universidade de aveiro
theoria poiesis praxis

DEPARTAMENTO DE ELECTRÓNICA, TELECOMUNICAÇÕES E INFORMÁTICA
MESTRADO EM ENG. DE COMPUTADORES E TELEMÁTICA
ANO 2023/2024

REDES E SISTEMAS AUTÓNOMOS

AUTONOMOUS NETWORKS AND SYSTEMS

PRACTICAL GUIDE 2 – B.A.T.M.A.N.

Objectives

- Set up B.A.T.M.A.N. ad-hoc network
- Validate the multi-hop ad-hoc network with simple tests
- Observe the status of the network
- Try new topologies of the ad-hoc network to force different paths
- Analyse metrics of the network

Duration

2 weeks

1st week

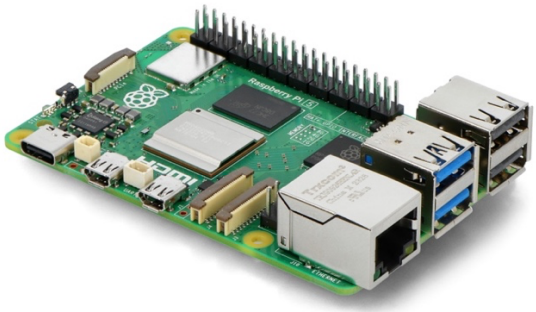
Introduction

The Better Approach to Mobile Ad-hoc Networking (B.A.T.M.A.N.) is a protocol for multi-hop mobile **ad hoc networks**. The official documentation is available here.

In this practical guide, we will learn how to deploy a Wi-Fi ad-hoc network with support for multi-hop. Supporting more than one hop means the network will behave like a mesh network, where the nodes can reach other nodes that are directly or indirectly connected, if they are part of the same batman group and Wi-Fi channel.

Environment

The setup we will be mounting for the experience uses Raspberry Pi (RPi) 5 4GB, and there is **one available per student**:



Raspberry Pi 5 4GB Dev Kit

Each RPi is identified with a sticker, example "NAP 706". For this example, the **id** of the device is **6**.

Every RPi is already flashed with the official Raspbian and it has already some pre-configuration (the batman binary is installed).

In terms of networking and authentication, the board is configured with the following:

- Hostname: raspberrypi-7**id**
- Ethernet interface: 192.168.3.**id** (it also accepts DHCP)
- SSH login
    - Username: nap
    - Password: openlab

The internal WiFi chipset of the RPi will be used for the batman setup. The important thing about this chipset is that it allows the configuration of the Wi-Fi interface in **IBSS mode** which is required for the configuration of ad-hoc networks.

Form groups of 4 people to perform this guide.

1. Connecting your PC to the RPi board

1.1. Configure **your PC** with an IP in the **same LAN** of the RPi Ethernet interface (192.168.3.0/24) and connect an Ethernet cable between your PC and RPi.

Example: 192.168.3.1/24 (**no** gateway needed)

1.2. Use **ssh** to connect to the RPi, using the IP and authentication described in the previous page. In Windows use Putty or ssh through WSL. In Linux use ssh directly.

2. Configure the Wi-Fi interface in preparation for the batman network.

2.1. Run the following commands, so the wpa_supplicant service doesn't interfere with the configuration we will do for batman:

```
sudo service wpa_supplicant stop
sudo systemctl mask wpa_supplicant.service
sudo killall wpa_supplicant
```

2.2. Change the Wi-Fi interface mode to IBSS

(check the Wi-Fi interface name (`ip a`), and use it to **replace** with the right name in the following commands)

```
sudo ip link set wlanX down

sudo iw wlanX set type ibss

sudo ifconfig wlanX mtu 1500
```

2.3. Select the right Wi-Fi channel. Every group with 4 boards will create a separated ad-hoc network. Select channels among the following list [3, 5, 7, 9, 11]

