Advanced Networking Lab

Thomas Hendriks Jakob Struye

March 15, 2017

Lab 1

Basic Configurations

The goal of the first lab is to make you acquainted with the hardware and software needed to perform the tasks in other modules. Furthermore we will have a look at the various sniffing possibilities in wireless networks.

1.1 Device Exploration

Have a closer look a the devices used for this course.

Exercise 1: Getting to know the interfaces

1. Log in on **wmn!** (**wmn!**)1.
   `ssh root@wmn1`

2. When asked to give a name, enter `AP`

3. Get a list of all available interfaces.
   `ap!:~# ifconfig -a`

4. List the interfaces and their **mac!** (**mac!**) addresses.
   L1-1-1

eth0	00:0D:B9:25:C5:70
lo	N/A
wlan0	D4:CA:6D:12:1B:A4
wlan1	00:0E:8E:20:83:DF

5. A **mac!** address is unique per card, but also carries a generic part, identifying the vendor of a card. List the prefixes and find out the vendors (name) of all different interfaces found in the node.

   Google is your friend!

   L1-1-2

eth0	00:0D:B9	PC Engines GmbH
lo	N/A	N/A
wlan0	D4:CA:6D	Routerboard.com
wlan1	00:0E:8E	SparkLAN Communications, Inc.

1.2 Access Point Setup

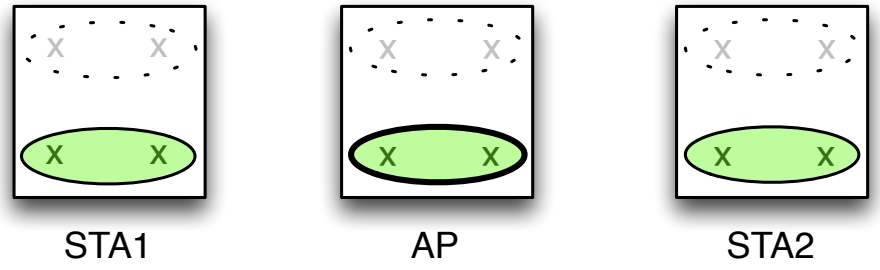

Figure 1.1: The basic infrastructure setup

The goal is to create a basic infrastructure setup as can be found at home. The setup consists of one **ap!** (**ap!**) and two connected stations as shown in figure **??**. In this case, two laptops are connected to the same **ap!** and can communicate to each other. We will not consider any communication to the outside world.

1.2.1 ap! Configuration

To set up this basic infrastructure network, start by configuring the **ap!**.

Exercise 2: Set up an **ap!**

1. Check the wireless parameters of wlan1:
   `ap!:~# iwconfig wlan1`

   This should give you some output like

```
wlan1     IEEE 802.11abgn  ESSID:off/any
          Mode:Managed  Access Point: Not-Associated   Tx-Power=0 dBm
          RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:off
```

   Remark that we did not configure the interface yet!

   Give an explanation for all available parameters which are displayed using `iwconfig` (hint: man pages).
   L1-2-1

Output of the command *iwconfig wlan1*

```
wlan1     IEEE 802.11abgn  ESSID:off/any
          Mode:Managed  Access Point: Not-Associated   Tx-Power=0 dBm
          RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:off
```

- **IEEE 802.11agbn**: - This first field shows which wireless MAC technologies the device supports on that interface.
- **ESSID:off/any** - The second field specifies the network name (if any) of a network consisting of a group of cells connected via repeaters or infrastructure where users may roam between the cells. When this field is set to off or any, ESSID checking is disabled.

- **Mode:managed**: Specifies the operating mode of the device and tells us what kind of network it operates in as well as what the function of the node is in the network.
  In this case the mode is set to Managed which means that the node connects to a network composed of many access points where roaming is enabled.

- **Access Point**: Specifies which access point the card is connected to. When configuring, it is possible to specify which access point the card should try to register to. An access point can be chosen, it can be set to ”any”, ”auto” or ”off”.
- **Tx-power=0dBm**: For network cards supporting multiple transmit powers, this setting allows you to set the transmit power in dBm. It also allows you to specify power control and radio using ”auto”, ”fixed” and ”on” or ”off” respectively.

- **RTS thr:off**: This parameter specifies when the RTS/CTS mechanism will be used and can be set to ”auto”, ”fixed” and ”off”. RTS/CTS adds overhead but increases throughput in situations with hidden nodes or a large number of active nodes.

