



ESD5 – Fall 2024

Problem Set 1

Department of Electronic Systems
Aalborg University

September 16, 2024

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.

- 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?
- 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?

In computer network terminology, **throughput** refers to the rate of successful messages (frames) transmitted over a communication channel during a certain time period. For example, in (b), the throughput would be equal to the data rate of the channel of 1 kbit/s, since the communication is error-free. However, a message is comprised of bits that are useless to Alice, but still useful for the protocol; these are called control bits. Moreover, the

preamble	pkt length	own seq number	other seq number	CRC error detection	data
1 byte	6 bits	1 bit	1 bit	1 byte	1-64 bytes

Figure 1: Protocol frame adopted by Alice and Bob.

$$A) \frac{5000}{64} = 78,12 = 79 \text{ frames}$$

$$B) \text{ The frame: } 64 \text{ byte} = 8 \cdot 64 = 512 \text{ bits}$$

$$\frac{512}{1000} = 0,512 \text{ [s]} \text{ for a single frame}$$

$$0,512 \cdot 79 = 40,4 \text{ [s]} \text{ for the 79 frames.}$$

bits that are useful for Alice are called data bits. Hence, another useful metric to measure the efficiency of the communication is the so-called **goodput**, which is the rate of successful useful information transmitted over a communication channel during a certain time period. Thus, the goodput excludes the protocol overhead, giving us a better sense of communication efficiency with respect to the complexity of the adopted protocol.

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

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?
- (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?

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

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

¹For reference, look at the ASCII table available on <https://www.asciitable.com/>. Hint: Note the difference between the ASCII of lowercase and uppercase letters.

$$c) \quad e_p = \frac{64}{67} \approx 0.96$$

$$e_g = 1000 \cdot 0.96 \approx 960 \text{ bit/s}$$

d) Leaving only error detection
leaves a size of 65 byte

$$\frac{64}{65} \approx 0.98, \quad 1000 \cdot 0.98 = 980 \left[\frac{\text{bit}}{\text{s}} \right]$$

a)

$$\min = 5 \text{ ms} \cdot 2 = 10 \text{ ms}$$

$$\max = 20 \text{ ms} \cdot 2 = 40 \text{ ms}$$

b)

$$10 \text{ ms} \cdot 2 = 20 \text{ ms per frame}$$

$$20 \text{ ms} \cdot 74 = 1580 \text{ ms} = 1.58 \text{ s}$$

- (b) Consider a baud rate of 9,600 bit/s. How long would it take for Alice to 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.
- (c) Assume the use of the parity bit now and consider the even parity bit scheme². The transmitter wants to send the following ASCII character: 0 1 1 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 1. Which character was received? Can the error be detected?
 - (ii) 0 1 1 1 0 1 1. Which character was received? Can the error be detected?

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 $p_{PER} = 0.1$.

- (a) Let us focus on the case where $M = 1$. Calculate the probability of successful packet transmission for $K = 1$, and then for $K = 2$.
- (b) Calculate the throughput for $K = 1$, and then for $K = 2$.
- (c) Characterize the relationship between the throughput, the probability of successful packet transmission, and the value of K .
- (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.

²Please, read more about parity bit in https://en.wikipedia.org/wiki/Parity_bit.

1.A
 $\frac{5000}{64} = 78,125$

1.B
 $78.67 \cdot 8 = 42344 \text{ [bits]}$
 $\frac{42344}{1000} = 42,4 \text{ [S]}$

1.C
 $e = \frac{64}{67} = 95,5\%$
 $\text{Good} = \frac{64}{67} \cdot 1000 = 955 \text{ [bit/s]}$

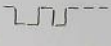
1.D
 Remove
 · Pkt length
 · own Seq
 · other Seq
 $e = \frac{64}{66} = 96,9\%$
 $\text{Good} = 969 \text{ [bit/s]}$

2.A
 $\text{Min} = 9\text{ms} + 5\text{ms} = 14\text{ms}$
 $\text{Max} = 20\text{ms} + 20\text{ms} = 40\text{ms}$

2.B
 $79 \cdot 20\text{ms} + 42,4\text{S} = 43,98 \text{ [S]}$
 $79 \cdot 20\text{ms} = 1,58 \text{ [S]}$
 $\frac{42344}{43,98} = 962 \text{ [bit/s]}$

2.C
 $\text{Retrans} = \frac{10000}{100} = 100$
 $\text{Retrans} = \frac{100}{100} = 1$
 $\text{errors} = 100 + 1 = 101$

