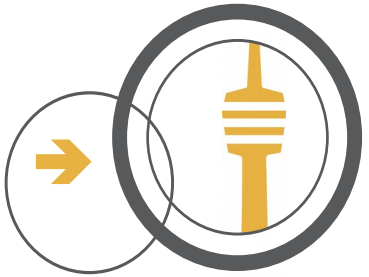
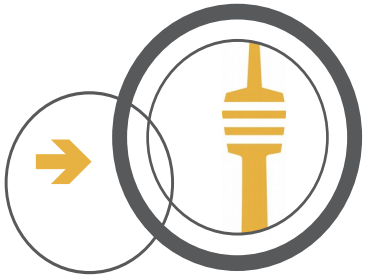
FFS Backbone



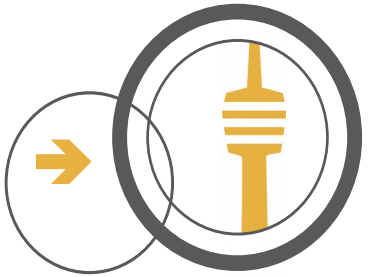
Oder:
Wie gut funktioniert ein Switch mit 3000 Ports verteilt über 900 Standorte

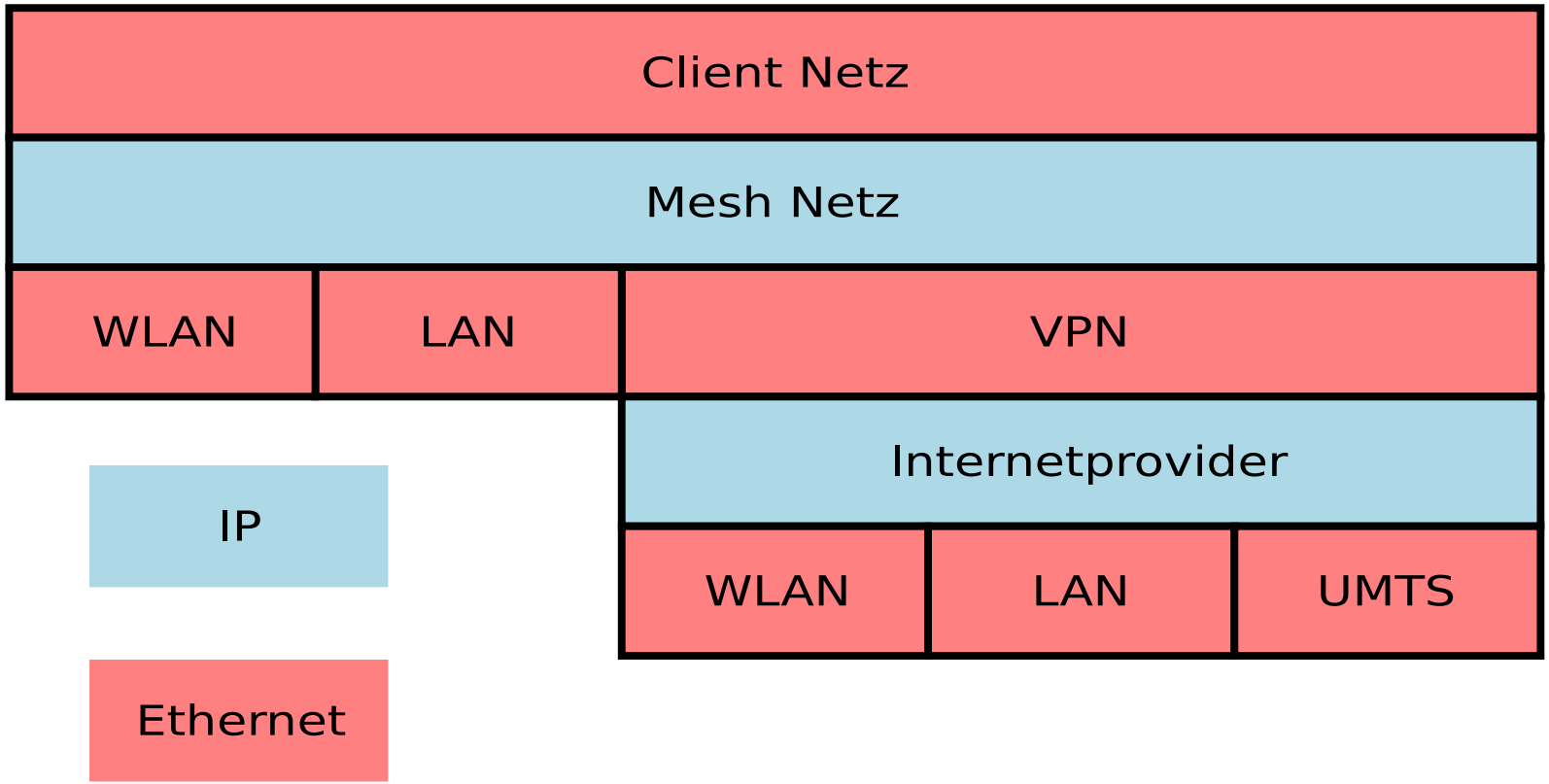
FFS Technik: B.A.T.M.A.N



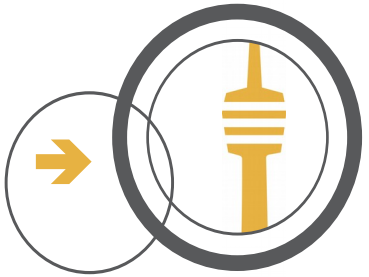
- Kommuniziert zwischen Nodes mit IPv6
- Baut aus den Nodes einen virtuellen Switch
- Alles Layer 2
  - Relevantes Protokoll ist Ethernet, nicht IP
  - Eine Broadcastdomain
    - ARP-Requests (100 in 2-3 Sekunden auf einem Node)
    - Netbios Namensauflösung
    - Avahi/Bonjour/Zeroconf
    - DHCP-Broadcasts
  - Firewall wäre nicht iptables, sondern ebtables
    - Firewall zwischen Switchports, nicht Netzen

„Netzwerkstack“



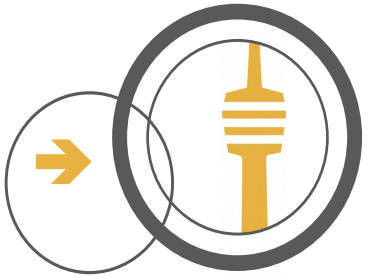


Layer 2 Vorteile



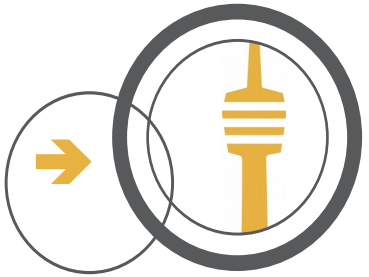
