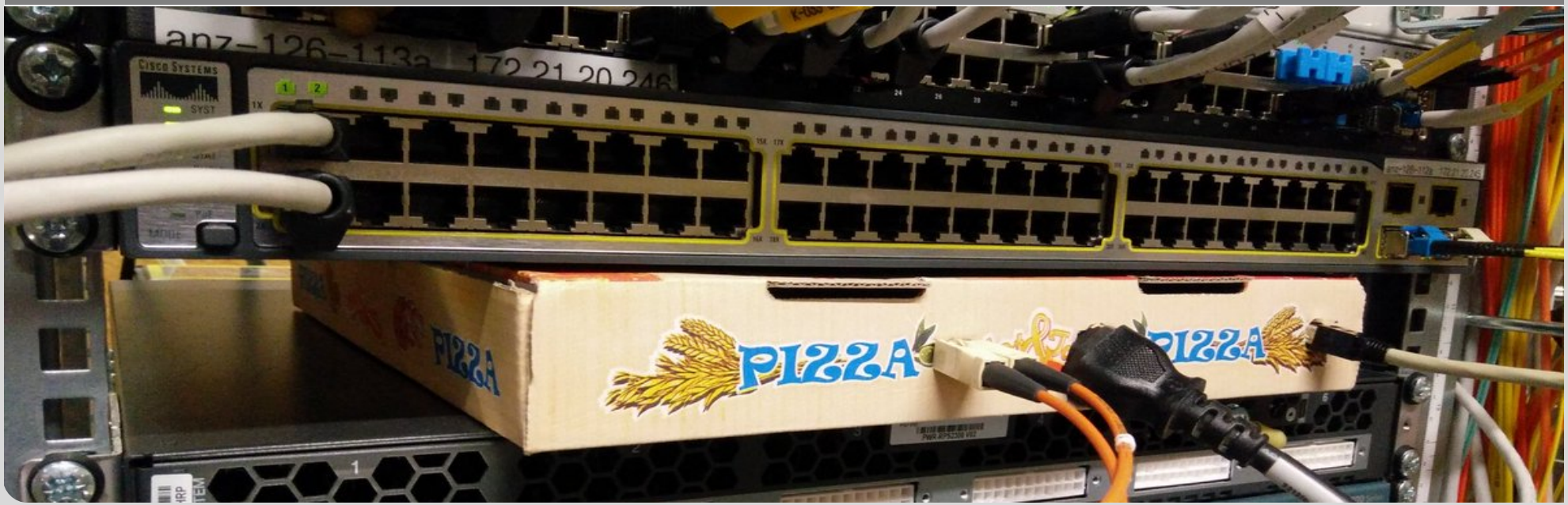
Karlsruher Institut für Technologie

Steinbuch Centre for Computing

Netzwerk-Konfiguration mit IPv6 am KIT

Klara Mall | 22. Juli 2019

STEINBUCH CENTRE FOR COMPUTING - SCC



IPv4 und IPv6

- IPv4 und IPv6 laufen parallel
- IPv4 und IPv6 sind nicht kompatibel
- gleicher Zweck, unterschiedliche Lösung
- IPv4 ist von 1982, IPv6 von 1998

Zahlen

- eine IPv4-Adresse kostet heute ca. 18 €
- 129.13.0.0/16 und 141.52.0.0/16 sind jeweils ca. 1,18 Mio € wert

- IPv4: 2^{32} IPv4-Adressen
- IPv6: 2^{128} IPv6-Adressen
- in ein einziges Subnetz bei IPv6 (/64) passt der komplette IPv4-Adressraum im Quadrat hinein

Hexadezimalsystem

- 1. Zeile: Dezimal = 10 Ziffern (0,1,2,3,4,5,6,7,8,9)
- 2. Zeile: Hexadezimal = 16 Ziffern (0,1,2,3,4,5,6,7,8,9,a,b,c,d,e,f)

dec:	0	1	2	3	4	5	6	7	8	9	10
hex:	0	1	2	3	4	5	6	7	8	9	a

dec:	11	12	13	14	15	16	17	18	19	20	21
hex:	b	c	d	e	f	10	11	12	13	14	15

dec:	22	23	24	25	26	27	28	...
hex:	16	17	18	19	1a	1b	1c	...

