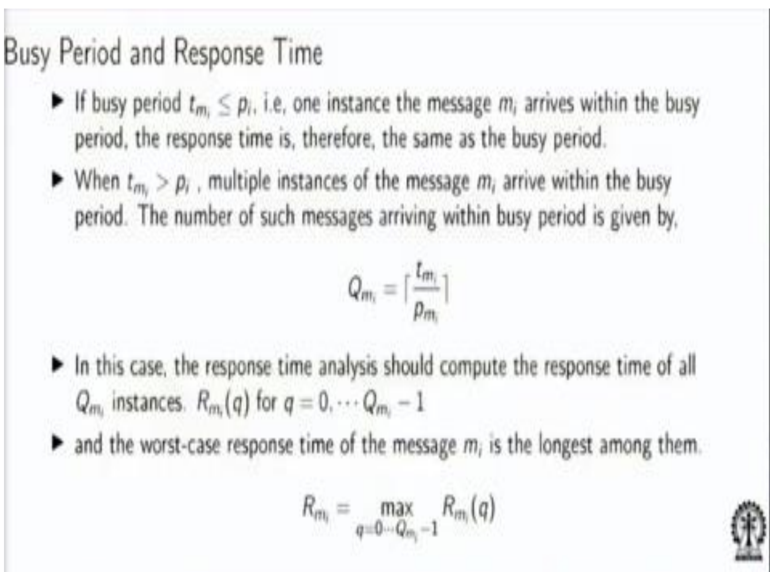
Foundation of Cyber Physical Systems

Prof. Soumyajit Dey
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 17
Real Time Task Scheduling for CPS (Continued)

Hi and welcome back to this lecture series on Foundations of Cyber Physical Systems. So, I believe in the last lecture, we have been talking about this CAN bus worst-case response time analysis and in the process, we were introducing this idea of busy period and how the busy period can be computed.

(Refer Slide Time: 00:47)



And we then say that well, for finding out a response time for a specific message what you need to do is, inside the busy period you have to figure out, how many instances of this message has really been transmitted. So, let us say the busy period is t mi for this message. That means, it has been blocked initially and then there has been multiple transmissions by other higher priority messages and eventually all the release instances of this particular message got transmitted, right.

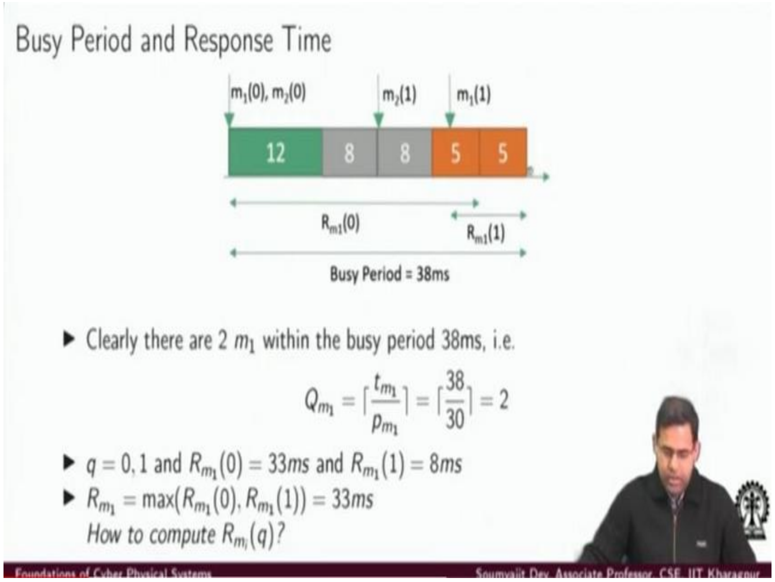
So, that has been our definition of the busy period. Now, so, if this length is greater than the period of the message then, I would be having multiple instances of the message and I would have to then figure out well if there are few instances, let us say from 0 to Q_{mi}–1, Q_{mi} is computed like this.

$$Q_{mi} = \left\lceil \frac{t_{mi}}{p_{mi}} \right\rceil$$

Then for all these Q instances 0 to Q -, Q_{mi}-1 number of instances, I will have to figure out what is the worst-case response time and what is the response time and eventually the worst case would be the maximum of this.

