfiguration

Bei SLAAC, der automatischen Adresskonfiguration vergeben sich die Hosts (Nodes) selbst Adressen. **Stateless** bedeutet, dass keine Instanz (Server) die Adressvergabe regelt und über die vergebenen Adressen Buch (z.B. Leases-DB bei dhcp) führt.

Der Präfix wird dabei hauptsächlich auf 2 Arten gebildet:

1. Für Link-Lokale-Adressen wird immer der Präfix FE80::/64 genommen
2. Für globale Unicast Adressen wird der Präfix mittels *Router Advertisements* vom Router des Subnetzes bezogen.

Der Interface-Identifizier kann auf verschiedene Arten gebildet werden. Vgl. dazu Abschnitt 4.

### 2.1 Ablauf von SLAAC und Duplicate Address Detection

Bevor ein Host/Node sich selbst eine Wunschadresse vergibt, prüft er mittels *Duplicate Address Detection* ob die Adresse nicht schon vergeben ist. Hierzu sendet er eine *Neighbor Solicitation* Nachricht. Erhält er keine Antwort auf die Nachricht, vergibt er sich selbst seine Wunschadresse.

Der folgende Mitschnitt mit `tcpdump` zeigt den Ablauf an einem Beispiel (Link-Lokale-Wunschadresse). Die Aufgezeichneten Daten zeigen den Ethernet-Frame mit eingebettetem IPv6-Paket (vgl. Abschnitt 5), das wiederum eine ICMP6-Neighbor-Solicitation-Nachricht enthält (vgl. Abschnitt 6).

```
IP6 :: > ff02::1:ff39:b75c: ICMP6, neighbor solicitation,
      who has fe80::5005:c938:6f39:b75c, length 32
```

```

0x0000: 3333 ff39 b75c dca6 320a 7012 86dd 6000
0x0010: 0000 0020 3aff 0000 0000 0000 0000 0000
0x0020: 0000 0000 0000 ff02 0000 0000 0000 0000
0x0030: 0001 ff39 b75c 8700 3a58 0000 0000 fe80
0x0040: 0000 0000 0000 5005 c938 6f39 b75c

```

**Ziel-MAC** : 3333 ff39 b75c : 3333 + die letzten 4 Oktette der IPv6 Multicast-Adresse

**Quell-MAC** : dca6 320a 7012 : MAC der RaspberryPI Foundation

**Ethertype** : 86dd (Type=IPv6)

**Quell-IP** : 0000 0000 0000 0000 0000 0000 0000 0000

**Ziel-IP** : ff02 0000 0000 0000 0000 0001 ff39 b75c

**ICMPv6-Type und Code** : 8700; 0x87 = 135; Code = 0x00 = 0

**Wunschadresse** : fe80 0000 0000 0000 5005 c938 6f39 b75c

Die Ziel-MAC hat den Wert: Man erkennt als Quell-IP die Adresse 0:0:0:0:0:0:0

### 3 Adressauflösung

IPv6 kennt kein ARP. Um dennoch MAC-Adressen (Layer2) in IPv6-Adressen auflösen zu können wird ähnlich wie bei SLAAC das *Neighbor Discovery Protocol* verwendet.

Hier ein Beispiel (Mitschnitt). Die zur IPv6-Adresse fe80::5005:c938:6f39:b75c gehörende MAC-Adresse ist gesucht. Die gesuchte Adresse ist im *Neighbor Advertisement Paket* enthalten (hier: dc:a6:32:0a:70:12).

Im Unterschied zur Duplicate Address Detection wird hier als Quelladresse nicht die Unspecified Address, sondern die eigene IPv6 verwendet:

```

IP6 fe80::683f:9847:5ab3:de80 > ff02::1:ff39:b75c: ICMP6,
  neighbor solicitation, who has fe80::5005:c938:6f39:b75c, length 32
  0x0000: 3333 ff39 b75c c860 00c7 0b1f 86dd 6000
  0x0010: 0000 0020 3aff fe80 0000 0000 0000 683f
  0x0020: 9847 5ab3 de80 ff02 0000 0000 0000 0000
  0x0030: 0001 ff39 b75c 8700 7731 0000 0000 fe80
  0x0040: 0000 0000 0000 5005 c938 6f39 b75c 0101
  0x0050: c860 00c7 0b1f
IP6 fe80::5005:c938:6f39:b75c > fe80::683f:9847:5ab3:de80: ICMP6,
  neighbor advertisement, tgt is fe80::5005:c938:6f39:b75c, length 32
  0x0000: c860 00c7 0b1f dca6 320a 7012 86dd 6000
  0x0010: 0000 0020 3aff fe80 0000 0000 0000 5005
  0x0020: c938 6f39 b75c fe80 0000 0000 0000 683f
  0x0030: 9847 5ab3 de80 8800 e1f9 6000 0000 fe80
  0x0040: 0000 0000 0000 5005 c938 6f39 b75c 0201
  0x0050: dca6 320a 7012