- **Fragment thr:off**: Similar to the RTS threshold this parameter can be ”auto”, ”fixed” or ”off” and allows you to specify the maximum fragment size. Fragmentation can reduce error penalties in a noisy environment. On some cards, this threshold can be higher than the maximum packet size which will enable Frame Bursting.

- **Encryption key:off**: This parameter allows manipulation of encryption, scrambling keys and security mode. It can be used to change security keys, disable or enable encryption and change the security mode (implications of this are dependent on the type of card).

- **Power Management:off**: Used to manipulate power management scheme parameters and mode. It allows you to disable or enable power management as well as specify which types of packets should be received and choose the period between wake ups.

---

2. Configure the wlan1 interface using your assigned channel and the **essid!** (**essid!**):

```
interface=wlan
driver=nl80211
logger_syslog=-1
logger_syslog_level=2
logger_stdout=-1
logger_stdout_level=2
debug=4
hw_mode=a
channel=x
macaddr_acl=0
auth_algs=3
eapol_key_index_workaround=0
eap_server=0
wpa=0
ssid=wmn-gid-A
```

This file can be found on the devices as `hostapd.conf`. Edit it to change the channel number, interface name and **essid!**, and then perform the following:
`ap!:~# hostapd -B hostapd.conf`

`ap!:~# iwconfig` should now return something like:

```
wlan1     IEEE 802.11abgn  Mode:Master  Tx-Power=20 dBm
          RTS thr:off   Fragment thr:off
          Power Management:off
```

1.2.2 Station Configuration

Exercise 3: Configuring the wireless stations

We will now configure *both* stations and verify they get associated with the **ap!** you just created. Make sure to repeat the commands to configure the second station.

1. Bring up the interface:
   `sta!1:~# ifconfig wlan1 up`

2. Configure the interface to connect to our **ap!**:
   `sta!1:~# iw dev wlan1 connect wmn!-gid!-A`

3. Do the same on **sta!** (**sta!**)2.

 When you run `iwconfig`, the `Access Point` parameter should show the **mac!** address of the **ap!** and show identical frequencies and **essid!**s on both stations.

```
wlan1     IEEE 802.11abgn  ESSID:"wmn-0-A"
          Mode:Managed  Frequency:5.24 GHz  Access Point: 00:00:11:22:33:44
          Bit Rate=1 Mb/s   Tx-Power=5 dBm
          RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:off
          Link Quality=12/70  Signal level=-164 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:4   Missed beacon:0
```

4. What parameters are different in your output compared to the output above?
L1-3-1

- ESSID is set to wmn-1-A and not wmn-0-A
- Frequency: 5.18GHz and not 5.24GHz because we used channel 36.
- Access Point shows the MAC Address of our AP.
- Bitrate: 6Mb/s and not 1Mb/s.
- Tx-Power: shows 20 dBm and not 5 dBm.
- Link quality: 48/70 and not 12/70.
- Sign level: -62 dBm and not -164 dBm.
- Invalid misc: We had 6 packets lost that were not caused by hardware failure and not 4.

If the output for both stations confirms the association between the two wireless stations and the **ap!**, the actual L2 connection between these three devices is up and running. To verify the connection, we will configure IP addresses on the stations and have them communicate.

Exercise 4: Verifying the basic setup

1. Configure the IP address of **sta!**1 and **sta!**2:

```
sta!1:~# ip addr add fc00::gid!::1/64 dev wlan1
sta!2:~# ip addr add fc00:gid!::2/64 dev wlan1
```

2. Start a ping session from **sta!**1 to **sta!**2. Give the minimum, maximum and average **rtt!** (**rtt!**):
   `sta!1:~# ping6 -c 30 fc00:gid!::2`
   L1-4-1

   Minimum RTT: 1.091ms
   Maximum RTT: 2.712ms
   Average RTT : 1.361ms

   ---

3. Perform a ping between the same devices as before, but this time, use the wired IP addresses. Again write down the same values.
   L1-4-2

   Minimum RTT: 0.254ms
   Maximum RTT: 0.636ms
   Average RTT: 0.274ms

Remark that we did not (yet) enable L3 capabilities at the **ap!**.

1.3 Wireless Sniffing

Using the basic network we constructed in the previous section, we will go through the various sniffing modes available in wireless networks. The main difference between wired and wireless networks is obviously the fact that the transmission medium (radio waves vs. cable) is shared amongst all wireless users. Therefore, sniffing the network offers a lot more opportunities in wireless systems.

The first setup is shown in figure **??** and corresponds to the basic setup we created in the previous section.