- LAN/WLAN wie zu Hause
- Clients können sich automatisch konfigurieren, z.B. DHCP, IPv6 Autoconfig
- Roaming zwischen Switchports (Nodes) ist möglich mit derselben IP

Layer 2 Nachteile



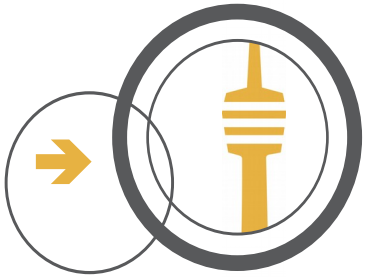
- Unnötige‘ Broadcasts gehen durchs ganze Netz (wo ist mein Drucker, wo ist mein Homeserver, wo ist mein Fernseher)
- Clients sehen das Freifunknetz als LAN an
- Die User wollen eigentlich Layer3, das gibt es aber nur als IPv6 einfach so.
- Anfällig für Layer2-Angriffe

Layer 2 mit B.A.T.M.A.N.



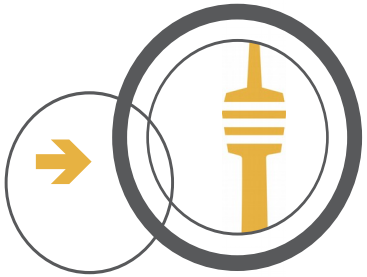
- Routing auf Layer 2, manche Broadcasts werden direkt an die richtigen Geräte geleitet statt an alle Ports
  - DHCP-Anfragen gehen nur zu B.A.T.M.A.N-Gateways
  - ARP-Anfragen werden in den Nodes gecached
- B.A.T.M.A.N. filtert manche Broadcasts um das Netz zu entlasten

Was ist ein Gateway?



