Alle Opgaver Løst igen ark :)

Lecture 5

Problem 1 – The Price of a Protocol

Consider that Alice wants to transmit 5,000 bytes of data to Bob. The communication protocol between Alice and Bob consists of the following elements: an error detection mechanism based on CRC, a feedback mechanism based on ARQ with sequence number, and a two-way connection with piggybacking in order to enable transmission/feedback from/to either Alice or Bob. The accordingly frame structure is shown in Fig. 1.

(a) What is the minimum number of frames required to be transmitted by Alice in order to convey the 5,000 bytes of data to Bob?

Solution:

```
number_frames = ceil(5000/64)% frames :)
```

```
number_frames =
    79
```

(b) Consider that the communication channel that connects Alice and Bob supports a data rate of 1 kbit/s. What would be the minimum time required for Alice's data to reach Bob if the ARQ procedure was not present and communication was error-free?

Solution:

```
totalbits = (24+(64*8))*number_frames; % Hele framen bruges.
min_time = totalbits/1e3; % 1e3 = 1kbits
```

(c) What is the efficiency of the protocol adopted? Then, what is the goodput in bit/s following the considerations made in (b)?

Solution:

```
effeciency = (64*8)/((64*8)+24) %mængde data tæller, totalmængde bits i nævner
```

```
effeciency =
         0.955223880597015
```

```
goodput = 1e3*effeciency %bits/s
```

```
goodput =
          955.223880597015
```

(d) Considering a very specific scenario where only Alice transmits and Bob receives with a fixed data packet length of 64 bytes. How would you change the frame to make the protocol more efficient? What is the efficiency gain and the new goodput when comparing the new protocol to the old one?

Solution:

Fjern pkt length nu hvor size er fixed.

```
new_effeciency = 512/530; % 530 = uden packet length
new_goodput = new_effeciency*1e3;
```

```
new_goodput =
          966.037735849057
```

Problem 2 – Feedback Latency

Consider the same scenario introduced in Problem 1 and the frame depicted in Fig. 1. Now, assume that a frame sent by Alice takes from 5 to 20 ms to be received by Bob, which is also valid from Bob to Alice. On average, this delay in receiving is 10 ms. Recall that Alice wants to transmit 5,000 bytes of data to Bob.

(a) What is the minimum time required by Alice to know that a frame has been lost? And the maximum?

Solution: 5ms+5ms (min) 20ms + 20ms(max) �

(b) Assuming that all frames are received successfully, what is the average throughput in bit/s of Alice's transmission by considering now the ACKs from Bob?

Solution: $\frac{536\text{bits} \cdot 79 \text{ frames}}{20\text{ms} \cdot 79 \text{ frames}} = 26800 \frac{\text{bits}}{s}$

```
average_throughput = (536*79)/(20e-3*79);
```

Consider now that Alice transmits 10,000 frames to Bob and assume that every 100-th frame is lost in transmission. Consider only that frames from Alice can be in error, while ACKs/NACKs are perfectly sent from Bob.

(c) How many retransmissions would occur?

```
retransmission = (10e3/100) + 1 % Husk at retransmission som man regner fra
10e3/100 også fejler derfor + 1 :)
```

```
retransmission =
   101
```

(d) How long would it take to transmit all frames on average? Assume that all retrans-mission is successful

Solution:

$$\frac{10100 \text{ frames} \cdot 536 \text{ bits}}{26.8\text{E}3 \text{ averagethroughput}} = 202\text{sekunder}$$

```
average_transmit = (10100*536)/26.8e3;
```

Alternative answer:

```
(10e3+100)*20e-3; %Total frame gange average delaye for receive
```

```
ans =
   202
```

Problem 3 – RS-232

For this problem, assume that Alice and Bob are connected via a serial link and their communication follows the recommended standard 232 (RS-232).

(a) Draw a diagram showing the evolution over time of the voltage levels within an RS-232 frame to transmit the lowercase ASCII character "h" from Alice to Bob.[1]

Solution: FUCKING TEGN :)

(b) Consider a baud rate of 9,600 bit/s. How long would it take for to Alice send the sentence "hello world" to Bob assuming error-free communication? Assume that each RS-232 frame is comprised of a start bit and a stop bit and that there are no parity bits.

```
x = (11*10)/(9600); % i sekunder for totalt mængde af karaktere(11) som fylder 10
bit
num2sip(x);
```

```
ans =
'11.458 m'
```

(c) Assume the use of the parity bit now and consider the even parity bit scheme[2]. The transmitter wants to send the following ASCII character: `0 1 1 0 0 0 0 1`. Which character is it? What is the parity check bit that needs to be appended to this data? Consider the following reception situations for the data sent:

(i) `0 1 1 1 0 0 0 1`. Which character was received? Can the error be detected?

