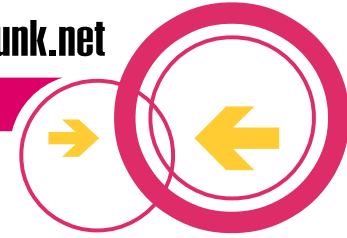


Freifunk

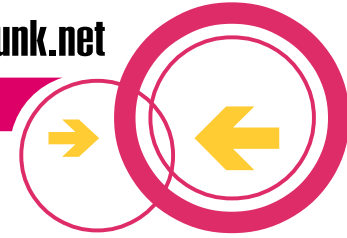




Heute

- Was ist das Freifunk?
- Wie gehts das?
- Exkurs: Mesh-Routing Protokolle
- Ziel: Gefühl bekommen wie das ganze technisch und theoretisch funktioniert.

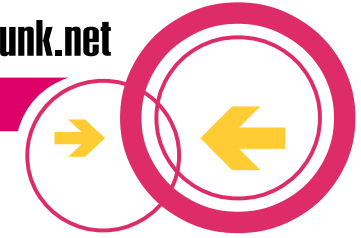




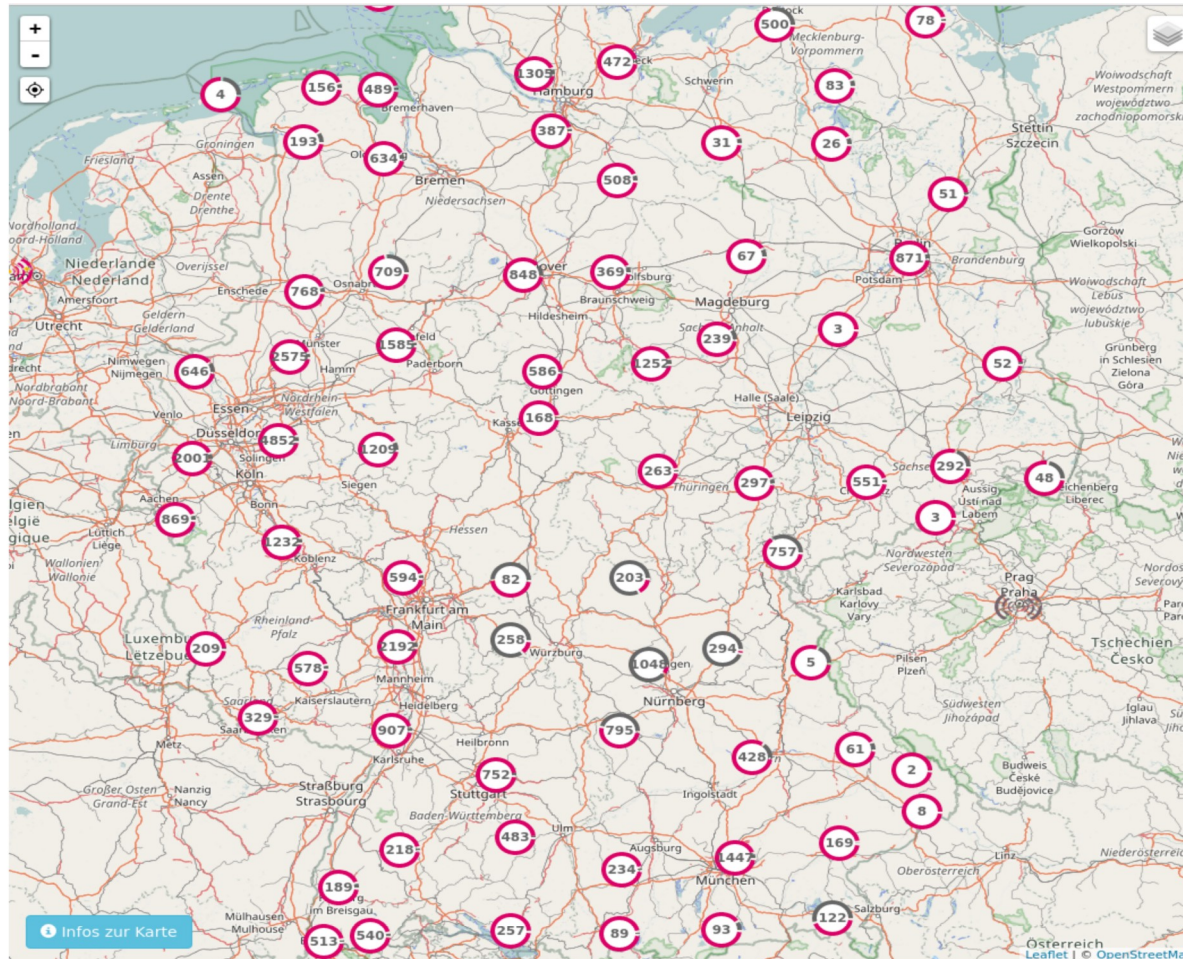
Was ist Freifunk?

