Tutorial 3

Real Time Scheduling -2 (CAN)

Example on CAN Busload

1. Consider a CAN bus with baud rate 125 kbps transmits the following messages.

Message	Data (bytes)	Rate (Period in ms)
Msg A	1	100 trx/s (10ms)
Msg B	5	20 trx/s (50ms)
Msg C	8	10 trx/s (100ms)

Calculate the % bus load considering the worst case traffic?

Example on CAN Busload

Soln. We know

Bandwidth consumption (%) = (1.1 total periodic bits consumed / total bits available) * 100*

Message	Data (bytes)	Message Size (bits)	Rate (Period in ms)	Message Bits Consumed
Msg A	1	1*8+47 = 55	100 trx/s (10ms)	55*100= 5500
Msg B	5	5*8 +47 = 87	20 trx/s (50ms)	87*20 = 1740
Msg C	8	8*8+47 = 111	10 trx/s (100ms)	111*10 = 1110
Total periodic bits consumed				8350

Bandwidth consumption (%) = (1.1 8350 / 125000) * 100 =* ***7.348 %***

Exercise on CAN Busload

2. CAN message traffic description is provided in the below Table.The cybersecurity engineer wants to do feasibility study on implementing the encryption of each CAN messages. Each CAN pay-load along with 256-bits of Hash-based Message Authentication Code (HMAC) together is encrypted with 128-bit AES blocks. Assuming the CAN baudrate to be 500kbps find the bus load *before encryption* and *after encryption*.

Message (ID)	Data (bytes)	Period (ms)
M1 (1)	8	20
M2 (2)	5	10
M3 (3)	3	30
M4 (4)	4	60
M5 (5)	1	60

Exercise on CAN Busload

Soln. What Happens During Encryption ?

If M1 is encrypted using 256 bit HMAC then total bit size = 8*8+256 = 320

To encrypt with 128 bit AES block cypher this needs to be divided into 128-bit size blocks.

Hence total number of such blocks = ceiling(320/128) = 3

(i.e. 128bits+128bits+64 zeros padded with 64bits)

And total number of bits become 3*128 i.e. 384 bits or (384/64) or **6 CAN frames**.

Hence **one** unencrypted CAN frame becomes **six CAN frames.**

Note that this is true for any message bigger than 0 bytes size. Do you agree?

Now solve the question.

Exercise on CAN Busload

Message (ID)	Data (bytes)	Period (ms)	Rate (trx/s)	Bps without encryption	**Bps with encryption**
M1 (1)	8	20	50	(8*8+47)*50 = 5550	**(111*6)*50 = 33300**
M2 (2)	5	10	100	(5*8+47)*10 = 8700	**(111*6)*10 = 6660**
M3 (3)	3	30	33.33	(3*8+47)*33.33 = 2366.43	**(111*6)*33.33 = 22198**
M4 (4)	4	60	16.66	(4*8+47)*16.66 = 1316.14	**(111*6)*16.66 = 11096**
M5 (5)	1	60	16.66	(1*8+47)*1666 = 916.3	**(111*6)*50 = 11096**
Total Number of Bits				18848.87	**144268.92**
Bus load				4.146%	**31.73%**

Exercise on TDMA

3. Consider there are 3 tasks (T1,T2,T3) in PU A and 2 tasks(T4,T5) in PU B. T3 being a control task transmits a message m1 via TDMA bus. T_{Tx}(m1) is the transmission task running in the PU A for that purpose. T4 is another control task that that receives m1 to serve its control logic and transmits another message m2. T_{Rx}(m1) is the receiving task running in the PU B for this purpose. Following are the details of the tasks.

PU A			
Tasks	Offset (s)	Period (s)	Execution Time (s)
T1	0	2	0.5
T2	0	3	0.5
T3	0	4	1
T_{Tx}(m1)	3	4	0.5

Exercise on TDMA

3. Contd..

   a. Are the tasks in PU A EDF Schedulable? If so, Derive the schedule.
   b. Suggest a suitable TDMA cycle and slot length such that the m1 is transmitted at a fixed slot in a cycle.

PU A			
Tasks	Offset (s)	Period (s)	Execution Time (s)
T1	0	2	0.5
T2	0	3	0.5
T3	0	4	1
T_{Tx}(m1)	3	4	0.5

Exercise on TDMA

3. a. Are the tasks in PU A EDF Schedulable? If so, Derive the schedule.

