ESD5 – Fall 2024
Problem Set 2

Department of Electronic Systems
Aalborg University

September 19, 2024

Problem 1 – TDMA-Based Scheduling

In this problem, we are going to develop a functional medium access control (MAC) scheme in the context of a cellular network. Consider the system depicted in Fig. 1, where a base station (Basil) wants to communicate with three terminals: Zoya, Yoshi, and Xia. The communication occurs according to a *Time Division Multiple Access* (TDMA) scheme. This means that the time domain is sliced into time slots and that at some point each slot is allocated to a single terminal to which Basil wants to transmit or receive data. Note that Basil owns a powerful role since she can control who is able to speak/hear. To develop the most simple functional MAC scheme, we will develop 4 different frame types, where each executes a different function. Throughout the problem, we will consider that Basil has the capacity of serving 3 users at most.

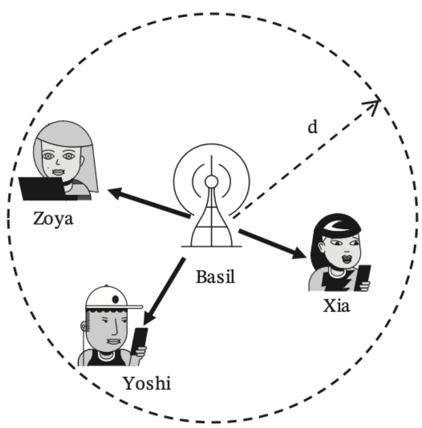

Figure 1: A cellular network that is operating according to a TDMA scheme.

(a) Basil needs to specify the frame that is meant to be sent in the frame header so that the users can understand what is happening at the moment. How many bits are necessary to differentiate 4 frame types?

Frame 1: Initial Access

In the beginning, Zoya, Yoshi, and Xia have not established a link with Basil and they are all in a receive state, that is, waiting for someone to speak to them so they can figure out what to do. Therefore, Basil needs to create a frame to enable the users to connect to her. Let's call this the link establishment frame. In this frame, there is no payload data being sent, just an invite packet from Basil asking the users: "does anyone want to connect?". After this invite packet, Basil changes to the receive mode in order to listen for some answers. If a user wants to connect, it should send an ACK to Basil. Basil then sends an ACK back to ensure that the link was established and with this ACK also a number between 1 and 3 in order to associate each user with a time slot that is going to be useful when performing data transmission. After this process, Zoya, Yoshi, and Xia can start to communicate with Basil.

(b) Enlist what can go wrong with this process. What do Basil and the users need to do when each one of the problems occurs?

Frame 2: Link Termination

Consider that Zoya, Yoshi, and Xia have links established to Basil. Assume that we now have a new user, Walt, that wants to communicate with Basil. In view that Basil can serve 3 users at most, the engineering question here is: "how can Walt be served by Basil?"

(c) Design a link termination frame that enables Basil to free resources when suitable.

Frames 3 and 4: Downlink/Uplink Transmissions

In wireless communication terminology, the transmitting direction of having Basil send data to the users is called *downlink*, while the other way around is called *uplink*. Basil needs to design a frame in order to differentiate the directions and specify who needs to listen/hear. Both downlink and uplink transmission frames contain 3 slots, which are assumed to agree to the size of a data packet.

(d) Let us consider a downlink transmission frame where Zoya, Yoshi, and Xia already have established links, and each one is associated with a single slot. Assume that the data rate of each data packet sent by Basil is 1 kbit/s. However, Xia needs to hear all the slots in order to receive the packet meant for her. What is the equivalent data rate of Xia?

A Very Simple Functional MAC Scheme

Now that we have defined all 4 frames, we can evaluate how this MAC scheme actually works.

(e) Draw a timing diagram showing the following steps:

- Zoya tries to establish a link.
- Basil transmits to Zoya.
- Zoya transmits to the Basil.
- Yoshi and Xia try to establish a link simultaneously.
- Basil transmits to Zoya.
- Xia tries to establish a link.
- Zoya and Xia transmit to Basil.

Make sure to specify the frame type used in each step.

Problem 2 – ALOHA Protocol

According to Fig. 2, consider a slotted ALOHA scheme involving five users competing to send their packets over 6 time slots. We assume that users are sending multiple copies of the same packet.