Exercise 5: First scan

1. Make sure the **nd!** (**nd!**) cache of both **sta!**s is cleared.
   `sta!2:~# ip neigh flush dev wlan1`

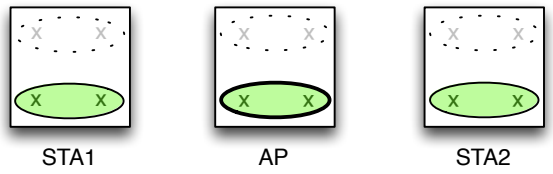

Figure 1.2: Wireless packet capture setup

```
sta!1:~# ip neigh flush dev wlan1
```

You can always check the state of the cache using `ip neigh show`

2. On the **ap!** and on both **sta!**s, start a packet capture using `tcpdump` and save it to `/mnt/L1-5-1.snif-location.pcap`

   This is the time to make sure that you have a remote location mounted with `sshfs` in /mnt!

   ```
   ap!:~# tcpdump -i wlan1 -w /mnt/L1-5-1.ap!.pcap
   sta!1:~# tcpdump -i wlan1 -w /mnt/L1-5-1.sta!1.pcap
   sta!2:~# tcpdump -i wlan1 -w /mnt/L1-5-1.sta!2.pcap
   ```

3. Start a ping session from **sta!**1 to **sta!**2. Limit the ping to only two requests.
   ```
   sta!1:~# ping6 -c 2 fc00:gid!::2
   ```

4. Which type of packets can be seen in the **ap!** trace file? You can open the trace file with wireshark on the remote computer.

L1-5-1

Looking at trace file traces/L1-5-1.AP.pcap, we observe the following message types:

- Neighbor Solicitation (e.g. packet 1): STA1 is trying to find STA2 by broadcasting these
- Neighbor Advertisement (e.g. packet 3): STA2 telling STA1 its L2 address
- Ping Request (e.g. packet 4) from STA1 to STA2
- Ping Reply (e.g. packet 5) from STA2 to STA1

---

5. Now, take a closer look at the **mac!** headers. Which type of link layer headers show up on the packets?
   L1-5-2

The type of the link layer headers is Ethernet II. These are so-called 'fake Ethernet' headers. Before passing the packets to tcpdump, the driver or network stack converts the 802.11 header to an Ethernet format.

From the tracefile made on the AP, it is impossible to see if we made a wired or wireless trace. Let's more closely examine the tracefiles made on the stations, where a difference will become apparent.

Exercise 6: Scanning other interfaces

Open the trace files made on both the sending and receiving station.

1. You should observe duplicate entries. Describe which duplicates are observed in each file (use packet numbers!):
   L1-6-1

STA1's trace file (traces/L1-5-1.STA1.pcap) shows duplicate ping requests (e.g. packets 3 and 4) and some duplicate neighbor advertisements (e.g. 10 and 11) and solicitations (e.g. 12 and 13).
According to the src MAC addresses, these are all packets sent by STA1.

Similarly, for the STA2 dump (traces/L1-5-1.STA2.pcap), everything that is sent by STA2 is duplicated.
This includes the neighbor advertisement in packets 2 and 3; the ping replies in 5 and 6; the neighbor solicitation in 10 and 11.

---

2. Take a closer look at the **mac!** addresses of the duplicate packets using Wireshark. What addresses are used on the frames? Are they identical?
   L1-6-2

Each pair of duplicates has the same source MAC addresses and destination MAC addresses.
For a ping request going from STA1 to STA2, the src and dst MAC addresses are

of STA1 and STA2, respectively. The AP's MAC address does not appear here, even though the packet was sent via the AP. This explains the duplicates: tcpdump logs both the packet sent by STA1 to AP (outgoing at STA1) and the packet sent by AP to STA2 (incoming at STA1, even though it was not the intended receiver).

`tcpdump` will automatically put the interface in *promiscuous mode*. This means all packets which can be read by the wireless interface will be delivered up the networking stack even though the destination **mac!** address does not correspond to the address of the specific card. Each interface receives all packets sent by the **ap!**. If the destination **mac!** address is not that of the receiving interface, the packet is normally dropped. In promiscuous mode, however, these packets are nevertheless stored in the trace file. Now repeat the previous exercise but disable the promiscuous mode.

