

IPv6

Das neue Internetprotokoll im Überblick

von

Torben Keil

Der Rote Faden

- Rechnernetzwerke
- ISO- / OSI-Schichtenmodell
- Verbindungsweiterleitung durch Hub / Switch / Router
- Verbindungsweiterleitung
- Protokollbeispiele
- Adressbereiche von IPv4 und IPv6
- IPv6 Adress-Schemata
- IPv6 „Classful Networking“
- IPv6 Router im Eigenbau
- Manuelle Konfiguration
- Automatische Konfiguration
- Stateless Autoconfiguration
- DHCPv6
- Globale IPv6-Adresse erhalten
- NAT64
- IPsec
- Mobile IPv6 (RFC 6275)

Rechnernetzwerke

... bestehen aus...

- einer physikalischen Verbindung
- Switches, Hubs
- Router
- (Server)

IPv6 ist eines von vielen Protokollen im Netzwerk

ISO- / OSI-Schichtenmodell



Schicht 1: Physikalische Schicht

- Binäre Zahlenfolgen werden in elektrische Signale umgewandelt und umgekehrt.
- Kodieralgorithmen nutzen den vorhandenen Sende- und Empfangskanal optimal aus.
- Effiziente Algorithmen für den Medienzugriff steuern die Sendeeinheit

Schicht 2: Sicherungsschicht

- Daten werden gegen Störeinflüsse geschützt.
- Fehlererkennungs- und -korrekturalgorithmen erkennen fehlerhaft empfangene Daten und rekonstruieren sie mit Hilfe redundanter Informationen.
- Effektive Algorithmen verhindern dass viele redundante Informationen gesendet werden müssen.

