CSIE 5452, Fall 2022: Homework 1

1 Timing Analysis of the CAN Protocol: Part I (12pts)

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

Iteration	LHS (Q_0)	B_0	RHS	Stop?
1	30	30	30	Yes

A: worse-case response time of μ_0 : 30+10 = 40ms

1-2. (4pts) What is the worst-case response time of μ1?

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	30	30	0	30.1	50	1	10	40	No
2	40	30	0	40.1	50	1	10	40	Yes

A: worst-case response time of μ1 : 40+30 = 70ms

1-3. (4pts) What is the worst-case response time of μ2?

<table>
<tr><th>Iteration</th><th>LHS (Q_1)</th><th>B_1</th><th>j</th><th>$Q_1 + \tau$</th><th>T_j</th><th>$\left\lceil \frac{Q_1+\tau}{T_j} \right\rceil$</th><th>$C_j$</th><th>RHS</th><th>Stop?</th></tr>
<tr><td rowspan="2">1</td><td rowspan="2">20</td><td rowspan="2">20</td><td>0</td><td rowspan="2">20.1</td><td>50</td><td>1</td><td>10</td><td rowspan="2">60</td><td rowspan="2">No</td></tr>
<tr><td>1</td><td>200</td><td>1</td><td>30</td></tr>
<tr><td rowspan="2">2</td><td rowspan="2">60</td><td rowspan="2">20</td><td>0</td><td rowspan="2">60.1</td><td>50</td><td>2</td><td>10</td><td rowspan="2">70</td><td rowspan="2">No</td></tr>
<tr><td>1</td><td>200</td><td>1</td><td>30</td></tr>
<tr><td rowspan="2">3</td><td rowspan="2">70</td><td rowspan="2">20</td><td>0</td><td rowspan="2">70.1</td><td>50</td><td>2</td><td>10</td><td rowspan="2">70</td><td rowspan="2">Yes</td></tr>
<tr><td>1</td><td>200</td><td>1</td><td>30</td></tr>
</table>

A: worst-case response time of μ1 : 70+20 = 90ms

2 Timing Analysis of the CAN Protocol: Part II (36pts)

2-1. Worst-case response time (Ri) of those messages.

1.440 ms
2.040 ms
2.560 ms
3.160 ms
3.680 ms
4.280 ms
5.200 ms
8.400 ms
9.000 ms
9.680 ms
10.200 ms
19.360 ms
19.800 ms
20.320 ms
29.400 ms
29.760 ms
30.280 ms

2-2. Source code

```
import numpy as np

import math


with open('input.dat', 'r') as f:

    d = f.readlines()

    n, tau = int(d[0]), float(d[1]) # d[0] is the number of messages.

    cis = [] # list for storing all transmission time (Ci)

    tis = [] # list for storing all period (Ti)


    # extract priority (Pi), the transmission time (Ci), and the period (Ti)

    of each message starting from col3

    for i in range(2, len(d)):

        mu = [float(x) for x in d[i].split()]

        cis.append(mu[1])

        tis.append(mu[2])
```

```python
schedulable = True
worst_response = []
for i in range(n):
    if i < n-1:
        qi = rhs = bi = max(cis[i:]) # blocking time for μi
    else:
        qi = rhs = bi = cis[i]

    first = True
    while first or (rhs + cis[i] <= tis[i] and qi != rhs):
        qi = rhs
        rhs = bi
        first = False
        for j in range(i):
            rhs += (math.ceil((qi+tau)/tis[j])*cis[j])

    if rhs + cis[i] > tis[i]:
        print('constraint violation')
        schedulable = False
        break
    elif qi == rhs:
        print('the system is schedulable')
        worst_response.append(qi+cis[i])

if schedulable:
    print("\n Worst-Case Response Times : ")
    for i in range(n):
        print('{:.3f} ms'.format(worst_response[i]))
```

3 Timing Analysis of Preemptive Fixed-Priority Scheduling

(16pts)

3-1. (4pts) What is the worst-case response time of τ0?

Iteration	LHS (R_0)	C_0	RHS	Stop?
1	10	10	10	Yes

A: worst-case response time of τ0 : 10ms

3-2. (4pts)) What is the worst-case response time of τ1?

