setting-up-B.A.T.M.A.N.-ADV-IV-batman-adv-on-Raspberry-Pis-4-Ubuntu-24.04 manual

TIM

March 2025

1 Implementierung eines Ad-hoc-Mesh-Netzwerks mit Batman-Adv

Implementierung des Ad-hoc-Mesh-Netzwerks mit dem Batman-Adv Netzwerkprotokoll in Ubuntu 24.04.2 LTS.

Zunächst erfolgt die Installation der benötigten Pakete:

```
$ sudo apt install iw
$ sudo apt install wireless-tools
$ sudo apt install ifupdown
$ sudo apt install net-tools
$ sudo apt-get install -y batctl
```

Zur Identifikation und überprüfung der vorhandenen Inferfaces und deren Einstellungen wird folgender Befehl ausgeführt:

```
$ iwconfig
```

```
eth0      no wireless extensions.

wlan0     IEEE 802.11  ESSID:off/any
          Mode:Managed  Access Point: Not-Associated   Tx-
              Power=off
          Retry short limit:7   RTS thr:off   Fragment thr:
              off
          Power Management:on
```

Um die Leistung des Ad-Hoc-Netzwerks zu erhöhen wird nun das Power Management auf off und die Sendeleistung auf 23 dBm (unterliegt Länderspezifischen Regularien siehe Technische Grundlagen) gestellt.

```
$ sudo iwconfig wlan0 power off
$ sudo iwconfig wlan0 txpower 23
```

Um sicherzustellen, dass der Systemstart nicht durch das Warten auf eine Netzwerkverbindung verzögert wird, kann der systemd-networkd-wait-online.service deaktiviert werden. Dieser Service ist standardmäßig aktiviert und wartet darauf, dass alle Netzwerkschnittstellen online sind, bevor der Boot-Vorgang fortgesetzt wird. Durch das Maskieren dieses Services wird verhindert, dass der Boot-Vorgang durch das Fehlen einer Netzwerkverbindung verzögert wird.

```
$ sudo systemctl mask systemd-networkd-wait-online.service
```

Mit dem folgenden Command wird das automatische Sperren und das Eintreten in den Ruhezustand deaktiviert:

```
$ sudo systemctl mask sleep.target suspend.target hibernate.
    target hybrid-sleep.target
```

Falls Ubuntu Dektop (beispielsweise zum Entwickeln) verwendet wird, muss vor dem Start von Batman-adv der Network-Manager von Ubuntu deinstalliert werden, da dieser sonst die Verwaltung der Netzwerkinterfaces übernimmt.

```
$ sudo apt-get rm network-manager
$ sudo apt-get purge network-manager
```

Um weiterhin Zugang zum Internet zu gewähren besteht die Möglichkeit der Netzwerkkonfiguration via Netplan. Hier ein Beispiel zur Konfiguration der eth0 Schnittstelle, alternativ kann eine weitere Wlan-Karte auch via Netplan konfiguriert werden:

```
$ sudo nano /etc/netplan/50eth0.yaml
```

Für DHCP (eth0):

```
network:
  version: 2
  renderer: networkd
  ethernets:
    eth0:
      dhcp4: yes
      dhcp6: yes
      optional: true
```

Beispiel für statische IPv4 addresse (eth0):

```
network:
    version: 2
    renderer: networkd
    ethernets:
       eth0:
         addresses: [192.168.137.50/24]
         gateway4: 192.168.137.1
         nameservers:
           addresses: [192.168.137.1, 8.8.8.8]
```

Beispiel für die Nutzung einer weiteren Wlan-Karte: wlan1 unter wifis wird durch den tatsächlichen Wlan-Interfacenamen ersetzt. "WIFI_SSID" durch tatsächlichen Namen des WLAN-Netzwerks ersetzen. "WIFI_PASSWORD" durch das tatsächliche Passwort des WLAN-Netzwerks ersetzen.

```
$ sudo nano /etc/netplan/50wlan1.yaml
```

```
network:
    version: 2
    renderer: networkd
    wifis:
       wlan1:
         dhcp4: yes
         dhcp6: yes
         access-points:
           WIFI_SSID:
            password: WIFI_PASSWORD
```

Es empfielt sich alle weiteren *.yaml-Dateien im Netplan Verzeichnis zu löschen, um Fehlerquellen zu vermeiden. Netzpan nimmt die Konfiguration an welche sich alphabetisch als letzte einordnet.