(ii) `0 1 1 1 0 1 0 1`. Which character was received? Can the error be detected?

Solution: ACII = 'a' og paritiy bit = 1 grundet even parity hvilket medfører at man tæller antal '1' og da dette er ulige = 1 og modsat for hvis det var lige. For ulige parity så ville ulige antal 1 give 0 og omvendt.

(i) q and yes.

(ii) u and no, fordi det er lige parity og der er lige antal 1 så skulle det sidste bit være 0?

Problem 4 – ARQ with Limited Retransmissions

In Example 3 of Lecture 5, we show the average delay under the assumption of infinite retransmissions. This assumption means that, if we have an unbounded time to receive data, then the transmission success probability over the *Binary Symmetric Channel* (BSC) is always 1. These are the theoretical insights obtained by our analysis. However, infinite retransmission attempts might not be practical, considering the communication system's stringent latency requirement. So, in this problem, we investigate the performance when we set the constraint for the number of retransmissions. Let K be the maximum number of retransmissions for a single packet. We consider a scenario where Alice wants to deliver M packets to Bob in a BSC where the *Packet Error Rate* (PER) is $p_{\mathrm{PER}} = 0.1$.

Solution:

$1 - (p_{\mathrm{PER}})^{\text{Total transmissions}(K+M)} = \text{proability of succesful transmission}$

(a) For $K = 1$, $P_s = \sum_{i=1}^{2} p_{\mathrm{PER}}^{(i-1)}(1 - p_{\mathrm{PER}}) = 1 - p_{\mathrm{PER}}^2 = 1 - 0.1^2 = 0.99$

For $K = 2$, $P_s = \sum_{i=1}^{3} p_{\mathrm{PER}}^{(i-1)}(1 - p_{\mathrm{PER}}) = 1 - p_{\mathrm{PER}}^3 = 1 - 0.1^3 = 0.999$

(b) Calculate the throughput for K = 1, and then for K = 2

$$\text{Number of packet}(M) \cdot \frac{\text{proability of succes transmissions}}{\text{number of transmissiomns}(M+K)} = 1 \cdot \frac{0.99}{2} = 0.495 \text{ packet per slot}$$

$$1 \cdot \frac{0.999}{3} = 0.333 \text{ Packet per slot}$$

(c) Characterize the relationship between the throughput, the probability of successful packet transmission, and the value of K.

Solution: Increasing the number of retransmissions leads to a higher probability of successful packet transmission for a single packet, while it deteriorates the throughput. **TLDR**: Betal ved kasse 1 ����

(d) Let us consider the case where $M = 2$, meaning that Alice wants to send 2 packets to Bob. In this case, we are interested in the probability of the 2 data packets being transmitted by a deadline of 5 slots. This probability is known as the *reliability* of the communication system. Does the number of retransmissions that Alice can send for each packet affect the reliability? Explain it.

Solution:

If the set number of retransmissions makes the total number of transmissions for the 2 packets higher than 5 slots, then the reliability is zero.

Fx: M=2 og K=1 så hvis hver message fejler og en retransmission sker så det totalt 4 hvilket er under 5 slots hvilket er godt. Mere end det så er reliability zero.

Notice that the reliability also depends on Bob successfully receiving the data packets. Hence, if the total number of transmissions for the 2 packets is below 5 slots, the communication reliability is defined by the packet error rate.

Lecture 6

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.

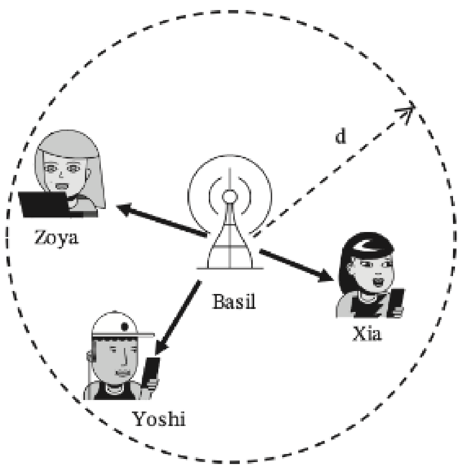

Figure 1: A cellular network that is operating according to a TDMA scheme.

(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?

Solution: $\log 2(4)$ Hvor 4 er de forskellige different 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?

Solution:

- Det kan gå galt hvis de alle gerne vil snak med Basil på samme tid og der sker ingen ting.
- ACK can be lost in transmission for both parties and nothing happens then try again to fix it :)

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.

