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

TIM

March 2025

1 Implementation of an Ad-hoc Mesh Network with Batman-Adv

Implementation of the Ad-hoc Mesh Network using the Batman-Adv network protocol in Ubuntu 24.04.2 LTS.

First, the required packages are installed:

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

To identify and verify existing interfaces and their settings, the following command is executed:

```
$ 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
```

To improve the performance of the Ad-Hoc network, power management is turned off and the transmit power is set to 23 dBm (subject to country-specific regulations, see Technical Basics).

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

To ensure that the system startup is not delayed by waiting for a network connection, the systemd-networkd-wait-online.service can be disabled:

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

The following command disables automatic locking and entering sleep mode:

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

If Ubuntu Desktop is used (for example, for development purposes), the Network Manager must be uninstalled before starting Batman-adv:

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

To still provide internet access, network configuration via Netplan is possible. Here is an example for configuring the eth0 interface:

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

For DHCP (eth0):

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

Example for a static IPv4 address (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]
```

Example for using an additional WLAN card:

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

```
network:
    version: 2
    renderer: networkd
    wifis:
       wlan1:
         dhcp4: yes
```

```
        dhcp6: yes
        access-points:
          WIFI_SSID:
           password: WIFI_PASSWORD
```

Apply the Netplan configuration:

```
$ sudo netplan apply
```

1.0.1 Configuration files in /etc/network/interfaces.d

Create the wlan0 configuration file:

```
$ 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
```

Configure DHCP client to not manage the interface:

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

Create the bat0 configuration file:

```
$ 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
```

Example configuration for another device:

```
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
```

Add batman-adv to startup modules:

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

1.0.2 Script to start Batman-adv

Create the startup script:

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

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

Make the script executable:

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

1.0.3 Integration into system startup using Crontab

Add to crontab:

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

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

Optional power management setting in crontab:

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

Reboot the system:

```
$ sudo reboot
```

1.0.4 Commands to verify Batman-adv functionality

Check interface recognition:

```
$ sudo batctl if
```

Expected output:

```
wlan0: active
```

Check for neighbors:

```
$ sudo batctl n
```

Example neighbor output:

```
[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 with Wireshark

Install Wireshark:

```
$ sudo apt install wireshark
```

1.1.1 Stopping and Masking NetworkManager

Stop and mask NetworkManager:

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

Check status:

```
$ sudo systemctl status NetworkManager
```

```
○ 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
```

Restore NetworkManager after debugging:

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

1.1.2 Capturing B.A.T.M.A.N. packets

Set interface to monitor mode:

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

Verify mode:

```
$ iwconfig wlan0
```

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

Start Wireshark:

```
$ sudo wireshark
```

Restore interface settings:

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

2 SSH Access via B.A.T.M.A.N. Advanced IV

Test connectivity:

```
$ ping 192.168.123.3
```

Example ping 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
```

Establish SSH connection:

```
$ ssh user@192.168.123.3
```

SSH fingerprint verification:

```
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])?
```

Password prompt:

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

Successful login example:

```
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:~$
```