Exercise 7: Disabling promiscuous mode

1. Make sure the **nd!** cache is cleared on **sta!**1 and **sta!**2.
   ```
   sta!1:~# ip neigh flush dev wlan1
   sta!2:~# ip neigh flush dev wlan1
   ```

2. On **sta!**1, start a packet capture not using promiscuous mode using `tcpdump` and save it to `/mnt/L1-7-1.sta!1.pcap`
   ```
   sta!1:~# tcpdump -i wlan1 -p -w /mnt/L1-7-1.sta!1.pcap
   ```

3. Start a ping session from **sta!**2 to **sta!**1.
   ```
   sta!2:~# ping6 -c 2 fc00:gid!::1
   ```

4. Do you still observe the duplicate entries?
   L1-7-1

No, as seen in traces/L1-7-1.STA1.pcap, the duplicate packets are now gone. The would-be duplicates are discarded before reaching tcpdump when promiscuous mode is disabled.

Exercise 8: Third party scanning

So far, we only sniffed the network on nodes which took part of the network activity. Let's introduce another party, which just wants to listen to what is happening on the

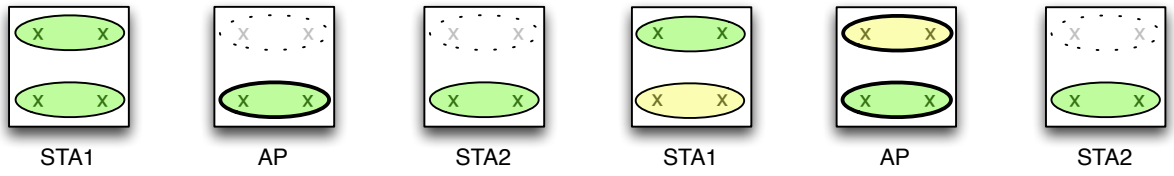

Figure 1.3: Third party scanning

channel. Imagine a notebook eavesdropping on the traffic of a wireless cell. First we will add it to the same **essid!** and later on change it to a different **essid!**. The intended setups are shown in figure **??** and **??**. The first situation then compares to a situation where a laptop is connected to your home network and is sniffing the traffic of other active stations in this network, while the latter situation can be seen as you being connected to your network, trying to sniff the network of a neighbour on the same channel, but using a different network name.

We'll start from the previous setup.

It is essential that you bring `wlan1` down on **sta!**1 first.

1. Bring `wlan1` down on **sta!**1.
   ```
   sta!1:~# ifconfig wlan1 down
   ```

2. Add `wlan0` of **sta!**1 to get the setup of **??**:
   ```
   sta!1:~# ifconfig wlan0 up
   sta!1:~# iw dev wlan0 connect essid wmn-gid!-A
   sta!1:~# ip addr add fc00:grID::1/64 dev wlan0
   ```

3. Now, reintroduce `wlan1` of **sta!**1 to the network.
   ```
   sta!1:~# ifconfig wlan1 up
   sta!1:~# iw dev wlan1 connect wmn-gid!-A
   ```

4. Configure `wlan1` with the following IP address:
   ```
   sta!1:~# ip addr add fc00:grID:1::1/64 dev wlan1
   ```
   Remark that this IP address belongs to a different subnet! This is in preparation of the next part of the exercise.

5. Repeat steps **??** to **??** from exercise **??** , but in step 2, start the capture only on `wlan1` of **sta!**1 and save it to `/mnt/L1-8-1.sta!1.A.pcap`

6. To change the `wlan1` interface to another **essid!**, we first need another **ap!** managing this **essid!** (figure **??**). Configure a second **ap!** interface on the **ap!** node, this time using "**wmn!**-**gid!** (**gid!**)-B" as essid

   Copy the previous hostapd.conf file and change the ssid and interface.

7. Make sure `wlan1` of **sta!**1 is connected to the **wmn!**-**gid!**-B network.

8. Repeat the same exercise again, saving the trace to `/mnt/L1-8-1.sta!1.B.pcap`

9. Compare the results from both tests. How do these two setups compare to a wired setup?
   L1-8-1

Setup A is similar to having each one of the involved interfaces connected to the same wired switch. All of the packets can be observed on all of the interfaces.

Setup B compares to having STA1 wlan1 and AP wlan0 on one wired switch and the other three interfaces together on another, completely separate switch. Packets sent over one of the two switches are not observed by the interfaces connected only to the other one. Setup B's trace indeed shows 0 packets.

Exercise 9: Pinging the **ap!**

In the previous exercises, we have used the **ap!** only as a L2 device. The **ap!** can of course also be configured as a L3 device. We will now configure the **ap!** as a L3 device and perform the same ping test again, but this time from **sta!**1 to **ap!**.

