Exercise 2:
IEEE 802.11 Hands-On Exercise

TECHNISCHE UNIVERSITÄT DARMSTADT

Mobile Networking
Secure Mobile Networking Lab - SEEMOO
Matthias Hollick, Allyson Sim, Dingwen Yuan and Robin Klose
30th November, 2018

Goal

This hands-on exercise will deepen your understanding of the IEEE 802.11 MAC by means of practical experiments. You will assess the behavior of the IEEE 802.11 MAC under various conditions and different protocol settings. In particular, you will examine the implications of equal channel access probability in IEEE 802.11 wireless networks.

Organization

- **Registration**: The exercise will be conducted in small groups. Please register in Moodle for a time slot. The exercise takes place in building S4|14 at several dates. At each date, we can serve 4 groups of up to 4 students.

  If you don't have a team yet, please fill up non-empty slots first, so that other teams can register more easily!

- **Preparation**: Prepare yourself for the exercise! We have a preparation class on 30th November 2018, right after the lecture. We will discuss the concepts of the 802.11 MAC that will be assessed practically in the hands-on exercise. In particular, we have a closer look at the fair chance of medium access in the 802.11 MAC and discuss its implications. Read the exercise sheet carefully to get an overview of the tasks. This saves your time during the exercise session.

- **Storage**: We recommend to bring an USB flash drive in order to save recorded Wireshark traces so that you can review and process your results later when writing your report.

- **Hardware**: During the exercise, each group will get four netbooks, one notebook and one access point (AP) to set up a wireless network. You may also bring your own notebooks in addition to the provided devices.

- **Experiments**: Plan your experiments! Think carefully about the meaning and purpose of each task before running experiments: clarify the goal of your experiment, think about reasonable parameter settings that might lead to meaningful results, then conduct the experiment. Some tasks might also require you to take several series of measurements with parameter sweeps. Do not forget to note down your setup, your parameter settings and your measurement results carefully!

- **Report**: Document your experiments thoroughly in a report. For each task, write down the purpose of your experiment in your own words, reason about how you set up your experiment and which parameter settings you choose. Then describe your results and give an explanation. If the results do not match your expected outcome of the experiment, describe the differences between what you expected and what you observed and try to give reasons. Your report should also include tables or plots. It is permissible to share the Wireshark traces and results among members of the same exercise group, still each student should write his or her own report!

  Upload your report to Moodle as a PDF named `201812DD_groupGG_lastname_firstname.pdf`, where `DD` is the day of your time slot and `GG` is your group number according to your Moodle registration (e.g., 1A).

- **Deadline**: Upload your report to Moodle until 20:00 on Wednesday, 19th December 2018.

- **Discussion**: We will discuss the hands-on exercise and your results on Friday, 21th December 2018.

Overview of Commands

You will use several GNU/Linux system tools to set up the network and to conduct experiments. Make yourself familiar with the following commands (in advance before the practical exercise). Below, you find a brief description for each command and a selection of its usages. Please refer to the manpage of a command (run `man <command>` in a terminal) for more detailed and comprehensive documentation.

ifconfig

The `ifconfig` command allows to configure wired and wireless network interfaces:

- `ifconfig`: Display the names of active network interfaces and their respective configurations.
- `ifconfig -a`: Display the names of all network interfaces and their respective configurations.
- `ifconfig <interface>`: Query a detailed status for a specific network interface, e.g., `eth0`.
- `ifconfig <interface> up`: Activate the specified interface.
- `ifconfig <interface> down`: Shut down the driver of the specified interface.

iwconfig

The `iwconfig` command allows to configure specifically *wireless* network interfaces:

- `iwconfig`: Display the names of wireless network interfaces and their respective configurations.
- `iwconfig <interface>`: Display the configuration of a specific wireless network interface, e.g., `wlan0`.
- `iwconfig <interface> mode managed`: Set the wireless network interface to managed mode.
- `iwconfig <interface> mode monitor`: Set the wireless network interface to monitor mode.
- `iwconfig <interface> essid <name>`: Set the *ESSID* (Extended Service Set Identifier), i.e., the network name.
- `iwconfig <interface> channel <number/frequency>`: Set the channel to a channel number or a frequency.
- `iwconfig <interface> rate <rate>`: Set the data rate.
- `iwconfig <interface> rts <threshold>`: Set the RTS threshold in bytes. You may also set this parameter to `auto`, `fixed` or `off`.
- `iwconfig <interface> frag <threshold>`: Set the MAC layer's fragmentation threshold.