IPv6-Adressen

Subnetzgrößen für Endnutzer:

- IPv4: viele verschiedene (z. B. /24 oder /27)
  IPv6: immer /64
- eine IPv6-Adresse besteht aus 128 bit, das sind 8 durch Doppelpunkte getrennte Blöcke, die jeweils aus vier Ziffern bestehen:

KIT-Präfix: 2a00:1398 ::/32
(2a00:1398: KIT Präfix)

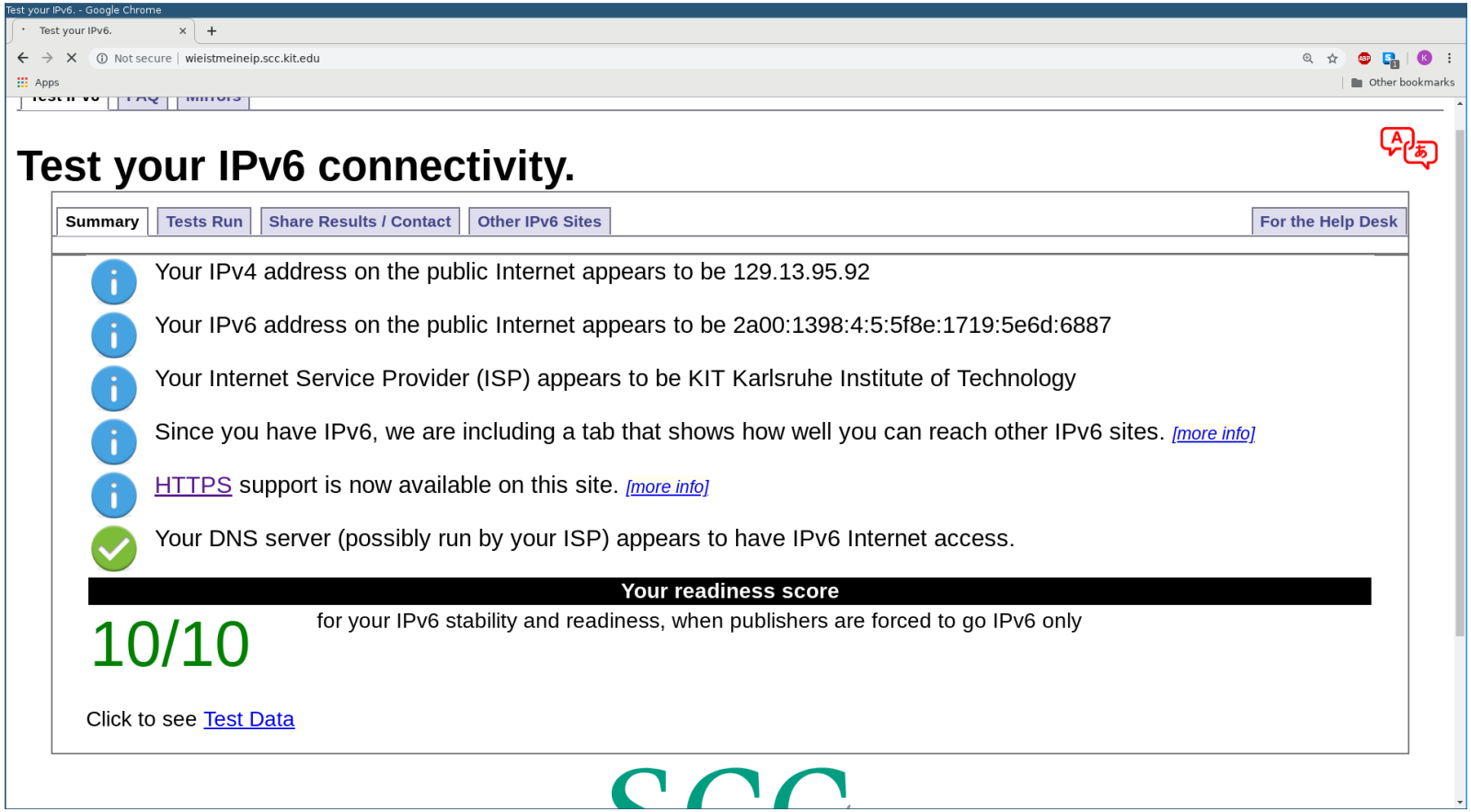
Subnetz: 2a00:1398: 0004:00a7 ::/64
(2a00:1398: KIT Präfix; 0004:00a7 Subnetz-ID)

IP-Adresse: 2a00:1398:0004:00a7: 0000:0000:0000:0000
(2a00:1398:0004:00a7: Netzwerk-Teil / Präfix; 0000:0000:0000:0000 Host-Teil)

