Exercise Sheet 3

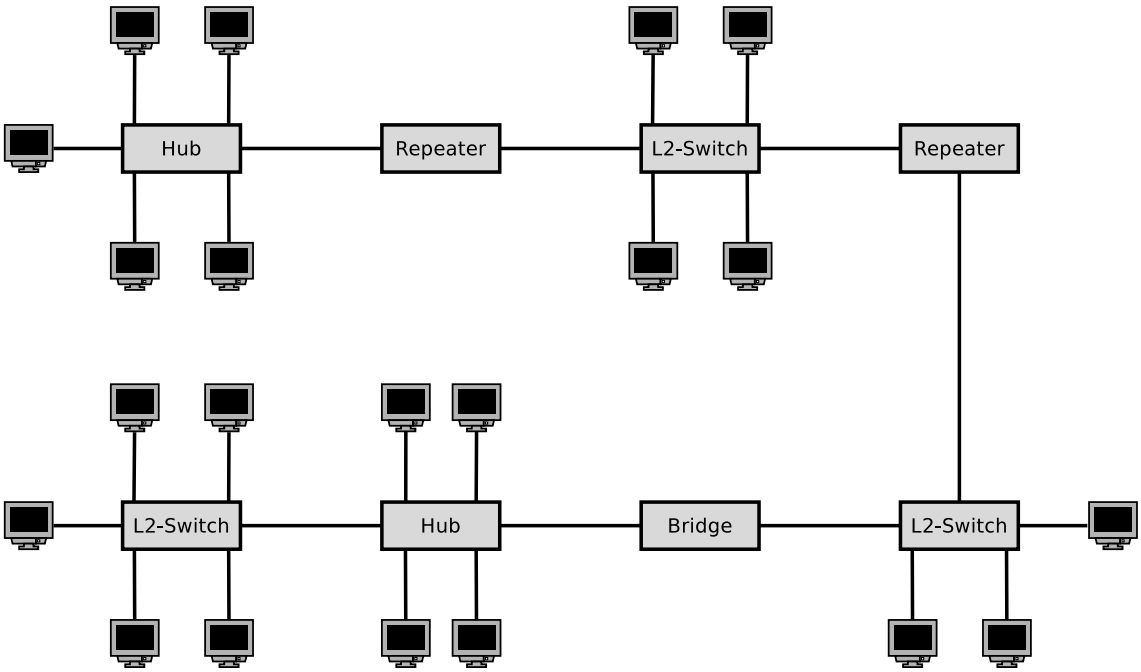
Exercise 1 (Bridges and Switches)

1. What is the purpose of **Bridges** in computer networks?
2. How many **interfaces** („*Ports*“) provides a Bridge?
3. What is the major difference between **Bridges** and **Layer-2-Switches**?
4. Why do Bridges and Layer-2-Switches not require **physical or logical addresses**?
5. Name at least two **examples** of Bridge implementations.
6. What is the advantage of **Learning Bridges** in contrast to „dumb“ Bridges?
7. What information is stored in the **forwarding tables** of Bridges?
8. What happens, if for a network device, no entry exists in the **forwarding table** of a Bridge?
9. Why do Bridges try to avoid **loops**?
10. What protocol use Bridges to **handle loops**?
11. What is a **spanning tree**?
12. What information contains the **Bridge ID** according to the IEEE?
13. What is the difference between the **Bridge ID** according to the IEEE and the **Cisco extended version** of the Bridge ID?
14. How many priority values can be encoded with the **Bridge ID** according to the IEEE?
15. How many priority values can be encoded with the **Cisco extended version** of the Bridge ID?
16. What is a **Bridge Protocol Data Unit** (BPDU) and for what is it used?
17. What is the selection criteria for determining, whether a Bridge becomes the **Root Bridge**?
18. What is a **Designated Bridge** and what is its task?
19. How many **Designated Bridges** does a computer network contain?
20. What is the selection criteria for determining, whether a Bridge becomes a **Designated Bridge**?