1. Add an IP address on `wlan0` of the **ap!**:
   ```
   ap!:~# ip addr add fc00:grID:1::3/64 dev wlan0
   ```

2. On **sta!**1, start a packet capture on `wlan1` and save it to `/mnt/L1-9-1.sta!1.pcap`
   ```
   sta!1:~# tcpdump -i wlan1 -w /mnt/L1-9-1.sta!1.pcap
   ```

3. Start a ping session from **sta!**1 to **ap!**. Limit the ping to only two requests.
   ```
   sta!1:~# ping6 -c 2 fc00:grID:1::3
   ```

4. In the trace files, do you observe duplicate entries? Why or why not?
   L1-9-1

We do not observe duplicate entries as the packets are not retransmitted by the AP: the AP is the intended receiver. There are no intermediary hosts in the transmission. As STA1 cannot receive its own packets while transmitting them, there are no duplicates in the trace.

Exercise 10: Sniffing in monitor mode

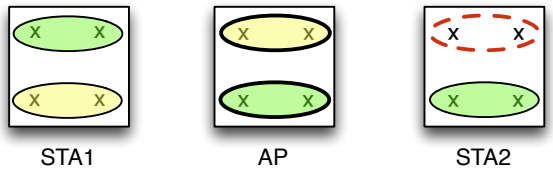

Figure 1.4: Introducing a monitoring interface.

Until now, not much difference has been observed compared to network sniffing on a wired network - apart from observing duplicate packets. In the next exercises we will broaden the setups to show the actual differences. Firstly, the monitor interface will be introduced. Configuring a **wnic!** in monitor mode will enable you to see all traffic that is present in a certain channel. In figure **??** the next setup is shown. We continue from the previous setup and add an extra interface.

1. Configure the monitoring device:
   ```
   sta!2:~# iw dev wlan0 set type monitor
   ```

2. Bring the monitor interface up:
   ```
   sta!2:~# ifconfig wlan0 up
   ```

3. Configure the channel:
   ```
   sta!2:~# iw dev wlan0 set channel x
   ```
   Remark that it is not necessary to configure an **essid!**. Why?
   L1-10-1

Because the station's wlan0 interface will be in monitor mode, it captures all packets of all ESSIDs observable on the currently selected channel. Monitor mode is passive: the interface transmits nothing and does not connect to any network: providing an ESSID would serve no purpose.

---

4. Clear the **nd!** caches:
   ```
   sta!2:~# ip neigh flush dev wlan1
   sta!1:~# ip neigh flush dev wlan0
   ```

5. Start a capture session on the monitor interface and save it to `/mnt/L1-10-2.sta!2.pcap`
   ```
   sta!2:~# tcpdump -i wlan0 -w /mnt/L1-10-2.sta!2.pcap
   ```

6. Start a ping session from **sta!**1 to **sta!**2.
   ```
   sta!1:~# ping6 -c 2 fc00:gid!::2
   ```

7. Open the capture file in Wireshark.

   - What type of frames are visible within the trace? Give an example (packet number) of each.
     L1-10-2

     There are three main types of frames observed, but each can be divided into separate subtypes. We observed the following types of frames in our trace traces/L1-10-2.STA2.pcap:

     - Management frames
       - Probe Request (e.g. 15, for SSID UAntwerpen)
       - Beacon Frame (e.g. 29, for SSID UAntwerpen)
       - Probe response (e.g. 313, for SSID wmn-1-A)
       - Action (e.g. 1172)
     - Control frames
       - Request-to-send (e.g. 1)
       - Clear-to-send (e.g. 7)
       - 802.11 Block Ack (e.g. 4)
       - Acknowledgement (e.g. 9)
     - Data frames
       - QoS Null function (e.g. 47)
       - Null function (e.g. 1178)

    - Data (e.g. 1967, these are the Neighbor soliciations/advertisements and pings)

  There were also some 802.11ac-specific frames such as the VHT NDP Announcement (e.g. 12)

- What type of link layer headers are visible now?
  L1-10-3

  Now we do see 802.11 headers, and no more Ethernet headers. In monitor mode the driver or network stack no longer converts the headers to Ethernet. Each frame contains the radio information header with some network statistics. This seems to contain roughly the same info as the Radiotap header described below, and is also a pseudo-header.
  What follows is the 802.11 header. The precise set of fields it contains depends on the (sub)type of the frame. The data frame headers seem to contain all possible fields: The frame control field containing a type + subtype byte, a duration field, source and destination MAC addresses of initial sender and final receiver, source and destination MAC address of current transmission, the AP's MAC, the STA's MAC, a fragment number, a sequence number and a frame check sequence.
  Only the data frames also have a Logical Link Control header. It identifies used L2 protocols and the type of L3 data contained.