Schreibweisen von IPv6-Adressen

- 2a00:1398:0004:00a7:0000:0000:0000:10
  = 2a00:1398:4:a7:0:0:0:10
  = 2a00:1398:4:a7::10
- $\Rightarrow$ führende Nullen werden weggelassen
- $\Rightarrow$ Doppelpunkt fasst mehrere Nuller-Blöcke zusammen, darf nur einmal vorkommen

Webseite wieistmeineip.scc.kit.edu



Subnetzvergabe

- Einheit: Institut, SCC-Abteilung, Hochschulgruppe, . . .
- Jeder Einheit wird ein /56-Subnetz zugewiesen.
- Daraus werden /64 Subnetze an die Einheit vergeben.

IPv6 Subnetz

- Subnetz: 2a00:1398: 4:a7 ::/64
- die ersten 16 Adressen sind von SCC-NET reserviert
- $\Rightarrow$ die erste freie Adresse ist die ::10

IPv6 - Gültigkeitsbereiche (scopes)

- neu in IPv6
- scope ist ein Teil des Netzwerks, in dem die zugehörige Adresse geroutet wird
- wichtigste scopes: link-local und global
- Link-lokale Adresse:
  - nur auf dem Netzwerk-Segment (VLAN) gültig. Kommunikation mit dem Router über link-lokale Adresse.
  - Auf jedem Interface gibt es eine Adresse mit scope link.
  - Zusätzlich beliebig viele Adressen mit scope global (statisch, SLAAC, temporäre Adressen)

IPv6 - link-lokale Adressen

- Link-local Prefix: fe80::/64
  Beispiel link-lokale Adresse: fe80:: a012:2729:5aab:113c
- Link-lokale Adressen sind nur pro Interface eindeutig
  $\Rightarrow$ dieses muss mit % an die Adresse angehängt werden:
  ping fe80:: a012:2729:5aab:113c %2
  (Interface-Namen unterschiedlich je nach Betriebssystem)

Neighbor Discovery Protocol - NDP

- neu in IPv6, nutzt ICMPv6
- nutzt immer link-local als Source-Adresse
- Neighbor discovery messages
  - Router solicitation
  - Router advertisement
  - Neighbor solicitation
    entspricht ARP Request bei IPv4
  - Neighbor advertisement
    entspricht ARP Reply bei IPv4

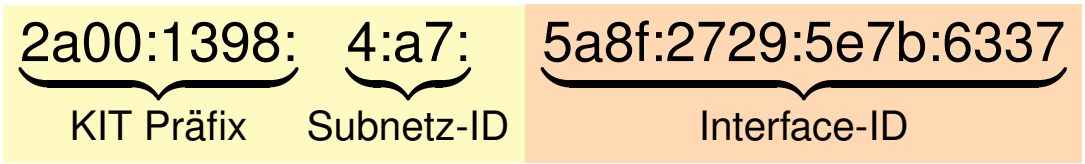
Router Advertisements

Wesentliche Bestandteile von IPv6:

- Router Advertisement:
  enthält Gateway und Präfix
- Stateless Address Autoconfiguration (SLAAC)

IPv6-Adressen für Clients

- IPv6-Autokonfiguration: Host-Teil legt der Host selbst fest
  5a8f:2729:5e7b:6337 = Interface-ID
- Woher weiß der Host den Netzwerk-Teil (Präfix) ?
- ⇒ Router Advertisements = RA



Netzwerk-Teil
Präfix

Host-Teil

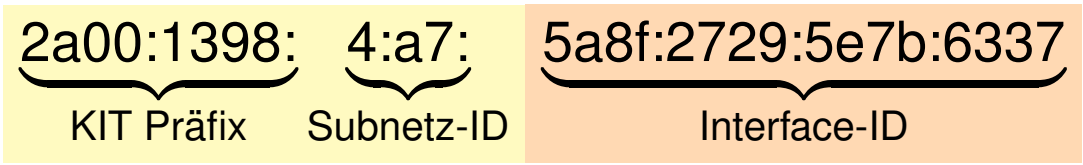
Duplicate Address Detection (DAD)

- DAD: vor dem Konfigurieren der Adresse auf dem Interface (solange ist Adresse tentative)
- im allgemeinen doppelte Adressen sehr unwahrscheinlich wegen des großen Adressraums

SLAAC: Interface-ID

Wie kommt der Host-Teil (Interface-ID) in der IPv6-Adresse zustande?