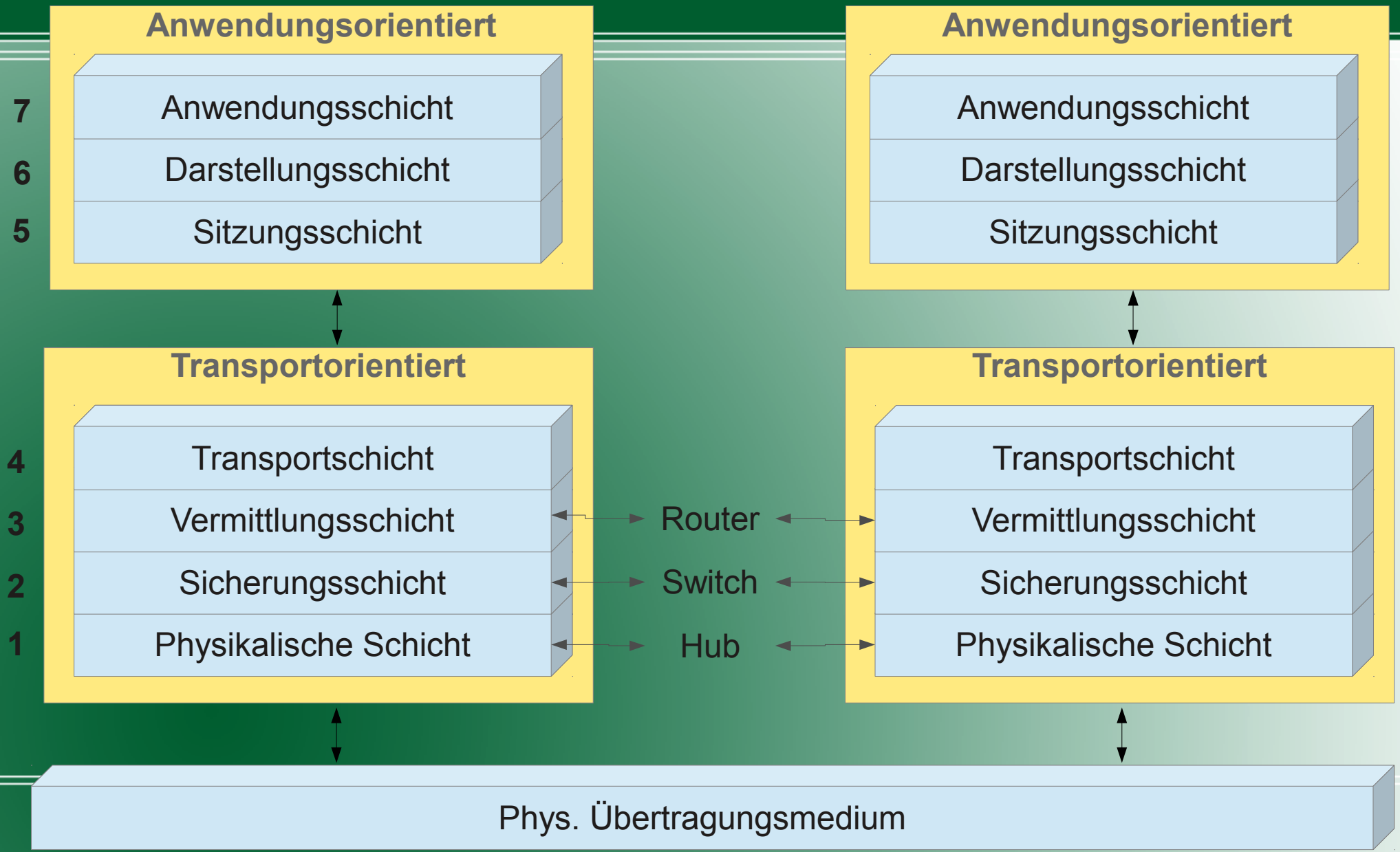
Schicht 3: Vermittlungsschicht

- Das IP-Protokoll befindet sich in der Vermittlungsschicht.
- Mit diesem Protokoll können mehrere Rechner über Netzwerkgrenzen hinweg verbunden werden.

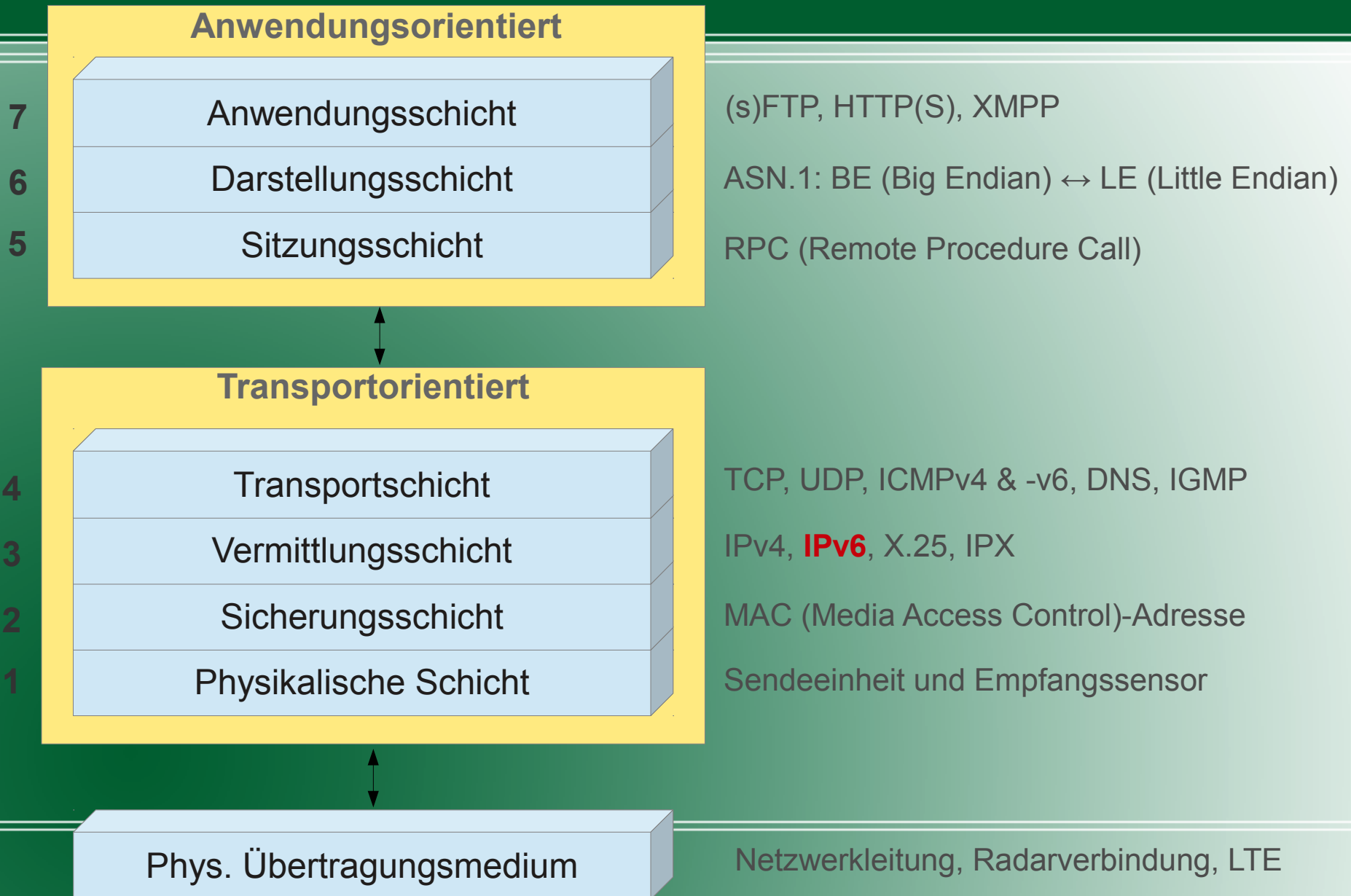
Verbindungsweiterleitung durch Hub / Switch / Router

