CSIE 5452, Fall 2018 — Homework 1

Due October 1 (Monday) at Noon

Points may be deducted for solutions that do not show intermediate work.

1 Data Efficiency (12pts)

In the lecture, we introduced the bit stuffing of the Controller Area Network (CAN) and calculate the concept of worst-case data efficiency of the CAN. Assume that there are 4 Electronic Control Units (ECUs), $\varepsilon_0, \varepsilon_1, \varepsilon_2, \varepsilon_3$, and 4 messages, $\mu_0, \mu_1, \mu_2, \mu_3$, as follows:

Message	Sender	Receiver(s)	Number of Bits	Period (msec)
μ_0	ε_0	ε_1	6	50
μ_1	ε_0	ε_1	10	50
μ_2	ε_1	$\varepsilon_2, \varepsilon_3$	10	50
μ_3	ε_0	ε_3	16	100

1. (4pts) What is the best-case data efficiency for μ_0?
2. (4pts) What is the worst-case data efficiency for μ_0?
3. (4pts) What is the worst-case data efficiency for μ_1?

2 Message Redesign (20pts)

1. (4pts) Following Question 1, a system designer redesigns the messages as follows:

Message	Sender	Receiver(s)	Number of Bits	Period (msec)
μ_0'	ε_0	ε_1	16	50
μ_2	ε_1	$\varepsilon_2, \varepsilon_3$	10	50
μ_3	ε_0	ε_3	16	100

   where the first 6 bits of μ_0' are the bits from μ_0 and the following 10 bits of μ_0' are the bits from μ_1. What is the worst-case data efficiency for μ_0'?
2. (4pts) Regarding the number of bits which need to be transmitted in the worst case, do you think that the new design is better? Please explain.
3. (4pts) Can you further merge μ_2 into μ_0'?
4. (8pts) In most cases, it does not hurt to have more frequent messages, but it is not allowed to have less frequent messages. Following this policy, can you further improve the number of bits which need to be transmitted in the worst case? Please explain.

3 Worst-Case Response Time Computation — Part I (20pts)

Given a set of periodic messages μ_0, μ_1, μ_2 with their priorities, transmission times, and periods as follows:

Message	Priority (P_i)	Transmission Time (C_i) (msec)	Period (T_i) (msec)
μ_0	0	10	50
μ_1	1	40	100
μ_2	2	10	200

The worst-case response time R_i of μ_i can be computed as

$$R_i = Q_i + C_i, \tag{1}$$

and

$$Q_i = B_i + \sum_{\forall j, P_j < P_i} \left\lceil \frac{Q_i + \tau}{T_j} \right\rceil C_j, \tag{2}$$

where $\tau = 0.1$ in this question. You can consider using the following tables to help you.

1. (4pts) What is the worst-case response time of μ_0?

Iteration	LHS (Q_0)	B_0	RHS	Stop?
1				

2. (8pts) What is the worst-case response time of μ_1?

Iteration	LHS (Q_1)	B_1	j	$Q_1 + \tau$	T_j	$\left\lceil \frac{Q_1+\tau}{T_j} \right\rceil$	C_j	RHS	Stop?
1			0						
2			0						
3			0						

3. (8pts) What is the worst-case response time of μ_2?

Iteration	LHS (Q_2)	B_2	j	$Q_2 + \tau$	T_j	$\left\lceil \frac{Q_2+\tau}{T_j} \right\rceil$	C_j	RHS	Stop?
1			0						
			1						
2			0						
			1						
3			0						
			1						

4 Worst-Case Response Time Computation — Part II (32pts)

Please download the benchmark "input1.dat" from the "Resources" area of Piazza. In the benchmark, the first number is n, the number of messages. The second number is τ. Each of the following lines contains the priority (P_i), the transmission time (C_i), and the period (T_i) of each message. You are required to do two things in your submission:

1. You should print out n numbers (one number per line) representing the worst-case response time (R_i) of those messages. Note that you need to follow the message ordering in the benchmark, *e.g.*, the first number in the list is the worst-case response time of the first message in the benchmark.
2. You should also print out your source codes. (For your information, my implementation is less than 100 lines.) We may ask you to provide your source codes which must be the same as those on your printout. If the worst-case response times above are correct but the source codes are clearly wrong implementation, it is regarded as academic dishonesty.

It is highly recommended to write your codes well (*e.g.*, capable of dynamically allocating memory based on n) so that you can reuse them in Homework 2 or Homework 3. Ideally, you can test your implementation with the small benchmark in Question 3 and verify its solution by your implementation. Just do not make the same mistake in Questions 3 and 4.

5 Timing Analysis of Another Protocol (16pts)

There is a communication protocol based on Time Division Multiple Access (TDMA). The protocol maintains a periodic schedule which consists of 8 time slots. Each time slot is 1-second long, and thus the period of the schedule is 8 seconds. Each time slot is assigned to at most one message, and each message can only be transmitted in the time slots assigned to the message.

There is a periodic message μ within the communication protocol. The transmission time of μ is 1 second, and the period of μ is 4 second. There are two "consecutive" time slots assigned to μ in every period of the schedule. Here, we allow the worst-case response time of a message to exceed the period of the message, but the instances of a message should be transmitted in order (*e.g.*, the first instance should be transmitted before the second instance).

You can answer this question by observation and explanation. Rigorous mathematical proofs are not necessary.

1. (8pts) If the releases of message μ and the schedule are aligned as follows:
   - The instances of message μ are release at time 0, 4, 8, ...
   - The time slots assigned to μ are at time 1–3, 9–11, 17–19, ...

   What is the worst-case response time of μ?
2. (8pts) If the releases of message μ and the schedule are not aligned, *i.e.*,
   - The instances of message μ are release at time $0 + s$, $4 + s$, $8 + s$, $\ldots$, where $0 \leq s < 4$.
   - The time slots assigned to μ are at time 1–3, 9–11, 17–19, ...

   What is the worst-case response time of μ?