Advanced Networking Lab

Johan Bergs Bart Braem

February 12, 2016

Part I

WiFi Lab

Introduction

Before diving into the “wireless lab” part of this course, we would like to set some conventions first and introduce you to the devices that are used. ***Keep in mind that the conventions presented here are introduced to make all of our lives easier.*** In case of problems, it is easier to have a look at a lab setup if everyone sticks to the same rules.

0.1 Equipment

0.1.1 Introduction

Throughout these labs, you will be using three ALIX PCs, each equipped with one wired network interface card (nic) and two wireless network interface cards (wnics). The nodes run OpenWRT [10], a Linux distribution for embedded devices. A basic knowledge of the Linux command line is expected. The commands needed to configure and set up the labs will be introduced throughout the course.

0.1.2 Saving Data

All nodes are labelled with their hostname. IP address and hostname are assigned using the Dynamic Host Configuration Protocol (DHCP). The nodes boot using preboot execution environment (PXE). This means that the entire operating system is running from Random Access Memory (RAM) and that the filesystem is read-only. A RAM overlay is available, so that you can create new or modify existing configuration files on the system. After each reboot, however, the system is restored to its original state.

Keep in mind that, as the system is running entirely in RAM, there is no way to recover data from a system crash or reboot!.

In case you need to save traces, use own shell scripts, ... you can use the `sshfs` tool to mount a directory from a remote system via secure shell (SSH) on the ALIX. `sshfs` is very simple to use: `sshfs user@host:directory/ /mnt/` mounts the directory `directory` on host `host` on `/mnt/`. The remote PC will ask for the password of the supplied user.


Example: `sshfs s0123456@student.mosaic.uantwerp.be:labreports/ /mnt/`

0.1.3 Group Practice

The labs will be performed in groups of 2 students. Each group will be assigned a group ID (grID) which will be used throughout this course to identify the group. Within the next week, you need to form your groups and you will be assigned grID. On Blackboard, you can find a forum topic that should be used to announce your groups.

grID	Student 1	Student 2
1		
2		
3		
4		
5		

0.1.4 Lab Reservation

Labs can be reserved using the online lab reservation system [9]. You should be able to log in using your UA credentials. Wireless labs 1 and 2 are available for this course. They both consist of 3 wireless mesh nodes (wmns). The IP addresses of the wireless nodes are summarized below. Access to the wireless nodes is only possible using SSH. Unless you connect using your own laptop, you should as well reserve one or more PCs from the lab.

Hostname	lab ID	IPv4 address	IPv6 address
wmn1	wireless 1	143.129.81.101	2001:6a8:500:e081:20d:b9ff:fe25:c570
wmn2	wireless 1	143.129.81.102	2001:6a8:500:e081:20d:b9ff:fe18:22c0
wmn3	wireless 1	143.129.81.103	2001:6a8:500:e081:20d:b9ff:fe18:235c
wmn4	wireless 2	143.129.81.104	2001:6a8:500:e081:20d:b9ff:fe25:c560
wmn5	wireless 2	143.129.81.105	2001:6a8:500:e081:20d:b9ff:fe25:c4f4
wmn6	wireless 2	143.129.81.106	2001:6a8:500:e081:20d:b9ff:fe18:22c4

During the labs, we will use wmn1 to wmn3 to indicate the specific wireless nodes. When using wireless lab 2, of course these correspond to wmn4 to wmn6.

0.1.5 Channels

In order to prevent interference on the wireless channels, all the lab setups will be performed using 802.11a unless stated otherwise. Two channels are available. Each channel is associated with a lab setup. When using wireless 1, only use the channel assigned to this lab, and likewise for wireless 2. These channels are reserved for lab use within our group, so you will not see any other traffic and your measurements will not be interfered by tests performed for our research. For the same reason, refrain from using other channels, as you might be interfering with the work of others.

Throughout the labs, a channel identifier (x) will be used to specify which channel to use. The mapping between this identifier and the actual channel is given in table 1.

Channel ID	wireless 1	wireless 2
x	36	40

Table 1: Mapping between channel identifier and channel.

Sometimes, instead of entering a channel number, you will need to enter a frequency. Table 2 shows the mapping between channel number and frequency in MHz.

Channel	Frequency (MHz)
36	5180
40	5200

Table 2: Mapping between channel and frequency.


The lab numbers refer to the physical machines and not to the numbers of the lab exercises! Wireless 1 consists of wmn1, 2 and 3 and wireless 2 consists of wmn4, 5 and 6.

0.1.6 Accessing the Devices and the Network

To configure the devices, you will have to access them using SSH and log in with the following credentials:

username root

password mvkbj1n


mvkbj1n is derived from the Dutch sentence *"Met veel kabels bouw je één netwerk"* ("With many cables, you build one network").

The wireless devices can be accessed from the Lab PCs or from your laptop. In order to have access to the network from the master class, check the information that is available in a separate document, which is available on BlackBoard.

To keep things easy, you can change the default shell prompt on the wireless devices to mimic the names provided in the drawings and the text by exporting the prompt variable PS1. `export PS1='STA1:\w\$ '` will produce `STA1:~#` as prompt. A script is provided to ease this task. ". `./prompt.sh STA1`" will change the active prompt to `STA1:~#`. At login, this script will run so you can initialise your prompt properly and easily distinguish the various nodes.


If you want to run the script, be sure to run ". `./prompt.sh`". The first dot and space are required!

0.2 Naming and IP Address Conventions

- To create unique IP address ranges and extended Service Set IDs (ESSIDs), your group ID will be used. ESSIDs will follow the following template: wmn-grID-[A-Z].
- IP ranges can be selected from the fc00:grID::/32 range, so for example group 2 can use IP ranges fc00:2::/64, fc00:2:1::/64, fc00:2:1234:abcd::/64...
- When IP addresses are just used as an example in the course text, the documentation IP range (2001:db8::/32) will be used. ***You should never use IP addresses in this range in your own lab setups!***

0.3 Figure Conventions

The schematic view used throughout the labs represents the devices as a box. Each wnic is wired to two antennas. Both antennas are used by the wnic simultaneously. No special configuration is required to make this work. Each wnic is represented by an ellipse around the antennas used by that wnic. The wnics are numbered top to bottom;

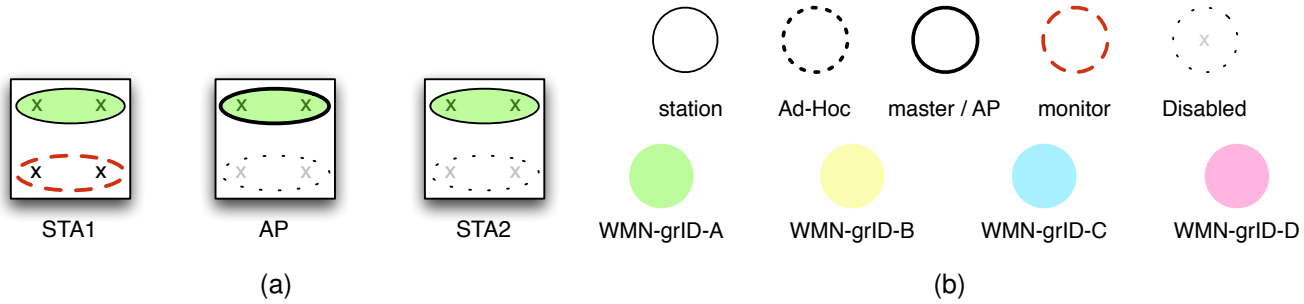

Figure 1: The basic infrastructure set up

`wlan0` is at the top and `wlan1` is at the bottom.The outline of the ellipse denotes the mode in which a wnic will operate. Make sure that when a disabled device is shown, the interface is effectively disabled to avoid interference and unwanted results. The letters in an active wnic correspond to the channel identifiers from table 1. The colors of an active wnic correspond to an ESSID as shown in the legend.

Basically, in figure 1, an access point (AP) and station (STA) is set up with one active wnic, the first one in master/AP mode and the latter in station mode. The STA1 device has two active interfaces. The first one is also configured in station mode. All mentioned interfaces will use the same channel x and use the same ESSID wmn-grID-A. The second interface of STA1 is set up as a monitor interface in channel x.

0.4 Device Introduction

In this section, we will summarize the basic commands to configure the wireless network interface cards. To start, log in to the AP device.

The mesh devices are equipped with two wnics and one wired nic. The wnics have an Atheros chipset and are controlled by the ath9k drivers [2].

Besides the wired interfaces (`eth0`), `wlanX` interfaces are available for each installed wnic.

Using the `hostapd` [7] command, APs can be configured. The general tool to configure wireless interfaces is `iw`. Sometimes, `iwconfig` is used to get information from the wireless interfaces, as it provides the information in a nice format. All three commands will be used throughout this course. Read up on their syntax in the man pages. The most important commands are summarised here:

- `hostapd -B <configfile>`

  Enables an AP. The configuration file that defines the parameters of the AP must be given as an argument. An example configuration file can be found below. This file is also present on the wireless nodes.

  
  The -B “background” flag is not strictly required. However, if you omit it, hostapd will run in the foreground until you stop it.

- `iwconfig`

  Lists the wireless parameters of all interfaces. Unsupported interfaces (e.g. wired interfaces) will return `no wireless extensions`.

- `iwconfig wlan0`

  Lists the wireless parameters of `wlan0`.

- `iw dev wlan0 set type ibss`

  Puts the `wlan0` interface in ad-hoc mode.

- `iw dev wlan0 set type monitor`

  Puts the `wlan0` interface in monitor mode.

- `iw dev wlan0 set type station`

  Puts the `wlan0` interface in station mode.