21. What is the impact of Bridges and Layer-2-Switches on the **collision domain**?
22. What is a **switched network**?
23. Name an advantage of a **switched network**.

Exercise 2 (Collision Domain)

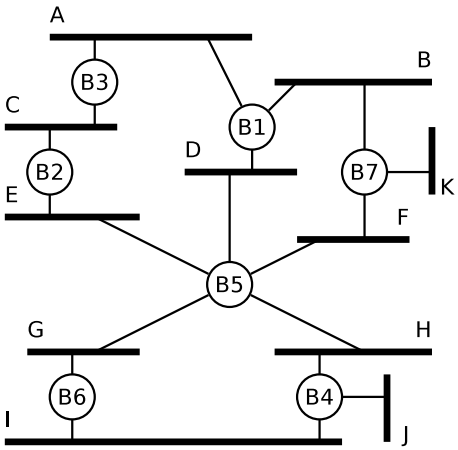
Sketch in the diagram of the network topology all **collision domains**.



Exercise 3 (Spanning Tree Protocol)

The figure shows the physical connections of a network. All Bridges boot up at the same time after a power failure. Highlight in the figure which ports and Bridges are not used when the Spanning Tree Protocol is used.

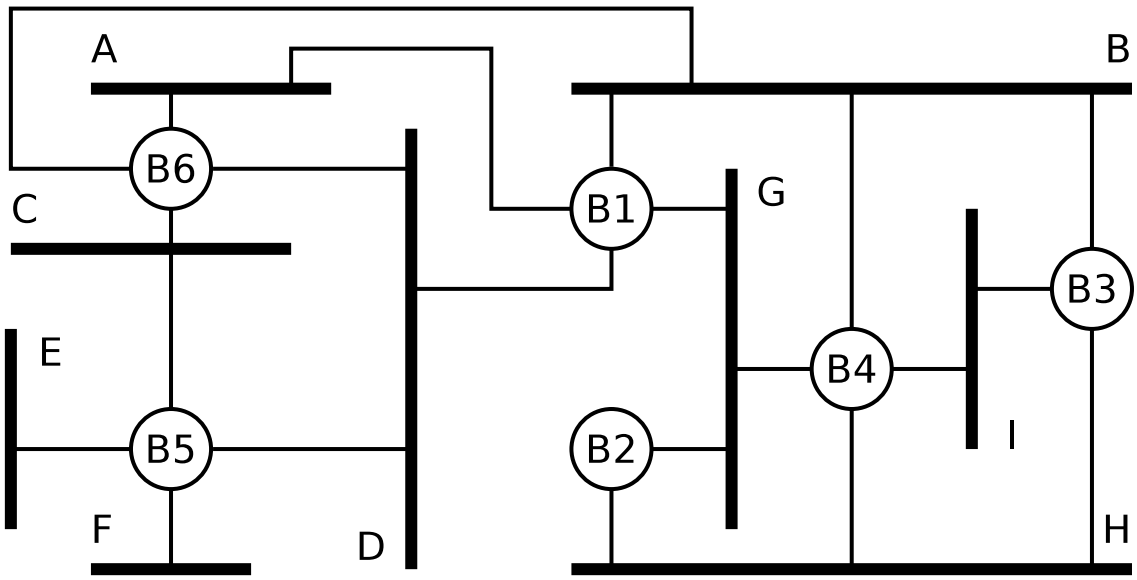
Attention: If multiple paths from a network to the root bridge have the same distance, then take the bridge IDs as decision criterion. The smaller the ID of a bridge is, the higher is its priority.



Exercise 4 (Spanning Tree Protocol)

The figure shows the physical connections of a network. All Bridges boot up at the same time after a power failure. Highlight in the figure which ports and Bridges are not used when the Spanning Tree Protocol is used.

Attention: If multiple paths from a network to the root bridge have the same distance, then take the bridge IDs as decision criterion. The smaller the ID of a bridge is, the higher is its priority.



