Universität Paderborn
Computer networks group

Winter term 2017/2018

Mobile communications

Homework assignment 6: OFDM

Due date: 2017-11-29

---

1. **OFDM Waterfilling in different SNR regimes**

(a) Describe an algorithm to solve the waterfilling problem – i.e., find the concrete value for μ given a set of subcarriers qualities h_i and, thence, the actual values for P_i.

Solution: Recall: We know from class that

$$1 \stackrel{!}{=} \sum_{i=1}^{N} P_i = \sum_{i=1}^{N} \max\{0, \frac{1}{\mu} - \frac{1}{\alpha_i}\}$$

must hold for the given, fixed channel coefficients α_i. We hence just have to find μ.

Idea: sort $1/\alpha_i$, increase $1/\mu$ in jumps from one $!/\alpha_i$ to the next, see where the sum becomes bigger than one. Find detail in between. So:

- Reorder α_i in descending order, or $1\alpha_i$ in increasing order.
- Find smallest index i^* such that $\sum_{i=1}^{N} \max\{0, 1/\alpha_i^* - 1/\alpha_i\} \leq 1 < \sum_{i=1}^{N} \max\{0, 1/\alpha_{i^*+1} - 1/\alpha_i\} \leq 1$
- With that, the number of terms in the sum is clear, and it turns into a simple linear equation: $1 + \sum_{i=1}^{i^*} = i^* \cdot \frac{1}{\mu}$.
- Example: Say, $\alpha_1 = 5, \alpha_2 = 3, \alpha_3 = 1$. Then:
  - $i^* = 1$: 1/5 - 1/5 < (1/3 - 1/5) + (1/3-1/3) < 1. So i^* is still too small .
  - $i^* = 2$: (1/3-1/5) + (1/3-1/3) < 1 < (1-1/5) + (1 - 1/3) + (1-1) This is the right i^*.

  Hence, $1/\mu$ must be between 1/3 and 1. That means, in particular, that only the first two parts of the sum play a role (because $1/mu - 1$ is negative, and so would any other terms.

  Hence, we only need to solve $1 \stackrel{!}{=} (1/\mu - 1/5) + (1/\mu - 1/3) \leftrightarrow 1 + 1/5 + 1/3 = 2/\mu$

(b) Assuming that you know that the channel has very good quality over all subcarrier – it works in the *high-SNR regime*. Is there a simplification plausible that achieves nearly optimal results? (Try to think in terms of the waterfilling visualization.)

(c) Similarly: what happens when the channel is in the *low-SNR regime*, i.e., all subcarriers are pretty bad. What would then be a simplification to compute nearly optimal P_i?

Solution: From Tse, Figure 5.25:

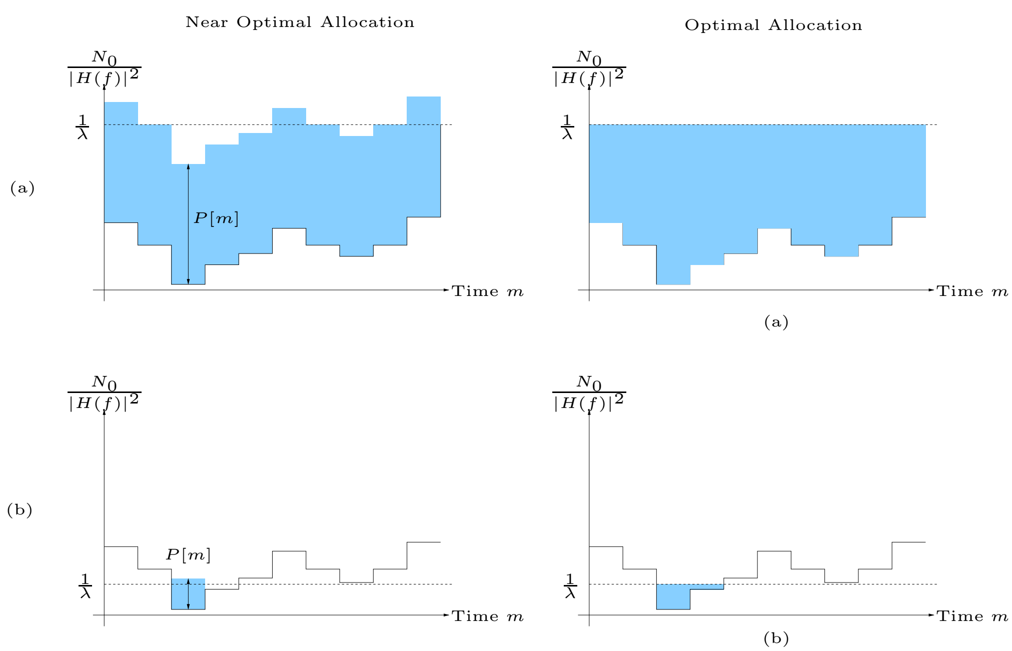

Figure 5.25: (a) High SNR: Allocating equal powers at all times is almost optimal. (b) Low SNR: Allocating all the power when the channel is strongest is almost optimal.