Iteration	LHS (R_1)	C_1	j	R_1	T_j	$\left\lceil \frac{R_1}{T_j} \right\rceil$	C_j	RHS	Stop?
1	30	30	0	30	50	1	10	40	No
2	40	30	0	40	50	1	10	40	Yes

A: worst-case response time of τ1 : 40ms

3-3. (4pts) What is the worst-case response time of τ2?

Iteration	LHS (R_1)	C_1	j	R_1	T_j	$\left\lceil \frac{R_1}{T_j} \right\rceil$	C_j	RHS	Stop?
1	20	20	0	20	50	1	10	60	No
			1		200	1	30		
2	60	20	0	60	50	2	10	70	No
			1		200	1	30		
3	70	20	0	70	50	2	10	70	Yes
			1		200	1	30		

A: worst-case response time of τ2 : 70ms

3-4. (4pts) Compared with non-preemptive fixed-priority scheduling, preemptive fixed-priority scheduling is expected to be disadvantageous to the lowest-priority message/task. Explain why the worst-case response time of $\tau 2$ is smaller than the worst-case response time of $\mu 2$ in Question 1.

A: Because we do not know if $\mu 2$ is schedulable, we need to include itself as blocking time into the worst-case response time. And since we assume $\mu 2$ has arrived just slightly earlier then $\mu 0$ and $\mu 1$, those messages with higher priority should be served before $\mu 2$ being served again, which is similar to the situation we discuss about $\tau 2$. However, since $\tau 2$ is preemptive, even though $\mu 2$ is earlier than those who are prior to $\mu 2$, it will be preempted immediately. Hence, $\tau 2$ does not need to consider itself as blocking time and which make it's worst-case response time smaller than that of $\mu 2$.

4. Timing Analysis of TDMA-Based Protocols (12pts)

4-1. (2pts) Please duplicate the schedule pattern (hint: (4, 10, 1, 2, . . .)). No intermediate work is needed here.

A: (4, 10, 1, 2, 6, 7)

4-2. (2pts) Please duplicate the arriving times of frames in the frame arrival pattern but fix m = 4 and p = 10. No intermediate work is needed here.

A: (4, 10, 0, 3, 5, 6, 10, 13, 15, 16)

4-3. (2pts) Please duplicate the starting times of time slots in the schedule pattern but fix n = 4 and q = 10. No intermediate work is needed here.

A: (4, 10, 1, 2, 6, 7, 11, 12, 16, 17)

4-4. (4pts) Please complete the following table:

k	$\max_{1\le j\le n}(s_{j+k} - s_j)$	=	$\min_{1\le i\le m}(a_{i+k-1} - a_i)$	=	(Col-3)−(Col-5)
1	$\max_{1\le j\le 4}(s_{j+1} - s_j)$	4	$\min_{1\le i\le 4}(a_i - a_i)$	0	4
2	$\max_{1\le j\le 4}(s_{j+2} - s_j)$	5	$\min_{1\le i\le 4}(a_{i+1} - a_i)$	1	4
3	$\max_{1\le j\le 4}(s_{j+3} - s_j)$	9	$\min_{1\le i\le 4}(a_{i+2} - a_i)$	3	6
4	$\max_{1\le j\le 4}(s_{j+4} - s_j)$	10	$\min_{1\le i\le 4}(a_{i+3} - a_i)$	6	4

4-5 (2pts) Please compute the worst-case response time (which is waiting time plus transmission time) of the message.

A: 1 + 6 = 7

5 MILP Linearization (12pts)

5-1 (4pts) Given α, β, γ which are binary variables, prove

$\alpha + \beta + \gamma \neq 2 \Longleftrightarrow \alpha + \beta - \gamma \leq 1 \wedge \alpha - \beta + \gamma \leq 1 \wedge -\alpha + \beta + \gamma \leq 1$

by filling "T" (True) or "F" (False) in the following table (if LHS=RHS in all cases, then LHS and RHS are equivalent):

a	b	γ	LHS	$\alpha + \beta - \gamma \leq 1$	$\alpha - \beta + \gamma \leq 1$	$\alpha + \beta + \gamma \leq 1$	RHS	LHS = RHS
0	0	0	T	T	T	T	T	T
0	0	1	T	T	T	T	T	T
0	1	0	T	T	T	T	T	T
0	1	1	F	T	T	F	F	T
1	0	0	T	T	T	T	T	T
1	0	1	F	T	F	T	F	T
1	1	0	F	F	T	T	F	T
1	1	1	T	T	T	T	T	T