- EUI-64 (abgeleitet von MAC-Adresse)
- Stable Privacy (stabil, aber zufällig)



EUI-64

- EUI-64 kann man bereits als überholt betrachten
- In der IETF[1] ist man der Ansicht, dass EUI-64 nicht mehr verwendet werden sollte (vgl. RFC 8064, RFC 7721, RFC 7217)
- Privacy-Problematik
- Anfälligkeit gegen Scans und gezielte Angriffe
- EUI-64 bereits nicht mehr der Default in Windows und Linux Network-Manager

[1] Internet Engineering Task Force (IETF): eine Organisation, die sich mit der technischen Weiterentwicklung des Internets befasst.

Stable Interface ID Generation

Random ID = RID als Alternative zu EUI-64 (RFC 7217)

- RID = hash (Prefix, Net_Iface, DAD_Counter, secret_key)
- ebenfalls stabil/konstant für jedes Netzwerk-Interface, aber mehr Privacy
- IIDs (Interface IDs) sind unterschiedlich in unterschiedlichen Netzwerken

Interface-ID: Privacy Extensions / Temporary Addresses

Temporäre Adressen zusätzlich zur stabilen Adresse für noch mehr Privacy:

- temporär: Adressen mit beschränkter Lebensdauer
- werden als Source Address präferiert
  (RFC 6724: Default Address Selection for IPv6)
- solange noch Verbindungen offen, konfiguriert als deprecated
- daher nach einer Weile u. U. sehr viele temporäre Adressen am Interface konfiguriert

Empfehlung: bei Servern abschalten
(bei Windows Server per default abgeschaltet)

Default Gateway

- bekannt durch Router Advertisement
- daher: Default Gateway muss auch bei statischer IPv6-Adresse nicht konfiguriert werden
- im neuen KIT-Core wird für jedes Subnetz das Gateway auf fe80::1 gesetzt
- Empfehlung für Hosts mit statischer IP-Adresse im neuen KIT-Core: Default Gateway fe80::1 fest eintragen
- Vorerst (alter KIT-Core): Default Gateway auch bei Servern nicht konfigurieren

Beispiel: Windows 10
IP-Adress-Konfiguration

```
Wireless LAN adapter WiFi:

   Connection-specific DNS Suffix  . : scc.kit.edu
   Description . . . . . . . . . . . : Qualcomm Atheros AR946x Wireless Network Adapter
   Physical Address. . . . . . . . . : [redacted]
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   IPv6 Address. . . . . . . . . . . : 2a00:1398:9:fb03:913:47eb:[redacted](Preferred)
   Temporary IPv6 Address. . . . . . : 2a00:1398:9:fb03:34e3:b80f:1009:a84d(Preferred)
   Link-local IPv6 Address . . . . . : fe80::913:47eb:[redacted]%18(Preferred)
   IPv4 Address. . . . . . . . . . . : 129.13.31.75(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.224.0
   Lease Obtained. . . . . . . . . . : Sunday, March 31, 2019 3:13:14 PM
   Lease Expires . . . . . . . . . . : Sunday, March 31, 2019 3:33:14 PM
   Default Gateway . . . . . . . . . : fe80::4255:39ff:fec0:3a80%18
                                       129.13.0.1
   DHCP Server . . . . . . . . . . . : 172.21.64.133
   DHCPv6 IAID . . . . . . . . . . . : [redacted]
   DHCPv6 Client DUID. . . . . . . . : [redacted]
   DNS Servers . . . . . . . . . . . : 129.13.64.5
                                       129.13.96.2
   NetBIOS over Tcpip. . . . . . . . : Enabled
```

Beispiel: Windows 10 Routingtabelle

`netstat -rn`

```
IPv6 Route Table
===========================================================================
Active Routes:
 If Metric Network Destination      Gateway
  9    281 ::/0                     fe80::4255:39ff:fec0:3a80
  1    331 ::1/128                  On-link
  9    281 2a00:1398:4:1::/64       On-link
  9    281 2a00:1398:4:1:963:82e1:[redacted]/128
                                    On-link
  9    281 2a00:1398:4:1:e098:a702:[redacted]/128
                                    On-link
  9    281 fe80::/64                On-link
  9    281 fe80::963:82e1:[redacted]/128
                                    On-link
  1    331 ff00::/8                 On-link
  9    281 ff00::/8                 On-link
===========================================================================
```