iwlist

The `iwlist` command allows to scan for APs and to query the capabilities of a wireless network interface:

- `iwlist <interface> scan`: Scan for APs.
- `iwlist <interface> channel`: Obtain a list of channels/frequencies supported by the network interface.
- `iwlist <interface> rate`: List data rates supported by the network interface.

iperf

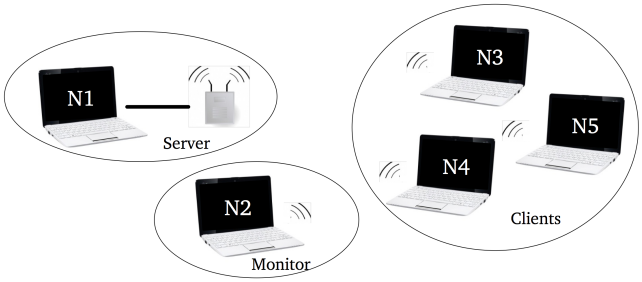
`iperf` is a tool for running network performance tests. It can test both UDP and TCP performance. In order to run a performance test with `iperf`, you have to set up a server that acts as a data sink as well as one or several clients that generate traffic to the server. The server will finally provide measurements of the network performance. Examples:

- `iperf --udp -s`: Start an `iperf` server listening for incoming UDP traffic.
- `iperf -s`: Start an `iperf` server listening for incoming TCP traffic.
- `iperf --udp -c <ip> -l <length> -b <bandwidth>`: Start `iperf` in client mode and generate UDP traffic to the specified IP address with frames of length `<length>` and an offered load `<bandwidth>`.

Press `Ctrl+C` to stop a server or a client. Please refer to the manpage of `iperf` for more options and details.

Test Network

Your test network is composed of a server, several clients and a monitor. The server runs on a netbook (N1) which is connected to an access point (AP) via Ethernet and which serves as the data sink for the clients. The clients (N3, N4, N5) generate traffic that is sent to the server. Depending on the experiment, they might have different parameter settings for their generated traffic. The monitor (N2) is passive and does not interact with the network at all. It just records received WLAN frames with Wireshark. It is most convenient to use the notebook with the large screen as the monitor.



Network Setup - Server

The server runs on a netbook (N1) connected to the AP via Ethernet. The wireless network interface of the netbook itself will not be used. To set up the netbook, perform the following steps:

- Log in with username `seemoo` and password `pw4mobnet`.
- Open a terminal. You can press `Ctrl+Alt+T` to open a terminal and `Alt+F7` to maximize it.
- Execute `sudo -i` to obtain administrator rights in a root shell.
- Execute `service network-manager stop`, then execute `killall wpa_supplicant` to disable the network manager and to prepare the device for the configuration of the network by yourself.
- Connect your assigned AP via Ethernet to the netbook.
- Look at the settings of your AP in the list below.
- Execute `ifconfig -a` and look for `ethy`, which is the identifier of the Ethernet interface. The value `y` is used in the subsequent commands.
- Execute `ifconfig ethy 192.168.xxx.2/24 up` with `xxx` corresponding to your AP settings.
- Execute `ping 192.168.xxx.1` to test the response of the AP. Press `Ctrl+C` to cancel pinging.

Depending on its number, your AP is configured according to the following settings:

- `AP 1: SSID: 'MOBNET Exercise 1', Channel 1, Subnet: 192.168.100/24, Static IP: 192.168.100.1`
- `AP 2: SSID: 'MOBNET Exercise 2', Channel 6, Subnet: 192.168.110/24, Static IP: 192.168.110.1`
- `AP 3: SSID: 'MOBNET Exercise 3', Channel 11, Subnet: 192.168.120/24, Static IP: 192.168.120.1`
- `AP 4: SSID: 'MOBNET Exercise 4', Channel 14, Subnet: 192.168.130/24, Static IP: 192.168.130.1`

Network Setup - Clients and Monitor

The clients and the monitor use their own wireless network interfaces to access the wireless network.