```

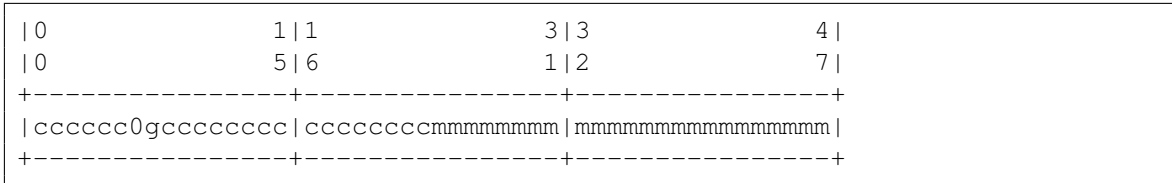
### 4 Interface Identifier

Der IID kann auf 3 Arten gebildet werden:

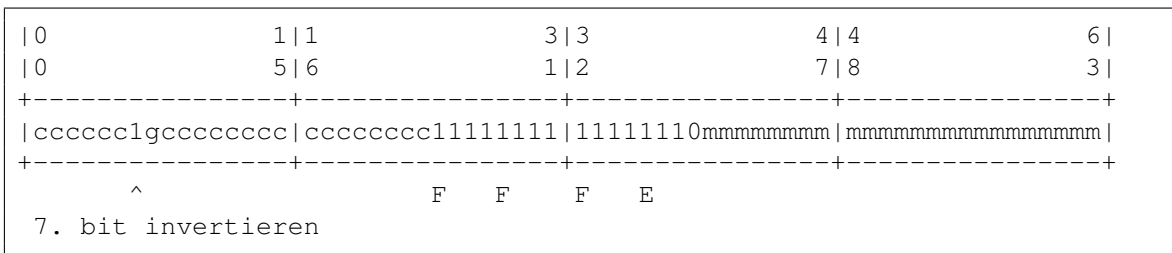
## 4.1 EUI64

IEEE EUI-64 : der IID enthält die eingebettete MAC-Adresse der Netzwerkschnittstelle.

Allgemeine Form einer MAC-Adresse. c=bits der Hersteller-ID, g=group-bit (Wert spielt hier keine Rolle) und m=bits der Hersteller- (manufacturer-) Seriennummer:



Daraus wird ein IID im EUI64-Format gebildet:



## 4.2 Temporäre IID

Temporäre IIDs werden aus dem linken Teil eines MD5-Hash abgeleitet. Bei der allerersten Erzeugung einer Adresse wird eine Zufallszahl in den MD5-Hash umgerechnet.

Beim nächsten Neustart des Rechners, wird der neue Hash aus den 64 niederwertigen bits des alten Hashes gebildet. Diese mussten zuvor persistent gespeichert werden. Ist persistente Speicherung nicht möglich, wird wieder eine Zufallszahl genommen.

D.h. bei jedem Neustart ändert sich die IID.

## 4.3 Stable Privacy IID

Stabile, aber nicht rückverfolgbare IIDs werden wie folgt gebildet:

$$IID = SHA(Prefix, Interface - Name, [Network - ID], DAD - Zaehler, secret\_key)$$

dabei sind:

**SHA** Secure Hash Algorithym; nur SHA-1 oder SHA-256 sind erlaubt (SHA-1 ist inzwischen abgekündigt)

**Prefix** der IPv6-Praefix des Subnetzes; also FE80::/64 oder der vom Router gelieferte globale Präfix

**Interface-Name** z.b. eth0, enp3s0, enp5s0, ...

**Network-ID** das ist optional, bei WLAN wäre das z.b. die SSID

**DAD-Zaehler** eine systemweite, persistente Variable, die bei der Installation des Hosts auf 0 gesetzt wird. Gibt es bei der Duplicate Address Detection (vgl. Abschnitt 2.1) einen Konflikt, wird die Variable um ein hochgezählt und eine neue IID wird errechnet.

**secret\_key** eine mindestens 128bit lange Zufallszahl, die bei der Erstinstallation des Rechners errechnet und persistent gespeichert wird.