Es folgt das Anwenden von Netplan.

```
$ sudo netplan apply
```

1.0.1 Konfigurationsdateien in /etc/network/interfaces.d

Nun werden zwei Konfigurationsdateien angelegt. `wlan0` dient der Konfiguration des Netzwerkinterfaces. `wireless-channel 1`: Legt den Wi-Fi-Kanal fest, auf dem die drahtlose Kommunikation stattfindet. Mit dem folgenden Befehl können die belegten Kanäle angezeigt werden. Um die Leistung des Netzwerks zu erhöhen, empfiehlt es sich, in der Konfiguration des Ad-Hoc-Netzwerks einen Kanal zu verwenden, der wenig bis gar nicht genutzt wird und bei dem möglichst keine Überlappung mit anderen Kanälen stattfindet. Im Frequenzbereich von 2,4 GHz stehen zwar 13 Kanäle zur Verfügung, jedoch sind nur die Kanäle 1, 6 und 11 nicht überlappend nutzbar, da sie im 20-MHz-Kanalraster mindestens 25 MHz Abstand zueinander aufweisen. Mit den folgenden Befehlen werden Analysen zu bereits genutzten Frequenzen und Kanälen durchgeführt:

```
$ sudo iwlist wlan0 scan | grep "Channel"
$ sudo iwlist wlan0 scan | grep "Frequency\|Quality"
```

`wireless-essid my-swarm`: Setzt die ESSID (Netzwerkname) des drahtlosen Netzwerks. `wireless-mode ad-hoc`: Konfiguriert den Wi-Fi-Modus als Ad-Hoc-Netzwerk, was eine direkte Kommunikation zwischen Geräten ohne Access Point ermöglicht. `wireless-ap 02:12:34:56:78:9A`: Gibt die MAC-Adresse des Access Points (AP) an, mit dem sich das Gerät verbinden soll. `wlan0`

entspricht dem tatsächlichen Namen des WLAN-Interfaces und muss gegebenenfalls angepasst werden. Alle weiteren Einstellungen müssen auf allen Geräten identisch sein.

```
$ sudo nano /etc/network/interfaces.d/wlan0
```

```
auto wlan0
iface wlan0 inet6 manual
        wireless-channel 1
        wireless-essid my-swarm
        wireless-mode ad-hoc
        wireless-ap 02:12:34:56:78:9A
```

Der folgende Befehl weist den DHCP-Client (dhcpcd) an die Netzwerkschnittstelle nicht zu verwalten, eine wichtige Konfiguration.

```
echo 'denyinterfaces␣wlan0' | sudo tee --append /etc/dhcpcd.
    conf
```

bat0 dient der Konfiguration der virtuellen Schnitstelle für das BATMAN-Advanced-Protokoll. Eine statische IPv4 adresse wird bat0 in zeile 3 zugeteilt. Entsprechende Addresse für jedes Gerät, letzte Stelle, individuell anpassen. In Ziele 6 wird bat0 eine individuelle MAC-Adresse zugeteilt, für jedes Gerät, individuell anpassen (die Auflösung erfolgt in Hexadezimalzahlen). In Ziele 7 wird wlan0 (je WLAN-Interface anpassen) der virtuellen Schnitstelle bat0 hinzugefügt.

```
$ sudo nano /etc/network/interfaces.d/bat0
```

```
auto bat0
iface bat0 inet static
        address 192.168.123.2
        netmask 255.255.255.0
        gateway 192.168.123.1
        pre-up /usr/sbin/ip link set addr 12:14:90:a0:d0:3b
            dev $IFACE
        pre-up /usr/sbin/batctl if add wlan0
```

Beispielkonfiguration für ein weiteres Gerät:

```
auto bat0
iface bat0 inet static
        address 192.168.123.3
        netmask 255.255.255.0
        gateway 192.168.123.1
        pre-up /usr/sbin/ip link set addr 12:14:90:a0:d0:3c
            dev $IFACE
        pre-up /usr/sbin/batctl if add wlan0
```

Mit dem folgenden Befehl wird Batman-adv zu den Modulen hinzugefügt die beim booten gestartet werden:

```
echo 'batman-adv' | sudo tee --append /etc/modules
```

1.0.2 Skript zum Starten von Batman-adv

Nun wird ein Script zum Starten von Batman-adv erstellt.

```
$ sudo nano ~/start-batman-adv.sh
```

```
#!/bin/bash
sudo batctl if add wlan0
sudo ifconfig wlan0 up
sudo ifconfig bat0 up
```

Das Script wird mit folgendem Befehl ausführbar gemacht.

```
$ sudo chmod +x ~/start-batman-adv.sh
```

1.0.3 Integration von Batman-Adv in den Systemstart mittels Crontab

Damit Batman-Adv sich bei jedem Systemstart automatisch startet, wird folgende Zeile in crontab am Ende hinzugefügt:

```
$ sudo nano /etc/crontab
```

```
@reboot        root     /home/user/start-batman-adv.sh
```

user wird entsprechend dem tatsächlichen user des Homeverzeichnisses in dem das start-batman-adv.sh Script liegt angepasst.