- Seit 2003
- Dezentrale Initiative
- Nicht-Kommerziell
- Öffentliche Infrastruktur
- mit WLAN-Routern
- Sehr oft:
 - Meshrouting
 - Mit Internetzugang



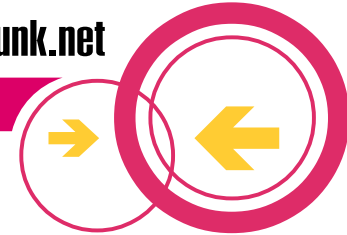


Freifunk-Router



Quelle: <https://www.freifunk-karte.de/>

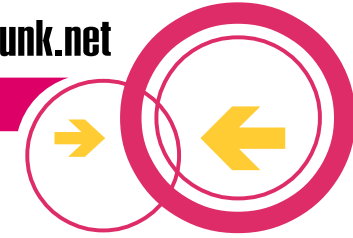




Hardware

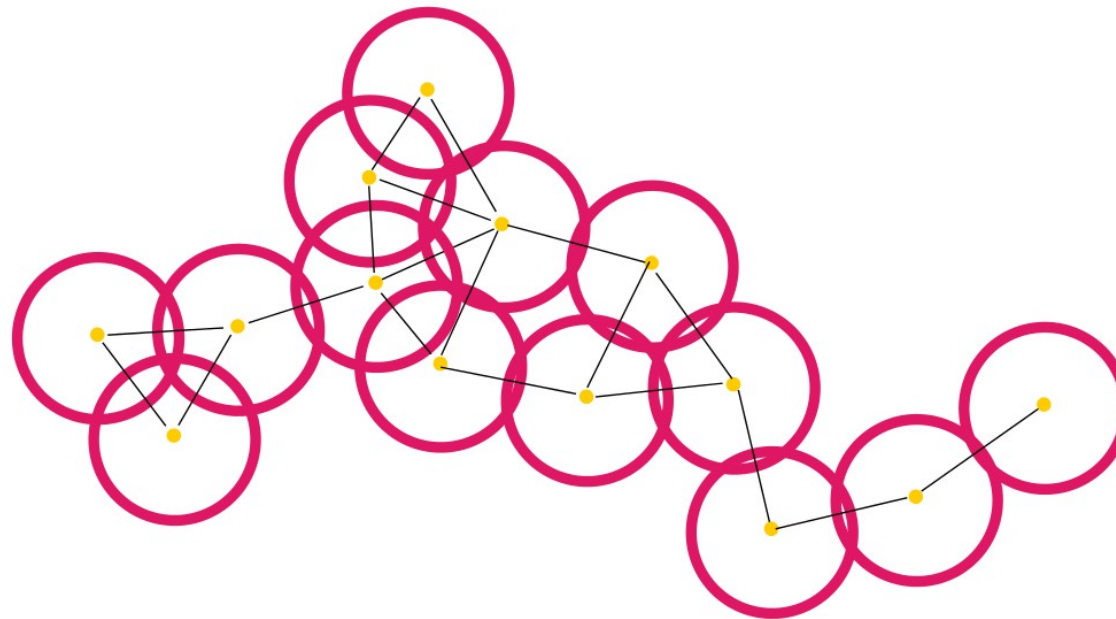
- Hauptsächlich “Plaste”-Router
 - z.B. tp-wr841
 - 4MB Flash, 32MB RAM, 650 MHz Mips
 - 2.4GHz WLAN
 - 5GHz eher weniger verbreitet wegen Treiber / Kosten / Verbreitung
- Richtfunkrouter
 - z.B. Ubiquiti Nanostation
 - Einige Kilometer Reichweite
- Server sind “normale” Angebote
 - z.B. 2Ghz, 2GB RAM, 25GB HDD, primär viel Traffic