2.D
 $\text{tot bits} = 10100 \cdot 67 \cdot 8 = 5413600$
 $\frac{5413600}{962} = 5628 \text{ [S]}$

3a
 $h = 0.68 = 104 \text{ DBC}$
 $\text{bit} = \frac{1011011009}{8}$

 High = logic 0 (4.5, 15)
 Low = logic 1 (1.5, 15)

3b
 1 Start bit + 8 bit Data + 1 Stop bit = 10 bit
 "hello_world" 11 characters = 10-11 = 110 bit
 Transmission time $\rightarrow \frac{110 \text{ [bit]}}{9600 \text{ [bps]}} = 12 \text{ ms}$

3c
 We want to get [01100001] = "a" \rightarrow Parity bit = 1
 We receive \rightarrow [0110000] = "a" (Error detected, Since even nb. of bits)
 We receive \rightarrow [01100010] = "a" (Error not detected)

4a
 $M=1$ $K=1$ $P_{\text{PER}}=0.1$
 $P_{\text{PER}} = 0.1^2 = 0.01 \rightarrow 1\%$ Success $\rightarrow 99\%$
 $K=2$ $0.1^3 = 0.001 \rightarrow 0.1\%$ Success $\rightarrow 99.9\%$

4b
 $R_{5232} \rightarrow 9600 \frac{\text{bit}}{\text{s}}$ $K=1$ 10% $\frac{9600}{1.1} \rightarrow 8727 \text{ bits}$ $K=2$ $\frac{9600}{1.11} \rightarrow 8648 \text{ bits}$

4c
 Throughput = $\frac{9600}{1 + P_{\text{PER}} K}$

4d
 $1 - 0.1^4 = 0.9999 \rightarrow 0.9999\%$ YES



Lecture 5
solutions

ESD5 – Fall 2024 Problem Set 1 – Solutions

Department of Electronic Systems
Aalborg University

September 16, 2024

Problem 1 – The Price of a Protocol

(a)

$$\text{min. number of frames} = \left\lceil \frac{5000}{64} \right\rceil = 79.$$

Alternative answer: since we have the flexibility of deciding the packet length, we can assume that we could have transmitted the following fractional number of frames:

$$\frac{5000}{64} = 78.125 \text{ frames.}$$

(b) A frame consists of 536 bits (64 byte \times 8 [bit/byte]). To transmit a frame with a data rate of 1 kbit/s we need 0.536 s. Then, we need 42.344 s to transmit the 79 frames.

Alternative answer: Following (a), it would take 41.875 ms to transmit the 78.125 frames.

(c) Each frame has 512 bits of the payload information. The efficiency is: $\frac{512}{536} \approx 0.96$. The goodput is then $1 \cdot 10^3 \cdot 0.96 = 960 \text{ bit/s}$.

(d) By excluding the packet length field the sequence numbers (used for piggybacking) and CRC, we get a new frame whose size is 520 bits. The new efficiency is $\frac{512}{520} \approx 0.98$; goodput of 980 bit/s. Consequently, we have an efficiency gain of approximately 2%.

Note: You could naturally remove more or less which would yield yet again a different efficiency. Fundamentally the point is you should only add in as much control data as you need to operate correctly, otherwise there will be a waste. Of course, if operations are not correct you need control to ensure a desirable behavior.

Problem 2 – Feedback Latency

- (a) The minimum is 10 ms and the maximum is 40 ms. Note that we are defining the data rate of the channel when giving the time that the message arrives at the receiver. Hence, we are substituting the value of 1 kbit/s used in the first exercise.
- (b) The average throughput is:

$$\text{avg. throughput} = \frac{79 \cdot 536 \text{ bits}}{79 \cdot (10 + 10) \cdot 10^{-3}} = 26.8 \text{ kbit/s.}$$

The numerator is the number of bits that Alice wants to transmit, where one should recall that the number of frames to transmit 5,000 bytes is 79 frames From Problem 1 – (a). The numerator corresponds to the time needed to Alice transmit a frame and receive an ACK from Bob, which is considered to be always successful.

- (c) 100 re-transmissions if we assume that the re-transmitted frames are received successfully.

Alternative and more complete answer: By thinking recursively, from the 100 re-transmissions, we will have one more in error. Hence, we would have a total of 101 re-transmissions.