   0.5/2+0.5/3+¼+0.5/4 < 1 ===> **satisfies the necessary condition**

   Hence, it should be EDF schedulable

PU A			
Tasks	Offset (ms)	Period (ms)	Execution Time (ms)
T1	0	2	0.5
T2	0	3	0.5
T3	0	4	1
T_{Tx}(m1)	3	4	0.5

Exercise on TDMA

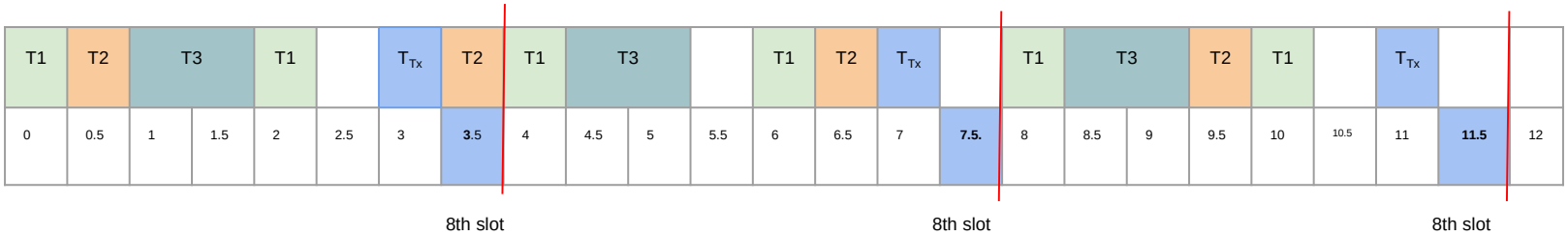
3. a. T1,1->T2->T3->T1->_->T_{Tx} ->T2->T1->T3->_-> T1-> T2->T_{Tx}->_->T1->T3->T2->T1->_->T_{Tx}->_

T1	T2	T3		T1		T_{Tx}	T2	T1	T3			T1	T2	T_{Tx}		T1	T3		T2	T1		T_{Tx}		
0	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10	10.5	11	11.5	12

PU A			
Tasks	Offset (ms)	Period (ms)	Execution Time (ms)
T1	0	2	0.5
T2	0	3	0.5
T3	0	4	1
T_{Tx}(m1)	3	4	0.5

Exercise on TDMA

3. b. Suggest a suitable TDMA cycle and slot length such that the m1 is transmitted at a fixed slot in a cycle.



We need to choose a cycle in a way such that transmissions happen *after 3.5*, *after 7.5* and *after 11.5.*

m1 is ready to dispatch in the bus at 3.5, 7.5 and 11.5 ms

Considering a cycle length of 4ms with 0.5ms slots makes m1 transmit at every 8th slot in a cycle.

Note that we also could have chosen a cycle of length 2ms with m1 transmitting every alternative cycles but always at the last/4th slot.

Exercise on TDMA

4. Consider there are 3 tasks (T1,T2,T3) in PU A and 2 tasks(T4,T5) in PU B. T3 being a control task transmits a message m1 via communication bus. T_{Tx}(m1) is the transmission task running in the PU A for that purpose. T4 is another control task that that receives m1 to serve its control logic and transmits another can message m2. T_{Rx}(m1) is the receiving task running in the PU B for this purpose. Following are the details of the tasks.

PU B			
Tasks	Offset (s)	Period (s)	Execution Time (s)
T4	x	8	1
T5	0	2	1
T_{Rx}(m1)	y	8	0.5

Exercise on TDMA

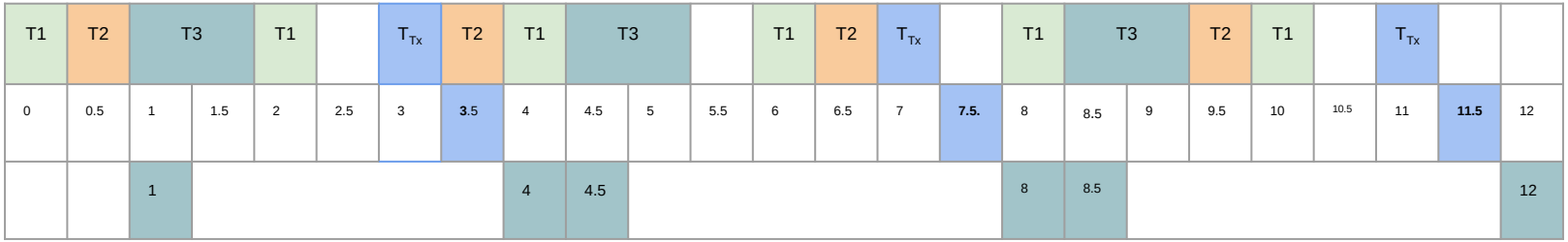
4. Contd..

a. Suggest a minimum offset for the receiving task (y).
b. Suggest a minimum offset value for T4 (x)
c. Find the EDF schedule to find the minimum end to end delay for the control calculation of T4.