WLAN-Mesh

Die Grundidee:

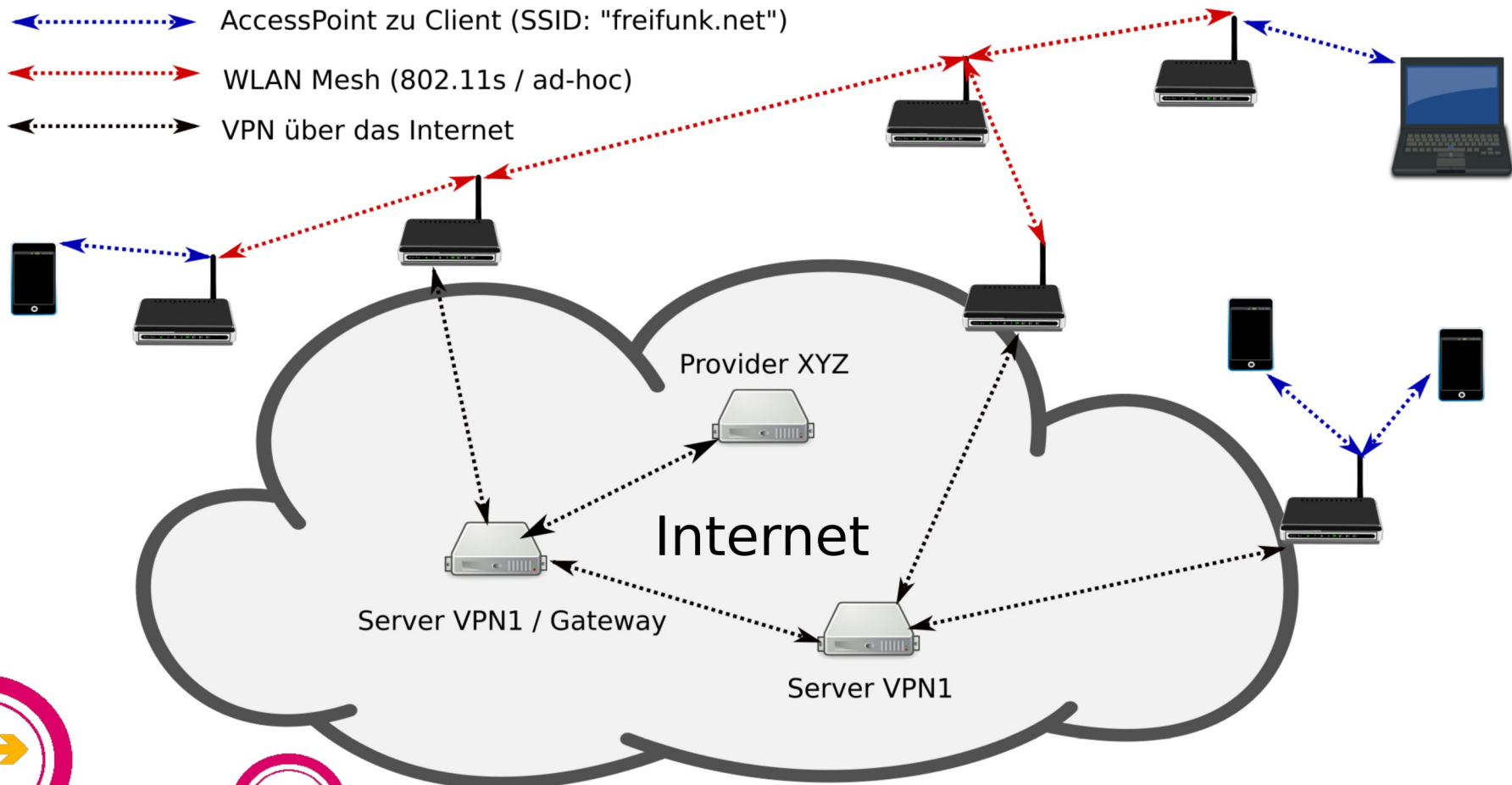
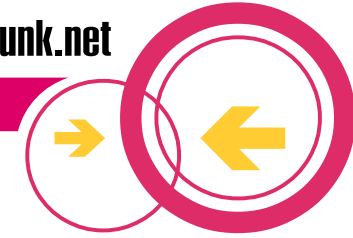