- (d) The transmission time can be computed as follows:

$$\text{Tx time} = \left(\underbrace{2 \cdot 10,000 \cdot 10 \cdot 10^{-3}}_{\text{time to Alice transmit and Bob ACK/NACK}} + \underbrace{2 \cdot 100 \cdot 10 \cdot 10^{-3}}_{\text{time to Alice re-transmit and Bob ACK}} \right) = 202 \text{ s.}$$

Problem 3 – RS-232

- (a) The diagram is shown below. Note that we did not consider the parity bit. If we consider it, it should be placed between “b7” and the “stop” bits. *The calculation of the parity bit does not consider the start bit, just the data bits.* **Assumptions.** Voltage levels: “0” \leftarrow -3 V and “1” \leftarrow +3 V. We also specify that the binary word is written from the Most Significant Bit (MSB) to the Least Significant Bit (LSB). We assume that before the transmission the link is in the “idle” state (0 V). According to the standard the payload data of an RS-232 frame consists of 8 bits.

With that, the transmitter should transmit: ‘0011010001’, where the blue bit is the start one while the red one is the stop. Thus, we obtain the following diagram:

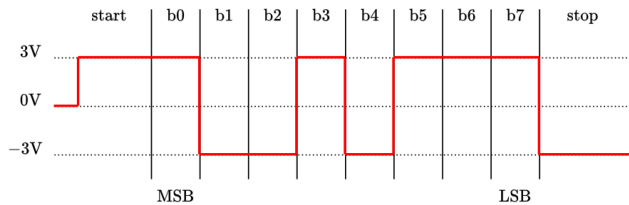


Figure 1: Diagram for (a). The signal is in red.

Alternative answer: one could also have considered the inverse form for the assignments of the voltage levels and the direction in which the payload data is interpreted. Moreover, one could also have assumed the parity bit¹.

- (b) It would take $\frac{11 \cdot 10}{9600} = 0.011$ s.
- (c) The character is “a” and the parity would be “1”.
- (i) “q”; yes.
 - (ii) “u”; no.

¹See on <https://en.wikipedia.org/wiki/RS-232> that the voltage levels can range from ± 3 V to ± 15 V and that the direction in which the payload data is read is not readily specified by the standard.

Problem 4 – ARQ with Limited Retransmissions

- (a) For $K = 1$, $P_s = \sum_{i=1}^{\infty} p_{\text{PER}}^{(i-1)} (1 - p_{\text{PER}}) = 1 - p_{\text{PER}}^2 = 1 - 0.1^2 = 0.99$
For $K = 2$, $P_s = \sum_{i=1}^{\infty} p_{\text{PER}}^{(i-1)} (1 - p_{\text{PER}}) = 1 - p_{\text{PER}}^3 = 1 - 0.1^3 = 0.999$
- (b) For $K = 1$, $G = \frac{1 \times 0.99}{1 + 1} = 0.495$ packet/slot
For $K = 2$, $G = \frac{1 \times 0.999}{1 + 2} = 0.333$ packet/slot
- (c) Increasing the number of retransmissions leads to a higher probability of successful packet transmission for a single packet, while it deteriorates the throughput.
- (d) 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. 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.



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.

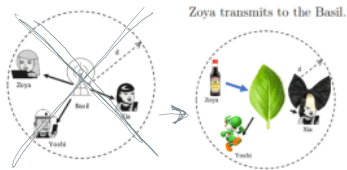


Figure 1: A cellular network that is operating according to a TDMA scheme.

1

- (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?

2

2a
 $4 = 2^d$
 $d = 2$
so 60 2 bits

1b
Der er free for all indtil at alle forbundet op.
Så hvis 2 sendes på samme tidspunkt, så vil Basil IKKE sende en ACK tilbage af så ved de to der sendte ved at de skal vente et random stykke tid, og så sende igen indtil at Basil sendes sin ACK tilbage.

2c
Så man bestemmer at være X'ne time slot (periodisk check in)
Og hvis der ikke kommer et svar tlabge så smider man dem af net værket og så kan man pinge ud og høre om der er andre der gerne vil komme på linjen.

2d
 $\frac{1}{3}$
Hun har en slot ud af tre.
Hun skal vente tre gange.

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

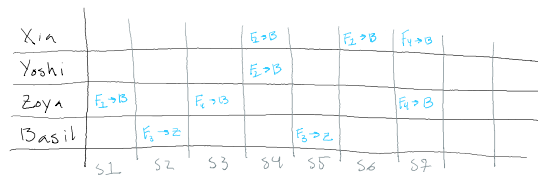
According to Fig. 2, consider a slotted ALOHA scheme involving five users competing to send their packets over 6 time slots. We assume that users are sending multiple copies of the same packet.