2. **Hands on: Orthogonal Frequency Division Multiplexing (OFDM)**

In OFDM the available bandwidth for a communication is split into S different subchannels where each subchannel is independently used to transmit data between two nodes A and B.

To successfully decode a data packet, a BER of at most 10^{-6} is needed. Table 1 shows the possible modulation schemes and corresponding SNR threshold to achieve a BER < 10^{-6}. If you transmit with a given modulation over a channel whose SNR is worse than the threshold, you must assume that this transmission fails.

Tabelle 1: Possible modulation schemes and their corresponding SNR threshold

M-PSK	4	8	16	32
SNR [dB]	10.3	14.1	18.2	23.3

On our website you will find a csv-file ("channels.csv") containing SNR information of 64 subchannels for 10000 time steps.

Design an algorithm that uses SNR information of the different subchannels to select one of the modulation techniques for each timestep and each subchannel. To do so, assume (a) you know the SNR for each time step perfectly and (b) you can change the modulation every timestep. Use that modulation to transmit data between the nodes.

Develop different algorithms. Possible relaxations/constraints could be:

- All subchannels must use the same modulation (see next assignment as well)
- Assume that switching modulations incurs overhead (it does, in practice), so only allow to change modulation every T timesteps.
- Try to incorporate forward error correction schemes (difficult!) to tolerate occasional violations of the SNR threshold, giving you the option to make more aggressive choices in modulation selection.
- Assume that you do not know the actual SNR trace when making decisions (that's highly unrealistic knowledge about the future), but only some erroneous estimates of it, or only some occasional values (outdated ones?).
- ...

Plot the data rate your algorithms achieve.

Hint: The table shows M-PSK modulations; the number of bits per symbol is only $\log_2(M)$!

Hint: You may want to use either Python/Numpy, Matlab or Gnu Octave to implement and test your algorithm.

The following assignment is really optional, but it should make sense to at least think it through.

3. **Hands on Part 2: Orthogonal Frequency Division Multiplexing (OFDM) reloaded**

In OFDM the available bandwidth for a communication is split into S different subchannels where each subchannel is independently used to transmit data between two nodes A and B.

Consider the following scenario: A and B are deployed with low-cost transceivers which are able to transmit in OFDM mode but they can only use the same modulation for all subchannels at a time. You are using an FEC that is capable to correct up to a 20% bit error rate and the transceivers are able to send and decode M-QAM with $M \in \{2, 4, 8, 16, 32, 64, 128, 256, 512\}$. This time the BER of a given modulation at a specific SNR is given by

$$\text{BER} = 2 \cdot \left(1 - \frac{1}{\sqrt{M}}\right) \cdot \textbf{erfc}\left(\sqrt{\frac{3}{2 \cdot (M-1)}} \cdot \frac{E_S}{N_0}\right) - \left(1 - \frac{2}{\sqrt{M}} + \frac{1}{M}\right) \cdot \textbf{erfc}^2\left(\sqrt{\frac{3}{2\,(M-1)}} \cdot \frac{E_S}{N_0}\right)$$

where erfc is the complementary error function given by

$$\textbf{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_x^\infty e^{-t^2} dt$$

On our website you will find a csv-file (snr.cvs) containing SNR information of 64 subchannels for an interval of 10000 timesteps.

(a) Design an algorithm that uses SNR information of the different subchannels to select one of the modulation techniques for each timestep. To be more realistic than in the previous assignment, assume channel sensing takes you 10 timesteps. So every time your algorithm accesses the SNR data you get a penalty of 10 timesteps in which you are not able to send data. The size of a packet is fixed at 48 bytes (including the FEC redundancy data).

Note: During the transmission of a paket you must not change the modulation.

(b) Implement your algorithm in either Matlab, Gnu Octave, or other tools. At the receiver draw a diagram of the decodable and non-decodable packets over time.

(c) Now assume there are two nodes B and C to which A wants to send data. Extend your algorithm to decide which subchannels should be used to send to B or C. Explain the type of fairness your algorithm provides.

For the transmission $A \rightarrow B$ use the SNR provided by snr.csv; for $A \rightarrow C$ use snr2.csv (also available on our homepage).