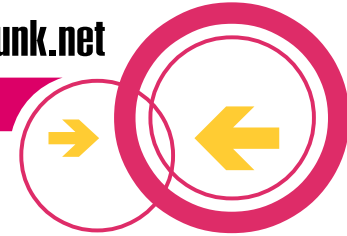
Nicht ganz realistisch, weil:

- Runde Sendebereiche
- Keine gegenseitige Überlappung



Netzwerkarchitektur

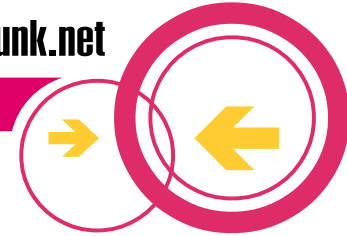




Netzstruktur

- Lokale und dezentrale Meshwolken
- Zentrale Backbonestruktur
 - Anders geht es z.Z. nicht
 - Wird eventuell mit IPv6 besser

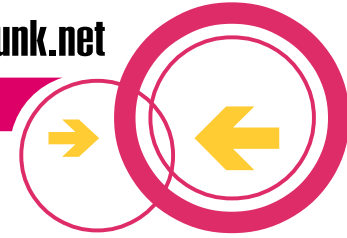




Adressvergabe

- IPv4-DHCP auf den Gateways
- IPv6 Router Advertisement daemon
 - z.B. radvd, selbst geschrieben (gluon-radvd) oder Teil des Mesh-Routing-Software
 - Läuft oft auf jedem Router
 - Gleiches Suffix an Clients
 - z.B. fdef:17a0:ffb1:300::/64
- Fast überall Pv4/IPv6 im Parallelbetrieb

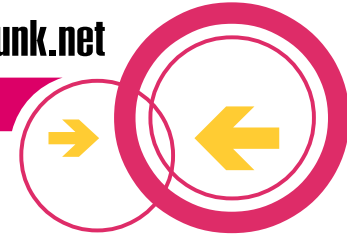




Konfiguration

- Teilweise sehr unterschiedlich
- Viele Communities experimentieren
- Hier nur ein Überblick der Komponenten und häufigen Konfiguration
- In der Realität komplexer
 - Effizienter und Leute wollen Spaß haben :>

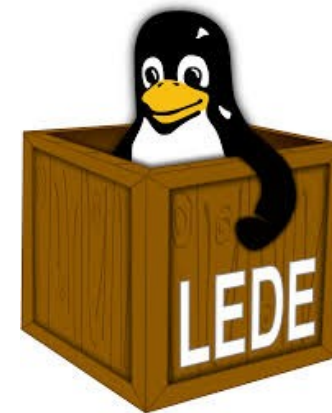


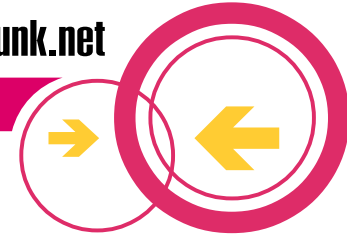


Router-Software (1)

- Betriebssystem
 - LEDE / OpenWrt
- Routing-Protokoll
 - OLSR, batman-adv
- VPN Program
 - Fastd
 - L2TP
 - OpenVPN

OpenWrt
Wireless Freedom

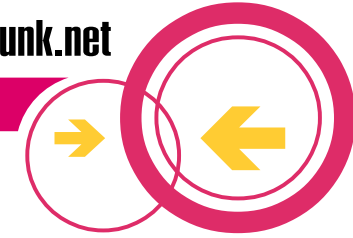




Router Software (2)

- Konfigurations-Tool
 - uci zum bequemen ändern der Konfigurationsdateien in /etc/config
- Adressvergabe:
 - Radvd, gluon-radvd, dnsmasq
- Webserver:
 - uhttpd + LuCI





LEDE / OpenWrt

- Linux Distribution für WLAN-Router
- Firmware bauen:

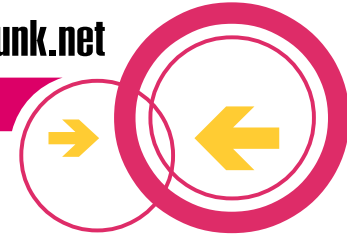
```
git clone git://git.lede-project.org/source.git  
cd source
```

```
./scripts/feeds update -a  
./scripts/feeds install -a
```

```
make defconfig  
make menuconfig
```

Live Demo!

