Lab 3: Addressing

1 Introduction to lab 3

In this third lab you will develop the necessary mechanisms to enable communication in a Local Area Network (LAN) with multiple access.

Your assignment in this lab is to implement the necessary functionality to succeed in making the two user nodes, i.e. the *Master Node* and the *Development Node*, communicate as in the physical layer lab, but this time in a multi-user environment via an *Access Point*. The *Master Node* and the *Access Point* will be provided for you and is fully functional according to the specifications found in the *Reference Manual*[1]. As in the previous labs, you will be working on the *Development Node*.

As in previous labs, the task is to remotely light one of the *Master Node*'s Light Emitting Diodes (LEDs) as selected on the *Development Node*. In this lab however, the two user nodes, i.e. the *Development Node* and the *Master Node*, can not communicate directly with each other; the communication has to go via a third node, the *Access Point*. The *Access Point* repeats all frames it receives out on the common link. Thus, the user nodes will receive copies of all frames they are transmitting. For this to work, addressing is required and each user node has to be assigned a unique 4-bit binary address. The nodes must only act on frames addressed to it.

The *Access Point* works in a store-and-forward mode. It catches frames and transfers whatever it receives (including possible errors) to the user nodes. The *Access Point* works in a promiscuous mode; it repeats any frame sent on the LAN. As with all repeaters, the *Access Point* has no address of its own.

2 Learning outcomes

After performing this lab you should have gained understanding of

- the concept and implementation of link frame addresses.

3 Lab setup

Starting from lab three, there should be three Arduinos on your working bench in the physical environment. The *Development Node* should be connected to the computer, and

the *Master Node* and *Access Point* should be connected to a USB power outlet. It is important that the nodes are aligned so that both the *Development Node* and the *Master Node* can send to and receive from the *Access Point*, while not being able to send to each other. Place the *Development Node* and the *Master Node* side by side, both facing the *Access Point*, somewhat like you see in Figure 1.

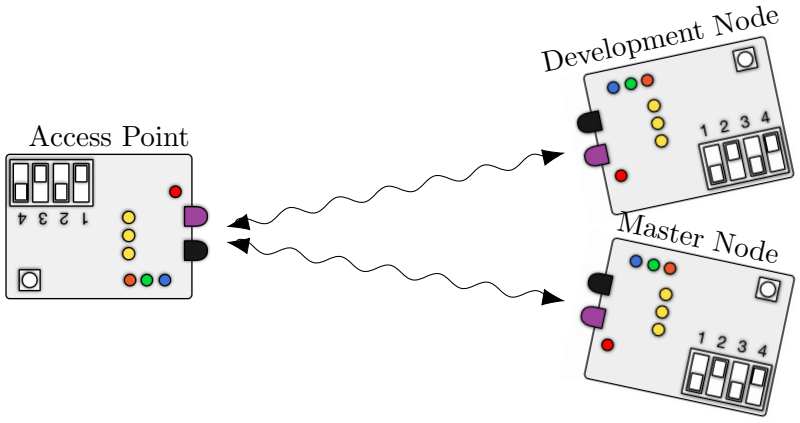

Figure 1: The setup of the network.

4 Lab tasks

The third and fourth lab is about the implementation of the necessary Data Link Layer (L2) mechanisms. The primary practical objectives of this lab are to:

- Implement the necessary functionalities in the *Development Node* such that the two nodes are able to communicate similarly to the physical layer lab, but this time via the *Access Point*.
- The *Development Node* should be able to construct a frame with a correct address for transmission and assess the address of a received frame

5 Task 1: Addressing

Data Link Layer (L2) frames should have assigned appropriate source and destination node addresses. The three nodes work in a LAN with multiple access; all frames repeated by the *Access Point* are received by all user nodes in the network, even the originating node. The purpose of addressing is that only frames addressed to the selected node should be processed by that node. A frame not addressed to a receiving node should silently be dropped. The task is done when the setup works with three nodes and when addressing is turned on on the nodes.

5.1 Hints

- Make sure that the Cyclic Redundancy Check (CRC) functions are de-activated both in the *Master Node* and the *Access Point*. See the *Reference Manual* for details.
- The two nodes have to be assigned unique addresses. The L2 frame from the *Development Node* to the *Master Node* must have the source address set to the *Development Node*'s address, and the destination address should be the *Master Node*'s address. And of course, vice versa.
- The *Master Node*'s address is set with this nodes Dual In-line Package (DIP) switches. Note that only DIP switches 1 and 2 are used for setting the *Master Node*'s address, see the description of the *Master Node* in the *Reference Manual* for details.
- The *Development Node*'s own address should be assigned using the `Shield::setMyAddress(int address)` method in the `setup()` function of the *Development Node*'s sketch. You can then later get the address by using the method `Shield::getMyAddress()`. Note that the addressing field occupies 4 bits, so the address value ranges from 0 to 15.
- At the application layer, read the current destination address from the DIP switches and store it in the message array - the interface between the application layer and Data Link Layer (L2).
  To read the DIP switches use the `Shield::get_address()` method.
  Use the message array when setting the corresponding frame field at the next layer.
- After the received framed has been decomposed into its individual fields, check that the destination field of the received frame contains the *Development Node*'s address. Only frames addressed to the *Development Node* should be processed further, other frames are silently dropped and the node should go back and wait for a new frame.
- Finally, you select which state should be the next one, depending on the outcome of the address check.

References

[1] Data Communication Lab: Reference Manual, 2023, 2024.