Exercise 5 (Addressing in the Data Link Layer)

1. The format of what **addresses** is defined by Data Link Layer protocols?

   ☐ physical network addresses ☐ logical network addresses

2. How are **physical network addresses** called?
3. What protocol uses Ethernet for the **address resolution**?
4. Who receives a frame with the **destination address** `FF-FF-FF-FF-FF-FF`?
5. What is **MAC spoofing**?

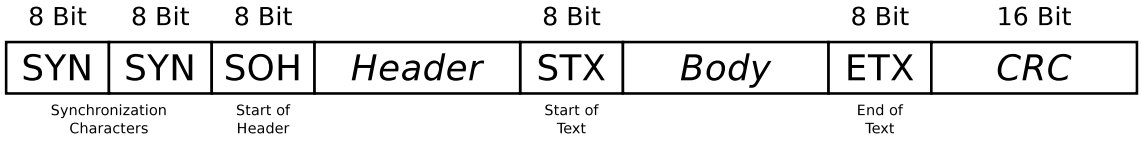
Exercise 6 (Framing)

1. One way to mark the frames' borders is via **character count in the frame header**. Name a potential issue that can arise from this method.
2. One way to mark the frames' borders is via **Byte Stuffing**. Name a drawback of this method.
3. Why work up-to-date Data Link Layer protocols, such as Ethernet and WLAN, **bit-oriented and not byte-oriented**?
4. What information contains an **Ethernet frame**?

   ☐ Sender IP address
   ☐ Sender MAC address
   ☐ Hostname of the receiver
   ☐ Information about the Transport Layer protocol used
   ☐ Preamble to synchronize the receiver
   ☐ Port number of the receiver
   ☐ CRC checksum
   ☐ Information about the Application Layer protocol used
   ☐ VLAN tag
   ☐ Receiver MAC address
   ☐ Receiver IP address
   ☐ Information about the Network Layer protocol used
   ☐ Hostname of the sender
   ☐ Signals, which are transmitted via the transmission medium
   ☐ Port number of the sender

Exercise 7 (Byte Stuffing)

The Data Link Layer splits the bit stream from the Physical Layer into frames. The character-oriented protocol BISYNC uses control characters to mark the structure of the frames. The start of a frame highlights the character `SYN`. The start of the header highlights the character `SOH` (*Start of header*). The payload is located between `STX` (*Start of text*) and `ETX` (*End of text*). The figure shows the structure of BISYNC frames:



Control character	SOH	STX	ETX	DLE	SYN
Hexadecimal notation	01	02	03	10	16

If the payload (body) contains the control characters `ETX` and `DLE` (*Data Link Escape*), they are protected (*escaped*) by the Data Link Layer protocol with a stuffed `DLE` caracter. A single `ETX` in the payload area is represented by the sequence `DLE ETX`. The `DLE` character itsef is represented by the sequence `DLE DLE`.

Mark the payload inside the following BISYNC frames?

1. `16 16 01 99 98 97 96 95 02 A1 A2 A3 A4 A5 03 A0 B7`
2. `16 16 01 99 98 97 96 95 02 05 04 10 03 02 01 03 76 35`
3. `16 16 01 99 98 97 96 95 02 10 03 10 10 10 03 03 92 55`
4. `16 16 01 99 98 97 96 95 02 10 10 10 10 10 03 01 02 A1 03 99 B2`

Source: Jörg Roth. Prüfungstrainer Rechnernetze. Vieweg (2010) and Wikipedia

Exercise 8 (Bit Stuffing)

The Data Link Layer protocol HDLC (High-Level Data Link Control) uses Bit Stuffing. If the sender discovers 5 consecutive 1 bits in the bitstream from the Network Layer, it *stuffs* a single 0 bit into the outgoing bit stream. If the receiver discovers 5 consecutive 1 bits, followed by a single 0 bit in the bit stream from the Physical Layer, it removes (*destuffs*) the 0 bit.