PU B			
Tasks	Offset (s)	Period (s)	Execution Time (s)
T4	x	8	1
T5	0	2	1
T_{Rx}(m1)	y	8	0.5

Exercise on TDMA

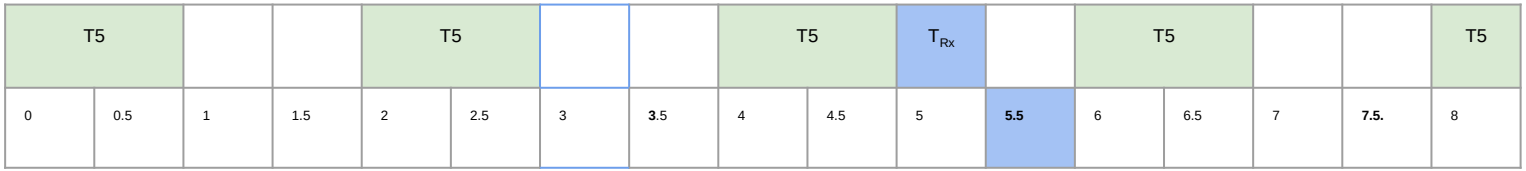
4. a. As we can see, m1 is done transmitting at 4ms after T3 arrives.



Hence the receiving task must at least have an offset of 4ms (y=4).

Exercise on TDMA

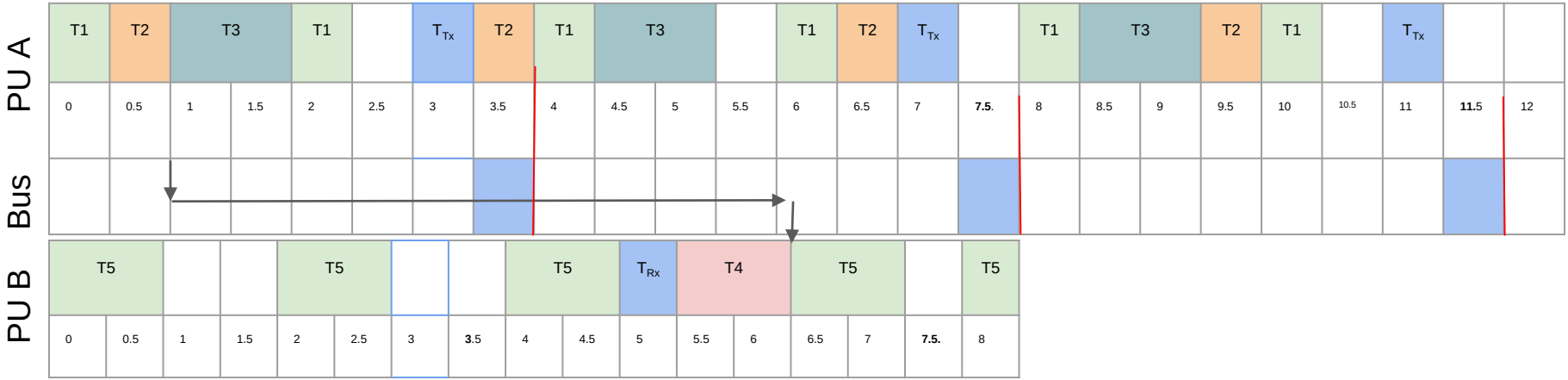
4. b. As we can see, T4 can only start after m1 is received



Hence its minimum offset can be 5.5ms (x=5.5ms)

Exercise on TDMA

4. c. The possible EDF schedule in PU B : T5->_->T5->T5->T_{Rx}->T4->T5->_->T5



Observing the EDF schedule in PU A the minimum total delay to execute T4 is (6.5 -1) = 5.5 ms

Bits in a CAN Frame

Field	Length (bits)	Description
Start of Frame (SOF)	1	Must be dominant
Identifier	11	Unique identifier indicates priority
Remote Transmission Request (RTR)	1	Dominant in data frames; recessive in remote frames
Reserved	2	Must be dominant
Data Length Code (DLC)	4	Number of data bytes (0–8)
Data Field	0–8 bytes	Length determined by DLC field
Cyclic Redundancy Check (CRC)	15	
CRC Delimiter	1	Must be recessive
Acknowledge (ACK)	1	Transmitter sends recessive; receiver asserts dominant
ACK Delimiter	1	Must be recessive
End of Frame (EOF)	7	Must be recessive