- **Hub (Schicht 1):**
Verstärkt die el. Signale auf den Leitungen
(z.B. USB-Hub, 100MBit-Hub)
- **Switch (Schicht 2):**
Speichert sich MAC-Adressen und schaltet
zwischen den Netzwerkports eine direkte Verbindung
- **Router (Schicht 3):**
Speichert sich IP-Adresszuordnungen und sendet diese an
dem richtigen Netzwerkport raus.

Verbindungsweiterleitung



Protokollbeispiele



Adressbereiche von IPv4 und IPv6

- IPv4-Adresse (32 Bit Adresse):

192.168.1.1 / 24

=> 3x8 Bit für die Netzwerkadresse

=> 1x8 Bit für die Rechneradresse im Netzwerk

- IPv6-Adresse (128 Bit Adresse):

2a01:0198:0158:0000:0000:0000:0000:0001 / 64

Alternative Darstellung: 2a01:198:158::1 / 64

=> 4x16 Bit für die Netzwerkadresse

=> 4x16 Bit für die Rechneradresse im Netzwerk

IPv6 Adress-Schemata

Drei Adress-Segmente bilden die IPv6-Adresse

1.	0 - 47 Bit	Global Routing Prefix (bei /48 -Netzen)	} Subnetz- Prefix
2.	48 - 63 Bit	Subnetz-ID	
3.	64 - 128 Bit	Interface-ID	

Beispiel:

fe80:0000:0000:0000:2676:13ff:fedb:5aa4

↓
65536 Subnetze

↓
18,45 Trillionen
Interfaces

IPv6 Adress-Schemata

IPv6 Adressen können als Unicast-, Anycast und Multicast-Adressen eingeordnet werden.

Unicast-Adressen identifizieren einen einzigen Netzanschluss.

Ein Paket mit einer Anycast-Zieladresse gelangt nur zu einem Empfänger aus einer Gruppe (mit der gleichen Unicast-Adresse).

Eine Multicast-Nachricht geht an alle Empfänger einer Gruppe.

Der Beginn der Adresse (Präfix) zeigt die Art der Nachricht an.