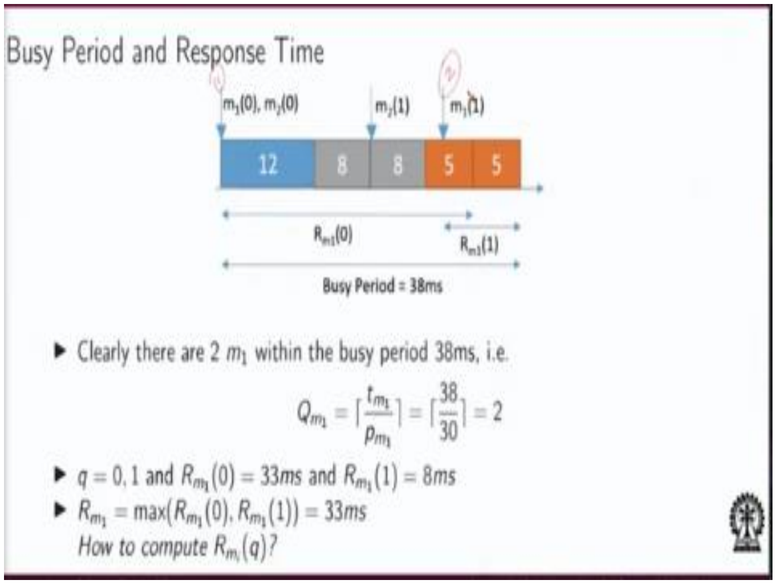
$$R_{mi} = \max_{q=0,\ldots,Q_{mi}-1} R_{mi}(q)$$

(Refer Slide Time: 01:55)



So, that is how we go forward and do this. So, for our example here, we figured out, this is the length of my busy period, right. For this particular message and inside this busy period, m_1 has got two release instances, one being here and the other being here, right. And, so this was m_1 first instance and this was m_1 second instance here.

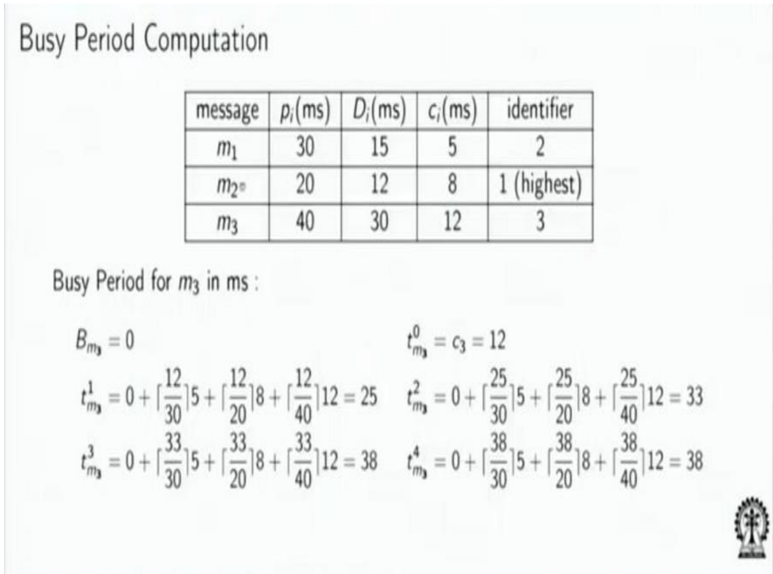
(Refer Slide Time: 02:30)



So, inside this overall busy period of 38 milliseconds that we calculated for the first instance as you can see. I mean, the first instance has the release here and the transmission ends here, right. So, the busy period is given by this value, which is like if you can see. It should be 12, 8, 8, so that 28 and 5 is the execution time of m_1, the transmission time of m_1 so that 33. And for the second messages response time. Again, inside the busy period we have to see when it arrives. It arrives here, right.