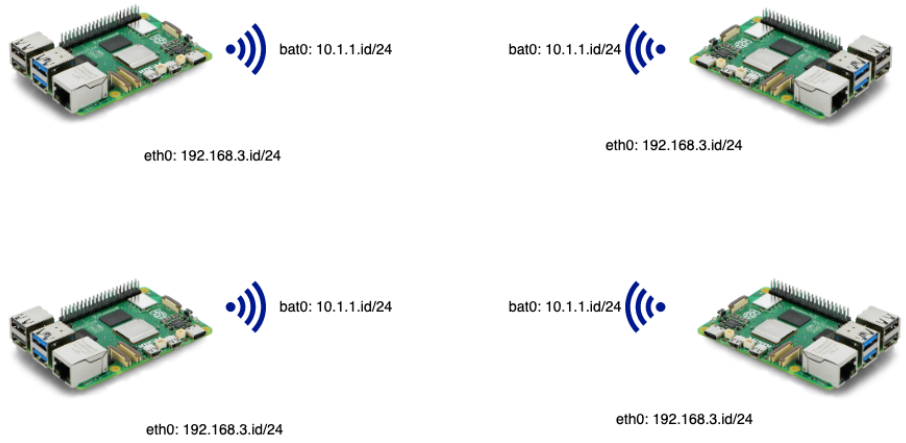
```
sudo iwconfig wlanX channel Y

sudo ip link set wlanX up
```

2.4. Confirm if the Wi-Fi interface is in the correct mode (**ibss**), with the following command:

```
sudo iw dev
```

3. First batman network configuration

3.1. The network of every group will look like the following figure:



3.2. Find out (**search online**) what is the frequency (**f0**), **in MHz**, for the Wi-Fi **channel** selected in the configuration of exercise **2.3**.

3.3. Then, invent a **network name** for your batman network.

3.4. On your RPi, run the following command, according to the frequency and the chosen ad-hoc network name:

```
sudo iw wlanX ibss join network_name f0
```

3.5. Finalize the batman configuration with the following:

```
sudo modprobe batman-adv
```

```
sudo batctl if add wlanX
```

```
sudo ip link set up dev wlanX
```

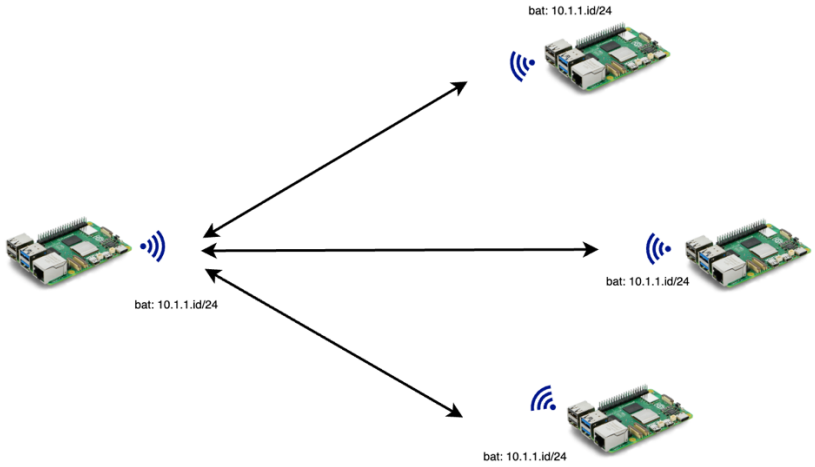
```
sudo ip link set up dev bat0
```

```
sudo ifconfig bat0 10.1.1.id/24
```

(Configure the IP for the batman LAN, according to the **id** of your board)

4. Testing the batman network for the first time

When the boards are placed together and close enough, each board connects to the other in a direct way, as seen in the following figure from the perspective of one of the RPis:



4.1. Together with the colleagues from the other groups, test the wireless connection to their RPis, using the ping command. Validate that you can reach every node of the ad-hoc network.