Präfix (bin)	Präfix (hex)	Adressraumgröße	Bedeutung
0000 0000	0x0000	1/256	Reserviert (IPv4-Kompatibilität)
0000 0010	0x0200	1/128	NSAP-Adressen
0000 0100	0x0400	1/128	IPX-Adressen
0010 0000	0x2000	1/8	Aggregatable Global Unicast Adressen
1111 1110 10	0xFE80	1/1024	Link-Local Unicast Adressen
1111 1110 11	0xFEC0	1/1024	Site-Local Unicast Adressen
1111 1111 00	0xFF00	1/256	Multicast-Adressen

IPv6 „Classful Networking“

- /12 Zuweisung der IANA an einen regionalen Internet Registrar (z.B. denic)
- /16 Zuweisung an die größten ISPs der Welt
- /32 Kleinste Zuteilung der RIPE RIR
- /48 Zuweisung an Unternehmen / Organisationen
- /56 Größe für kleine Unternehmen / Homeuser
- /64 Typische Größe eines Subnetzes
- /126 P2P Verbindungen

IPv6 Router im Eigenbau

Um im eigenen Netzwerk IPv6 zu verteilen, benötigt man nicht viel.

- Adresskonfiguration: Fest, statisch oder dynamisch
- Firewall: ip6tables / iptables (Tunnel Broker)
- DNS-Server (z.B. bind9, DNS64)
- Das Weiterleiten der IP-Pakete einschalten:
`sudo sysctl -w net.ipv6.conf.all.forwarding=1`
- Für IPv6-Only-Betrieb:
Routerseitiges NAT64 (tayga)

Manuelle Konfiguration mit ip

- Aktuelle Konfiguration anzeigen

```
ip addr show
```

```
wlan0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP  
qlen 1000 link/ether 24:76:13:db:5a:a4 brd  
ff:ff:ff:ff:ff:ff  
inet6 2a01:198:418:2::100:100/64 scope global  
valid_lft forever preferred_lft forever  
inet6 2a01:198:418:2:2676:13ff:fedb:5aa4/64 scope global dynamic  
valid_lft 86396sec preferred_lft 14396sec  
inet6 fe80::2676:13ff:fedb:5aa4/64 scope link  
valid_lft forever preferred_lft forever
```

- Adresse hinzufügen

```
ip addr add 2a01:198:418:3::100:100/64 dev wlan0
```

Jede Netzwerkkarte kann mehrere IP-Adressen besitzen.

- Direkte Verbindung zum nächsten Rechner/Router anzeigen

```
ip neigh show
```

```
fe80::240:e4ff:fea3:c1f4 dev wlan0 lladdr 00:40:e4:a3:c1:f4 router  
REACHABLE
```

- Privater Adressbereich:

```
fd00::/7
```


Automatische Konfiguration

Es existieren zwei Konfigurationsmöglichkeiten, die statische und/oder dynamisch Adressen vergeben können:

- Stateless Autoconfiguration
(neuer Standard)
- DHCPv6
(Angelehnt an alte DHCPv4-Funktionalitäten)

Stateless Autoconfiguration

- Client sendet Solicitation Message (ICMPv6)
- Alle Router im lokalen Netz antworten mit Router Advertisement Message
- Antwortmessage enthält Subnetz-Präfix für den Client
- Client baut aus Subnetz-Präfix und MAC-Adresse die eigene Interface-ID

Beispiel:

MAC-Adresse: 24:76:13:db:5a:a4

IPv6: 2a01:198:418:2:2676:13ff:fedb:5aa4

Stateless Autoconfiguration

- Neighbour Discovery Protocol (Teil von ICMPv6, Schicht 4)
- Auf dem Server muss der Daemon **radvd** installiert sein.
- Radvd antwortet auf Solicitation Messages
- Auf dem Client muss **rdnssd** installiert sein
rdnssd is a daemon program. Its purpose is to supply IPv6 DNS resolvers through stateless autoconfiguration, carried by Router Advertisements.