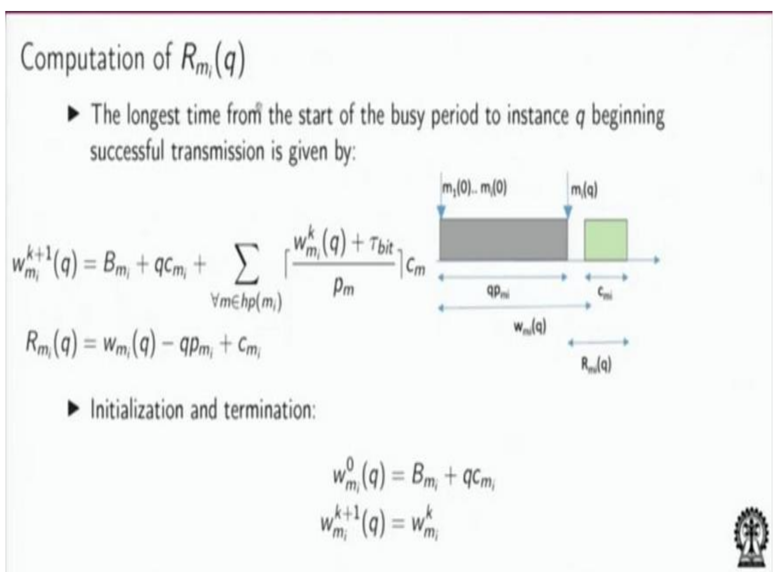
And it has a period, I believe, of 30, right, and it has to wait for some time, I mean, this is 3 milliseconds and then it will be executing for 5 milliseconds, right. So, that would have a response time of 8, right. So, if we want to check let us just go back and see the task setting here, yes.

(Refer Slide Time: 03:31)



So, m1 has the periodicity 30 itself, so as we can, and m1 as a transmission time of 5. So that means this indeed comes at 30 and the busy period length is 38, so definitely the transmission time is 8 milliseconds here, okay. Now definitely the max would be 33, so that is the worst-case response time. Now the question is, how do I compute this response time for each of these messages? Because here, we are doing it just by looking at the picture, the schedule on the busy period. But how do I do it computationally?

(Refer Slide Time: 04:04)



Well, we will have a similar kind of recurrence equation for that. So, what we will do is we will figure out the longest time from the start of a busy period to the instance q. So, let us say we are looking for some q^{th} instance of message mi and we are trying to figure out its response time. So, the first thing I will check out is, what is the time starting from the initial, I mean, from the start of the busy period up to the situation, when this q^{th} an instance of mi has started transmission.

So, let us call this thing, is for q^{th} instance, 0, 1, 2, up to q. So, for this many to start what is the duration. So, let us denote it by $w_{mi}(q)$. q^{th} instance of message mi start of successful transmission. Now how can I figure it out. Well, it would be blocked initially by the blocking time for this particular message mi and then we would have, if you see this is the q^{th} instance and the indexing in this notation here is starting from 0 up to q.

So, before q^{th} instance we already have q number of messages with index 0 to q -1. So, for this, q-1 messages you have this much of transmission time taken up, right. So, you have an initial blocking time, at the time when this was getting blocked by lower priority messages, you have, I mean inside this space what are the things, we are having the blocking time and when the previous instances of these messages themselves got transmitted there are q such instances 0 up to q-1.

And then you have the interference time. So, the way I will calculate the interference time is I will see what are the higher priority messages and I have a previous estimate of this quantity. Add with it tau bit, which is the transmission time of 1 bit, assuming this is where the first bit of this particular message starts transmission. And you are dividing this by this message's periodicity. So, for whichever message has got the higher priority.

So, let us say you have multiple messages we have got higher priority than mi, right. So, let us pick one of them, so you divide this previous estimate plus tau bit this small delay divided by the period of this message. So, that would give you how many instances of this higher priority message is kind of interfering with this and those instances definitely will be transmitted earlier. So, you have to multiply this with the transmission time of each of those messages for transmission time of those message instances.