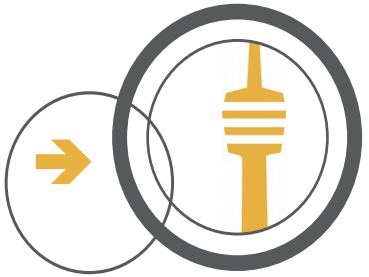
- Ähnlich einem normalen Node, aber
- B.A.T.M.A.N. läuft im Gateway Mode
- Gateway-Mode heisst im Wesentlichen, dass DHCP-Anfragen dort ankommen
- Ist normalerweise VPN-Server für Nodes
- Service Internet wird normalerweise durch Gateways angeboten

Problem:



- B.A.T.M.A.N.-Verwaltungsverkehr für 900 Nodes und 2700 Clients schluckt Bandbreite.
- Schmale Verbindungen können kaum noch für ‚echte‘ Aufgaben genutzt werden
- Freifunk mit Volumenbegrenzung geht nur kurz

Lösungsansatz:

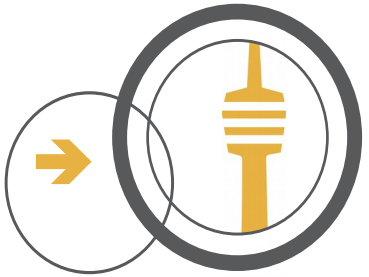


Segmentierung an den Gateways

- Separate VPNs für jedes Segment
- Separates Layer2
- Separate IP-Bereiche

Klingt einfach.

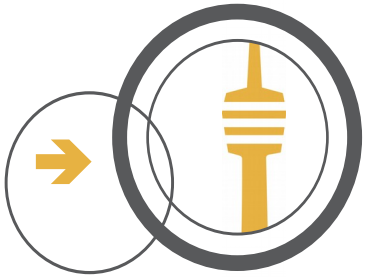
Roaming



Geht nicht mehr zwischen Segmenten

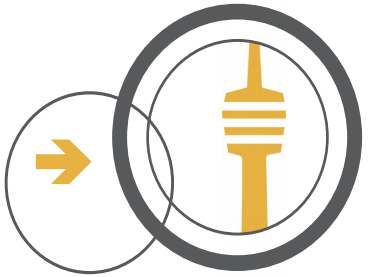
- Segmentgrenzen so setzen, dass das nichts aus macht
- Routing zwischen den Segmenten haben

Segmentgrenzenwahl



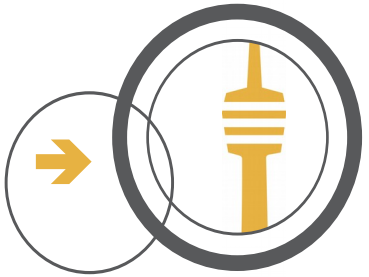
- Anhand geographischer Gegebenheiten, KFZ-Kennzeichen kommen da recht gut ran
- Ausgeglichene Verteilung von Nodes/Clients

Node → Segment

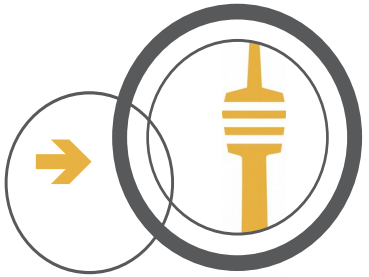


- http://gw01.freifunk-stuttgart.de/nodes.html
  - Wird erzeugt von einem Skript das die Geolocation auswertet
    - Bitte Koordinaten setzen, es reicht auch ungefähr
- Welche anderen Nodes meshen?
  - Verschiedene Skripte von Leuten auf gw-admins@
- Handarbeit im ersten Anlauf

Region	Nodes	Clients
Unknown Region	26	36
Hohenlohekreis	1	1
Alb-Donau-Kreis	1	0
Tuebingen	51	136
Calw	11	10
No Location	89	224
Neckar-Odenwald-Kreis	9	50
Rheinland-Pfalz	1	0
Goeppingen	21	66
Ludwigsburg	69	202
Ortenaukreis	2	1
Bodenseekreis	3	3
Rems-Murr-Kreis	116	572
Rottweil	6	40
Bayern	0	0
Saarland	0	0
Boeblingen	60	145
Hessen	0	0
Nordrhein-Westfalen	0	0
Reutlingen	65	203
Schwaebisch-Hall	0	0
Esslingen	72	209
Stuttgart	200	474
Pforzheim	1	1
Zollernalbkreis	0	0
Ostalbkreis	26	21
Heilbronn	15	24
Schwarzwald-Baar-Kreis	0	0