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

hostapd.conf

0.5 Communication

Do not hesitate to seek help! The objective of these labs is to get a hands on experience in wireless networking, not in debugging devices or software. If you get stuck, try to find a solution using one of the following channels:

- Send us an e-mail: johan.bergs@uantwerpen.be
- Drop by the office (G.326). Be aware I am not always at the office, so in case you want to discuss something, send me an e-mail to make an appointment.

All information about this course will be posted on Blackboard [4]. In case we need to contact you directly, we only use your UA e-mail address. Blackboard will also be used to collect your reports, so make sure you are registered there when taking this course.

0.6 Evaluation

0.6.1 Grades

This part of the "Advanced Networking Lab" course accounts for 10 points. The evaluation for this part of the course is organised as follows:

1. ***Reports***: This exercises are divided over various labs. Each lab will guide you step-by-step in building a wireless setup and perform some tests on it. During these sessions, you will need to write down some answers to questions. The questions are clearly marked in the course text. You will receive LaTeX [8] source files for each lab, with instructions on how to use them. Using these source files, you can include your answers in the reports and create a pdf file to submit on Blackboard. Answers are supposed to be short and to the point. No long essays are expected! During the labs, you will also have to make some capture files, containing test results. ***These files are an integral part of the report!*** Without the trace files, the labs cannot be evaluated. Make sure to stick to the naming scheme. Each time you need to create a capture file, the name to use is clearly stated. The generated lab reports and the corresponding capture files need to be handed in on Blackboard before the deadline. Submission other than via Blackboard (e-mail, ...), will not be accepted.
2. ***Oral examination***: at the end of the semester, an oral examination will be held to asses your personal knowledge on the course material.

For the lab reports, each group member receives the same grades. The grades for the oral examination are individual. All reports and the examination are graded on a total of 20 points. The final grade is calculated as a weighted average of each of the different reports and the examination. The weights of each part can be found in table 3.

Course part	Weight
lab 1	10%
lab 2	10%
lab 3	25%
lab 4	25%
exam	30%

Table 3: Evaluation

Lab report	Deadline
lab 1 + 2	March 18
lab 3 + 4	May 20

Table 4: Lab Report Deadlines

0.6.2 Evaluation Criteria and Deadlines

1. The lab reports are subject to deadlines. Consult table 4 for this year's deadlines. If a lab report is not handed in by 23:59 on the day of the deadline, you will not get any grades for that lab report.

2. During the examination period in June, the oral examination will take place. The date and room will be published on Blackboard. Each student will be assigned a time slot, which will also be announced on Blackboard. The examination is open book, so you can bring your lab reports with you. During the examination, you will get 15 minutes of preparation time, followed by 15 minutes of examination.

0.6.3 Re-examination

If necessary, the re-examination for this course will require you to redo labs 3 and 4 individually. An oral examination is also part of the re-examination. The deadlines for both reports are the same: August 16, at 23:59. The weights to calculate your final grade are listed in table 5.

Course part	Weight
lab 3	35%
lab 4	35%
exam	30%

Table 5: Evaluation

0.7 Writing Lab Reports

0.7.1 LaTeX Template

This course expects you to write your lab report using LaTeX. To make things easier you will be provided with a template for each lab that already contains the questions you need to fill in. The idea is that you make a .tex file for each of the questions you need to answer in the solutions folder. You are supposed to submit a compiled PDF file, the .tex solution files and your traces.

Download the lab report templates from Blackboard, which will be called lab1.tgz, lab2.tgz, etc. Unpack lab1.tgz and you will see the following structure (files you need to edit are in bold):

- labo.tex: compile this file with LaTeX to create labo.pdf
- labX.tex: contains all the questions of the lab.
- header.tex: leave this unchanged.
- ***groupid.tex***: edit this file so that it contains your names and grID.
- solution: a folder that will contain your solutions
- traces: a folder that will contain your lab traces.

Try to compile the labo.tex file and you will notice an error message like this:

```
! LaTeX Error: File ‘solutions/L1-1-1.tex’ not found.
```

This tells you that you did not yet provide an answer for question 1 of exercise 1 of lab 1. Now create a file with the exact same name, i.e. `solutions/L1-1-1.tex`. Then fill in the answer and compile again. You will now get an error message for the next

question. Once you created an answer file for every question, compilation should be successful.

To include test files you can use the lstlisting environment.

```
\begin{lstlisting}
       place your trace data here.
\end{lstlisting}
```

Which will end up looking something like this:

```
STA1:~# ping6 -c 3 wmn2
PING wmn2 (2001:6a8:20a:7:20d:b9ff:fe18:22c0): 56 data bytes
64 bytes from 2001:6a8:20a:7:20d:b9ff:fe18:22c0: seq=0 ttl=64 time=0.604 ms
64 bytes from 2001:6a8:20a:7:20d:b9ff:fe18:22c0: seq=1 ttl=64 time=0.236 ms
64 bytes from 2001:6a8:20a:7:20d:b9ff:fe18:22c0: seq=2 ttl=64 time=0.231 ms

--- wmn2 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 0.231/0.357/0.604 ms
```

To include a text file that contains this output you can use the input command, which would result in the same output as above.

```
\lstinputlisting{traces/ping.out}
```

When you encounter a LaTeXerror that you cannot work around, use the `verbatim` environment.

```
\begin{verbatim}

\end{verbatim}
```


When using the `verbatim` environment, make sure that you manually use newlines, as LaTeXwill not automatically wrap lines.

Installing LaTeX

Having issues installing LaTeX on your own laptop? Check this website: http://www.latex-project.org/ftp.html

0.8 Answering Questions

0.8.1 General Remarks

When answering a question in your lab report, keep the following in mind:

- Read the question carefully. If you are asked to compare two things, then do so. Do not describe one and forget about the other. Comparing also means that you do not describe both items separately, but that you describe the similarities and differences between them.
- Most answers can be short, but make sure you include all necessary information.
- Only answer the question, otherwise you may lose points even when you include the correct answer. If you are asked to give a Media Access Control (MAC) address, for example, only give the MAC address and don't give the IP address as well.
- If you are asked to give examples by giving packet IDs from specific packets in a trace file, then do so. Take extra care that you are referring to the correct trace file, especially if you made several traces for the same exercise.
- Give scientific answers. For example, you should write "The ping in this setup is five times as fast as in the previous setup". Do not write "There is a huge speed difference." or "In this scenario, the ping is way faster."

0.8.2 Example Q&A

This section contains an example question and answer to indicate what is expected.

Exercise 1: MAC Address Lookup

Give the MAC address of interface `eth0` of wmn1. Compare this to the MAC address of `eth0` of wmn2

- Good answer:

```
wmn1: 00:0D:B9:25:C5:70
wmn2: 00:0D:B9:18:22:C0

We observe that the first 3 bytes are the same.
```

```
The last 3 bytes are different.
This means that both devices have a eth0 card of the same manufacturer.
The last 3 bytes, which are NIC specific, are different.
```

- Wrong answer:

```
wmn1:
eth0      Link encap:Ethernet  HWaddr 00:0D:B9:25:C5:70
          inet addr:143.129.81.15  Bcast:143.129.81.255  Mask:255.255.255.0
          inet6 addr: 2001:6a8:20a:7:20d:b9ff:fe25:c570/64 Scope:Global
          inet6 addr: fe80::20d:b9ff:fe25:c570/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:10639 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6185 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1265729 (1.2 MiB)  TX bytes:1364221 (1.3 MiB)

wmn2:
eth0      Link encap:Ethernet  HWaddr 00:0D:B9:18:22:C0
          inet addr:143.129.81.16  Bcast:143.129.81.255  Mask:255.255.255.0
          inet6 addr: 2001:6a8:20a:7:20d:b9ff:fe18:22c0/64 Scope:Global
          inet6 addr: fe80::20d:b9ff:fe18:22c0/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:4617 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2448 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:601874 (587.7 KiB)  TX bytes:464740 (453.8 KiB)

We observe that the first 3 bytes are the same.
The last 3 bytes are different.
This means that both devices have a eth0 card of the same manufacturer.
The last 3 bytes, which are NIC specific, are different.
```

- Wrong answer:

```
wmn1: 00:0D:B9:25:C5:70
wmn2: 00:0D:B9:18:22:C0
```

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 wmn1.
   `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 addresses.
   L1-1-1

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

1.2 Access Point Setup

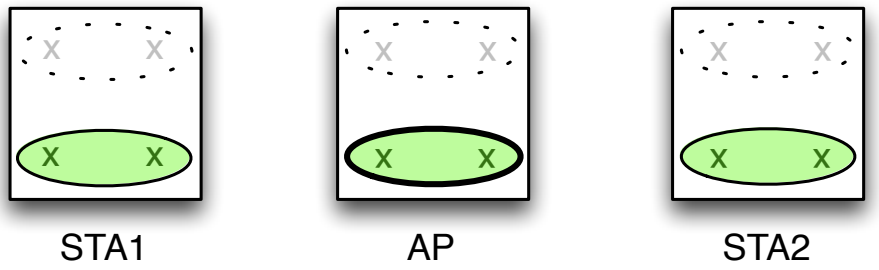

Figure 1.1: The basic infrastructure setup

The goal is to create a basic infrastructure setup as can be found at home. The setup consists of one AP and two connected stations as shown in figure 1.1. 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

.....

2. Configure the wlan1 interface using your assigned channel and the 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:
   ```
   STA1:~# ifconfig wlan1 up
   ```

2. Configure the interface to connect to our AP:
   ```
   STA1:~# iw dev wlan1 connect wmn-grID-A
   ```

3. Do the same on STA2.

   When you run `iwconfig`, the `Access Point` parameter should show the MAC address of the AP and show identical frequencies and ESSIDs 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

   .....