- A radiotap header should also be visible before the link layer header. This header is not actually transmitted, but is used to communicate transmission statistics between the driver and kernel. As such, it reports on various transmission statistics as the signal strength and the used antenna. Select a packet (give the packetID within the trace), and list which items are present. Although the format is standardized, the actual content depends on the driver and hardware used.
  L1-10-4

  This is the fully expanded radiotap header of packet 1977 (ping request).

```
Radiotap Header v0, Length 26
    Header revision: 0
    Header pad: 0
    Header length: 26
    Present flags
        .... .... .... .... .... .... .... ...1 = TSFT: Present
        .... .... .... .... .... .... .... ..1. = Flags: Present
        .... .... .... .... .... .... .... .1.. = Rate: Present
```

```
9           .... .... .... .... .... .... .... 1... = Channel: Present
            .... .... .... .... .... .... ...0 .... = FHSS: Absent
11          .... .... .... .... .... .... ..1. .... = dBm Antenna
               Signal: Present
            .... .... .... .... .... .... .0.. .... = dBm Antenna Noise
               : Absent
13          .... .... .... .... .... .... 0... .... = Lock Quality:
               Absent
            .... .... .... .... .... ...0 .... .... = TX Attenuation:
               Absent
15          .... .... .... .... .... ..0. .... .... = dB TX Attenuation
               : Absent
            .... .... .... .... .... .0.. .... .... = dBm TX Power:
               Absent
17          .... .... .... .... .... 1... .... .... = Antenna: Present
            .... .... .... .... ...0 .... .... .... = dB Antenna Signal
               : Absent
19          .... .... .... .... ..0. .... .... .... = dB Antenna Noise:
                Absent
            .... .... .... .... .1.. .... .... .... = RX flags: Present
21          .... .... .... .0.. .... .... .... .... = Channel+: Absent
            .... .... .... 0... .... .... .... .... = MCS information:
               Absent
23          .... .... ...0 .... .... .... .... .... = A-MPDU Status:
               Absent
            .... .... ..0. .... .... .... .... .... = VHT information:
               Absent
25          ...0 0000 00.. .... .... .... .... .... = Reserved: 0
               x00000000
            ..0. .... .... .... .... .... .... .... = Radiotap NS next:
                False
27          .0.. .... .... .... .... .... .... .... = Vendor NS next:
               False
            0... .... .... .... .... .... .... .... = Ext: Absent
29       MAC timestamp: 66921462
         Flags: 0x10
31          .... ...0 = CFP: False
            .... ..0. = Preamble: Long
33          .... .0.. = WEP: False
            .... 0... = Fragmentation: False
35          ...1 .... = FCS at end: True
            ..0. .... = Data Pad: False
37          .0.. .... = Bad FCS: False
            0... .... = Short GI: False
39       Data Rate: 48,0 Mb/s
         Channel frequency: 5180 [A 36]
41       Channel flags: 0x0140, Orthogonal Frequency-Division
            Multiplexing (OFDM), 5 GHz spectrum
            .... .... ...0 .... = Turbo: False
43          .... .... ..0. .... = Complementary Code Keying (CCK):
               False
```

```
        .... .... .1.. .... = Orthogonal Frequency-Division
            Multiplexing (OFDM): True
45      .... .... 0... .... = 2 GHz spectrum: False
        .... ...1 .... .... = 5 GHz spectrum: True
47      .... ..0. .... .... = Passive: False
        .... .0.. .... .... = Dynamic CCK-OFDM: False
49      .... 0... .... .... = Gaussian Frequency Shift Keying (GFSK
            ): False
        ...0 .... .... .... = GSM (900MHz): False
51      ..0. .... .... .... = Static Turbo: False
        .0.. .... .... .... = Half Rate Channel (10MHz Channel
            Width): False
53      0... .... .... .... = Quarter Rate Channel (5MHz Channel
            Width): False
    SSI Signal: -50 dBm
55  Antenna: 1
    RX flags: 0x0000
57      .... .... .... .... .... ..0. = Bad PLCP: False
```

We'll only go over the actual statistics in the header and not every field.
There are some flags identifying the network's configuration in the Flags field (no WEP, fragmentation disabled, ...)
The data rate (48Mbps) and channel frequency (5180MHz, channel 36) are listed.
Then there are some channel flags identifying channel properties such as used multiplexing and modulating techniques.
Finally there is some info on signal strength and the used antenna.