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? *User 1 in slot 3, User 5 in 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?

3



Så hvad vi lige læser ud fra det så:

MAN KAN IKKE DECODE NOGET HVIS MAN IKKE HAR FÅET DET RENT FØR.

Så, hvis vi KENDTE user 1, inden det her timetable, så du kunne man succesfuldt få user 3 ud af det og så ville man have decoded 3, på slot 6.

NEVER MIND!

MAN GEMMER ALT ENS DATA, SÅ MAN KAN DECODE TILBAGE.

Slot 1: user 3

Slot 2: user 2

Slot 3: user 1

Slot 5: user 5

Da, vi på slot 3 for user 1

Og på slot 5 for 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, ...) are reserved for the 4-user group. Odd time slots (i.e., 1, 3, 5, ...) 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.

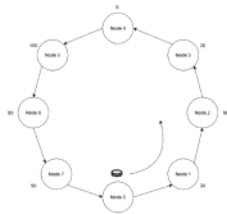


Figure 3: Architecture of the token ring system.

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:

4

$$\begin{aligned}
 & \text{for 4 } \left(1 - \frac{1}{4}\right)^3 = 0.42 \quad (50\%) \\
 & \text{for 16 } \left(1 - \frac{1}{16}\right)^{15} = 0.38 \quad (50\%) \\
 & \frac{0.42 + 0.38}{2} = 0.40
 \end{aligned}$$

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?

5

$$3a) 60 + 30 + 60 + 20 + 0 + 160 + 80$$

$$3c) 50 + 30 + 50 + 20 + 0 + 50 + 50 + 10 + 0 + 10 + 0 + 0 + 50 + 30 + 50$$

So in second loop i

$$3d) \text{ Increase: } 0, 2, 5 \\ \text{Decrease: } 1, 3, 4, 6, 7$$

3e) With Round Robin:

$$\frac{\text{Return time}}{50 \text{ [ms]}} = \frac{\text{min number of users}}{\text{max} = \text{NoIPBA}}$$

Without Round Robin
 NO Fucking CHUB

ESD5 – Fall 2024 Problem Set 2 – Solutions

Department of Electronic Systems
Aalborg University
September 19, 2024

Problem 1 – TDMA-Based Scheduling

- (a) $\log_2 4 = 2$ bits
 (b) The following events can happen:
- No user responds to the invitation sent by Basil. Basil does nothing.
 - More than one user answers the invitation from Basil. The users will not receive an ACK from Basil or they can receive something with the wrong address. The users need to try again in the future randomly in order to reduce the chance of collision (tossing a coin).
 - ACK sent by Basil may be lost. Then, the user needs to try again in the future when another initial access frame is sent by Basil.
 - Please, feel free to exploit your creativity here and propose other events.
- (c) 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.
- (d) The equivalent data rate is $R_{\text{SLA}} = \frac{R}{K}$, where K is the number of users, which is assumed to be the size of the downlink frame. Therefore, $R_{\text{SLA}} = \frac{1}{3}$ kbit/s.
- (e) The diagram can be seen in Fig. 1.

1

Assume that:

H_{00} - initial access frame

H_{01} - link reconnection frame

H_{10} - DL transmission

H_{11} - UL transmission

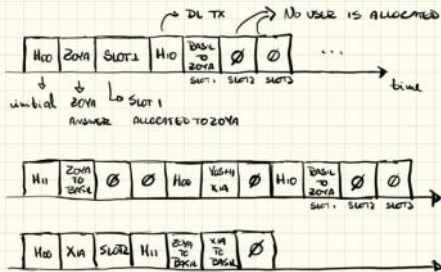


Figure 1: Solution (c).

2

Problem 2 – ALOHA Protocol

(a) 2 users: User 1 and User 5

(b) Applying the SIC method, 4 users' packets can be successfully decoded. The decoded order could be one of the following possible solutions:

- User 1 → User 5 → User 3 → User 2
- User 1 → User 5 → User 2 → User 3
- User 5 → User 1 → User 3 → User 2
- User 5 → User 1 → User 2 → User 3

(c) The 4-user group and even time slots establish a slotted ALOHA system with 4 nodes. Then the maximum throughput could be achieved when $p = \frac{1}{4}$, resulting in throughput at $(1 - \frac{1}{4})^4 = \frac{27}{64}$.