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

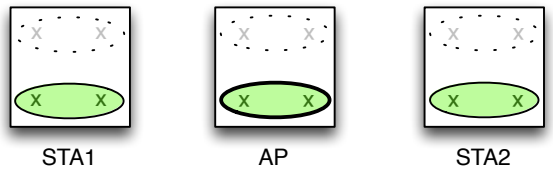

Figure 1.2: Wireless packet capture setup

1. Configure the IP address of STA1 and STA2:
   ```
   STA1:~# ip addr add fc00::grID::1/64 dev wlan1
   STA2:~# ip addr add fc00:grID::2/64 dev wlan1
   ```

2. Start a ping session from STA1 to STA2. Give the minimum, maximum and average round trip time (RTT):
   ```
   STA1:~# ping6 -c 30 fc00:grID::2
   ```
   L1-4-1

   .....

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

   .....

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 1.2 and corresponds to the basic setup we created in the previous section.

Exercise 5: First scan

1. Make sure the Neighbor Discovery (ND) cache of both STAs is cleared.
   ```
   STA2:~# ip neigh flush dev wlan1
   STA1:~# 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 STAs, 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
   STA1:~# tcpdump -i wlan1 -w /mnt/L1-5-1.STA1.pcap
   STA2:~# tcpdump -i wlan1 -w /mnt/L1-5-1.STA2.pcap
   ```

3. Start a ping session from STA1 to STA2. Limit the ping to only two requests.
   ```
   STA1:~# ping6 -c 2 fc00:grID::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
   .....

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

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
   .....

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
   .....

`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 STA1 and STA2.
   ```
   STA1:~# ip neigh flush dev wlan1
   STA2:~# ip neigh flush dev wlan1
   ```

2. On STA1, start a packet capture not using promiscuous mode using `tcpdump` and save it to `/mnt/L1-7-1.STA1.pcap`
   ```
   STA1:~# tcpdump -i wlan1 -p -w /mnt/L1-7-1.STA1.pcap
   ```

3. Start a ping session from STA2 to STA1.
   ```
   STA2:~# ping6 -c 2 fc00:grID::1
   ```

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

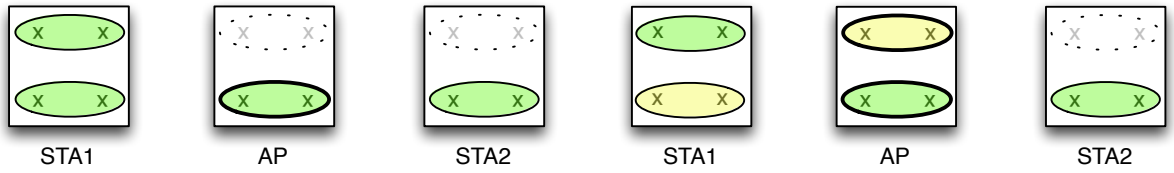

Figure 1.3: Third party scanning

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 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 1.3a and 1.3b. 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 STA1 first.

1. Bring `wlan1` down on STA1.
   ```
   STA1:~# ifconfig wlan1 down
   ```

2. Add `wlan0` of STA1 to get the setup of 1.3a:
   ```
   STA1:~# ifconfig wlan0 up
   STA1:~# iw dev wlan0 connect essid wmn-grID-A
   STA1:~# ip addr add fc00:grID::1/64 dev wlan0
   ```

3. Now, reintroduce `wlan1` of STA1 to the network.
   ```
   STA1:~# ifconfig wlan1 up
   STA1:~# iw dev wlan1 connect wmn-grID-A
   ```

4. Configure `wlan1` with the following IP address:
   `STA1:~# 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 1 to 3 from exercise 5 , but in step 2, start the capture only on `wlan1` of STA1 and save it to `/mnt/L1-8-1.STA1.A.pcap`

6. To change the `wlan1` interface to another ESSID, we first need another AP managing this ESSID (figure 1.3b). Configure a second AP interface on the AP node, this time using "wmn-grID-B" as essid

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

7. Make sure `wlan1` of STA1 is connected to the wmn-grID-B network.

8. Repeat the same exercise again, saving the trace to `/mnt/L1-8-1.STA1.B.pcap`

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

   L1-8-1

   .....

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 STA1 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 STA1, start a packet capture on `wlan1` and save it to `/mnt/L1-9-1.STA1.pcap`
   `STA1:~# tcpdump -i wlan1 -w /mnt/L1-9-1.STA1.pcap`

3. Start a ping session from STA1 to AP. Limit the ping to only two requests.
   `STA1:~# ping6 -c 2 fc00:grID:1::3`

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

   .....

Exercise 10: Sniffing in monitor mode

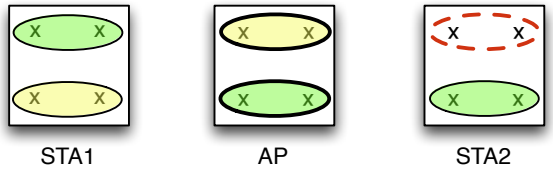

Figure 1.4: Introducing a monitoring interface.

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 wireless network interface card in monitor mode will enable you to see all traffic that is present in a certain channel. In figure 1.4 the next setup is shown. We continue from the previous setup and add an extra interface.

1. Configure the monitoring device:
   `STA2:~# iw dev wlan0 set type monitor`

2. Bring the monitor interface up:
   `STA2:~# ifconfig wlan0 up`

3. Configure the channel:
   `STA2:~# iw dev wlan0 set channel x`

   Remark that it is not necessary to configure an ESSID. Why?
   L1-10-1

   .....

4. Clear the ND caches:
   `STA2:~# ip neigh flush dev wlan1`

```
STA1:~# ip neigh flush dev wlan0
```

5. Start a capture session on the monitor interface and save it to `/mnt/L1-10-2.STA2.pcap`
   ```
   STA2:~# tcpdump -i wlan0 -w /mnt/L1-10-2.STA2.pcap
   ```

6. Start a ping session from STA1 to STA2.
   ```
   STA1:~# ping6 -c 2 fc00:grID::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

     .....

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

     .....

   - 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

     .....

   - Describe the path a packet takes to be delivered from one station 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

     .....

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 station 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 access point 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 8. We will be sending traffic on both wireless networks and monitor the channel. The setup remains unchanged (see figure 1.4).

1. Clear the ND caches as before. Be sure to delete all ND entries on all wnics!
2. Start a scan on the monitor interface of STA2 and save it to `/mnt/L1-11-1.STA2.pcap`
   ```
   STA2:~# tcpdump -i wlan0 -w /mnt/L1-11-1.STA2.pcap
   ```
3. Perform a ping from STA1 to AP and start a ping from STA2 to STA1, 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.
   ```
   STA1:~# ping6 -c 2 fc00:grID:1::3
   STA2:~# ping6 -c 2 fc00:grID::1
   ```
4. Which ping session(s) is/are visible in the trace?

   L1-11-1

   .....

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

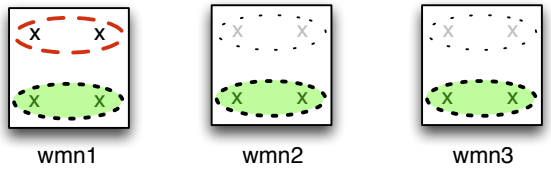

Figure 1.5: Basic ad-hoc network.

1. To ensure all interfaces are in the default state, reboot all devices:
```
AP:~# reboot
STA1:~# reboot
STA2:~# reboot
```

2. The basic setup is shown in figure 1.5. All nodes will have the same configuration: mode, ESSID and channel.
```
STA1:~# iw dev wlan1 set type ibss
STA2:~# iw dev wlan1 set type ibss
STA3:~# iw dev wlan1 set type ibss
```

3. Assign IP addresses to all interfaces and activate them:
```
STA1:~# ip addr add fc00:grID::1/64 dev wlan1
STA2:~# ip addr add fc00:grID::2/64 dev wlan1
STA3:~# ip addr add fc00:grID::3/64 dev wlan1
STA1:~# ifconfig wlan1 up
STA2:~# ifconfig wlan1 up
STA3:~# iwconfig wlan1 up
```

4. Configure all interfaces with a ESSID and a channel:
```
STA1:~# iw dev wlan1 ibss join wmn-grID-A <frequency>
STA2:~# iw dev wlan1 ibss join wmn-grID-A <frequency>
STA3:~# iw dev wlan1 ibss join wmn-grID-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 STA1:
```
STA1:~# iw dev wlan0 set type monitor
STA1:~# ifconfig wlan0 up
STA1:~# iw dev wlan0 set freq <frequency>
```

6. Scan using the monitor interface and save it to `/mnt/L1-12-1.STA1.pcap`
   ```
   STA1:~# tcpdump -i wlan0 -w /mnt/L1-12-1.STA1.pcap
   ```

7. Verify that the nodes now all can reach each other:
   ```
   STA1:~# ping6 -c 2 fc00:grID::2
   STA2:~# ping6 -c 2 fc00:grID::3
   STA3:~# ping6 -c 2 fc00:grID::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

   .....

9. Repeat the ping tests from exercise 4 (from STA1 to STA2 and write down the requested values. Do you observe a significant change in timings?
   L1-12-2

   .....

Lab 2

Frequencies and Channels

In the IEEE 802.11 standards, various channels are defined. Two main frequency bands are used, namely 2.4 GHz and 5GHz. IEEE 802.11a networks use the 5 GHz band, while IEEE 802.11b/g networks operate in the 2.4 GHz band. In both bands, various "channels" have been defined. However, channel separation is not as clean cut as one might expect. In the following setups, you will illustrate this using some simple tests.

2.1 Available channels

Exercise 1: Frequencies

Within the IEEE 802.11 specification, various channels are defined. Depending on the local authorities (Belgisch Instituut voor Postdiensten en Telecommunicatie - Belgian Institute for Postal services and Telecommunications (BIPT) in Belgium [3]), the list of allowed channels can vary. Using `iw` you can get the list of available channels.