Radvd Configuration:

```
interface wlan1 {
    AdvSendAdvert on;
    prefix 2a01:0198:418:2::/48 {
        AdvOnLink on;
        AdvRouterAddr on;
        AdvAutonomous on;
    };
# clients {
#     fe80::2677:3ff:fedb:5ff4;
#     fe80::ee9b:5bff:fed3:7d04;
# };
RDNSS 2a01:198:418:2::80:80 {
    AdvRDNSSLifetime 30;
};
DNSSL private.keil-tech.de {
    AdvDNSSLLifetime 30;
};
```

DHCPv6

- Dynamic Host Configuration Protocol (UDP, Schicht 4)
- Auf dem Server muss ein DHCPv6-Server installiert sein.
Z.B. **wide-dhcpv6**
- DHCP-Server antwortet auf Clientanfragen.
- Funktioniert ähnlich wie in IPv4 (außer bei Netboot)

dhcp6s Configuration:

```
option domain-name-servers
2a01:0198:0418:0002::0080:0080;

option domain-name
"private.keil-tech.de";

interface wlan1 {
    address-pool    poolWlan1 3600;
};

pool poolWlan1 {
    range
2a01:0198:0418:0002::0100:0100 to
2a01:0198:0418:0002::0100:01ff;
};
```

IPv6 Adresse erhalten

Es gibt mehrere Möglichkeiten, an eine global geroutete IPv6-Adresse zu gelangen. Z.B.:

- Internet Provider mit IPv6-Unterstützung
(z.B. NetCologne)
- Tunnel Broker
(z.B. Sixxs)

NAT64

Müssen Internetseiten über IPv6 erreichbar sein, die nur eine IPv4-Adresse besitzen, dann kann man eine IPv6 / IPv4-Translation durchführen lassen

```
/etc/tayga.conf:  
tun-device nat64  
wan-device ppp0  
ipv4-addr 192.168.192.254  
ipv6-addr 2a01:0198:0418:ffff::1  
prefix  
2A01:0198:0418:AFFE::/96  
data-dir /var/spool/tayga
```

- Tayga verwendet Kerneleigene TUN-Devices um die einzelnen Netzwerke zu verbinden.

```
# ip addr show  
nat64: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500  
qdisc pfifo_fast state UP qlen 500  
link/none  
inet 192.168.192.254/32 scope global nat64  
inet6 2a01:198:418:ffff::1/128 scope global  
valid_lft forever preferred_lft forever
```

NAT64 & DNS64 für IPv4-Adressen

- Bind sucht nach IPv6-Adressen bei der Auflösung von Domainnamen
- Fehlt eine IPv6-Adresse, so wird die IPv4-Adresse in den IPv6-Adressraum gemappt
- Einfache Konfiguration von bind:

```
options {  
    [...]  
    dns64 2A01:0198:0418:AFFE::/96 {  
        clients { any; };  
    };  
};
```

IPsec

- RFCs 2401 bis 2409, RFC 4301
- Verschlüsselung & Authentisierung von Paketen
- Internet Key Exchange Protocol (IKE)
- IKEv2: RFC 5996
- Tunnel-Modus \Leftrightarrow Host-Modus
(aka Transport M.)

IPsec

- Authentication Header (AH)
Authentifiziert IPv6-Pakete
- Encapsulating Security Payload (ESP)
Verschlüsselung von Paketen

	AH	ESP V	ESP VA
Integriät der Daten	X		X
Sender Authentifizierung	X		X
Verschlüsselung der Daten		X	X

Mobile IPv6 (RFC 6275)

- Ähnlich wie VPN, verwendet IPsec
- z.B. Smartphone meldet aktuelle Adresse einem Homeagent zu Hause
- Homeagent empfängt Nachrichten für Handy und sendet in seinem Namen welche
- Homeagent im lokalen Netz „vertritt“ Client
- Handy/Smartphone ist mit seiner festen globalen Adresse im Internet erreichbar

Mobile IPv6

- UMIP (Userland Mobile IP, <http://umip.org>)
- Funktioniert erst ab Kernel 3.8.1 out-of-the-box
- Debian Packages (i386 only!):
deb <http://umip.org/debian/> unstable main
deb-src <http://umip.org/debian/> unstable main
- Für alle anderen Architekturen müssen Kernel und Userland-Tools kompiliert werden.
- Auf umip.org findet sich eine gute Installationsanleitung

Häh?

Fragen ?

Ah!

Danke!