Falls das WLAN-Interface nach dem Reboot auf Power Management:on zurücksetzt, empfielt sich das Einfügen der folgenden Zeile, oberhalb der Zeile die das start-batman-adv.sh ausführt, in crontab.

```
$ sudo nano /etc/crontab
```

```
@reboot        root     sudo iwconfig wlan0 power off
```

Abschließend folgt ein Neustart des Systems:

```
$ sudo reboot
```

1.0.4 Commands um funktionalität von Batman-adv zu überprüfen

Testen ob WLAN-Interface von Batman-adv erkannt wird:

```
$ sudo batctl if
```

Wenn das Interface richtig erkannt wurde erscheint:

```
wlan0: active
```

Testen ob Nachbar gefunden wird:

```
$ sudo batctl n
```

Wenn ein Nachbar gefunden wird:

```
[B.A.T.M.A.N. adv 2024.2, MainIF/MAC: wlan0/12:34:56:78:9a:
    bc (bat0/12:34:56:78:9a:3a BATMAN_IV)]
IF             Neighbor              last-seen
wlan0          12:34:56:78:9a:3b     0.870s
```

1.1 Debugging mit Wireshark

Um zu überprüfen, ob andere Geräte im Netzwerk B.A.T.M.A.N. Advanced IV-Pakete versenden, kann Wireshark verwendet werden. Im Folgenden wird beschrieben, wie das WLAN-Interface in den Monitor-Modus versetzt und die Paketerfassung mit Wireshark durchgeführt wird, um die Kommunikation zu analysieren.

Die Installation von Wireshark erfolgt mit:

```
$ sudo apt install wireshark
```

Es erscheint ein Dialogfenster zur Konfiguration von wireshark-common. Um Fehlerquellen zu vermeiden empfielt es sich non-superusers das Erfassen von Paketen zu erlauben, was allerdings zu Sicherheitsrisiken führen kann.

1.1.1 Stoppen und Maskieren des NetworkManagers

Falls das Networking auf dem Gerät zum Debugging vom NetworkManager verwaltet wird, muss dieser zunächst gestoppt und gemaskt werden, damit er nicht erneut automatisch startet und das WLAN-Interface managed:

```
$ sudo systemctl stop NetworkManager
$ sudo systemctl mask NetworkManager
```

Zur Überprüfung ob der NetworkManager gestoppt und gemaskt ist wird folgender Command ausgeführt:

```
$ sudo systemctl status NetworkManager
```

```
O NetworkManager.service
     Loaded: masked (Reason: Unit NetworkManager.service is
         masked.)
     Active: inactive (dead) since Sun 2025-03-23 20:14:49
         CET; 27s ago
   Duration: 12min 9.728s
   Main PID: 1028949 (code=exited, status=0/SUCCESS)
```

```
        CPU: 667ms

Mar 23 20:14:49 t-ThinkPad-T460s NetworkManager[1028949]: <
    info>  [1742757289.3>
Mar 23 20:14:49 t-ThinkPad-T460s NetworkManager[1028949]: <
    info>  [1742757289.3>
Mar 23 20:14:49 t-ThinkPad-T460s NetworkManager[1028949]: <
    info>  [1742757289.3>
Mar 23 20:14:49 t-ThinkPad-T460s NetworkManager[1028949]: <
    info>  [1742757289.3>
Mar 23 20:14:49 t-ThinkPad-T460s NetworkManager[1028949]: <
    info>  [1742757289.3>
Mar 23 20:14:49 t-ThinkPad-T460s NetworkManager[1028949]: <
    info>  [1742757289.4>
Mar 23 20:14:49 t-ThinkPad-T460s NetworkManager[1028949]: <
    info>  [1742757289.4>
Mar 23 20:14:49 t-ThinkPad-T460s NetworkManager[1028949]: <
    info>  [1742757289.5>
Mar 23 20:14:49 t-ThinkPad-T460s systemd[1]: NetworkManager.
    service: Deactivate>
Mar 23 20:14:49 t-ThinkPad-T460s systemd[1]: Stopped
    NetworkManager.service - N>
```