The 16-user group and even time slots establish a slotted ALOHA system with 16 nodes. Then the maximum throughput could be achieved when $p = \frac{1}{16}$, resulting in throughput at $(1 - \frac{1}{16})^{16} = \frac{360}{65536}$.

Summing up, the average throughput is $\frac{1}{2}(\frac{27}{64} + \frac{360}{65536}) \approx 0.4$.

3

Problem 3 – Token Ring and Round-Robin

- (a) $60 + 30 + 60 + 20 + 0 + 160 + 80 = 410$ ms
- (b) Long communication time from certain nodes, like Node 5, and long delay for the following nodes even if they have little to communicate. This causes uncertainty in the availability of the channel for others, and removes any guarantees that could be made on latency from wanting to communicate to actually doing so. If nodes produce more to transmit than the channel supports, we would also have single nodes using it indefinitely.
- (c) $50 + 30 + 50 + 20 + 0 + 50 + 50 + 10 + 0 + 10 + 0 + 0 + 50 + 30 = 400$ ms
- (d) Increased: 0, 2, 5
Decreased: 1, 3, 6, 7
Unchanged: 4 (cannot be increased or reduced, as nothing was communicated)
- (e) If we think of time when doing a full circle of the ring, Node 0 may communicate again after 300 ms. If we consider the maximum time we may use the medium (50 ms), Node 0 knows at least 6 nodes are in the system since $\frac{300}{50} = 6$.
- No, single nodes like Node 5 can take long making it look like many nodes, while nodes like 4 are hidden away, and would not be noticed. The missing guarantee on communication time makes it impossible.
 - The nodes know 2 things: how long before they get the token back again and how much they need to communicate. If the token were to start at Node 1, 3, or 4 they would know that there are at least 7 nodes in the system. Exemplifying, for Node 1, $\text{num_nodes} = 1 + \lceil \frac{300-30}{50} \rceil = 7$, while, for Node 0, it would be $\text{num_nodes} = 1 + \lceil \frac{300-50}{50} \rceil = 6$.



ESD5 – Fall 2024 Problem Set 4

Department of Electronic Systems
Aalborg University
September 30, 2024

Problem 1 – Bellman-Ford Algorithm

The *Bellman-Ford algorithm* is an algorithm for finding and computing the shortest paths in a graph.¹ In this exercise, we are going to see an example of how this algorithm can be used when considering a weighted graph. But first, why are we interested in graphs and algorithms over them? A weighted graph can be used as a mathematical structure to represent a communication network, where nodes are devices such as computers and edges are links (cables). The weights of each edge can correspond to the communication cost/efficiency of the corresponding link. Therefore, to improve a practical network's performance, we can use graph algorithms to perform routing or other functionalities we would like our network to perform.

Consider the graph and weights illustrated in Fig. 1. Consider that the weight of each edge corresponds to the distance between the nodes. We want to find the shortest distance paths between the source node S and all the other nodes. To do this, we will apply the Bellman-Ford algorithm to find the shortest network paths. Answer the following:

- What is the maximum number of iterations required to complete the Bellman algorithm?
- Describe the Bellman algorithm through each iteration and construct the shortest path tree.

Challenge: You could also try to run the algorithm when changing the weight between A-D to -4. However, we should not interpret the weights as distances anymore because it does not make physical sense to have negative distances. In this case, instead of finding the

¹https://en.wikipedia.org/wiki/Bellman%E2%80%93Ford_algorithm

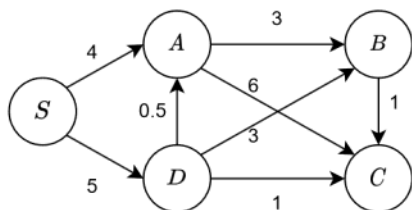


Figure 1: Weighted graph with weights representing distance.

shortest path, let's assume the **objective** is that we would like to find the path with the lowest weight. We also assume that the end path cannot have a negative weight!

Note: We could also have set the **objective** to find the path with the largest weight! Everything depends on what we want in a given context.

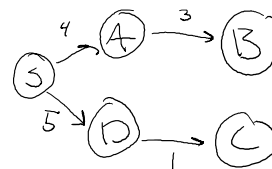
Problem 2 – Routing

The purpose of *routing* in a network is to find the best path to transmit a packet from a node X to another node Y . In Problem 1, we have learned how graphs can represent networks and how to use the Bellman-Ford algorithm. In this part, we will apply what we have learned to perform routing!