`sudo batctl ping 10.1.1.X` (layer 2 test)

`ping 10.1.1.X`

4.2. Use the following to check the neighbors table. The command `watch` is to keep observing in real time:

```
sudo watch -n 0.1 batctl n
```

Example of output:

```
Every 0.1s: batctl n
raspberrypi-706: Thu Mar 16 19:52:03 2023[B.A.T.M.A.N. adv
2021.3…
IF             Neighbor              last-seen
   wlan0     24:ec:99:b4:0f:a6    0.444s
   wlan0     24:ec:99:8b:e7:08    0.128s
```

2nd week

5. Automatic configuration of batman

5.1. To allow an easier configuration of batman from now on, modify the following script (with vim/nano) with the same settings used before in exercises **2.** and **3.** **(channel, network name, and frequency):**

Go to `~/batman_installation` and run the following to edit the script:

```
vi create_batman_interface.sh
```

5.2. After booting of the board, now you only need to run the script to configure batman:

```
./batman_installation/create_batman_interface.sh wlan0 \
10.1.1.id
```

6. Analysis of the batman topology and conditions

6.1. Utilizing the ssh connection to each RPi, save the **MAC address** of the wireless interface (wlan0) and map it with the id of the board, so you can identify the devices.

6.2. Use again the command to check the neighbors table. Confirm in the neighbors table that the registered boards of your mesh are the ones appearing there.

```
sudo watch -n 0.1 batctl n
```

6.3. Now run the command to check the **originators table**. This command lists all other nodes in the network and **registers which direction to send the packets** (marked with *). Identify who is the nexthop. In the current topology, all should be direct.

```
sudo watch -n 0.1 batctl o
```

Example of output:

```
Every 0.1s: batctl o              jetson-14: Thu Mar 16 19:53:05
2023[B.A.T.M.A.N. adv 2021.3…
Originator       last-seen (#/255) Nexthop      [outgoingIF]
  24:ec:99:8b:e7:08 0.696s ( 12) 24:ec:99:b4:0f:a6 [ wlan0]
* 24:ec:99:8b:e7:08 0.696s ( 43) 24:ec:99:8b:e7:08 [ wlan0]
  24:ec:99:b4:0f:a6 0.756s ( 10) 24:ec:99:8b:e7:08 [ wlan0]
* 24:ec:99:b4:0f:a6 0.756s ( 78) 24:ec:99:b4:0f:a6 [ wlan0]
```

The column `#/255` is the sequence number.

6.4. Use the space of the classroom beyond the table, and also the **power cable (USB-C) directly to your laptop for at least one RPi** so you can carry it far away. The idea is to change the topology to have at least one indirect connection (check through the originators table):

7. Power transmission change on the physical interface

7.1. Change the power transmission on the physical interface. This will allow the creation of more scenarios with more ease (less distance apart to create hops). Use **0dBm**, as seen in the following command example:

```
sudo iw wlan0 set txpower limit 0
```

7.2. Run again the batman script `create_batman_interface.sh` to reconfigure the network.

7.3. Run the following command to check if the new power transmission setting was successfully applied:

```
iw wlan0 info
```

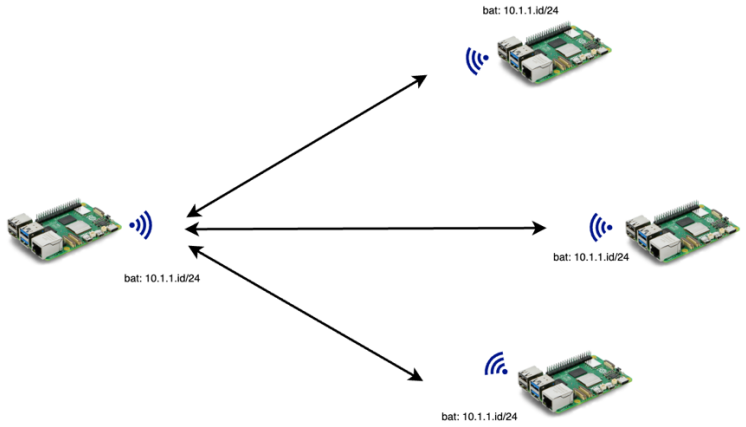
Example of the output:

```
Interface wlan0
ifindex 3
wdev 0x1
addr d8:3a:dd:e1:4a:85
ssid adhocdrones
type IBSS
wiphy 0
channel 11 (2462 MHz), width: 20 MHz, center1: 2462 MHz
txpower 0.00 dBm
```