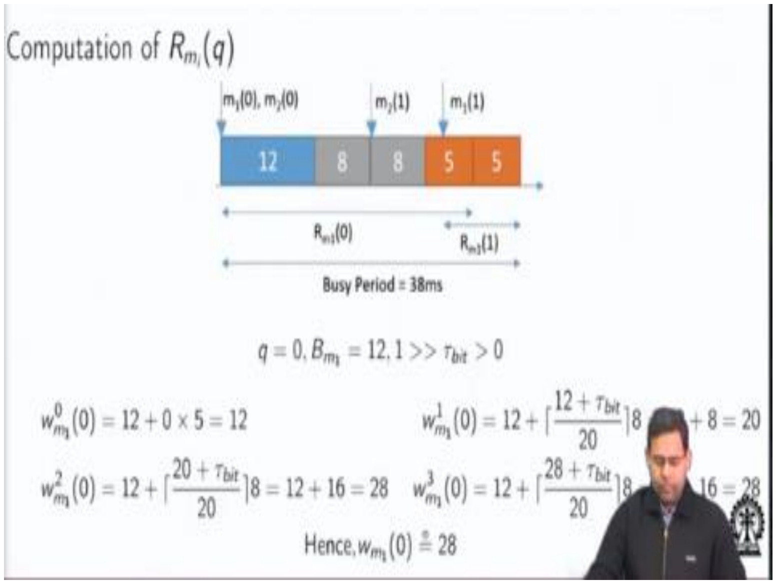
And then you have to similarly sum it up for all such other high priority messages. So, that is how you do it and this gives you the time starting from the time, I mean, from the start of the busy period up to this time, when this particular message, this q^{th} instance of mi starts its transmission. So, you can compute it like this. Now this is not really what we are looking for, right. We are looking for the response time. That means, if I go by response time this is the definition.

It is the time, so this is basically a star this current estimate is giving me when this message actually starts the transmission. So, this message will start the transmission here and if you add the transmission time, right. So, if you add this quantity with the transmission time, then you get the time value when this message transmission is successfully finished, right. But we want to subtract from it, the time when this message actually arrived into the system, right. Because we are calculating response time.

The time when the message actually arrived in the system, the q^{th} instance is basically given by the period of mi multiplied by q. So, you just subtract this q times the period of mi, right, and that would give you this response time of q. So, now you can just like our previous computations you can keep on solving this recurrence multiple times until it stabilizes and you get the overall waiting time, right. And then, you use this waiting time in this equation to get the response time.

So, that is how you can simply do it. So, you can start your initial iteration with this being 0 and you are having this B, the blocking time plus qc_{mi} and then eventually you will stop when under two successive iterations these values are the same.

(Refer Slide Time: 09:06)

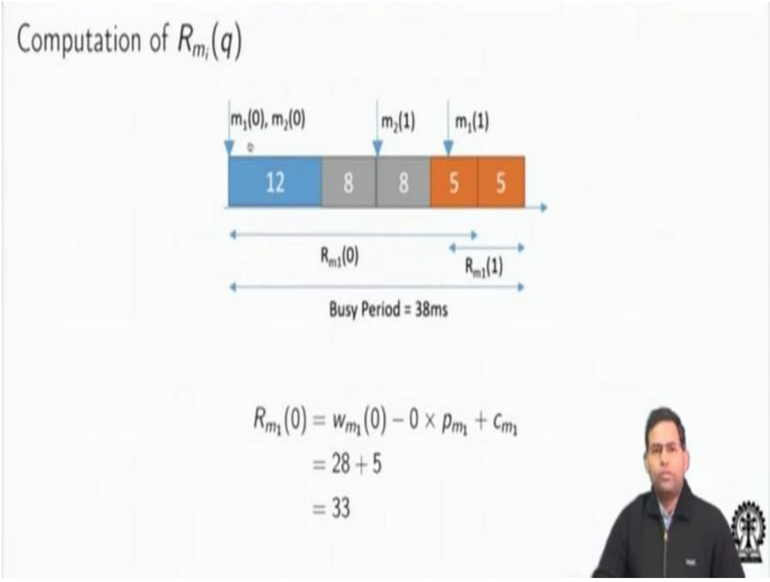


So, if you can see this example let us say we are talking about message zero, right, message zero has a blocking time 0 and so you have this as 12, and then you are looking really for the 0th instance of message 1, right not message zero. This message is 1's 0th instance. So, that is 0 multiplied by 5, then basically is just the initial blocking due to the other lowest priority message right. And then you can compute w_1 so that is 12 and then you can compute in the next iteration you can use this here.

This is the busy period's current estimate. This w's current estimate. τ_{bit}, we are always neglecting tau bit here for all practical purposes we will just neglect it unless each other sees bigger value, okay. So, we will just neglect it and we will have this, and this message is transmission time is 8. And the periodicity is 20. So, this is about the message which has the higher priority with respect to m_1.

