Computer Networks

Exercise Session 01

Prof. Dr. Oliver Hahm

Frankfurt University of Applied Sciences
Faculty 2: Computer Science and Engineering
oliver.hahm@fb2.fra-uas.de
https://teaching.dahahm.de

October 21, 2022

General Schedule

All exercises will follow this general schedule

- Identify potential understanding problems
  → Ask your questions
  → Recap of the lecture
- Address the understanding problems
  → Answer your questions
  → Repeat certain topics
- Walk through the exercises/solutions → Some hints and guidance
  → Work time or presentation of results

About the lecture

Are there any questions regarding . . .

- . . . the organization of the course?
- . . . the information about the lecture?
- . . . the objectives of the course?

Components and Terms

You have seen ...

- what the general purpose of a Computer Network is
- which components are required for a Computer Network
- how Computer Networks can be distinguished by their dimension
- the difference between unicast, broadcast, multicast, and anycast
- what connection-orientation means
- what the directional dependence of data transmission is
- what bandwidth, throughput, goodput, and latency are

Reference Models

You have seen . . .

- how a Computer Network can be broken down into layers
- what a reference model is and which relevant ones exist
- which layers exist in the hybrid reference model and what tasks they have

Topologies

You have seen . . .

- what a topology is
- what the difference between the physical and the logical topology is
- the advantages and drawbacks of the different topologies
- which topologies are used in current networks

Any other questions left?

Exercise 1: Data Encoding

- How many bits do we need to encode letters (lower case $\rightarrow$ a..z)?

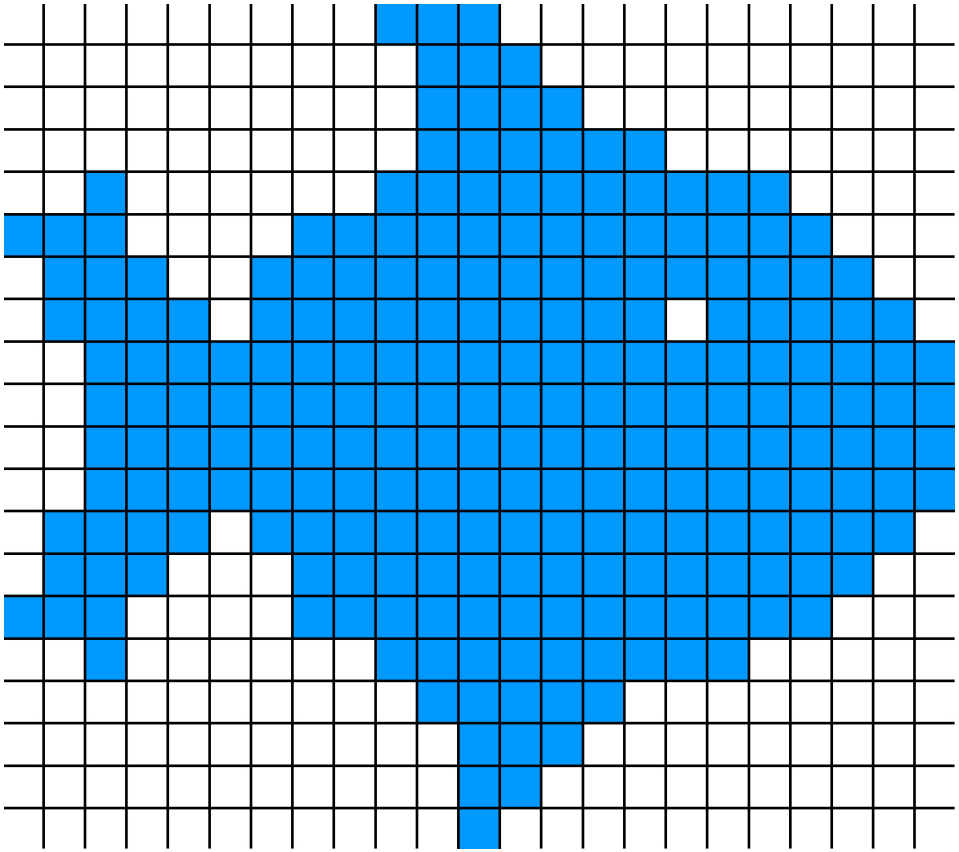
Exercise 1: Data Encoding

- How many bits do we need to encode letters (lower case → a..z)?
- ⇒ 26 letters ⇒ smallest possible power of 2: $2^5 = 32$
  ⟶ 5 bits are required

Possible Encoding

```
a → 0x00 // 0b00000
b → 0x01 // 0b00001
c → 0x02 // 0b00010
d → 0x03 // 0b00011
e → 0x04 // 0b00100
f → 0x05 // 0b00101
g → 0x06 // 0b00110
h → 0x07 // 0b00111
i → 0x08 // 0b01000
j → 0x09 // 0b01001
k → 0x0a // 0b01010
l → 0x0b // 0b01011
m → 0x0c // 0b01100
n → 0x0d // 0b01101
o → 0x0e // 0b01110
p → 0x0f // 0b01111
q → 0x10 // 0b10000
r → 0x11 // 0b10001
s → 0x12 // 0b10010
t → 0x13 // 0b10011
u → 0x14 // 0b10100
v → 0x15 // 0b10101
w → 0x16 // 0b10110
x → 0x17 // 0b10111
y → 0x18 // 0b11000
z → 0x19 // 0b11001
```

Exercise 3: Bitmapped Images



- Simple way to store an image
- Each pixel is stored separately
- The more colors, the more bits are required to store one pixel

Exercise 4: SI Units vs. IEC Units

- The International System of Units (SI) defines the prefixes *kilo*, *mega*, *giga* etc. as powers of 10
- Traditionally these prefixes has been used for powers of 2
  $\rightarrow$ 1 kB referred to 2^{10} bytes
- In 1996 the International Electrotechnical Commission (IEC) introduced new prefixes *kibi*, *mebi*, *gibi* etc. for these powers of 2
- While persistent storage is typically expressed using SI prefixes correctly, some operating systems (e.g., Microsoft Windows) still label powers of 2 with SI prefixes
- On most UNIX-like systems one can choose