5-2 (4pts) Given α, β, γ which are binary variables, prove

$\alpha\beta = \gamma \Longleftrightarrow \alpha + \beta - 1 \leq \gamma \wedge \gamma \leq \alpha \wedge \gamma \leq \beta$

by filling "T" (True) or "F" (False) in the following table (if LHS=RHS in all cases, then LHS and RHS are equivalent):

a	b	γ	LHS	$\alpha + \beta - 1 \leq \gamma$	$\gamma \leq \alpha$	$\gamma \leq \beta$	RHS	LHS = RHS
0	0	0	T	T	T	T	T	T
0	0	1	F	T	F	F	F	T
0	1	0	T	T	T	T	T	T
0	1	1	F	T	F	T	F	T
1	0	0	T	T	T	T	T	T
1	0	1	F	T	T	F	F	T
1	1	0	F	F	T	T	F	T
1	1	1	T	T	T	T	T	T

5-3 (4pts) Given β which is a binary variable, x, y which are non-negative real variables, and a constraint $x \leq 2022$, select a value of M to guarantee $\beta x = y \Longleftrightarrow 0 \leq y \leq x \wedge x - M(1 - \beta) \leq y \wedge y \leq M\beta$, where you can refer to the following table:

β	LHS	0≤y≤x	x−M(1−β)≤y	y≤Mβ	RHS
0	0=y	0≤y≤x	x−M≤y	y≤0	x−M≤y=0≤x
1	x=y	0≤y≤x	x≤y	y≤M	0≤y=x≤M

A: To select a value for M, we can discuss with 4 situations.

(1) LHS is True and RHS is True where $\beta = 0$
Since $y = 0$, and x is non-negative real variables, $y \leq x$ will hold true. And M need to be at least 2022 to let $x - M \leq y$ hold true, since x could be any real number from 0 to 2022.

Conclusion: $M \geq 2022$

(2) LHS is False RHS is False where $\beta = 0$
Because $y \neq 0$ and y need to be greater than 0, y is a positive real number, which make $y \leq 0$ false at the same time. And this already make RHS false. Hence, M could be any number in this situation.

Conclusion: M can be any number

(3) LHS is True and RHS is True where $\beta = 1$
Here we have $x = y$, which clearly make $0 \leq y \leq x$ and $x \leq y$ true, so we only need to consider $y \leq M$. And since x could be at most 2022, and so does y, M should at least 2022 to make RHS hold true.

Conclusion: $M \geq 2022$

(4) LHS is False and RHS is False where $\beta = 1$
Since $x \neq y$, either $0 \leq y \leq x$ or $x \leq y$ must hold false, which make RHS false. Consequently, it does not matter if $y \leq M$ hold true or false.

Conclusion: M can be any number

As a result, $M \geq 2022$ is the constraint to select M.

6 Signal Packing (12pts)

6-1 (4pts) Regarding the number of bits which need to be transmitted, do you think that the new design is better? Please explain.

A: Yes, fewer bits need to be transmitted. Before the new design µ0 and µ1 need to send (44+3+8) + (44+4+16) = 118bits every 50ms. However, after the new design, µ'0 only need to transmit (44+3+6+10) = 63nits every 50ms.

6-2 (4pts) Can you further merge µ2 into µ'0?

A: No, neither the sender nor the receivers of µ'0 and µ2 are the same.

6-3 (4pts) 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? Please explain.

A: Yes, if we merge µ'0 with µ3 and send this new message, hereinafter call µ4, every 50ms. Reason as follows. Since µ'0 and µ3 are sent by the same sender, we merge them together and send it to ε1 and ε3 every 50ms, which made it (44+3+16+16) = 79bits every 50ms. Because the question state that it does not hurt to have more frequent messages, here we assume that ε3 is allowed to ignore µ4 every 50ms since the period of µ3 is 100ms. In original, µ'0 need to send (44+3+16) = 63bits every 50ms and µ3 need to send (44+3+16) = 63bits every 100ms. That is, 63*2 + 63 = 189bits in 100ms in total. However, we only need to send 79*2 = 158bits in 100ms with the new method. Hence, it can be further improve.