Give the encoding for each one of the following bit sequences, when the sender *stuffs* after 5 consecutive 1 bits a single 0 bit into the bit stream from the Network Layer.

1. `01111110 10100111 11111000 11110010 10011111 10111111 11100101`

2. `00111111 01110001 11110011 11111100 10101010 11001111 11100001`
3. `11111111 11111111 11111111 11111111 11111111 11111111 11111111`

Exercise 9 (Error Detection – CRC)

1. Calculate the frame to be transferred.

   ```
   Generator polynomial: 100101
   Payload: 11010011
   ```

2. Check, if the received frame was transmitted correctly.

   ```
   Transferred frame: 1101001110100
   Generator polynomial: 100101
   ```

3. Check, if the received frame was transmitted correctly.

   ```
   Transferred frame: 1101001111100
   Generator polynomial: 100101
   ```

4. Calculate the frame to be transferred.

   ```
   Generator polynomial: 100101
   Payload: 10110101
   ```

5. Check, if the received frame was transmitted correctly.

   ```
   Transferred frame: 1011010110110
   Generator polynomial: 100101
   ```

6. Check, if the received frame was transmitted correctly.

   ```
   Transferred frame: 1011010110100
   Generator polynomial: 100101
   ```

7. Check, if the received frame was transmitted correctly.

   ```
   Transferred frame: 1010010110100
   Generator polynomial: 100101
   ```

8. Calculate the frame to be transferred.

   ```
   Generator polynomial: 100000111
   Payload: 1101010101110101
   ```

9. Check, if the received frame was transmitted correctly.

```
Transferred frame: 11010101011111010110111
Generator polynomial: 100000111
```

10. Check, if the received frame was transmitted correctly.

```
Transferred frame: 11010101011101010110111
Generator polynomial: 100000111
```

Exercise 10 (Error Correction – Simplified Hamming Code)

Transmission errors can be detected via CRC checksums. If it is important to not only recognize errors, but also to be correct them, then the data to be transmitted must be encoded in a way, that error-correction is possible. Error correction can be realized e.g. via the **Simplified Hamming Code** we discussed in the computer networks course.

1. A message of 8 bits payload (`10011010`) need to be transfered. Calculate the message, that will be transmitted (payload inclusive parity bits).
2. The following messages have been received. Verify, if they were transmitted correctly.
   a) `00111101`
   b) `101110100010`
   c) `001101100100`
   d) `0001101100101101`

Exercise 11 (Media Access Control)

1. Why do computer networks use protocols for **media access control**?
2. Which media access control method is implemented by **Ethernet**?
   ☐ Deterministic media access control
   ☐ Non-deterministic media access control
3. Which media access control method is implemented by **Token Ring**?
   ☐ Deterministic media access control
   ☐ Non-deterministic media access control

4. Which media access control method is implemented by **WLAN**?

   ☐ Deterministic media access control
   ☐ Non-deterministic media access control

5. What is the advantage of the media access control method of **Token Ring** in contrast to the media access control method of **Ethernet**?

6. Why use Ethernet and WLAN different **media access control methods**?

7. How do Ethernet devices react, when they detect a **collision**?

8. Why is it important that the transmission of a frame is not completed when a collision occurs in an **Ethernet** network?

9. What is done to ensure that the transmission of a frame is not completed when a collision occurs in an **Ethernet** network?

10. Which two **special characteristics** of the transmission medium in **wireless networks** cause **undetected collisions** at the receiver?

11. Describe both **special characteristics** of subtask 10.

12. What is the **Network Allocation Vector** (NAV) for what purpose is it used?

13. What is the **Contention Window** (CW) and for what purpose is it used??

14. Name a benefit and a drawback of using the control frames **Request To Send** (RTS) and **Clear To Send** (CTS)?

Exercise 12 (Address Resolution Protocol)

1. What is the function of the **Address Resolution Protocol**?

2. What is the **ARP cache**?