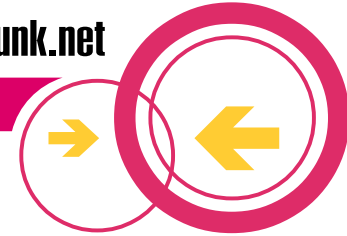




Fertige images

- Falls die Hersteller-Software drauf ist:
 - bin/targets/ar71xx/generic/lede-ar71xx-generic-tl-wr841-v9-squashfs-**factory**.bin
- Falls LEDE /OpenWrt bereits drauf ist:
 - bin/targets/ar71xx/generic/lede-ar71xx-generic-tl-wr841-v9-squashfs-**sysupgrade**.bin

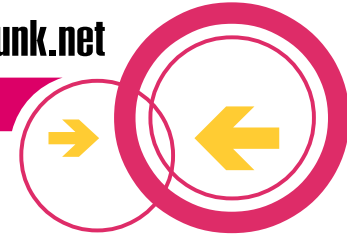




Zugriff

- LEDE
 - ssh `root@192.168.1.1`
- OpenWrt
 - telnet 192.168.1.1
 - ssh über `/etc/config/dropbear` dauerhaft aktivieren
- Im Fehlerfall Reset-Taste im richtigen Moment drücken
 - LED blinkt schnell
 - Sich 192.168.1.2/24 geben
 - Telnet 192.168.1.1

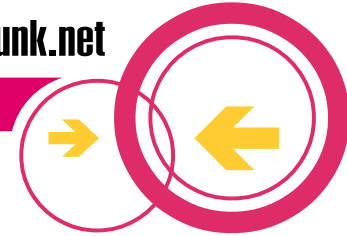




Router Konfigurieren

- /etc/config/firewall
 - Pakete blocken
- /etc/config/network
 - Netze und Adressen konfigurieren
- /etc/config/wireless
 - WLAN Anschlüsse (AP, Mesh)
 - SSID

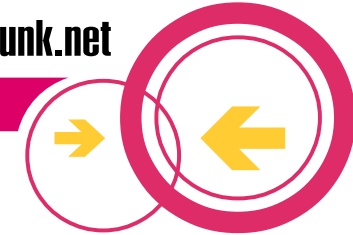




Dateien ins Image

- Dateien für das RootFS unter files/ im build-Ordner, z.B.:
 - files/etc/config
 - files/www/
- Landen im gebauten Image
- Scripte zum einmaligen Ausführen:
 - /etc/config/uci-defaults/



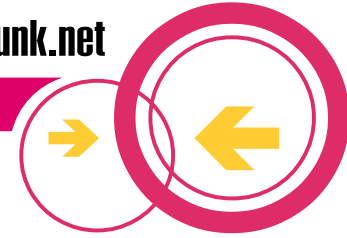


Router Konfiguration

- Firewall
 - Oft werden alle Broadcasts und fast alle Multicast-Pakete geblockt.
 - Ansonsten hoher Datenverkehr
 - Geschwätzige Geräte im FF-Netz; z.B. Apple-Geräte
- WLAN:
 - Ein AccessPoint Interface (z.B. SSID “<stadtname>.freifunk.net”)
 - Ein WLAN-Mesh Interface
 - 802.11s (mit eigenem Mesh-Protokoll) oder Ad-Hoc
- IP Adressen werden oft vom Router und Gateway vergeben.

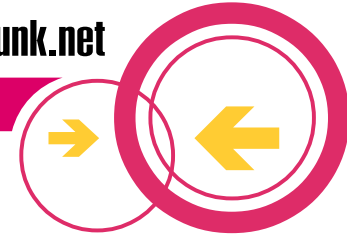


FF-Firmwarelandschaft



- Gluon
 - Weit verbreitet
 - Framework zu Firmware bauen
- Meshkit
 - Leipzig, Augsburg, ...
 - Webseite zum selber bauen von images über Profile
- Andere:
 - Ulm ;-)
 - Berlin
 - Viele andere