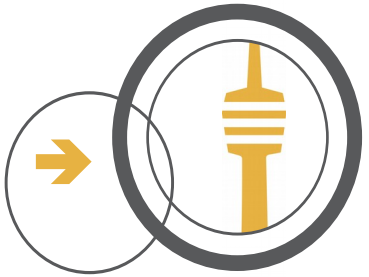


Node → Segment



- Aktuelle Firmware:
  - VPN-Keys für 10 Gateways
  - VPN-Konfigurationen für 5 verschiedene Ports
    - 1 Port = 1 Segment
- Gatewayseitig:
  - 5 verschiedene fastd-Instanzen für die VPNs
  - Darauf 5 verschiedene B.A.T.M.A.N.-Instanzen
  - Key für einen Node ist immer nur für eine fastd-Instanz gültig
    - Verbindungsaufbau dauert länger

Routing zwischen Segmenten

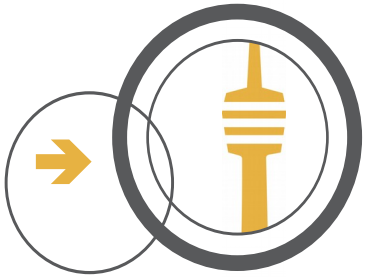


Wenn ein Gateway alle Segmente hat:

Einfach, kann TCP-IP schon immer:

```
# ip r l table stuttgart
default via 10.3.54.254 dev tun9
10.190.0.0/18 dev br01  scope link
10.190.64.0/18 dev br02  scope link
10.190.128.0/18 dev br03  scope link
10.190.192.0/18 dev br04  scope link
172.21.0.0/18 dev br00  scope link
```