Configuring a Client

To configure a netbook to run as a client, perform the following steps:

- Log in, get a root shell, and disable the network manager and wpa_supplicant (as described above for the server).
- Execute `ifconfig -a` and look for `wlany`, which is the identifier of the wireless network interface. The value `y` is used in the subsequent commands.
- If the wireless interface was previously activated, deactivate it: `ifconfig wlany down`
- Set the network interface to managed mode: `iwconfig wlany mode managed`
- Set the ESSID: `iwconfig wlany essid "MOBNET Exercise x"`, where `x` is the number of your AP.
- Enable the wireless network interface: `ifconfig wlany 192.168.xxx.z/24 up`, where `xxx` corresponds to the subnet address of your AP and `z` is an arbitrary unique number within your network in the range from 3 to 254.
- Check the configuration: `iwconfig`

Configuring a Monitor

To configure the notebook as a monitoring station, perform the following steps:

- Log in, get a root shell, disable the network manager and wpa_supplicant and look up the wireless network interface identifier `wlany` (as described above).
- If the wireless interface was previously activated, deactivate it: `ifconfig wlany down`
- Set up the monitor mode: `iwconfig wlany mode monitor`
- Enable the wireless interface: `ifconfig wlany up`
- Set the channel to the channel `c` of your AP: `iwconfig wlany channel c`
- Close the root shell: `exit`
- Run wireshark as the `seemoo` user: `wireshark`
- Alternatively, you can run Wireshark from the menu bar (type `wireshark` into the field and press enter).
- Select `wlany` from the offered interfaces in Wireshark.

Tasks

Task 1: Maximum Throughput and Saturation Throughput in Practice

Bianchi concentrates on the saturation throughput in his analysis. Fig. 3 of his paper shows simulation results of a wireless network with 20 nodes under conditions of increasing offered load. He demonstrates that there is a maximum throughput for a particular setting of the offered load, and that for further increasing offered load, the achieved throughput becomes unstable, i.e., it slightly decreases.

Assess the throughput behavior of 802.11 practically with clients sending UDP traffic to the server. What maximum throughput and what saturation throughput is achieved? For which offered load is the maximum throughput achieved? Start with a single client first, then conduct the experiment with multiple clients sending at the same time.

Task 2: Fairness criterion of the 802.11 MAC

The key assumption of Bianchi's model is that each packet collides with constant and independent probability p. This immediately translates to an equal stationary chance for all stations to successfully access the medium on a transmission attempt, which we refer to as the fairness criterion of the 802.11 MAC. Assess the throughput of IEEE 802.11 under saturated and unsaturated conditions while

- varying the offered load of different clients.
- varying the frame length of different clients.

Explain your observations by means of the fairness criterion of the 802.11 MAC.

Task 3: RTS/CTS Overhead

Assess the overhead of RTS/CTS by comparing its performance to the basic access mechanism under saturated conditions. Run an experiment with a single client first, then with multiple clients sending at the same time. For which parameter settings does the overhead become most obvious, when is it negligible? Relate your observations to the results of Bianchi.

Task 4: 802.11 MAC Fragmentation

802.11 supports fragmentation at the MAC layer. How does it work and what was it originally meant for? Is it possible to achieve a better throughput fairness by fragmenting frames at the MAC layer? What are the limitations of this technique to harmonize the clients' throughputs? Compare the jitter and saturation throughput in both cases with and without fragmentation. Also try out different fragmentation threshold settings. Plot your observations and reason about them.

Additional Information

During the course of this exercise you will use *Wireshark* to capture WLAN frames and to analyze the workings of the 802.11 MAC layer. Information can be found at `http://www.wireshark.org/`. The Wireshark wiki offers, among other things, an overview of the capture setup process: `http://wiki.wireshark.org/CaptureSetup`.

Acknowledgements

This exercise is based in part on an exercise held during the summer term 2006 at Technische Universität Kaiserslautern.