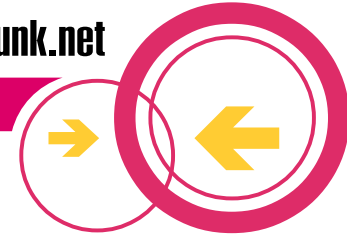




FF - Server / Gateway

- Über Server Im Internet sind die Router miteinander verbunden
 - VPN-Server
 - Oft auch Mesh-Routing Software
- Gateway ist ein Server der Internetzugang bietet
 - Über Schweden VPN, anderweitiges Ausland
 - Über Provider in Deutschland
 - z.B. Freifunk Rheinland ;-)

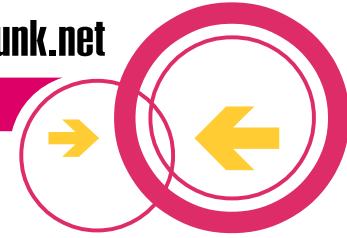




Sonstige Software

- (Dezentrale) Kartenlösungen
 - Mit alfred + meshviewer ([Link](#))
 - Server:
 - Alfred => Konverter => meshviewer
 - HTML+Javascript+3D.js
 - WLAN-Router:
 - Alfred
 - Andere ..
- Image Auswahl:
 - Gluon-Firmware-Wizard ([Link](#))
- DNS
 - Oft mit Bind, dem DNS-Server

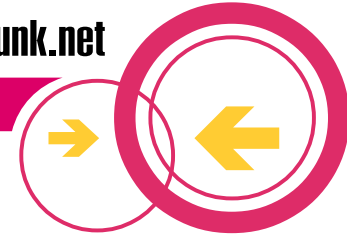




Meshrouting Allgemein

- Szenario
 - Knoten können willkürlich...
 - Sich bewegen
 - Aus- oder Angehen
 - Ändernde Verbindungsstärken
- Problemstellung:
 - Knoten A bekommt ein Paket das nach X möchte. Aber an welchen Nachbarn weiterreichen?

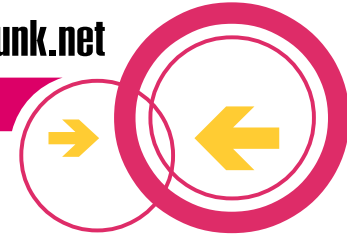




Meshrouting

- Viele Unterscheidungsmöglichkeiten
 - Layer 2 / 3
 - Proaktiv / Reaktiv
 - Distance Vector Routing / Link State / ...
- Weitere Eigenschaften
 - Keine Routing-Loops
 - Konvergent
 - Alle Knoten haben die Gleichen Informationen
 - Nutzdaten vs. Grundrauschen
 - Routing-Metrik (Durchsatz, Paketloss, Hop-Count)

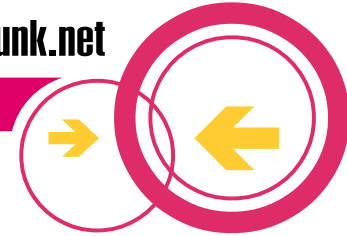




Grundrauschen

- Routing-Daten von einem Knoten werden an alle weitergeschickt.
 - “Hello”-Paket, OGM, Echolot
- Beim Mesh-Routing ist bisher oft bei ~1000 Knoten Schluss
 - Wesentlich mehr Verwaltungstraffic als Nutzdaten...
 - ... Segmentierung

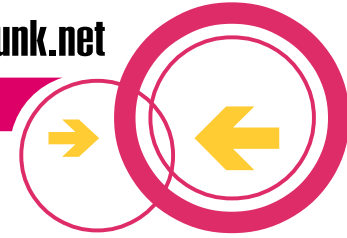




Layer 2 / 3

- Netzwerkebene die Simuliert wird
 - Layer 2
 - Es wird ein großer Netzwerkschicht simuliert
 - Ein Subnetz (z.B. 10.x.y.z)
 - Jeder Anschluss ein AccessPoint (freifunk.net)
 - Roaming geht aus dem Stand heraus
 - Layer 3
 - “Gebiete” werden in Subnetze unterteilt
 - Aber Subnetze müssen auch wieder verteilt werden
 - Für Roaming muss getrickst werden, geht aber auch