Powershell: Konfiguration der Netzwerk-Einstellungen

- Set-NetIPv6Protocol -RandomizeIdentifiers Disabled
  (EUI-64 statt Stable Privacy)
- Set-NetIPv6Protocol -UseTemporaryAddresses Disabled
  (keine Privacy Extensions)
- Set-NetIPInterface -RouterDiscovery Disabled
  (keine Auswertung von Router Advertisements)

Werte auslesen mit Get-NetIPv6Protocol bzw. Get-NetIPInterface.
Default-Werte:

- RandomizeIdentifiers: enabled
- UseTemporaryAddresses: disabled (Server), enabled (Client)
- RouterDiscovery: enabled

Beispiel: Linux IP-Adress-Konfiguration

```
ip addr show dev wlp58s0
```

```
3: wlp58s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen
    link/ether [redacted] brd ff:ff:ff:ff:ff:ff
    inet 129.13.95.92/25 brd 129.13.95.127 scope global dynamic wlp58s0
       valid_lft 5398sec preferred_lft 5398sec
    inet6 2a00:1398:4:5:21d7:4f7b:7748:654/64 scope global temporary tentative dynamic
       valid_lft 604800sec preferred_lft 85803sec
    inet6 2a00:1398:4:5:5f8e:1719:[redacted]/64 scope global tentative mngtmpaddr noprefixrou
       valid_lft 2592000sec preferred_lft 604800sec
    inet6 fe80::16d3:bb16:[redacted]/64 scope link
       valid_lft forever preferred_lft forever
```

Beispiel: Linux Routingtabelle

```
ip -6 route
```

```
2a00:1398:4:5::/64 dev wlp58s0 proto ra metric 600  pref medium
fe80::1 dev wlp58s0 proto static metric 600  pref medium
fe80::/64 dev wlp58s0 proto kernel metric 256  pref medium
default via fe80::1 dev wlp58s0 proto static metric 600  pref medium
```

Linux: Konfiguration der Netzwerk-Einstellungen

- sysctl net.ipv6.conf.all.autoconf=0 (kein SLAAC)
- sysctl net.ipv6.conf.all.accept_ra=0 (keine RA akzeptieren)
- sysctl net.ipv6.conf.all.use_tempaddr=0 (keine Privacy Extensions)

Network-Manager:

- nmcli con modify “KIT” ipv6.addr-gen-mode stable-privacy
- nmcli con modify “KIT” ipv6.ip6-privacy 0

Auswahl einer statischen IPv6-Adresse (z. B. für Server)

- **Empfohlen**:
  völlig zufällig oder vorne anfangen und hochzählen:
  - 2a00:1398:4:a7:7e04:ac49:1231:62f5
  - 2a00:1398:4:a7::10, ::11, ..., ::1f, ::20
  - 2a00:1398:4:a7::23:1, ::23:2, ..., ::23:f, ::23:20
- **Gültige Schreibweise, aber nicht mehr empfohlen**:
  - 2a00:1398:4:a7::141.52.17.3 = 2a00:1398:4:a7::8d34:1103
  - 2a00:1398:4:a7:141:52:17:3

Alternativen für Spaßvögel:
2a00:1398:4:a7::dead:beef oder 2a00:1398:4:a7::cafe

DNS-Resolver

DNS-Resolver KIT:

- 2a00:1398::1
- 2a00:1398::2
- (2a00:1398::3)
- (2a00:1398::4)

Empfehlung:

- IPv6-Resolver eintragen (::1 und ::2)
- bei Linux mehr als drei nicht möglich

DNS-Einträge

- IPv4: A-Record, IPv6: AAAA-Record (sprich: Quad-A-Record)
- wenn ein Name A- und AAAA-Record hat: IPv6 wird präferiert
- IPv4 und IPv6 jeweils eigener DNSVS-Bereich

DNS-Einträge für Server

- Eintrag der statischen Adresse im DNSVS als AAAA-Record (z. B. 2a00:1398:4:a7::1:1)
- NATVS: falls Freischaltung auf den Namen vorhanden $\Rightarrow$ Freischaltungen gelten automatisch für IPv4 und IPv6

DNS-Einträge für Clients

DNSVS-Eintrag empfohlen auch bei Clients.