The wireless drivers we use make a difference between the logical interface (`wlan0` as we have used it so far) and the actual physical radio interface. All things related to physical characteristics are actually handled by the physical interface, which is called `phy0` or `phy1`, respectively.

1. To get a correct overview of which frequencies/channels are available in Belgium, use the command `iw phy1 info`.
2. List all frequencies/channels that the `phy1` interface supports.
   L2-1-1

You should observe several frequencies that are marked *disabled*. This indicates that the card supports these frequencies, but cannot use them because of Belgian regulations. Also, several frequencies should be marked with *radar detection*. These can be used, but special mechanisms must be put in place in order to avoid interference with radar installatiuons (e.g. airport radar).

Exercise 2: Available UA hotspots

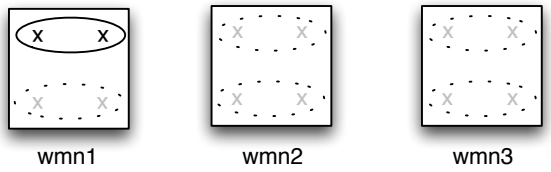

Figure 2.1: Channel separation setup 1

1. Start by building the setup as illustrated in figure 2.1.

   Remark that it is not necessary to configure the SSID or channel, but bring up the interface to activate it.

2. Find out which APs are active using `iw wlan0 scan`. This command provides an interface to list various settings from the wnic. The manpage or command line help should be self-explanatory.

3. Fill in the following table:
   L2-2-1

SSID	Frequency (MHz)	BSS

2.2 Channel Separation

Exercise 3: Channel overhearing

In the previous exercise, you made an overview of a lot of access points. The next exercise will illustrate the channel overhearing. The information about which channel an AP is working on, is provided in the beacon frames the AP periodically transmits. In figure 2.2, a Wireshark [11] screen shot shows a beacon frame. The Radiotap header indicates the frame was received on channel 5, while the beacon's content shows the AP sending this beacon only operates at channel 6. Channels are thus not cleanly separated. In the next exercise you'll create an overview of the channel overhearing.

1. Start from the setup as shown in figure 2.3.

   Note that the AP should be configured in channel 4 from the b/g range. To be able to do this, you should change the line `hw_mode=a` to `hw_mode=b` in the hostapd.conf file.

2. On STA1, on every channel, make a capture `/mnt/L2-3-1.chanID.pcap`

3. Use the following wireshark display filter to only show beacons:
   `wlan.fc.type_subtype == 8`

4. Fill out the following table using the obtained information.
   L2-3-1

Selected channel	Observed channels in received beacons
1	
2	

3	
4	
5	
6	
7	
8	
9	
10	
11	

From this exercise, it should be clear that the channels within the IEEE 802.11b/g range are not strictly separated. Figure 2.4[1] gives you an idea why: the consecutive channels overlap to a certain extent. Therefore, a careful channel planning is crucial in building and deploying wireless networks on these frequencies. In the next exercise, we will take a look at the channel separation in the IEEE 802.11a band.

Exercise 4: Channel separation in IEEE 802.11a

1. Now, change the AP so that it is on channel `x`.
2. Using `tcpdump` as in the previous exercise, find out in which channels IEEE 802.11a the beacons of this AP can be seen. Save your traces in `/mnt/L2-4-1.chanID.pcap`

L2-4-1

2.3 Using the Wireless Channel

Exercise 5: Beacons

In the next few exercises, we will again use an ad-hoc network setup. It is best to reboot the devices before proceeding.

1. Create the network setup as shown in figure 2.5. Perform the following commands on all three stations. Substitute `nodeNumber` with a different number (e.g. 1, 2 and

[1] By Michael Gauthier, Wireless Networking in the Developing World [CC-BY-SA-3.0 (http://creativecommons.org/licenses/by-sa/3.0)], via Wikimedia Commons

No.	Time	Source	Destination	Protocol	Info
1	0.000000	ArubaNet_a6:26:80	Broadcast	IEEE 80.	Beacon frame,SN=2648,FN=0,BI=100, SSID: "UA-vpn"
2	0.001109	ArubaNet_a6:26:81	Broadcast	IEEE 80.	Beacon frame,SN=2649,FN=0,BI=100, SSID: "UA-visit"
3	0.001684	ArubaNet_a6:26:82	Broadcast	IEEE 80.	Beacon frame,SN=2650,FN=0,BI=100, SSID: "UA-dwep"
4	0.002089	ArubaNet_a6:26:83	Broadcast	IEEE 80.	Beacon frame,SN=2651,FN=0,BI=100, SSID: "UA-tkip"
5	0.002511	ArubaNet_a6:26:84	Broadcast	IEEE 80.	Beacon frame,SN=2652,FN=0,BI=100, SSID: "UA-aes"

```
▷ Frame 1 (126 bytes on wire, 126 bytes captured)
▽ Radiotap Header v0, Length 26
    Header revision: 0
    Header pad: 0
    Header length: 26
  ▷ Present flags: 0x0000186f
    MAC timestamp: 7916884150
  ▷ Flags: 0x12
    Data Rate: 1.0 Mb/s
    Channel: 5
    Channel frequency: 2432
    Channel type: 802.11g (0x0480)
    SSI Signal: -73 dBm
    SSI Noise: -96 dBm
    Antenna: 1
    SSI Signal: 23 dB
▷ IEEE 802.11
▽ IEEE 802.11 wireless LAN management frame
  ▷ Fixed parameters (12 bytes)
  ▽ Tagged parameters (60 bytes)
    ▷ SSID parameter set: "UA-vpn"
    ▷ Supported Rates: 1.0(B) 2.0(B) 5.5 11.0
    ▷ DS Parameter set: Current Channel: 6
    ▷ (TIM) Traffic Indication Map: DTIM 0 of 1 bitmap empty
    ▷ Country Information: Country Code: BE, Any Environment
    ▷ Power Constraint: Tag 32 Len 1
    ▷ ERP Information: no Non-ERP STAs, do not use protection, short or long preambles
    ▷ Extended Supported Rates: 6.0 9.0 12.0 18.0 24.0 36.0 48.0 54.0
    ▷ Reserved tag number: Tag 171 Len 11
```

Figure 2.2: Channels in a captured beacon frame.

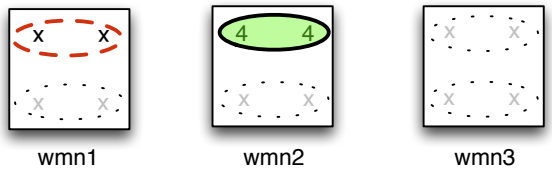

Figure 2.3: Setup for exercise 3.

3) on each node.

```
wmn:~# iw dev wlan0 set type ibss
wmn:~# ip addr add fc00:grID::nodeNumber/64 dev wlan0
wmn:~# ifconfig wlan0 up
wmn:~# iw dev wlan0 ibss join wmn-grID-A <frequency>
```

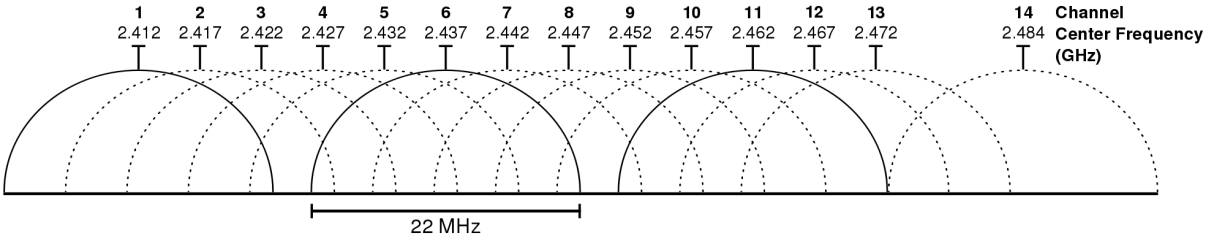

Figure 2.4: 2.4 GHz channels

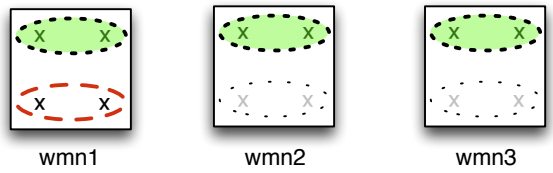

Figure 2.5: Basic ad-hoc network.

2. Put interface wlan1 of STA1 in monitor mode on the same frequency.
   ```
   STA1:~# iw dev wlan1 set type monitor
   STA1:~# ifconfig wlan1 up
   STA1:~# iw dev wlan1 set freq <frequency>
   ```

3. Check if all stations can reach each other.

4. Perform a `tcpdump` on the monitor interface. Save it to `/mnt/L2-5-1.STA1.pcap` Let it run for a few seconds and then stop the trace.

5. In your trace file, you should observe beacon frames. Carefully inspect one of the beacon frames in this trace and compare it to a beacon frame captured in the previous exercise (exercise 4). Give the packet IDs of the beacon frames you are comparing and identify the differences in the management frame part.
   L2-5-1
   .....

6. Start a new trace on the monitor interface and save it to `/mnt/L2-5-2.STA1.pcap`

7. While the trace is running, perform two ping tests. You may perform the simultaneously, but it will be easier to answer the next question if you perform them consecutively.
   ```
   STA2:~# ping6 -c 5 fc00:grID::3
   STA3:~# ping6 -s 1400 -c 5 fc00:grID::1
   ```

8. Open your trace file and filter out the beacon frames with the following display filter: `!(wlan.fc.type_subtype == 0x08)`. Apart from the data frames, which kind of frames do you observe?
   L2-5-2
   .....

9. Explain the purpose of those frames.
   L2-5-3
   .....

Exercise 6: Request to Send (RTS)/Clear to Send (CTS)

For this exercise, you will study the RTS/CTS mechanism. This mechanism will clear the channel for each transmission, minimizing the chance of collisions in the channel. A threshold value is used to determine if RTS/CTS should be used. Larger packets will receive RTS/CTS protections while smaller packet will not. Using RTS/CTS for small packets adds a lot of overhead and hence degrades performance.