Solution:

Based on the initial access frame, a link termination frame would be comprised of a reference signal sent by Basil to establish that this frame has begun. Then, Basil sends the address of the user that she wants to be terminated. The user listens to it and withdraws its connection. The user no longer has access to the network.

��

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?

Solution: 1kbit /s bliver delt i mellem 3 som skal høre det hele

```
1e3/3;
```

```
ans =
        333.333333333333
```

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.

Solution: Jeg har lagt det på github og sæt billede ind.

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.

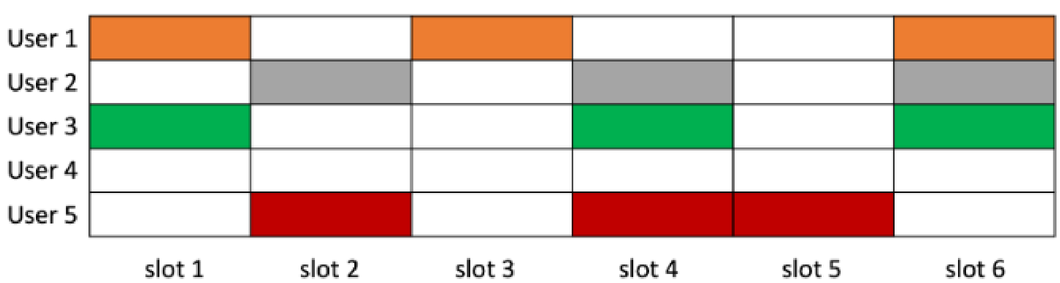

Figure 2: Example of slotted ALOHA. Each user sends multiple copies of the same packet.

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

Solution:

User 1, Slot 3 and User 5 Slot 5

(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?

Solution: U have to pick either slot 3 or slot 5 since there are only one package which means the sequence will either start with user 1 or user 5.

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.*, $0, 2, 4, \dots$) are reserved for the 4-user group. Odd time slots (*i.e.*, $1, 3, 5, \dots$) 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.

Solution:

$\frac{\left(1-\frac{1}{4}\right)^{4-1}+\left(1-\frac{1}{16}\right)^{(16-1)}}{2}=0.4$ Brug formlen som er i givet i opgave beskrivelsen.

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.

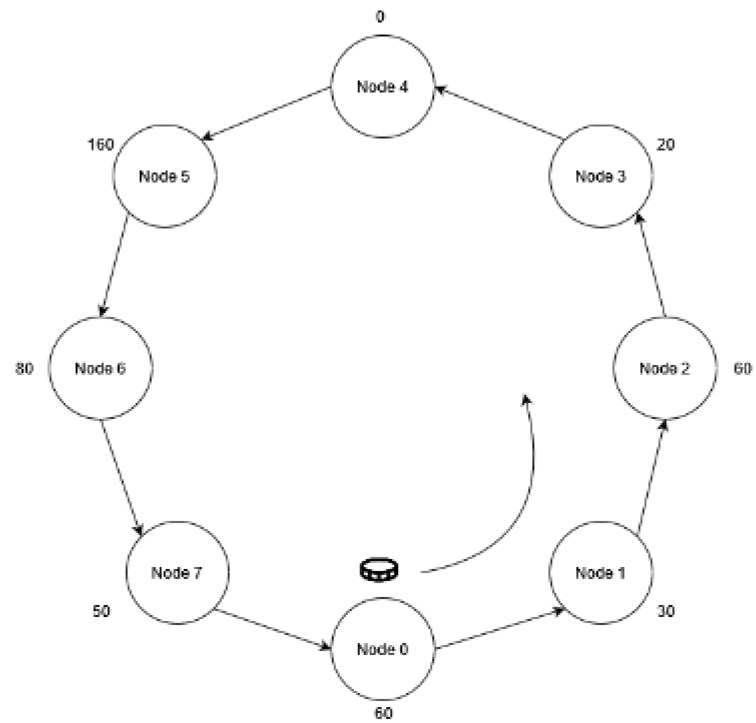

Figure 3: Architecture of the token ring system.

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?

Solution:

$60 + 30 + 60 + 20 + 0 + 160 + 80 = 410$

(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?

Solution: Det betyder at en node kan bruge lang tid på at sende så rimelig unfair :)

Ingen garanti på latens for at kommunikere.

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?

Solution: 50+30+50+20+50+50+50+10+10+50+30=400 (Så man skal to gange rund eftersom at 50 er maksimal det der kan sendes så der er en rest for nogen)

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

- Increased: Node; 2,0,5
- Decreased: Node; 1,3,6,7
- Unchanged: Node; 4

(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?