Vergleich IPv4-M-DHCP versus IPv6-Autokonfiguration:

- IPv4: MAC-Adresse am Interface ablesen, IP-Adresse zuordnen und mit Namen eintragen
- IPv6: stabile SLAAC-Adresse am Interface ablesen und mit Namen eintragen

Vorteile DNS-Eintrag:

- mit AAAA-Record Zugriff von Remote über IPv6 möglich
- Dokumentationsmöglichkeit im DNSVS nutzen

DNS-Reverse-Einträge

- PTR-Record: Reverse-DNS-Eintrag (IP-Adresse wird in Name aufgelöst)
- im DNSVS bei AAAA-Host-Record-Eintrag automatisch mit dabei
- Reverse-DNS-Einträge unmöglich für zufällige temporäre Adressen
- aber manche Anwendungen verlangen immernoch Reverse-DNS-Einträge (z. B. Mailserver)
- DNS-Dienst-seitige Lösung in Arbeit (z. B. automatisches Generieren von DNS-Einträgen)

IPv6 kurz gefasst

Client:

- Rechner anschalten und es funktioniert
- Optional AAAA-Record für stabile SLAAC-Adresse im DNSVS eintragen:
  - mit AAAA-Record Zugriff von Remote über IPv6 möglich
  - Dokumentationsmöglichkeit im DNSVS nutzen

Server:

- Statische Adresse raussuchen, auf dem Server konfigurieren
- Default Gateway im neuen KIT-Core:
  fe80::1 (vorerst noch leer lassen)
- wenn Adresse funktional: AAAA-Record im DNSVS eintragen

Fragen?

DHCPv6

- DHCPv6 existiert, ist aber ungleich DHCPv4
- kein Default Gateway
- keine Netzmaske
- basiert nicht auf der MAC-Adresse
- Router Advertisements braucht man immer, außer bei komplett statischer Konfiguration
- Android unterstützt kein DHCPv6 („won't fix“)

SLAAC versus DHCPv6

Vorteile SLAAC für die IP-Adressvergabe:

- Skalierbarkeit: ein Router Advertisement für alle Clients
- keine zusätzlichen redundanten Server nötig, die den State halten müssen
- Ausfallsicherheit
- Komplexitätsreduktion

Other configuration options

Unterscheidung stateful versus stateless DHCPv6:

- stateful: IP-Adressen werden vergeben
- stateless: nur sonstige Einstellungen wie Name-Server, Search-Domain, NTP-Server, . . . werden vergeben

Sonstige Einstellungen:

- Name-Server und Search-Domains können auch über RA mitgeteilt werden (RDNSS, RFC 8106)
- Windows 7 und 8: unterstützen kein RDNSS, aber stateless DHCPv6

Am KIT:

- IP-Adresse: SLAAC, DNS-Einstellungen: RDNSS/DHCPv6
- stateful DHCPv6 nur in Ausnahmefällen (z. B. Prefix Delegation)

Unbekannte Geräte in meinem VLAN

- mit IPv6-Autokonfiguration kann sich jeder Host selbst eine IP-Adresse konfigurieren
- Link-local-Adresse ist bei IPv6 sowieso vorhanden
- Malware kommt auch ohne DHCP oder Prefix-Advertisement ins Netz
- kein Adressmangel im Gegensatz zu IPv4

- Wirksamer Schutz gegen unbekannte Geräte: Port Security
- Nutzung des Gäste-Netzes LTA (Laptop Access) oder KIT-WLAN
- ungenutzte Ports dekonfigurieren lassen

IPv6-only

- bei beschränktem Kommunikationskreis bereits möglich (alle Kommunikationspartner sprechen IPv6)
- sobald alle Clients im KIT IPv6 haben, können rein interne KIT-Dienste IPv6-only betrieben werden
- Client-Netze IPv6-only mit Übersetzungstechnologien wie NAT64/DNS64 oder 464XLAT oder mit VPN

VLAN: Failure Domain und Trust Domain

- Schleifen, Broadcast-Stürme
- Autodiscovery
- Mögliche Angriffe innerhalb eines VLANs:
  - IP-Spoofing
  - Rogue DHCP-Server
  - ARP-Proxy / ARP-Spoofing
  - Rogue Router Advertisements
  - Neighbor Discovery DoS Attack
- Vorteile der Auftrennung in mehrere VLANS:
  - Trennung des Non-Unicast-Traffic
  - Reduzierung von Layer-2-Overhead
  - Verkleinerung der Angriffsfläche durch gleichartige Systeme
  - Einheitliche Security Policy
  - Vereinfachung der Administration