1. Start from the setup as for the previous exercise.
2. The RTS/CTS threshold will be set with `iwconfig wlan0 rts 1000`. Perform this command on all nodes.
3. Performing the same `ping6` commands as in the previous exercise. Start a capture session on the monitor interface and save it to `/mnt/L2-6-1.STA1.pcap`
4. You should observe RTS/CTS packets. Which packets are protected by this mechanism? Give an example (packet ID).
   L2-6-1
   .....

5. Compare an RTS and CTS frame. How do they differ? Illustrate using frames from your last trace file. L2-6-2

.....

As you can see in the RTS/CTS frames, they do not contain any information about the network on which they are transmitted. The RTS/CTS is meant to avoid collisions on the wireless medium, so any node that receives a CTS frame is required to remain silent for the duration included in the CTS frame. The only exception, for obvious reasons, is the node to which the CTS frame is sent. Capturing RTS/CTS frames (or acknowledgement frames) on a certain channel is always a good indication that some activity is going on in that channel, even if it is impossible to overhear the actual data transmission.

2.4 IEEE 802.11n

Until now, we have only used the wnics in IEEE 802.11a or b/g mode. “b” is the oldest mode. “g” improves upon “b” by offering significantly higher maximal throughput (54 Mbps versus 11 Mbps) .“a” operates at different frequencies and offers the same throughput as “g”.

Another mode of operation is called “n”. IEEE 802.11n is an amendment to the IEEE 802.11 standard in order to improve throughput over both “a” and “g”. It operates at both frequency bands and is defined for bit rates up to 600 Mbps, but very few setups will be able to obtain these speeds.

IEEE 802.11n can use different optimizations, such as the multiple-input, multiple-output (MIMO) principle, 40 MHz wide channels (versus 20 MHz in IEEE 802.11a/b/g), and frame aggregation to improve throughput. The following exercise will demonstrate that IEEE 802.11n channels can indeed be twice as wide as those in IEEE 802.11a, and thus interfere with adjacent IEEE 802.11a channels.

Exercise 7: IEEE 802.11n

As this exercise will use both channels allocated to the mobile and wireless lab, two groups cannot perform this exercise at the same time! Check if no other group is currently working on this course!

1. Reboot all devices. In order to enable IEEE 802.11n mode without any problems, we will start from a blank setup.

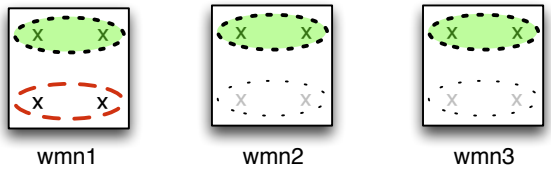

Figure 2.6: IEEE 802.11n ad-hoc network.

2. After they have been rebooted, perform the following on all devices in order to create an ad-hoc network with IEEE 802.11n support in the 5 GHz band:
```
STA:~# iw dev wlan0 set type ibss
STA:~# ip addr add fc00:grID::nodeNumber/64 dev wlan0
STA:~# ifconfig wlan0 up
STA:~# iw dev wlan0 ibss join wmn-grID-A 5180 HT20
```

3. Check that all nodes can reach each other.

4. Set up a monitor interface on channel 36 (5180 MHz) on STA1.

5. Start a trace on the monitor interface and save it to `/mnt/L2-7-1.STA1.pcap`

6. Start a ping from STA2 to STA3 and from STA3 to STA1. Like before, use different ping sizes:
```
STA2:~# ping6 -c 5 fc00:grID::3
STA3:~# ping6 -s 1400 -c 5 fc00:grID::1
```

7. Filter out the beacon frames from your trace. Do you observe packets from both ping sessions? Did you see all packets from these sessions? Why or why not?
L2-7-1

.....

8. Set the monitor interface to channel 40 (5200 MHz).

9. Perform the same ping test again, but now save your trace to `/mnt/L2-7-2.STA1.pcap`

10. Did you capture any packets?
L2-7-2

.....

Exercise 8: 40 MHz channels

In the previous exercise, we enabled IEEE 802.11n, but we did not enable 40 MHz wide channels yet. We will do that now.

1. Deconfigure all `wlan0` interfaces.
   ```
   STA:~# iw dev wlan0 ibss leave
   ```

2. Now, create an ad-hoc network that uses 40 MHz wide channels:
   ```
   STA:~# iw dev wlan0 ibss join wmn-0-A 5180 HT40+
   ```

3. As in the previous exercise, set the monitor interface of STA1 on channel 36 and perform a capture while sending both pings. Save it to `/mnt/L2-8-1.STA1.36.pcap`

4. Do the same again, but this time monitor channel 40.
   Save your trace in `/mnt/L2-8-1.STA1.40.pcap`

5. Do you observe any different behaviour in the trace file on channel 36, compared to the previous exercise?
   L2-8-1
   .....

6. In the trace file on channel 40, which packets have you captured? Why?
   L2-8-2
   .....

Lab 3

Performance Measurements

In this lab, the performance and throughput in wireless networks will be investigated. Using tools like `iperf`, we will record the maximum throughput which can be achieved in wireless networks and have a look at the parameters influencing this throughput.

3.1 Bit Rates

Exercise 1: Basic throughput in IEEE 802.11a

This first exercise will give you an insight into the difference in usable throughput and the available bit rate. To determine the throughput, we will be using a tool called `iperf`. This is a client-server based tool which sends Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) traffic and reports the measured throughput. Consult the man pages for the details about this tool. A second tool to be used is `gnuplot`, in order to plot your results in a graph. More info about `gnuplot` can be found in [5] and a nice tutorial in [6]. A basic gnuplot script to generate your first plots is provided on the course website. The `iperf` tool is preinstalled on the wireless nodes, while `gnuplot` is available on the lab PCs.

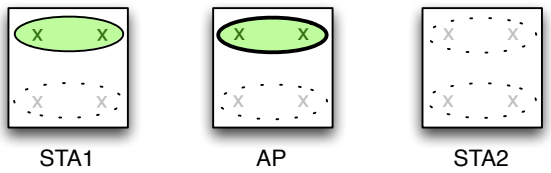

Figure 3.1: Basic throughput setup.

1. Start by configuring the setup as shown in figure 3.1. Use `fc00:grID::3/64` for the AP and `fc00:grID::1/64` for STA1.

2. IEEE 802.11 supports various bit rates. Using `iw list` you can query the supported rates:
   Which rates are supported in IEEE 802.11a?
   L3-1-1

   .....

   Which rates are supported in IEEE 802.11b/g?
   L3-1-2

   .....

3. `iperf` can be used to measure the maximum throughput between two stations. Therefore, a server is started on one end and a client connects on the other end to this server. A connection is set up and as TCP tries to maximize the throughput on a connection, an estimate of the maximum throughput on a link can be calculated. When using `iperf` with the basic parameters, it will perform a 10 seconds test and report the achieved throughput at the client side.

4. Start a basic `iperf` session between the STA and AP, with the `iperf` server on the AP:
   ```
   AP:~# iperf -V -s
   STA1:~# iperf -V -c fc00:grID::3
   ```
   Copy the output of the `iperf` client:
   L3-1-3

   .....

5. Now, using `iperf`, collect the throughput for each available rate and write the results in a file `/mnt/L3-1-4.tcp.txt` On each line, first put the rate followed by a space and then the result in Mbit/s obtained from `iperf`, e.g. `54 30` denotes that a 30Mbit/s was measured when using a rate of 54 Mbps. You can change the rate used at the STA using `iw`, e.g. to 54Mbps, as follows:
   ```
   STA1:~# iw dev wlan0 set bitrates legacy-5 54
   ```
   You can check the actual used bit rate from the output of `iwconfig`:
   ```
   STA1:~# iwconfig wlan0
   ```
   As we will be repeating the collection of these results in the following exercises, it will be easier to use some bash scripting to speed up this process. A helper script can be found in the file `iperf-tcp.sh`. Change this script so it loops over

the correct bitrates, and use it with the command `./iperf-tcp.sh fc00:grID::3`. This script generates output that can be directly copied to `/mnt/L3-1-4.tcp.txt`

6. `iperf` will by default use TCP to check the connection, but it is also possible to use a unidirectional UDP stream. Therefore, one can try to feed more data to the network than the network can support and as such measure how much can be actually delivered. Thus, repeat the previous scenario and collect the maximum achievable throughput using `iperf` in UDP mode for each available bit rate. Save these results in the same format as in the previous item in a file `/mnt/L3-1-4.udp.txt` Again, a script called `udp.sh` is provided that automates this process. The script will try to send a UDP stream that fills the wireless link.

   ```
   AP:~# iperf -V -s -u -l 1452
   STA1:~# ./iperf-udp fc00:grID::3
   ```

   
   `-l` is the letter l, not the number one!

7. Now plot these results using `gnuplot`. On the course website, a `gnuplot` script, `tput1.gnuplot`, is provided to plot the throughput and the relative link usage over the various available rates. The used commands are straightforward and the script is inline commented so should be self-explanatory. Make sure you understand the various commands in the file. The script produces two PDF files which can be viewed with any regular PDF viewer. Place the `.txt` files you created in the previous steps in the same directory as the `gnuplot` script. Save the generated files in your lab report as `L3-1-4-tput.pdf` and `L3-1-4-usage.pdf`. Add the plots here and shortly discuss what can be observed. Also shortly discuss the difference between UDP and TCP. To generate the PDF files, use:

   ```
   gnuplot tput.gnuplot
   ```

   L3-1-4

.....