The *distance-vector routing protocol* is a routing protocol based on finding the shortest distances. Consider the network of Fig. 2 and that distance-vector routing protocol is used. To be familiarized with the nomenclature used by the protocol, please read https://en.wikipedia.org/wiki/Distance-vector_routing_protocol. Now, consider that a message is sent through the network in the figure and nodes B, D, and E have the following routing tables: B: (5, 0, 8, 12, 6, 2); D: (16, 12, 6, 0, 9, 10); E: (7, 6, 3, 9, 0, 4). Note that each routing table entry (A, B, C, D, E, F) corresponds to the *distance between the current node and all other nodes*. For example, B: (5, 0, 8, 12, 6, 2) means that the distance to transmit from B to A is 5, B to B is 0, B to C is 8, and so on. Consider that the distance between the C to B, D, and E links are 6, 3, and 5, respectively. Based on these three routing tables, *what is C's new routing table? Give the routing table and the corresponding best nodes.*

Hint: For example, to obtain the value for the first element of C's routing table, you have to determine the path that will cost the less from C to A based on the possible ways to reach A from C: through B, D, and E.

S	A	D	B	C
0	4	5	2	∞
	5,5		7	6
			8	10
				8
0	4	5	7	6



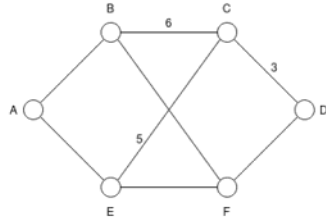


Figure 2: Example of a network.

Problem 3 – Flow control

A CPU executes instructions at the rate of 1000 MIPS (million instructions per second). A transmitter wants to send a packet of 64 bits to a receiver that uses this CPU. Consider that the processing of each packet costs 10 instructions. Consider that the CPU must copy the packet 4 times, where the copy operation over a packet can occur simultaneously. Can the receiver's CPU handle it if the transmitter and receiver are connected using a link to transmit data with 1 Gbps? For simplicity, assume that all instructions, even those that read or write memory, run at the total 1000-MIPS rate.

Motivation: This exercise represents an example of flow control, where the data rate of the link between the transmitter and the receiver should adapt to the processing capability of the receiver! There is no point in transmitting faster than the receiver can handle it.

Problem 4 – TCP/IP

This exercise suggests that you read and familiarize yourself with the TCP/IP, the protocol that enabled the internet boom.² Without this protocol, you would not be able to have Netflix, YouTube, and social media.

- To address the limitations of IP version 4, a major effort had to be undertaken via IETF that resulted in the design of IP version 6. There is still a significant reluctance to adopt this new version. However, no such major effort is needed to address the limitations of TCP. Explain why this is the case.
- In the figure of the TCP segment header as in Fig. 3, we saw that in addition to the 32-bit acknowledgment field, there is an ACK bit in the fourth word. Does this add

²For a quick intro, you can find YouTube videos, such as https://www.youtube.com/watch?v=PpsEaqJV_A0&ab_channel=Techquickie.

	B	D	E	C
A	5	6	7	10
B	0	12	6	8
C	8	6	3	0
D	12	0	9	6
E	6	9	0	3
F	2	10	4	7

Info:

64 bits = 10 instructions

no copy

64 bits = 10 · 4 = 40 inst.

bits per inst $\frac{64}{40} = 1.6 \frac{\text{bit}}{\text{inst}}$

as 1000 MIPS

$\frac{10^9}{1.6} = 625 \cdot 10^6$

som er $< 10^9$ og derfor kan den følge med

anything? Why or why not?

		TCP segment header																																			
Offsets	Octet	0								1								2								3											
Octet	Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0				
0	0	Source port																Destination port																			
4	32	Sequence number																																			
8	64	Acknowledgment number (if ACK set)																																			
12	96	Data offset	Reserved		0 0 0		N	S	C	W	R	E	U	R	G	A	C	K	P	S	S	R	Y	N	F	I	N	Window Size									
16	128	Checksum																Urgent pointer (if URG set)																			
20	160	Options (if data offset > 5. Padded at the end with "0" bits if necessary.)																																			
...	...																																				
60	480																																				

Figure 3: TCP segment header.