- Describe the path a packet takes to be delivered from one **sta!** to another. Give a detailed overview of the various addresses in the headers. Identify the frames (packet ID) from the trace you use to illustrate this.
  L1-10-5

The ping requests clearly show how the packets are transmitted. Consider the first request in packets 1975 and 1977. The ping is from STA1 to STA2.
The first contains these addresses

```
1   Receiver address: Sparklan_20:83:df (00:0e:8e:20:83:df)
    Destination address: Sparklan_26:13:b0 (00:0e:8e:26:13:b0)
3   Transmitter address: Routerbo_12:1b:a8 (d4:ca:6d:12:1b:a8)
    Source address: Routerbo_12:1b:a8 (d4:ca:6d:12:1b:a8)
5   BSS Id: Sparklan_20:83:df (00:0e:8e:20:83:df)
    STA address: Routerbo_12:1b:a8 (d4:ca:6d:12:1b:a8)
```

The source and destination addresses are of STA1 and STA2 respectively. These show the initial source and the final destination of the frame. Conversely, the transmitter and receiver addresses are of the interface that actually transmitted this frame and the interface that should receive it. Here they are STA1 and the AP, respectively. The BSS Id is the AP's address. Here it equals the receiver address so other stations know to ignore this frame. Finally the STA address is that of the involved station, here being STA1.

These are the addresses for the second occurrence of the same request.

```
    Receiver address: Sparklan_26:13:b0 (00:0e:8e:26:13:b0)
2   Destination address: Sparklan_26:13:b0 (00:0e:8e:26:13:b0)
    Transmitter address: Sparklan_20:83:df (00:0e:8e:20:83:df)
4   Source address: Routerbo_12:1b:a8 (d4:ca:6d:12:1b:a8)
    BSS Id: Sparklan_20:83:df (00:0e:8e:20:83:df)
6   STA address: Sparklan_26:13:b0 (00:0e:8e:26:13:b0)
```

As this is still the same request logically, the source, destination and BSS Id addresses are the same as before. What did change are the transmitter and receiver addresses: these are now the AP's and STA2's respectively. The STA address is now that of STA2. This shows that the frame was first sent to the AP by STA1, and then repeated to STA2 by the AP.

Remember the traces you made containing duplicate packets? The duplicates can now be clearly observed in the traces from a monitor interface. Each transmission from one **sta!** to another, both connected to the same **ap!** is always relayed over the **ap!**. The sender thus first recorded its transmission and then detected the relayed frame on the air. Frames which are sent to an **ap!**, are discarded by other stations. This explains why a third station did not show duplicate packets. Finally, an **ap!** will only report one occurrence in `tcpdump` traces, as the relaying of a frame is done transparently to the kernel in the hardware/driver.

Exercise 11:

Finally, we will repeat exercise **??**. We will be sending traffic on both wireless networks and monitor the channel. The setup remains unchanged (see figure **??**).

1. Clear the **nd!** caches as before. Be sure to delete all **nd!** entries on all **wnic!**s (**wnic!**s)!

2. Start a scan on the monitor interface of **sta!**2 and save it to `/mnt/L1-11-1.sta!2.pcap`
   ```
   sta!2:~# tcpdump -i wlan0 -w /mnt/L1-11-1.sta!2.pcap
   ```

3. Perform a ping from **sta!**1 to **ap!** and start a ping from **sta!**2 to **sta!**1, both on `wlan1`. This means we will inject traffic on both wireless networks.

   These pings can be performed consecutively. Make sure they are in the same capture session.
   ```
   sta!1:~# ping6 -c 2 fc00:gid!:1::3
   sta!2:~# ping6 -c 2 fc00:gid!::1
   ```

4. Which ping session(s) is/are visible in the trace?
   L1-11-1

This time both ping sessions are fully visible in the tracefile (traces/L1-11-1.STA2.pcap). The monitor interface does not distinguish between ESSIDs and sees pings in both of them as they are both on the same channel as the monitor. Note that the requests and replies have duplicates only for the second ping: the first is STA to AP so not repeated.

1.4 Ad-Hoc Networks

In the next section, we will set up a basic ad-hoc network. Contrary to the infrastructure network we used in the previous section, ad-hoc networks are built from identically configured hosts, without a central entity in charge. Let's start right away to build our first ad-hoc network.