Exercise 2: Client to client throughput

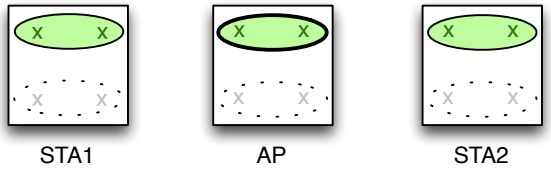

Figure 3.2: Client to client throughput setup.

In the previous exercise, the throughput was measured between a STA and the AP it was connected to. In this exercise, we are interested in the throughput between two STAs associated with the same AP.

1. Create the test setup as shown in figure 3.2. Configure STA2 with IP address `fc00:grID::2/64`.
2. Collect the TCP and UDP throughput measures as in the previous exercise and save them to `/mnt/L3-2-1.tcp.txt` and `/mnt/L3-2-1.udp.txt`
3. Adapt the provided `gnuplot` script and create graphs for the new measurements. Include them in your report.
4. Comment on the obtained results and compare the results from this exercise with those from the previous exercise. Explain the difference.
   L3-2-1

.....

3.2 Network Settings

Exercise 3: RTS/CTS

In lab 2, we enabled the RTS/CTS mechanism to show that in IEEE 802.11, stations can reserve the medium in order to avoid collisions. In this exercise, we will study the effect of RTS/CTS on throughput.

1. Enable RTS/CTS on all devices.
   ```
   :~# iw phy phy0 set rts 1000
   ```
2. Repeat the iperf tests from STA1 to AP. Save your logs to `/mnt/L3-3-1.tcp.txt` and `/mnt/L3-3-1.udp.txt` Again modify the gnuplot script to create new graphs and include them in your report. To make the comparison more easy, modify the gnuplot script so that you plot both the results from exercise 1 and this exercise on the same graph. What do you observe? Why?
   L3-3-1

.....

Exercise 4: Frame length

The standard Maximum Transferable Unit (MTU) for ethernet frames is 1500 bytes. IEEE 802.11 however allows an MTU up to 2274 bytes. In this exercise, you will measure the effect of frame size on throughput.

1. Continue from the previous setup, but make sure RTS/CTS is turned off on all nodes:
   `iw phy phy0 set rts off`

2. On STA1, reset rate control to its default (automatic) value:
   `STA1:~# iw dev wlan0 set bitrates`

3. Perform the following tests with 4 different maximum frame sizes: 1500, 1758, 2016 and 2274. The frame size can be controlled by changing the MTU. *If you do this, do this on all nodes!*
   `ifconfig wlan0 mtu 1758`

4. For each of the MTUs mentioned, start with a TCP `iperf` test between STA and AP as in the previous exercises and make plots for each frame size. Generate a results file called `/mnt/L3-4-1.tcp.txt` containing MTU and bit rate achieved. The file should have the same format as the ones produced this far, except that the first column now contains your MTU setting rather than the link speed.
   `STA1:~# iperf -V -c fc00:grID::3`

5. Repeat the tests, now using UDP. For `iperf`, use the `-l` parameter, followed by the current MTU setting minus 48 (e.g. 1452 if the MTU is set to 1500) to generate packets that are large enough to fill the MTU. ***This must be done on both client and server!*** Save your results to `/mnt/L3-4-1.udp.txt`.
   `STA1:~# iperf -V -u -l 1452 -c fc00:grID::3 -b 54M`

6. Modify the `gnuplot` script to generate the same graph as before. Plotting only the throughput versus MTU suffices. A relative link usage graph is not required. What do you observe? Why? Be as precise as possible.
   L3-4-1

.....

3.3 Packet Size

Exercise 5: IP Fragmenting

Changing frame lengths has an effect on the throughput, as you have shown in the previous exercise. However, in that case, traffic was flowing between segments with the same fragment size. When we consider the default setup of an AP which is connected to the Internet, the wide area network (WAN) link is most likely limited to 1500 bytes per frame and thus IP fragmenting comes into play. In IPv6, fragmenting in intermediate hops is not allowed. The end hosts may, however, fragment the IP packets end-to-end to make sure they fit on the link with the smallest MTU. Depending on the original payload length, fragments of various lengths will be created. In this exercise we will have a look at the impact of fragmenting on the overall throughput.

To do so, we will run the `iperf` server on the third node, which is only connected by wire to the AP. Using additional routes, we will create a setup where STA1 and STA2 communicate via the AP. Traffic between STA1 and the AP will be transmitted on the wireless link, while traffic between the AP and STA2 will be transmitted on the wired link.

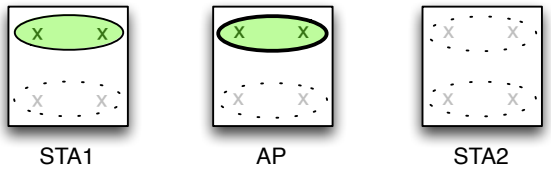

Figure 3.3: Fragmenting setup.

1. Start by configuring your setup as shown in figure 3.3. STA2 will only be used as `iperf` server and does not need an active wnic.

2. Set `fc00:grID::3/64` as IP for the AP and `fc00:grID::1/64` for STA1.

3. Traffic from STA1 to STA2 will by default use the wired network. To force a wireless hop, we will add some new routes:
   ```
   STA2:~# ip route add fc00:grID::/64 via <wired IP address of AP>
   STA1:~# ip route add <wired IP address of STA2>/128 via fc00:grID::3
   ```

4. IPv6 forwarding is by default not enabled on the lab nodes. We must enable it on the AP for the setup to work. However, due to a bug, the AP will clear its default route when we do this, making it instantly inaccessible over IPv6. Perform the following as a workaround:

```
AP:~# ip -6 route show | grep default
```
You should get output like this:
```
default via fe80::225:90ff:fe61:efac dev eth0 proto kernel metric 1024
expires 1670sec
```

5. Enable IPv6 routing functionality on the AP:
```
AP:~# sysctl -w net.ipv6.conf.all.forwarding=1 && ip route add default
via fe80::225:90ff:fe61:efac dev eth0
```

   The IP address in that command should be the same as that in the output of the previous step.

6. Perform a `traceroute6` from STA1 to STA2 to check if the routes are set up correctly. Include your traceroute6 output below:
```
STA1:~# traceroute6 <wired IP address of STA2>
```

L3-5-1

.....

7. Now, for each MTU (1500, 1758, 2016 and 2274), perform the following:

   - Set the MTU on the wireless link.
   - Perform a TCP `iperf` from STA1 to STA2.
   - Perform a UDP `iperf` from STA1 to STA2. In each trace, set your frame size to the MTU of the wireless link minus 48. Send at a bit rate of 54 Mbps.

8. The results of your tests should once again be saved to text files like in the previous exercise. Save them to `/mnt/L3-5-2.tcp.txt` and `/mnt/L3-5-2.udp.txt`. Create a throughput graph as in the previous exercise.

9. You should also make `.pcap` trace files from these experiments. However, as the link gets fully saturated, the trace file would become very large. Therefore, we will repeat the experiments sending far less traffic. This of course means that you will not see a difference in `iperf` results. The traces will help you, however, to understand what has happened in the previous exercise.

10. In order to limit the amount of traffic, we will do the following on STA1:

    - Set the rate of `wlan0` to 6 Mbps.
    - for both TCP and UDP `iperf`, add the `-t 2` parameter. This limits the `iperf` trace to two seconds.

- for UDP `iperf`, omit the `-b 54M` parameter. This way, `iperf` will only transmit at 1 Mbps.

11. Now, do the same `iperf` tests again, using the above parameters for `iperf`. For each run, make a `tcpdump` trace on interface `wlan0` of the AP. Save the traces to `/mnt/L3-5-2.tcp.<MTU size>.pcap` and `/mnt/L3-5-2.udp.<MTU size>.pcap`, respectively.

12. What do you observe? Distinguish between the behaviour for TCP and UDP. Explain your findings by including your graphs and refering to the trace files you made. Be as precise as possible.

L3-5-2

. . . . . .

Lab 4

Wireless Security

In this lab, you will have a look at the impact of various security measures which are available for wireless networks. The main goal is to show what the effect is of a certain security method and what a possible attacker/sniffer can do on the protected network. In the first lab, you already performed some sniffing using the monitor mode. It should be clear that unless we take some security measures, it is quite easy to intercept and read any management information and data from open networks. It is even quite easy to perform denial of service (DoS) attacks on open networks as you will see at the end of this lab.

In the first part of the lab, you will investigate the more basic security features like hiding a network and MAC filtering. In the second part, you will have a look at encryption mechanisms and try to hack some of these. In the last part, we will introduce some basic attacks. These attacks are introduced to illustrate the mechanisms behind wireless networks and should only be used for this purpose.

4.1 Basic Security Measures

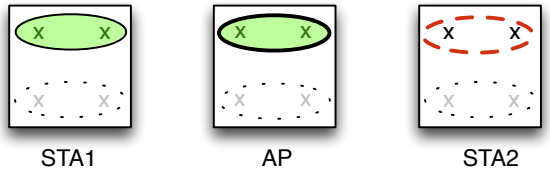

Figure 4.1: MAC filter setup

Exercise 1: Network hiding

A first step in preventing someone to use your wireless network is to hide it from being detected by a regular scan. Hiding a Service Set ID (SSID) ensures it will not show up in the list of available APs when a scan is performed. You will however show that the clever sniffer is still able to detect the network.

1. Set up your network as illustrated in figure 4.1, but do not set up the wnic of STA1 yet.
2. Start a trace on the monitor interface (`/mnt/L4-1-1.STA2.pcap` of STA2. While the trace is running, bring up the wnic from STA1. Make sure the station is associated with the AP before you stop your trace.
3. Indicate in which frames the SSID is visible. Give a packet ID of an example packet in your trace file for each.
   L4-1-1
   .....
4. The next step is to hide the SSID. This can be done adding the following line to the hostapd.conf file: `ignore_broadcast_ssid=1`. Bring the interface down on STA1, restart `hostapd` on the AP and repeat steps 1 and 2. Save your trace to `/mnt/L4-1-2.STA2.pcap`
5. Compare the two trace files and indicate how the SSID is hidden. In what way can a SSID still be detected?
   L4-1-2
   .....