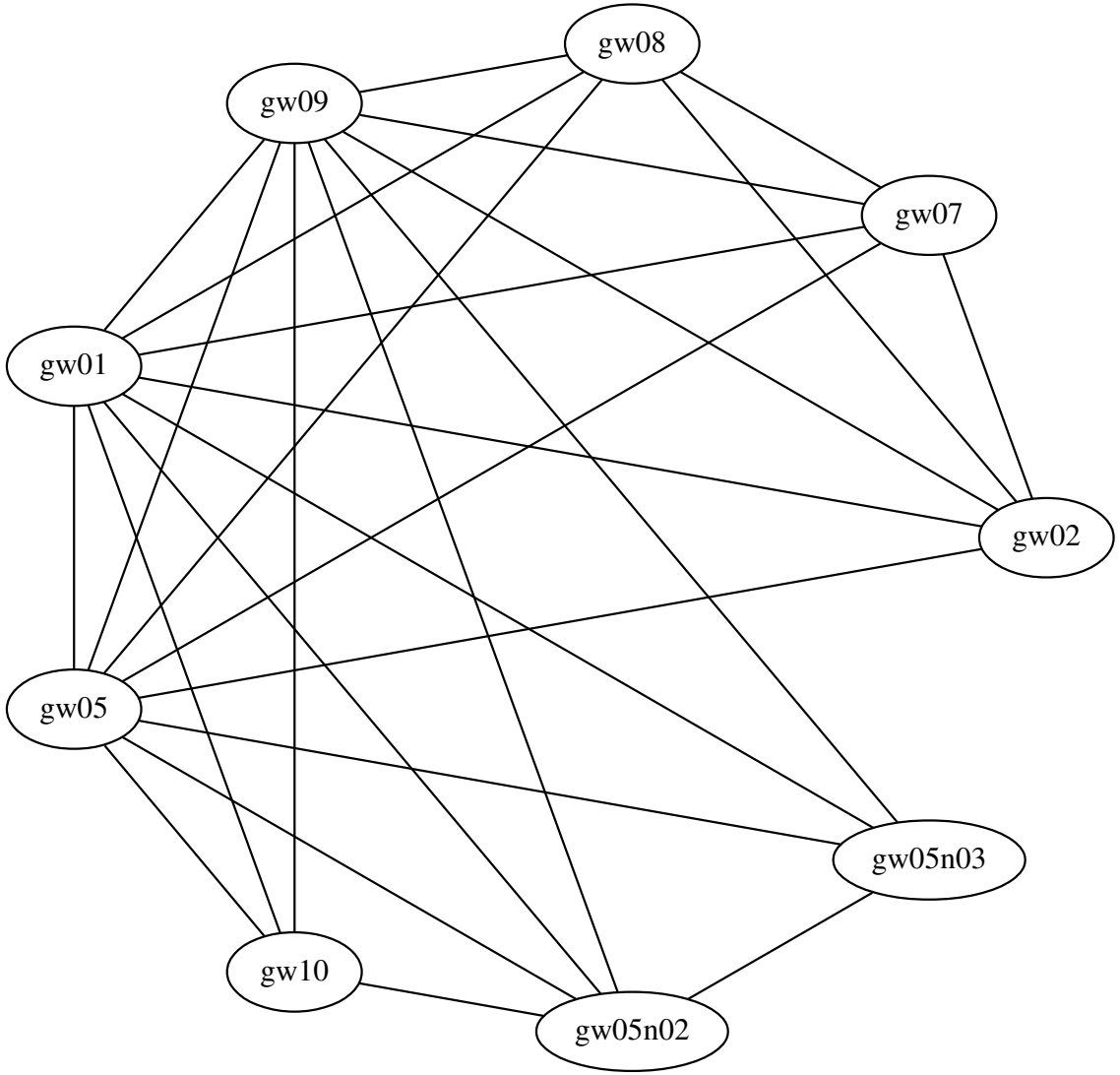
Routing zwischen Segmenten

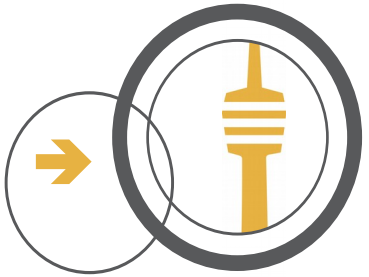


Wenn ein Gateway nicht alle Segmente hat:

- TINC-VPN ‚ffsbb‘ zwischen den Gateways
- Routingprotokoll OSPFv2 (IPv4), bzw. OSPFv3 (IPv6) im TINC-VPN

ffsbb TINC VPN

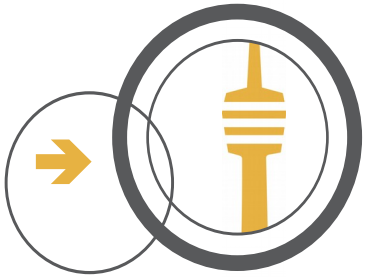




```
bird> show ospf state
area 0.0.0.0
        router 10.191.255.5
                distance 0
                network 10.191.255.0/24 metric 100
                stubnet 172.21.20.0/22 metric 10
                stubnet 10.190.24.0/21 metric 10
                stubnet 10.190.96.0/21 metric 10
                stubnet 10.190.160.0/21 metric 10
                stubnet 10.190.224.0/21 metric 10
                external 0.0.0.0/0 metric2 10000
                external 10.190.0.0/15 metric2 10000
                external 172.21.0.0/16 metric2 10000
        router 10.191.255.7
                distance 100
                network 10.191.255.0/24 metric 100
                stubnet 172.21.0.0/18 metric 10
                stubnet 10.190.0.0/18 metric 10
                stubnet 10.190.64.0/18 metric 10
                stubnet 10.190.128.0/18 metric 10
                stubnet 10.190.192.0/18 metric 10
```

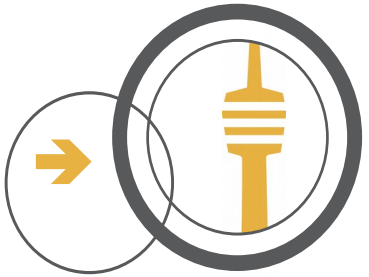
```
network 10.191.255.0/24
        dr 10.191.255.53
        distance 100
        router 10.191.255.53
        router 10.191.255.10
        router 10.191.255.7
        router 10.191.255.2
        router 10.191.255.5
        router 10.191.255.1
```

IPv4 Routing mit OSPFv2



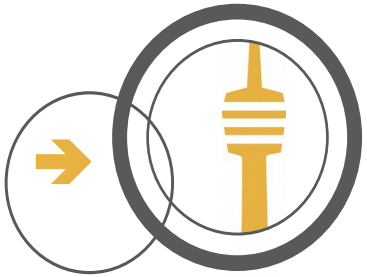
```
10.190.0.0/18 dev br01  scope link
10.190.0.0/18 via 10.191.255.7 dev ffsbb  proto bird  metric 100
10.190.0.0/15 via 172.21.28.1 dev br00
10.190.64.0/18 dev br02  scope link
10.190.64.0/18 via 10.191.255.7 dev ffsbb  proto bird  metric 100
10.190.128.0/18 dev br03  scope link
10.190.128.0/18 via 10.191.255.7 dev ffsbb  proto bird  metric 100
10.190.192.0/18 dev br04  scope link
10.190.192.0/18 via 10.191.255.7 dev ffsbb  proto bird  metric 100
172.21.0.0/18 via 10.191.255.7 dev ffsbb  proto bird  metric 100
172.21.0.0/16 dev br00  scope link
```

