Exercise Sheet 4

Exercise 1 (Error Control)

1. An upper-layer packet is split into 16 frames, each of which has a 75 percent chance of arriving undamaged. If no error control is done by the data link protocol, how many frames are required to be sent on average to get the entire thing through?

2. If error detection is done by the data link protocol, how many frames are required?

3. If forward error correction is done, how many frames are required?

4. Given the following valid codewords on the data link layer:

   - $w_1 =$ `0001 1111`
   - $w_2 =$ `0111 1111`
   - $w_3 =$ `1100 1111`
   - $w_4 =$ `1011 1111`
   - $w_5 =$ `0001 0000`
   - $w_6 =$ `0111 0000`
   - $w_7 =$ `1100 0000`
   - $w_8 =$ `1011 0000`

   What is the minimum Hamming distance of this code? How many flipped bits could be detected? How many of them could be automatically be corrected?

5. Most data link layer protocols put the CRC in the end of a frame (*trailer*) rather than in the beginning (*header*). Why?

6. For the data `0xDE 0xAD 0xBE 0xEF` the *CRC16-CCITT* results in `0x19 0x15`. Which of the following blocks of data will certainly result in a different CRC16-CCITT checksum?

   - `0xDE 0xAD 0xBE 0xFF`
   - `0xDE 0xAD 0xBE 0xE8`
   - `0xFF 0xFD 0xBE 0xEF`
   - `0x9E 0xAD 0xBE 0xED`
   - `0xDE 0xAD 0xBE 0xD0`

Source: Andrew Tanenbaum, Computer Networks, Fourth Edition. Pearson (2003)

Exercise 2 (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 3 (Media Access Control)

1. Why do computer networks use protocols for **media access control**?
2. Describe the **differences** between contention-based (non-deterministic) and contention-free (deterministic) media access control.
3. Which media access control method is implemented by **Ethernet**?

   ☐ Deterministic media access control
   ☐ Non-deterministic media access control
4. Which media access control method is implemented by **Token Ring**?

   ☐ Deterministic media access control
   ☐ Non-deterministic media access control
5. Which media access control method is implemented by **WLAN**?

   ☐ Deterministic media access control
   ☐ Non-deterministic media access control
6. What is the advantage of the media access control method of **Token Ring** in contrast to the media access control method of **Ethernet**?
7. Why use Ethernet and WLAN different **media access control methods**?
8. How do Ethernet devices react, when they detect a **collision**?
9. Explain why it is important that the transmission of a frame is not completed when a collision occurs in an Ethernet network.
10. Explain what is done to ensure that the transmission of a frame is not completed when a collision occurs in an **Ethernet** network.
11. Why is the MAC protocol less relevant for modern Ethernet networks?

Exercise 4 (p-persistent CSMA, CSMA/CD, and CSMA/CA)

1. Which prerequisite needs to be fulfilled to use a CSMA MAC protocol?
2. For p-persistent CSMA the size of p determines the performance of the network. In which cases is preferable to use a higher value for p?
3. Explain how the actual delay of a network with CSMA/CD as MAC protocol is affected by the random number generator of the host.

4. The maximum number of transmission attempts may be configurable. What are the consequences of an increased number?

5. In CSMA/CD only data frames and JAM signals are required. In CSMA/CA an additional frame type is needed. Which one? Why is it required?

Exercise 5 (Address Resolution Protocol)

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

2. What are the main differences between **ARP** and **NDP**?

3. Which effect does *MAC spoofing* have on ARP and NDP?

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

Exercise 6 (Do some research)

1. For **Wireless Sensor Networks** MAC protocols have played an important role. Elaborate on its impact for these networks and name at least two WSN specific MAC protocols.

2. Why can CRC not be used for digital signatures?

3. As a convention for configuration you may find **7E1**. Explore the context of this configuration setting and explain its meaning.

4. Explain why it sometimes happen that a host sends an ARP request for its own IPv4 address.

Exercise 7 (Lab Exercise: Address Resolution)

The *Address Resolution Protocol (ARP)* is an auxiliary protocol which maps IP address from the *Network Layer* to physical addresses from the *Data Link Layer*.

1. After booting and logging in, bring up your Ethernet network interface by opening a terminal emulator and executing
   `sudo ip link set eno2 up`

2. Start *Wireshark* and monitor the traffic on the device *eno2*. (You can set a filter rule to ignore *mdns* traffic by setting the filter to `!mdns`.)

3. In the terminal emulator and configure an IPv4 address by executing the following command:
   `sudo ip addr add 10.0.<T>.<C>/24 dev eno2`.
   Replace `<T>` by the number of your table and `<C>` by the number of the computer (counting from right to left).

4. Open a second terminal emulator and run the command
   `watch -n1 ip neigh`

5. In the first terminal run the command
   `ping 10.0.<T>.<N>`
   Replace `<T>` by the number of your table and `<N>` by the number of the computer of your neighbor.

6. Observe what is happening in Wireshark and in the other terminal window.

7. On your neighbor's computer:

   - Figure out the **MAC address** by running
     `ip link show eno2`
   - Change the MAC address via
     `sudo ip link set eno2 addr <NEW MAC ADDRESS>`

8. What do you observe over time - both in the terminals and in Wireshark.