Exercise 12: Basic ad-hoc network

1. To ensure all interfaces are in the default state, reboot all devices:
   ```
   ap!:~# reboot
   sta!1:~# reboot
   sta!2:~# reboot
   ```

2. The basic setup is shown in figure **??**. All nodes will have the same configuration: mode, **essid!** and channel.

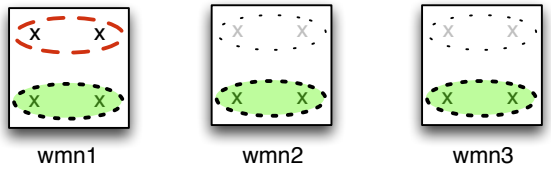

Figure 1.5: Basic ad-hoc network.

```
sta!1:~# iw dev wlan1 set type ibss
sta!2:~# iw dev wlan1 set type ibss
sta!3:~# iw dev wlan1 set type ibss
```

3. Assign IP addresses to all interfaces and activate them:
```
sta!1:~# ip addr add fc00:gid!::1/64 dev wlan1
sta!2:~# ip addr add fc00:gid!::2/64 dev wlan1
sta!3:~# ip addr add fc00:gid!::3/64 dev wlan1
sta!1:~# ifconfig wlan1 up
sta!2:~# ifconfig wlan1 up
sta!3:~# iwconfig wlan1 up
```

4. Configure all interfaces with a **essid!** and a channel:
```
sta!1:~# iw dev wlan1 ibss join wmn!-gid!-A <frequency>
sta!2:~# iw dev wlan1 ibss join wmn!-gid!-A <frequency>
sta!3:~# iw dev wlan1 ibss join wmn!-gid!-A <frequency>
```
The frequency you should use here can be found in the introduction section.

5. To monitor the traffic in the channel, set up a monitor interface on **sta!**1:
```
sta!1:~# iw dev wlan0 set type monitor
sta!1:~# ifconfig wlan0 up
sta!1:~# iw dev wlan0 set freq <frequency>
```

6. Scan using the monitor interface and save it to `/mnt/L1-12-1.sta!1.pcap`
```
sta!1:~# tcpdump -i wlan0 -w /mnt/L1-12-1.sta!1.pcap
```

7. Verify that the nodes now all can reach each other:
```
sta!1:~# ping6 -c 2 fc00:gid!::2
sta!2:~# ping6 -c 2 fc00:gid!::3
```

```
sta!3:~# ping6 -c 2 fc00:gid!::1
```

8. Describe how data is exchanged between the various hops. How does this compare to infrastructure mode? Motivate your findings by selecting frames (give the packet ID) from the trace file and describe their **mac!** header and addressing scheme.
L1-12-1

Let's again consider the first ping request (packet 3878). It contains the following addresses.

```
    Receiver address: Sparklan_26:13:c2 (00:0e:8e:26:13:c2)
2   Destination address: Sparklan_26:13:c2 (00:0e:8e:26:13:c2)
    Transmitter address: Sparklan_20:83:df (00:0e:8e:20:83:df)
4   Source address: Sparklan_20:83:df (00:0e:8e:20:83:df)
    BSS Id: 8a:89:24:d2:26:b8 (8a:89:24:d2:26:b8)
```

The source and transmitter addresses and the destination and receiver addresses are the same. This indicates the frame was sent directly from original source to final destination without any intermediary stops. There is no longer an AP. The BSS Id is still filled in but is a random address. The STA address is no longer part of the header.

Note that there are no longer any duplicates as there is no AP to repeat the frames.

---

9. Repeat the ping tests from exercise **??** (from **sta!**1 to **sta!**2 and write down the requested values. Do you observe a significant change in timings?
L1-12-2

Minimum RTT: 0.853ms
Maximum RTT: 1.536ms
Average RTT: 1.194ms

Looking at the average RTT, this is 12% faster than in managed mode. The frame no longer travels via the AP. The additional physical distance to be covered by the signal, plus the processing time at the AP explains why the RTT in exercise 4 is higher.

Note that the physical distance between STA1 and AP, and between STA1 and STA2 are roughly the same. This means the direct transmission in ad-hoc mode for this specific ping does not have to travel significantly farther than when the AP was involved.

Acronyms

AP access point

ESSID extended **ssid!**

grID group ID

MAC Media Access Control

ND Neighbor Discovery

RTT round trip time

SSID Service Set ID

STA station

wmn wireless mesh node

wnic wireless **nic!**