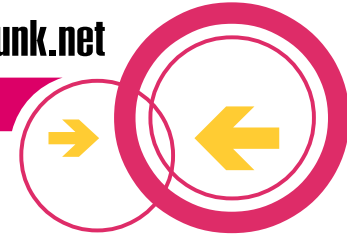




Proaktiv / Reaktiv

- Proaktiv
 - Alle nötigen Routinginformationen sammeln bevor ein Paket ankommt
 - Schnelle Routingentscheidung
 - Mehr Datenverkehr für Routing
- Reaktiv
 - Erst wenn ein Paket geroutet werden soll, werden die notwendigen Routinginformationen ermittelt
 - Langsame Routingentscheidungen
 - Wenig Datenverkehr



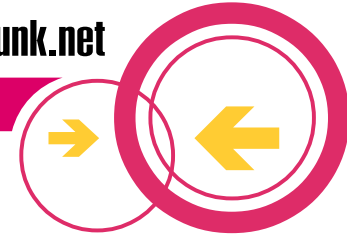


Protokoll-Kategorien

- Link State:
 - Knoten wissen die gesamte Topologie
 - Beispiel: OSPF, ISIS
- Distance Vector Routing:
 - Knoten wissen nur die Richtung
 - Beispiel: RIP, IGRP
- Szenarien:
 - Straßenkarte (jeder hat eine) – ok, schlechtes Beispiel :>
 - Kleine-Welt-Phänomen (jeder kennt nur sein Nachbarn)



Hinweis: Die Unterscheidung gibt es nicht nur bei Mesh-Routing Proto

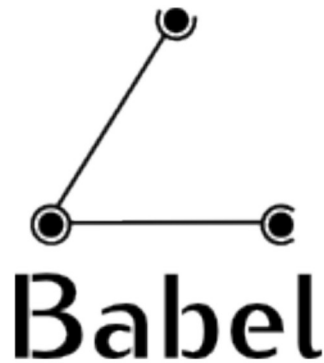


Mesh-Routing-Protokolle

```

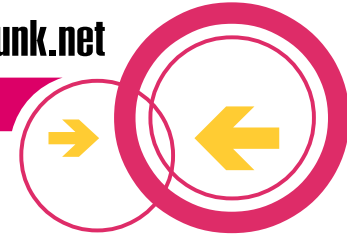
/* re-insert on candidate tree with the better metric */
new tc->path_stx = new stx; Insert into the global tc tree.
olsrd_sp_add_candidate(tc, cand_tree, new tc);
    avl_insert(&tc_tree, &tc->vertex_n
    and bump the hop count */
    * Initialize subtrees for edges
new tc->hops = tc->hops + 1; avl_init(&tc->prefix_tree, avl_co
  
```

olsrd
an adhoc wireless mesh routing daemon



- OLSR
 - Link State
 - Dijkstra-Algorithmus
 - Layer 3
 - Proaktiv
- Batman-adv
 - Distance Vector
 - Layer 2
 - Proaktiv
- Babel
 - Distance Vector
 - Bellman-Ford Algorithmus
 - Layer 3
 - Proaktiv

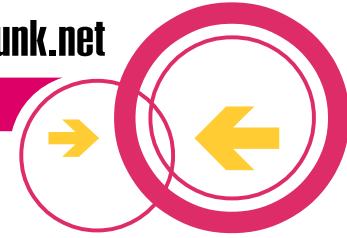




Zukunft

- OpenWrt => LEDE
- Ad-hoc => 802.11s
- Bessere skalierende Routing-Protokolle
 - Experimente mit Babel...
- Dezentrales Backbone *wish*

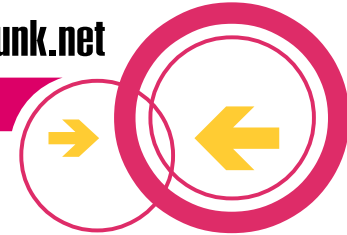




Fragen - Danke

???





Links

- Freifunk: <https://freifunk.net/>
- LEDE: <https://lede-project.org/>
- Batman-adv: <https://www.open-mesh.org/projects/batman-adv>
- Babel: <https://www.irif.fr/~jch/software/babel/>
- OLSR: <http://www.olsr.org>
- Fastd: <https://projects.universe-factory.net/projects/fastd/wiki>
- Alfred: <https://www.open-mesh.org/projects/alfred/wiki>
- Meshviewer: <https://github.com/ffnord/meshviewer>
- Gluon-Firmware-Wizard:
<https://github.com/freifunk-darmstadt/gluon-firmware-wizard>