(zu erwartende) Probleme



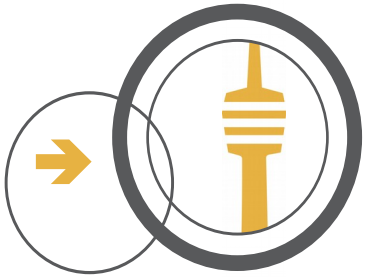
- Nodes verbinden zwei Segmente über Mesh → Segmentierungsvorteil ist weg, Nodes bekommen falsche IPs, schwache Router Rebooten wegen zu vielen IPv6-Routenannouncements
- Mit jedem zusätzlichen Segment dauert die mittlere Startupdauer eines VPN-Nodes länger
- IP-Vorrat reicht bei aktuellem Verbrauch nur für 8 Segmente, eine andere Aufteilung muss her

Segmentbrücken über Nodes



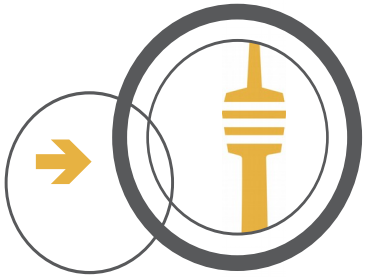
- Kill-Skripte die die betreffenden Nodes aus dem VPN werfen
- Im besten Fall auch gleich wieder im korrekten VPN eintragen
- WIP

VPN Startuplatenz reduzieren



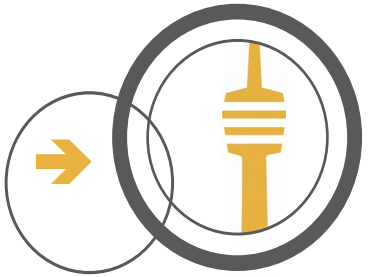
- Reduzierung der Gateways pro Segment, die ‚fehlenden‘ Gateways werden in diesem Segment auch aus dem DNS genommen.
- Sub-Gateways mit dem gleichen VPN-Schlüssel und DNS-Namen zur Lastverteilung
  - Wird derzeit erprobt mit gw05n02, gw05n03, gw072, gw082

IPs einsparen



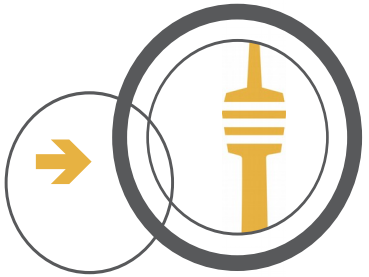
- B.A.T.M.A.N. reicht einen DHCP-Request nur an das nächste Gateway weiter → jedes Gateway muss alle potentiellen Clients mit IPs versorgen können, das können auch mal mehrere hundert an einem einzelnen VPN-Node sein (z.B. Schule mit VPN-Offloader)
- 1 DHCP-Server(-Cluster) für mehrere Gateways → der IP-Range des DHCP-Servers muss nur noch so groß sein wie alle Clients des Segmentes maximal benötigen. 1 DHCP-Server pro Segment reicht potentiell.
- Die derzeitigen Sub-Gateways sind alle so konfiguriert
- Bei 4k IPs/Segment würde das für 30 Segmente reichen

Besseren fastd haben



- Der kann beim Connect der Clients entscheiden an welches Netzwerkdevice (Segment) die gehören, d.h. es wird nur noch ein Port benötigt.

Auto-Segment Firmware



- VPN-Node hat keine VPN-Konfiguration per Default
- VPN-Node verbindet sich mit einem Konfigurationsserver und gibt dem alle Information die er hat
  - Mesh-Nachbarn, Koordinaten, Postleitzahl, Autokennzeichen, was der Betreiber halt angegeben hat
  - Konfigurationsserver antwortet mit Konfiguration für das passende Segment
- Aufwändige Implementierung, Nodes wären aber schneller online

Folienende