So, that is m_2. So, you are giving the interference due to m_2, and you have the wait time estimate to be defined to 20, and then it gets further defined to 28, and it stabilizes at 28. So, this waiting time is 28 here.

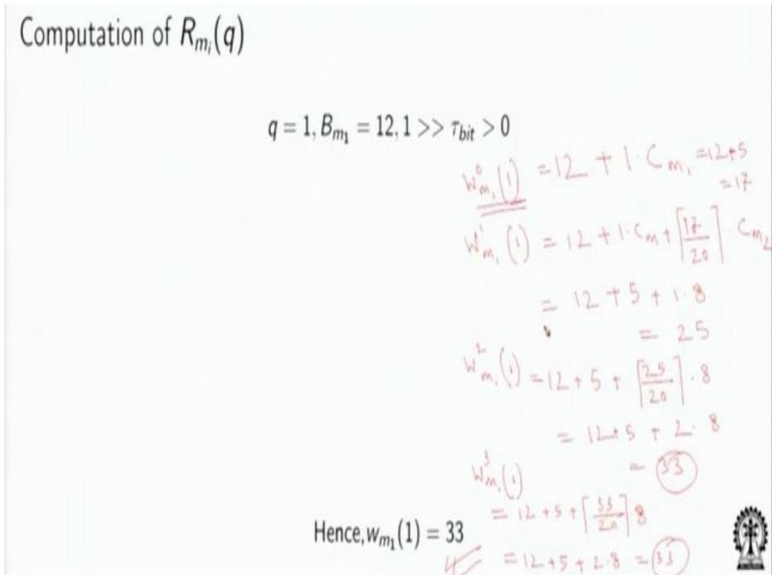
(Refer Slide Time: 10:34)



Then you can use 28 for the 0^{th} instance of m_1. So, basically this message, right and as you can see for this you can calculate the response time like this 28 + 5, which is the actual transmission time. So, that would give you 33. And if you check, this checks out, because the arrival is at 0 so nothing really to subtract. So, and this interval is really 28 + 5. So, then this is the wait time, up to the time when this really starts this transmission and before that you see, you had 2 iterations of this equation, right.

I mean, that is really happening because there are 2, twice you have to factor in the interference due to this message instance of m_2. You have message instance of m_2 coming right here 1. So, you have this interference, another is coming and that is kind of getting added up through these 2 iterations of these equations, and then it stabilizes.

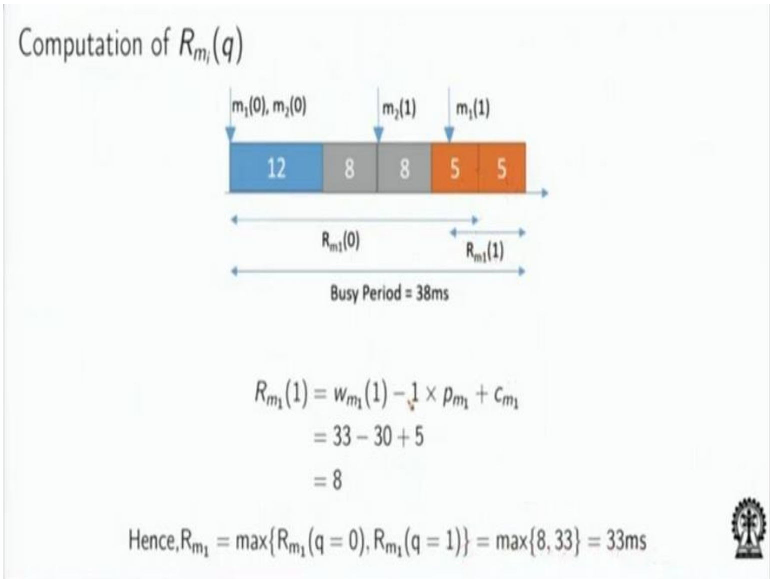
(Refer Slide Time: 11:43)



So, now similarly if we are trying to do it for the next iteration. So, let us if you see here, we talked about 2 messages, 2 different instances of m1, right. So, we computed here for the 0^{th} instance but you also have the first instance here, right. So, if I try to compute this weight time for the first instance. So, the usual blocking time plus now you have 1 instance earlier, so you will multiply its execution time here. So, the transmission time for this message is 5, and for m_2 that is 8, right.