Exercise 2: MAC filtering

Continuing from the setup you used in the previous exercise, a possible intruder has still complete access to our network. In the next step, you will prevent him from joining the network by MAC filtering.

1. The management of the MAC filtering is also performed by modifying the `hostapd.conf` file at the AP. The entries needed for MAC filtering are:

   macaddr_acl= controls the behaviour of the mac filtering and expects an integer value from the following list:

0 accept unless in deny list,

1 deny unless in accept list,

2 use external RADIUS server (accept/deny lists are searched first

accept_mac_file= append the name of the file here that contains the list of MAC addresses to accept.

deny_mac_file= append the name of the file here that contains the list of MAC addresses to deny.

2. Configure the AP in such a way that the MAC address of the `wlan0` interface of STA1 is blocked from the network. SSID broadcasting may be enabled again.

3. Now bring down the wnic at STA1 again, restart a capture session on STA2 and save it to `/mnt/L4-2-1.STA2.pcap` and bring the wnic back up at STA1 when this capture session is active. Compare the results from this capture file with the first one you made and discuss the differences between both. Illustrate with packet IDs.
L4-2-1

4. Of course, in a real world setup, blocking unwanted MAC addresses is not feasible. One would use the option with the `accept_mac_file` parameter, allowing only known MAC addresses to connect. How would you circumvent this security measure? If an attacker has joined the network, will this have an effect on legitimate stations on that network?
L4-2-2

4.2 Encryption

4.2.1 WEP

The security techniques from the previous section will "prevent" a possible attacker from joining the network by hiding the network or preventing him access to the network. However, it is still possible to read any communication over the network by using a wnic in monitor mode. Encryption mechanisms will counter this and obfuscate the network traffic for an eavesdropper. You will take a look at Wired Equivalent Privacy (WEP) here

as this method is still sometimes being used although it is very vulnerable to attacks. Wi-Fi Protected Access (WPA)2 is far more resilient.

Exercise 3: WEP encryption

WEP was the first available encryption method, standardised together with the IEEE 802.11 standard. It was however quickly been proven insecure. In fact, as of 2004, WEP was declared deprecated by IEEE because of its flaws. In this exercise, you will first set up a WEP encrypted network to check what is exactly encrypted. In the next exercise you will crack the encryption key.

Because WEP is quite old, the tools you will use assume that IPv4 is used, as they will use Address Resolution Protocol (ARP) messages. Therefore, for this exercise, you will configure IPv4 addresses instead of IPv6 addresses.

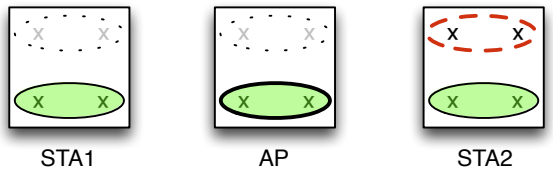

Figure 4.2: WEP setup

1. Start from the network setup as shown in figure 4.2. Disable any MAC filter that is still in place.

2. In order to avoid capturing needless IPv4 traffic, disable the `avahi-daemon` on all three nodes:
   `wmn:~# killall avahi-daemon`

3. Create a tracefile on the monitor interface of STA2 (`/mnt/L4-3-1.STA2.cap`), containing the scan, authentication and association phase of STA1 and a 5 packet ping session between STA1 and STA2. Use `10.0.grID.1/24` for STA1 and `10.0.grID.2/24` for STA2.
   `STA:~# ip addr add 10.0.grID.X/24 broadcast 10.0.grID.255 dev wlan1`

4. Make sure that STA2 is connected to the AP and that the wnic of STA1 is down before the start of the capture session. To ping, use `ping` instead of `ping6`. This trace will be our reference trace.

5. Activate the WEP encryption on the AP. You can choose a random key as long as it is either 5 or 13 characters long (”arandomwepkey” in this example):

Change the `hostapd.conf` file, so that it includes the following lines:
```
wep_key0="arandomwepkey"
wep_default_key=0
```

The WEP key must be ***exactly 5 or 13*** characters long!

6. Make sure the AP is running. Connect STA2 to the AP:
```
STA2:~# iw dev wlan1 connect wmn-0-A key 0:arandomwepkey
```

7. Make sure the interface of STA1 is down and then start a new trace on the monitor interface of STA2. Save it to `/mnt/L4-3-1.STA2.wep.pcap`.

8. Perform the same test as in step 4, but be sure to add `key 0:arandomwepkey` when you connect STA1 to the network.

9. Compare both trace files. In the second trace file, you should no longer be able to see the `ping` session in the clear.

   - Which frame types are encrypted?
     L4-3-1

     .....

   - Which part of a frame is encrypted and which part is still readable?
     L4-3-2

     .....

   Refer to specific frames from the second trace file to illustrate your answers.

Exercise 4: Cracking the WEP keys

Without going into details about the flaws in WEP, the main problem arises from the fact that traffic keys are easily repeated on a busy network. A traffic key is the key actually used by the encryption algorithm RC4. As RC4 is a stream cipher, the same key should not be used twice. Therefore, the shared WEP key, which is configured in each AP and STA, is combined with a random initialization vector (IV). This IV is transmitted in plaintext between two wireless clients so the receiver can recreate the encryption key used for a specific packet, by combining the chosen (shared) network key with the received IV. As the IV has a length of only 3 bytes, the number of possibilities is relatively low and in busy networks, the same IVs are frequently reused, weakening the

cryptographic strength of WEP. Furthermore, statistical relations between used keys and encrypted text can be exploited to find the used key.

Various methods have been developed to crack a WEP key, but you will just illustrate the mere ease with which a WEP encrypted network can be hacked using the aircrack tool [1]. This tool implements various approaches, but you will limit ourselves to Pychkine, Tews, Weinmann (PTW) method. The attack is based on the fact that the first 16 bytes of an ARP packet are fixed. This is also the reason we switched to IPv4 for this exercise. . . For more details read [14], specifically sections 1 and 5 for a general overview. Exploiting the relation between cleartext and the captured IVs makes it possible to quite easily determine the network key.

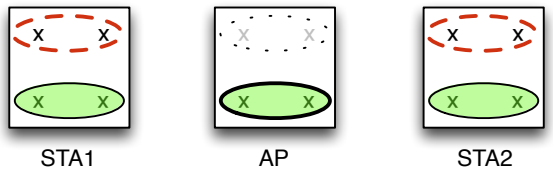

Figure 4.3: WEP cracking setup

1. Before you start to crack the configured WEP key, first have a second look at the traces from the previous exercise. Using Wireshark [11], you can decrypt an encrypted trace. Using the menu Edit→ Preferences → Protocols → IEEE 802.11 you can specify the key in HEX. To convert your key to HEX, use e.g. the following website: `http://www.dirtymonday.net/key_convert.html` For our example it will be `6172616E646F6D7765706B6579`.
   Make sure you also check the 'decrypt packets' box. Select an ICMP request in both trace files (unencrypted and encrypted, but decrypted by Wireshark) and identify the extra information added to a frame in order to encrypt it. Also indicate where in the frame this information is added.
   L4-4-1

   .....

2. Start from the WEP encrypted network you created in the previous exercise.

3. Add an additional monitor interface on STA1, resulting in the network scheme shown in figure 4.3. The monitor interface on STA2 will be used to perform the actual attack. ARP packets will be injected into the network in order to trigger ARP responses which will contain new IVs with every response . These packets will be captured and stored in files. In parallel with this capturing of new ARP responses

and the associated IVs, we can start the process of analyzing these packages and searching for the key. The monitor interface on STA1 will be used to monitor the channel so we can afterwards take a look at what happened.

4. Start a packet capture to `/mnt/L4-4-2.STA1.pcap` on the monitor interface of STA1.

5. Start the IV capture process at STA2. This command will store its captured frames in files `output-xy.cap` and `output-xy.txt`. Make sure to run this command with your active directory set to the `/mnt` folder, in which a remote directory is mounted.
   `STA2:~# airodump-ng --band a --channel x --bssid <MAC addr AP> -w output wlan0`
   This should give you an overview screen where at least the incoming beacon count is rising. If no activity is shown, bring the interface down and then bring it up again and retry. This command will store its captured frames in files `output-xy.cap` and `output-xy.txt`. Make sure to run this command in `/mnt`, where a remote location is mounted.

6. On a new terminal, perform a fake authentication to the AP from the attacking machine (STA2). This ensures the frames we will be injecting for `wlan1` will be accepted by the AP:
   `STA2:~# aireplay-ng --fakeauth 0 -e wmn-grID-A -a <MAC addr AP> -h <MAC addr wlan0> wlan0.`

7. It is now time to actively inject packets into the wireless network. This is done by the following command: `STA2:~# aireplay-ng --arpreplay -b <MAC addr AP> -h <MAC addr wlan0> -x 1024 -o 512 wlan0`

   This command will monitor incoming packets and when an ARP request passes, it will re-inject it at a rate of several 100 frames/s. You should be able to see this in the output from `airodump-ng`.

   Due to a bug in the driver, the packet injection rate will not be high enough. You should run 10 of the above `aireplay-ng` commands in parallel to generate enough packets. You can do so easily by editing the `wepcrack.sh` script found on the nodes. Insert the correct MAC addresses in that script and run it. To stop the replay attack, simply type `killall aireplay-ng`.

8. In order to get an ARP request, just start a ping from STA1 to AP. This should trigger an ARP request. If the ARP tables already contained the IP addresses from both hosts (check it with `STA1:~# ip neigh`), remove the entry using `ip neigh flush dev wlan1`.