ESD5 – Fall 2024
Problem Set 4 – Solutions

Department of Electronic Systems
Aalborg University

September 30, 2024

Problem 1 – Bellman Ford Algorithm

- (a) Max. number of iterations required = num. of nodes - 1 = 4.
(b) The Bellman algorithm (one of the possible solutions)

Initial

Node	S	A	B	C	D
Cost	0	∞	∞	∞	∞
Pre-Node	-	-	-	-	-

Iteration 1

Node	S	A	B	C	D
Cost	0	4	∞	∞	5
Pre-Node	-	S	-	-	S

Iteration 2

Node	S	A	B	C	D
Cost	0	4	7	6	5
Pre-Node	-	S	A	D	S

Converged here: we did not need the 4 iterations at all.

Iteration 3

Node	S	A	B	C	D
Cost	0	4	7	6	5
Pre-Node	-	S	A	D	S

Iteration 4

Node	S	A	B	C	D
Cost	0	4	7	6	5
Pre-Node	-	S	A	D	S

Then, the shortest path tree: S-D-A-B-C

Path	Shortest Path	Distance
S-A	S-A	4
S-B	S-A-B	7
S-C	S-D-C	6
S-D	S-D	5

Challenge:

Initial

Node	S	A	B	C	D
Cost	0	∞	∞	∞	∞
Pre-Node	-	-	-	-	-

Iteration 1

Node	S	A	B	C	D
Cost	0	4	∞	∞	5
Pre-Node	-	S	-	-	S

Iteration 2

Node	S	A	B	C	D
Cost	0	1	7	6	5
Pre-Node	-	D	A	D	S

Iteration 3

Node	S	A	B	C	D
Cost	0	1	4	5	5
Pre-Node	-	D	A	B	S

Iteration 4

Node	S	A	B	C	D
Cost	0	1	4	5	5
Pre-Node	-	D	A	D	S

Then the shortest path tree: S-D-A-B-C

Path	Shortest Path	Distance
S-A	S-D-A	1
S-B	S-D-A-B	4
S-C	S-D-A-B-C	5
S-D	S-D	5

(c) Checking negative weight cycle

Path	Condition	Checking
A-B	$d_B \leq d_A + c(A,B)$	$4 \leq 1 + 3$
A-C	$d_C \leq d_A + c(A,C)$	$5 \leq 1 + 6$
B-C	$d_C \leq d_B + c(B,C)$	$5 \leq 4 + 1$
D-B	$d_B \leq d_D + c(D,B)$	$4 \leq 5 + 3$
D-C	$d_C \leq d_D + c(D,C)$	$5 \leq 5 + 1$
D-A	$d_A \leq d_D + c(D,A)$	$1 \leq 5 + (-4)$
S-D	$d_D \leq d_S + c(S,D)$	$5 \leq 0 + 5$
S-A	$d_A \leq d_S + c(S,A)$	$1 \leq 0 + 4$

→ All conditions are satisfied, and the graph has no negative cycle.

Problem 2 – Routing

(a,b)

Going from C via B to all the other nodes gives (11, 6, 14, 18, 12, 8).

Going from C via D to all the other nodes gives (19, 15, 9, 3, 9, 10).

Going from C via E to all the other nodes gives (12, 11, 8, 14, 5, 9).

Taking the minimum for each destination except C gives C's new routing table as

Destination	A	B	C	D	E	F
Distance	11	6	0	3	5	8
Outgoing line	B	B	-	D	E	B

Problem 3 – Flow control

Each packet consumes 10 instructions. With 4 copies, we end up with 40 instructions. The CPU can process 40 instructions in 40 nanoseconds. Thus, a byte (8 bits) requires 5 nanoseconds of CPU time. Thus, the system can handle 200 megabytes/second, 1600 megabits/second, or 1.6 gigabits/second. Therefore, the answer is yes: the system can handle a transmission line of 1 gigabit/second.

Problem 4 – TCP-IP

- (a) IP is a network-level protocol, while TCP is an end-to-end transport-level protocol. Any change in the protocol specification of IP must be incorporated on all routers on the Internet. On the other hand, TCP can work fine as long as the two endpoints are running compatible versions. Thus, it is possible to have many different versions of TCP running at the same time on different hosts, but not this is not the case with IP.
- (b) The ACK bit is used to tell whether the 32-bit field is used. But if it were not there, the 32-bit field would always have to be used, if necessary, acknowledging a byte that had already been acknowledged. In short, it is not essential for normal data traffic. However, it plays a crucial role during connection establishment, where it is used in the second and third messages of the three-way handshake.