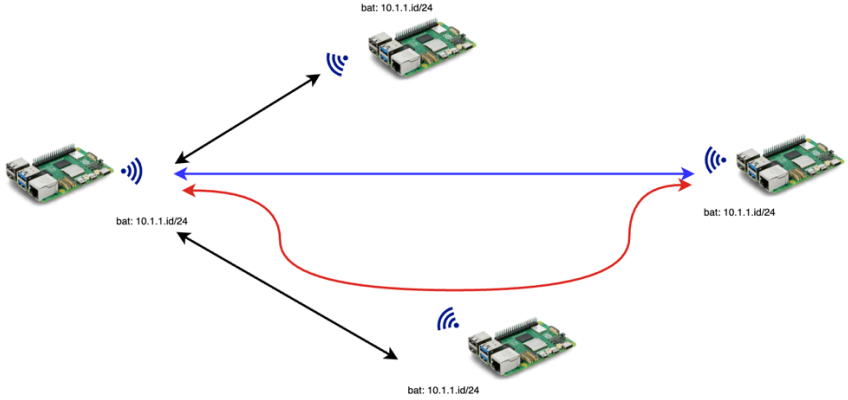
8. More scenarios

8.1. Now we will work with three topologies:

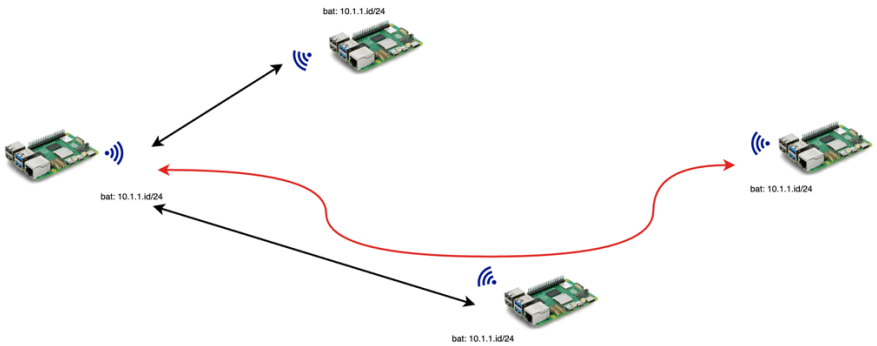
- 1 – All nodes within direct connection



- 2 – One of the nodes more far, utilizing one hop, but **still reachable with direct connection**



- 3 – One of the nodes even more far, **only reachable with one hop**.



8.2. Fill the following table, with tests for the three scenarios, and interpret the results:

Scenario	Source (MAC)	Next hop (MAC)	Destination (MAC)	**tp**	**rtt**
1			d8:3a:dd:ed:f0:fb	47,28	1.00 ms
2 (hop)	:65		fb	10306	1111
2 (direct)					
3					

tp = throughput (Mbps)
Measure with the command:

```
sudo batctl tp <dest_MAC_address>
```

rtt = round trip time (ms)
Measure with the ping command.

9. Originator Messages

9.1. Analyse the flow of originator messages among the nodes with the following command:

```
sudo batctl td wlan0
```

9.2. With some of the scenarios of the exercise **8.**, check in which conditions the board stops receiving originator messages from other boards. Correlate that behaviour with the selection of the next hop.

More info and more commands to analyse batman:
https://www.open-mesh.org/projects/batman-adv/wiki/Understand-your-batman-adv-network