Mit folgenden Befehlen kann der NetworkManager nach dem Nutzen von Wireshark und dem zurücksetzten des WLAN-Interfaces (in mode:managed) wieder geunmasked und gestartet werden:

```
$ sudo systemctl unmask NetworkManager
$ sudo systemctl start NetworkManager
```

1.1.2 B.A.T.M.A.N. Advanced IV-Pakete mit Wireshark suchen

Um Netzwerkpakete mit Wireshark zu erfassen, muss das WLAN-Interface in den Monitor-Modus versetzt werden. "wlan0" wird durch den tatsächlichen WLAN-Interface ersetzt.

```
$ sudo ip link set wlan0 down
$ sudo iwconfig wlan0 mode monitor
$ sudo ip link set wlan0 up
```

Überprüfung ob das Interface tatsächlich in den Monitormode versetzt wurde mit:

```
$ iwconfig wlan0
```

```
wlan0     IEEE 802.11  Mode:Monitor
          Retry short limit:7   RTS thr:off   Fragment thr:
              off
          Power Management:off
```

Wireshark starten mit:

```
$ sudo wireshark
```

Auf der Grafischen Oberfläche von Wireshark das WLAN-Interface wählen und eine weile warten. Falls mindestens ein Gerät B.A.T.M.A.N. Advanced IV-Pakete sendet sollten erscheint der folgende Output. Dieser kann unter anderem auf die MAC-Addresse des Sendenden Interfaces aufschluss geben.

Figure 1: Wireshark Output

Um das Interface des Geräts zum Debuggen wieder in den vorherigen Zustand zurückzusetzen werden folgende Befehle ausgeführt:

```
$ sudo ip link set wlan0 down
$ sudo iwconfig wlan0 mode managed
$ sudo ip link set wlan0 up
```

Falls das Networking des Geräts mit dem NetworkManager verwaltet wird, wird dieser wieder geunmasked und gestartet (siehe oben).

2 SSH-Zugriff via B.A.T.M.A.N. Advanced IV

Die statische IPv4 Adresse wurde in der Konfigurationsdatei /etc/network/interfaces.d/bat0 festgelegt. Nun wird überprüft, ob sich die Geräte im Mesh-Netzwerk problemlos anpingen können:

```
$ ping 192.168.123.3
```

Beispielhafter Output:

```
PING 192.168.123.3 (192.168.123.3) 56(84) bytes of data.
64 bytes from 192.168.123.3: icmp_seq=1 ttl=64 time=1.23 ms
64 bytes from 192.168.123.3: icmp_seq=2 ttl=64 time=0.987 ms
64 bytes from 192.168.123.3: icmp_seq=3 ttl=64 time=1.05 ms
64 bytes from 192.168.123.3: icmp_seq=4 ttl=64 time=1.12 ms
```

```
--- 192.168.123.3 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 4006
    ms
rtt min/avg/max/mdev = 0.956/1.068/1.230/0.103 ms
```

Es ist eine Grundvoraussetzug, dass der SSH-Server auf dem Zielgerät installiert ist um SSH-Zugriff zu erlangen.

Mit folgendem Befehl wird eine SSH-Verbindung aufgebaut (user mit tatsächlichem Benutzer auf Zielgerät ersetzen):

```
$ ssh user@192.168.123.3
```

Es erfolgt eine SSH-Fingerprint-Abfrage:

```
The authenticity of host '192.168.123.3 (192.168.123.3)' can't be established.
ECDSA key fingerprint is SHA256:AbCdEfGhIjKlMnOpQrStUvWxYz0123456789ZaBcDeFgHi.
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

Es wird mit yes bestätigt und der Fingerabdruck des Geräts wird in der Datei /.ssh/known_hosts gespeichert, und zukünftige Verbindungen zu diesem Server werden nicht mehr diese Abfrage anzeigen.

Es folgt die Passwort abfrage von user:

```
$ user@192.168.123.3's password:
```

Nach der Eingabe des richtigen Passworts wird der Zugang zur Shell des Serwers sichtbar. Beispielhafte Ansicht:

```
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.1.0-raspi aarch64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

Last login: Mon Mar  24 12:34:56 2025 from 192.168.123.2
user@ubuntu:~$
```