EcX9, incl sol9

Monday, 7 October 2024 19.30



Lecture 9
exercises

ESD5 – Fall 2024

Problem Set 5

Department of Electronic Systems
Aalborg University

October 7, 2024

Problem 1

In encryption using block-cipher, what potential problem can occur when using Electronic Code Book? Using Cipher Block Chaining?

Problem 2

Assume that two parties know each other's public keys. If one message is sent from A to B, what can be verified? If two messages are exchanged, what can be verified?

Problem 3

With Diffie-Hellman key exchange, is it a man-in-the-middle attack possible? If possible, draw how this could be achieved.

Problem 4

Assume the following scenario: A and B both have the knowledge of a secret key K (e.g. pre-shared). They communicate over an insecure channel. Define a protocol (by writing down a message sequence chart) in which A and B use the pre-shared key K to mutually authenticate each other and to agree on a common session key (different from the long-term pre-shared key K). Try to keep the number of exchanged messages as low as possible.

Problem 5

Asymmetric encryption and symmetric encryption are both actively used to ensure security. While Asymmetric encryption leads to more security it comes with computational expensive algorithms for encryption and decryption. Meanwhile, symmetric keys provide less security but are less computationally expensive. Assuming A has a public/private key pair, where B can ask what the public key is through an unsecured channel, how could a middle ground between the computational cost and security be met?

- (a) To further enhance security enhancing the authentication is critical. What kind of technology could A possess, and how would this make a Man-in-the-middle attack more difficult?



ESD5 – Fall 2024
Problem Set 5 – [Solutions](#)

Department of Electronic Systems
Aalborg University

October 7, 2024

Problem 1

- Too long messages can cause repetition in the cipher which is vulnerable to cryptanalysis.
- Losing a block means we cannot decrypt the following blocks.

Problem 2

- If one is sent, the authenticity of A can be verified by B, and the integrity of the message, confidentiality as only B can decrypt it. At this point, A can however not verify the authenticity of B, but is sure that no one but B can read or alter the message.
- If both are sent the authenticity of both users can be verified by both. Of course, the properties of confidentiality and integrity remain and are verifiable by both.

Problem 3

- Yes, if the MitM can establish 2 sessions with each user, i.e. create an asymmetric key session with both users then a MitM attack is viable, Fig. 1 shows an example by drawing.

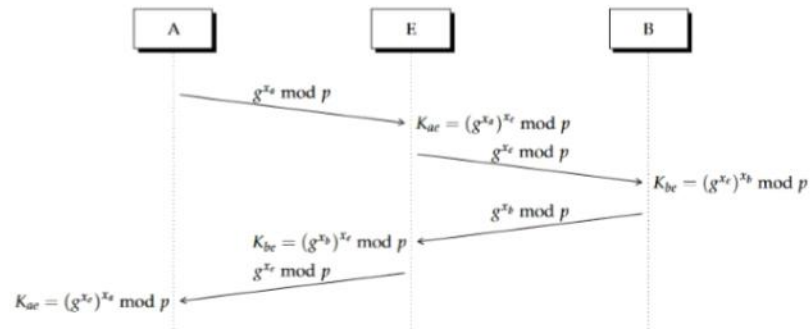


Figure 1: Example of man in the middle attack with Diffie-Hellman. From the paper <https://www.mdpi.com/1099-4300/23/2/226>.

Problem 4

- This is just a proposal to a solution, as long as the authentication is ensured and a new session key can be generated then the principles are correct.

- **Step 1.** A and B send starts a connection.

Step 2. Authentication via a "challenge"; A sends a random string, and B hashes it with the key and sends it back to A. A can validate B by hashing the challenge and the key together and comparing the hash with what was received. A key point is that the challenge should be random, otherwise, it may be vulnerable to a replay attack. While the initial communication could be encrypted with a key, it is not necessary to authenticate as the one way function property of a hash makes reversing the operation computationally infeasible.

Step 3. When authenticated, any key generation algorithm could be used e.g. RSA/DH to create the session key.

Problem 5

We use asymmetric encryption to create a secure channel, through which B can define a symmetric key that both can use for a session. By redefining the session key “often” the security may be further enhanced, of course at the cost of more computational complexity.

- (a) By adding certificates to the mix, B can be sure that A is who A says he is. This is a critical part of e.g. HTTPS, where websites authenticate their identity through a trusted CA, hence establishing a server-client connection is asymmetricly encrypted. The user can then assume that the website is who they say and that the public key is not false, and data transmissions be done through a symmetrically encrypted channel.