9. To crack the key, the tool `aircrack-ng` can be used. This tool processes the files generated by `airodump-ng`. The needed number of packets to crack the key depends on the bit length of the key. In general, you need at least 20,000 packets for a 64-bit key and between 40,000 and 85,000 for a 128-bit key. To start the analysis, run the following command in the folder containing the output from `airodump-ng`:
   `STA2:~# aircrack-ng -b <MAC addr AP> output*.cap -q`

10. If you captured enough packets, after a while, the `aircrack-ng` tool should provide you with the WEP key used in your network. Copy its output here:
    L4-4-2
    .....

11. Using the following `tshark` command on a lab PC, you can get a list of IVs used during your session.
    `tshark -r <trace file> -T fields -e wlan.wep.iv`

12. How many IVs did you capture? And how many of them occured at least twice?
    L4-4-3
    .....

13. When handing in the report, you must include the `output-*.csv` and `*.netxml` files. `output-*.cap` is not required, as it will be quite large. Limit the trace file of STA1 to only a few (e.g. a hundred) of packets that show the ARP injection in progress.

4.2.2 WPA2

As you have demonstrated in the previous exercise, WEP encryption is easily circumvented. You will now take a look at WPA2 encryption. You will no try to crack it, as WPA2 uses the Counter Cipher Mode with Block Chaining Message Authentication Code Protocol (CCMP) algorithm with Advanced Encryption Standard (AES) encryption, which is extremely resilient to attacks. To give you an idea, the best known method until now to crack a 128 bit AES key, which is used by WPA2, still takes at least $2^{126.1}$ computations [13].

A security flaw concerning Wi-Fi Protected Setup (WPS) has been discovered that makes APs running WPS vulnerable to a relatively easy attack (which would still take

far too much time for this lab exercise) [15]. This, of course, does not hold true for APs that do not have WPS enabled. If you want to read up on how WPA2 works and on the evolution in security from WEP over WPA to WPA2, you should read [12].

Exercise 5: WPA2

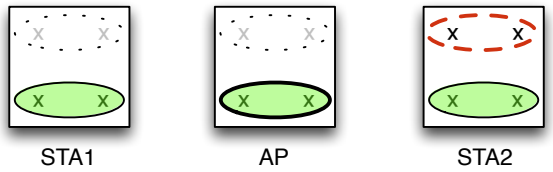

Figure 4.4: WPA setup

1. Start from the WPA2 enabled network setup as seen in figure 4.4.
2. On the AP, to enable WPA2, make sure that the line `wpa=0` in the `hostapd.conf` file is changed to `wpa=2`, and add the following lines to `hostapd.conf`:

   ```
   wpa_passphrase=aRandomPassphrase
   wpa_key_mgmt=WPA-PSK
   rsn_pairwise=CCMP
   ```

   Use your own passphrase if you like. Unlike WEP, it is not restricted to a certain amount of characters.
3. Set the IP addresses of `wlan1` on STA1 and STA2 to `fc00:grID::1/64` and `fc00:grID::2/64`, respectively.
4. Connect STA2 to the network. In order to do so, modify the `wpa_supplicant.conf` file so that it includes the correct ESSID and passhprase. Afterwards, run:
   `STA2:~# wpa_supplicant -B -c/root/wpa_supplicant.conf -iwlan1`

   You might get an error like `ioctl[SIOCSIWENCODEEXT]: Invalid argument`. You can safely ignore this error.
5. Make sure the monitor interface on STA2 is configured and start a trace. Save it in `/mnt/L4-5-1.STA2.pcap`.
6. Now bring the interface on STA1 up an do `ping6 -c 5` to STA2.

7. Which frame types are encrypted? Give examples.
   L4-5-1
   .....

8. Which part of a frame is encrypted and which part is still readable?
   L4-5-2
   .....

9. Compare this trace file with the one made in exercise 3. Compare the authentication/association phase of STA1. What are the similarities and what are the differences? Refer to packet IDs!
   L4-5-3
   .....

10. Now, compare an encrypted frame in both traces. Indicate where they differ.
    L4-5-4
    .....

4.3 Denial of Service Attacks

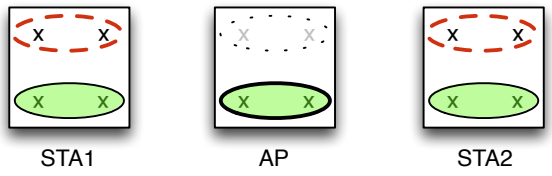

Figure 4.5: DoS setup

In the previous section, you have gained access to a WEP encrypted network by actively injecting packets and cracking the key. This kind of attack makes it possible that you, the attacker, can afterwards associate with the network and sniff all traffic on that

network. Alternatively, you can also use the network yourself as if you were a normal user.

Within this section, you will illustrate a DoS attack that shows the vulnerability of a wireless network. You will not gain access to the network as such, but you will disturb normal network operation. Tools to do so are freely available and current encryption techniques cannot prevent these attacks as they make use of management frame injection. As you have seen, management frames are not encrypted and thus can be easily injected into a wireless network.

Exercise 6: Disassociation Attack

The principle behind a disassociation attack is quite easy. A disassociation frame is sent by either a STA or AP to signal the disconnection from the wireless network. For instance, it is sent by a STA just before you disable an associated wnic. Although the sending of a disassociation message is not mandatory by the standard, at reception of such frame, a client should consider its connection lost and an AP will remove a STA from the list of connected stations. Injecting a malicious disassociation frame for a wireless node will thus force this node to scan for and associate again to its AP, resulting in a connectionless period. If the disassociation frame is sent periodically, this can render the wireless network useless to the attacked STA.

1. Start from the network scheme as illustrated in figure 4.5.
2. Be sure to have configured your network with WPA2 encryption.
3. Configure the IP addresses of STA1 and STA2 to `fc00:grID::1/64` and `fc00:grID::2/64` respectively.
4. On the monitor interface of STA1, start a capture session and save it to `/mnt/L4-6-1.STA1.pcap`.
5. Start a `ping6` from STA1 to STA2.
6. On STA2, use the following command to send a disassociation to STA1:
   `STA2:~# aireplay-ng --deauth 1 wlan0 -a <MAC addr AP> -c <MAC addr STA1>`
7. What is the result for the ping?
   L4-6-1

   .....

8. Open the trace file and explain using this trace what exactly happened.
   L4-6-2

Acronyms

AES Advanced Encryption Standard

AP access point

ARP Address Resolution Protocol

BIPT Belgisch Instituut voor Postdiensten en Telecommunicatie - Belgian Institute for Postal services and Telecommunications

CCMP Counter Cipher Mode with Block Chaining Message Authentication Code Protocol

CTS Clear to Send

DHCP Dynamic Host Configuration Protocol

DoS denial of service

ESSID extended Service Set ID

grID group ID

IV initialization vector

MIMO multiple-input, multiple-output

MAC Media Access Control

MTU Maximum Transferable Unit

ND Neighbor Discovery

nic network interface card

PTW Pychkine, Tews, Weinmann

PXE preboot execution environment

RAM Random Access Memory

RTS Request to Send

RTT round trip time

SSH secure shell

SSID Service Set ID

STA station

TCP Transmission Control Protocol

UDP User Datagram Protocol

WAN wide area network

WEP Wired Equivalent Privacy

wmn wireless mesh node

wnic wireless network interface card

WPA Wi-Fi Protected Access

WPS Wi-Fi Protected Setup

Bibliography

[1] Aircrack-ng [online]. URL: http://www.aircrack-ng.org/ [cited August 24th, 2010].

[2] ath9k - Linux Wireless [online]. URL: http://linuxwireless.org/en/users/Drivers/ath9k [cited October 3rd, 2012].

[3] BIPT - Welkom [online]. URL: http://www.bipt.be/ [cited August 2rd, 2010].

[4] Blackboard Learn [online]. URL: https://blackboard.ua.ac.be/ [cited August 20th, 2010].

[5] gnuplot homepage [online]. URL: http://www.gnuplot.info/ [cited September 15th, 2009].

[6] gnuplot tips (not so Frequently Asked Questions) [online]. URL: http://t16web.lanl.gov/Kawano/gnuplot/index-e.html [cited September 15th, 2009].

[7] hostapd: IEEE 802.11 AP, IEEE 802.1X/WPA/WPA2/EAP/RADIUS Authenticator [online]. URL: http://hostap.epitest.fi/hostapd/ [cited September 27th, 2011].

[8] LaTeX - A document preparation system [online]. URL: http://www.latex-project.org/ [cited September 27th, 2011].

[9] Meeting Room Booking System [online]. URL: https://patsstudent.cmi.ua.ac.be/telecomlabo/ [cited September 15th, 2009].

[10] Openwrt [online]. URL: http://openwrt.org/ [cited October 3rd, 2012].

[11] Wireshark · Go deep. [online]. URL: http://www.wireshark.org/ [cited August 20th, 2010].

[12] WiFi Alliance®. The State of WiFi® Security, 2012. http://www.wi-fi.org/sites/default/files/uploads/files/wp_State_of_Wi-Fi_Security_20120125.pdf.

[13] Andrey Bogdanov, Dmitry Khovratovich, and Christian Rechberger. Biclique Cryptanalysis of the Full AES. Cryptology ePrint Archive, Report 2011/449, 2011. http://eprint.iacr.org/2011/449.pdf.

[14] Erik Tews, Ralf-Philipp Weinmann, and Andrei Pyshkin. Breaking 104 bit WEP in less than 60 seconds. Cryptology ePrint Archive, Report 2007/120, 2007. http://eprint.iacr.org/2007/120.pdf.

[15] Stefan Viehböck. Brute forcing Wi-Fi Protected Setup, 2011. http://sviehb.files.wordpress.com/2011/12/viehboeck_wps.pdf.