Now if I just try to define it further. So, you will have another 5 here. You have another 8 here. That is 33. And if you try to refine it further. So, if you just check this out, so you see that now this has stabilized, right, and this is kind of the final answer here. Because eventually it is now always coming to be the same expression. So, we can just stop here.

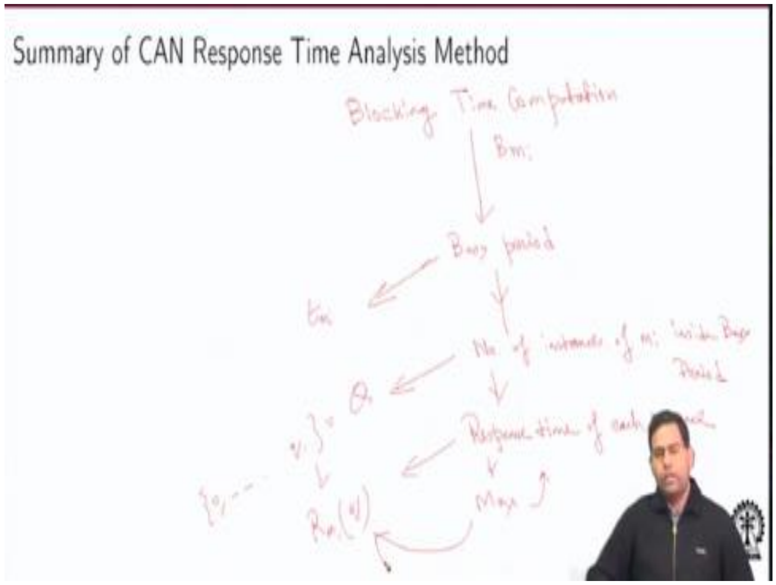
(Refer Slide Time: 15:02)



And now you have got one idea, like well, from my previous calculation for the 0^{th} instance of the message m_1, we computed just recapping here that this waiting time was 28 and you use the response time equation for this 0th message instance and you got that entire response time as 33 here, right. And we are now trying to calculate for the first instance, and what we saw is for the first instance, the response, the wait time was 33 and then you apply the response time equation for the first instance.

So, 33 minus the arrival time when it came, there is 30 plus 5 that is 8. So, if you see this equational method that we discussed, that is kind of agreeing with how we analyze the busy period and response time from the figure, and now you can just apply our previous definition of worst-case response time and calculate and infer that the worst-case response time would be 33 here. So, that is how we calculate the worst-case response time of messages in the bus, in case of any fixed priority non-preemptive scheduling algorithm, which we took as an example as CAN. But it should hold in general as you can understand.

(Refer Slide Time: 16:25)



So, if I just quickly summarize what had been this response time analysis method, what we started with was blocking time computation and for some message mi, it gives me this B_{mi}, right. And then using that we computed busy period and then using busy period we computed number of instances of mi inside this busy period. And then what we did is we calculated the response time of each instance, and then we took the max of the previous calculation, right.

So, from B_{mi} we calculated the busy period using that recurrence equation and we called we got this time for the busy period, and then we computed the q number of instances, and that was containing this 0 to up to some q elements whatever, and for each of these elements here we computed the response times, right. And then you apply the max over them, and that kind of gives you the workspace response time here, right.

So, this is how overall you are given, if you are given a CAN or some other bus space system it may be a TDMA or this kind of a fixed priority non-preemptive, I mean, bus scheduling protocol. And they are connected with multiple compute units where these individual computer control units, whatever you call them, they are contain messages, the tasks which have precedences. And they are scheduled with respect to some scheduling algorithm. How you schedule the tasks in the processors and how you schedule the tasks in the bus.

So, that is overall our coverage of how real time scheduling works in general and in our later lectures we will see that how this effects (1915) cyber physical system designs. For any kind of such design timing analysis as you can understand would have been a very important criteria. And these are some of the basic techniques of timing analysis which are applied in the industry and well, if you are trying to do research in this area in both places. So, with this we will end this lecture. Thanks for your attention.