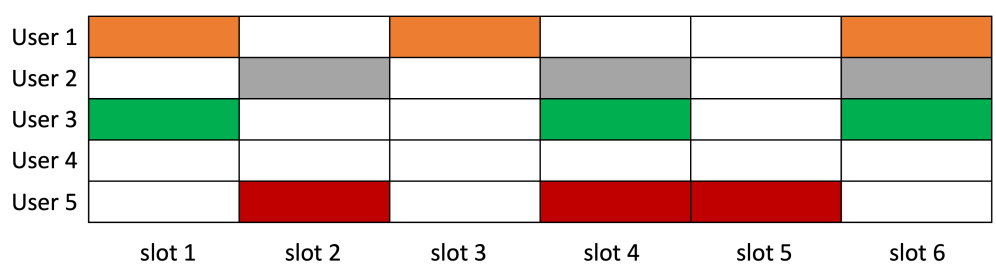

Figure 2: Example of slotted ALOHA. Each user sends multiple copies of the same packet.

(a) How many users have successfully sent their packets without colliding with one another? Who are those users?

(b) Consider the decoding technique of *Successive Interference Cancellation* (SIC), which can be applied at the receiver to improve the throughput of the communication. In principle, the SIC algorithm works by removing the replicas of the already resolved packets. For example in slot 1, assuming User 1's packet has been resolved, SIC can be used to exclude User 1's packet. From there, User 2's packet can be easily decoded without collision. So by assuming the use of the SIC technique, how many users' packets can be successfully decoded? What is the decoding order to achieve it?

For slotted ALOHA of N nodes, the throughput is $Np(1-p)^{N-1}$, where p is the transmission probability. The optimal p is $1/N$, resulting in a throughput of $(1-\frac{1}{N})^{N-1}$. Now, consider the following multiple access scheme that combines TDMA and slotted ALOHA. There are 20 users, separated into two groups, one of 4 users and the other of 16 users. Even time slots (*i.e.*, $0, 2, 4, \ldots$) are reserved for the 4-user group. Odd time slots (*i.e.*, $1, 3, 5, \ldots$) are reserved for the 16-user group. Contention within each group is resolved by the slotted ALOHA protocol (*e.g.*, when a user in the 16-user group wants to send, it waits for an odd slot and then transmits with a probability p).

(c) Determine the *average* throughput (in packets/slot) of the system, assuming that every user always has something to send and, in each group, the users use the optimal transmission probability.

Problem 3 – Token Ring and Round-Robin

Consider the system in Fig. 3, where we have 8 communicating nodes positioned in a ring architecture. Here we have a *token ring system*, where Node 0 starts with the token and hence is allowed to communicate first. Then, after a node finishes its transmission, the node successively passes the token to the next one following the counterclockwise direction. Table 1 reports the time each node takes to execute its transmission.

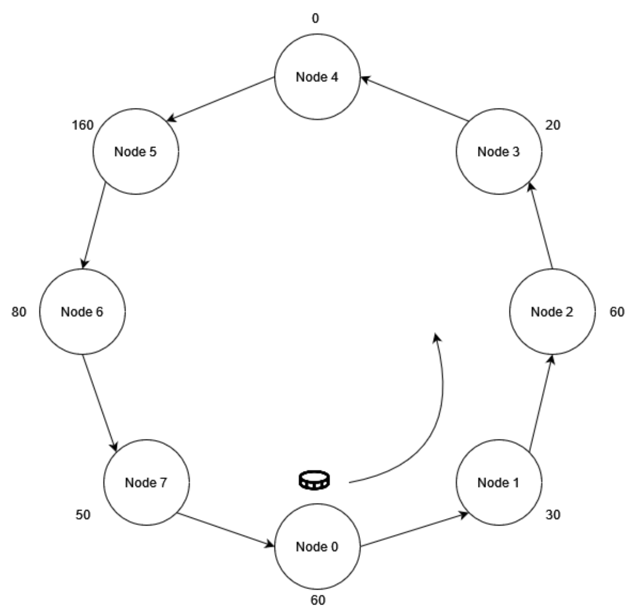

Figure 3: Architecture of the token ring system.

Time is set to 0 when the original token holder (Node 0) starts its first transmission. Considering that the token handover is instantaneous, answer the following:

Node ID	0	1	2	3	4	5	6	7
Time to execute (ms)	60	30	60	20	0	160	80	50

Table 1: Execution times.

(a) How long does it take before Node 6 has sent its message?

(b) What are the consequences of having a system where we do not limit communication time? How does this affect the performance of the various nodes?

To make sharing of the communication medium fairer we add a *round-robin* approach to the system. Now our nodes can only transmit for 50 ms at a time before they have to pass the token on to the next node.

(c) How long time before Node 6 has finalized transmitting its message?

(d) Which nodes have had the time before they finish their communication increased, which has it decreased, and which are unchanged?

(e) Nodes in this architecture would normally only know the existence of their neighbors and nothing more. From the perspective of Node 0, can you make any assumptions on the minimum or the maximum number of nodes, from when Node 0 gets the token back?

- Is it the same if we do not apply the round-robin approach?
- If the original token holder changed, would the same amount of information on the number of nodes in the system be available, and which nodes would know more?